A Technical Memorandum ERL WPL-22

DEPARTMENT OF COMMERCE
UNITED STATES OF AMERICA

COMPUTER PROGRAMS FOR MAXIMUM ENTROPY SPECTRAL ANALYSIS OF REAL AND COMPLEX MULTICHANNEL TIME SERIES (WITH MICROFILM PLOTS)

Otto Neall Strand

Wave Propagation Laboratory
Boulder, Colorado
April 1977

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Environmental Research Laboratories

NOAA Technical Memorandum ERL WPL-22

COMPUTER PROGRAMS FOR MAXIMUM ENTROPY SPECTRAL ANALYSIS OF REAL AND COMPLEX MULTICHANNEL TIME SERIES (WITH MICROFILM PLOTS)

Otto Neall Strand

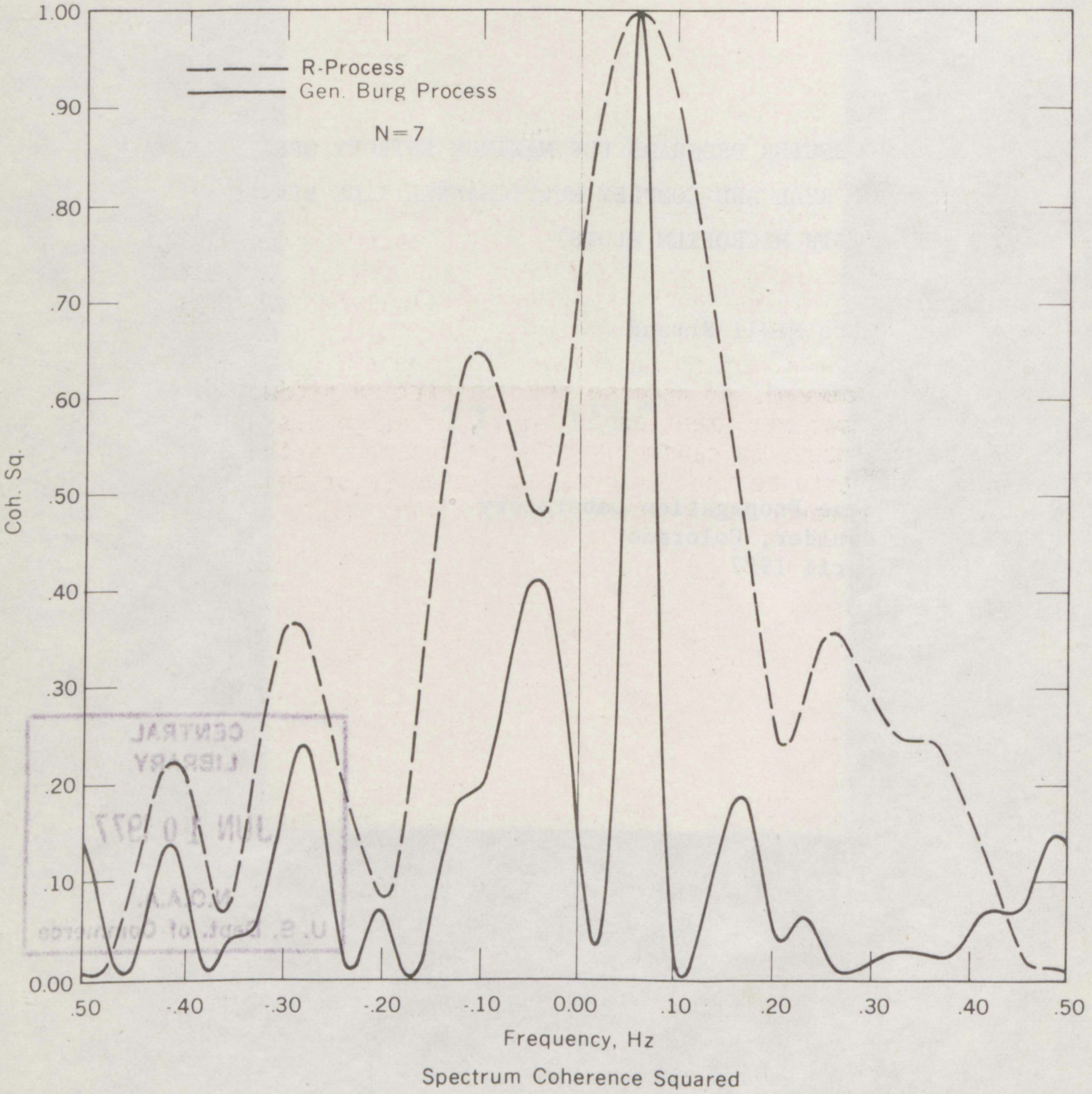
Wave Propagation Laboratory
Boulder, Colorado

April 1977

UNITED STATES
DEPARTMENT OF COMMERCE
Juanita M. Kreps, Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Robert M. White, Administrator

Environmental Research
Laboratories
Wilmot N. Hess, Director



Frontispiece. Comparison between coherence squared obtained from the R-(Yule-Walker) process (dashed) and the multichannel Burg process for the nearly monochromatic signal described in section 2.

NOTICE

The Environmental Research Laboratories do not approve, recommend, or endorse any proprietary product or proprietary material mentioned in this publication. No reference shall be made to the Environmental Research Laboratories or to this publication furnished by the Environmental Research Laboratories in any advertising or sales promotion which would indicate or imply that the Environmental Research Laboratories approve, recommend, or endorse any proprietary product or proprietary material mentioned herein, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of this Environmental Research Laboratories publication.

NOTICE

The Environmental Research Laboratories do not approve, recommend, or endorse any proprietary product or proprietary material mentioned in this publication. No reference shall be made to the Environmental Research Laboratories or to this publication furnished by the Environmental Research Laboratories in any advertising or sales promotion which would indicate or imply that the Environmental Research Laboratories approve, recommend, or endorse any proprietary product or proprietary material mentioned herein, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of this Environmental Research Laboratories publication.

CONTENTS

COMPUTER PROGRAMS FOR MAXIMUM ENTROPY SPECTRAL ANALYSIS OF REAL AND COMPLEX MULTICHANNEL TIME SERIES (WITH MICROFILM PLOTS)

Otto Neall Strand
NOAA/ERL/Wave Propagation Laboratory
Boulder, Colorado 80302

ABSTRACT

This report describes various FORTRAN computer programs for maximum entropy spectral analysis of a complex or real vector time series. The descriptions include sufficient detail to permit the programs to be implemented correctly. All programs have been carefully checked out, and are available from the author on request.

1. INTRODUCTION

This report describes several computer programs written to implement multichannel maximum entropy spectral analysis. The main objective of these programs is to obtain an estimate of the multivariate power spectral density for a real or complex multichannel time series. Although most of the programs described here can be used for single-channel series by setting the number of channels equal to 1, a separate report (Strand, et al., 1977) and separate programs have been written for the single-channel case because of reduced storage and program complexity. In the present report we strive to provide the user with enough information to enable him to implement the programs correctly. Many mathematical derivations have been intentionally omitted, although we attempt to state most of the formulas required for the solutions. For details one may consult Strand (1977). Ioannidis (1975) gives a clear exposition of multichannel methods similar to some of those presented here; derivations and discussions of the single-channel case are presented by Burg (1975), Kanasewich (1973), Haykin and Kesler (1976), and Ulrych and Bishop (1975). In this report many specific details regarding the computer programs are also deliberately omitted. If such specific details are desired, the user should consult the relevant source deck; in some cases COMMENT cards indicate programming options.

We consider the complex p-channel vector time-series

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \cdot \\ \cdot \\ \cdot \\ y_p(t) \end{bmatrix}, \quad t = \left\{ n\Delta t \right\}_{t=-\infty}^{\infty}, \qquad (1.1)$$

where we assume that y(t) is wide-sense stationary with zero mean. The <u>N-element forward filter</u> (or <u>forward filter of length N</u>) has the form

$$y(t) + \sum_{k=1}^{N} F_{kN}^{*}\, y(t - k\Delta t) = e_N(t). \qquad (1.2)$$

This is called a forward filter because y(t) is predicted ahead, in terms of previous values $y(t - k\Delta t)$. The term $e_N(t)$ is called the output of the filter and plays the role of an error. Similarly, we may define the N-element backward filter

$$y(t) + \sum_{k=1}^{N} B_{kN}^{*} \, y(t + k\Delta t) = b_N(t). \tag{1.3}$$

In (1.2) and (1.3), the forward and backward filter coefficients, F_{kN} and B_{kN}, respectively, are complex p x p matrices. The star * is used to denote Hermitian conjugation; for instance, F_{kN}^{*} is the conjugate transpose of F_{kN}. For optimum forward or backward filters (2.2) or (2.3) we impose the condition that the expected mean-square value of $e_N(t)$ or $b_N(t)$ should be a minimum. For (2.2) this involves determining F_{1N}, F_{2N} ... F_{NN} such that

$$E\{[e_N(t)]^{*}[e_N(t)]\} = \text{minimum} \tag{1.4}$$

where E{ } denotes the expected value. Inserting the expression (1.2) for $e_N(t)$ into (1.4) and minimizing with respect to the coefficients F_{kN}^{*} gives the system

$$\begin{bmatrix} R_0 R_1 & \cdots & R_N \\ R_{-1} R_0 R_1 & \cdots & R_{N-1} \\ \cdots & & \cdots \\ R_{-N} R_{-N+1} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} I \\ F_{1N} \\ \cdot \\ \cdot \\ \cdot \\ F_{NN} \end{bmatrix} = \begin{bmatrix} P_N \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{1.5}$$

where I is the p x p identity matrix and the square block submatrices in this system are defined by

$$R_k = E[y(t)y^{*}(t - k\Delta t)], \; k = 0,1,\ldots N \tag{1.6}$$

so that

$$R_{-k} = R_k^{*} . \tag{1.7}$$

We sometimes call the coefficient matrix in (1.5) the R-matrix or simply R. The forward power matrix P_N for the resulting optimum filter (that is, for

F_{kN} satisfying (1.5)) is

$$P_N = E\{e_N(t)e_N^*(t)\} \ . \tag{1.8}$$

Because the F_{kN} satisfy (1.5), it can be shown that

$$P_N = F_N^* \ R \ F_N, \tag{1.9}$$

where we define F_N (single subscript) by

$$F_N^* = [I|F_{1N}^*| \ \ldots \ F_{NN}^*] \ .$$

System (1.5) is identical to the transposed form of the system resulting from the multichannel equations cited by Wiggins and Robinson (1965) for the real case, except that matrix transposes have been replaced by Hermitian conjugates. Because of (1.7) it is apparent that the optimum backward filter (1.3) satisfies

$$\begin{bmatrix} R_0 R_{-1} & \ldots & R_{-N} \\ R_1 R_0 & \ldots & R_{-N+1} \\ \ldots & & \ldots \\ R_N R_{N-1} & \ldots & R_0 \end{bmatrix} \begin{bmatrix} I \\ B_{1N} \\ \cdot \\ \cdot \\ \cdot \\ B_{NN} \end{bmatrix} = \begin{bmatrix} P_N' \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{1.10}$$

We sometimes call the coefficient matrix in (1.10) the R'-matrix, or simply R'. The backward power matrix P_N' for the resulting optimum filter is

$$P_N' = E\{b_N(t)b_N^*(t)\} \tag{1.11}$$

and it follows from (1.10) that

$$P_N' = B_N^* \ R' \ B_N \tag{1.12}$$

where

$$B_N^* = [I|B_{1N}^*| \ \ldots \ B_{NN}^*] \ .$$

If one has suitable approximations to the R_k, he may solve (1.5) to obtain the matrix coefficients F_{kN}, after which the power matrices P_N and P_N' are determined by (1.9) and (1.12). However, it can be shown that P_N and P_N' may be computed by the simple recursion formulas

$$P_N = P_{N-1} - C_N^* P_{N-1}' C_N \tag{1.13}$$

and

$$P_N' = P_{N-1}' - (C_N')^* P_{N-1} C_N' , \tag{1.14}$$

where the <u>forward</u> and <u>backward</u> matrix reflection coefficients C_N and C_N' are defined by

$$C_N = F_{NN} \qquad \text{and} \qquad C_N' = B_{NN} . \tag{1.15}$$

Because the expected values in (1.6) cannot be obtained from a finite sample of a single realization of y(t), it is necessary to use approximations. We denote by P_N and P_N' the result of substituting <u>approximate</u> values of R_k into (1.9) and (1.12) or equivalently (1.13) and (1.14). Thus an arbitrary set of R_k may or may not give rise to a positive definite set P_0, P_1, ... P_N. In case R_k does give rise to such a set of power matrices, it is known (Burg, 1975; Jones, 1974; Ioannidis, 1975) that the maximum entropy power spectral density S(f) can be calculated in terms of the forward filter elements by

$$S(f) = \Delta t [F^{-1}(\tfrac{1}{z})]^* \; P_N [F^{-1}(\tfrac{1}{z})] \tag{1.16}$$

where

$$F(z) = I + F_{1N} z + \ldots + F_{NN} z^N \tag{1.17}$$

and z is the complex scalar defined by

$$z = \exp[-2\pi i f \Delta t] . \tag{1.18}$$

The matrices F(1/z) are calculated in our programs by fast Fourier transform (FFT).

If $y_1, y_2, \ldots y_{N_d}$ is a sample of a realization of the time series y(t) in (1.1), then the usual biased estimator of R_k is

$$\tilde{R}_k = \frac{1}{N_d} \sum_{\ell=1}^{N_d-k} y_{k+\ell}\, y_\ell^*,\ k = 0,\ 1,\ \ldots\ N\quad , \tag{1.19}$$

where it is assumed that $N << N_d$. It can be shown that the estimator $\tilde{R}_k$ always gives rise to a positive definite R-matrix. The corresponding unbiased estimator, obtained by replacing the factor $1/N_d$ by $1/(N_d-k)$ in (1.19), does not have this virtue. It can also be shown (as is suggested by (1.9)) that $P_0, P_1, \ldots P_N$ (and $P_0', P_1' \ldots P_N'$) are positive definite if and only if R is positive definite. All methods used in our programs to solve (1.5) and (1.10) are recursive; that is, the solutions construct the filter coefficients $F_{1N}, F_{2N}, \ldots F_{NN}$ and $B_{1N}, B_{2N}, \ldots B_{NN}$ for the filter of length N from the corresponding solutions $F_{1,N-1}, F_{2,N-1} \ldots F_{N-1,N-1}$ and $B_{1,N-1}, B_{2,N-1} \ldots B_{N-1,N-1}$ of the systems (1.5) and (1.10) arising for filters of length N-1.

Two basic methods of solution are used in our programs. In the R-method, sometimes called Yule-Walker estimation, we generate approximations $\tilde{R}_k$ by (1.19) for k = 0, 1, ... N_{max}, where N_{max} is at least as large as the maximum number of lags to be considered, and then solve (1.5) and (1.10) recursively. We calculate spectra as desired by using (1.16), (1.17) and (1.18). Further details on our implementation of the R method are presented in section 3.

In the multichannel Burg process we first estimate the reflection coefficients C_N and C_N' by minimizing a certain weighted sum of squares of residuals for forward and backward filters, after which all forward and backward filter coefficients F_{kN} and B_{kN} are obtained by the modern Levinson algorithm. Further computational detail is presented in section 2; a complete theoretical discussion and all mathematical derivations are given elsewhere (Strand, 1977; Burg, 1975).

A rough indication of the performance of either the multichannel Burg process or the R method is given by Akaike's "Final prediction error" (FPE) criterion (Akaike, 1969, 1971), calculated by

$$\mathrm{FPE}(N) = (\det P_N)\left(\frac{N_d + 1 + pN}{N_d - 1 - pN}\right)^p \tag{1.20}$$

where det ( ) denotes the determinant. According to this criterion, the filter length N is properly chosen if it minimizes FPE(N). Formula (1.20) is implemented in our programs, and FPE(N) is usually presented both as printout and in graphical form.

2. MULTICHANNEL BURG REFLECTION-COEFFICIENT PROGRAMS

2.1 Discussion of Computing Methods

The programs described in this section implement a new generalization of Burg's single-channel reflection-coefficient method. We summarize the equations used; further details and mathematical derivations are given by Strand (1977). Our programs assume an unbroken sample $y_1, y_2, \ldots y_{N_d}$. If forward and backward positive definite weight matrices Q_1 and Q_2, respectively, are used, then the system to be solved to obtain the p x p reflection-coefficient matrix C_N is:

$$T_1 C_N + C_N T_2 = T_3 \tag{2.1}$$

where

$$T_1 = (P'_{N-1}\, Q_2\, P'_{N-1})^{-1}\, B$$

$$T_2 = (P^{-1}_{N-1}\, E\, P^{-1}_{N-1})\, Q_1^{-1} \tag{2.2}$$

$$T_3 = -\,(P'_{N-1}\, Q_2\, P'_{N-1})^{-1}\, G - (P'_{N-1})^{-1}\, G\, P^{-1}_{N-1}\, Q_1^{-1}$$

$$E = \frac{1}{M}\sum_{m=1}^{M} e_m^N \,(e_m^N)^*$$

$$G = \frac{1}{M}\sum_{m=1}^{M} b_m^N \,(e_m^N)^* \tag{2.3}$$

$$B = \frac{1}{M}\sum_{m=1}^{M} b_m^N \,(b_m^N)^* .$$

The solution of the $p^2 \times p^2$ system (2.1) is implemented in our multichannel Burg programs. The theory of systems such as (2.1) is given briefly by Bellman (1960), pp. 175-76 and 231. Thus our programs permit the use of arbitrary weight matrices Q_1 and Q_2. However, not all choices of Q_1 and Q_2 will give rise to positive definite power matrices P_N and P'_N. It has been shown by Strand (1977) that the choice

$$\begin{aligned} Q_1 &= P_{N-1}^{-1} \\ Q_2 &= (P'_{N-1})^{-1} \end{aligned} \qquad (2.4)$$

gives a direct generalization of Burg's single-channel reflection-coefficient method (Burg, 1975), and that all resulting power matrices are positive definite. We use (2.4) in all our programs, although it is a simple matter to implement other choices for Q_1 and Q_2, as is indicated by COMMENT cards in the source decks. In case the choice (2.4) is made, then the equations (2.1) to be solved can be written in the form

$$B\, C_N + P'_{N-1}\, C_N\, P_{N-1}^{-1}\, E = -\,2G. \qquad (2.5)$$

In (2.3), e_m^N is the residual resulting from applying the fitted forward filter of length N-1 to the last N vectors of the m^{th} N+1-tuple and b_m^N is the residual resulting from applying the fitted backward filter of length N-1 to the first N vectors of the m^{th} N+1-tuple. $M=N_d-N$ is the number of consecutive N+1-tuples in the data string of length N_d. As an example, let $N_d=7$, $N=3$, and suppose the forward and backward filters of length 2, as well as P_2 and P'_2 have been determined. Then the data consist of the 7 vectors $(y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ and for N = 3 the N+1-tuples are (y_1, y_2, y_3, y_4), (y_2, y_3, y_4, y_5), (y_3, y_4, y_5, y_6) and (y_4, y_5, y_6, y_7) so that M = 4. The required residuals resulting from the fitted filters of length (N-1) = 2 are

$$\begin{aligned} e_1^3 &= y_4 + F_{12}^* y_3 + F_{22}^* y_2 \\ e_2^3 &= y_5 + F_{12}^* y_4 + F_{22}^* y_3 \\ e_3^3 &= y_6 + F_{12}^* y_5 + F_{22}^* y_4 \\ e_4^3 &= y_7 + F_{12}^* y_6 + F_{22}^* y_5 \end{aligned} \qquad (2.6)$$

and

$$\begin{aligned} b_1^3 &= y_1 + B_{12}^* y_2 + B_{22}^* y_3 \\ b_2^3 &= y_2 + B_{12}^* y_3 + B_{22}^* y_4 \\ b_3^3 &= y_3 + B_{12}^* y_4 + B_{22}^* y_5 \\ b_4^3 &= y_4 + B_{12}^* y_5 + B_{22}^* y_6 \end{aligned} \tag{2.7}$$

from which E, G and B are determined by (2.3) for N = 3 and C_3 may be obtained by solving (2.1). It has been shown (Strand, 1977) that the residuals e_m^N and b_m^N obey the recursion

$$\begin{aligned} e_m^{N+1} &= e_{m+1}^N + C_N^* b_{m+1}^N \\ b_m^{N+1} &= b_m^N + (C_N')^* e_m^N \end{aligned} \tag{2.8}$$

for m = 1,2, ..., (M-1) = N_d-N-1 and that

$$C_N' = P_{N-1}^{-1} C_N^* P_{N-1}'. \tag{2.9}$$

(The apparent lack of symmetry in (2.8) is caused by the particular notation employed.) Furthermore, the filter coefficients obey the recursions

$$\begin{aligned} F_{NN} &= C_N \\ F_{kN} &= F_{k,N-1} + B_{N-k,N-1} C_N, \quad k = 1,2, \ldots, (N-1) \end{aligned} \tag{2.10}$$

and

$$\begin{aligned} B_{NN} &= C_N' \\ B_{kN} &= B_{k,N-1} + F_{N-k,N-1} C_N', \quad k = 1,2, \ldots, (N-1). \end{aligned} \tag{2.11}$$

In our example above, we compute C_3' by (2.9), e_m^4 and b_m^4, m = 1,2,3, by (2.8), and we update the power matrices by (1.13) and (1.14) and the filter coefficients by (2.10) and (2.11). The recursive solution is initialized by setting

$$\begin{aligned} e_m^1 &= y_{m+1} \\ b_m^1 &= y_m, \quad m = 1,2, \ldots, N_d-1 \end{aligned} \tag{2.12}$$

and

$$P_o = P_o' = \frac{1}{N_d} \sum_{m=1}^{N_d} y_m y_m^* \tag{2.13}$$

in (2.3) and (2.2). Equation (2.1) is equivalent to a linear system of size $p^2 \times p^2$, and is computed by using a complex matrix inversion subroutine, CMINV, where the matrix to be inverted is first calculated as a Kronecker product. Also note that, according to (2.8), computer storage is required only for the set of e_m^N and b_m^N corresponding to the current value of N. To compute the p x p block autocovariance matrices R_k, one may recursively apply the formula

$$R_k = - \sum_{j=0}^{k-1} F_{j+1,k}^* R_{k-j-1}, \; k = 1,2, \ldots \tag{2.14}$$

obtained by taking the Hermitian conjugate of the last of equations (1.5) with N replaced by k. Although our present Burg programs do not provide the R_k, as they have not been needed, we have used (2.14) for checkout. A series of R_k were computed by (2.14) and punched onto cards which were then used as input to one of the R-method programs of section 3. Exact agreement between the results of the multichannel Burg process and the R method was observed.

For further clarification, we briefly review the computing procedure for the multichannel Burg programs. First, we compute P_o and P_o' by (2.13) and E, B and G by (2.12) and (2.3) with N = 1 and M = N_d - 1. Then (2.1) is solved for C_1 and (2.9) is used to compute C_1'. The filter coefficients are then computed by (2.10) and (2.11), where only the formulas $F_{11} = C_1$ and $B_{11} = C_1'$ apply for N = 1. We compute the forward and backward residuals e_m^2 and b_m^2

resulting from N = 1 by formulas (2.8) and the resulting power matrices P_1 and P_1' by (1.13) and (1.14). We continue the algorithm for N = 2, computing E, G and B by (2.3), solving (2.1) for C_2, computing C_2' by (2.9), etc., repeating the process for successively larger values of N. Whenever desired, the spectra are computed by (1.16), (1.17) and (1.18).

2.2 General Discussion of Computer Programs

All programs have been written in a relatively unsophisticated dialect of FORTRAN. Because zero subscripts are not allowed, all subscripts that can be zero in the formulas of this section have been increased by 1. Thus, for instance, R(I+1) is the same as R_i. Because the filter coefficients are being updated by (2.10) and (2.11), and need not remain in storage separately for each N, they are denoted by

```
(((CF(I,J,K),CB(I,J,K),I=1,NP),J=1,NP),K=1,N+1);
```

that is, NP = p, K corresponds to k+1 in (2.10) and (2.11), and only three subscripts are used.

The program BRGPC1 will be described in detail here, as it is the most general of the programs implementing the new multichannel Burg algorithm. As presently dimensioned, it accommodates complex time series having up to p = 4 components. No effort has been made to minimize the amount of storage used. If a saving of storage is required, those matrices that are Hermitian symmetric could be stored in "Hermitian mode" to eliminate redundancy; this is left to the user. Program BRGPC1 has been implemented on the CDC 6600 at Boulder, Colorado, and uses two IMSL (1975) subroutines, VCVTCH and EIGCH, as noted under subroutine CPRINT. The program also uses AUTOPLT, a microfilm plotting subroutine written by J. Leise. For successful operation on the 6600, both IMSL and AUTOPLT must be attached. For other implementations the eigenvalue calculation could be eliminated or some other eigenvalue routine for complex Hermitian symmetric matrices could be used. If the eigenvalue calculation is eliminated, however, some other calculation of the determinant in (2.15) must be provided if one desires to calculate FPE(N), as the current program uses the product of the eigenvalues. If a computer system other than our local 6600 is used, different graphics would also be required. Several of our programs are provided without graphics, so that the user may implement his own plotting routines.

2.3 Program BURGPC1

We begin the description of the program BRGPC1 with a simplified flow chart, presented as fig. 1, and follow this by a detailed description of the function of each subroutine.

2.3.1 BRGPC1, the Main Program

This program reads in the data and calls subroutines as indicated in fig. 1. The data input is described as follows

NC Maximum filter length.

NP Number of channels (components) in time series (at present $1<NP\leq 4$).

NSPE Minimum filter length for which spectrum is desired.

NDAT Number of data points.

NPRS Minimum filter length for which full printout is desired.

((YT(I,K),I=1,NP),K=1,NDAT) Time series data, complex.

In performing the calculations, this program calls subroutines DETREND and CALFIL. The READ formats should be modified by the user as desired. The present program uses card input.

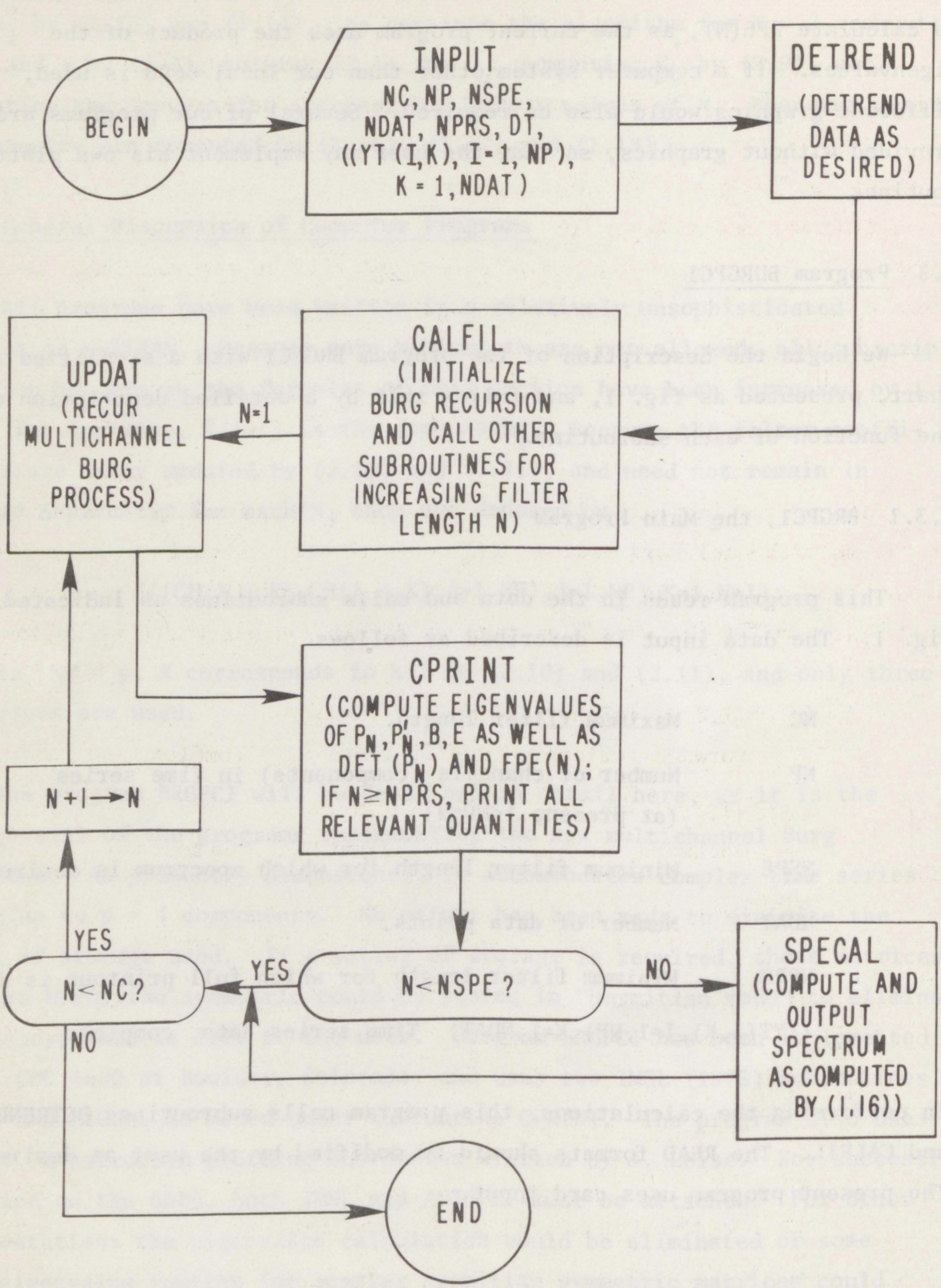


Figure 1. Simplified flow chart for the program BRGPC1

2.3.2 Subroutine DETREND

This subroutine detrends the time-series data in three possible ways. If the control cards ISKIP=1 and IF(ISKIP.EQ.1)GO TO 30 are placed between the COMMENT cards marked with repeated 1's, the data will not be detrended; if these control cards are placed between the 2's, the data will be reduced by subtracting the mean vector; finally, if these control cards are placed between the 3's a least-squares linear trend will be removed from the data. These options are also indicated by COMMENT cards in the source deck. The printouts of this subroutine are self explanatory.

2.3.3 Subroutine CALFIL

This subroutine initializes all quantities required for the recursion of the multichannel Burg process and calls the recursion in a DO loop on N. In this subroutine N is called I9 and N+1 is called IFP1. For orientation we define some of the variables used in CALFIL and elsewhere as follows:

(((CF(I,J,K),I=1,NP),J=1,NP),K=1,N+1)

Forward reflection coefficients F_{kN}, complex

(((CB(I,J,K),I=1,NP),J=1,NP),K=1,N+1)

Backward reflection coefficients B_{kN}, complex

SS(I,J)

Sum of squares P_o as given by (2.13), complex

BT(I,K),ET(I,K)

Forward and backward residuals, complex

P(I,J)

Forward power matrix P_N, complex

PP(I,J)

Backward power matrix P_N', complex

AEIG(I,J)

Buffer matrix used for eigenvalue routines

ELP(N+1,1)

Maximum eigenvalue of P_N, real

ELP(N+1,2)

Minimum eigenvalue of P_N, real

ELPP(N+1,1)

Maximum eigenvalue of P_N', real

ELPP(N+1,2)

Minimum eigenvalue of P_N', real

DETP(N+1)

Determinant of P_N, real

FPE(N+1)

Akaike FPE criterion, FPE(N), as defined by (1.20), real.

Calls to the two IMSL subroutines typically appear in the forms

```
CALL VCVTCH(AEIG,NP,4,AEIG)
```

and

```
CALL EIGCH(AEIG,NP,0,EL,V1,4,WK,IER).
```

If one changes the DIMENSION NPMAX describing the maximum number of channels for the time series y(t), then the constant 4 in all such calls must be changed to equal NPMAX. Such modifications must also be made in SPECAL and CPRINT. The printout of CALFIL consists of the matrix SS(I,J) and its trace.

2.3.4 Subroutine SPECAL

This subroutine computes the spectrum by (1.16), (1.17) and (1.18). The computation of F(1/z) is accomplished by fast Fourier transform (FFT), (Cooley and Tukey, 1965). The parameters N2 and M2 appearing in the statement COMMON/FFDATA/A(256),M2,N2,W(10), as well as N3 = N2/2 must be set (as the first executable statement) to correspond to the DIMENSION of A. We require N2=DIMENSION of A and 2**M2=N2. Because of peculiarities in the subroutine CMINV, described below, the same DIMENSION must be used for every matrix to be inverted. This DIMENSION is determined by the Kronecker product computed in subroutine KRONPR described below. Thus whenever a matrix is to be inverted, it is first stored as a buffer matrix EINV whose dimension is $(NPMAX)^2 X(NPMAX)^2$, where NPMAX is the DIMENSION describing the maximum number of channels in y(t); in the present case NPMAX=4 and EINV has dimension 16x16.

All printouts of the spectrum and the corresponding microfilm plots are generated in SPECAL. Other outputs generated by SPECAL are maximal and minimal eigenvalues of the spectrum, determinants of P_N, and Akaike's FPE criteria as given by (1.20). Figures 2-11 present the graphical microfilm output for the signal

$$\begin{aligned}
&\mathrm{Re}(YT(I,K)) = \cos(\frac{2\pi K}{16} + (I-1)\mathrm{rad.}) + .25(\mathrm{Ranf}-.5) \\
&\mathrm{Im}(YT(I,K) = \sin(\frac{2\pi K}{16} + (I-1)\mathrm{rad.}) + .25(\mathrm{Ranf}-.5) \\
&I = 1,2;\ NP = 2;\ K = 1,2,\ \ldots,\ 128;\ NDAT \equiv N_d = 128; \\
&DT = 1\ \mathrm{sec};\ NSPE = NC = 7.
\end{aligned} \tag{2.15}$$

The quantities Ranf are independent samples from a random number generator having a uniform distribution between 0 and 1.

2.3.5 Subroutine CPRINT

The main function of this subroutine is to print most quantities relevant to the multichannel Burg recursion. It also calculates and prints out eigenvalues of the matrices E, B, P_N and $P_N^{'}$, defined by (2.3), (1.8) and (1.11), and determinants of P_N and $P_N^{'}$. If either P_N or $P_N^{'}$ is found not to be positive semi-definite, the program stops with a diagnostic printout. As has been mentioned, this can only happen by computation error if Q_1 and Q_2 are assigned as inverse power matrices as in (2.4). If other choices of Q_1 and Q_2 are made and the program runs satisfactorily, it can be concluded (because of the tests made) that all power matrices encountered

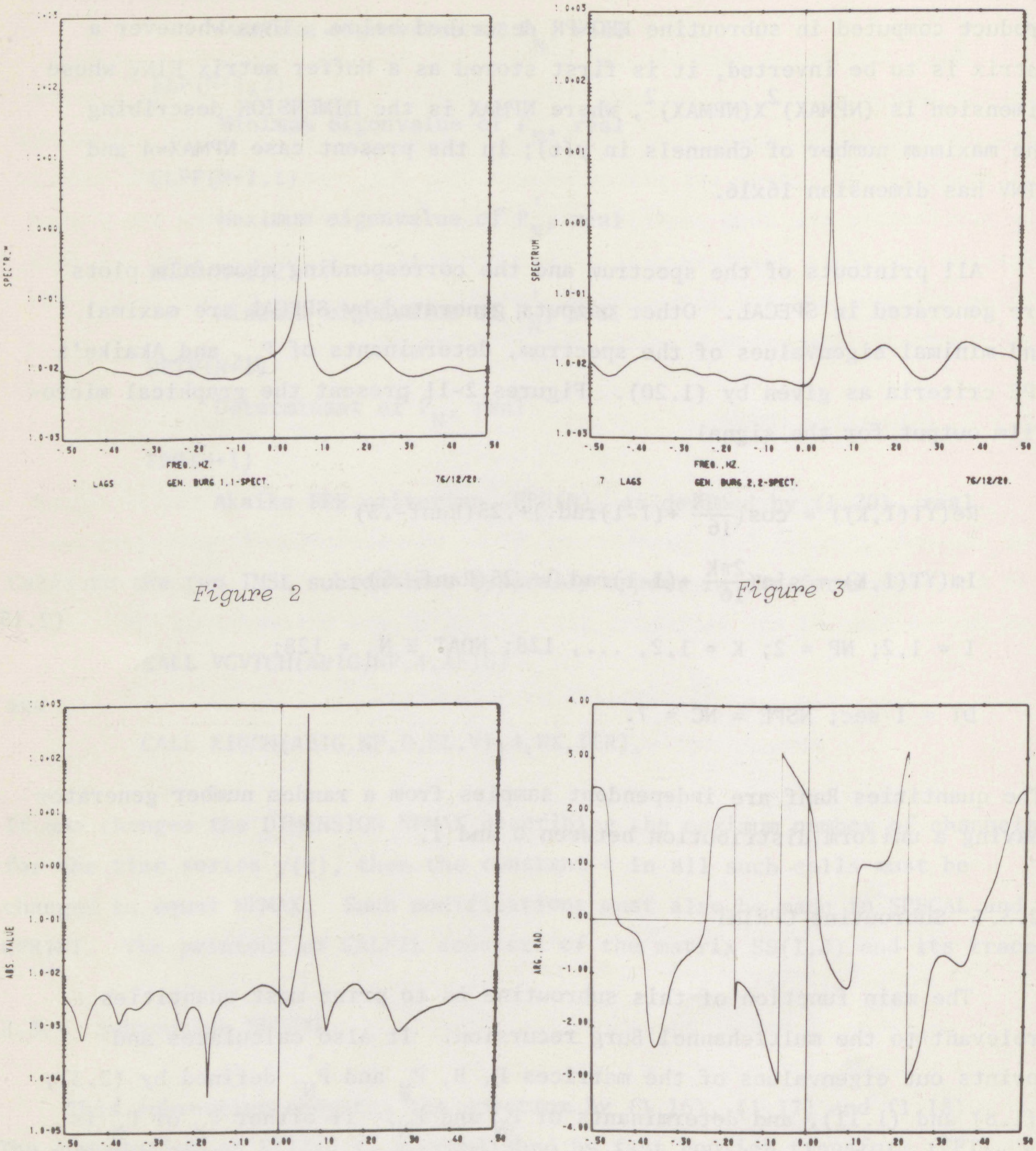


Figure 2

Figure 3

Figure 4

Figure 5

Figures 2-5. Graphical microfilm output for the program BRGPC1

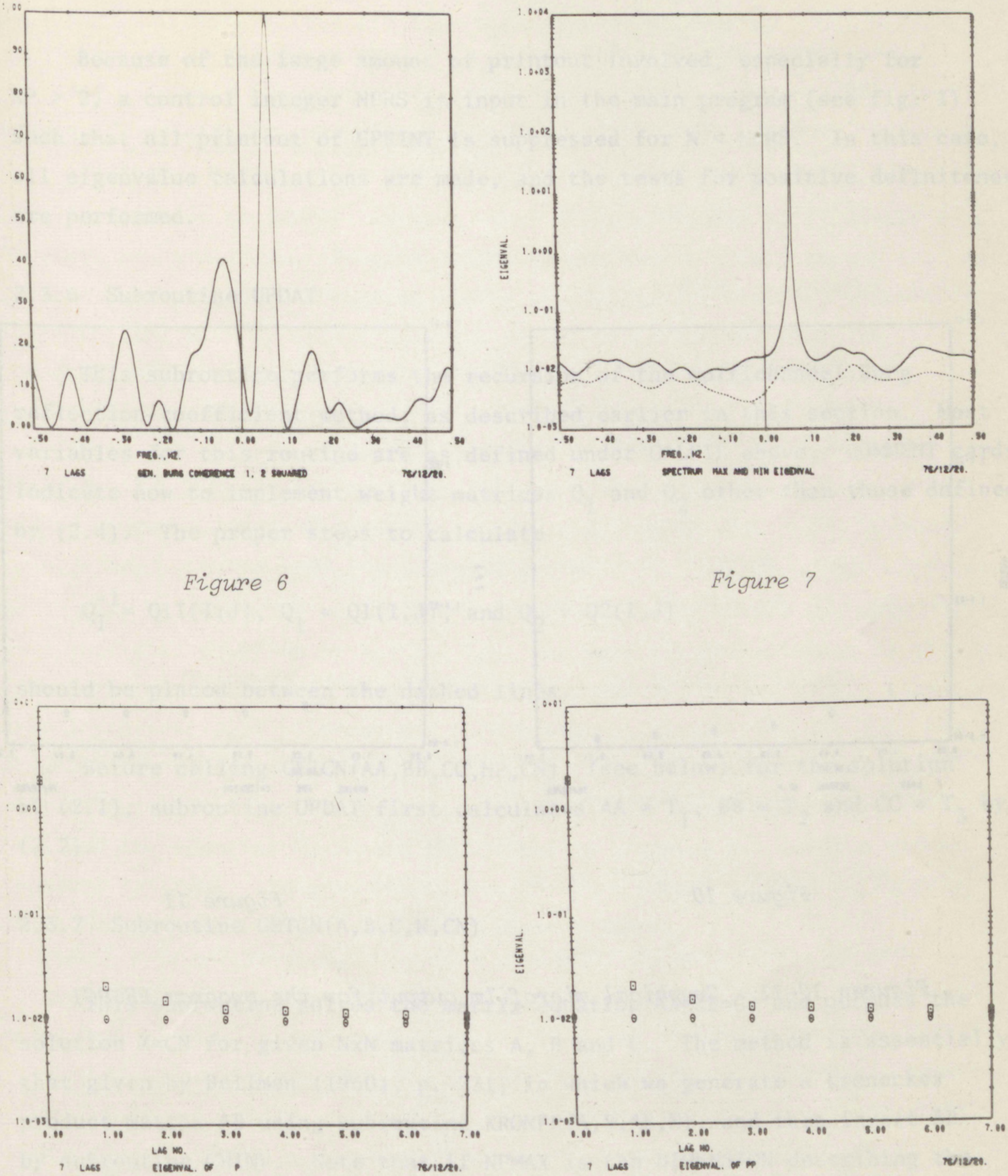


Figure 6

Figure 7

Figure 8

Figure 9

Figures 6-9. Graphical microfilm output for the program BRGPC1

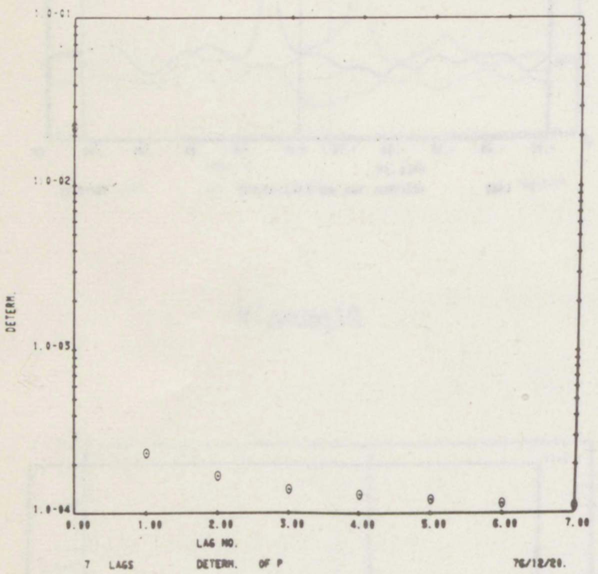


Figure 10

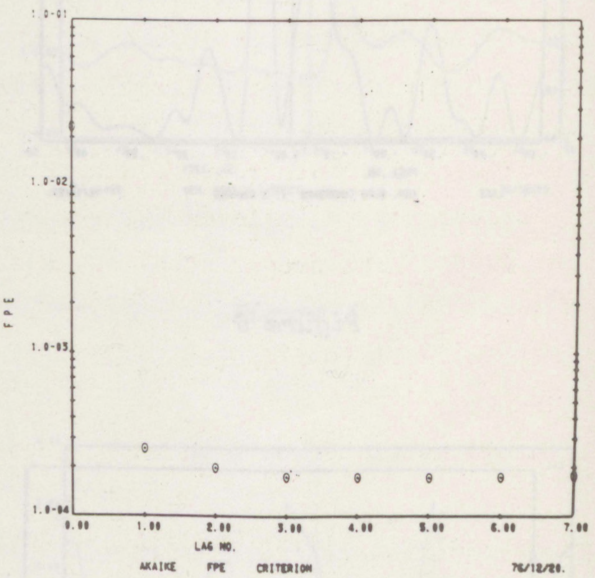


Figure 11

Figures 10-11. Graphical microfilm output for the program BRGPC1

are positive definite.

Because of the large amount of printout involved, especially for NP > 2, a control integer NPRS is input in the main program (see fig. 1) such that all printout of CPRINT is suppressed for N < NPRS. In this case, all eigenvalue calculations are made, and the tests for positive definiteness are performed.

2.3.6 Subroutine UPDAT

This subroutine performs the recursion of the multichannel Burg reflection-coefficient method, as described earlier in this section. Most variables for this routine are as defined under CALFIL above. COMMENT cards indicate how to implement weight matrices Q_1 and Q_2 other than those defined by (2.4). The proper steps to calculate

$$Q_1^{-1} = \text{Q1I(I,J)},\ Q_1 = \text{Q1(I,J)},\ \text{and}\ Q_2 = \text{Q2(I,J)}$$

should be placed between the dashed lines.

Before calling GETCN(AA,BB,CC,NP,CN), (see below) for the solution of (2.1), subroutine UPDAT first calculates AA = T_1, BB = T_2 and CC = T_3 by (2.2).

2.3.7 Subroutine GETCN(A,B,C,N,CN)

This subroutine solves the matrix equation AX+XB=C, and outputs the solution X=CN for given NxN matrices A, B and C. The method is essentially that given by Bellman (1960), p. 231, in which we generate a Kronecker product matrix AB using subroutine KRONPR(A,B,AB,N), and then invert AB by subroutine CMINV. Note that if NPMAX is the DIMENSION describing the maximum number of channels for the time series YT, then A,B,C and CN should have DIMENSION (NPMAX,NPMAX),BB requires DIMENSION (NPMAX**2) and AB requires DIMENSION (NPMAX**2,NPMAX**2).

2.3.8 Subroutine KRONPR(A,B,AB,N)

This subroutine generates the $N^2 \times N^2$ complex matrix AB defined by $AB=A\otimes I+I\otimes B^T$, where ^{T} denotes the ordinary transpose and $\otimes$ denotes the Kronecker product. The matrix AB is the coefficient matrix for the elements of C_N in the equation $AC_N+C_NB=C$. For further clarification, consult Bellman (1960) and Strand (1977).

2.3.9 Subroutine CMINV(A,N,B)

This subroutine inverts the complex N x N matrix A and replaces the matrix argument A by A^{-1}. If desired, it also simultaneously solves the complex vector equation AX=B and replaces the argument B by the solution vector X. In its present form, the matrix to be inverted must have DIMENSION consistent with that given here; this is achieved by using a buffer matrix of the proper DIMENSION whenever CMINV is called from various other subroutines. In the present version, we should have DIMENSIONS

IPIVOT(N2),A(N2,N2),B(N2),INDEX(N2,2) and PIVOT(N2),

where N2 is the square of the DIMENSION describing the maximum number of channels for the given time series YT.

2.3.10 Subroutines FFT,INIFFT,REVBIT

This is a fast Fourier transform package which uses the well-known Cooley-Tukey (1965) algorithm. It calls FORTRAN subroutines INIFFT and REVBIT, both of which are included in the source deck. If it is desired to change the DIMENSIONS of the variables in FFT, one should change the COMMON/FFDATA/ statements appearing in SPECAL, FFT, INIFFT and REVBIT to give A( ) a DIMENSION equal to the desired power of 2. Note that in subroutine SPECAL the statements N2=256$ M2=8$ N3 = 128 should be changed such that N2 is the DIMENSION of A( ), 2**M2=N2, and N3=N2/2. In the multichannel Burg programs the FFT routines are used only to compute F(z) by (1.17) as needed to obtain the spectrum by (1.16). In the R-method programs of section 3 they are also used to obtain covariance estimates such as (1.19).

2.4 Other Multichannel Burg Programs

In addition to BRGPC1, various other closely related multichannel Burg programs are available. Program BRGPC2 is similar to BRGPC1 except that no graphics are provided. Programs BRGPR1 and BRGPR2 are similar, respectively to BRGPC1 and BRGPC2, except that the signal YT is assumed real; the computation is simpler and less core storage is required. Programs BRG2C1, BRG2C2, BRG2C3 and BRG2C4 are pilot programs written earlier than the others. They are only valid for NP=2; however this may suffice for many applications, for example, obtaining the coherence between two single-channel real or complex signals. These subroutines use no external subroutines except graphics for BRG2C1 and BRG2C2. General weight matrices Q_1 and Q_2 may be used; the eigenvalue subroutine EIG implements a simple solution to a quadratic equation in λ. The test for positive definiteness is simply a test for the positiveness of the 1,1-element and the determinant. Program BRG2C1 provides full printout and 6600 graphics; BRG2C2 provides full 6600 graphics but prints only the original time series data before and after detrending; BRG2C3 has no graphics and prints only the original time-series data and the spectral data; and BRG2C4 has no graphics but gives a rather complete printout. Program BRG2C4 has the main advantage that it is self contained and requires no attached systems routines.

3. R-METHOD (YULE-WALKER) PROGRAMS

3.1 Theoretical Considerations and Computing Methods

A rather simple and efficient recursive solution of (1.5) and (1.10) can be constructed by using the modern Levinson algorithm, (Levinson, 1947), as has been pointed out by Burg (1975); however, the R-method solution presented here was based on earlier analysis, and employs the simultaneous triangular decomposition of the R and R' matrices (Rissanen, 1973), as presented below. Our method does not differ substantially in efficiency from the modern Levinson algorithm and, as the solutions of (1.5) and (1.10) are unique, is necessarily equivalent to it. No loss of computing accuracy has ever been observed, although the method has not been severely tested by using many lags and a large number of channels.

Rissanen (1973) has developed an algorithm giving the recursive simultaneous decomposition of the matrices R and R' into the block-diagonal forms

$$TRT^{*}=D \tag{3.1}$$

and

$$T'R'(T')^{*}=D' \tag{3.2}$$

where

$$T = \begin{bmatrix} I & & & & O \\ T_{21} & I & & & \\ T_{31} & T_{32} & I & & \\ \cdot & \cdot & \cdot & \ddots & \\ \cdot & \cdot & & \cdot & \\ T_{N+1,1} & T_{N+1,2} & \cdots & T_{N+1,N} & I \end{bmatrix} \tag{3.3}$$

$$T' = \begin{bmatrix} I & & & & O \\ T'_{21} & I & & & \\ T'_{31} & T'_{32} & I & & \\ \cdot & \cdot & \cdot & \ddots & \\ \cdot & \cdot & & \cdot & \\ T'_{N+1,1} & T'_{N+1,2} & \cdots & T'_{N+1,N} & I \end{bmatrix} \tag{3.4}$$

$$D = \text{block diag}[D_1, D_2, \ldots D_{N+1}] \tag{3.5}$$

$$D' = \text{block diag}[D'_1, D'_2, \ldots D'_{N+1}] \tag{3.6}$$

and, as before, the stars denote Hermitian conjugation. Substitute (3.3) and (3.5) into (3.1) and consider the last block row of (3.1). Because of the form of D and the last factor T^*, taking note of the (N+1,1) block element of TRT^* shows that

$$\sum_{k=0}^{N} T_{N+1,k+1} R_{-k} = 0, \tag{3.7}$$

where we let $T_{N+1,N+1} = I$. Now taking account of (3.7) and considering the (N+1,2)-element of TRT^* shows that

$$\sum_{k=0}^{N} T_{N+1,k+1} R_{-k+1} = 0. \tag{3.8}$$

Taking account of (3.7) and (3.8) and examining the (N+1,3)-element of TRT^*, and continuing in this manner gives

$$\sum_{k=0}^{N} T_{N+1,k+1} R_{-k+j} = 0,\ j = 0,1,\ \ldots,\ (N-1) \tag{3.9}$$

and finally,

$$\sum_{k=0}^{N} T_{N+1,k+1} R_{N-k} = D_{N+1}. \tag{3.10}$$

However, equations (3.9) and (3.10) may be written in the equivalent form

$$[I|T_{N+1,N}|T_{N+1,N-1}| \ldots |T_{N+1,1}] \begin{bmatrix} R_0 & R_{-1} & \ldots & R_{-N} \\ R_1 & R_0 & \ldots & R_{-N+1} \\ \ldots & & & \ldots \\ R_N & R_{N-1} & \ldots & R_0 \end{bmatrix}$$

$$= [D_{N+1}|0| \ldots |0]. \tag{3.11}$$

Comparison of the Hermitian conjugate of (3.11) with (1.10), which has a unique solution, shows that

$$\begin{bmatrix} I \\ T^{*}_{N+1,N} \\ \cdot \\ \cdot \\ \cdot \\ T^{*}_{N+1,1} \end{bmatrix} = \begin{bmatrix} I \\ B_{1N} \\ \cdot \\ \cdot \\ \cdot \\ B_{NN} \end{bmatrix} \tag{3.12}$$

and that

$$D_{N+1} = P'_{N}, \; N = 0, 1, \ldots \; . \tag{3.13}$$

In order to obtain a formula like (3.12) for the forward filter coefficients, we apply the same analysis as above to (3.2) and compare with (1.5) to obtain

$$\begin{bmatrix} I \\ T'^{*}_{N+1,N} \\ \cdot \\ \cdot \\ \cdot \\ T'^{*}_{N+1,1} \end{bmatrix} = \begin{bmatrix} I \\ F_{1N} \\ \cdot \\ \cdot \\ \cdot \\ F_{NN} \end{bmatrix} \tag{3.14}$$

or equivalently

$$F_{jN} = (T'_{N+1,N-j+1})^{*}, \; j = 1, 2, \ldots N, \tag{3.15}$$

and

$$D'_{N+1} = P_{N} \quad N = 0, 1, \ldots \; . \tag{3.16}$$

It follows from (3.1), (3.5) and (3.13) and (3.2), (3.6) and (3.16) that

$$\det R = \prod_{k=0}^{N} \det P'_{k} \tag{3.17}$$

and

$$\det R' = \prod_{k=0}^{N} \det P_{k}. \tag{3.18}$$

It is easily shown by direct block multiplication that

$$R' = URU^* \tag{3.19}$$

where

$$U = \begin{bmatrix} 0 \ldots 0\ 0\ I \\ 0 \ldots 0\ I\ 0 \\ \ldots \quad \ldots \\ 0\ I\ 0 \ldots 0 \\ I\ 0\ 0 \ldots 0 \end{bmatrix}.$$

Since det U = ± 1, it follows from (3.19) that

$$\det R = \det R'. \tag{3.20}$$

Because $\det P_o = \det P_o' = \det R_o$, (3.17), (3.18) and (3.20) imply that

$$\det P_k = \det P_k', \; k = 0, 1, \ldots \tag{3.21}$$

Finally, note that if y(t) is a single-channel time series, i.e., p = 1, then $P_k = P_k'$, k = 0, 1, ..., that is corresponding forward and backward powers are the same.

The computation of T and T' is accomplished by a complex version of the Rissanen (1973) algorithm, for which the modification of the original algorithm was slight. It was found that storage of only two rows of the T and T'-matrices was required, a "current" row and a "previous" row. Because of differences between Rissanen's notation and ours, notably his use of the star as indicating an interchange of R_k and R_{-k} (i.e., his star is the same as our prime) and the fact that he denotes our T and T' matrices by B and B^*, respectively, we denote the current rows of T and T' by (((B(I,J,K),BS(I,J,K),I=1,NP),J=1,NP),K=1,N+1) and the previous rows by B1(I,J,K) and BS1(I,J,K). Similarly, we denote D_{k+1} by D(k+1) and D_{k+1}' by DS(k+1). Finally, we note that the Rissanen decomposition was checked out by direct multiplication and comparison with the R matrix, and the solutions were checked by substitution into (3.5) and (3.10).

3.2 Program RYWPC1

This program provides an R-method solution in which the estimates $\tilde{R}_k$ of the autocovariances (see (1.19)) are calculated by FFT for all k up to the DIMENSION of R(I,J,K), after which the solution is initialized and proceeds recursively by the Rissanen algorithm. It assumes a complex time series YT, and will accommodate an arbitrary number, NP, of channels, provided that all variables are properly dimensioned, and sufficient storage exists. The present version assumes NP $\leq$ 4. It uses the same IMSL (1975) eigenvalue raoutines as BRGPC1 (see section 2) and also uses the same DDLIB 6600 microfilm plotting routines. We begin the specific description of program RYWPC1 with a simplified flow chart, presented as fig. 12, and follow this with a detailed description of the function of each subroutine.

3.2.1 Subroutine RYWPC1

This is the main program. It reads in the data and calls subroutines DETREND and INIT. All other calls to the various subroutines are made in subroutine INIT. The inputs have the same meaning as those described in section 2 under BRGPC1.

3.2.2 Subroutine DETREND

This subroutine detrends the data, and is identical to subroutine DETREND as described in detail in section 2.

3.2.3 Subroutine RCALC(R)

This subroutine calculates autocovariances by (1.19), using the fast Fourier transform (FFT). Because FORTRAN cannot use zero subscripts, $\tilde{R}_k$ is denoted by R(I,J,K+1). The DIMENSION NMAX, of R(, ,NMAX) should exceed that of the filter coefficients by at least 1. The integer NN should be set equal to NMAX. This subroutine then computes R(I,J,K) for K = 1, 2, ... NMAX in one loop of FFT calls. The constants M2 and N2

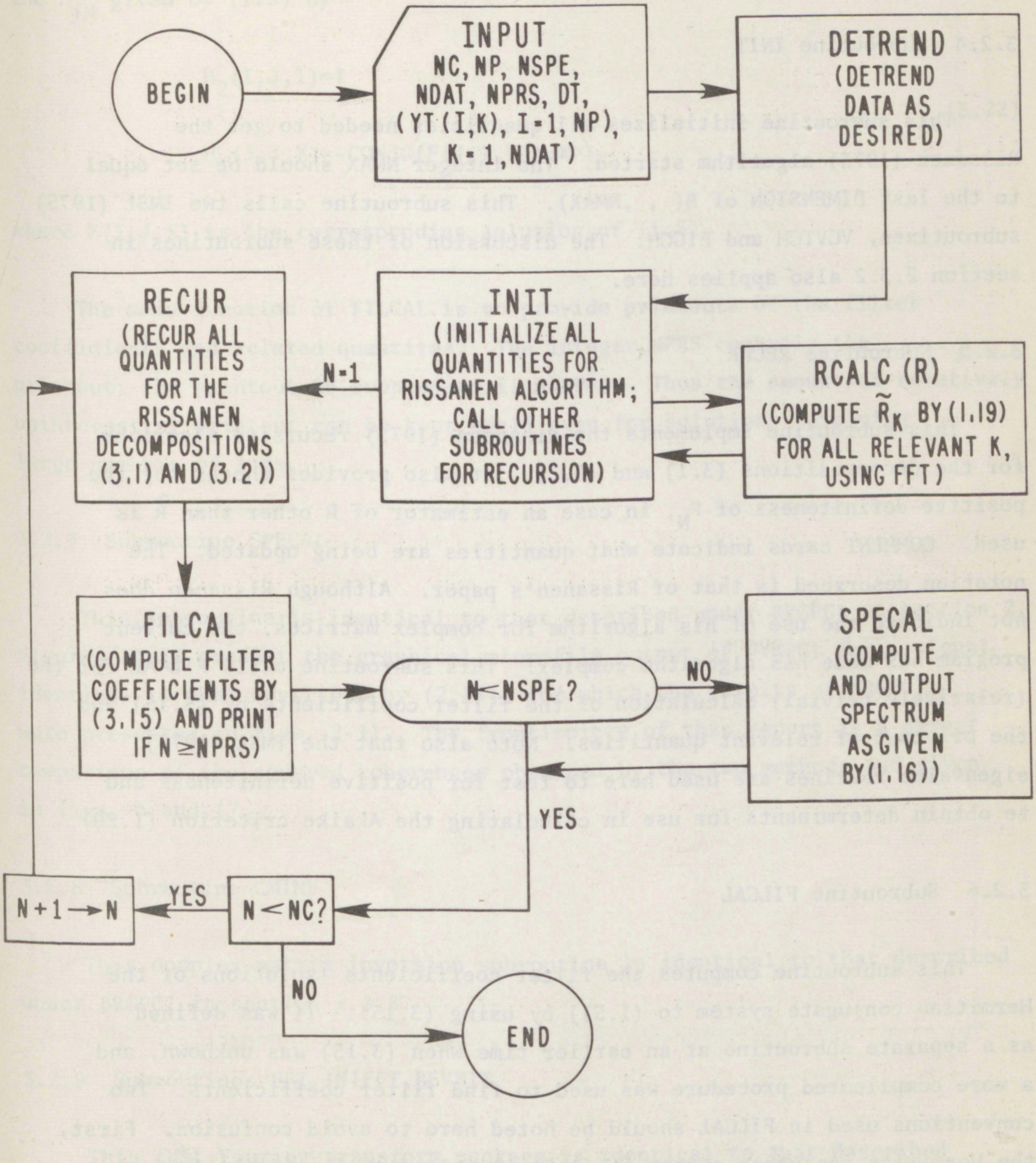


Figure 12. Simplified flow chart for the program RYWPC1

should be set as indicated under SPECAL in section 2. An unbiased estimator of R_k can be obtained by modifying statement 12.

3.2.4 Subroutine INIT

This subroutine initializes all quantities needed to get the Rissanen (1973) algorithm started. The integer NMAX should be set equal to the last DIMENSION of B(, ,NMAX). This subroutine calls two IMSL (1975) subroutines, VCVTCH and EIGCH. The discussion of these subroutines in section 2.3.2 also applies here.

3.2.5 Subroutine RECUR

This subroutine implements the Rissanen (1973) recursive algorithm for the decompositions (3.1) and (3.2). It also provides checks for the positive-definiteness of P_N, in case an estimator of R other than $\tilde{R}$ is used. COMMENT cards indicate what quantities are being updated. The notation described is that of Rissanen's paper. Although Rissanen does not indicate the use of his algorithm for complex matrices, the current program has made his algorithm complex. This subroutine calls FILCAL for the (relatively trivial) calculation of the filter coefficients by (3.15) and the printout of relevant quantities. Note also that the IMSL (1975) eigenvalue routines are used here to test for positive definiteness and to obtain determinants for use in calculating the Akaike criterion (1.20)

3.2.6 Subroutine FILCAL

This subroutine computes the filter coefficients (solutions of the Hermitian conjugate system to (1.5)) by using (3.15). It was defined as a separate subroutine at an earlier time when (3.15) was unknown, and a more complicated procedure was used to find filter coefficients. Two conventions used in FILCAL should be noted here to avoid confusion. First, the Hermitian conjugate system to (1.5) is solved to obtain the F^*_{jN} instead of F_{jN}. Furthermore, we use a sign convention occasionally seen in the literature in which all filter coefficients after the first are negated. Thus H2, the solution obtained in FILCAL, is related to the F_{jN} given by (1.5) by

$$\begin{aligned} &H_2(I,J,1)=I \\ &H_2(I,J,K)=-CONJG(F(J,I,K)),K>1, \end{aligned} \qquad (3.22)$$

where F(I,J,K) is the corresponding solution of (1.5).

The main function of FILCAL is to provide printouts of the filter coefficients and related quantites. The integer NPRS controls the printout; the printout is suppressed if N<NPRS. Thus the amount of relatively uninteresting printout can be kept small even for solutions involving large numbers of lags.

3.2.7 Subroutine SPECAL

This subroutine is identical to that described under BRGPC1 in section 2. Figures 13-22 present the graphical microfilm output of FYWPC1 for a signal identical to that described by (2.15), for which the results of BRGPC1 were presented in figs. 2-11. The frontispiece of this report is a direct comparison of the squared coherences obtained by the two methods and given in figs. 6 and 17.

3.2.8 Subroutine CMINV

This complex matrix inversion subroutine is identical to that described under BRGPC1 in section 2.3.8.

3.2.9 Subroutines FFT,INIFFT,REVBIT

This fast Fourier transform package is identical to that described under BRGPC1 in section 2.3.9. If FFT DIMENSION changes are made, they must be made in RCALC(R) and SPECAL, as well as here.

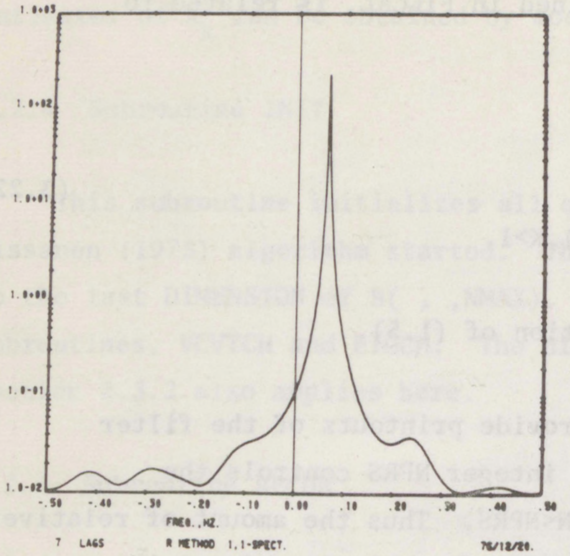


Figure 13

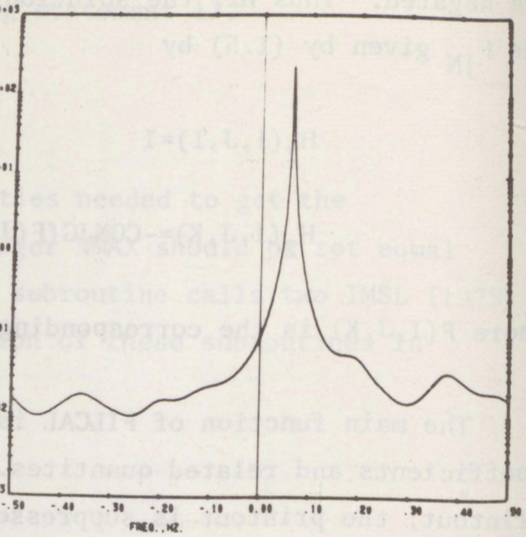


Figure 14

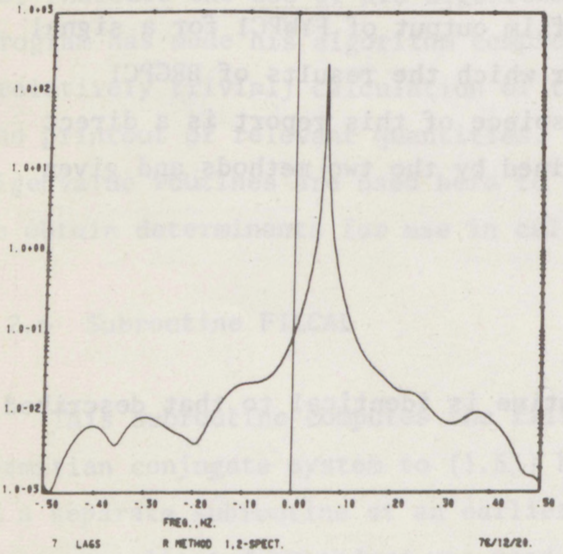


Figure 15

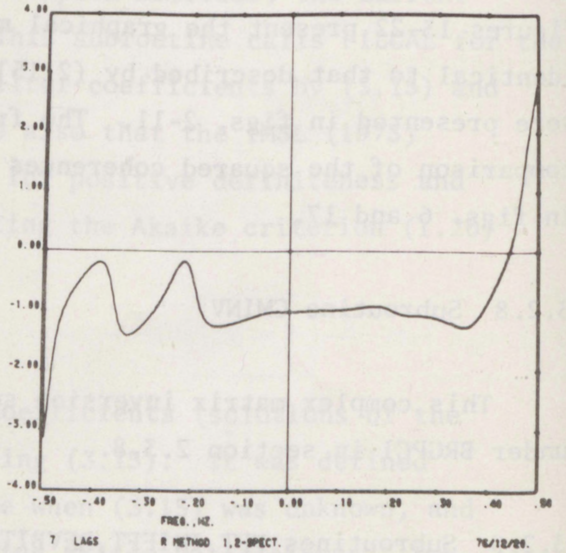


Figure 16

Figures 13-16. Graphical microfilm output for the program RYWPC1

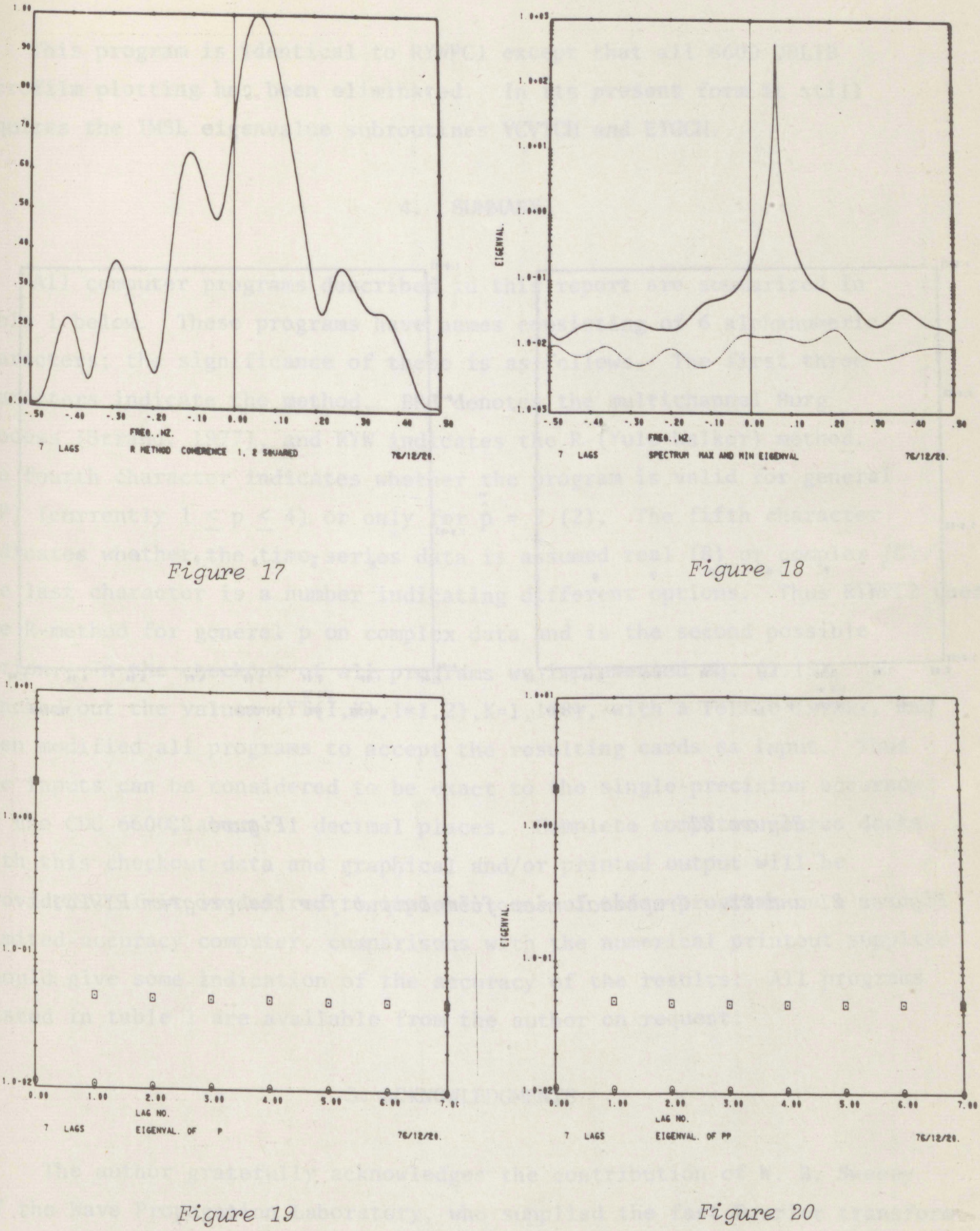


Figure 17

Figure 18

Figure 19

Figure 20

Figures 17-20. Graphical microfilm output for the program RYWPC1

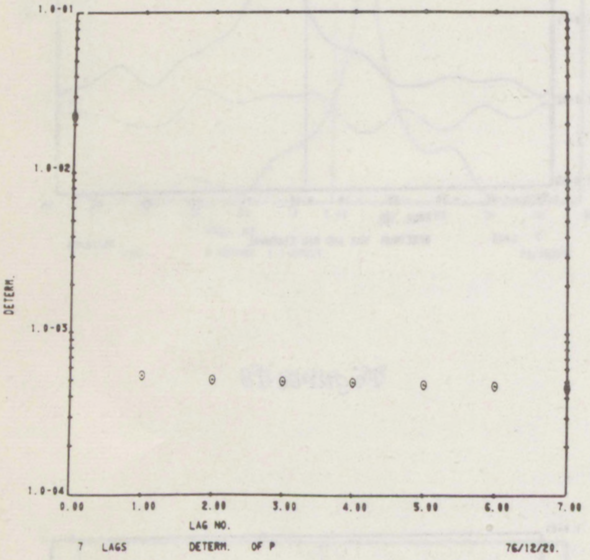


Figure 21

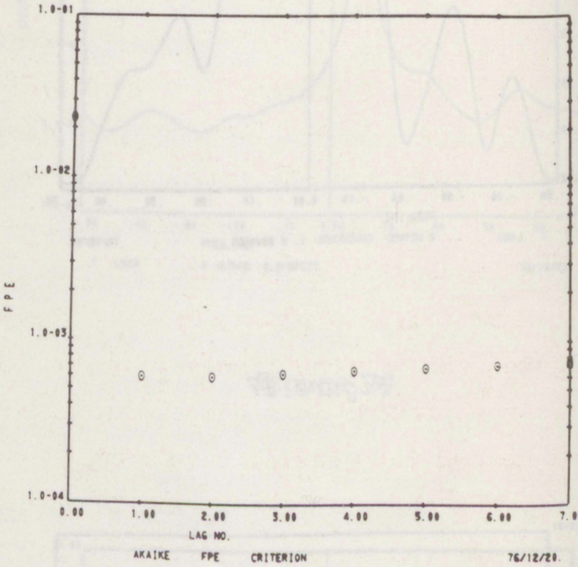


Figure 22

Figures 21 and 22. Graphical microfilm output for the program RYWPC1

3.3 Program RYWPC2

This program is identical to RYWPC1 except that all 6600 DDLIB microfilm plotting has been eliminated. In its present form it still requires the IMSL eigenvalue subroutines VCVTCH and EIGCH.

4. SUMMARY

All computer programs described in this report are summarized in table 1 below. These programs have names consisting of 6 alphanumeric characters; the significance of these is as follows. The first three characters indicate the method. BRG denotes the multichannel Burg process (Strand, 1977), and RYW indicates the R-(Yule-Walker) method. The fourth character indicates whether the program is valid for general p(P) (currently $1 \leq p \leq 4$) or only for p = 2 (2). The fifth character indicates whether the time-series data is assumed real (R) or complex (C). The last character is a number indicating different options. Thus RYWPC2 uses the R-method for general p on complex data and is the second possible option. In the checkout of all programs we implemented eq. (2.15), punched out the values (YT(I,K),I=1,2),K=1,128), with a format ±x.xxx, and then modified all programs to accept the resulting cards as input. Thus the inputs can be considered to be exact to the single-precision accuracy of the CDC 6600, about 11 decimal places. Complete computer source decks with this checkout data and graphical and/or printed output will be provided. If it is desired to implement one of these programs on a limited-accuracy computer, comparisons with the numerical printout supplied should give some indication of the accuracy of the results. All programs listed in table 1 are available from the author on request.

5. ACKNOWLEDGMENTS

The author gratefully acknowledges the contribution of W. B. Sweezy of the Wave Propagation Laboratory, who supplied the fast Fourier transform subroutines, and the contribution of J. A. Leise of CIRES, who supplied the 6600 microfilm plotting routines.

Table 1. List of Computer Programs for Multichannel Maximum Entropy Spectral Analysis

Program name	Type of data	Method	No. of Channels, p[1]	Approx. source deck thickness[2]	Approx. octal core[4]	Total octal storage[5]	FFT dimension	Time series dimension	External subroutines required[6]	Description
BRGPC1	Complex	Multich. Burg	$1 \leq p \leq 4$	7"	120 000	147 000	256	(4,300)	IMSL DDLIB	Complete printout and 6600 graphics.
BRGPC2	Complex	Multich. Burg	$1 \leq p \leq 4$	6 1/4"	106 000	134 000	256	(4,300)	IMSL	Complete printout, no graphics.
BRGPR1	Real	Multich. Burg	$1 \leq p \leq 4$	6 1/4"	73 000	124 000	256	(4,300)	IMSL DDLIB	Same as BRGPC1 but signal is real.
BRGPR2	Real	Multich. Burg	$1 \leq p \leq 4$	5 1/2"	62 000	110 000	256	(4,300)	IMSL	Same as BRGPC2 but signal is real.
BRG2C1	Complex	Multich. Burg	p=2	7"	73 000	123 000	512	(2,512)	DDLIB	Same as BRGPC1 but valid only if p=2 and requires no IMSL routines.
BRG2C2	Complex	Multich. Burg	p=2	5"	63 000	113 000	512	(2,512)	DDLIB	Same as BRG2C1 but has minimal printout.
BRG2C3	Complex	Multich. Burg	p=2	4 3/4"	54 000	101 000	512	(2,512)	NONE	Same as BRG2C1 but has no graphics and only prints spectral quantities.
BRG2C4	Complex	Multich. Burg	p=2	6 1/2"	62 000	108 000	512	(2,512)	NONE	Same as BRG2C1 but has no graphics; has full printout.
RYWPC1	Complex	R-method[3]	$1 \leq p \leq 4$	5 1/2"	160 000	210 000	512	(4,512)	IMSL DDLIB	Complete printout and 6600 graphics.
RYWPC2	Complex	R-method	$1 \leq p \leq 4$	5"	147 000	175 000	512	(4,512)	IMSL	Complete printout, no graphics.

[1] For $1 \leq p \leq 4$ below, one need only change DIMENSIONS for larger p.

[2] Exclusive of data.

[3] Sometimes called "Yule-Walker".

[4] Storage varies linearly with maximum number of data points and FFT DIMENSION and as the square of p_{MAX}.

[5] The CDC 6600 at Boulder, Colorado will run with this amount of storage.

[6] DDLIB refers to 6600 microfilm plotting routines. See references and section 2 for IMSL.

6. REFERENCES

Akaike, H. (1969): Fitting autoregressive models for prediction. *Ann. Inst. Statist. Math.*, 21, 243-247.

Akaike, H. (1971): Autoregressive model fitting for control. *Ann. Inst. Statist. Math.*, 23, 163-180.

Bellman, R. (1960): *Introduction to Matrix Analysis.* McGraw-Hill, New York.

Burg, J. P. (1975): Maximum entropy spectral analysis. Ph.D. dissertation, Dept. of Geophysics, Stanford University.

Cooley, J. W. and J. W. Tukey (1965): An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19, 297-301.

Haykin, S. and S. Kesler (1976): The complex form of the maximum entropy method for spectral estimation. Proc. IEEE, 64, 822-823.

IMSL (1975): (International Mathematical and Statistical Libraries, Inc.). Library 3, Edition 5 (FORTRAN) CDC 6000/7000, CYBER 70/170 series, LIB3-0005; IMSL, Sixth floor, GNB Bldg., 7500 Bellair Blvd., Houston, Tex. 77036, (713)-772-1927; Revised, Nov. 1975.

Ioannidis, G. A. (1975): Application of multivariate autoregressive spectrum estimation to ULF waves. *Radio Science*, 10, no. 12, 1043-1054.

Kanasewich, E. R. (1973): *Time Sequence Analysis in Geophysics.* University of Alberta Press, Edmonton, Canada.

Levinson, N. (1947): The Wiener RMS (root mean square) error criterion in filter design and prediction. *J. Math. Phys.*, 25, no. 4, 261-278.

Rissanen, J. (1973): Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring matrix polynomials. *Math. Comp.*, 27, no. 121, 147-154.

Strand, O. N. (1977): Multichannel complex maximum entropy (autoregressive) spectral analysis. To appear, *IEEE Trans. Autom. Control.*

Strand, O. N., J. M. Young, and R. B. Chadwick (1977): Computer programs for maximum entropy spectral analysis of real and complex single-channel time series. NOAA Tech. Memo ERL WPL-19, U.S. Dept. of Commerce, Environmental Research Laboratories, Boulder, Colorado 80302.

Ulrych, T. J. and T. N. Bishop (1975): Maximum entropy spectral analysis and autoregressive decomposition. *Rev. Geophys. and Space Phys.*, 13, no. 1, 183-200.

Wiggins, R. A. and E. A. Robinson (1965): Recursive solution to the multichannel filtering problem. *J. Geophys. Research,* 70, no. 8, 1885-1891.