LIBDRB: A Subroutine Library For Elementary Time Series Analysis

Keith A. Haycock
and
David R. Brillinger

Technical Report No. 48
November 1985

Department of Statistics
University of California
Berkeley, California
Introduction

LIBDRB is a collection of Fortran subroutines and functions for the elementary analysis of time series. Most of the routines are for spectral analysis, though some time domain routines are included (e.g., AUTREG).

LIBDRB developed from DRBLIB, a Fortran library written by David Brillinger in the late 60's and early 70's (except for AR1DFT, which was written by G. Sande in the mid 60's). Additional routines were added during the following years as need arose. (These include a second fast Fourier transform, FFTSNG, written by R. C. Singleton in 1968.) The early routines had been written quickly and were rather messy in places. Ross Ihaka cleaned up some of the code and in 1982 added versions of many of the routines to isp. In the spring of 1985, Keith Haycock began a major revision of the library. The remainder of the code was further cleaned up, some routines were completely rewritten (as were the isp programs) and a number of new routines were added.

The resulting library, renamed LIBDRB, differs in a number of ways from the earlier versions. The routines now perform only numerical calculations. The task of displaying and printing the results is kept separate. This permits greater flexibility, portability and efficiency. Several of the routines now allow the user more control over the estimation procedure. For example, the new spectral estimation routines allow the user to specify the filter used to smooth the periodogram. In addition, these routines allow cross-spectral analysis of multivariate series and use fit-based smoothers for increased efficiency. The fits now check the series length to ensure that it meets the factorization criteria and report on its failure to do so, while the functions NEXTLG and NEXTSM find nearby lengths which do satisfy the criteria. (Several DRBLIB routines for the spectral analysis of point processes are not included, but there are plans to include them in a future version.)

Documentation for the subroutines follows, but a few general comments can be made here. The routines are written so as to be compatible with either Fortran 66 or 77. The argument names follow Fortran typing conventions, i.e., arguments whose names begin with the letters I-N are integers, others are reals. Unless otherwise noted, routines return immediately upon encountering an illegal value (e.g., a series length less than 1).

The routines were tested on a number of data sets, including the Vienna-Berlin temperature data used in D. R. Brillinger (1975) Time Series: Data Analysis and Theory, Holt-Rinehart, but no claim is made that they are 100% error free. Comments/suggestions/problems with regard to both the library and the documentation are welcome and should be sent to Keith Haycock. Electronic mail can be sent to haycock@bach (at Berkeley) or haycock@bach.berkeley.edu or ucbvax@ucbbach.haycock (uucp). Suggestions may be implemented in future versions of LIBDRB.
Double Precision LIBDRB

We have found that, for series of moderate length, the single precision LIBDRB routines give unacceptably differing results on different cpu types (for example, the SUN and the VAX). There is now a double precision version of LIBDRB whose results have much better agreement across machines. On UNIX, the double precision routines are normally available—together with the single precision routines—using the compile/link-edit flag `-ldrb'.

The calls are identical to the single precision version (except of course that all floating point arguments are double precision instead of single). The routine names differ as follows:

<table>
<thead>
<tr>
<th>Single Precision</th>
<th>Double Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar1dft</td>
<td>dar1df</td>
</tr>
<tr>
<td>autreg</td>
<td>dautre</td>
</tr>
<tr>
<td>coher2</td>
<td>dcohr2</td>
</tr>
<tr>
<td>coherent</td>
<td>dcohrk</td>
</tr>
<tr>
<td>crscov</td>
<td>dcrscov</td>
</tr>
<tr>
<td>decim</td>
<td>ddecim</td>
</tr>
<tr>
<td>demean</td>
<td>ddemea</td>
</tr>
<tr>
<td>demod</td>
<td>ddemod</td>
</tr>
<tr>
<td>demodu</td>
<td>ddemodu</td>
</tr>
<tr>
<td>deseas</td>
<td>ddesea</td>
</tr>
<tr>
<td>detmd</td>
<td>ddetm</td>
</tr>
<tr>
<td>diff</td>
<td>ddiff</td>
</tr>
<tr>
<td>fft</td>
<td>dfft</td>
</tr>
<tr>
<td>fftsng</td>
<td>dfftsn</td>
</tr>
<tr>
<td>filter</td>
<td>dfilter</td>
</tr>
<tr>
<td>freqrm</td>
<td>dfreqr</td>
</tr>
<tr>
<td>fx</td>
<td>dfx</td>
</tr>
<tr>
<td>fy</td>
<td>dfy</td>
</tr>
<tr>
<td>imprs2</td>
<td>dimpr2</td>
</tr>
<tr>
<td>imprsk</td>
<td>dimprk</td>
</tr>
<tr>
<td>lag</td>
<td>dlag</td>
</tr>
<tr>
<td>mksym</td>
<td>dmkSYM</td>
</tr>
<tr>
<td>pgrmc</td>
<td>dpgrmc</td>
</tr>
<tr>
<td>pgrmr</td>
<td>dpgrmr</td>
</tr>
<tr>
<td>smthc</td>
<td>dsmtHC</td>
</tr>
<tr>
<td>smthr</td>
<td>dsmtHR</td>
</tr>
<tr>
<td>spec1</td>
<td>dspec1</td>
</tr>
<tr>
<td>spec1u</td>
<td>dspec1u</td>
</tr>
<tr>
<td>specck</td>
<td>dspeck</td>
</tr>
<tr>
<td>specku</td>
<td>dspeku</td>
</tr>
<tr>
<td>taper</td>
<td>dtaper</td>
</tr>
<tr>
<td>tmfr2</td>
<td>dtmfr2</td>
</tr>
<tr>
<td>tmfrk</td>
<td>dtmfrk</td>
</tr>
<tr>
<td>wghts</td>
<td>dwghts</td>
</tr>
</tbody>
</table>

The BLSS Project (formerly 'Berkeley ISP') now maintains LIBDRB source code in double precision; the single precision routines are mechanically created from the double precision routines.

D. Mark Abrahams and Fran Rizzardi
BLSS Project
Department of Statistics
University of California, Berkeley
June 1988
List Of Subroutines And Functions

ARIDFT(X, Y, N, IERR)
AUTREG(CXX, A, NA, V, AIC, NX)
COHER2(FXX, FYY, FXYR, FXYI, R, NF)
COHERK(F, R, NF, K)
CRSCOV(X, Y, NXY, CXY, NC)
DEMEAN(X, NX, XMEAN)
DEMODO(X, Y, NXY, FREQ, W, L, AMP, PH, NAP, XPOW, BW)
DEMODU(X, Y, NXY, FREQ, W, L, AMP, PH, NAP, XPOW, BW)
DESEAS(X, NX, SM, NM)
DETRND(X, NX, B)
DIFF(X, NX, D, ND)
FFT(X, Y, N, INV, IERR)
FFTSNG(X, Y, NTOT, N, NSPAN, ISN, IERR)
FILTER(X, NX, F, NF, Y)
FREQRM(X, NX, ANG, Y)
IMPRS2(GXY, PHXY, AR, Al, NF, JOB, IERR)
IMPRSK(G, PH, AR, Al, NF, K, JOB, IERR)
NEXTLG(N)
NEXTSM(N)
LAG(X, NX, L)
PGRMC(X, Y, PGR, PGI, N, IERR)
PGRMR(X, PG, N, IND, IERR)
SMTHC(XR, XI, HR, HI, N, WK, JOB, IERR)
SMTHR(X, H, N, WK, JOB, IERR)
SPEC1(X, PG, W, N, BW, NBW, F, NF, DF, WK, JOB, IERR)
SPEC1U(X, PG, W, N, F, NF, NS, DF, WK, JOB, IERR)
SPEC2(X, PG, W, N, K, BW, F, NF, DF, WK, JOB, IERR)
SPEC2K(X, PG, W, N, K, F, NF, DF, WK, JOB, IERR)
TRNFR2(FXX, FXYR, FXYI, GXY, PHXY, NF)
TRNFRK(F, G, PH, NF, K)
TAPER(X, NX, START, END)
Subroutine Dependencies

Below is a list of those LIBDRB subroutines/functions which call other routines. Italicized names are utility routines/functions not intended for external use and, hence, are not documented.

<table>
<thead>
<tr>
<th>Routine/Function</th>
<th>Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>COHERK</td>
<td>COHER2</td>
</tr>
<tr>
<td>DEMOD</td>
<td>DEMODU</td>
</tr>
<tr>
<td>FFT</td>
<td>AR1DFT OR FFTSNG</td>
</tr>
<tr>
<td>IMPRS2</td>
<td>FFT</td>
</tr>
<tr>
<td>IMPRSK</td>
<td>IMPRS2</td>
</tr>
<tr>
<td>NEXTLG</td>
<td>FFT</td>
</tr>
<tr>
<td>NEXTSM</td>
<td>FFT</td>
</tr>
<tr>
<td>PGRMC</td>
<td>FFT, FX, FY</td>
</tr>
<tr>
<td>PGRMR</td>
<td>FFT, MKSYM</td>
</tr>
<tr>
<td>SMTHC</td>
<td>FFT</td>
</tr>
<tr>
<td>SMTHR</td>
<td>FFT, MKSYM</td>
</tr>
<tr>
<td>SPEC1</td>
<td>PGRMR, SPEC1U, WGHTS</td>
</tr>
<tr>
<td>SPEC1U</td>
<td>DECIM, PGRMR, SMTHR</td>
</tr>
<tr>
<td>SPECK</td>
<td>SPECKU, WGHTS</td>
</tr>
<tr>
<td>SPECKU</td>
<td>DECIM, PGRMC, PGRMR, SMTHC</td>
</tr>
<tr>
<td>TRNFRK</td>
<td>TRNFR2</td>
</tr>
</tbody>
</table>
SUBROUTINE AR1DFT(X, Y, N, IERR)

Arbitrary radix one dimensional fast Fourier transform of a one-dimensional complex array.
WARNING: This routine destroys the existing X and Y values.

X A one-dimensional array containing the real part of the sequence. On return, contains
the real part of the Fourier transform.

Y A one-dimensional array containing the imaginary part of the sequence. On return, con-
tains the imaginary part of the Fourier transform.

N The number of points in the sequence. N should be reasonably composite.

IERR A non-zero value on return indicates that an error has occurred while factoring N. The
interpretation of the values is as follows:

IERR = 0: no error.
IERR = 1: N < 2.
IERR = 2: too many factors in N
(current maximum is 16).
IERR = 3: factor of N too large
(current maximum is 19).

See NEXTLG (NEXTSM) to find the next largest (smallest) acceptable N.

The discrete Fourier transform

\[ \sum_{i=0}^{N-1} \frac{2\pi i}{N} \ast (X_s + iY_s) \]

is returned. Refer to the source listing for further details.

SUBROUTINE AUTREG(CXX, A, NA, V, AIC, NX)

Fits autoregressive schemes of order 0, 1, ..., NA-1 to the series with autocovariance function
CXX, by recursively solving the Yule-Walker equations.

CXX An array of the order 0 through NA-1 autocovariances of the series X, e.g., as produced
by CRSCOV. The lag L value should be stored in CXX(L+1).

A An NA by NA array. On return, the i-th column contains the coefficients produced by
the least squares fit of the model

A(1)\ast X(t) + ... + A(i)\ast X(t-i+1) = e(t)

with A(1)=1 and the e(t) independently distributed with mean 0. The array is filled out
with zeros.

NOTE: the diagonal of A contains the negatives of the order 0 through NA-1 partial
autocorrelations of the series X (except that A(1,1) contains 1, the order 0 partial
autocorrelation, rather than -1).

NA  A positive integer. NA-1 is the highest order model fitted.
V   An array of NA values. On return, contains one residual variance for each model fitted.
AIC An array of NA values. On return, contains one Akaike information statistic for each model fitted.
NX  A positive integer, the number of data points from which CXX was originally computed.

SUBROUTINE COHER2(FXX, FYY, FXYR, FXYI, R, NF)
Computes the coherence of two series X and Y from their estimated (power- and cross-) spectra. The spectra can be obtained using SPECK or SPECKU.

FXX  An array of NF elements, the estimated power spectrum of the first series.
FYY  An array of NF elements, the estimated power spectrum of the second series.
FXYR An array of NF elements, the real part of the estimated cross spectrum of the two series.
FXYI An array of NF elements, the imaginary part of the estimated cross spectrum of the two series.
R    An array of NF elements. On return, holds the estimated coherence of the two series.
NF   A positive integer, the number of frequencies at which the spectra were estimated.

SUBROUTINE COHERK(F, R, NF, K)
Computes all pairwise coherences of a K-variate series X from the spectral array produced by SPECK or SPECKU.

F    An NF by K by K array, the spectral array returned by SPECK or SPECKU.
R    An NF by K by K array. On return, holds all pairwise coherences of the components of the series X.
NF   A positive integer, the number of frequencies at which the spectra were estimated.
K    A positive integer (> 1), the dimension of the series X.

SUBROUTINE CRSCOV(X, Y, NXY, CXY, NC)
Estimates the autocovariance function of a single time series or the crosscovariance function of a pair of series. (See AUTREG for the partial autocorrelation function.)

X    An array of NXY time series values.
Y    An array of NXY time series values. Set Y equal to X for a single series.
NXY  A positive integer, the length of the series X and Y.
CXY  An array of NC values. On return, contains the cross-covariance of X with Y at lags 0,...,NC-1. The estimate of cov(X(T+L),Y(T)), the cross-covariance at lag L, is in location CXY(L+1).
NC   A positive integer. The maximum lag at which the cross-covariance is estimated is NC-1.
In the case of two series, for a full set of second-order estimates, one must call CRSCOV(X,X,...), CRSCOV(X,Y,...), CRSCOV(Y,X,...) and CRSCOV(Y,Y,...).

**SUBROUTINE DEMEAN(X, NX, XMEAN)**

Removes the sample mean of the series X. **WARNING:** this routine destroys the existing X values.

X An array of NX time series values. On return, holds the transformed series.
NX A positive integer, the length of the series X.
XMEAN The mean removed.

**SUBROUTINE DEMOD(X, Y, NXY, FREQ, W, L, AMP, PH, NAP, XPOW, BW)**

Calculates the demodulates at frequency FREQ of the complex-valued series X + iY. **WARNING:** This routine destroys the existing X and Y values.

X An array of the real parts of a sequence of NXY time series values.
Y An array of the imaginary parts of the NXY time series values.
NXY A positive integer, the length of the series X and Y.
FREQ The frequency of demodulation, in cycles per unit time.
W An array of L+1 elements, used to hold the weights of the smoothing filter. The weights are calculated by the routine. To specify the weights to be used, see DEMODU.
L A positive integer. Demodulation is carried out by means of a raised cosine window of length 2*L+1. The first and last N values in AMP and PH are set to 0, where N is the integer part of 1 + L*(NAP-1)/NXY.
AMP An array of NAP values. On return, contains the estimated instantaneous log10 amplitude at NAP equispaced time points covering the entire time span of the input series.
PH An array of NAP values. On return, contains the estimated instantaneous phase at NAP equispaced time points covering the entire time span of the input series.
NAP A positive integer, the number of (equispaced) points at which the instantaneous amplitude and phase are computed.
XPOW On return, contains a number which, when added to the instantaneous log10 amplitude, gives an estimate of one half times the instantaneous log10 power.
BW On return, contains the bandwidth of the smoothing filter (equal to 1/(L + .5)).

**SUBROUTINE DEMODU(X, Y, NXY, FREQ, W, L, AMP, PH, NAP, XPOW, BW)**

Calculates the demodulates at frequency FREQ of the complex-valued series X + iY using a user-specified smoothing filter. **WARNING:** This routine destroys the existing X and Y values.

X An array of the real parts of a sequence of NXY time series values.
Y An array of the imaginary parts of the NXY time series values.
NXY A positive integer, the length of the series X and Y.
FREQ The frequency of demodulation, in cycles per unit time.
W An array of L+1 elements containing the weights of the smoothing filter. W(i) should hold the lag i-1 weight (= the lag 1-i weight). The weights should all be positive.
L  A positive integer. The symmetric smoothing filter has \(2L + 1\) non-zero coefficients. The first and last \(N\) values in AMP and PH are set to 0, where \(N\) is the integer part of \(1 + L*(NAP-1)/NXY\).

AMP  An array of NAP values. On return, contains the estimated instantaneous log10 amplitude at NAP equispaced time points covering the entire time span of the input series.

PH  An array of NAP values. On return, contains the estimated instantaneous phase at NAP equispaced time points covering the entire time span of the input series.

NAP  A positive integer, the number of (equispaced) points at which the instantaneous amplitude and phase are computed.

XPOW  On return, contains a number which, when added to the instantaneous log10 amplitude, gives an estimate of one half times the instantaneous log10 power.

BW  On return, contains the bandwidth of the smoothing filter (equal to \(1/(L + .5)\)).

SUBROUTINE DESEAS(X, NX, SM, NM)

Simple seasonal adjustment by removing the seasonal means of period NM. WARNING: this routine destroys the existing X values.

X  An array of NX time series values. On return, holds the transformed series.

NX  A positive integer, the length of the series X.

SM  The array of the NM seasonal means removed.

NM  A positive integer, the number of seasonal means.

SUBROUTINE DETRND(X, NX, B)

Removes a linear trend from the series X. WARNING: this routine destroys the existing X values.

X  An array of NX time series values. On return, holds the transformed series.

NX  A positive integer, the length of the series X.

B  A vector containing the two parameters of the fitted line:

\[ X(t) = B(1) + B(2)*(t-tmean) \]

SUBROUTINE DIFF(X, NX, D, ND)

Differences the time series X. WARNING: this routine destroys the existing X values.

X  An array of NX time series values. On return, X contains the differenced series, padded with zeros at the start or end as appropriate (see below).

NX  A positive integer, the length of the series X.

D  A vector of ND integers, the orders of the successive differences to be performed. At stage \(j\), \(X(i)\) (as modified by previous differences) is replaced by \(X(i) - X(i-D(j))\). The series is padded with zeros. The number of zeros added at the start of the series is the sum of the positive elements of D; the number added at the end is equal to the absolute value of the sum of the negative elements of D. (Thus, the sum of these two numbers must be less than NX.)
ND A positive integer, the number of differences to be performed.

NOTE:
If both differencing and lagging (see LAG) are to be performed, the former should be done first.

SUBROUTINE FFT(X, Y, N, INV, IERR)
Generic one-dimensional fast Fourier transform. Calls one of AR1DFT or FFTSNG.

X A one-dimensional array containing the real parts of the data. On return, contains the real part of the Fourier transform.
Y A one-dimensional array containing the imaginary parts of the data. On return, contains the imaginary part of the Fourier transform.
N The total number of complex data values. N should be reasonably composite.
INV An indicator specifying whether or not to compute the inverse transform. If INV = 0, the sign of the complex exponential in the transform equation is -1. If INV = 1, the sign is 1 and the result is divided by N.
IERR A non-zero value on return indicates that an error has occurred while factoring N. The interpretation of the values is as follows:

IERR = 0: no error.
IERR = 1: N < 2.
IERR = 2: too many factors in N
(maximum is routine-dependent).
IERR = 3: factor of N too large
(maximum is routine-dependent).
IERR = 4: product of non-square factors too large
(maximum is routine-dependent).

See NEXTLG (NEXTSM) to find the next largest (smallest) acceptable N.

SUBROUTINE FFTSNG(X, Y, NTOT, N, NSPAN, ISN, IERR)
Multivariate (i.e. spatial) complex Fourier transform, computed in place using a mixed radix fast Fourier transform algorithm.

X An array containing the real parts of the data. On return, contains the real part of the Fourier transform.
Y An array containing the imaginary parts of the data. On return, contains the imaginary part of the Fourier transform.
NTOT The total number of complex data values.
N The dimension of the current variable.
NSPAN The value of NSPAN/N gives the spacing between values of the current variable.
ISN An variable giving the sign of the complex exponential in the transform equation (+1 or -1).
IERR A non-zero value on return indicates that an error has occurred while factoring N. The interpretation of the values is as follows:
IERR = 0: no error.
IERR = 1: N < 2.
IERR = 2: too many factors in N
          (current maximum is 11).
IERR = 3: factor of N too large
          (current maximum is 23).
IERR = 4: product of non-square factors too large
          (current maximum is 209).

See NEXTLG (NEXTSM) to find the next largest (smallest) acceptable N.

See the listing for a more detailed description.

SUBROUTINE FILTER(X, NX, F, NF, Y)
    Applies the filter with coefficients in F to the series in X, putting the result in Y.

    X       An array of NX time series values.
    NX      A positive integer, the length of the series X.
    F       An array of NF filter coefficients.
    NF      A positive integer, the number of coefficients in filter F.
    Y       An array of NX values. On return, contains the filtered series given by
             \[ Y_t = \sum_{i=1}^{NF} F_i \cdot X_{t+i-NF} \]
             for t=1,...,NX-NF+1; Y_t=0 for t=NX-NF+2,...,NX.

SUBROUTINE FREQRM(X, NX, ANG, Y)
    Removes a component of frequency ANG (in radians per unit time) from the series X and
    puts the result in Y.

    X       An array of NX time series values.
    NX      A positive integer, the length of the series X.
    ANG     The frequency of the component to be removed, in radians per unit time.
    Y       An array of NX elements. On return, holds the series with the specified component removed.

SUBROUTINE IMPRS2(GXY, PHXY, AR, AI, NF, JOB, IERR)
    Computes the impulse response function of Y on X from the corresponding transfer function,
    e.g., as produced by TRNFR2.

    GXY      An array of NF elements, the gain of Y on X.
    PHXY     An array of NF elements, the phase of Y on X.
AR  An array of NF elements. On return, holds the real part of the estimated impulse response function.

AI  An array of NF elements. On return, holds the imaginary part of the estimated impulse response function.

NF  A positive integer, the number of frequencies at which the gains/phases were estimated.

JOB  An indicator specifying whether or not to use convergence factors in computing the impulse response. If JOB = 1, Tukey-Hamming factors are used.

IERR  A non-zero value on return indicates that NF has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.)

**SUBROUTINE IMPRSK(G, PH, AR, AI, NF, K, JOB, IERR)**

Computes all pairwise impulse response functions of the K-variate series X from the corresponding transfer function arrays produced by TRNFRK.

G  The NF by K by K array of gains produced by TRNFRK.

PH  The NF by K by K array of phases produced by TRNFRK.

AR  An NF by K by K array. On return, holds the real parts of all the pairwise impulse response functions of the components of X on each other. AR(1:NF,i,j) contains the real part of the impulse response function of the j-th component of X on the i-th component.

AI  An NF by K by K array. On return, holds the imaginary parts of all the pairwise impulse response functions of the components of X on each other. AI(1:NF,i,j) contains the imaginary part of the impulse response function of the j-th component of X on the i-th component.

NF  A positive integer, the number of frequencies at which the gains/phases were estimated.

K  A positive integer (> 1), the dimension of the series X.

JOB  An indicator specifying whether or not to use convergence factors in computing the impulse response. If JOB = 1, Tukey-Hamming factors are used.

IERR  A non-zero value on return indicates that NF has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.)

**SUBROUTINE LAG(X, NX, L)**

Shifts the time series X by L time units. WARNING: this routine destroys the existing X values.

X  An array of NX time series values. On return, X(i) is replaced by X(i-L) and the first (if L > 0) or last (if L < 0) L places contain zeros.

NX  A positive integer, the length of the series X.

L  An integer, the amount by which the series X is to be lagged. The absolute value of L must be less than NX.

NOTE:

If both differencing (see DIFF) and lagging are to be performed, the former should be done first.
INTEGER FUNCTION NEXTLG(N)

Returns the smallest positive integer greater than or equal to N which satisfies the FFT factorization criteria. (0 is returned if no such integer exists (i.e., if N > the largest acceptable series length).)

INTEGER FUNCTION NEXTSM(N)

Returns the largest positive integer less than or equal to N which satisfies the FFT factorization criteria. (0 is returned if no such integer exists (i.e., if N ≤ 1).)

SUBROUTINE PGRMC(X, Y, PGR, PGI, N, IERR)

Computes the cross periodogram of X and Y.

X An array of N time series values.
Y An array of N time series values.
PGR An array of N elements. On return, holds the real part of the cross periodogram.
PGI An array of N elements. On return, holds the imaginary part of the cross periodogram.
N A positive integer, the length of the series X and Y. N should be reasonably composite.
IERR A non-zero value on return indicates that N has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.)

SUBROUTINE PGRMR(X, PG, N, IND, IERR)

Computes the periodogram of the series X. WARNING: This routine destroys the existing X (and PG) values.

X An array of N time series values.
PG An array of N elements. On return, holds the periodogram.
N A positive integer, the length of the series X. N should be reasonably composite.
IND An indicator describing the contents of X and PG when PGRMR is called. Usually, X will contain the time series for which the periodogram is desired. In this case, IND should be 0 and any values in PG will be ignored. If the Fourier transform of X has been previously computed, however, this may be passed instead. In this case, X should contain the real part of the Fourier transform, PG the imaginary part, and IND should be 1. In either case, the desired periodogram is returned in PG.
IERR A non-zero value on return indicates that N has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.)

SUBROUTINE SMTHC(XR, XI, HR, HI, N, WK, JOB, IERR)

Smooths the complex valued series X with the coefficients specified by H. This is a fast smooth using FFT. The result will be exact if X is a periodic series with period equal to its length (e.g., a periodogram). Otherwise, the values at the ends of the smoothed series will be incorrect. WARNING: this routine destroys the existing X and H values.
XR  An array of N elements, the real part of the series to be smoothed. On return, holds the real part of the smoothed series

\[ \text{smooth}(X_t) = \sum_{i=1}^{N} H_s \cdot X_{t-s+1} \]

where \( t-s+1 \) is interpreted modulo N (in the sense that it is replaced by \( t-s+1-N \) if it is \( N \) and by \( t-s+1+N \) if it is \( -N \)). The multiplication here is true complex multiplication, as opposed to component-wise multiplication.

XI  An array of N elements, the imaginary part of the series to be smoothed. On return, holds the imaginary part of the smoothed series.

HR  An array of N elements, the coefficients of the real part of the smoothing filter. The coefficients should be stored as described in SMTHR.

HI  An array of N elements, the coefficients of the imaginary part of the smoothing filter. The coefficients should be stored as described in SMTHR.

N   A positive integer, the length of the series X. N should be reasonably composite.

WK  An array of 2*N elements, used as workspace by the routine. On return, WK holds the Fourier transform of either the filter or the series. This may be used to more efficiently smooth several series with the same filter or one series with several filters (see JOB, below).

JOB A positive integer of the decimal form AB which provides instructions for the routine. The value of A should be 1 if WK already contains the Fourier transform of either the series or the filter and 0 otherwise. B specifies which Fourier transform is (if A = 1) or is to be (if A = 0) stored in WK: 1 for the transformed filter, 0 for the transformed series. In other words,

\[
\text{JOB} = 0: \text{nothing in WK; store transformed series there} \\
\text{JOB} = 1: \text{nothing in WK; store transformed filter there} \\
\text{JOB} = 10: \text{transformed series in WK} \\
\text{JOB} = 11: \text{transformed filter in WK}
\]

IERR A non-zero value on return indicates that N has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.)

SUBROUTINE SMTHR(X, H, N, WK, JOB, IERR)

Smoothes the real valued series X with the coefficients in H. This is a fast smooth using FFT. The result will be exact if X is a periodic series with period equal to its length (e.g., a periodogram). Otherwise, the values at the ends of the smoothed series will be incorrect. WARNING: this routine destroys the existing X and H values.

X   An array of N elements, the series to be smoothed. On return, holds the smoothed series

\[ \text{smooth}(X_t) = \sum_{i=1}^{N} H_s \cdot X_{t-s+1} \]

where \( t-s+1 \) is interpreted modulo N (in the sense that it is replaced by \( t-s+1-N \) if it is \( N \) and by \( t-s+1+N \) if it is \( -N \)).
**An array of N elements, the coefficients of the smoothing filter. Because of the periodicity of the series, the coefficients should be stored as follows. Suppose the smoothed value at time \( t \) is a weighted sum of the values at times \( t-p, \ldots, t+q \). Then the lag \( i \) weight (i.e., the coefficient of \( X(t-i) \)) should be stored in \( H(i+1) \) for \( i=0, \ldots, p \) and in \( H(N+1+i) \) for \( i=-1, \ldots, -q \). The remainder of \( H \) (if any) should be filled out with zeros.**

**A positive integer, the length of the series \( X \). \( N \) should be reasonably composite.**

**An array of 2*N elements, used as workspace by the routine. On return, \( WK \) holds the Fourier transform of either the filter or the series. This may be used to more efficiently smooth several series with the same filter or one series with several filters (see \( JOB \), below).**

**A positive integer of the decimal form \( AB \) which provides instructions for the routine. The value of \( A \) should be 1 if \( WK \) already contains the Fourier transform of either the series or the filter and 0 otherwise. \( B \) specifies which Fourier transform is (if \( A = 1 \)) or is to be (if \( A = 0 \)) stored in \( WK \): 1 for the transformed filter, 0 for the transformed series. See \( SITHC \) for a more complete explanation.**

**A non-zero value on return indicates that \( N \) has failed to meet the \( FFT \) factorization criteria. (See \( FFT \) for the interpretation of the value, \( NEXTLG \) to find the next largest acceptable \( N \), and \( NEXTSM \) to find the next smallest such \( N \).)**

**Estimates the power spectrum of the series \( X \). WARNING: this routine destroys the existing \( X \) values.**

**An array of \( N \) time series values.**

**An array of \( N \) elements. On return, contains the periodogram of \( X \).**

**An array of \( N \) elements; used to hold the coefficients of the smoothing filter. The coefficients are calculated by the routine. To specify the coefficients to be used, see \( SPECKU \).**

**A positive integer, the length of the series \( X \). \( N \) should be reasonably composite.**

**An array of \( NBW \) elements. \( BW(i) \) gives the bandwidth, in cycles per unit time, desired for the \( i \)-th spectral estimate. The \( i \)-th smoothing filter has \( 2L + 1 \) non-zero coefficients, where \( L \) is the integer part of \((N\cdot BW(i) - 1)/2\). All bandwidths should be in the interval \([3/N, (N-1)/N]\). If a bandwidth lies outside this interval, it is set to the nearest acceptable value.**

**A positive integer, the number of different bandwidths of interest. If \( NBW \) is less than 1, only the periodogram is computed. (This can be done more directly using \( PGRMR \).)**

**An NF by \( NBW \) array. On return, the \( i \)-th column contains the \( i \)-th spectral estimate at NF equispaced frequencies between 0 and .5 cycles per unit time.**

**A positive integer, the number of equispaced points at which the spectrum is to be estimated. \( NF \) must be \( \leq n2 \), where \( n2 \) is the integer part of \( N/2 + 1 \).**

**An array of \( NBW \) elements. On return, \( DF(i) \) holds the (number of) degrees of freedom of the chi-square approximation to the distribution of the \( i \)-th spectral estimate.**

**An array of 2*N elements, used as workspace by the routine. On return, \( WK \) contains the periodogram as transformed by the smoothing routine. This may be used to obtain additional spectral estimates more efficiently (see \( JOB \), below).**

**A positive integer of the decimal form \( CBA \) which provides instructions for the routine. **

A specifies the shape of the smoothing filter used:
A = 0 : rectangular
A = 1 : cosine
A = 2 : triangular

B describes the input to the routine:
B = 0 : X contains the series for which spectral estimates are desired.
B = 1 : X contains the real part of the Fourier transform of the series, PG contains the imaginary part.
B = 2 : PG contains the periodogram of the series.
B = 3 : WK contains the periodogram of the series as transformed by a previous call to SPEC1 or SPEC1U.

Ordinarily, \( \log_{10} \) of the power spectrum is returned in F. If C = 1, the spectrum itself is returned.

IERR A non-zero value on return indicates that N has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.) Here IERR will also be 1 on return if NF \( \leq 1 \) or NF > n2.

SUBROUTINE SPEC1U(X, PG, W, N, F, NF, NS, DF, WK, JOB, IERR)

Estimates the power spectrum of the series X using user-specified weights to smooth the periodogram. WARNING: this routine destroys the existing X values.

X An array of N time series values.
PG An array of N elements. On return, contains the periodogram of X.
W An N by NS array. The i-th column contains the weights to be used for the i-th spectral estimate. The value of this estimate at the j-th Fourier frequency is the weighted average

\[
\sum_{k=-n2}^{j} W_{2+k,i} \cdot PG_{j+k}
\]

where n2 is the integer part of \( N/2 + 1 \) and j+k is interpreted modulo N (in the sense that j+k is replaced by j+k-N if j+k > N and by j+k+N if it is < 1). If there are less than N non-zero weights in the i-th filter, that column should be filled out with 0's. (The weights should be non-negative and sum to 1.)

N A positive integer, the length of the series X. N should be reasonably composite.
F An NF by NS array. On return, the i-th column contains the i-th spectral estimate at NF equispaced frequencies between 0 and .5 cycles per unit time.
NF A positive integer, the number of equispaced points at which the spectrum is to be estimated. NF must be \( \leq n2 \) (see above).
NS A positive integer, the number of spectral estimates desired. If NS is less than 1, only the periodogram is computed. (This can be done more directly using PGRMR.)
DF An array of NS elements. On return, DF(i) holds the (number of) degrees of freedom of the chi-square approximation to the distribution of the i-th spectral estimate.
WK An array of 2*N elements, used as workspace by the routine. On return, WK contains the periodogram as transformed by the smoothing routine. This may be used to obtain
additional spectral estimates more efficiently (see JOB, below).

**JOB**

A positive integer of the decimal form BA which provides instructions for the routine.

A describes the input to the routine:

- A = 0: X contains the series for which spectral estimates are desired.
- A = 1: X contains the real part of the Fourier transform of the series, PG contains the imaginary part.
- A = 2: PG contains the periodogram of the series.
- A = 3: WK contains the periodogram of the series as transformed by a previous call to **SPEC1** or **SPEC1U**.

Ordinarily, log10 of the power spectrum is returned in F. If B = 1, the spectrum itself is returned.

**IERR**

A non-zero value on return indicates that N has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.) Here IERR will also be 1 on return if NF \< 1 or NF > n2.

**SUBROUTINE SPECK(X, PG, W, N, K, BW, F, NF, DF, WK, JOB, IERR)**

Estimates all power- and cross- spectra of the K-variate series X. WARNING: this routine destroys the existing X values.

**X**

An N by K array of time series values.

**PG**

An N by K by K array. On return, contains the periodograms and cross-periodograms of X in the following form. The elements PG(1:N,j,i) contain the periodogram of the i-th component of X. The elements PG(1:N,j,i) contain, for i \< j, the real part of the cross-periodogram of the i-th and j-th components of X, while the elements PG(1:N,j,i) contain the imaginary part of this cross-periodogram.

**W**

An array of 2*N elements, used to hold the coefficients of the smoothing filter. The coefficients are calculated by the routine. To specify the coefficients to be used, see **SPECKU**.

**N**

A positive integer, the length of the series X. N should be reasonably composite.

**K**

A positive integer (> 1), the dimension of the series X.

**BW**

The bandwidth of the smoothing filter, in cycles per unit time. The smoothing filter has 2*L + 1 non-zero coefficients, where L is the integer part of (N*BW - 1)/2. BW should be in the interval [3/N,(N-1)/N]. If it is not, it is set to the nearest acceptable value.

**F**

An NF by K by K array. On return, contains the estimates of the power- and cross-spectra of X, at NF equispaced frequencies between 0 and .5 cycles per unit time, in the following form. The elements F(1:NF,j,i) contain the power spectrum of the i-th component of X. The elements F(1:NF,j,i) contain, for i \< j, the real part of the cross-spectrum of the i-th and j-th components of X, while the elements F(1:NF,j,i) contain the imaginary part of this cross-spectrum.

**NF**

A positive integer, the number of equispaced points at which the spectra are to be estimated. NF must be \< n2, where n2 is the integer part of N/2 + 1.

**DF**

On return, DF holds the (number of) degrees of freedom of the complex Wishart approximation to the distribution of the spectral estimate.

**WK**

An array of 2*N elements, used as workspace by the routine.
JOB     An integer of the decimal form BA which provides instructions for the routine.

A specifies the shape of the smoothing filter used:
   A = 0 : rectangular
   A = 1 : cosine
   A = 2 : triangular

B describes the input to the routine:
   B = 0 : X contains the series for which spectral estimates are desired.
   B = 1 : PG contains the periodogram of the series.

In addition, if JOB = -1, only the periodogram is computed.

IERR     A non-zero value on return indicates that N has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable N, and NEXTSM to find the next smallest such N.) Here IERR will also be 1 on return if NF ≤ 1 or NF > n2, or if K < 2.

SUBROUTINE SPECKU(X, PG, W, N, K, F, NF, DF, WK, JOB, IERR)

Estimates all power- and cross-spectra of the K-variate series X using user-specified weights to smooth the periodograms. WARNING: this routine destroys the existing X values.

X     An N by K array of time series values.

PG     An N by K by K array. On return, contains the periodograms and cross-periodograms of X in the following form. The elements PG(1:N,i,i) contain the periodogram of the i-th component of X. The elements PG(1:N,i,j) contain, for i < j, the real part of the cross-periodogram of the i-th and j-th components of X, while the elements PG(1:N,j,i) contain the imaginary part of this cross-periodogram.

W     An array of 2*N elements. The first N elements are the coefficients of the smoothing filter. These should be non-negative and sum to 1. (If there are less than N non-zero coefficients, the remainder of these (first N) elements should be set to 0.) The remaining N elements are used by the smoothing routine.

The estimate at the l-th Fourier frequency of the j,k-th (j < k) cross-spectrum is the weighted average

\[ \sum_{m=-n2}^{n2} W_{n2+m} \ast (PG_{l+m,j,k} + iPGL_{l+m,k,j}) \]

where n2 is the integer part of N/2 + 1 and l+m is interpreted modulo N (in the sense that l+m is replaced by l+m-N if l+m > N and by l+m+N if it is < 1). (For the j-th power spectrum, the m-th term in the above sum is \( W_{n2+m} \ast PG_{l+m,j,j} \).)

N     A positive integer, the length of the series X. N should be reasonably composite.

K     A positive integer (> 1), the dimension of the series X.

F     An NF by K by K array. On return, contains the estimates of the power- and cross-spectra of X, at NF equispaced frequencies between 0 and .5 cycles per unit time, in the following form. The elements F(1:MF,i) contain the power spectrum of the i-th component of X. The elements F(1:MF,i,j) contain, for i < j, the real part of the cross-spectrum of the i-th and j-th components of X, while the elements F(1:MF,j,i) contain
the imaginary part of this cross-spectrum.

NF  A positive integer, the number of equispaced points at which the spectra are to be estimated. NF must be \( \leq n^2 \) (see above).

DF  On return, DF holds the (number of) degrees of freedom of the complex Wishart approximation to the distribution of the spectral estimate.

WK  An array of \( 2 \times N \) elements, used as workspace by the routine.

JOB  An indicator which describes the input to the routine.

\( \text{JOB} = 0 : X \) contains the series for which spectral estimates are desired.

\( \text{JOB} = 1 : PG \) contains the periodograms of the series.

IERR  A non-zero value on return indicates that \( N \) has failed to meet the FFT factorization criteria. (See FFT for the interpretation of the value, NEXTLG to find the next largest acceptable \( N \), and NEXTSM to find the next smallest such \( N \).) Here IERR will also be 1 on return if \( NF \leq 1 \) or \( NF > n^2 \), or if \( K < 2 \).

SUBROUTINE TRNFR2(FXX, FXYR, FXYI, GXY, PHXY, NF)
Computes the gain and phase of \( Y \) on \( X \) from the power spectrum of \( X \) and the cross spectrum of the two series. The spectra can be obtained using SPECK or SPECKU.

FXX  An array of \( NF \) elements, the estimated power spectrum of the first series.

FXYR  An array of \( NF \) elements, the real part of the estimated cross spectrum of the two series.

FXYI  An array of \( NF \) elements, the imaginary part of the estimated cross spectrum of the two series.

GXY  An array of \( NF \) elements. On return, holds the gain of \( Y \) on \( X \).

PHXY  An array of \( NF \) elements. On return, holds the phase of \( Y \) on \( X \).

NF  A positive integer, the number of frequencies at which the spectra were estimated.

SUBROUTINE TRNFRK(F, G, PH, NF, K)
Computes all pairwise gains/ phases of the \( K \)-variate series \( X \) from the spectral array produced by SPECK or SPECKU.

F  An \( NF \) by \( K \) by \( K \) array, the spectral array returned by SPECK or SPECKU.

G  An \( NF \) by \( K \) by \( K \) array. On return, holds all pairwise gains of the components of \( X \) on each other. \( G(1:NF,i,j) \) contains the gain of the \( j \)-th component of \( X \) on the \( i \)-th component.

PH  An \( NF \) by \( K \) by \( K \) array. On return, holds all pairwise phases of the components of \( X \) on each other. \( PH(1:NF,i,j) \) contains the phase of the \( j \)-th component of \( X \) on the \( i \)-th component.

NF  A positive integer, the number of frequencies at which the spectra were estimated.

K  A positive integer (\( > 1 \)), the dimension of the series \( X \).

SUBROUTINE TAPER(X, NX, START, END)
Tapers the ends of a time series with raised cosine tapers. WARNING: this routine destroys the existing \( X \) values.
X An array of NX time series values. On return, holds the transformed series.
NX A positive integer, the length of series X.
START The first 100*START percent of the array is to be tapered.
END The last 100*END percent of the array is to be tapered.


17. DRAPER, D. C. (Feb 1983). Rank-based robust inference in regression models with several observations per cell.


32. BREIMAN, L. (May 1984). Nail finders, edifices, and Oz.


43. BICKEL, P. J. and YAHAV, J. A. (1985). On estimating the number of unseen species: how many executions were there?


56. LE CAM, L. and YANG, G. L. (January 1986). Replaced by No. 68.


64. O'SULLIVAN, F. (May 1986). Nonparametric Estimation in the Cox Proportional Hazards Model.


70. BICKEL, P.J. and YAHAV, J.A. (July 1986). Richardson Extrapolation and the Bootstrap.


76. O'SULLIVAN, F. (September 1986). Deconvolution of episodic hormone data.


80. DABROWSKA, D.M. (November 1986). Nonparametric regression with censored survival time data.


90. DABROWSKA, D.M. (March 1987). Rank tests for matched pair experiments with censored data.


95. CANCELLED


114. RITOV, Y. (September 1987). Estimation in a linear regression model with censored data.
118. ALDOUS, D.J. (October 1987). Meeting times for independent Markov chains.
127. Same as No. 133.
128. DONOHO, D. and GASKO, M. (December 1987). Multivariate generalizations of the median and trimmed mean II.
131. Same as No. 140
166. FAN, JIANQING (July 1988). Nonparametric estimation of quadratic functionals in Gaussian white noise.
172. ADLER, R.J. and EPSTEIN, R. (September 1988). Intersection local times for infinite systems of planar brownian motions and for the brownian density process.


Copies of these Reports plus the most recent additions to the Technical Report series are available from the Statistics Department technical typist in room 379 Evans Hall or may be requested by mail from:

Department of Statistics
University of California
Berkeley, California 94720

Cost: $1 per copy.