General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

TECHNICAL MEMORANDUM

TIME SERIES ANALYSIS PACKAGE (TSAP)
FOR
TERMINAL USE

Bellcomm

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W., WASHINGTON, D.C. 20024

COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE-Time Series Analysis Package (TSAP) for Terminal Use

FILING CASE NO(S)- 103-7, 630

FILING SUBJECT(S)
(ASSIGNED BY AUTHOR(S))- Time Series Analysis
Remote Terminal

TM- 70-1031-3

DATE- December 31, 1970

AUTHOR(S)- G. R. Andersen
R. R. Singers

ABSTRACT

The time series analysis package (TSAP) presented here in the form of a user's manual, is a set of eleven interactive, functionally dependent time series programs which are accessible in any order from a remote terminal. TSAP has been designed so that an analyst can conduct a basic portion of the statistical calculations involved in multiple time series analysis during one session at a terminal. This allows the analyst to avoid the waiting times between outputs that occur with standard batch processing. In addition, it takes advantage of the sequential nature of the statistical computations involved in the analysis of time series.

TECHNICAL MEMORANDUM

INTRODUCTION

The result of any particular statistical computation
produces two types of information, one type leading to a conclu-
sion and the other suggesting a particular direction for further
analysis. With respect to the analysis of time series, the
analyst usually has a general idea of the avenues of investigation
likely to provide a broad preliminary understanding of the particular
problem, e.g., how two series are statistically related or what is
the statistical mechanism generating a single series. Since the
direction of the investigation after the completion of any particular
segment of the computations depends on the results of that segment,
he must view the results of a large number of intermediate steps of
the analysis before he can reach a point where a detailed evaluation
and summary of the data can begin. Unfortunately, each segment
usually requires at least one run on a computer and, more often
than not, several hours elapse before the results can be viewed.
On the other hand, the decision concerning the next step of the
analysis can usually be made immediately, say in the length of time
it would take the results to print out at a terminal.

TSAP, the terminal time series analysis package presented
here, is designed to eliminate this miss-match between the computer
and analyst by furnishing him with a set of interactive, function-
ally dependent time series programs; these programs are accessible
in any order from a remote terminal.

The particular set of programs which are presently
available in TSAP are necessary for the first steps in the analysis
of any time series data. Further reduction of such data may require
programs that are best kept in a batch mode (e.g., the method using
the cross-spectral information of several series to improve the
least squares estimates of the coefficients in a multiple time
series regression). More programs will be added as they prove
useful or are required in the solution of a particular problem.
For this reason TSAP is provided with an index (cf., Appendix III)
that may be consulted to determine what programs are available.
The Bellcomm Univac 1108 implementation of TSAP is in file
RRS*TSAP.

Descriptions of the TSAP programs are given in section 1 together with a brief nontechnical explanation of their use or meaning. This section also contains a diagram indicating the principal ways the programs in TSAP may be combined. The way in which data files are generated and used is described in Section 2.

Section 3 contains a series of examples (pages, 10, 22 and 30) that illustrate most of the present capabilities of TSAP and is intended as a users guide.[1]

Section 4 contains a discussion of program techniques and the flow charts of each of the programs; the listings are given in Appendix III. Some elementary remarks concerning time series together with the computational formulae used in the various programs are given in Appendix I. Details concerning the various filters available in TSAP are given in Appendix II.

1.0 PROGRAMS

This section contains a list of the programs in TSAP together with a brief description of their use.

(1) Data Transformations and Filters (FILTER)

The purpose of this program is to allow the user to prepare the data for time series analysis. For example, this might include transforming the series into the logarithms of its absolute values and then centering the resulting series at its mean in an effort to obtain a series that approximates a zero mean Gaussian time series. A series could be pre-whitened for spectral estimation by applying an appropriate high-pass filter; alternately, a trend in the series could be estimated or a cycle removed by using one or more of the available filters. The program is designed so that any subset of its transformations or filters may be applied sequentially to a particular time series with a single request for the element labeled FILTER. The transformations that are available in this program are listed when the user calls @filter; details concerning these transformations are given in Appendix II.

(2) Auto-Covariance and Auto-Correlation (ACF)

The sample auto-correlation function calculated with this program is a measure of the "memory" or inertia of the process generating the time series; this information can be used to some extent in model discrimination. The primary purpose of the auto-covariance function is in calculating the spectrum of the series.

---

[1]Although some remarks concerning statistics have been made to prevent misunderstanding and to motivate the use of a particular sequence of programs, this section is not meant to be tutorial in statistics; it assumes an elementary knowledge of this area.

(3) Partial-Auto-Correlation Function (PACF)

This function describes the correlation between two distinct members of a time series with the linear affect of intervening members removed. The program allows the user to calculate (up to) the first sixty values of the partial-auto-correlation function. It also computes the residual variance of auto-regression (a.r.) models,

$$y(t) = \alpha_0 + \alpha_1 \, y(t-1) + \ldots + \alpha_p \, y(t-p) + \varepsilon(t) \quad ,$$

up to order p = 60. On the basis of these results the user may determine the order of the best fitting a.r. model. Once the order is furnished, the program will print out the coefficients of the corresponding a.r. process. The appropriateness of the model may be judged by submitting the residuals of the fitted a.r. model to ACF and ASF.

(4) Auto-Spectrum (ASF)

An estimate of the auto-spectrum of a stationary time series is calculated and plotted at the terminal; if requested, its upper and lower 90% confidence limits are superimposed on the spectrum plot. One of the main purposes of this program is to determine which frequency bands contribute significantly to the total variance of the series. Another is in suggesting and testing the appropriateness of parametric models for the series being studied.

(5) Cross-Covariance (XCF)

This program is used primarily as data for the cross-spectral routine.

(6) Cross-Spectrum (XSF)

This program calculates estimates of the co-spectrum and the quadrature spectrum of pairs of (jointly stationary) time series. These quantities measure the covariance between the "in-phase" and the "$\pi/2$ out-of-phase" frequency components of the two series, respectively. The co- and quadrature spectrums, together with results from ASF, are used to calculate the coherence and phase of the pair of series. The coherence is a measure of linear relationships between frequency components of the series, while the phase measures lead and lag relationships. The arc-tanh of the coherence is plotted to obtain confidence bands which are independent of frequency.

(7)   <u>Least Squares Regression</u> (LSRA and LSRP)

     Least squares regression is included in the system for
several purposes. It provides a method of estimating
the trend component of a series, furnishes the standard
errors of auto-regression coefficients (the order of the
a.r. is determined with PACF) and gives preliminary esti-
mates of the coefficients in a multiple regression on
time series.

     Two separate least squares regression programs[1] are
included in TSAP: LSRA and LSRP. LSRA is a forward
stepwise regression program that includes an analysis
of variance table at each step together with a table of
observed values, expected values and percent deviation,
printed out at the request of the user. LSRP is a back-
ward regression program whose main feature is that
it yields predicted values with upper and lower predic-
tion limits.

(8)   <u>Save</u>

     After an application of the program FILTER the user is
asked whether or not he wants the transformed series
stored. If the answer is yes and the data format is
specified, then the series is automatically stored, but
the user must supply the statement @SAVE FILTER,[series
name] to be able to reference the series for later use.[2]
Similar statements are required after executing the LSRA,
LSRP or COMB programs (e.g., @SAVE LSRA,[series name]).

(9)   <u>Combination of Series</u> (COMB)

     Given two time series of the same length, this program
enables the user to perform term by term addition (ADD),
subtraction (SUB), multiplication (MPY) or division (DIV)
of the two series. COMB also allows one to concatenate
(CON) series of arbitrary length.

(10)   <u>MATH Conversion</u> (MATHCON)

     In some applications, the user requires a series of values
of some known or constructed function and a way to easily
combine this series algebraically with an observed series

---

    [1] Both of these programs are modifications of regression
programs available in the UNIVAC Stat-pack program library.

    [2] This control statement stores the series in the TPF; if the
user has a catalogued file he may store the series in this file by
typing @SAVE FILTER,[FILE.SERIES NAME].

being analyzed with TSAP. This can be accomplished without
leaving the terminal by generating the required series with
MATH and combining it in TSAP (using COMB) with the observed
series. MATHCON eliminates the incompatibilities in I/O[1]
between the two series.

Additional programs that are available for batch runs, but
have not been included in TSAP at the present time include

(A) Time Series Regression (TSR) [2]

This program combines the preliminary least squares estimates
in a multiple regression on time series with cross-spectral
information between series to produce asymptotically efficient
estimates of the regression coefficients.

(B) Histogram and Scatter Plot Programs (HIST, SCAT) [2]

Work is in progress on the following programs: a means of
estimating mixed regression, auto-regression and moving average models
and a program for forecasting with such models.

The efficiency and effectiveness of TSAP would be enhanced
with the addition of CRT (Cathode Ray Tube) display capabilities to
allow more outputs. This would provide the user with a more compre-
hensive system containing programs dealing with graphics.

Another possible improvement which will be looked into
concerns the formatting structure. At present each input or output
series requires a format; this reduces data format conversions at
the expense of answering questions. An alternative method would be
to have a fixed format, say G20.10, in which all data is read and
written. This would reduce the number of user responses but also
reduce the generality of the input data structures. A way around
this is to add an option, say F, to allow the present dialogue to be
used.

The following diagram indicates the ways in which the
TSAP programs may be linked together in applications. These will be
further described by means of the examples given in Section 3.

---

[1] MATH "SAVED" elements are in fixed point and floating point
notation, using *10**n instead of the FORTRAN E+n. A format of
(G20.10) will suffice to read the converted series.

[2] TSR will be documented in the near future; HIST is a
corrected UNIVAC Stat-pack routine and SCAT is obtainable from
R. R. Singers.

TSAP
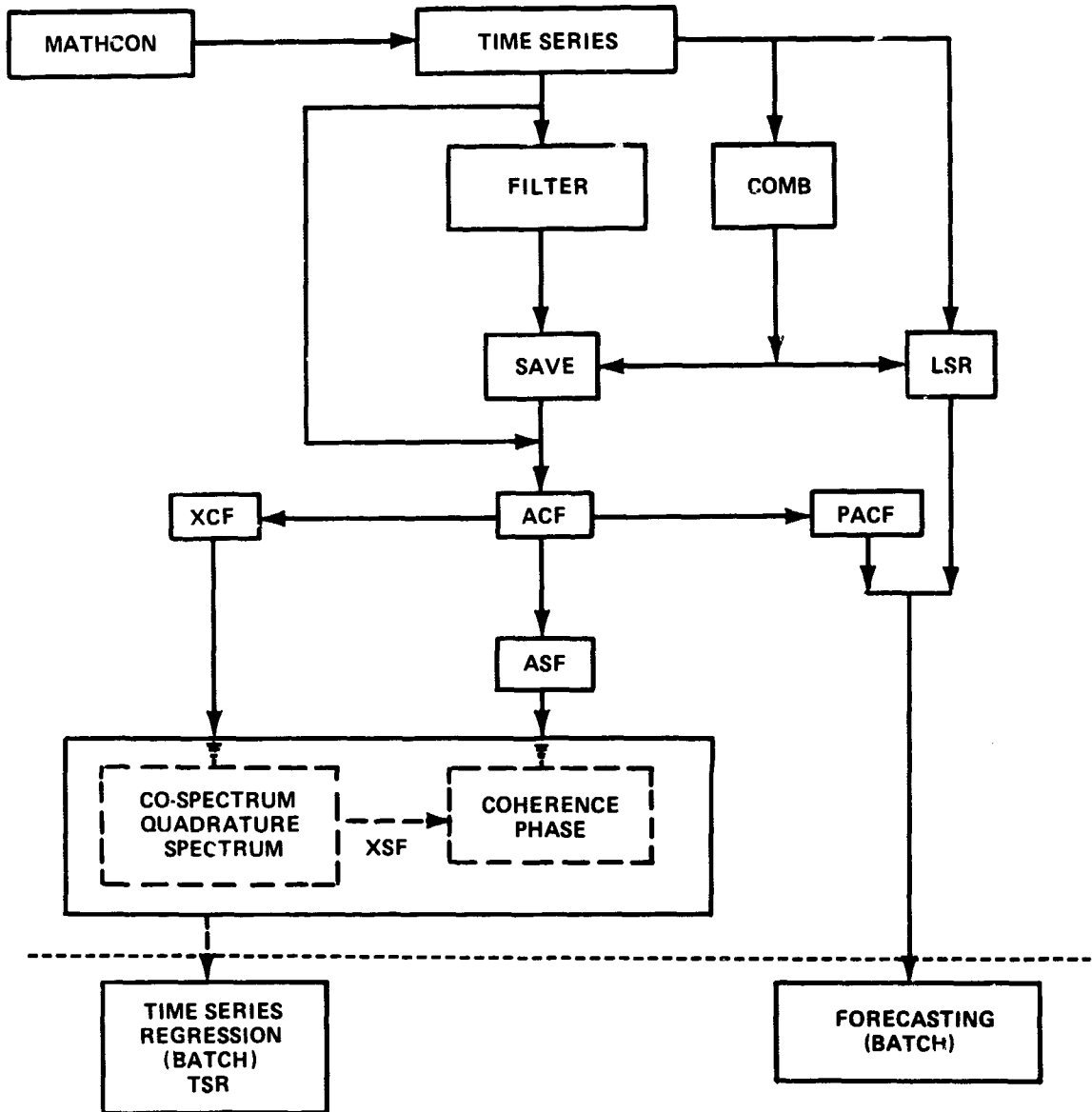
FIGURE 1

2.0   GENERATION AND USE OF STORED DATA ELEMENTS

The first step in applying TSAP to a particular set of time series data is to make this data accessible from the terminal. This can be accomplished by using the EXEC 8 control statements @ELT and @ADD. The control statement ELT allows the user to place the data into the machine, via the card reader or the terminal, with a name attached to it. This name together with the @ADD statement makes the data available for use in TSAP.

The ELT processor allows one to generate a data element in a file using the following control card (starting in column 1):

@ELT,I˯QUAL*FILE.ELEMENT

where ˯ indicates one or mare spaces.

QUAL*FILE. is the name of the catalogued file[1] into which the user wants the data stored. This field may be left blank, in which case the user temporary file, TPF$, is used.

ELEMENT is the name of the data series. The data follows immediately after this card and the element is terminated when the next control card is read.

When the data is asked for within TSAP, the element can be obtained by using the following control card (starting in column 1):

@ADD˯QUAL*FILE.ELEMENT

The following are fo˯r examples of ELT processor use:

1) For external card deck

    (a)   without previously catalogued file

        1.   @RUN,/NR˯INIT,ID,PROJECT,15,150

        2.   @ASG,CP˯QUAL*FILE.,F2

        3.   @ELT,IL˯QUAL*FILE.ELEMENT

        4.   (Data cards)

        5.   @FIN

---

[1]See [14] for definition of catalogued file.

(b)   with previously catalogued file

    1.   @RUN,/NR⌃INIT,ID,PROJECT,15,150

    2.   @ASG,A⌃QUAL*FILE.

    3.   @ELT,IL QUAL*FILE.ELEMENT

    4.   (Data cards)

    5.   @FIN

2) For terminal input

  (a)   without previously catalogued file

    1.   @RUN⌃INIT,ID,PROJECT,70

    2.   @ASG,CP⌃QUAL*FILE.,F2

    3.   @ELT,I⌃QUAL*FILE.ELEMENT

    4.   (Type in data)

  (b)   with previously catalogued file

    1.   @RUN⌃INIT,ID,PROJECT,70

    2.   @ASG,A⌃QUAL*FILE.

    3.   @ELT,I⌃QUAL*FILE.ELEMENT

    4.   (Type in data)

I/O. [1]   It is necessary for all data to be in FORTRAN formatted

----

[1] For information on formats see [13].

3.0   USING TSAP

The mechanics of using TSAP will now be explained by means of several examples. These examples fall into essentially three groups and begin on pages 10, 22 and 30.

The first group involves a single time series and uses that portion of TSAP indicated in Figure 2. The first example within this group illustrates spectral estimation using a Tukey window at two different bandwidths. These spectra together with their 90% confidence bands are plotted against frequency in cycles per unit time. The spectrum of the time series is then estimated by means of both the Daniell and Tukey windows and plotted against period with no confidence bands. The second and third examples within the first group illustrate two types of Goodman filters applied to the series and the spectra of the resulting series.

To aid the reader, an arrow has been added to the left margin of the terminal printout each time a new request is made for a TSAP program.

Finally, it should be remembered that each group of examples was executed in one session at a remote terminal; depending on the user, the amount of his time required for computations similar to those in one of these groups would be between 10 and 30 minutes.

```
┌────────┐     ┌────────┐     ┌────────┐     ┌────────┐
│ FILTER │ ──▶ │  SAVE  │ ──▶ │  ACF   │ ──▶ │  ASF   │
└────────┘     └────────┘     └────────┘     └────────┘
```

FIGURE 2

We will assume that the series has been stored in the machine, as explained in the previous section, under the name y10e; it contains 362 terms.

In order to activate the terminal, assign the file and copy it over into a temporary file we type the terminal number, say

BTL1,

followed by

```
    @RUN INITIALS,ACCOUNT,PROJECT,MAXTIME,MAXPAGES

    @ASG,AX RRS*TSAP.

    @COPY,P RRS*TSAP.,TPF$.
```

The first program called is @filter[1]:

@filter

```
'FILTER' CONTAINS THE FOLLOWING FILTERS:
CENTER,EXP,LOG,DIFF,10**,LOG10,SQRT,LAG,ABS,TUKEY(HIGH AND LOW
PASS),GOODMAN,SCALE.
TO TERMINATE THE FILTER SEQUENCE,REPLY WITH 'END'.
THE FINAL SERIES IS STORED ON UNIT 8(IF DESIRED).
LENGTH OF SERIES TO BE READ IN?
362                            ◄─┐ The entire series that is stored
  INPUT DATA FORMAT?             │ in the data file must be read in
(4e15.8)                         └ at this point.
  INPUT DATA:
@add y10e
  LENGTH OF SERIES TO BE USED?   ┌ Here we could have deleted any
362                            ◄─┤ number, m, of consecutive terms from
  MEAN=    .37433816+01          └ the end of the series by typing 362-m.
  LENGTH=  362

  FILTER?                        ┌ After each request for a filter the
center                           │ mean of the filtered series together
  MEAN=    .35338186-05        ◄─┤ with its length is automatically
  LENGTH=  362                   └ printed out.

  FILTER?
end
  DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
no
  DO YOU WANT THE FINAL SERIES STORED(YES/NO)?
yes
  OUTPUT DATA FORMAT?
(4e15.8)
  SERIES STORED.
  FILTER TERMINATES.
@save filter, y1000
```

---

[1] The underlined quantities are the users responses to the preceeding program questions. The italicized type indicates explanatory comments that have been placed onto the terminal printout to aid the reader of this memorandum.

The name of the yl0e series after the above transformations is yl000. The auto-correlation function of yl000 is calculated next:

→ @acf

  ACF COMPUTES THE AUTO-COVARIANCE/CORRELATION FUNCTION OF A SERIES.
  THE COVARIANCES ARE STORED ON UNIT 1 IN BINARY(IF DESIRED)
  IN CONSECUTIVE ORDER.

  DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
cor
  LENGTH OF SERIES?
362
  NUMBER OF LAGS(ZERO INCLUDED)?
41
  INPUT DATA FORMAT?
(4e15.8)
  INPUT DATA:
@add yl000

0	.10000000+01	13	.18489365-01	26	.11289490+00
1	.46415033+00	14	.11077084+00	27	.15901410+00
2	.17615860+00	15	.17462706+00	28	.11500745+00
3	.49235427-01	16	.35404109-01	29	.93957893-01
4	-.50011224-01	17	-.16164341-01	30	.97580075-01
5	-.26090787-01	18	.35023679-03	31	.59549038-01
6	-.19535766-01	19	.60941407-01	32	-.80215645-02
7	-.31009051-01	20	.61706792-01	33	-.28728484-01
8	-.47952917-01	21	.17506226-01	34	.53134954-01
9	.23769772-02	22	-.79266704-01	35	.18498416-01
10	.21561115-01	23	-.79293265-01	36	.18898459-01
11	.10452646-02	24	.13103552-01	37	-.31084192-02
12	.25509858-02	25	.82080608-01	38	.88336019-01
				39	.14070644+00
				40	.96740634-01

  VARIANCE=   .34657791+00
  MEAN=   .35338186-05
  DO YOU WANT THE COVARIANCES STORED(YES/NO)?
yes
  COVARIANCE STORED.

  DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
end
  ACF TERMINATES.

→ @asf

  ASF COMPUTES THE AUTO SPECTRUM FUNCTION.
  IT IS ASSUMED THAT ACF WAS CALLED AND THE DESIRED COVARIANCES STORED.
  THE AUTO-SPECTRA ARE STORED ON UNIT 2(IF DESIRED) CONSECUTIVELY.
  DATA SPACING?
1.
  WINDOW TO BE USED(TUKEY/DANIELL)?

tukey
  BANDWIDTH PARAMETER?
18
  NUMBER OF FREQUENCY POINTS TO BE VIEWED?
33
  DO YOU WANT CONFIDENCE LIMITS(YES/NO)?
yes
  DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
freq.
  AUTO-SPECTRUM:

FREQ.	SPECTRUM
.0000	.74730+00
.0156	.74350+00
.0313	.74541+00
.0469	.76602+00
.0625	.78415+00
.0781	.76448+00
.0938	.70337+00
.1094	.62974+00
.1250	.56432+00
.1406	.50291+00
.1563	.44168+00
.1719	.39207+00
.1875	.35903+00
.2031	.32513+00
.2188	.27371+00
.2344	.21620+00
.2500	.17947+00
.2656	.17243+00
.2813	.18018+00
.2969	.18700+00
.3125	.18969+00
.3281	.18850+00
.3438	.18017+00
.3594	.16534+00
.3750	.15253+00
.3906	.14782+00
.4063	.14651+00
.4219	.14004+00
.4375	.12689+00
.4531	.11207+00
.4688	.99940-01
.4844	.91983-01
.5000	.89126-01
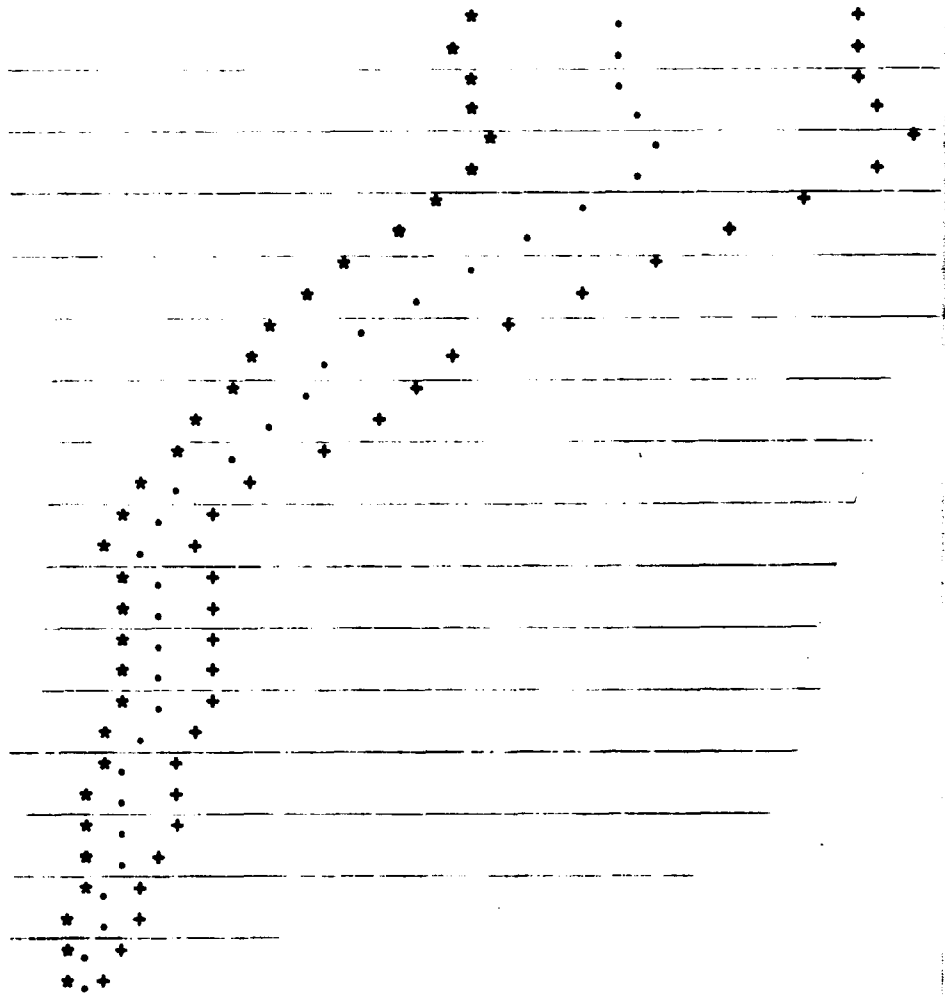
THE 90% CONFIDENCE LIMITS FOR NU=57[1] ARE: CU= 1.39722 , CL= .75097

---

[1] NU, the number of degrees of freedom of the spectral estimator, is defined in Appendix I.

```
THE SQRT OF VARIANCE RATIO IS:  19.311%
DO YOU WANT THE SPECTRUM STORED(YES/NO)?
no
DO YOU HAVE ANOTHER SERIES(YES/NO)?
yes
IS THE SERIES THE SAME(YES/NO)?
yes (1)
DATA SPACING?
1.
WINDOW TO BE USED(TUKEY/DANIELL)?
tukey
BANDWIDTH PARAMETER?
33
NUMBER OF FREQUENCY POINTS TO BE VIEWED?
33
DO YOU WANT CONFIDENCE LIMITS(YES/NO)?
yes
DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
freq.
```

---

[1] This illustrates the usual method of calculating the spectrum at successively smaller bandwidths ("window closing") in order to determine the structure of the true but unknown spectrum.

AUTO-SPECTRUM:

FREQ.	SPECTRUM
.0000	.88953+00
.0156	.73221+00
.0313	.62610+00
.0469	.73374+00
.0625	.90672+00
.0781	.84932+00
.0938	.62948+00
.1094	.59679+00
.1250	.56681+00
.1406	.56516+00
.1563	.45911+00
.1719	.26265+00
.1875	.35751+00
.2031	.45512+00
.2188	.27104+00
.2344	.12386+00
.2500	.15475+00
.2656	.20220+00
.2813	.19860+00
.2969	.16452+00
.3125	.16827+00
.3281	.22187+00
.3438	.20527+00
.3594	.14452+00
.3750	.12858+00
.3906	.14862+00
.4063	.16991+00
.4219	.14789+00
.4375	.11107+00
.4531	.10553+00
.4688	.11495+00
.4844	.91251-01
.5000	.64649-01

THE 90% CONFIDENCE LIMITS FOR NU=30 ARE: CU=   1.63145 ,CL=   .68925
THE SQRT OF VARIANCE RATIO IS: 26.148 %
DO YOU WANT THE SPECTRUM STORED(YES/NO)?
no

In this last example we see that the square root of the variance ratio is about 26%; that is, fractional errors of about 26% in either direction can be expected. It was decided that with this length of series it would be difficult to obtain both high resolution (small bandwidth) and reliability. However, suppose for illustration that a bandwidth parameter of 33 gives us the proper resolution and then compare the spectral estimate obtained using the Daniell window with that obtained using the Tukey window.

→ @asf,n [1]
  DATA SPACING?
  1.
  WINDOW TO BE USED(TUKEY/DANIELL)? [2]
  daniell
  BANDWIDTH PARAMETER?
  33
  NUMBER OF COVARIANCES TO BE USED?
  362
  NUMBER OF FREQUENCY POINTS TO BE VIEWED?
  33
  DO YOU WANT CONFIDENCE LIMITS(YES/NO)?
  no
  DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
  period

---

[1] The "n" option is supplied to suppress the initial commentary for the experienced user.

[2] If there are N terms in the series, then to employ the Daniell window, N covariances (including the variance) must be calculated in ACF and used in responding to the 4th question in ASF. If any number of covariances less than this number are either calculated or used, a request for the Daniell window will yield a truncated Daniell window.

AUTO-SPECTRUM:

PERIOD	SPECTRUM
99.9999	.88676+00
64.0000	.75138+00
32.0000	.63410+00
21.3333	.69779+00
16.0000	.10041+01
12.8000	.85815+00
10.6667	.50042+00
9.1429	.57420+00
8.0000	.59914+00
7.1111	.62584+00
6.4000	.43762+00
5.8182	.23736+00
5.3333	.43552+00
4.9231	.45420+00
4.5714	.21520+00
4.2667	.10984+00
4.0000	.16376+00
3.7647	.19306+00
3.5556	.19815+00
3.3684	.16836+00
3.2000	.13186+00
3.0476	.24439+00
2.9091	.23136+00
2.7826	.14539+00
2.6667	.13965+00
2.5600	.12276+00
2.4615	.16503+00
2.3704	.15110+00
2.2857	.10209+00
2.2069	.11633+00
2.1333	.11029+00
2.0645	.95078-01
2.0000	.67187-01

DO YOU WANT THE SPECTRUM STORED(YES/NO)?

no

DO YOU HAVE ANOTHER SERIES(YES/NO)?

yes

IS THE SERIES THE SAME(YES/NO)?

yes

DATA SPACING?

1.

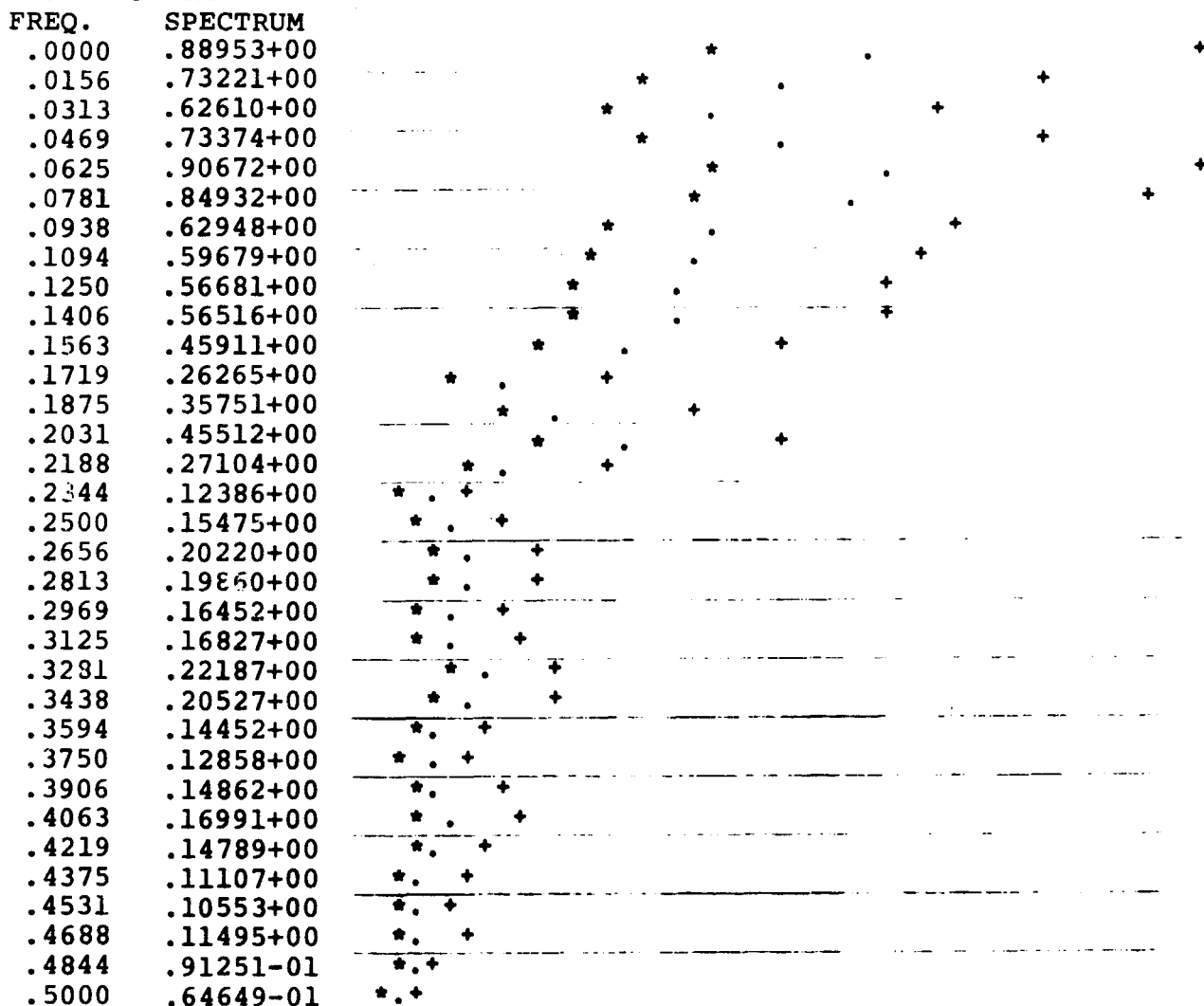WINDOW TO BE USED(TUKEY/DANIELL)?

tukey

BANDWIDTH PARAMETER?

33

NUMBER OF FREQUENCY POINTS TO BE VIEWED?

33

DO YOU WANT CONFIDENCE LIMITS(YES/NO)?

no

DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
period
AUTO-SPECTRUM

PERIOD	SPECTRUM
99.9999	.88953+00
64.0000	.73221+00
32.0000	.62610+00
21.3333	.73374+00
16.0000	.90672+00
12.8000	.84932+00
10.6667	.62948+00
9.1429	.59679+00
8.0000	.56681+00
7.1111	.56516+00
6.4000	.45911+00
5.8182	.26265+00
5.3333	.35751+00
4.9231	.45512+00
4.5714	.27104+00
4.2667	.12386+00
4.0000	.15475+00
3.7647	.20220+00
3.5556	.19860+00
3.3684	.16452+00
3.2000	.16827+00
3.0476	.22287+00
2.9091	.20527+00
2.7826	.14452+00
2.6667	.12858+00
2.5600	.14862+00
2.4615	.16991+00
2.3704	.14789+00
2.2857	.11107+00
2.2069	.10553+00
2.1333	.11495+00
2.0645	.91251-01
2.0000	.64649-01

DO YOU WANT THE SPECTRUM STORED(YES/NO)?
no
DO YOU HAVE ANOTHER SERIES(YES/NO)?
no
ASF TERMINATES.

We now return to the filter program and illustrate the use of the Goodman filters (cf., Appendix III):

@filter,n
LENGTH OF SERIES TO BE READ IN?
362
INPUT DATA FORMAT?

```
(4e15.8)
 INPUT DATA:
@add y1000
 LENGTH OF SERIES TO BE USED?
362
 MEAN=       .35338186-05
 LENGTH=     362

 FILTER?
goodman
 LENGTH OF MOVING AVERAGE(ODD NUMBER)?
41
 NUMBER OF TRIANGLES?
1
 DATA SPACING?
1.
 HIGH PASS(HIGH), LOW PASS(LOW), PASS BAND(PASS) OR REJECT BAND(REJECT)?
reject
 CENTER FREQUENCY?
.074
 MEAN=      -.25713835-01
 LENGTH=     322

 FILTER?
end
 DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
no
 DO YOU WANT THE FINAL SERIES STORED(YES/NO)?
yes
 OUTPUT DATA FORMAT?
 (4e15.8)
 SERIES STORED.
 FILTER TERMINATES.
@save filter, y1000r13
```

Thus, after filtering the series consists of 322 terms;
20 terms off of each end of the series y10b were required to remove
the spectral content at the frequency .074 (cycles/day) or at
13.5 (days). The filter is triangular and frequencies in a
neighborhood, of length .1, centered at .074 are attenuated.
The new series was saved, and its auto covariance function was
calculated and stored. The known affect (Appendix I) of this
filter on the spectrum may be observed by calculating its spectrum:

```
@asf,n
 DATA SPACING?
1.
 WINDOW TO BE USED(TUKEY/DANIELL)?
tukey
 BANDWIDTH PARAMETER?
33
 NUMBER OF FREQUENCY POINTS TO BE VIEWED?
33
 DO YOU WANT CONFIDENCE LIMITS(YES/NO)?
no
 DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
period
```

AUTO-SPECTRUM

PERIOD	SPECTRUM
99.9999	.95803+00
64.0000	.73078+00
32.0000	.46987+00
21.3333	.28421+00
16.0000	.77602-01
12.8000	.29739-01
10.6667	.17326+00
9.1429	.40914+00
8.0000	.53168+00
7.1111	.57579+00
6.4000	.44403+00
5.8182	.25050+00
5.3333	.37526+00
4.9231	.49792+00
4.5714	.30044+00
4.2667	.13079+00
4.0000	.16472+00
3.7647	.22356+00
3.5556	.21855+00
3.3684	.18145+00
3.2000	.18210+00
3.0476	.23152+00
2.9091	.20816+00
2.7826	.14103+00
2.6667	.13133+00
2.5600	.14793+00
2.4615	.16166+00
2.3704	.14852+00
2.2857	.11957+00
2.2069	.11832+00
2.1333	.12439+00
2.0645	.91969-01
2.0000	.63272-01

```
DO YOU WANT THE SPECTRUM STORED(YES/NO)?
no
DO YOU HAVE ANOTHER SERIES(YES/NO)? ·
no
ASF TERMINATES.
```

In order to illustrate the use of a low pass filter and view its affect on the spectrum of the filtered series we return to FILTER and call an adaptation of Goodman's trapezoidal filter (cf., Appendix II):

```
FILTER?
goodman
LENGTH OF MOVING AVERAGE(ODD NUMBER)?
41
NUMBER OF TRIANGLES?
5
HIGH PASS(HIGH), LOW PASS(LOW), PASS BAND(PASS) OR REJECT BAND(REJECT)?
low
DATA SPACING?
1.
CUT-OFF FREQUENCY IS     .2250000
DO YOU WANT TO CHANGE INPUT VALUES(YES/NO)?
no
MEAN=    -.27138643-01
LENGTH=    322

FILTER?
end
DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
no
DO YOU WANT THE FINAL SERIES STORED(YES/NO)?
yes
OUTPUT DATA FORMAT?
(4e15.8)
SERIES STORED.
FILTER TERMINATES.
@save filter, y10low
@acf,n

DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
cor
LENGTH OF SERIES?
322
NUMBER OF LAGS(ZERO INCLUDED)?
33
INPUT DATA FORMAT? ·
(4e15.8)
INPUT DATA:
@add y10low
```

0	.10000000+01	12	-.25210146-01	24	-.73984930-02
1	.74639370+00	13	.52257420-01	25	.10968976+00
2	.24772014+00	14	.16964401+00	26	.16955650+00
3	-.62077395-01	15	.18732763+00	27	.17834130+00
4	-.58770868-01	16	.83261370-01	28	.18380984+00
5	.31103434-01	17	-.17561690-01	29	.19426458+00
6	.13050401-01	18	-.22876046-02	30	.17436985+00
7	-.49668995-01	19	.92481258-01	31	.10661660+00
8	-.21093979-01	20	.12817274+00	32	.32856451-01
9	.69109890-01	21	.42624820-01		
10	.87358192-01	22	-.76860528-01		
11	.14558297-01	23	-.10038714+00		

VARIANCE= .26590697+00
MEAN= -.27138641-01
DO YOU WANT THE COVARIANCES STORED(YES/NO)?
yes
COVARIANCE STORED.

DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
end
ACF TERMINATES.
➡ @asf,n
DATA SPACING?
1.
* WINDOW TO BE USED(TUKEY/DANIELL)?
tukey
BANDWIDTH PARAMETER?
33
NUMBER OF FREQUENCY POINTS TO BE VIEWED?
33
DO YOU WANT CONFIDENCE LIMITS (YES/NO)?
no
DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
freq.

AUTO-SPECTRUM:

FREQ.	SPECTRUM
.0000	.97734+00
.0156	.75601+00
.0313	.56642+00
.0469	.64053+00
.0625	.79873+00
.0781	.76774+00
.0938	.62412+00
.1094	.62409+00
.1250	.58374+00
.1406	.58306+00
.1563	.45604+00
.1719	.25828+00
.1875	.39156+00
.2031	.50674+00
.2188	.29786+00
.2344	.10037+00
.2500	.37529-01
.2656	.94984-02
.2813	.17673-02
.2969	.11131-02
.3125	.14980-02
.3281	.15829-02
.3438	.15418-02
.3594	.12205-02
.3750	.11983-02
.3906	.10520-02
.4063	.11808-02
.4219	.11860-02
.4375	.10052-02
.4531	.98564-03
.4688	.12136-02
.4844	.10427-02
.5000	.88202-03

DO YOU WANT THE SPECTRUM STORED(YES/NO)?

no

DO YOU HAVE ANOTHER SERIES(YES/NO)?

no

ASF TERMINATES.

The second group involves two time series and uses that portion of TSAP shown in Figure 3. This example shows the sequence required to obtain the coherence and phase.
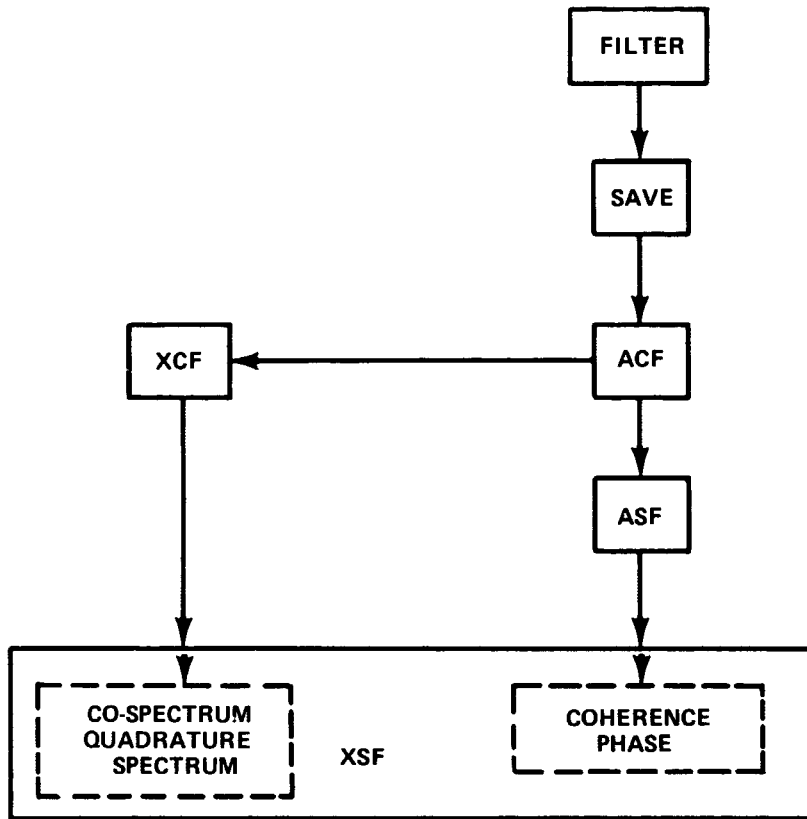
FIGURE 3

Since the filter program has already been demonstrated, we will assume that both series are prepared for spectral analysis and have been saved under the names ybh and sskp[1]. The first program called is ACF:

► @acf,n

 DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
cor
 LENGTH OF SERIES?
362
 NUMBER OF LAGS(ZERO INCLUDED)?
33
 INPUT DATA FORMAT?
(4e15.8)
 INPUT DATA:
@add ybh

---

[1] SSKP is saved in the file gra while ybh is in TPF$.

0	.10000000+01	11	-.31216709-01	22	-.63481236-01
1	.60598906+00	12	-.34091541-01	23	-.51319714-01
2	.26923198+00	13	-.21461296-01	24	.28734243-01
3	.75024208-01	14	.30060796-01	25	.97583456-01
4	-.63190014-01	15	.52854193-01	26	.11406376+00
5	-.58960785-01	16	-.37186533-01	27	.13826372+00
6	-.56973809-01	17	-.79431379-01	28	.12823801+00
7	-.35041797-01	18	-.37290229-01	29	.14123514+00
8	-.18516959-01	19	.24989689-01	30	.14814627+00
9	-.64888234-02	20	.19790754-01	31	.98589633-01
10	.41145700-02	21	-.21608890-01	32	.20618195-01

VARIANCE=   .14191380+00
MEAN=   .40113709-05
DO YOU WANT THE COVARIANCES STORED(YES/NO)?
yes
COVARIANCE STORED.

DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
cor
LENGTH OF SERIES?
362
NUMBER OF LAGS(ZERO INCLUDED)?
33
INPUT DATA FORMAT?
(4e15.8)
INPUT DATA:
@add gra.sskp

0	.10000000+01	11	.34387418-01	22	.15657139-01
1	.48397417+00	12	.85019480-01	23	-.32998001-02
2	.16358888+00	13	.18485049-01	24	.19564041-02
3	.34291061-01	14	.71320571-01	25	.20206913-01
4	-.82355165-01	15	.29714237-01	26	.47620085-01
5	-.48359534-01	16	-.17095344-01	27	.65758524-01
6	-.44122326-01	17	.44739600-02	28	.61266199-01
7	-.32026744-01	18	.17245420-02	29	.77879018-02
8	-.34230735-01	19	.19832755-01	30	-.39413126-01
9	-.71012059-02	20	.87463330-02	31	-.93015461-02
10	.96968840-02	21	.34905837-02	32	-.10927683-01

VARIANCE=   .11447779+01
MEAN=   .40104705-05
DO YOU WANT THE COVARIANCES STORED(YES/NO)?
yes
COVARIANCE STORED.

DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
end
ACF TERMINATES.
@asf,n
DATA SPACING?
1.

```
WINDOW TO BE USED(TUKEY/DANIELL)?
tukey
 BANDWIDTH PARAMETER?
33
 NUMBER OF FREQUENCY POINTS TO BE VIEWED?
33
 DO YOU WANT CONFIDENCE LIMITS(YES/NO)?
no
 DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD) DISPLAYED?
period
 AUTO-SPECTRUM:
  PERIOD    SPECTRUM
 99.9999    .33286+00
 64.0000    .34424+00
 32.0000    .39179+00
 21.3333    .38674+00
 16.0000    .38696+00
 12.8000    .39049+00
 10.6667    .33785+00
  9.1429    .31070+00
  8.0000    .26813+00
  7.1111    .23930+00
  6.4000    .18930+00
  5.8182    .10181+00
  5.3333    .10485+00
  4.9231    .13927+00
  4.5714    .91497-01
  4.2667    .44466-01
  4.0000    .50774-01
  3.7647    .62834-01
  3.5556    .55293-01
  3.3684    .41898-01
  3.2000    .44758-01
  3.0476    .53232-01
  2.9091    .46325-01
  2.7826    .38710-01
  2.6667    .40210-01
  2.5600    .44652-01
  2.4615    .46159-01
  2.3704    .35595-01
  2.2857    .23144-01
  2.2069    .18414-01
  2.1333    .19327-01
  2.0645    .18276-01
  2.0000    .15677-01
 DO YOU WANT THE SPECTRUM STORED(YES/NO)?
yes
 SPECTRUM STORED.
 DO YOU HAVE ANOTHER SERIES(YES/NO)?
yes
 IS THE SERIES THE SAME(YES/NO)?
no
 DATA SPACING?
1.
```

```
 WINDOW TO BE USED (TUKEY/DANIELL)? .
tukey
 BANDWIDTH PARAMETER?
33
 NUMBER OF FREQUENCY POINTS TO BE VIEWED?
33
 DO YOU WANT CONFIDENCE LIMITS (YES/NO)?
no
 DO YOU WANT FREQUENCY (FREQ.) OR PERIOD (PERIOD) DISPLAYED?
period
 AUTO-SPECTRUM:
  PERIOD      SPECTRUM
 99.9999     .25931+01
 64.0000     .23221+01
 32.0000     .21251+01
 21.3333     .23342+01
 16.0000     .27527+01
 12.8000     .29730+01
 10.6667     .26473+01
  9.1429     .21566+01
  8.0000     .17926+01
  7.1111     .17529+01
  6.4000     .16743+01
  5.8182     .13744+01
  5.3333     .11421+01
  4.9231     .97125+00
  4.5714     .83534+00
  4.2667     .66678+00
  4.0000     .61252+00
  3.7647     .67283+00
  3.5556     .62783+00
  3.3684     .50060+00
  3.2000     .49504+00
  3.0476     .61500+00
  2.9091     .69453+00
  2.7826     .60262+00
  2.6667     .39595+00
  2.5600     .36967+00
  2.4615     .53734+00
  2.3704     .56806+00
  2.2857     .37592+00
  2.2069     .20381+00
  2.1333     .17313+00
  2.0645     .23317+00
  2.0000     .27656+00
 DO YOU WANT THE SPECTRUM STORED (YES/NO)?
yes
 SPECTRUM STORED.
 DO YOU HAVE ANOTHER SERIES (YES/NO)?
no
```

ASF TERMINATES.
@xcf

XCF COMPUTES THE CROSS-COVARIANCE/-CORRELATION FUNCTION OF TWO
SERIES. IT IS ASSUMED THAT THE AUTO-COVARIANCES OF THE TWO SERIES
ARE STORED ON UNIT 1. THE CROSS-COVARIANCES ARE STORED ON UNIT 1
(IF DESIRED) AFTER THE AUTO-COVARIANCES.
DO YOU WANT COVARIANCE(COV) OR CORRELATION(COR)?
cor
INPUT DATA FORMAT OF SERIES 1?
(4e15.8)
INPUT DATA OF SERIES 1:
@add ybh
INPUT DATA FORMAT OF SERIES 2?
(4e15.8)
INPUT DATA OF SERIES 2:
@add gra.sskp

CROSS-COR. FOR SERIES 1 VS.   2.

0	-.31068495+00	11	.68149552-01	22	-.19024821-01
1	-.32688250+00	12	.19405418-01	23	.28423370-01
2	-.23031710+00	13	.67258818-01	24	.65661986-01
3	-.15878268+00	14	.33894985-01	25	.71599943-01
4	-.97850593-02	15	.15917252-01	26	.77692746-01
5	.70688121-02	16	.44837943-01	27	.54928598-01
6	-.78098371-02	17	.68729260-01	28	.34432033-01
7	.32410393-04	18	.30546963-01	29	-.31944079-02
8	-.14137039-01	19	-.21630687-01	30	-.27489146-01
9	.92356610-02	20	-.22341036-01	31	-.65063784-01
10	.56534438-01	21	-.36277876-01	32	-.22737283-01

CROSS-COR. FOR SERIES 2 VS.   1.

0	-.31068495+00	11	.68149863-02	22	.22976052-01
1	.16895355+00	12	-.29895622-01	23	-.29432392-01
2	.40202706+00	13	.19673600-01	24	-.10881614+00
3	.40670389+00	14	-.28525013-01	25	-.13978173+00
4	.38908781+00	15	-.15708253-01	26	-.98344548-01
5	.27146963+00	16	.39682694-01	27	-.48471882-01
6	.16915689+00	17	.24792291-01	28	.39428172-01
7	.90782260-01	18	-.34359416-01	29	.11597794+00
8	.41205884-01	19	-.67074667-01	30	.10655324+00
9	.65060452-02	20	-.42766543-01	31	.97206097-01
10	.64103368-02	21	.89084057-02	32	.10046362+00

```
   DO YOU WANT THE COVARIANCES STORED(YES/NO)?
yes
   COVARIANCES STORED.
    XCF TERMINATES.
@xsf
   XSF COMPUTES THE CROSS-SPECTRA FUNCTIONS FOR TUKEY WINDOW.
   IT IS ASSUMED THAT THE AUTO- AND CROSS-COVARIANCES ARE STORED
   ON UNIT 1 AND THAT THE AUTO-SPECTRA ARE STORED ON UNIT 2.
   THE CROSS-SPECTRA ARE STORED ON UNIT 2 AFTER THE AUTO-SPECTRA(IF DESIRED).
   DO YOU WANT FREQUENCY(FREQ.) OR PERIOD (PERIOD)
   DISPLAYED?
freq


   FREQ.      ARCTANH(COHERENCE) FOR SERIES 1 VS. 2 .
    .0000      .45577+00                              +
    .0156      .37510+00                         +
    .0313      .14873+00             +
    .0469      .17172+00             +
    .0625      .24303+00               +
    .0781      .33788+00                 +
    .0938      .52812+00                    +
    .1094      .63082+00                      +
    .1250      .67882+00                       +
    .1406      .71782+00                       +
    .1563      .73696+00                        +
    .1719      .55091+00                      +
    .1875      .35149+00                  +
    .2031      .45226+00                    +
    .2188      .51547+00                     +
    .2344      .60792+00                      +
    .2500      .75909+00                        +
    .2656      .69457+00                       +
    .2813      .49093+00                    +
    .2969      .44707+00                   +
    .3125      .84579+00                         +
    .3281      .12234+01                                   +
    .3438      .10487+01                              +
    .3594      .78859+00                         +
    .3750      .64568+00                      +
    .3906      .59831+00                     +
    .4063      .56528+00                    +
    .4219      .54029+00                    +
    .4375      .41398+00                  +
    .4531      .11776+00            +
    .4688      .13592+00             +
    .1844      .36685+00                 +
    .5000      .45305+00                   +
NU=  30.                +                 +
```

FREQ.   PHASE FOR SERIES 1 VS. 2 .

FREQ.	PHASE
. 000	.00000
.0156	-.52353+01
.0313	-.39472+02
.0469	-.13202+03
.0625	-.14719+03
.0781	-.16059+03
.0938	-.16853+03
.1094	-.17007+03
.1250	-.17200+03
.1406	-.17227+03
.1563	-.17250+03
.1719	-.17193+03
.1875	-.16744+03
.2031	-.17176+03
.2188	-.17493+03
.2344	-.17545+03
.2500	-.17624+03
.2656	-.17626+03
.2813	-.17471+03
.2969	-.17732+03
.3125	-.17998+03
.3281	-.17938+03
.3438	-.17893+03
.3594	-.17812+03
.3750	-.17657+03
.3906	-.17693+03
.4063	-.17884+03
.4219	-.17872+03
.4375	-.17625+03
.4531	-.17222+03
.4688	.17008+03
.4844	.17622+03
.5000	.18000+03

-180   0   .180

```
DO YOU WANT THE CO-SPECTRA AND QUADRATURES
STORED FOR TIME SERIES REGRESSION(YES/NO)?
no
XSF TERMINATES.
```

THE NEXT GROUP USES THE FOLLOWING PARTS OF TSAP:
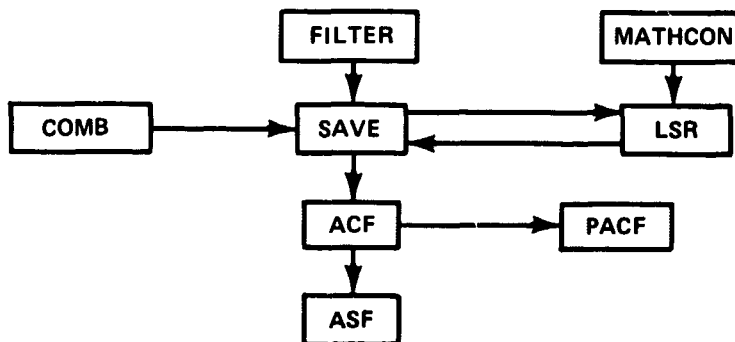
        The main object of this group of examples is to illustrate
the regression and partial auto-correlation programs. This is accom-
plished by regressing one series onto two others with LSRA and then
applying ACF and PACF to the residuals. On the basis of the output
from PACF, the residual series is modeled as a second order auto-
regression. The residuals corresponding to this model are obtained
by successive applications of FILTER and COMB and then examined using
PACF and ASF.

        The final example of this group illustrates LSRP.

        The forward stepwise regression program, LSRA, is
illustrated first by regressing a series x1, consisting of 123
terms, onto two series, y1 and time123, of the same length. The
series time 123 consists of the positive integers from 1 to 123
and was generated using MATH and MATHCON:

```
       @asg,a gra.
  ───▶ @math
       MATH VERSION 3.   WHAT CAN I DO FOR YOU?

       1.1 set a(i)=i.
       do part 1 for i=1(1)123.
       use file gra.
       ROGER.
       save a as a.
       DONE.
  ───▶ @mathcon,i  ,gra*gra.time123
       READY.
       @add gra*gra.a
       @nop
       MATHCON  TERMINATES.
```

➤ @lsra

   LSRA IS A STEPWISE MULTIPLE REGRESSION PROGRAM
   WITH ANALYSIS OF VARIANCE PRINTOUT.
   THE RESIDUALS ARE STORED ON UNIT 8(IF DESIRED).

   NUMBER OF OBSERVATIONS.
123
 NUMBER OF VARIABLES(DEPENDENT AND INDEPENDENT)?
3
 F LEVEL
3.25
 DEPENDENT VARIABLE FORMAT?

(4e15.8)
 DEPENDENT VARIABLE DATA:
@add xl
 INDEPENDENT VARIABLE 1 FORMAT?
(4e15.8)
 INDEPENDENT VARIABLE 1 DATA:
@add yl
 INDEPENDENT VARIABLE 2 FORMAT?
(e15.8)
 INDEPENDENT VARIABLE 2 DATA:
@add gra.time123
 MEANS:
    1   .39642049+01   2   .62000000+02

   CORRELATION COEFFICIENTS:
      1 VS 2 =   -.116816
      1 VS Y =   .787702   2 VS Y[1] = .515992
   STANDARD ERROR OF Y =   .478109+00

SSVAR=**************	SSRES=	27.887794	DFRES=	122.000000
MSRES= .228588	T=	.000000	MULTCOR=	.000000

   STEP NO. 1
      VARIABLE ENTERING: 1
      FLEVEL   197.819456
      STANDARD ERROR OF Y   .295757
      CONSTANT   .289935+00

VARIABLE	COEFFICIENT	STD. ERROR OF COEFF.
X 1	.92686272+00	.65899315-01

SSVAR=	17.303675	SSRES=	10.584119	DFRES=	121.000000
MSRES=	.087472	T=	14.064830	MULTCOR=	.620475

---

[1] In the computer responses, Y is a generic symbol for the
dependent variable while the symbol X, with subscripts, represents
an independent variable. The first X-variable entered is that
independent variable having the highest correlation with the
dependent variable.

```
STEP NO.  2
     VARIABLE ENTERING:  2
     FLEVEL    9494.080078
     STANDARD ERROR OF Y      .033180
     CONSTANT  -.558482+00
```

	VARIABLE	COEFFICIENT	STD. ERROR OF COEFF.
	X 1	.10115818+01	.74439476-02
	X 2	.82666745-02	.84840725-04

```
 SSVAR=    10.452011  SSRES=       .132108   DFRES=   120.000000
 MSRES=       .001101     T=    97.437571   MULTCOR=       .995263
```

DO YOU WANT TO SEE SOME OF THE DEVIATIONS(YES/NO)?
<u>yes</u>
WHAT PART(INITIAL,FINAL,STEP)?
<u>1,120,12</u>

```
PREDICTED VS. ACTUAL RESULTS:
OB. NO.    ACTUAL       PREDICTED      %DEVIATION
   1     .385775+01    .377441+01        2.16
  13     .321840+01    .317747+01        1.27
  25     .337288+01    .336292+01         .30
  37     .342785+01    .347028+01       -1.24
  49     .432563+01    .437717+01       -1.19
  61     .344014+01    .348335+01       -1.26
  73     .400280+01    .399166+01         .28
  85     .357452+01    .357611+01        -.04
  97     .464195+01    .463681+01         .11
 109     .479612+01    .476218+01         .71
```
DO YOU WANT THE RESIDUALS STORED(YES/NO)?
<u>yes</u>
OUTPUT DATA FORMAT?
<u>(4e15.8)</u>
SERIES STORED.
LSR TERMINATES.
➤ <u>@save lsra, rl</u>

      We have not printed out enough of the deviations of predicted from actual values to decide if the residuals are close to being white. However, this can be determined by submitting the residuals, saved under the name rl, to ACF and PACF:

➤ <u>@acf,n</u>

DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
<u>cor</u>
LENGTH OF SERIES?
<u>123</u>
NUMBER OF LAGS(ZERO INCLUDED)?
<u>21</u>

INPUT DATA FORMAT?
(4e15.8)
INPUT DATA:
@add rl

0	.10000000+01	7	.63026889+00	14	.30409733+00
1	.91360913+00	8	.58674648+00	15	.27090824+00
2	.86974473+00	9	.53077113+00	16	.24346690+00
3	.82899847+00	10	.48667117+00	17	.21895623+00
4	.77865961+00	11	.45664223+00	18	.18133523+00
5	.72503306+00	12	.41310194+00	19	.16843886+00
6	.66401155+00	13	.35039130+00	20	.14462271+00

VARIANCE=   .10740904-02
MEAN= -.14518410-06
DO YOU WANT THE COVARIANCES STORED(YES/NO)?
yes
COVARIANCE STORED.

DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR END?
end
ACF TERMINATES.

@pacf
PACF COMPUTES THE PARTIAL AUTO-CORRELATION AND A.R. COEFFICIENTS
UP TO ORDER K TOGETHER WITH THE RESIDUAL VARIANCES OF THE
FITTED A.R.'S UP TO ORDER K AND THE FIRST ORDER M.A. COEFFICIENTS.

MAXIMUM ORDER OF K?
15

PACF:

1	.91360912+00	6	-.99956975-01	11	.72968422-01
2	.21209436+00	7	.10971080+00	12	-.60200907-01
3	.57967539-01	8	-.56858845-02	13	-.15911209+00
4	-.57376178-01	9	-.99147042-01	14	-.18779177-01
5	-.71001749-01	10	-.14166919-01		

DO YOU WANT RESIDUAL VARIANCES(YES/NO)?
yes

RESIDUAL VARIANCES:

1	.18052634-03	6	.17648832-03	11	.18078887-03
2	.17389055-03	7	.17607510-03	12	.18216872-03
3	.17483730-03	8	.17784500-03	13	.17962297-03
4	.17584129-03	9	.17792252-03	14	.18171295-03
5	.17658256-03	10	.17978376-03		

ORDER OF A.R. PROCESS TO BE VIEWED(0=NONE OR NO MORE)?
2

.71983778+00   .21209436+00
ORDER OF A.R. PROCESS TO BE VIEWED(0=NONE OR NO MORE)?
0
DO YOU WANT THE FIRST ORDER M.A. COEFFICIENTS(YES/NO)?
no
PACF TERMINATES.

Because the acf attenuates, the pacf appears to truncate[1]
at the 2nd lag and the 2nd residual variance is minimum for a
2nd order a.r. model, it is reasonable to examine the hypothesis
that the residuals rl satisfy the following 2nd order a.r. model:

$$\varepsilon(t) = rl(t) - .7198rl(t-1) - .2121rl(t-2)$$

for t=3,4,...,123.

The construction of $\varepsilon(t)$ allows us to illustrate the
use of the "length of series to be used", "lag", and
"scale" transformations in FILTER together with the series
subtraction operator in COMB. (Of course, once it is noticed
that the residuals, rl, are auto-correlated the regression of
xl onto yl and timel23 must be re-examined along the lines
suggested in Durbin [5] or Hannan [10].)

@filter,n
LENGTH OF SERIES TO BE READ IN?
123
INPUT DATA FORMAT?
(4e15.8)
INPUT DATA:
@add rl
LENGTH OF SERIES TO BE USED?
123

---

[1]Under the hypothesis that the process is a pth order auto-
regression the kth value of the pacf has a normal distribution with
mean zero and standard deviation equal to $1/\sqrt{N}$, where the N is the
length of the series. In this particular example, the two standard
deviation band for significance is $\pm.18$.

```
 MEAN=     -.14518410-06
 LENGTH=    123

 FILTER?
lag
 NUMBER OF LAGS?
2
 MEAN=     -.12660986-02
 LENGTH=    121

 FILTER?
end
 DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
no
 DO YOU WANT THE FINAL SERIES STORED(YES/NO)?
yes
 OUTPUT DATA FORMAT?
(4e15.8)
 SERIES STORED.
 FILTER TERMINATES.
```
➤ @save filter, rll

```
@filter,n
 LENGTH OF SERIES TO BE READ IN?
123
 INPUT DATA FORMAT?
(4e15.8)
 INPUT DATA:
@add rl
 LENGTH OF SERIES TO BE USED?
122
 MEAN=     -.42588965-03
 LENGTH=    122

 FILTER?
lag
 NUMBER OF LAGS?
1
 MEAN=     -.11181355-02
 LENGTH=    121

 FILTER?
scale
 SCALE OPERATOR(ADD/MPY)?
mpy
 SCALE FACTOR?
.7198
 MEAN=     -.80483410-03
 LENGTH=    121
```

```
     FILTER?
   end
     DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
   no
     DO YOU WANT THE FINAL SERIES STORED(YES/NO)?
   yes
     OUTPUT DATA FORMAT?
   (4e15.8)
     SERIES STORED.
     FILTER TERMINATES.
```

➡ @save filter, rl2

➡ @filter,n
```
     LENGTH OF SERIES TO BE READ IN?
   123
     INPUT DATA FORMAT?
   (4e15.8)
     INPUT DATA:
   @add rl
     LENGTH OF SERIES TO BE USED?
   121
     MEAN=      -.16543380-03
     LENGTH=    121


     FILTER?
   scale
     SCALE OPERATOR(ADD/MPY)?
   mpy
     SCALE FACTOR?
   .2121
     MEAN=      -.16543380-03
     LENGTH=    121


     FILTER?
   end
     DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
   no
     DO YOU WANT THE FINAL SERIES STORED(YES/NO)?
   yes
     OUTPUT DATA FORMAT?
   (4e15.8)
     SERIES STORED.
     FILTER TERMINATES.
```
➡ @save filter, rl3

      The COMB program is now employed to subtract the series
rl2 and rl3 from rll to form εl.

→ @comb,n
  LENGTH OF SERIES 1 TO BE READ IN?
121
  INPUT DATA 1 FORMAT?
(4e15.8)
  INPUT DATA 1 :
@add r11
  LENGTH OF SERIES 2 TO BE READ IN?
121
  INPUT DATA 2 FORMAT?
(4e15.8)
  INPUT DATA 2 :
@add r12
  LENGTH OF OUTPUT SERIES?
121
  OPERATOR TO BE USED?
sub
  DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
no
  DO YOU WANT THE RESULTANT SERIES STORED(YES/NO)?
yes
  OUTPUT DATA FORMAT?
(4e15.8)
  SERIES STORED.
  COMB TERMINATES.
→ @save comb, kl
→ @comb,n
  LENGTH OF SERIES 1 TO BE READ IN?
121
  INPUT DATA 1 FORMAT?
(4e15.8)
  INPUT DATA 1 :
@add kl
  LENGTH OF SERIES 2 TO BE READ IN?
121
  INPUT DATA 2 FORMAT?
(4e15.8)
  INPUT DATA 2 :
@add r13
  LENGTH OF OUTPUT SERIES?
121
  OPERATOR TO BE USED?
sub
  DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?
no
  DO YOU WANT THE RESULTANT SERIES STORED(YES/NO)?
yes
  OUTPUT DATA FORMAT?
(4e15.8)
  SERIES STORED.
  COMB TERMINATES.
→ @save comb, el

→ **@pacf,n**

  MAXIMUM ORDER OF K?
  <u>10</u>

  PACF:

      1   -.85908396-01   4   .12719820+00   7   .14001438+00
      2   -.24426123-01   5   .31113816-01   8   .59418262-01
      3   .12837213+00   6   -.12368082+00   9   .17003972-01
  DO YOU **WANT** RESIDUAL VARIANCES (YES/NO)?

  <u>no</u>
  ORDER OF A.R. PROCESS TO BE VIEWED (0=NONE OR NO MORE)?
  <u>0</u>
  DO **YOU WANT** THE FIRST ORDER M.A. COEFFICIENTS (YES/NO)?
  <u>yes</u>

  FIRST ORDER M.A. COEFFICIENTS:

      1   -.92659724-01   4   .12719820+00   7   -.14001438+00
      2   -.92405158-01   5   .31113816-01   8   .59418262-01
      3   -.77088748-01   6   -.12368082+00   9   .17003972-01

→ **@asf,n**
  DATA SPACING?

  <u>1.</u>
  WINDOW **TO BE** USED (TUKEY/DANIELL)?

  <u>tukey</u>
  BANDWIDTH PARAMETER?

  <u>10</u>
  NUMBER **OF** FREQUENCY POINTS TO BE VIEWED?

  <u>24</u>
  DO YOU **WANT** CONFIDENCE LIMITS (YES/NO)?

  <u>yes</u>
  DO YOU **WANT** FREQUENCY (FREQ.) OR PERIOD (PERIOD) DISPLAYED?
  <u>freq.</u>

```
AUTO-SPECTRUM:
  FREQ.      SPECTRUM
  .0000     .10650-03
  .0217     .10318-03
  .0435     .94606-04
  .0652     .83922-04
  .0870     .74053-04
  .1087     .66422-04
  .1304     .61409-04
  .1522     .58782-04
  .1739     .63310-04
  .1957     .73549-04
  .2174     .89675-04
  .2391     .10746-03
  .2609     .12082-03
  .2826     .12535-03
  .3043     .12115-03
  .3261     .11286-03
  .3478     .10652-03
  .3696     .10551-03
  .3913     .10848-03
  .4130     .11091-03
  .4348     .10888-03
  .4565     .10228-03
  .4783     .94936-04
  .5000     .91759-04
```
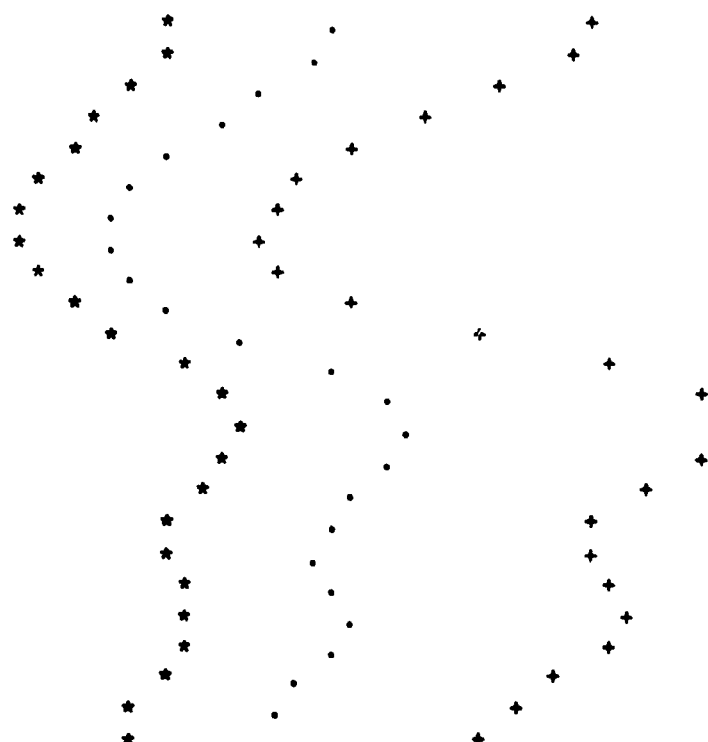
THE 90% CONFIDENCE LIMITS FOR NU=36 are: CU= 1.54095 , CL= .70309
THE SQRT OF VARIANCE RATIO IS: 24.896 %
DO YOU WANT THE SPECTRUM STORED(YES/NO)?
<u>no</u>
DO YOU HAVE ANOTHER SERIES(YES/NO)?
<u>no</u>
ASF TERMINATES.

        The final example involves an application of the second LSR program, LSRP:

→ <u>@lsrp</u>

LSRP IS A STEPWISE MULTIPLE REGRESSION PROGRAM WITH PREDICTIONS.
THE PREDICTION RESULTS ARE STORED(IF DESIRED) ON UNIT 8.
NUMBER OF OBSERVATIONS?
<u>7</u>
NUMBER OF VARIABLES(DEPENDENT AND INDEPENDENT)?
<u>2</u>
F LEVEL?
<u>3.25</u>
DEPENDENT VARIABLE FORMAT?
<u>(e20.10)</u>
DEPENDENT VARIABLE DATA:
<u>@add g35log</u>

INDEPENDENT VARIABLE 1 FORMAT?
(e20.10)
INDEPENDENT VARIABLE 1 DATA:
@add gra.time
T-VALUE FOR NU=   5 DEGREES OF FREEDOM?
2.015
STANDARD ERROR OF Y   .0108512
VARIABLE COEFFICIENT   S.D.   UPPER T   LOWER T
(0=CONST.)
        0   .76394+01   .91710-02   .76579+01   .76209+01
        1   .34170-01   .20507-02   .38302-01   .30038-01
PREDICTION DATA:
NUMBER OF PREDICTION VALUES?
17
INDEPENDENT VARIABLE 1 FORMAT?
(e20.10)
INDEPENDENT VARIABLE 1 DATA:
@add gra.time17
PREDICTED VALUES:
NUMBER   PREDICTION   UPPER T   LOWER T
      1   .7673548+01   .7700006+01   .7647089+01
      2   .7707718+01   .7732510+01   .7682925+01
      3   .7741888+01   .7765625+01   .7718150+01
      4   .7776058+01   .7799432+01   .7752683+01
      5   .7810228+01   .7833965+01   .7786490+01
      6   .7844398+01   .7869190+01   .7819605+01
      7   .7878568+01   .7905026+01   .7852109+01
      8   .7912738+01   .7941366+01   .7884109+01
      9   .7946908+01   .7978105+01   .7915711+01
     10   .7981078+01   .8015152+01   .7947003+01
     11   .8015248+01   .8052437+01   .7978059+01
     12   .8049418+01   .8089904+01   .8008931+01
     13   .8083588+01   .8127513+01   .8039663+01
     14   .8117758+01   .8165232+01   .8070283+01
     15   .8151928+01   .8203039+01   .8100816+01
     16   .8186098+01   .8240917+01   .8131279+01
     17   .8221268+01   .8278851+01   .8161685+01
DO YOU WANT THE PREDICTION VALUES AND LIMITS SAVED(YES/NO)?
yes
OUTPUT FORMAT?
(e20.10)

SERIES STORED IN FOLLOWING ORDER:
PREDICTION SERIES,LOWER LIMIT SERIES,UPPER LIMIT SERIES.
LSRP TERMINATES.
⟶ @save lsrp, pul

4.0   PROGRAM TECHNIQUES AND FLOW CHARTS

The programs are designed to operate in an interactive mode for user ease and convenience. The system, TSAP, is a set of individual main programs to allow interaction with system processors, e.g., FURPUR or MATH, without the need to exit TSAP. This also allows the user to devise similar programs to be used in conjunction with TSAP, e.g., tape I/O or unpacking data. Data output at the terminal is kept minimal since terminal I/O is so much slower than printer I/O.

Intermediate results of interdependent programs are stored in binary on dynamically assigned FASTRAND files 1, 2 and 8 to save redundant calculations and/or I/O. Two such "linked" programs are ACF and PACF (PACF needs the covariances generated by ACF). This also prevents errors in answering the same questions in both programs.

At Bellcomm, terminal core is restricted so a maximum series length of 1000 is built into all programs. This length is more of a "practical computation" size than an upper core limit and may be changed fairly easily. LSR is presently restricted to 5 series of length 500, again changeable.

An N option is available on most of the programs to skip the introductory commentary for the experienced user.

In ASF, the confidence limits use 90% chi square tables for $\gamma \leq 60$ degrees of freedom. Exceeding this limit may cause problems which can best be avoided by not asking for confidence limits for such series. The on-line plots of ASF (and XSF) consist of a 50 x NQ (NQ = number of frequency points to be viewed) grid which may cause distortion problems. It also makes comparisons between different series more difficult. XSF plots only the coherence ($\tanh^{-1}$) and phase to retain terminal efficiency while giving the most pertinent data output. These problems will be removed if a CRT display is incorporated into the system.

Flow charts of the main programs follow.

XSF

READ IN NUMBER
OF LAGS, LENGTH
OF SERIES

READ IN CROSS-
COVARIANCES

READ IN NUMBER
OF FREQUENCY
POINTS, AUTO-
SPECTRA

COMPUTE CO-
SPECTRUM
QUADRATURE
COHERENCE
PHASE

PRINT OUT
COHERENCE
AND PHASE
WITH PLOTS

STORE CO-SPECTRUM
AND QUADRATURE?   NO

YES

STORE CO-SPECTRUM
AND QUADRATURE

STOP

(REBSOM IS A UNIVAC STAT-PACK
BACKWARD REGRESSION SUBROUTINE)

LSRA

READ IN LENGTH
AND NUMBER OF
SERIES, F LEVEL,
SERIES

CALL
RESTEM

(RESTEM IS A UNIVAC STAT-PACK STEPWISE
FORWARD REGRESSION SUBROUTINE)

PRINT OUT CORRE-
LATION COEFFICI-
ENTS, STANDARD
ERRORS, COEFFICI-
ENTS FOR EACH
STEP I

FINAL I?   NO

YES

SEE % DEVIATIONS?   NO

YES

PRINT OUT
DEVIATIONS

STORE RESIDUALS?   NO

YES

STORE
RESIDUALS

STOP
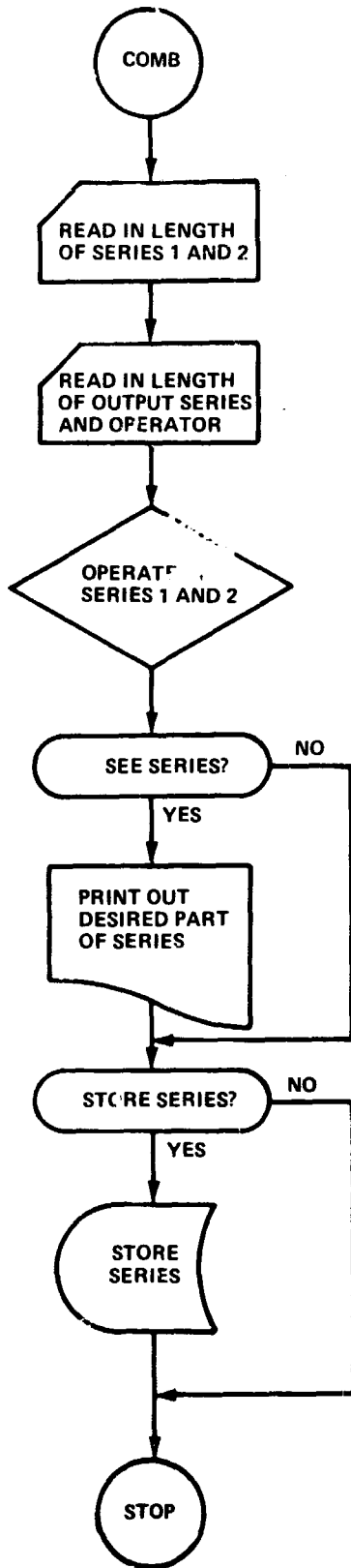
SUMMARY

          The original TSAP system was developed to aid in the
study of electron flux data from the earth's outer trapped
radiation belts. Since then it has been applied to a variety
of time series problems and each of these problems has influenced
its present form. As noted in the introduction, the present
content of TSAP will be increased in accordance with the needs
of future applications.

          The principal feature of TSAP is the fact that it is a
terminal system. Time series analysis is ideally suited to such
a mode of operation because it requires relatively long sequences
of statistical computations which ordinarily depend directly either
on the previous computations or on statistical judgements derived
from these computations.

          Simply stated, TSAP allows real time statistical judgements
to be made in time series analysis on direction of the analysis.
The versatility of TSAP is enhanced by its capability of interacting
with machine processors and user designed terminal programs.

G. R. Andersen

R. R. Singers

1031-$\frac{GRA}{RRS}$-jf

Attachments
  Appendices I - III
  References

APPENDIX I.

The stochastic properties of stationary stochastic processes are well known and are concisely presented in U. Grenander and M. Rosenblatt [8]. A recent practical approach to the subject may be found in the book by G. Jenkins and D. Watts [12]. The definitive text in the area of multiple time series analysis is the work by E. J. Hannan[1].

A.1 Spectral Analysis (ACF, ASF, XCF, XSF):

Let $\{X_j(t)\}$, $j=1,2$, and $t=1,2,\ldots,N$, be finite realizations of two jointly stationary stochastic processes. These time series may be realizations of discrete processes or discrete series formed by sampling a realization, $\xi_j(t)$, of a continuous time stochastic process at equidistant time points, $t \cdot \Delta$, $(\Delta > 0)$, and defining $X_j(t) = \xi_j(t \cdot \Delta)$, $t=1,\ldots,N$.

In either case, let the centered series be denoted by

$$\tilde{X}_j(t) = X_j(t) - \overline{X}_j \tag{1}$$

where

$$\overline{X}_j = \frac{1}{N} \sum_{t=1}^{N} X_j(t), \quad (j=1,2) \tag{2}$$

and form the lagged (sample) cross covariance, C_{jk}, by setting

$$C_{jk}(\nu) = \frac{1}{N} \sum_{d=1}^{N-\nu} \tilde{X}_j(d) \tilde{X}_k(d+\nu), \tag{3}$$

---

[1] Hannan's book will be published by J. Wiley and Sons late in 1970.

for $\nu = 0, 1, \ldots, N-1$ and extend this definition to $-1, \ldots, -N+1$ by setting

$$C_{jk}(-\nu) = C_{kj}(\nu) \quad . \tag{4}$$

In an attempt to be consistent with most mathematical treatments of stationary stochastic processes, we will introduce the basic computational definitions concerning spectra for a unit sampling interval and consider frequency to have the dimensions of radians per unit of time. It is then a simple matter to transform the spectral estimators to the more practical form used in the programs where the data spacing is an arbitrary positive constant, Δ, and frequency is in terms of cycles per unit time.

The sample auto-spectrum, $\hat{g}_{jj}$, of the $j\underline{th}$ time series is defined in terms of the sample auto-covariance function, C_{jj}, and a lag window, ω, by setting

$$\hat{g}_{jj}(\lambda) = \hat{g}_{jj}(\lambda; \omega) =$$

$$= \frac{1}{2\pi} \left(C_{jj}(0) + 2 \sum_{\nu=1}^{N-1} \omega\left(\frac{\nu}{M}\right) C_{jj}(\nu) \cos(\lambda\nu) \right), \tag{5}$$

where $|\lambda| \leq \pi$. The positive integer M, the bandwidth parameter, determines the bandwidth of the spectral window with Fourier coefficients $\omega\left(\frac{\nu}{M}\right) = \omega\left(\frac{-\nu}{M}\right)$, $\nu = 0, 1, \ldots, N-1$.

Two types of spectral windows are available in TSAP: the Tukey window and the Daniell window. The lag windows corresponding to these spectral windows are as follows:

(i)   Tukey:

$$\omega(y) = \omega_T(y) = \begin{cases} \frac{1}{2}(1+\cos(\pi y)), & |y| \leq 1 \\ \\ 0, & |y| > 1 \end{cases}$$

(ii)   Daniell:

$$\omega(y) = \omega_D(y) = \begin{cases} \frac{\sin \pi y}{\pi y}, & y \neq 0 \\ \\ 1, & y = 0. \end{cases}$$

The use of the Tukey window in (5) requires that only M lagged covariances (including the variance) be calculated; whereas, using the Daniell window requires all lagged covariances that can be calculated from a series of length N, namely N (including zero). For this reason the Daniell window should only be used on relatively short series (say, $N \leq 500$).

If we denote the spectral window corresponding to $\omega(\frac{\cdot}{M})$ by W_M, then

$$W_M(s) = \frac{1}{2\pi} \sum_{-N+1}^{N-1} \omega(\frac{\nu}{M}) e^{i\nu s}, \tag{6}$$

$|s| \leq \pi$.

Although the reader should consult any one of the above mentioned references for details on spectral analysis (cf. the bibliography for additional works), we will indicate the relationship between the spectral window and $\hat{g}_{jj}$.

If we set ω identically equal to 1 in (5) we obtain the (continuous) periodogram $\hat{g}_{jj}(\lambda;1)$. Integrating the product $W_M(\lambda-s)\hat{g}_{jj}(s;1)$ with respect to s over the interval $(-\pi,\pi)$ we obtain

$$\hat{g}_{jj}(\lambda;\omega) = \int_{-\pi}^{\pi} W_M(\lambda-s)\hat{g}_{jj}(s;1)ds \quad , \tag{7}$$

$|\lambda| \leq \pi$. This exhibits the sample auto-spectrum as a weighted average of the periodogram. With the Daniell window this averaging is approximately uniform over the interval $(\lambda-h, \lambda+h)$ where $h = \pi/M$. The relationship between the sample auto-spectrum formed with the Tukey window and the periodogram is more easily seen directly from (5) where a simple cosine identity leads to

$$\hat{g}_{jj}(\lambda_k;\omega_T) = \begin{cases} \frac{1}{4}\hat{g}_{jj}(\lambda_{k-1};1)+\frac{1}{2}\hat{g}_{jj}(\lambda_k;1)+\frac{1}{4}\hat{g}_{jj}(\lambda_{k+1};1) & \text{for } k=1,2,\ldots,Q-1 \\ \frac{1}{2}\hat{g}_{jj}(0;1)+\frac{1}{4}\hat{g}_{jj}(\lambda_1;1) & , \quad \text{for } k = 0 \\ \frac{1}{2}\hat{g}_{jj}(\pi;1)+\frac{1}{4}\hat{g}_{jj}(\lambda_{Q-1};1) & , \quad \text{for } k = Q \end{cases} \tag{8}$$

where $\lambda_k = \pi k/Q$, $k=0,1,\ldots,Q$.

Since $\hat{g}_{jj}(\cdot;1)$ is proportional to the modulus squared of the discrete Fourier transform of the time series, (8) is often used for computation when a fast Fourier transform (FFT) is available. The first extension of TSAP will include a FFT routine for use in model estimation. At that time either (8) will be used or, following Gentleman and Sande [6], (5) will be used with the lagged covariances calculated using the FFT and the convolution theorem.

If the time series $\{X_j(t)\}$, $t = 1,2,\ldots,N$, is related to a finite realization of $\xi_j(\tau)$, $0 < \tau \leq T$, of a stationary process by $X_j(t) = \xi_j(\Delta t)$, $\Delta>0$, $N\Delta \leq T$, then the right-hand side of (5), multiplied by Δ and taking λ in $(-\pi/\Delta, \pi/\Delta)$, defines an estimate (call it $g_j^*(\lambda)$) of

$$g_j(\lambda) = \sum_{r=-\infty}^{+\infty} \phi_j(\lambda + \frac{2\pi r}{\Delta}), \quad |\lambda| < \pi/\Delta \quad ,$$

where ϕ_j is the spectrum of ξ_j.

Hence, g_j will be approximately equal to ϕ_j only if Δ is chosen in such a way that $\phi_j(s) \cong 0$ for $|s| > \frac{\pi}{\Delta}$, the so-called Nyquist frequency (cf., R. B. Blackman and J. W. Tukey [3]). Then

$$\phi_j(\lambda) \cong \begin{cases} g_j(\lambda) \ , & |\lambda| \leq \pi/\Delta \\ \\ 0 & , \ |\lambda| > \pi/\Delta \end{cases}$$

and $g_j^*(\lambda)$, $|\lambda| \leq \pi/\Delta$ is an estimate of the spectrum of ξ_j.

We now state the computational forms used in TSAP, allowing for data spacing, Δ, and requiring that frequency be in dimensions of cycles per unit time [1]:

$$\hat{f}_{jj}(\lambda_k) = \Delta \left(C_{jj}(0) + 2 \sum_{\nu=1}^{M-1} \omega_T\left(\frac{\nu}{M}\right) C_{jj}(\nu) \cos\left(\frac{\pi k \nu}{M_1}\right) \right) , \qquad (10)$$

where $\lambda_k = \frac{k}{2M_1\Delta}$ [2], $k=0,1,\ldots,M_1$. M_1 is selected by the user and is generally between M and 2M. The proper bandwidth parameter, M, depends on the unknown spectrum and is usually found (if possible) by the technique of "window closing" (cf., [12]). Taking M to be 5, 10 and 20% of the record length N is a reasonable rule of thumb.

A compromise must usually be made between resolution and accuracy. As a guide to the accuracy of the estimate, the square root of the variance ratio, $\mathrm{var}(\hat{f}_{jj}(\lambda)/f_{jj}(\lambda))$, is printed out at the end of ASF. This quantity is $(3/4)(M/N)$ for the Tukey window and M/N for the Daniell window. The user also has the option of having 90% confidence limits plotted with the sample auto-spectrum. These are the usual limits based on the approximation which states that $\nu \cdot \hat{f}_{jj}(\lambda)/f_{jj}(\lambda)$, where f_{jj} denotes the true (but unknown) auto-spectrum, is a chi-square random variable with ν degrees of freedom, with $\nu = (8/3)(N/M)$ in the case of the Tukey window and $\nu = 2\,N/M$ for the Daniell window. (Cf., Grenander and Rosenblatt [8].)

---

[1] For simplicity we will state them for the Tukey window only since this is all that is available in the cross-spectral segment of TSAP.

[2] The user may choose to plot f_{jj} against period, λ_k^{-1}, $k=1,2,\ldots,M_1$.

The sample co-spectrum and quadrature spectrum of $\{X_j(t)\}$ and $\{X_k(t)\}$ are computed by

$$\hat{p}_{jk}(\lambda_\ell) = \Delta \{C_{jk}(0) + \sum_{\nu=1}^{M-1} (C_{jk}(\nu) + C_{kj}(\nu)) \omega_T(\frac{\nu}{M}) \cos(\frac{\pi \ell \nu}{M_1})\} \qquad (11)$$

and

$$\hat{q}_{jk}(\lambda_\ell) = \Delta \sum_{\nu=1}^{M-1} (C_{jk}(\nu) - C_{kj}(\nu)) \omega_T(\frac{\nu}{M}) \sin(\frac{\pi \ell \nu}{M_1}) \qquad (12)$$

respectively, where $\lambda_\ell = \frac{\ell}{2M_1 \Delta}$, $\ell = 0, 1, \ldots, M_1$.

These functions are used to compute the sample coherence [2]

$$\hat{\gamma}_{jk}(\lambda_\ell) = \begin{cases} \dfrac{\hat{p}_{jk}^2(\lambda_\ell) + \hat{q}_{jk}^2(\lambda_\ell)}{\hat{f}_{jj}(\lambda_\ell) \hat{f}_{kk}(\lambda_\ell)} , & \hat{f}_{jj}(\lambda_\ell) \cdot \hat{f}_{kk}(\lambda_\ell) > 0 \\ \\ 0 , & \text{otherwise} \end{cases} \qquad (13)$$

and the sample phase

$$\hat{\phi}_{jk}(\lambda_\ell) = \begin{cases} \tan^{-1}\left(\dfrac{\hat{q}_{jk}(\lambda_\ell)}{\hat{p}_{jk}(\lambda_\ell)}\right) \quad , \quad \hat{p}_{jk}(\lambda_\ell) > 0 \\[4em] \tan^{-1}\left(\dfrac{\hat{q}_{jk}(\lambda_\ell)}{\hat{p}_{jk}(\lambda_\ell)}\right) - \pi, \quad \hat{p}_{jk}(\lambda_\ell) < 0 \end{cases}$$

$\lambda_\ell = \dfrac{\ell}{2M_1\Delta}$, $\ell=0,1,\ldots,M_1$. For definiteness the phase is set equal to $\dfrac{\pi}{2}$ if $\hat{p}_{jk}(\lambda_\ell) = 0$.

A.2 Auto-Regression Coefficients, Residual Variance, and Partial Auto-Correlation Function (PACF)

All estimates in this section[1] are based on the sample auto-correlation function $r(\nu) = C(\nu)/C(0)$, where

$$C(\nu) = N^{-1} \sum_{t=1}^{N-\nu} X_t\, X_{t+\nu}.$$

We use the results of Durbin [5]:

Letting $a_1^{(1)} = r(1)$, $V_1 = 1$, choose a positive integer $k \leq N-1$, and define $a_1^{(k)}$, $a_2^{(k)}$,..., $a_k^{(k)}$ by setting

---

[1] We assume that the time series under discussion is $\{X(t): t=1,\ldots,N\}$, with $\overline{X}=0$.

$$a_s^{(s)} = (r(s) - a_1^{(s-1)} r(s-1) - a_2^{(s-1)} r(s-2) - \ldots - a_{s-1}^{(s-1)} r(1))/V_{s-1} \quad (1)$$

where

$$V_{s-1} = 1 - a_1^{(s-1)} r(1) - \ldots - a_{s-1}^{(s-1)} r(s-1) \quad (2)$$

for s=2,...,k and

$$a_r^{(s)} = a_r^{(s-1)} - a_s^{(s)} a_{s-r}^{(s-1)} \quad , \quad r=1,\ldots,s-1 \ . \quad (3)$$

Then $a_s^{(s)}$, s=2,...,k, are the partial correlation coefficients measuring the correlation between members of the time series 2,...,k time units apart with the linear effect of intervening observations removed. The terms $a_1^{(s)},\ldots,a_s^{(s)}$ are the least squares estimators of the autoregression coefficients in the s$\underline{\text{th}}$ order a. r. model.

$$y(t) = \alpha_1 y(t-1) + \ldots + \alpha_s y(t-s) + \varepsilon(t) \ . \quad (4)$$

The quantity

$$\sigma_\varepsilon^2(s) = \frac{N-s}{N-2s-1} C(0) \ V_s \quad (5)$$

is an estimator of the residual variance of the model in (4), i.e., the variance of $\varepsilon(t)$. V_s satisfies

$$V_s = V_{s-1} (1 - (a_s^{(s)})^2)$$

APPENDIX II

This appendix contains an explanation of the transformations available in FILTER. Let $\{X(t): t=1,\ldots,N\}$ be a given time series, then a request for "center" transforms $X(t)$ into $X(t) - \overline{X}$, where $\overline{X}$ is the sample mean (cf. Appendix I); symbolically

Center: $X(t) \rightarrow X(t) - \overline{X}$.

Similarly,

Exp: $X(t) \rightarrow e^{X(t)}$;

Log: $X(t) \rightarrow \log_e X(t)$, if $X(t) > 0$, for all t;

Log10: $X(t) \rightarrow \log_{10} X(t)$, if $X(t) > 0$, for all t;

Diff: $X(t) \rightarrow X(t) - X(t-1)$;

10**: $X(t) \rightarrow 10^{X(t)}$

Sqrt: $X(t) \rightarrow \sqrt{X(t)}$, if $X(t) \geq 0$, for all t;

Abs: $X(t) \rightarrow |X(t)|$;

Scale: $X(t) \rightarrow \alpha X(t)$, where the real number α is supplied by the user.

The transformation "LAG", accompanied by a user response k, $1 \leq k \leq N-1$, deletes the first k term from the series, i.e, it maps

$$(X(1), X(2), \ldots, X(k), X(k+1), \ldots, X(N))$$

into

$$(X(k+1), \ldots, X(N))$$

When the high pass Tukey filter (A. S. Alvia and G. M. Jenkins, [1]) is requested, the following linear transformation

$$(F_T^{(m)} X)(t) = \sum_{j=-m}^{m} \lambda_j X(t-j) , \qquad (1)$$

is applied to the series $\{X(\tau): \quad \tau=1,\ldots,N\}$, where $t=m+1,\ldots,N-m$ [1] and $\lambda_0 = 1 - \dfrac{1}{m+1}$,

$\lambda_j = \lambda_{-j} = \dfrac{-1}{2(m+1)} (1 + \cos(\dfrac{j\pi}{m+1}))$, $(j=1,2,\ldots,m)$.

The corresponding frequency response function $A_m(\lambda) = \sum_{-m}^{m} \lambda_j \cos(\lambda_j)$ constitutes the Tukey high pass filter; its effect on the auto-spectrum, f, of the process generating the time series $\{X(t)\}$ is well-known and given by $f_X^{\cdot}(\lambda) = |A_m(\lambda)|^2 f(\lambda)$. The gain $|A_m|^2$ is shown in Figure A.II.1 for m=7.
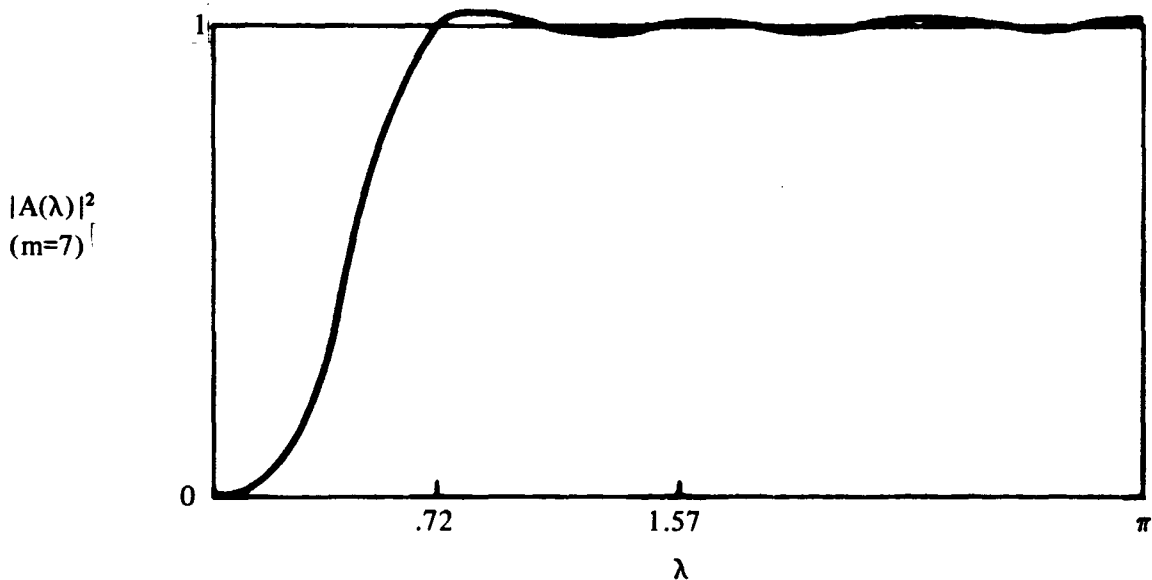
FIGURE A.II.1

---

[1] The length, 2m+1, of the moving average (1) is specified by the user.

The low frequency portion of the series $X(t)$ corresponding to the lcw pass filter with frequency response $1 - A_m(\lambda)$ is just $X(t) - (F_T^{(m)} X)(t)$, $t=m+1,\ldots,N-m$.

The Goodman band pass filter, as presented by N. R. Goodman [7] and extended in the Biomedical Computer Program [4], is a trapezoidal band pass filter constructed from a series of overlapping triangles of unit height with base length $2/m\Delta$ [1], using $2m+1$ pieces of data. The high pass filter that we refer to as a Goodman high pass filter is a simple modification of Goodman's results constructed for use in TSAP.

The linear transformation of $\{X(t)\}$ which produces a bandpass filter with center frequency α_0 using n triangles [2] and $2m+1$ pieces of data is given by

$$(F_{N,\alpha_0}^{(m)} X)(t) = \sum_{j=-m}^{m} \beta_j X(t-j) ,$$

$t=m+1,\ldots,N-m$, where $\beta_0 = 2n/m$,

$$\beta_j = \beta_{-j} = \frac{2}{m}[.54 + .46 \cos(\tfrac{j\pi}{m})]\cos(j\alpha_0) \frac{\sin(\tfrac{nj\pi}{m})}{\sin(\tfrac{j\pi}{m})}$$

$(j=1,2,\ldots,m-1)$ and

$$\beta_m = (-1)^{n+1} \frac{n}{m} (.08)\cos(m\alpha_0) ,$$

---

[1] When frequency is in dimensions of cycles per unit time and data spacing is $\Delta > 0$.

[2] See Figures A.II.2 and 3 for examples with $n=2$ and $n=3$.

provided that $(n+1)/2m\Delta \leq \alpha_0 \leq \frac{1}{2}\Delta(1 - \frac{n+1}{m})$
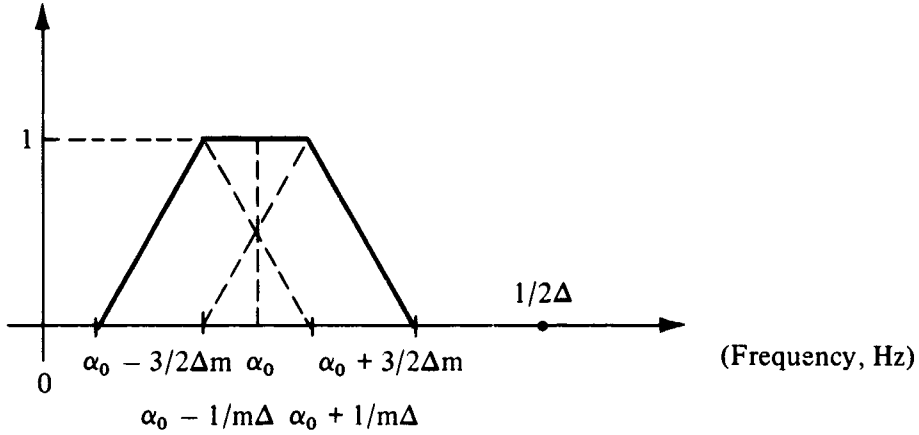
FIGURE A.II.2
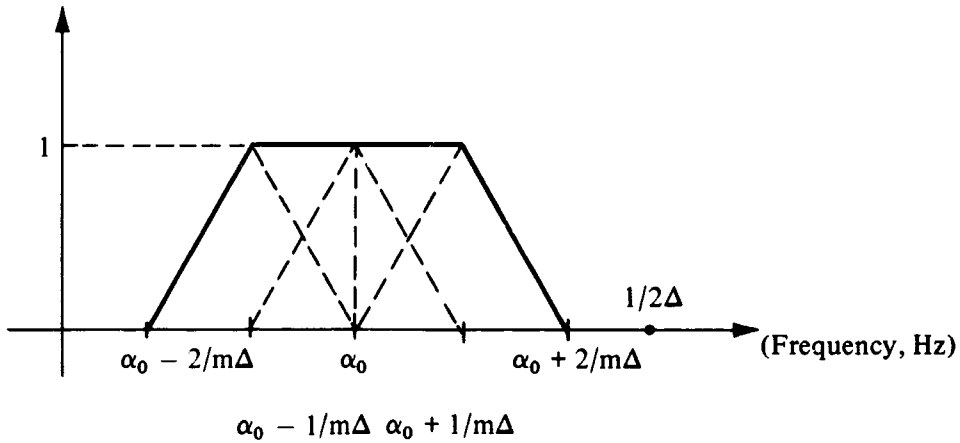
FIGURE A.II.3

The length of the base of the trapezoid is then $(n+1)/m\Delta$.

      The modification of the band pass to high pass filter mentioned earlier is accomplished by observing that the Goodman filters are symmetric with respect to the origin and add if the base of the trapezoid defining the band pass filter on $(0, 1/2\Delta)$ overlaps $1/2\Delta$ (or the origin).
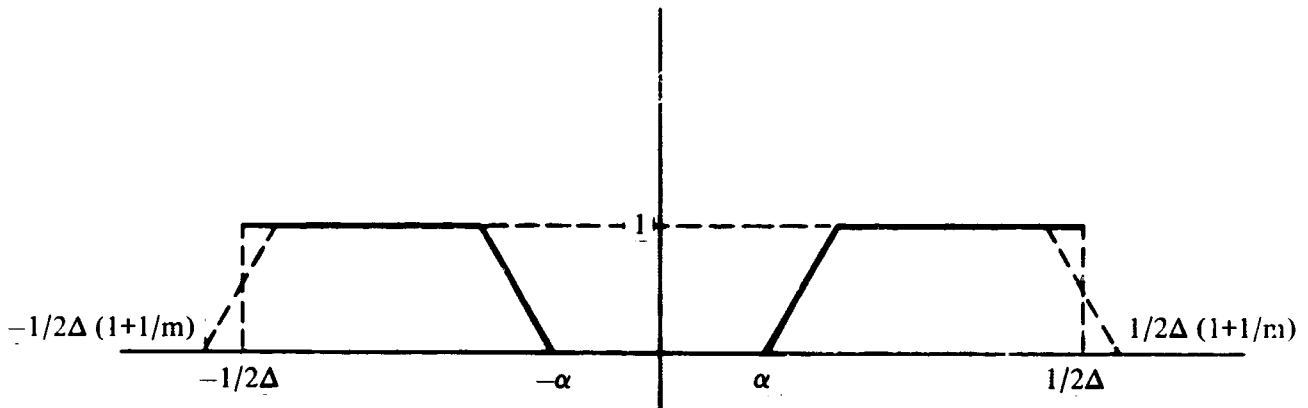
FIGURE A.II.4

      The program requests the number of triangles, **n**, and 2m+1 and tells the user the cut-off point:

$$\alpha = \frac{1}{2\Delta m} \, [m - 2n - 1] \, .$$

BELLCOMM, INC.

<div align="center">

APPENDIX III

</div>

This appendix contains listings of all main programs presently implemented in TSAP except for SAVE[1] and MATHCON[1] both to be in BAPL (BELLCOMM APPLICATIONS PROGRAM LIBRARY) on SYS*BLIB along with the TSAP INDEX.

@PRT	R.INDEX	
000001	000	DATE OF PRESENT ENTRY — 12/1/70
000002	000	DATE OF LAST ENTRY — 11/30/70
000003	000	
000004	000	PRESENT CONTENTS:
000005	000	
000006	000	0) SAVE — PROGRAM TO CONVERT A FORTRAN GENERATED DATA
000007	000	FILE TO SDF(SYSTEM DATA FORMAT) ELEMENT FILE.
000008	000	1) FILTER* — PROGRAM TO FILTER OR TRANSFORM A DATA SERIES.
000009	000	2) ACF — AUTO-COVARIANCE/CORRELATION PROGRAM.
000010	000	3) PACF — PARTIAL AUTO-CORRELATION PROGRAM.
000011	000	4) ASF* — AUTO-SPECTRUM PROGRAM.
000012	000	5) LSRA — LEAST SQUARES MULTIPLE REGRESSION PROGRAM
000013	000	WITH AN ANALYSIS OF VARIANCE PRINT OUT.
000014	000	6) XCF — CROSS-COVARIANCE/CORRELATION PROGRAM.
000015	000	7) XSF* — CROSS-SPECTRUM PROGRAM.
000016	000	8) COMB — PROGRAM TO COMBINE TWO TIME SERIES.
000017	000	9) MATHCON* — PROGRAM TO CONVERT MATH DATA SERIES TO
000018	000	FORTRAN DATA SERIES.
000019	000	10) LSRP¤ — LEAST SQUARES MULTIPLE REGRESSION PROGRAM
000020	000	WITH PREDICTION LIMITS.
000021	000	* — REVISED ENTRY
000022	000	¤ — NEW ENTRY

---

[1] Both programs were written by C. Mee.

R.FILTER,J.J
000 COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:28:17.

N PROGRAM

RAGE USED: CODE(1) 001565; DATA(0) 002410; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

03	OPTION
04	CSF
05	EXIT
06	NINTRS
07	NRDUS
10	NI01S
11	N102S
12	NWDUS
13	ALOG
14	NEXP2S
15	ALOG1U.
16	SQRT
17	EXP
20	COS
21	NEXP1S
22	SIN
23	NSTOPS

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

01	000030 1L	0001	000245 10L	0000	002012 100F	0000	002065 1
00	002076 103F	0000	002103 104F	0000	002114 105F	0000	002125 1
00	002141 108F	0000	002234 109F	0000	002274 110F	0000	002300 I
00	002312 112F	0000	002321 113F	0000	C02177 114F	0000	002150 1
00	002203 117F	0000	002211 118F	0000	002222 119F	0001	000264 1
00	002255 121F	0000	002324 122F	0000	002246 123F	0000	002155 1
00	002106 140F	0001	000071 142G	0001	000111 153G	0001	000325 1
01	000345 18L	0001	000132 2L	0001	000365 20L	0000	002074 2
01	000204 211G	0001	000424 22L	0001	000221 221G	0001	000235 2
01	000257 242G	0001	000614 25L	0001	000274 252G	0001	000315 2
01	000335 272G	0001	000746 28L	0001	001022 29L	0001	001270 3
01	000420 321G	0001	001440 33L	0001	000500 346G	0001	000545 3
01	001066 455G	0001	001117 463G	0001	001355 50L	0001	001531 5
01	001336 523G	0001	001554 53L	0001	001430 554G	0001	001500 5
01	001517 606G	0001	000225 8L	0000 R	001777 A	0000 R	002001 A
00 R	002005 CP	0000 R	002003 DEL	0000 R	001750 FMI	0000 R	001756 F
00 I	001773 IFL	0000 I	002000 II	0000 I	002006 ISO	0000 I	001766 I
00 I	001775 K	0000 I	001767 L	0000 I	001776 M	0000 I	001770 N
03 L	000000 OPTION	0000 R	002007 SF	0000 R	001772 SM	0000 R	002004 W

| 1* | C |
| 2* | C TITLE | FILTER - A PROGRAM TO FILTER OR TRANSFORM A |

```
 3*     C                          DATA SERIES.
 4*     C
 5*     C AUTHOR         R.R.SINGERS
 6*     C
 7*     C SPONSOR        G.R.ANDERSEN
 8*     C
 9*     C DATE           AUGUST,1970
10*     C
11*     C KEY WORDS      FILTER
12*     C               TRANSFORMATION
13*     C
14*     C PURPOSE        TO APPLY ONE OR MORE OF A LIST OF FILTERS OR
15*     C                TRANSFORMS TO A DATA SERIES.
16*     C
17*     C METHOD         THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
18*     C                ASKING FOR CERTAIN PARAMETERS AND APPLYING THEM TO
19*     C                THE INPUT DATA SERIES. THE PROCESS IS SEQUENTIAL
20*     C                ALLOWING MORE THAN ONE FILTER OR TRANSFORMATION
21*     C                TO BE APPLIED TO THE SAME SERIES.
22*     C
23*     C NOTE           1) A DYNAMICALLY ASSIGNED FILE (UNIT 8) IS
24*     C                USED TO STORE THE FINAL SERIES.
25*     C
26*                      PARAMETER MX=1000
27*                      DIMENSION X(MX),FMI(6),FMO(6)
28*                      DIMENSION ASG8(2)/'@ASG,T 8,F'/
29*                      LOGICAL OPTION
30*                      READ(5,201,END=53) FMI
31*                      IF(OPTION('N'))GO TO 1
32*                      L=MX
33*                      WRITE(6,100)
34*     100              FORMAT(/' ''FILTER'' CONTAINS THE FOLLOWING FILTERS:'/' C
35*            .ENTER,EXP,LOG,DIFF,10**,LOG10,SQRT,LAG,ABS,TUKEY(HIGH AND LOW PASS
36*            .),GOODMAN,'/' SCALE.'/' TO TERMINATE THE FILTER SEQUENCE,REPLY WIT
37*            .H ''END''.'/' THE FINAL SERIES IS STORED ON UNIT 8(IF DESIRED).')
38*     1                WRITE(6,101)
39*     101              FORMAT(' LENGTH OF SERIES TO BE READ IN?')
40*                      READ(5,200,END=53)N
41*     200              FORMAT( )
42*                      IF(N.GT.MX)GO TO 51
43*     201              FORMAT(6A6)
44*                      WRITE(6,103)
45*     103              FORMAT(' INPUT DATA FORMAT?')
46*                      READ(5,201,END=53)FMI
47*                      WRITE(6,104)
48*     104              FORMAT(' INPUT DATA:')
49*                      READ(5,FMI,END=53)(X(I),I=1,N)
50*                      WRITE(6,140)
51*     140              FORMAT(' LENGTH OF SERIES TO BE USED?')
52*                      READ(5,200,END=53)N
53*     2                SM=0.
54*                      DO 3 I=1,N
55*     3                SM=SM+X(I)
56*                      SM=SM/N
57*                      WRITE(6,105)SM,N
58*     105              FORMAT(' MEAN=',E15.8/' LENGTH=',I5///' FILTER?')
59*     4                READ(5,201,END=53) IFL
```

```
 60*                              IF(IFL.EQ.'END    ')GO TO 50
 61*                              IF(IFL.NE.'CENTER')GO TO 6
 62*                              DO 5 I=1,N
 63*         5                    X(I)=X(I)-SM
 64*                              GO TO 2
 65*         6                    IF(IFL.NE.'ABS    ')GO TO 8
 66*                              DO 7 I=1,N
 67*         7                    X(I)=ABS(X(I))
 68*                              GO TO 2
 69*         8                    IF(IFL.NE.'LOG    ')GO TO 10
 70*                              DO 9 I=1,N
 71*         9                    X(I)=ALOG(X(I))
 72*                              GO TO 2
 73*        10                    IF(IFL.NE.'DIFF   ')GO TO 12
 74*                              N=N-1
 75*                              DO 11 I=1,N
 76*        11                    X(I)=X(I+1)-X(I)
 77*                              GO TO 2
 78*        12                    IF(IFL.NE.'10**   ')GO TO 14
 79*                              DO 13 I=1,N
 80*        13                    X(I)=10**X(I)
 81*                              GO TO 2
 82*        14                    IF(IFL.NE.'LOG10 ')GO TO 16
 83*                              DO 15 I=1,N
 84*        15                    X(I)=ALOG10(X(I))
 85*                              GO TO 2
 86*        16                    IF(IFL.NE.'SQRT   ')GO TO 18
 87*                              DO 17 I=1,N
 88*        17                    X(I)=SQRT(X(I))
 89*                              GO TO 2
 90*        18                    IF(IFL.NE.'EXP    ')GO TO 20
 91*                              DO 19 I=1,N
 92*        19                    X(I)=EXP(X(I))
 93*                              GO TO 2
 94*        20                    IF(IFL.NE.'LAG    ')GO TO 22
 95*                              WRITE(6,106)
 96*       106                    FORMAT(' NUMBER OF LAGS?')
 97*                              READ(5,200,END=53)J
 98*                              N=N-J
 99*                              DO 21 I=1,N
100*        21                    X(I)=X(I+J)
101*                              GO TO 2
102*        22                    IF(IFL.NE.'TUKEY ')GO TO 25
103*                              WRITE(6,107)
104*       107                    FORMAT(' LENGTH OF MOVING AVERAGE(ODD NUMBER)?')
105*                              READ(5,200,END=53)J
106*                              K=(J-1)/2
107*                              WRITE(6,108)
108*       108                    FORMAT(' HIGH PASS(HIGH) OR LOW PASS(LOW)?')
109*                              READ(5,201,END=53)M
110*                              DO 24 I=1,N
111*                              IF(I.LE.K.OR.I.GT.N-K)GO TO 24
112*                              A=0.
113*                              DO 23 II=-K,K
114*                              AL=-1./(K+1.)*(.5+.5*COS(II*3.14159/(K+1.)))
115*                              IF(II.EQ.0)AL=1.-1./(K+1.)
116*        23                    A=A+AL*X(I+II)
```

```
117*                  IF(M.EQ.'LOW   ')A=X(I)-A
118*                  X(I-K)=A
119*      24          CONTINUE
120*                  N=N-J+1
121*                  GO TO 2
122*      25          IF(IFL.NE.'GOODMA')GO TO 30
123*                  WRITE(6,107)
124*                  READ(5,200,END=53) J
125*                  K=(J-1)/2
126*                  WRITE(6,115)
127*      115         FORMAT(' NUMBER OF TRIANGLES?')
128*                  READ(5,200,END=53) NO
129*                  WRITE(6,124)
130*      124         FORMAT(' DATA SPACING?')
131*                  READ(5,201,END=53)DEL
132*                  WRITE(6,116)
133*      116         FORMAT(' HIGH PASS(HIGH), LOW PASS(LOW), PASS BAND',
134*             .          '(PASS) OR REJECT BAND(REJECT)?')
135*                  READ(5,201,END=53)M
136*                  IF(M.EQ.'HIGH'.OR.M.EQ.'LOW') GO TO 28
137*                  WRITE(6,114)
138*      114         FORMAT(' CENTER FREQUENCY?')
139*                  READ(5,200,END=53) W
140*                  W=W*DEL*6.28318
141*                  GO TO 29
142*      28          CP=.5*(K-1.-2.*NO)/K/DEL
143*                  WRITE(6,117) CP
144*      117         FORMAT(' CUT-OFF FREQUENCY IS ',F10.7)
145*                  WRITE(6,118)
146*      118         FORMAT(' DO YOU WANT TO CHANGE INPUT VALUES(YES/NO)?')
147*                  READ(5,201,END=53)I
148*                  IF(I.EQ.'YES') GO TO 25
149*                  W=3.14159*(1.-FLOAT(NO)/FLOAT(K))
150*      29          DO 27 I=1,N
151*                  IF(I.LE.K.OR.I.GT.N-K) GO TO 27
152*                  A=0.
153*                  DO 26 II=-K,K
154*                  IF(II.EQ.ABS(K))AL=(-1)**(NO+1)*.08*NO/K*COS(K*W)
155*                  IF(II.EQ.0) AL=2.*NO/K
156*                  IF(II.NE.0.AND.II.NE.ABS(K))AL=2./K*(.54+.46*COS(
157*             .    II*3.14159/K))*COS(II*W)*SIN(NO*II*3.14159/K)/SIN
158*             .    (3.14159*II/K)
159*      26          A=A+AL*X(I+II)
160*                  IF(M.EQ.'LOW'.OR.M.EQ.'REJECT')A=X(I)-A
161*                  X(I-K)=A
162*      27          CONTINUE
163*                  N=N-J+1
164*                  GO TO 2
165*      30          IF(IFL.NE.'SCALE')GO TO 52
166*                  WRITE(6,119)
167*      119         FORMAT(' SCALE OPERATOR(ADD/MPY)?')
168*                  READ(5,201)ISO
169*                  WRITE(6,120)
170*      120         FORMAT(' SCALE FACTOR?')
171*                  READ(5,200)SF
172*                  DO 31 I=1,N
173*                  IF(ISO.EQ.'ADD')X(I)=X(I)+SF
```

```
I74*     31              IF(ISO.EQ.'MPY')X(I)=X(I)*SF
175*                     GO TO 2
176*     50              WRITE(6,109)
177*     109             FORMAT(' DO YOU WANT TO SEE PART OF THE SERIES',
178*        .            '(YES/NO)?')
179*                     READ(5,201,END=53)IFL
180*                     IF(IFL.NE.'YES')GO TO 33
181*                     WRITE(6,123)
182*     123             FORMAT(' WHAT PART(INITIAL,FINAL,STEP)?')
183*                     READ(5,200,END=53)I,J,K
184*                     DO 32 L=I,J,K
185*     32              WRITE(6,200)X(L)
186*     33              WRITE(6,121)
187*     121             FORMAT(' DO YOU WANT THE FINAL SERIES STORED',
188*        .            '(YES/NO)?')
189*                     READ(5,201,END=53)IFL
190*                     IF(IFL.NE.'YES  ')GO TO 53
191*                     WRITE(6,102)
192*     102             FORMAT(' OUTPUT DATA FORMAT?')
193*                     READ(5,201,END=53)FMO
194*                     CALL CSF(2,ASGB,I)
195*                     WRITE(8,FMO)(X(I),I=1,N)
196*                     WRITE(6,110)
197*     110             FORMAT(' SERIES STORED.')
198*                     GO TO 53
199*     51              WRITE(6,111)L
200*     111             FORMAT(' LENGTH OF SERIES MUST BE LESS THAN',I5,',TRY AGA
201*        .IN.')
202*                     GO TO 1
203*     52              WRITE(6,112)
204*     112             FORMAT(' FILTER NOT RECOGNIZED,TRY AGAIN.')
205*                     WRITE(6,113)
206*     113             FORMAT(' FILTER?')
207*                     GO TO 4
208*     53              WRITE(6,122)
209*     122             FORMAT(' FILTER TERMINATES.')
210*                     CALL EXIT
211*             END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

R.ACF,J,J

000 COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:28:49.

N PROGRAM

RAGE USED: CODE(1) 000501; DATA(0) 004145; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

```
03    OPTION
04    CSF
05    EXIT
06    NINTR$
07    NRDU$
10    NIO1$
11    NIO2$
12    NREW$
13    NWDU$
14    NABU$
15    NWEF$
16    NSTOP$
```

RAGE ASSIGNMENT   (BLOCK, TYPE, RELATIVE LOCATION, NAME)

01	000040 1L	0001	000422 10L	0000	004004 100F	0000	004017 1
00	004033 103F	0000	004040 104F	0000	004043 105F	0000	004046 1
00	004055 108F	0000	004066 109F	0000	004073 110F	0000	004105 1
00	004121 122F	0001	000160 163G	0001	000200 174G	0001	000061 2
00	004023 201F	0001	000211 202G	0001	000226 207G	0001	000274 2
01	000340 251G	0001	000103 3L	0001	000416 301G	0001	000431 3
01	000452 51L	0001	000462 52L	0001	000470 53L	0000	003750 9
00 R	003726 AC	0000 R	003744 B	0000 R	000000 C	0000 R	003720 F
00 I	003745 IC	0000 I	003733 ICO	0000 I	003743 II	0000 I	003730 I
00 I	003731 L	0000 I	003735 M	0000 I	003742 MR	0000 I	003736 N
00 I	003737 NI	0003 L	000000 OPTION	0000 R	001750 Y		

```
 1*   C
 2*   C TITLE      ACF - AUTO-COVARIANCE/CORRELATION PROGRAM.
 3*   C
 4*   C AUTHOR     R.R.SINGERS
 5*   C
 6*   C SPONSOR    G.R.ANDERSEN
 7*   C
 8*   C DATE       AUGUST,1970
 9*   C
10*   C KEY WORDS  COVARIANCE
11*   C            CORRELATION
12*   C
13*   C PURPOSE    TO COMPUTE THE AUTO-COVARIANCES OF A DATA SERIES
14*   C            AND IF DESIRED, THE AUTO-CORRELATIONS.
15*   C
```

```
16*     C METHOD         THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
17*     C               ASKING FOR CERTAIN PARAMETERS AND A DATA SERIES.
18*     C               THE STANDARD METHOD OF COMPUTING THE AUTO-COVARIANCES
19*     C               IS THEN APPLIED.
20*     C
21*     C NOTE          1) A DYNAMICALLY ASSIGNED FILE (UNIT 1) IS
22*     C               USED TO SAVE THE AUTO-COVARIANCES (WHEN
23*     C               SPECIFIED) FOR LATER USE.
24*     C
25*                     PARAMETER MX=1000
26*                     DIMENSION C(MX),Y(MX),FMT(6),AC(2)/'@ASG,T 1,F'/
27*                     LOGICAL OPTION
28*                     L=MX
29*                     READ(5,200,END=53)FMT
30*                     CALL CSF(2,AC,I)
31*                     REWIND 1
32*                     IF(OPTION('N'))GO TO 1
33*                     WRITE(6,99)
34*     99              FORMAT(/' ACF COMPUTES THE AUTO-COVARIANCE/CORRELATION FU
35*            .NCTION OF A SERIES.'/' THE COVARIANCES ARE STORED ON UNIT 1 IN BINA
36*            .RY(IF DESIRED)'/' IN CONSECUTIVE ORDER.')
37*     1               WRITE(6,100)
38*     100             FORMAT(/' DO YOU WANT COVARIANCE(COV),CORRELATION(COR) OR
39*            . END?')
40*                     READ(5,200,END=53)ICO
41*     200             FORMAT(6A6)
42*                     IF(ICO.EQ.'END')GO TO 50
43*     2               WRITE(6,101)
44*     101             FORMAT(' LENGTH OF SERIES?')
45*                     READ(5,201,END=53)N
46*     201             FORMAT( )
47*                     IF(N.GT.MX)GO TO 51
48*     3               WRITE(6,102)
49*     102             FORMAT(' NUMBER OF LAGS(ZERO INCLUDED)?')
50*                     READ(5,201,END=53)M
51*                     IF(M.GT.N)GO TO 52
52*     4               WRITE(6,103)
53*     103             FORMAT(' INPUT DATA FORMAT?')
54*                     READ(5,200,END=53)FMT
55*                     WRITE(6,104)
56*     104             FORMAT(' INPUT DATA:')
57*                     READ(5,FMT,END=53)(Y(I),I=1,N)
58*                     M1=M-1
59*                     DO 6 I=0,M1
60*                     NI=N-I
61*                     A=0.
62*                     DO 5 J=1,NI
63*     5               A=A+Y(J)*Y(J+I)
64*                     C(I+1)=A/N
65*     6               IF(ICO.EQ.'COR'.AND.I.NE.0)C(I+1)=C(I+1)/C(1)
66*                     A=C(1)
67*                     IF(ICO.EQ.'COR') C(1)=1.
68*                     MR=N/4
69*                     II=MR+1
70*                     IF(MR.GT.10)MR=10
71*                     DO 7 I=0,MR
72*     7               WRITE(6,105)(J,C(J+1),J=I,M1,II)
```

```
73*      105          FORMAT(4(I4,E15.8))
74*                   IF(ICO.EQ.'COR')WRITE(6,106)A
75*      106          FORMAT(' VARIANCE=',E15.8)
76*                   B=0.
77*                   DO 8 I=1,N
78*      8            B=B+Y(I)
79*                   B=B/N
80*                   WRITE(6,107)B
81*      107          FORMAT(' MEAN=',E15.8)
82*                   WRITE(6,108)
83*      108          FORMAT(' DO YOU WANT THE COVARIANCES STORED(YES/NO)?')
84*                   READ(5,200,END=53)IC
85*                   IF(IC.NE.'YES')GO TO 1
86*                   WRITE(1)N,M
87*                   IF(ICO.NE.'COR')GO TO 10
88*                   DO 9 I=1,M
89*      9            C(I)=A*C(I)
90*      10           WRITE(1)(C(I),I=1,M)
91*                   WRITE(6,109)
92*      109          FORMAT(' COVARIANCE STORED.')
93*                   GO TO 1
94*      50           END FILE 1
95*                   REWIND 1
96*                   GO TO 53
97*      51           WRITE(6,110)L
98*      110          FORMAT(' SERIES LENGTH MUST BE LESS THAN',I10,' -TRY AGAI
99*          .N.')
100*                  GO TO 2
101*     52           WRITE(6,111)
102*     111          FORMAT(' NUMBER OF LAGS CANNOT BE GREATER THAN LENGTH OF
103*          .SERIES-TRY AGAIN.')
104*                  GO TO 3
105*     53           WRITE(6,122)
106*     122          FORMAT(' ACF TERMINATES.')
107*                  CALL EXIT
108*            END
```

END OF COMPILATION: NO DIAGNOSTICS.

R.PACF,J.J

)00 COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:29:01.

N PROGRAM

RAGE USED: CODE(1) 000555; DATA(0) 007457; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

03	OPTION
04	EXIT
05	NINTR$
06	NRDU$
07	NIO1$
10	NIO2$
11	NWDU$
12	NRBU$
13	NSTOP$

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

)1	000026 1L	0001	000413 10L	0000	007253 100F	0000	007317 1				
)0	007330 104F	0000	007343 105F	0000	007376 106F	0000	007405 1				
)0	007350 109F	0000	007363 110F	0001	000011 110G	0000	007420 1				
)1	000110 147G	0001	000155 157G	0001	000213 172G	0000	007250 2				
)1	000243 206G	0001	000273 224G	0001	000310 230G	0001	000350 2(				
)1	000455 277G	0001	000470 304G	0001	000525 323G	0001	000354 5(				
)1	000544 53L	0000 R	000000 A	0000 D	007224 A0	0000 D	007226 A				
)0 D	007232 DEN	0000 R	007210 FMT	0000 I	007240 I	0000 I	007234 I'				
)0 I	007241 K	0000 I	007246 K0	0000 I	007242 L	0000 I	007243 L				
)0 I	007247 MA	0000 I	007237 ML	0000 I	007245 M1	0000 I	007236 N				
)3 L	000000 OPTION	0000 D	007216 R0	0000 D	007220 R1	0000 D	007222 R				

```
 1*    C
 2*    C TITLE       PACF - PARTIAL AUTO-CORRELATION PROGRAM.
 3*    C
 4*    C AUTHOR      R.R.SINGERS
 5*    C
 6*    C SPONSOR     G.R.ANDERSEN
 7*    C
 8*    C DATE        AUGUST,1970
 9*    C
10*    C KEY WORDS   PARTIAL AUTO-CORRELATION
11*    C
12*    C PURPOSE     TO COMPUTE THE PARTIAL AUTO-CORRELATION
13*    C             COEFFICIENTS AND THE CORRESPONDING VARIANCES
14*    C             OF UP TO 30 LAGS GIVEN THE AUTO-COVARIANCES.
15*    C             THE FIRST ORDER MOVING AVERAGE COEFFICIENT
16*    C             IS ALSO COMPUTED.
17*    C
18*    C METHOD      THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
```

```
19*     C                        ASKING FOR CERTAIN PARAMETERS. IT IS ASSUMED
20*     C                        THAT ACF WAS EXECUTED AND THE AUTO-COVARIANCES
21*     C                        SAVED. THE STANDARD METHOD OF COMPUTATION IS USED.
22*     C
23*     C NOTE                   1) THE AUTO-COVARIANCES ARE ASSUMED TO
24*     C                        BE STORED (ON UNIT 1) FROM ACF.
25*     C
26*                             PARAMETER MX=60
27*                             DIMENSION A(MX,MX),CO(MX),V(MX),FMT(6)
28*                             DOUBLE PRECISION R0,R1,R2,A0,A1,NUM,DEN
29*                             LOGICAL OPTION
30*                             READ(5,200,END=53)FMT
31*     200                     FORMAT(6A6)
32*                             IF(OPTION('N'))GO TO 1
33*                             WRITE(6,100)
34*     100                     FORMAT(' PACF COMPUTES THE PARTIAL AUTO-CORRELATION AND A
35*              ..R. COEFFICIENTS'/' UP TO ORDER K TOGETHER WITH THE RESIDUAL',
36*              .' VARIANCES OF THE'/' FITTED A.R.''S UP TO ORDER K AND THE FIRST',
37*              .' ORDER M.A. COEFFICIENTS.')
38*     1                       WRITE(6,101)
39*     101                     FORMAT(/' MAXIMUM ORDER OF K?')
40*                             READ(5,201,END=53)M
41*     201                     FORMAT( )
42*                             IF(M.GT.MX)GO TO 51
43*                             READ(1)N,ML
44*                             READ(1)(CO(I),I=1,ML)
45*                             A(1,1)=CO(2)/CO(1)
46*                             DO 4 K=3,M
47*                             L=K-1
48*                             L1=L-1
49*                             R0=CO(1)
50*                             NUM=CO(K)/R0
51*                             DEN=1.D0
52*                             DO 2 I=1,L1
53*                             A1=A(L1,I)
54*                             R1=CO(L-I+1)
55*                             R2=CO(I+1)
56*                             NUM=NUM-A1*R1/R0
57*     2                       DEN=DEN-A1*R2/R0
58*                             A0=NUM/DEN
59*                             A(L,L)=A0
60*                             DO 3 J=1,L1
61*     3                       A(L,J)=A(L1,J)-A(L,L)*A(L1,L-J)
62*     4                       CONTINUE
63*                             M1=M-1
64*                             WRITE(6,103)
65*     103                     FORMAT(/' PACF:')
66*                             WRITE(6,104)(A(J,J),J=1,M1)
67*     104                     FORMAT(/(5E15.8))
68*                             WRITE(6,108)
69*     108                     FORMAT(' DO YOU WANT RESIDUAL VARIANCES',
70*              .               '(YES/NO)?')
71*                             READ(5,200,END=53)I
72*                             IF(I.NE.'YES')GO TO 5
73*                             DO 7 L=1,M1
74*                             V(L)=CO(1)
75*                             DO 6 J=1,L
```

```
76*       6              V(L)=V(L)-A(L,J)*CO(J+1)
77*       7              V(L)=(N-L)/(N-2.*L-1.)*V(L)
78*                      WRITE(6,105)
79*       105            FORMAT(/' RESIDUAL VARIANCES:')
80*                      WRITE(6,104)(V(I),I=1,M1)
81*       5              WRITE(6,109)
82*       109            FORMAT(' ORDER OF A.R. PROCESS TO BE VIEWED',
83*       .              '(0=NONE OR NO MORE)?')
84*                      READ(5,201,END=53)K0
85*                      IF(K0.EQ.0)GO TO 10
86*                      WRITE(6,104)(A(K0,I),I=1,K0)
87*                      GO TO 5
88*       10             WRITE(6,110)
89*       110            FORMAT(' DO YOU WANT THE FIRST ORDER M.A.',
90*       .              ' COEFFICIENTS(YES/NO)?')
91*                      READ(5,200,END=53) MA
92*                      IF(MA.NE.'YES') GO TO 53
93*                      DO 9 K=1,M1
94*                      DEN=1.D0
95*                      NUM=A(M1,1)
96*                      DO 8 I=1,K
97*                      R1=A(M1,1)
98*                      R2=A(M1,I+1)
99*                      DEN=DEN+R1**2
100*      8              NUM=NUM+R1*R2
101*                     DEN=DEN+(DBLE(A(M1,M1)))**2
102*      9              V(K)=NUM/DEN
103*                     WRITE(6,106)
104*      106            FORMAT(/' FIRST ORDER M.A. COEFFICIENTS:')
105*                     WRITE(6,104)(V(I),I=1,M1)
106*                     GO TO 53
107*      51             L=MX
108*                     WRITE(6,107)L
109*      107            FORMAT(' NUMBER OF LAGS CANNOT BE GREATER THAN',I3,' -TRY
110*      .AGAIN.')
111*                     GO TO 1
112*      53             WRITE(6,111)
113*      111            FORMAT(' PACF TERMINATES.')
114*                     CALL EXIT
115*              END
```

END OF COMPILATION: NO DIAGNOSTICS.

SINGERS/TSAP - TIME SERIES ANALYSIS PACKAGE//

R.ASF,J,J

JUU COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:29:30.

N PROGRAM

RAGE USED: CODE(1) 001037; DATA(0) 004532; BLANK COMMON(2) 000000

LRNAL REFERENCES (BLOCK, NAME)

J3	OPTION
J4	CSF
J5	EXIT
J6	NINTRS
J7	NRDU$
.U	N101$
.1	N102$
.2	NWDU$
.3	NRBU$
.4	COS
.5	SIN
.6	SQRT
.7	NWBU$
.0	NWEF$
.1	NSTOP$

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

J1	000151 1L	0000	004250 100F	0000	004317 102F	0000	004327 1
J0	004377 105F	0000	004407 106F	0000	004413 107F	0000	004440 1
J0	004454 110F	0000	004334 111F	0000	004463 112F	0000	004472 1
J0	004353 115F	0000	004313 116F	0001	000011 121G	0001	000170 2
J0	004326 201F	0001	000221 214G	0001	000322 240G	0001	000347 2
J1	000500 302G	0001	000637 327G	0001	000754 371G	0001	001026 5
J1	001023 7L	0001	000026 99L	0000 R 004214 ANU	0000 R 003722 A
J0 R 000002 C	0000 R 003733 CHI	0000 R 004224 CL	0000 R 004231 C
J0 R 004210 DEL	0000 R 004242 EF	0000 R 004235 EL1	0000 R 004213 E
J0 R 001752 F	0000 R 004226 FM	0000 R 003725 FMT	0000 R 004241 F
J0 I 004220 I	0000 I 004222 ICL	0000 I 004244 IF	0000 I 004205 I
J0 I 004243 IU	0000 I 004211 IW	0000 I 003724 I$	0000 I 004227 K
J0 I 004212 M	0000 I 004206 N	0000 I 004216 NM	0000 I 004207 N
J0 I 004225 NU	0003 L 000000 OPTION	0000 R 000001 PI	0000 R 000000 P
J0 R 004245 VR	0000 R 004232 V0	0000 R 004233 V1	0000 R 004237 V

1*	C		
2*	C TITLE	ASF - AUTO-SPECTRUM PROGRAM.	
3*	C		
4*	C AUTHOR	R.R.SINGERS	
5*	C		
6*	C SPONSOR	G.R.ANDERSEN	
7*	C		
8*	C DATE	AUGUST,1970	

```
 9*     C
10*     C KEY WORDS      AUTO-SPECTRUM
11*     C               SPECTRAL DENSITY
12*     C
13*     C PURPOSE        TO COMPUTE THE AUTO-SPECTRUM OF A DATA SERIES
14*     C               AND TO PLOT THE SPECTRUM WITH ITS CONFIDENCE
15*     C               LIMITS.
16*     C
17*     C METHOD         THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
18*     C               ASKING FOR CERTAIN PARAMETERS AND A DATA SERIES.
19*     C               IT IS ASSUMED THAT ACF WAS EXECUTED AND THE
20*     C               AUTO-COVARIANCES SAVED. THE STANDARD METHOD OF
21*     C               COMPUTATION IS USED.
22*     C
23*     C NOTES          1) THE COVARIANCES ARE ASSUMED TO BE
24*     C               STORE (ON UNIT 1) FROM ACF.
25*     C               2) A DYNAMICALLY ASSIGNED FILE ( UNIT 2)
26*     C               IS USED TO STORE THE AUTO-SPECTRUM
27*     C               (WHEN SPECIFIED).
28*     C
29*                     PARAMETER MX=1000
30*                     DATA PIV,PI/.1591549,3.1415926/
31*                     DIMENSION C(MX),F(MX),AS(2)/'QASG,T 2,F'/,FMT(6)
32*                     DIMENSION CHI(60,2)/
33*           . 0.004,0.103,0.352,0.711,1.145,1.635,2.167,2.733,3.325,3.940,4.575
34*           . ,5.226,5.392,6.571,7.261,7.962,8.672,9.390,10.117,10.851,11.591,1
35*           . 2.338,13.091,13.848,14.611,15.379,16.151,16.928,17.708,18.493,19.
36*           . 281,20.072,20.867,21.664,22.465,23.269,24.075,24.884,25.695,26.50
37*           . 9,27.326,28.144,28.965,29.787,30.612,
38*           . 31.439,32.268,33.098,33.930,34.764,35.6,36.437,37.276,38.116,38.9
39*           . 58,39.801,40.646,41.492,42.339,43.188,
40*           .                             3.841,5.991,7.815,9.488,11.0
41*           . 71,12.592,14.067,15.507,16.919,18.307,19.675,21.026,22.362,23.685
42*           . ,24.996,26.296,27.587,28.869,30.144,31.410,32.671,33.924,35.172,3
43*           . 6.415,37.652,38.885,40.113,41.337,42.557,43.773,44.985,46.194,47.
44*           . 400,48.602,49.802,50.999,52.192,53.384,54.572,55.758,56.942,58.12
45*           . 4,59.304,60.481,61.656,
46*           . 62.83,64.001,65.171,66.339,67.505,68.669,69.832,70.993,72.153,73.
47*           . 311,74.468,75.624,76.778,77.931,79.082/
48*                     DIMENSION IPT(50)/50*' '/
49*                     DATA IL/'NO'/
50*                     LOGICAL OPTION
51*                     READ(5,201,END=53)FMT
52*                     IF(OPTION('N'))GO TO 99
53*                     WRITE(6,100)
54*     100             FORMAT(/' ASF COMPUTES THE AUTO SPECTRUM FUNCTION.'/
55*           .          ' IT IS ASSUMED THAT ACF WAS CALLED AND THE DESIRED'
56*           .          ' COVARIANCES STORED.'/' THE AUTO-SPECTRA ARE'
57*           .          ' STORED ON UNIT 2(IF DESIRED) CONSECUTIVELY.')
58*     99              IF(IL.NE.'YES')READ(1)N,NN
59*     200             FORMAT( )
60*                     WRITE(6,116)
61*     116             FORMAT(' DATA SPACING?')
62*                     READ(5,200,END=53) DEL
63*                     WRITE(6,102)
64*     102             FORMAT(' WINDOW TO BE USED(TUKEY/DANIELL)?')
65*                     READ(5,201,END=53)IW
```

```
66*     201             FORMAT(6A6)
67*                     WRITE(6,103)
68*     103             FORMAT(' BANDWIDTH PARAMETER?')
69*                     READ(5,200,END=53)M
70*                     IF(IW.EQ.'TUKEY')GO TO 1
71*                     EM1=M-1
72*                     ANU=2*N/EM1
73*                     AU=1.
74*                     WRITE(6,111)
75*     111             FORMAT(' NUMBER OF COVARIANCES TO BE USED?')
76*                     READ(5,200,END=53) NM
77*                     GO TO 2
78*     1               EM1=M-1.
79*                     ANU=2.667*N/EM1
80*                     AU=.75
81*                     NM=M
82*     2               WRITE(6,104)
83*     104             FORMAT(' NUMBER OF FREQUENCY POINTS TO BE VIEWED?')
84*                     READ(5,200,END=53)NQ
85*                     IF(IL.NE.'YES')READ(1)(C(I),I=1,NN)
86*                     ENQ1=NQ-1
87*                     WRITE(6,115)
88*     115             FORMAT(' DO YOU WANT CONFIDENCE LIMITS(YES/NO)?')
89*                     READ(5,201,END=53)ICL
90*                     CU=1.
91*                     CL=1.
92*                     IF(ICL.NE.'YES')GO TO 25
93*                     NU=IFIX(ANU+.5)
94*                     CU=ANU/CHI(NU,1)
95*                     CL=ANU/CHI(NU,2)
96*     25              FM=0.
97*                     DO 4 K=1,NQ
98*                     TAU=K-1
99*                     CO=COS(PI*TAU/ENQ1)
100*                    V0=0.
101*                    V1=0.
102*                    DO 3 L=NM,2,-1
103*                    EL1=L-1
104*                    W=1.
105*                    IF(IW.EQ.'DANIEL')W=SIN(EL1*PI/EM1)/(EL1*PI/EM1)
106*                    IF(IW.EQ.'TUKEY')W=.5*(1.+COS(PI*EL1/EM1))
107*                    V2=2.*CO*V1-V0+W*C(L)
108*                    V0=V1
109*    3               V1=V2
110*                    F(K)=DEL*(C(1)+2.*(V1*CO-V0))
111*    4               FM=AMAX0(FM,F(K))
112*                    FX=FM*CU/49.
113*                    WRITE(6,114)
114*    114             FORMAT(' DO YOU WANT FREQUENCY(FREQ.) OR'
115*          .         ' PERIOD(PERIOD) DISPLAYED?')
116*                    READ(5,201,END=53)FP
117*                    WRITE(6,105)FP
118*    105             FORMAT(' AUTO-SPECTRUM:'/2X,A6,'  SPECTRUM')
119*                    DO 5 I=1,NQ
120*                    L=I-1
121*                    IF(FP.NE.'PERIOD')EF=.5*L/ENQ1/DEL
122*                    IF(FP.EQ.'PERIOD'.AND.L.EQ.0)EF=99.9999
```

```
123*                    IF(FP.EQ.'PERIOD'.AND.L.NE.0)EF=2.*ENQI/L*DEL
124*                    IU=IFIX(F(I)*CU/FX)+1
125*                    IL=IFIX(F(I)*CL/FX)+1
126*                    IF=IFIX(F(I)/FX)+1
127*                    IF(IU.GT.50)IU=50
128*                    IPT(IU)='+'
129*                    IPT(IL)='*'
130*                    IPT(IF)='.'
131*                    WRITE(6,106)EF,F(I),IPT
132*     106            FORMAT(F8.4,E12.5,2X,50A1)
133*                    IPT(IU)=' '
134*                    IPT(IF)=' '
135*     5              IPT(IL)=' '
136*                    VR=SQRT(AU*M/N)*100.
137*                    IF(ICL.EQ.'YES')WRITE(6,107)NU,CU,CL,VR
138*     107            FORMAT(' THE 90% CONFIDENCE LIMITS FOR NU=',I2,' ARE: CU=
139*          ,'F10.5,' ,CL=',F10.5/' THE SQRT OF VARIANCE RATIO IS:',F8.3,' %')
140*                    WRITE(6,108)
141*     108            FORMAT(' DO YOU WANT THE SPECTRUM STORED(YES/NO)?')
142*                    READ(5,201,END=53)IW
143*                    IF(IW.NE.'YES')GO TO 6
144*                    CALL CSF(2,AS,I)
145*                    WRITE(2)M,DEL,NQ
146*                    WRITE(2)(F(I),I=1,NQ)
147*                    WRITE(6,109)
148*     109            FORMAT(' SPECTRUM STORED.')
149*     6              WRITE(6,110)
150*     110            FORMAT(' DO YOU HAVE ANOTHER SERIES(YES/NO)?')
151*                    READ(5,201,END=53)IW
152*                    IF(IW.NE.'YES')GO TO 7
153*                    WRITE(6,112)
154*     112            FORMAT(' IS THE SERIES THE SAME(YES/NO)?')
155*                    READ(5,201) IL
156*                    GO TO 99
157*     7              END FILE 2
158*     53             WRITE(6,113)
159*     113            FORMAT(' ASF TERMINATES.')
160*                    CALL EXIT
161*            END
```

END OF COMPILATION: NO DIAGNOSTICS.

SINGERS/TSAP - TIME SERIES ANALYSIS PACKAGE.//

    R.LSRA,J.J
000 COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:29:18.

N PROGRAM

RAGE USED: CODE(1) 001076; DATA(0) 007410; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

03	OPTION
04	RESTEM
05	EXIT
06	CSF
07	NINTR$
10	NRDU$
11	NIO1$
12	NIO2$
13	NWDU$
14	SQRT
15	NSTOP$

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

01	000026 1L	0001	000617 10L	0000	007003 100F	0000	007042 1
00	007217 103F	0000	007053 105F	0000	007056 107F	0000	007064 1
00	007102 110F	0001	000011 110G	0000	007116 111F	0000	007124 1
00	007137 114F	0000	007145 115F	0000	007151 116F	0000	007156 1
01	000732 12L	0000	007265 120F	0000	007306 121F	0000	007335 1
00	007243 124F	0000	007255 125F	0000	007311 126F	0000	007324 1
00	007363 129F	0001	000140 160G	0001	000160 171G	0001	000173 1
00	007000 200F	0000	007041 201F	0001	000212 207G	0001	000232 2
01	000345 253G	0001	000366 260G	0001	000412 272G	0001	000455 3
01	000603 366G	0001	000705 423G	0001	000770 447G	0001	001015 4
01	000246 5L	0001	001041 51L	0001	001053 52L	0001	001065 5
01	000511 8L	0001	000520 9L	0000	007034 99F	0000 R 005675 A
00 R 006765 C	0000 R 006744 D	0000 R 006777 DEV	0000 R 006756 E
00 R 006767 FL	0000 R 006736 FMT	0000 I 006760 I	0000 I 006763 I
00 I 005733 IVAR	0000 I 006755 IW	0000 I 006752 I$	0000 I 006774 J
00 I 006762 L	0000 I 006772 M	0000 I 006761 N	0000 I 006753 N
00 I 006754 NV	0000 I 006766 NVAR	0003 L 000000 OPTION	0000 R 005752 R
00 R 005726 SIG	0000 R 006776 SSK	0000 R 006775 SS0	0000 R 006770 S
00 R 000000 X	0000 R 005670 XBAR		

1*	C	
2*	C TITLE	LSRA - LEAST SQUARES MULTIPLE REGRESSION PROGRAM
3*	C	WITH ANALYSIS OF VARIANCE PRINTOUT.
4*	C	
5*	C AUTHOR	R.R.SINGERS
6*	C	
7*	C SPONSOR	G.R.ANDERSEN
8*	C	

```
 9*     C DATE          AUGUST,1970
10*     C
1'*     C KEY WORDS      REGRESSION
12*     C                LEAST SQUARES
13*     C
14*     C PURPOSE        TO COMPUTE THE LEAST SQUARES REGRESSION COEFFICIENTS
15*     C                AND THE ANALYSIS OF VARIANCE DATA OF TWO OR MORE
16*     C                DATA SERIES.
17*     C
18*     C METHOD         THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
19*     C                ASKING FOR CERTAIN PARAMETERS AND THE DATA SERIES.
20*     C                THE STANDARD FORWARD LEAST SQUARES REGRESSION
21*     C                TECHNIQUE IS USED.
22*     C
23*     C SUBPROGRAMS    RESTEM - UNIVAC 1108 STAT-PACK FORWARD
24*     C USED           REGRESSION SUBROUTINE.
25*     C
26*     C NOTE           1) A DYNAMICALLY ASSIGNED FILE (UNIT 8) IS
27*     C                USED TO SAVE THE RESIDUALS (WHEN SPECIFIED).
28*     C
29*                      PARAMETER MXL=500,MXV=5
30*                      DIMENSION X(MXL,MXV),W(MXL),XBAR(MXV),A(MXV,MXV),SIG(MXV)
31*           .          ,IVAR(MXV),B(MXV),SB(MXV),R(MXL),FMT(6)
32*                      DIMENSION D(6)
33*                      LOGICAL OPTION
34*                      READ(5,200,END=53)FMT
35*     200              FORMAT(6A6)
36*                      IF(OPTION('N'))GO TO 1
37*                      WRITE(6,100)
38*     100              FORMAT(/' LSRA IS A STEPWISE MULTIPLE REGRESSION PROGRAM'
39*           .          /' WITH ANALYSIS OF VARIANCE PRINTOUT.'
40*           .          /' THE RESIDUALS ARE STORED ON UNIT 8(IF DESIRED).'/)
41*     1                WRITE(6,99)
42*     99               FORMAT(' NUMBER OF OBSERVATIONS?')
43*                      READ(5,201,END=53)NL
44*     201              FORMAT( )
45*                      IF(NL.GT.MXL)GO TO 51
46*                      WRITE(6,101)
47*     101              FORMAT(' NUMBER OF VARIABLES(DEPENDENT AND INDEPENDENT)?'
48*           .          )
49*     2                READ(5,201,END=53)NV
50*                      IF(NV.GT.MXV)GO TO 52
51*                      IW=0
52*     3                WRITE(6,105)
53*     105              FORMAT(' F LEVEL?')
54*                      READ(5,201,END=53)EFIN
55*                      EFOUT=EFIN
56*                      WRITE(6,107)
57*     107              FORMAT(' DEPENDENT VARIABLE FORMAT?')
58*                      READ(5,200,END=53)FMT
59*                      WRITE(6,108)
60*     108              FORMAT(' DEPENDENT VARIABLE DATA:')
61*                      READ(5,FMT)(X(I,NV),I=1,NL)
62*                      N=NV-1
63*                      DO 4 L=1,N
64*                      WRITE(6,109)L
65*     109              FORMAT(' INDEPENDENT VARIABLE',I2,' FORMAT?')
```

```
66*                         READ(5,200,END=53)FMT
67*                         WRITE(6,110)L
68*          110            FORMAT(' INDEPENDENT VARIABLE',I2,' DATA:')
69*          4              READ(5,FMT)(X(I,L),I=1,NL)
70*                         IND=0
71*                         ISTEP=-1
72*          5              CALL RESTEM(X,NL,NV,MXL,MXV,W,IW,EFIN,EFOUT,XBAR,A,SIG,C,
73*          •             NVAR,FL,SY,NOIN,IVAR,B,SB,R,IND)
74*                         IF(IND.EQ.0)GO TO 10
75*                         ISTEP=ISTEP+1
76*                         IF(ISTEP.NE.0)GO TO 7
77*                         WRITE(6,102)(I,XBAR(I),I=1,N)
78*          102            FORMAT(/' MEANS:'/(3(I5,E15.8)))
79*                         WRITE(6,111)
80*          111            FORMAT(//' CORRELATION COEFFICIENTS:')
81*                         M=N-1
82*                         DO 6 I=1,M
83*                         K=I+1
84*          6              WRITE(6,112)(I,J,A(I,J),J=K,N)
85*          112            FORMAT(3(4X,I2,' VS',I2,' = ',F10.6))
86*                         WRITE(6,113)(I,A(I,NV),I=1,N)
87*          113            FORMAT(3(4X,I2,' VS Y = ',F10.6))
88*                         WRITE(6,114)SY
89*          114            FORMAT(' STANDARD ERROR OF Y =',E12.6)
90*                         D(4)=SY**2
91*                         D(3)=NL-1.
92*                         D(2)=D(4)*D(3)
93*                         D(1)=1.E30
94*                         SS0=D(2)
95*                         WRITE(6,103)D
96*                         GO TO 5
97*          7              WRITE(6,115)ISTEP
98*          115            FORMAT(/' STEP NO.',I3)
99*                         IF(NVAR.GE.0)GO TO 8
100*                        NVAR=-NVAR
101*                        FL=-FL
102*                        WRITE(6,116)NVAR
103*         116            FORMAT(5X,' VARIABLE REMOVED:',I3)
104*                        GO TO 9
105*         8              WRITE(6,117)NVAR
106*         117            FORMAT(5X,' VARIABLE ENTERING:',I3)
107*         9              WRITE(6,118)FL,SY,C,(IVAR(I),B(I),SB(I),I=1,NOIN)
108*         118            FORMAT(5X,' FLEVEL',F13.6/6X,'STANDARD ERROR OF Y',F13.6/
109*         •             6X,'CONSTANT',E13.6//13X,'VARIABLE      COEFICIENT     STD.
110*         .ERROR OF COEFF'/(17X,' X',I2,E16.8,E18.8))
111*                        D(4)=SY**2
112*         •              D(3)=D(3)-1.
113*                        SSK=D(4)*D(3)
114*                        D(1)=D(2)-SSK
115*                        D(2)=SSK
116*                        D(5)=SQRT(D(1)/D(4))
117*                        D(6)=1.-D(2)/SS0
118*                        WRITE(6,103)D
119*         103            FORMAT(/' SSVAR=',F13.6,2X,'SSRES=',F13.6,2X
120*         •             ,'DFRES=',F13.6/' MSRES=',F13.6,6X,'T=',
121*         •             F13.6,' MULTCOR=',F12.6/)
122*                        IF(ISTEP.LT.NV)GO TO 5
```

```
123*                         IND=-1
124*                         GO TO 5
125*          10             WRITE(6,124)
126*          124            FORMAT(' DO YOU WANT TO SEE SOME OF THE ',
127*            .            'DEVIATIONS(YES/NO)?')
128*                         READ(5,200,END=53)I
129*                         IF(I.NE.'YES') GO TO 12
130*                         WRITE(6,125)
131*          125            FORMAT(' WHAT PART(INITIAL,FINAL,STEP)?')
132*                         READ(5,201,END=53)J,K,L
133*                         WRITE(6,120)
134*          120            FORMAT(/' PREDICTED VS. ACTUAL RESULTS:'/' OB. NO.',4X,
135*            .            'ACTUAL',4X,'PREDICTED',4X,'%DEVIATION')
136*                         DO 11 I=J,K,L
137*                         DEV=(X(I,NV)-R(I))/X(I,NV)*100.
138*          11             WRITE(6,121)I,X(I,NV),R(I),DEV
139*          121            FORMAT(I5,2E13.6,F10.2)
140*          12             WRITE(6,126)
141*          126            FORMAT(' DO YOU WANT THE RESIDUALS STORED(YES/NO)?')
142*                         READ(5,200,END=53) I
143*                         IF(I.NE.'YES') CALL EXIT
144*                         DO 13 I=1,NL
145*          13             R(I)=X(I,NV)-R(I)
146*                         CALL CSF(2,'@ASG,T 8,F',I)
147*                         WRITE(6,127)
148*          127            FORMAT(' OUTPUT DATA FORMAT?')
149*                         READ(5,200,END=53)FMT
150*                         WRITE(8,FMT)(R(I),I=1,NL)
151*                         WRITE(6,128)
152*          128            FORMAT(' SERIES STORED.')
153*                         GO TO 53
154*          51             I=MXL
155*                         WRITE(6,122)I
156*          122            FORMAT(' NUMBER OF OBSERVATIONS MUST BE LESS THAN',I5,' T
157*            .RY AGAIN')
158*                         GO TO 1
159*          52             I=MXV
160*                         WRITE(6,123)I
161*          123            FORMAT(' NUMBER OF VARIABLES MUST BE LESS THAN',I5,' TRY
162*            .AGAIN')
163*                         GO TO 2
164*          53             WRITE(6,129)
165*          129            FORMAT(' LSR TERMINATES.')
166*                         CALL EXIT
167*                 END
```

END OF COMPILATION: NO DIAGNOSTICS.

R.XCF,J,J

000 COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:29:45.

N PROGRAM

RAGE USED: CODE(1) 000502, DATA(0) 010063, BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

03	OPTION
04	EXIT
05	NINTR$
06	NRDU$
07	NIO1$
10	NIO2$
11	NWDU$
12	NREW$
13	NRBU$
14	SQRT
15	NWBU$
16	NWEF$
17	NSTOP$

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

00	007667 100F	0000	007746 101F	0000	007761 103F	0000	007771 1
00	010011 106F	0000	010014 107F	0001	000011 107G	0000	010025 1
01	000061 133G	0001	000104 143G	0001	000144 161G	0001	000156 1
01	000213 200G	0000	007760 201F	0001	000232 204G	0001	000241 2
01	000332 240G	0001	000356 244G	0001	000421 267G	0001	000422 2
01	000446 303G	0001	000471 53L	0001	000432 8L	0001	000026 9
00 R	007650 B	0000 R	000000 C	0000 R	007640 FMT	0000 I	007654 1
00 I	007647 ICO	0000 I	007662 II	0000 I	007646 IS	0000 I	007651 J
00 I	007653 M	0000 I	007661 MR	0000 I	007655 M1	0000 I	007652 N
03 L	000000 OPTION	0000 R	003720 Y				

1*	C	
2*	C TITLE	XCF - CROSS-COVARIANCE/CORRELATION PROGRAM.
3*	C	
4*	C AUTHOR	R.R.SINGERS
5*	C	
6*	C SPONSOR	G.R.ANDERSEN
7*	C	
8*	C DATE	AUGUST,1970
9*	C	
10*	C KEY WORDS	CROSS-COVARIANCE
11*	C	CROSS-CORRELATION
12*	C	
13*	C PURPOSE	TO COMPUTE THE CROSS-COVARIANCES OF TWO DATA SERIES
14*	C	AND IF DESIRED, THE CROSS-CORRELATIONS OF TWO DATA
15*	C	SERIES.

```
16*     C
17*     C METHOD        THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
18*     C               ASKING FOR CERTAIN PARAMETERS AND TWO DATA SERIES.
19*     C               IT IS ASSUMED THAT ACF WAS EXECUTED AND THE AUTO-
20*     C               COVARIANCES SAVED. THE STANDARD METHOD OF COMPUTATION
21*     C               IS USED.
22*     C
23*     C NOTES         1) IT IS ASSUMED THAT THE AUTO-COVARIANCES
24*     C               ARE STORED (ON UNIT 1) FROM ACF.
25*     C               2) THE CROSS-COVARIANCES ARE STORED AFTER
26*     C               THE AUTO-COVARIANCES (ON UNIT 1).
27*     C
28*                     PARAMETER MX=1000
29*                     DIMENSION C(MX,2),Y(MX,2),FMT(6)
30*                     LOGICAL OPTION
31*                     READ(5,200,END=53)FMT
32*     200             FORMAT(6A6)
33*                     IF(OPTION('N'))GO TO 99
34*                     WRITE(6,100)
35*     100             FORMAT(/' XCF COMPUTES THE CROSS-COVARIANCE/-CORRELATION',
36*       .             ' FUNCTION OF TWO SERIES.'/' IT IS ASSUMED THAT THE ',
37*       .             'AUTO-COVARIANCES OF THE TWO SERIES ARE'/' STORED ON ',
38*       .             'UNIT 1. THE CROSS-COVARIANCES ARE STORED ON '/' UNIT',
39*       .             ' 1(IF DESIRED) AFTER THE AUTO-COVARIANCES.')
40*     99              REWIND 1
41*                     WRITE(6,101)
42*     101             FORMAT(' DO YOU WANT COVARIANCE(COV) OR CORRELATION(COR)?')
43*                     READ(5,200,END=53)ICO
44*     201             FORMAT( )
45*                     B=1.
46*                     DO 3 J=1,2
47*                     WRITE(6,103)J
48*     103             FORMAT(' INPUT DATA FORMAT OF SERIES',I2,'?')
49*                     READ(5,200,END=53)FMT
50*                     WRITE(6,104)J
51*     104             FORMAT(' INPUT DATA OF SERIES',I2,':')
52*                     READ(1)N,M
53*                     READ(5,FMT,END=53)(Y(I,J),I=1,N)
54*                     READ(1)(C(I,J),I=1,M)
55*     3               B=B*C(1,J)
56*                     B=SQRT(B)
57*                     M1=M-1
58*                     DO 5 I=0,M1
59*                     NI=N-I
60*                     DO 5 J=1,2
61*                     A=0.
62*                     DO 4 K=1,NI
63*     4               A=A+Y(K,J)*Y(K+I,3-J)
64*                     C(I+1,J)=A/N
65*     5               IF(ICO.EQ.'COR')C(I+1,J)=C(I+1,J)/B
66*                     MR=M/4
67*                     II=MR+1
68*                     IF(MR.GT.10)MR=10
69*                     DO 6 K=1,2
70*                     I=3-K
71*                     WRITE(6,105)ICO,K,I
72*     105             FORMAT(/' CROSS-',A3,'. FOR SERIES',I3,' VS.',I3,'.')
```

```
73*                    DO 6 I=0,MR
74*        6           WRITE(6,106)(J,C(J+1,K),J=I,M1,II)
75*        106         FORMAT(4(I4,E15.8))
76*                    WRITE(6,107)
77*        107         FORMAT(' DO YOU WANT THE COVARIANCES STORED(YES/NO)?')
78*                    READ(5,200,END=53)IC
79*                    IF(IC.NE.'YES')GO TO 53
80*                    IF(ICO.NE.'COR')GO TO 8
81*                    DO 7 J=1,2
82*                    DO 7 I=1,M
83*        7           C(I,J)=B*C(I,J)
84*        8           WRITE(1)((C(I,J),I=1,M),J=1,2)
85*                    WRITE(6,108)
86*        108         FORMAT(' COVARIANCES STORED.')
87*                    END FILE 1
88*                    REWIND 1
89*        53          WRITE(6,111)
90*        111         FORMAT(' XCF TERMINATES.')
91*                    CALL EXIT
92*         END
```

END OF COMPILATION: NO   DIAGNOSTICS.

R.XSF,J,J

00 COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:30:29.

N PROGRAM

RAGE USED: CODE(1) 001174; DATA(0) 014152; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

03	OPTION
04	ATANH
05	EXIT
06	NINTR$
07	NRDU$
10	NIO1$
11	NIO2$
12	NWDU$
13	NRBU$
14	SQRT
15	COS
16	SIN
17	ATAN2
20	NWBU$
21	NSTOP$

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

00	013713 100F	0000	013774 105F	0000	014010 106F	0000	014024 1
00	014060 109F	0000	014101 110F	0000	014030 111F	0000	014045 1
01	000011 113G	0001	000050 134G	0001	000077 146G	0001	000110 1
01	000210 175G	0000	013710 200F	0000	013773 201F	0001	000223 2
01	000413 244G	0001	000436 257G	0001	000445 264G	0001	000452 2
01	000657 334G	0001	000723 351G	0001	000750 367G	0001	001100 4
01	001146 442G	0001	001163 53L	0001	000026 99L	0000 R	013660 A
04 R	000000 ATANH	0000 R	000000 C	0000 R	013664 CO	0000 R	013702 D
00 R	013703 EF	0000 R	013673 EL1	0000 R	013656 EM1	0000 R	013657 E
00 R	003720 F	0000 R	013560 FMT	0000 R	013701 FP	0000 I	013651 I
00 I	013706 IP	0000 I	013562 IPT	0000 I	013644 IS	0000 I	013652 J
00 I	013672 L	0000 I	013653 M	0000 I	013647 N	0000 I	013650 N
00 R	013676 OD	0003 L	000000 OPTION	0000 R	013705 PH	0000 R	013646 P
00 R	007640 Q	0000 R	013665 SI	0000 R	013663 TAU	0000 R	013707 T
00 R	013667 V1	0000 R	013677 V2	0000 R	013674 W	0000 R	013670 Z
00 R	013700 Z2						

1*	C		
2*	C	TITLE	XSF - CROSS-SPECTRUM FUNCTION.
3*	C		
4*	C	AUTHOR	R.R.SINGERS
5*	C		
6*	C	SPONSOR	G.R.ANDERSEN
7*	C		

```
 8*    C DATE          SEPTEMBER,1970
 9*    C
10*    C KEY WORDS      CO-SPECTRUM
11*    C                QUADRATURE
12*    C
13*    C PURPOSE        TO COMPUTE THE CO-SPECTRUM,QUADRATURE,PHASE,AND
14*    C                COHERENCE BETWEEN TWO DATA SERIES.
15*    C
16*    C METHOD         THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
17*    C                ASKING FOR CERTAIN PARAMETERS AND TWO DATA SERIES.
18*    C                IT IS ASSUMED THAT ACF,XCF,AND ASF HAVE BEEN
19*    C                EXECUTED AND THE AUTO-COVARIANCES,CROSS-COVARIANCES,
20*    C                AND THE AUTO-SPECTRA(USING THE TUKEY WINDOW)
21*    C                HAVE BEEN SAVED. THE STANDARD METHOD OF COMPUTATION
22*    C                IS USED.
23*    C
24*    C NOTES          1) THE AUTO AND CROSS COVARIANCES ARE ASSUMED TO BE
25*    C                STORED(ON UNIT 1) FROM ACF AND XCF.
26*    C                2) THE AUTO SPECTRA ARE ASSUMED TO BE STORED(ON
27*    C                UNIT 2) FROM ASF.
28*    C
29*                     PARAMETER MX=1000
30*                     DIMENSION C(MX,2),F(MX,2),Q(MX,2),FMT(2),IPT(50)/50*
31*         .           ' '/
32*         .           DATA PIV,PI/.1591549,3.1415926/
33*                     LOGICAL OPTION
34*                     READ(5,200,END=53)FMT
35*    200              FORMAT(2A6)
36*                     IF(OPTION('N'))GO TO 99
37*                     WRITE(6,100)
38*    100              FORMAT(/' XSF COMPUTES THE CROSS-SPECTRA FUNCTIONS FOR',
39*         .           /' THE TUKEY WINDOW.'/' IT IS ASSUMED THAT THE AUTO- AND C
40*         .           .ROSS-COVARIANCES ARE STORED ON'/' UNIT 1 AND THAT THE AUTO-SPECTRA
41*         .           .ARE STORED ON UNIT 2.'/' THE CROSS SPECTRA ARE STORED ON UNIT 2 AF
42*         .           .TER THE AUTO-SPECTRA(IF DESIRED).')
43*    201              FORMAT( )
44*     99              READ(1)N,NN
45*                     READ(1)(C(I,1),I=1,NN)
46*                     READ(1)N,NN
47*                     READ(1)(C(I,1),I=1,NN)
48*                     READ(1)((C(I,J),I=1,NN),J=1,2)
49*                     READ(2)M,DEL,NQ
50*                     EM1=M-1
51*                     ENQ1=NQ-1.
52*                     AMU=2.667*N/EM1
53*                     ANU=1.65/SQRT(AMU)
54*                     DO 2 I=1,2
55*                     J=MOD(I,2)+1
56*                     DO 2 K=1,NQ
57*                     TAU=K-1
58*                     CO=COS(PI*TAU/ENQ1)
59*                     SI=SIN(PI*TAU/ENQ1)
60*                     V0=0.
61*                     V1=0.
62*                     Z0=0.
63*                     Z1=0.
64*                     DO 1 L=M,2,-1
```

```
65*                      EL1=L-1
66*                      W=.5*(1.+COS(PI*EL1/EM1))
67*                      EV=.5*(C(L,I)+C(L,J))
68*                      OD=.5*(C(L,I)-C(L,J))
69*                      V2=2.*CO*V1-V0+W*EV
70*                      Z2=2.*CO*Z1-Z0+W*OD
71*                      V0=V1
72*                      V1=V2
73*                      Z0=Z1
74*          1           Z1=Z2
75*                      F(K,I)=DEL*(C(1,I)+2.*(V1*CO-V0))
76*          2           Q(K,I)=2.*PIV*Z1*SI
77*                      WRITE(6,105)
78*          105         FORMAT(' DO YOU WANT FREQUENCY(FREQ.) OR PERIOD(PERIOD)',
79*            .         /' DISPLAYED?')
80*                      READ(5,200,END=53)FP
81*                      READ(2)(C(I,1),I=1,NQ)
82*                      READ(2)I,D,J
83*                      READ(2)(C(I,2),I=1,NQ)
84*                      DO 3 I=1,NQ
85*                      D=C(I,1)*C(I,2)
86*                      DO 3 J=1,2
87*                      C(I,J)=ATANH(SQRT((F(I,J)**2+Q(I,J)**2)/D))
88*          3           FMT(J)=MAX(ABS(C(I,J)),FMT(J))
89*                      FMT(1)=FMT(1)/49.
90*                      FMT(2)=FMT(2)/49.
91*                      I=1
92*                      J=2
93*                      WRITE(6,106)FP,I,J
94*          106         FORMAT(/2X,A6,'   ARCTANH(COHERENCE) FOR SERIES',I2,' VS.'
95*            .         ,I2,' .')
96*                      DO 4 K=1,NQ
97*                      L=K-1
98*                      IF(FP.NE.'PERIOD')EF=.5*L/ENQ1/DEL
99*                      IF(FP.EQ.'PERIOD'.AND.L.NE.0)EF=2.*ENQ1/L*DEL
00*                      IF(FP.EQ.'PERIOD'.AND.L.EQ.0)EF=99.9999
01*                      IF=IFIX(C(K,I)/FMT(I))+1
02*                      IF(IF.GT.50)IF=50
03*                      IF(IF.LT.1)IF=1
04*                      IPT(IF)='+'
05*                      WRITE(6,107)EF,C(K,I),IPT
06*          107         FORMAT(F8.4,E12.5,2X,50A1)
07*          4           IPT(IF)=' '
08*                      IF=IFIX(ANU/FMT(I))+1
09*                      IPT(1)='*'
10*                      IPT(IF)='*'
11*                      WRITE(6,111)AMU,IPT
12*          111         FORMAT(' NU=',F5.0,13X,50A1)
13*                      IPT(1)=' '
14*                      IPT(IF)=' '
15*                      WRITE(6,108)FP,I,J
16*          108         FORMAT(/3X,A6,'   PHASE FOR SERIES',I2,' VS.',I2,' .')
17*                      DO 5 K=1,NQ
18*                      PH=57.2958*ATAN2(Q(K,I),F(K,I))
19*                      L=K-1
20*                      IF(FP.NE.'PERIOD')EF=.5*L/ENQ1/DEL
21*                      IF(FP.EQ.'PERIOD'.AND.L.EQ.0)EF=99.9999
```

```
122*                      IF(FP.EQ.'PERIOD'.AND.L.NE.0)EF=2.*ENQ1/L*DEL
123*                      IP=IFIX(PH/7.16)+25
124*                      IF(IP.GT.50)IP=50
125*                      IF(IP.LT.1)IP=1
126*                       IPT(25)='*'
127*                      IPT(IP)='+'
128*                      WRITE(6,107)EF,PH,IPT
129*                      IPT(25)=' '
130*        5             IPT(IP)=' '
131*                      WRITE(6,112)
132*        112           FORMAT(T21,'*',25X,'*',25X,'*'/T19,'-180',
133*          .           24X,'0',24X,'180')
134*                      WRITE(6,109)
135*        109           FORMAT(/' DO YOU WANT THE CO-SPECTRA AND QUADRATURES',/
136*          .           ' STORED FOR TIME SERIES REGRESSION(YES/NO)?')
137*                      READ(5,200,END=53)TS
138*                      IF(TS.NE.'YES')GO TO 53
139*                      WRITE(2)((F(I,J),I=1,NQ),J=1,2)
140*                      WRITE(6,110)
141*        110           FORMAT(' SERIES STORED.')
142*        53            WRITE(6,113)
143*        113           FORMAT(' XSF TERMINATES.')
144*                      CALL EXIT
145*              END
```

END OF COMPILATION: NO DIAGNOSTICS.

R.COMB.J.J
JUO COMPILED BY 1201 BCS7E ON 01 DEC 70 AT 12:31:15.

N PROGRAM

RAGE USED: CODE(1) 000522; DATA(0) 004161; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

J3	OPTION
J4	CSF
J5	EXIT
J6	NINTRS
07	NRDUS
10	NIOIS
11	N1O2S
12	NWDUS
13	NSTOPS

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

J1	000026 1L	0001	000240 IOL	0000	003743 100F	0000	003776 I
JU	004015 103F	0000	004022 104F	0000	004030 105F	0020	004035 1
JU	004055 108F	0000	004066 109F	0000	004073 110F		000011 1
JU	004110 112F	0000	004122 113F	0000	004136 114F		000255 1
J1	000301 14L	0001	000102 145G	0001	000364 15L		000124 1
JU	004006 201F	0001	000201 205G	0001	000216 215G	0001	000233 2
J1	000272 247G	0001	000360 276G	0001	000132 3L	0001	000424 3
J1	000154 4L	0001	C00455 50L	0001	000465 51L	0001	000477 5
J1	000206 6L	0001	000223 8L	0000 R	003726 ASG8	0000 R	003720 F
JU I	003735 IOP	0000 I	003730 IS	0000 I	003736 J	0000 I	003733 K
JU I	003732 M	0000 I	003734 N	0003 L	000000 OPTION	0000 R	000000 X

1*	C		
2*	C TITLE	COMB - A PROGRAM TO COMBINE TWO DATA SERIES.	
3*	C		
4*	C AUTHOR	R.R.SINGERS	.
5*	C		
6*	C SPONSOR	G.R.ANDERSEN	
7*	C		
8*	C DATE	NOVEMBER,1970	
9*	C		
10*	C KEY WORDS	COMBINE,	
11*	C	CONCATINATE	
12*	C		
13*	C PURPOSE	TO COMBINE TWO DATA SERIES USING ONE OF FIVE	
14*	C	OPERATORS:ADD,SUB,MPY,DIV,CON.	
15*	C		
16*	C METHOD	THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE	
17*	C	ASKING FOR CERTAIN PARAMETERS AND APPLYING THEM TO	
18*	C	TWO INPUT DATA SERIES. THE RESULTANT SERIES MAY THEN	

```
19*    C                    BE STORED.
20*    C
21*    C NOTE              1) A DYNAMICALLY ASIGNED FILE (UNIT 8) IS USED TO
22*    C                    STORE THE FINAL SERIES.
23*    C
24*                         PARAMETER MX=1000
25*                         DIMENSION X(MX,2),FMT(6),
26*        .          ASG8(2)/'@ASG,T 8,F'/
27*                         LOGICAL OPTION
28*                         READ(5,200,END=53)FMT
29*    200                  FORMAT(6A6)
30*                         IF(OPTION('N'))GO TO 1
31*                         WRITE(6,100)
32*    100                  FORMAT(' COMB COMBINES TWO SERIES BY ONE OF THE FOLLOWING
33*        . SERIES:'/' ADD,SUB,MPY,DIV,CON(CONCATINATE)'/' THE RESULTANT SERI
34*        .ES IS STORED ON UNIT 8(IF DESIRED).')
35*    1                    DO 2 I=1,2
36*                         WRITE(6,101)I
37*    101                  FORMAT(' LENGTH OF SERIES',I2,' TO BE READ IN?')
38*                         READ(5,201,END=53)M
39*    201                  FORMAT( )
40*                         IF(M.GT.MX)GO TO 51
41*                         WRITE(6,102)I
42*    102                  FORMAT(' INPUT DATA',I2,' FORMAT?')
43*                         READ(5,200,END=53)FMT
44*                         WRITE(6,103)I
45*    103                  FORMAT(' INPUT DATA',I2,' :')
46*    2                    READ(5,FMT)(X(K,I),K=1,M)
47*    3                    WRITE(6,104)
48*    104                  FORMAT(' LENGTH OF OUTPUT SERIES?')
49*                         READ(5,201,END=53)N
50*                         IF(N.GT.2*MX)GO TO 52
51*    4                    WRITE(6,105)
52*    105                  FORMAT(' OPERATOR TO BE USED?')
53*                         READ(5,200,END=53)IOP
54*                         IF(IOP.NE.'ADD')GO TO 6
55*                         DO 5 J=1,N
56*    5                    X(J,1)=X(J,1)+X(J,2)
57*                         GO TO 14
58*    6                    IF(IOP.NE.'SUB')GO TO 8
59*                         DO 7 J=1,N
60*    7                    X(J,1)=X(J,1)-X(J,2)
61*                         GO TO 14
62*    8                    IF(IOP.NE.'MPY')GO TO 10
63*                         DO 9 J=1,N
64*    9                    X(J,1)=X(J,1)*X(J,2)
65*                         GO TO 14
66*    10                   IF(IOP.NE.'DIV')GO TO 12
67*                         DO 11 J=1,N
68*    11                   X(J,1)=X(J,1)/X(J,2)
69*                         GO TO 14
70*    12                   IF(IOP.NE.'CON')GO TO 50
71*                         K=N-M+1
72*                         L=0
73*                         DO 13 J=K,N
74*                         L=L+1
75*    13                   X(J,1)=X(L,2)
```

```
76*     14           WRITE(6,106)
77*     106          FORMAT(' DO YOU WANT TO SEE PART OF THE SERIES(YES/NO)?')
78*                  READ(5,200,END=53)IOP
79*                  IF(IOP.NE.'YES')GO TO 15
80*                  WRITE(6,107)
81*     107          FORMAT(' WHAT PART (INITIAL,FINAL,STEP)?')
82*                  READ(5,201,END=51)I,J,K
83*                  WRITE(6,201)(X(L,1),L=I,J,K)
84*     15           WRITE(6,108)
85*     108          FORMAT(' DO YOU WANT THE RESULTANT SERIES STORED(YES/NO)'
86*            .     )
87*                  READ(5,200,END=53)IOP
88*                  IF(IOP.NE.'YES')GO TO 53
89*                  WRITE(6,109)
90*     109          FORMAT(' OUTPUT DATA FORMAT?')
91*                  READ(5,200,END=53)FMT
92*                  CALL CSF(2,ASG8,I)
93*                  WRITE(8,FMT)(X(I,1),I=1,N)
94*                  WRITE(6,110)
95*     110          FORMAT(' SERIES STORED.')
96*                  GO TO 53
97*     50           WRITE(6,111)IOP
98*     111          FORMAT(1X,A6,' IS NOT A RECOGNIZED OPERATOR,TRY AGAIN.')
99*                  GO TO 4
100*    51           L=MX
101*                 WRITE(6,112)L
102*    112          FORMAT(' LENGTH OF SERIES MUST BE LESS THAN',I2,
103*           .     ' TRY AGAIN.')
104*                 GO TO 1
105*    52           L=2*MX
106*                 WRITE(6,113)L
107*    113          FORMAT(' LENGTH OF OUTPUT SERIES MUST BE LESS THAN',I5,
108*           .     ' TRY AGAIN.')
109*                 GO TO 3
110*    53           WRITE(6,114)
111*    114          FORMAT(' COMB TERMINATES.')
112*                 CALL EXIT
113*           END
```

END OF COMPILATION: NO DIAGNOSTICS.

R.LSRP,J,J
000 COMPILED BY 1201 BCS7E ON 03 DEC 70 AT 08:37:28.

N PROGRAM

RAGE USED: CODE(1) 001012; DATA(0) 007326; BLANK COMMON(2) 000000

ERNAL REFERENCES (BLOCK, NAME)

03	OPTION
04	REBSOM
05	CSF
06	EXIT
07	NINTR$
10	NRDU$
11	NIO1$
12	N1O2$
13	NWDU$
14	SQRT
15	NSTOP$

RAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

01	000026 1L	0000	006764 100F	0000	007014 101F	0000	007022 1
00	007036 104F	0000	007044 105F	0000	007052 106F	0000	007062 1
00	007101 109F	0000	007116 110F	0001	000011 110G	0000	007120 1
00	007141 113F	0000	007156 114F	0000	007160 115F	0000	007174 1
00	007225 118F	0000	007240 119F	0000	007253 120F	0000	007257 1
01	000150 157G	0001	000171 170G	0001	000205 176G	0001	000055 2
00	007021 201F	0001	000224 206G	0001	000245 217G	0001	000260 2
01	000455 301G	0001	000474 310G	0001	000515 322G	0001	000530 3
01	000551 343G	0001	000553 347G	0001	000574 356G	0001	000703 4
01	000723 427G	0001	000741 51L	0001	000753 52L	0001	000765 5
01	001002 55L	0000 R 006654 ATA	0000 R 006731 EFOUT	0000 R 006717 F
00 I 006736 INDIC	0000 I 006712 IVAR	0000 I 006725 I$	0000 I 006745 I
00 I 006752 JK	0000 I 006750 JL	0000 I 006760 K	0000 I 006751 K
00 I 006754 LM	0000 I 006733 N	0000 I 006726 NL	0000 I 006746 N
00 I 006730 NV	0000 I 006727 NV1	0003 L 000000 OPTION	0000 R 005670 R
00 R 006737 SY	0000 R 006744 TL	0000 R 006741 TNU	0000 R 006743 1
00 R 000000 X	0000 R 006753 Y	0000 R 006757 YL	00G0 R 006755 Y

1*	C
2*	C TITLE	LSRP - LEAST SQUARES MULTIPLE REGRESSION PROGRAM
3*	C	WITH PREDICTIONS.
4*	C
5*	C AUTHOR	R.R.SINGERS
6*	C
7*	C SPONSOR	G.R.ANDERSEN
8*	C
9*	C DATE	DECEMBER,1970
10*	C

```
11*     C KEY WORDS      REGRESSION
12*     C               LEAST SQUARES
13*     C               PREDICTIONS
14*     C
15*     C PURPOSE        TO COMPUTE THE LEAST SQUARES REGRESSION COEFFICIENTS
16*     C               AND PREDICTION VALUES WITH UPPER AND LOWER
17*     C               PREDICTION LIMITS.
18*     C
19*     C METHOD         THE PROGRAM IS DESIGNED IN AN INTERACTIVE MODE
20*     C               ASKING FOR CERTAIN PARAMETERS AND DATA SERIES.
21*     C               THE STANDARD BACKWARD REGRESSION TECHNIQUE IS
22*     C               USED.
23*     C
24*     C SUBROUTINES    REBSOM - UNIVAC 1108 STAT-PACK BACKWARD
25*     C USED           REGRESSION SUBROUTINE.
26*     C
27*     C NOTE           1) A DYNAMICALLY ASSIGNED FILE (UNIT 8) IS
28*     C               USED TO SAVE THE PREDICTED VALUES AND THE
29*     C               UPPER AND LOWER PREDICTION LIMITS.
30*     C
31*                     PARAMETER MXN=500,MXNP=5
32*                     PARAMETER MXNP1=MXNP+1
33*                     DIMENSION X(MXN,MXNP1),R(MXN),ATA(MXNP,MXNP1),
34*          .          IVAR(MXNP),FMT(6)
35*                     LOGICAL OPTION
36*                     READ(5,200,END=53)FMT
37*     200             FORMAT(6A6)
38*                     IF(OPTION('N'))GO TO 1
39*                     WRITE(6,100)
40*     100             FORMAT(/' LSRP IS A STEPWISE MULTIPLE REGRESSION PROGRAM'
41*          .          ,' WITH PREDICTIONS.'/' THE PREDICTION RESULTS ARE STORED
42*          .'          ,'(IF DESIRED) ON UNIT 8.')
43*     1               WRITE(6,101)
44*     101             FORMAT(' NUMBER OF OBSERVATIONS?')
45*                     READ(5,201,END=53)NL
46*     201             FORMAT( )
47*                     IF(NL.GT.MXN)GO TO 51
48*                     WRITE(6,102)
49*     102             FORMAT(' NUMBER OF VARIABLES(DEPENDENT AND INDEPENDENT)?'
50*          .          )
51*     2               READ(5,201,END=53)NV1
52*                     NV=NV1+1
53*                     IF(NV1.GT.MXNP)GO TO 52
54*                     WRITE(6,103)
55*     103             FORMAT(' F LEVEL?')
56*                     READ(5,201,END=53)EFOUT
57*                     WRITE(6,104)
58*     104             FORMAT(' DEPENDENT VARIABLE FORMAT?')
59*                     READ(5,200,END=53)FMT
60*                     WRITE(6,105)
61*     105             FORMAT(' DEPENDENT VARIABLE DATA:')
62*                     READ(5,FMT,END=53)(X(I,NV),I=1,NL)
63*                     N=NV1-1
64*                     DO 3 L=1,N
65*                     WRITE(6,106)L
66*     106             FORMAT(' INDEPENDENT VARIABLE',I2,' FORMAT?')
67*                     READ(5,200,END=53),MT
```

```
68*                         WRITE(6,112)L
69*      3                  READ(5,FMT,END=53)(X(I,L+1),I=1,NL)
70*                         DO 9 I=1,NL
71*      9                  X(I,1)=1.
72*                         CALL REBSOM(X,NL,NV1,MXN,MXNP,EFOUT,VARY,R,ATA,INDIC,IVAR
73*      .                  ,$54,$55)
74*                         SY=SQRT(VARY)
75*                         NU=NL-INDIC
76*                         WRITE(6,107)NU
77*      107                FORMAT(' T-VALUE FOR NU=',I3,' DEGREES OF FREEDOM?')
78*                         READ(5,201,END=53)TNU
79*                         WRITE(6,108)SY
80*      108                FORMAT(' STANDARD ERROR OF Y',F13.7)
81*                         WRITE(6,109)
82*      109                FORMAT(' VARIABLE COEFFICIENT        S.D.    UPPER T    LOWE
83*      .R T'/' (0=CONST.)')
84*                         DO 4 I=1,INDIC
85*                         SDC=SY*SQRT(ATA(I,I))
86*                         TU=ATA(I,NV)+TNU*SDC
87*                         TL=ATA(I,NV)-TNU*SDC
88*                         I1=I-1
89*      4                  WRITE(6,110)I1,ATA(I,NV),SDC,TU,TL
90*      110                FORMAT(I10,4E11.5)
91*                         WRITE(6,111)
92*      111                FORMAT(' PREDICTION DATA:'/' NUMBER OF PREDICTION VALUES?
93*      .')
94*                         READ(5,201,END=53)NPR
95*                         DO 5 I=1,N
96*                         WRITE(6,106)I
97*                         READ(5,200,END=53)FMT
98*                         WRITE(6,112)I
99*      112                FORMAT(' INDEPENDENT VARIABLE',I2,' DATA:')
100*     5                  READ(5,FMT,END=53)(X(J,I+1),J=1,NPR)
101*                        DO 10 J=1,NPR
102*     10                 X(J,1)=1.
103*                        WRITE(6,113)
104*     113                FORMAT (' PREDICTED VALUES:'/' NUMBER   PREDICTION        UPP
105*     .ER T    LOWER T')
106*                        DO 8 JL=1,NPR
107*                        DO 6 KL=1,NV
108*                        R(KL)=0.
109*                        DO 6 JK=1,INDIC
110*     6                  R(KL)=R(KL)+X(JL,JK)*ATA(JK,KL)
111*                        Y=0.
112*                        DO 7 LM=1,INDIC
113*     7                  Y=Y+R(LM)*X(JL,LM)
114*                        Y=SY*SQRT(1.+Y)
115*                        YPR=R(NV)
116*                        YU=YPR+TNU*Y
117*                        YL=YPR-TNU*Y
118*                        X(JL,1)=YPR
119*                        X(JL,2)=YL
120*                        X(JL,3)=YU
121*     8                  WRITE(6,114)JL,YPR,YU,YL
122*     114                FORMAT(I7,3E13.7)
123*                        WRITE(6,115)
124*     115                FORMAT(/' DO YOU WANT THE PREDICTION VALUES AND LIMITS SA
```

```
125*             .VED(YES/NO)?')
126*                          READ(5,200,END=53)I
127*                          IF(I.NE.'YES')GO TO 53
128*                          WRITE(6,116)
129*     116              FORMAT(' OUTPUT FORMAT?')
130*                          READ(5,200,END=53)FMT
131*                          CALL CSF(2,'@ASG,T 8,F',J)
132*                          WRITE(8,FMT)((X(J,K),J=1,NPR),K=1,3)
133*                          WRITE(6,117)
134*     117              FORMAT(/' SERIES STORED IN FOLLOWING ORDER:'/
135*            .          ' PREDICTION SERIES,LOWER LIMIT SERIES,UPPER LIMIT SERIES
136*            ..')
137*                          GO TO 53
138*     51              I=MXN
139*                          WRITE(6,118)I
140*     118              FORMAT(' NUMBER OF OBSERVATIONS MUST BE LESS THAN',I5,
141*            .          ' TRY AGAIN.')
142*                          GO TO 1
143*     52              I=MXNP
144*                          WRITE(6,119)I
145*     119              FORMAT(' NUMBER OF VARIABLES MUST BE LESS THAN',I5,
146*            .          ' TRY AGAIN.')
147*                          GO TO 2
148*     53              WRITE(6,120)
149*     120              FORMAT(' LSRP TERMINATES.')
150*                          CALL EXIT
151*     54              WRITE(6,121)
152*     121              FORMAT(/' *** ATA MATRIX OF REBSOM SINGULAR ***')
153*                          GO TO 53
154*     55              WRITE(6,122)
155*     122              FORMAT(/' *** NO VARIABLES LEFT IN REGRESSION ***')
156*                          GO TO 53
157*              END
```

END OF COMPILATION: NO DIAGNOSTICS.

REFERENCES

1. Alvia, A. S. and Jenkins, G. M., An Example of Digital Filtering, Applied Statistics 14, 70 (1965).

2. Amos, D. E. and Koopmans, L. H., Tables of the Distribution of the Coefficient of Coherence for Stationary Bivariate Gaussian Processes, Sandia Corporation Monograph, March 1963.

3. Blackman, R. B. and Tukey, J. W., The Measurement of Power Spectra, Dover Publications, Inc., New York (1958).

4. Dixon, W. J., (Editor), BMD, Biomedical Computer Programs, University of California Publications in Automatic Computation, 1967.

5. Durbin, J., The Fitting of Time-Series Models, Revue Inst. Int. De Stat. 28, 3, 1960.

6. Gentleman, W. M. and Sande, G., Fast Fourier Transforms--For Fun and Profit, Proceedings of the Fall Joint Computer Conference, 1966.

7. Goodman, N. R., Measuring Amplitude and Phase, Journal of the Franklin Inst., 270, 6 (1960).

8. Grenander, U. and Rosenblatt, M., Statistical Analysis of Stationary Time Series, John Wiley & Sons, New York, 1957.

9. Hannan, E. J., Multiple Time Series Analysis, John Wiley & Sons (to appear).

10. Hannan, E. J., Regression for Time Series, Proceedings of the Symposium on Time Series Analysis, June 11-14, 1962, edited by Murray Rosenblatt.

11. Harris, B., (Editor), Advanced Seminar on Spectral Analysis of Time Series, John Wiley & Sons, 1967.

12. Jenkins, G. M. and Watts, D. G., Spectral Analysis and its Applications, Holden Day, San Francisco, 1968.

13. Univac 1108 - FORTRAN V, Programmer's reference, UP-4060 Rev. 1.

14. Univac 1108 OPERATING SYSTEM - EXEC 8, Programmer's reference, UP-4144 Rev. 1.