AUSTRALIAN NATIONAL ANTARCTIC RESEARCH EXPEDITIONS

A N A R E

R E S E A R C H

N O T E S

25

An impulse function program

J.S. Reid and G.B. Burns

ANTARCTIC DIVISION
DEPARTMENT OF SCIENCE

ANARE RESEARCH NOTES (ISSN 0729-6533)

This series allows rapid publication in a wide range of disciplines. Copies of this and other ANARE Research Notes are available from the Antarctic Division. Any person who has participated in Australian National Antarctic Research Expeditions is invited to publish through this series. Before submitting manuscripts authors should obtain a style guide from:

The Publications Office
Antarctic Division
Channel Highway
Kingston
Tasmania 7150
Australia.

Published March 1985
ISBN: 0 642 07453 4

CONTENTS

FIGURES

APPENDIXES

AN IMPULSE FUNCTION PROGRAM

by

J.S. Reid
Reid Research,
21 Service Street,
Glebe, Tasmania, Australia

and

G.B. Burns
Antarctic Division,
Department of Science,
Hobart, Tasmania, Australia

ABSTRACT

This Research Note details the mathematics behind the development of a program to calculate the impulse function relating two arrays of data. These data arrays are discrete samples of the proposed driving and driven functions. The program is tested for its susceptibility to noise and non-linear variations between the input functions. The impulse function program and relevant test programs are listed and their use described.

1. MATHEMATICAL DEVELOPMENT

The source of the information contained in this section is predominantly communications with Dr J.S. Reid, with supplementary detail provided by Jenkins and Watts (1968) and Breiman (1973). The mathematical ideas and the initial program are the work of Dr J.S. Reid.

Impulse response estimation is a method of time series analysis in which two time series are assumed to be linearly related. One time series, the driven function $\{y_i, i = 1, 2, \ldots, N\}$, is assumed to result from the application of a linear filter to the other time series, the driving function $\{x_i, i = 1, 1, \ldots, N\}$. The object is to estimate, from the time series', the impulse response, h, which describes the filter.

Two functions are linearly related when

$$y(t) = \int_{-\infty}^{\infty} h(u)x(t-u)du + y_0 \qquad (1)$$

where y_0 is a constant.

It is assumed that

(i) the two functions are causally related by a physically realizable filter. This implies that

$$h(u) = 0 \quad \text{when } u < 0 \qquad (2)$$

and $$\int_0^{\infty} h(u)du < K, \text{ where K is some finite value.} \qquad (3)$$

(ii) The given time series' $\{x_i\}$, $\{y_i\}$ are obtained by sampling continuous functions at discrete intervals of time.

$$x_i = x\,(i \cdot \Delta t)$$

(iii) There is a random element, Z_i, introduced by the observing process, or inherent in the physical process under examination, such that

$$Y_i = \sum_{r=0}^{\infty} h_r x_{i-r} + y_0 + Z_i \tag{4}$$

Y_i and Z_i are infinite series for each i. The data set $\{y_i\}$ is assumed to be the realization of an infinite set of random variables $\{Y_i\}$ given by equation (4).

The continuous impulse response, h(u), and the discrete impulse response, h_r, are related by equation (5).

$$h_r = \int_{(r-\frac{1}{2})\Delta t}^{(r+\frac{1}{2})\Delta t} h(u)du \tag{5}$$

(iv) The random variables, Z_i, are assumed to have the following properties:

(a) they are Gaussian

(b) they have zero mean

$$E(Z_i) = 0 \quad \text{for } i = 1,2,...N \tag{6}$$

(c) they have constant variance

$$E(Z_i^2) = \sigma^2 \quad \text{for } i = 1,2,...N \tag{7}$$

and (d) they are un-selfcorrelated or 'white'

$$E(Z_i Z_j) = 0 \quad \text{for } i \neq j \tag{8}$$

(v) In order to be consistent and physically realistic, a random component should be included in the input series $\{x_i\}$ as well. This however makes the problem insoluble. The assumption is made that any random noise present in the observation of the driving function is much smaller than $E(Z^2)$ and can be neglected.

(vi) A final assumption is that h_r is negligibly small for r greater than some number m, which is itself much smaller than the number of data in the set, N. With these assumptions, $\{y_i\}$ is a realization of equation (4).

$$y_i = \sum_{0}^{m} h_r x_{i-r} + y_0 + z_i. \qquad (9)$$

The series $\{z_i\}$ is the outcome of a white Gaussian process and the values, z_i, are known as residuals.

For notational convenience, but without loss of generality, it is assumed that the series $\{x_i\}$ and $\{y_i\}$ have zero mean. This implies $y_0 = 0$. In the programs contained herein, the mean of each series is subtracted.

For mathematical simplicity, consider only the subset of $\{y_i\}$ for which the x_{i-r} in equation (9) are available. With these simplifications and considerations, equation (9) reduces to equation (10).

$$y_i = \sum_{0}^{m} h_r x_{i-r} + z_i \quad \text{for } i = m+1,\ldots,N. \qquad (10)$$

Equation (10), with the conditions imposed on Z_i, satisfies the conditions of a multivariate regression model in which the h_r are the regression coefficients to be estimated. The solution is found by minimizing the sum of squares of the residuals with respect to the unknown coefficients. When, as assumed, the z_i are Gaussian, the solution is an example of the 'maximum likelihood' method (Breiman, 1973).

$$\text{Let } R = \sum_{i=m+1}^{N} z_i^2$$

$$= \sum_{i=m+1}^{N} (y_i - \sum_{r=0}^{m} h_r x_{i-r})^2 \qquad (11)$$

Hence for $q = 0,1,2,\ldots,m$

$$\frac{\partial R}{\partial h_q} = \sum_{i=m+1}^{N} 2x_{i-q}(y_i - \sum_{r=0}^{m} h_r x_{i-r})^2 \tag{12}$$

The sum of the squares of the residuals, R, is a minimum when

$$\frac{\partial R}{\partial h_q} = 0 \qquad \text{for } q = 0,1,2,\ldots,m \tag{13}$$

For the specific case of R a minimum, let $\{h_r\}$ be represented by $\{\hat{h}_r\}$.

Combining equations (12) and (13) implies

$$\sum_{i=m+1}^{N} x_{i-q}y_i = \sum_{r=0}^{m} \hat{h}_r [\sum_{i=m+1}^{N} x_{i-q}x_{i-r}] \tag{14}$$

$$\text{for } q = 0,1,2,\ldots,m.$$

These m+1 equations for the m+1 unknowns, $\hat{h}_0,\hat{h}_1,\hat{h}_2,\ldots,\hat{h}_m$ can be summarized by the vector equation

$$\underset{\sim}{\Gamma}_{xy} = \Gamma_{xx} \underset{\sim}{\hat{h}} \tag{15}$$

Equation (15) is the sample, vector-equivalent, of the Weiner-Hopf integral equation (Jenkins and Watts, 1968).

In equation (15), $\underset{\sim}{\Gamma}_{xy}$ is an m+1 element column vector whose elements are given by

$$\gamma_p = \sum_{i=m+1}^{N} x_{i-p}y_i \qquad \text{for } p = 0,\ldots,m \tag{16}$$

Γ_{xx} is an (m+1) × (m+1) matrix whose elements are given by

$$\gamma_{pq} = \sum_{i=m+1}^{N} x_{i-p}x_{i-q} \qquad \text{for } p,q = 0,1,\ldots,m \tag{17}$$

and $\underset{\sim}{\hat{h}}$ is an (m+1) element column vector whose elements are the desired impulse response estimates.

If y_0 in equation (9) is not assumed to be zero, and $\{x_i\}$ is not assumed to have zero mean, then equations (16) and (17) become

$$\gamma_p = \sum_{i=m+1}^{N} (x_{i-p}-\bar{x}_p)(y_i-\bar{y}) \tag{18}$$

$$\gamma_{pq} = \sum_{i=m+1}^{N} (x_{i-p}-\bar{x}_p)(x_{i-q}-\bar{x}_q) \tag{19}$$

where

$$\bar{x}_p = \sum_{i=m+1}^{N} x_{i-p}/(N-m-1) \tag{20}$$

$$\bar{y} = \sum_{i=m+1}^{N} y_i/(N-m-1) \tag{21}$$

Equation (14) may be solved directly by matrix inversion methods.

$$\hat{\underset{\sim}{h}} = \Gamma_{xx}^{-1} \underset{\sim}{\Gamma}_{xy} \tag{22}$$

It is this equation which is solved to yield the 'maximum likelihood' elements of the impulse function $\hat{\underset{\sim}{h}}$.

Γ_{xx} is a symmetric, positive-definite matrix and in the programs described here it is inverted using the Cholesky method (p.334, Applied Numerical Methods, Carnahan, Luther and Wilkes, 1969).

2. THE IMPULSE FUNCTION PROGRAM

The impulse function main program, IMPULZ.FOR, and all the associated subroutines are written in FORTRAN. To be made operational the main program must be compiled and linked with compiled versions of the following subroutines and functions; STATS.FOR, FASTNT.FOR, REV.FOR, INVSYM.FOR, FACSYM.FOR, IPP.FOR and PEF.FOR. The main program and the associated subroutines are listed in the Appendix.

The IMPULZ program expects data to be in integer format with a carriage return, line feed separating each record. Provision is made for ignoring file headers, but these must also be in integer format. In 1980 era Antarctic Division data, an integer file header is used to indicate the number of minutes since the start of the UT day at which data collection for the file in question commenced. Subsequent expansion of the information stored in headers has expanded to six the number of integers used to describe the data contained in the file.

As it is presented in this ANARE Research Note, the IMPULZ program has provision for driving (XV) and driven (YV) arrays of up to 3000 points. In order to vary the maximum number of points allowed, it is necessary to alter the dimension of the 'COMMON'ed arrays XV and YV in the main program IMPULZ, and the subroutines STATS and FASTNT.

The IMPULZ program as listed will evaluate impulse functions of up to 100 points. To alter this to a maximum length of N Points, the dimension of the 'DOUBLE PRECISION'ed arrays A, W, B, ALPHA and T must be altered to N and the array S to N (N-1)/2. It is only necessary to alter these values in the main program, IMPULZ.

The order, NU, of the impulse function is the number of points in the function. The delay, NULM, is the number of these points before the zero impulse time. This provides a means of determining the validity of the causality assumption inherent in the impulse analysis method. If the relation between the driving and driven arrays is causal, then all impulse points before the zero impulse time will be approximately zero.

The program asks for the number of data headers in the input files and then for the number of data records which follow the headers. It assumes that these are equal for both driving and driven arrays. These values are held constant throughout a program run. After evaluating the impulse function, the program loops to statement label 50 in the main program. It then requests new input arrays, these may be the same as for the previous run, and then requests input of the order and delay of the new impulse function to be evaluated. In order to exit from the program, it is necessary to hit a 'CONTROL C' or system equivalent.

A second main program, IMPHOT.FOR, is also listed in the Appendix. This must also be compiled and linked with STATS, FASTNT, REV, INVSYM, FACSYM, IPP and PEF. It differs from IMPULZ in that the parameters are set as desired for evaluating the impulse function relating the N_2^+1NG (0,1) 427.8 nm band emission (the driving data) and the O($^1S-^1D$), 557.7 nm line emission (the driven data) from the photometric data collected at Macquarie Island during 1980. The OPEN statements for the input files are more complicated for the IMPHOT version due to the manner in which the data was collected. The photometric data was packed in binary form with one data point per word. This was done in order to save space on the initial data storage medium which was single density, eight inch, floppy disks.

A FORTRAN program for converting data stored in this fashion to the appropriate machine code, BINCON, is also listed in the Appendix.

3. TESTING THE IMPULSE ANALYSIS PROGRAM

The impulse analysis program was tested to determine its susceptibility to noise. A sample of 427.8 nm photometric data was convoluted with a 25 point digital filter with a 7 point decay constant. White noise, amplitude limited to a percentage of the range of the initial 427.8 nm data, was added to this initial data string. The resultant array is used as the driving function. The array resulting from the convolution with the digital filter is the driven data. The IMPULZ program, with an impulse function length set at 45 points and a delay set at 10 points, was used to recover the digital filter. The results of these tests are shown in Figure 1 as a superposition of the 'no-added noise' case, indistinguishable on the scale shown from the original filter, and four cases of varying levels of added noise. The four levels of 'added noise' tested are 5, 10, 15 and 20% of range, amplitude limited, white noise.

The important features of the 'noisy' data responses are (1) the rounding of the calculated impulse near the zero time, (2) the existence of a significant acausal part of the impulse which is approximately equal in area to the discrepancy in the casual part of the response, and (3) the more accurate approximation to the filter at large displacements from the impulse zero time.

As the amplitude of the added white noise, measured as a percentage of the range of the initial 427.8 nm data, is increased these discrepancies become more pronounced.

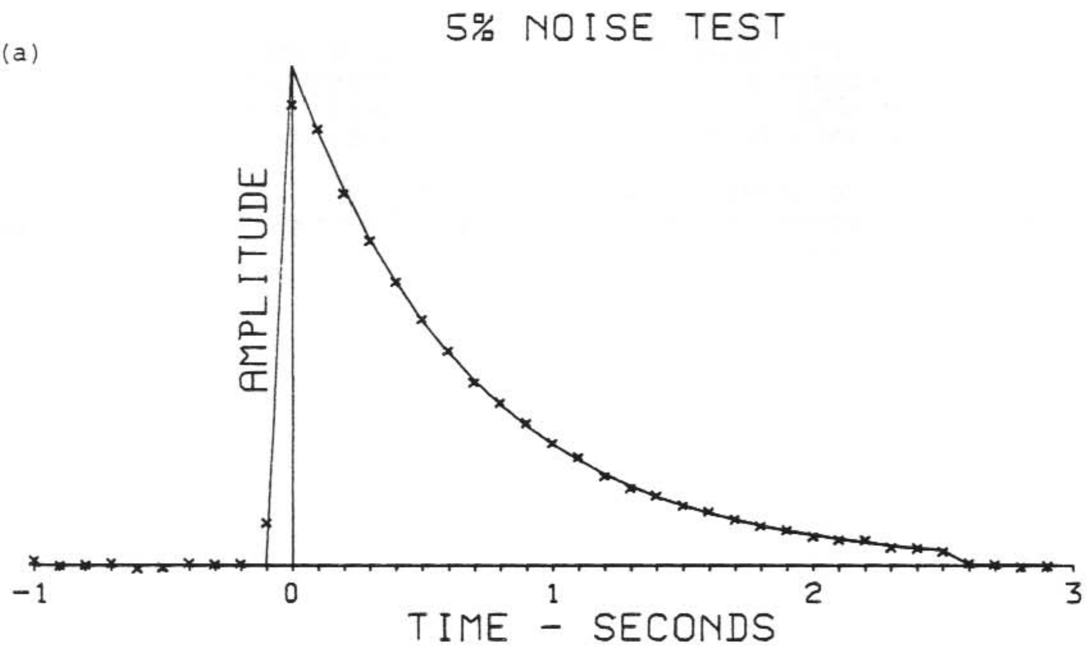
An accurate estimation of the decay time of the filter can still be obtained from the low level 'noisy' data cases by fitting to points away from the zero time of the response. Determination of the decay time of an impulse function was the initial application of the IMPULZ program. The decay time was determined from the least squares 'best-fit' exponential filter to 15 consecutive points of the calculated impulse function. At this stage it is convenient to note that the 427.8 nm data was collected at 10 Hz. A 7 point decay constant thus corresponds to a 0.7 second decay constant. Using the fitting technique described above, estimates of the decay constant of 0.70, 0.70, 0.71, 0.73 and 0.75 seconds were obtained for the 0, 5, 10, 15 and 20% of range added noise cases respectively.

A further test of the impulse program was to generate the filtered data, and then to add a linear increase or decrease to the initial or filtered data prior to calculation of the impulse function. This corresponds to the case of a variation in the 557.7 nm or 427.nm channel, without an associated variation in the other channel. Figure 2(a) shows the superposition of the calculated responses resulting from a 20% of range linear increase and decrease in the driving 427.8 nm array, and the uncontaminated response. Figure 2(b) shows the corresponding responses for a 20% linear variation in the driven 557.7 nm function. In all cases the variation in the calculated response is significantly different from the initial 25 point digital filter. The estimated decay times for the responses due to the linear increase and decrease in the driving function are respectively 0.59 and 0.74 seconds. The corresponding figures for a 20% linear increase and decrease in the driven function data are 0.85 and 0.53 seconds. The key feature of the calculated responses is the large variation from a zero value of the first and last values of the response. This feature may be used in the data reduction to discriminate against these events.

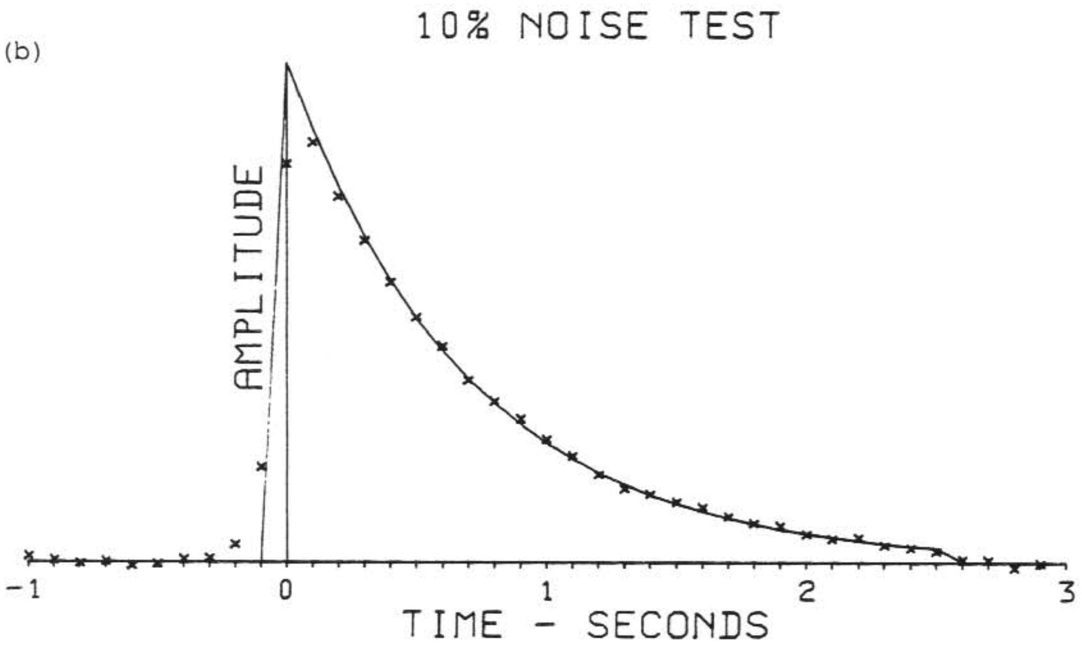
The initial 427.8 nm data string is shown in Figure 3 along with the 557.7 nm line emission measured at the same time and a constructed data string which consists of the 427.8 nm data convoluted with a least squares best fit, twin exponential function relating the two photometric data strings.

The program used to generate the test data, FAKE.FOR, and the program used to determine the exponential decays of the calculated impulse functions, EXPFIT.FOR, are listed in the Appendix.

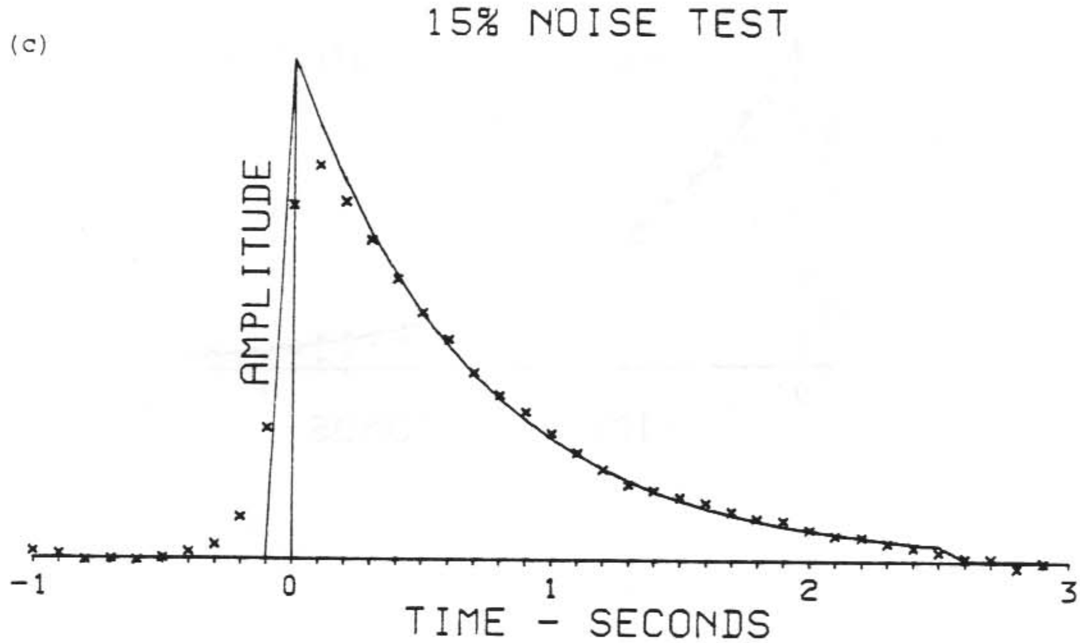
(a)


5% NOISE TEST
AMPLITUDE
-1
0
1
2
3
TIME - SECONDS

(b)


10% NOISE TEST
AMPLITUDE
-1
0
1
2
3
TIME - SECONDS

(c)



(d)

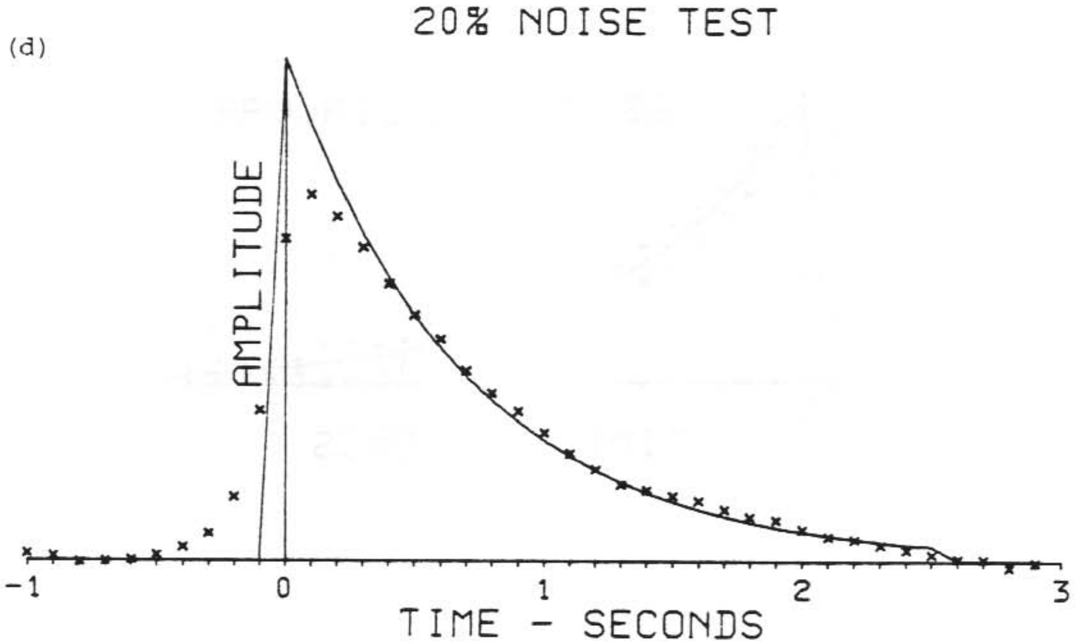
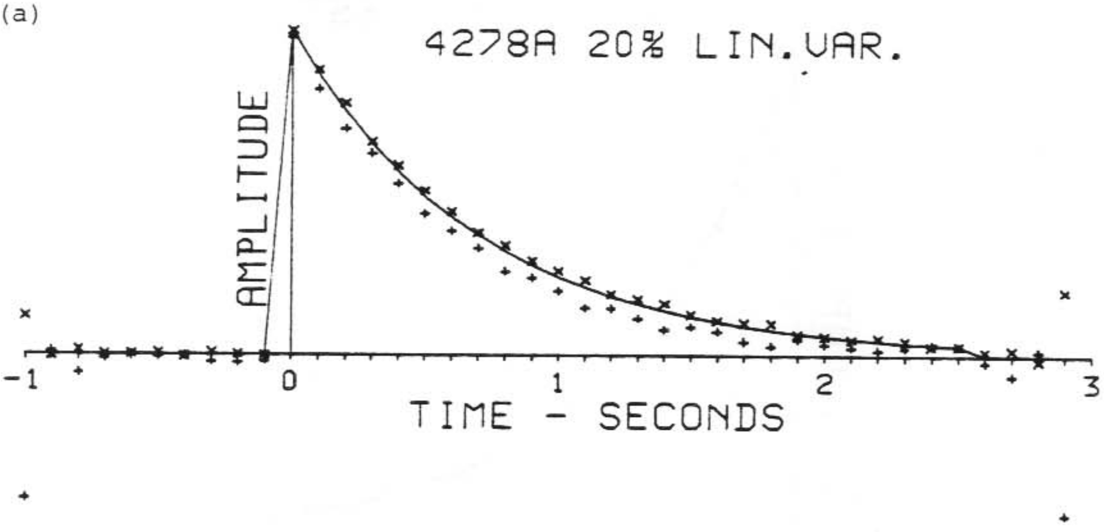


Figure 1. Tests of the susceptibility of the impulse response program to noise. The headings indicate the amplitude limitations of the added noise as a percentage of the range of the initial 427.8 nm data. The solid line marks the 'no-added noise' case, indistinguishable on this scale from the initial filter.

(a)



(b)

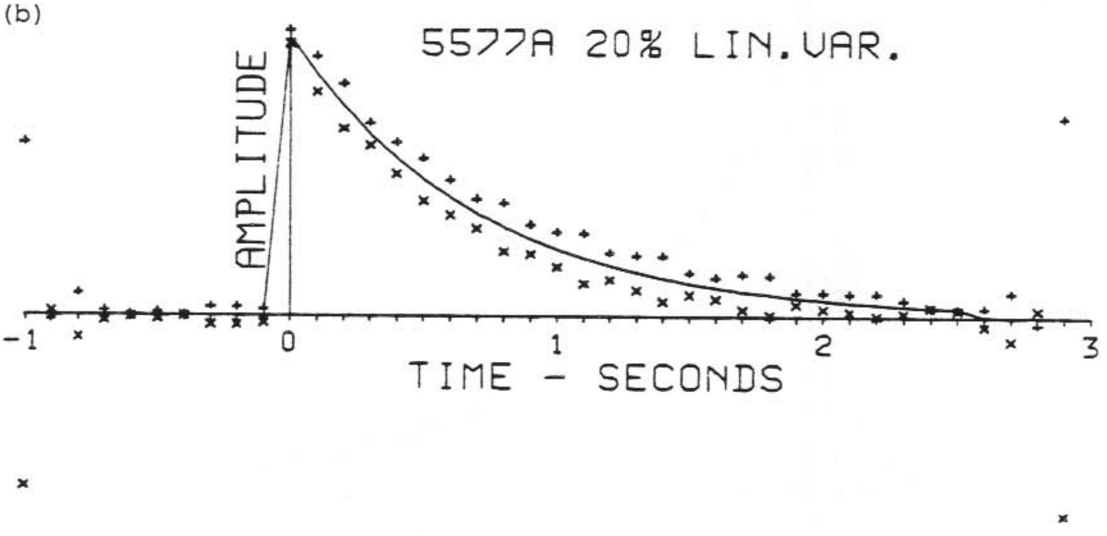


Figure 2. Tests of the response of the impulse program to linear background variations in the respective channels. The results of an added linear variation are denoted by addition signs (+) and a linear decrease by multiplication signs (x). The solid line marks the shape of the initial filter.

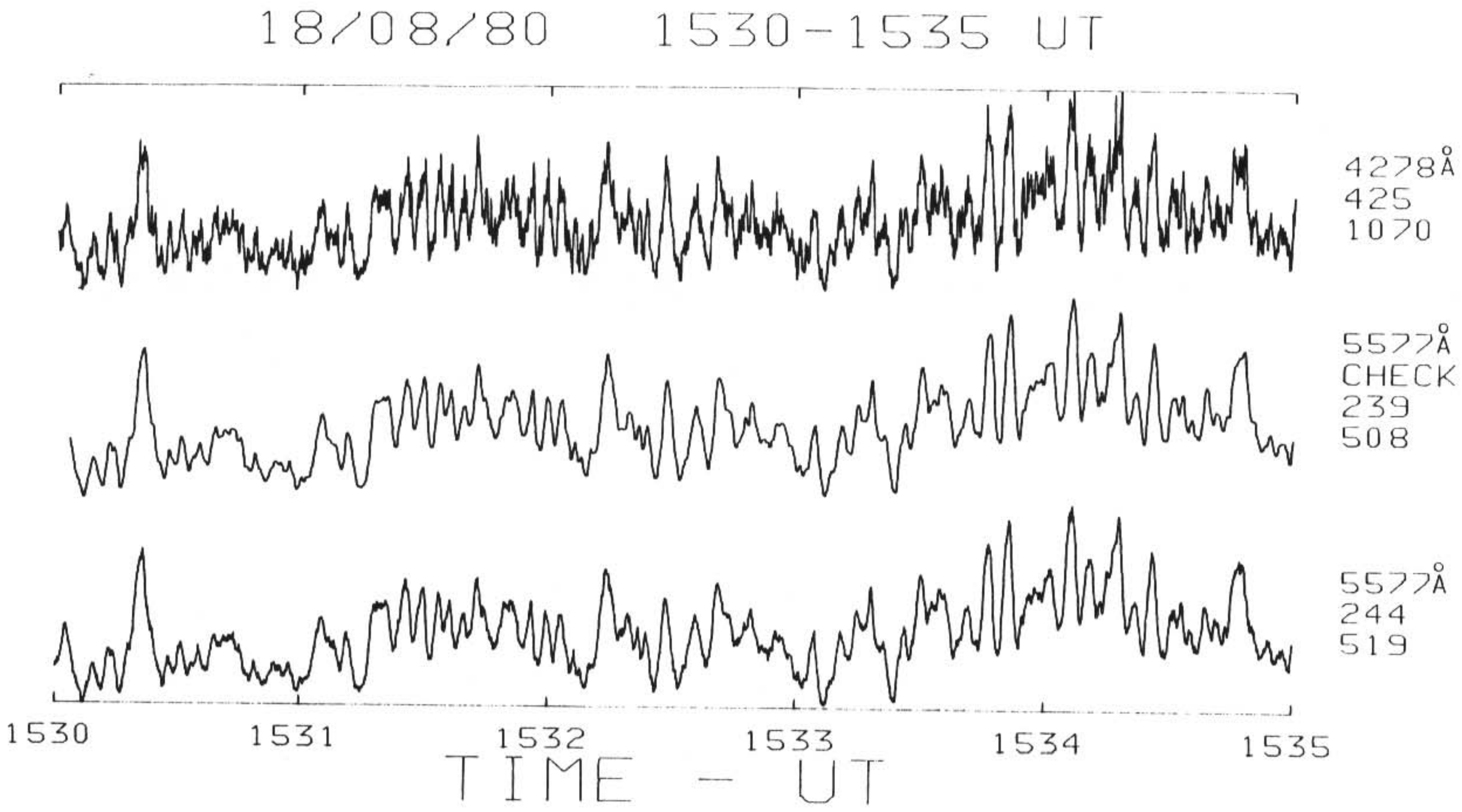


Figure 3. The measured 427.8 nm emission profile used as a starting array for the tests. Also shown is the measured 557.7 nm profile and the result of convoluting the initial 427.8 nm array with a best fit twin exponential impulse function to the impulse function calculated using IMPHOT.

```
C
C         PROGRAM IMPULZ
C
C   TO BE MADE OPERATIONAL THIS MAIN PROGRAM MUST BE 'LINKED'
C   WITH THE SUBROUTINES.....
C         STATS, FASTNT, REV, INVSYM, FACSYM, IPP AND PEF.
C
C   THIS IMPULSE PROGRAM CALCULATES AN IMPULSE FUNCTION OF UP TO
C   100 POINTS IN LENGTH FROM DRIVING (XV) AND DRIVEN (YV) DATA
C   ARRAYS OF UP TO 3000 POINTS.
C
C   TO VARY THE MAXIMUM NUMBER OF POINTS IN THE IMPULSE FUNCTION
C   TO 'N', IT IS NECESSARY TO ALTER THE DIMENSION OF THE DOUBLE
C   PRECISIONED ARRAYS A, W, B, ALPHA AND T TO 'N' AND OF THE
C   ARRAY S TO 'N*(N-1)/2'.
C
C   TO VARY THE MAXIMUM NUMBER OF POINTS ALLOWED FOR AN INPUT DATA
C   ARRAY TO 'NN', IT IS NECESSARY TO ALTER THE DIMENSION OF THE
C   COMMONED ARRAYS XV AND YV TO 'NN'. THE DIMENSION OF XV AND YV
C   MUST ALSO BE CHANGED TO 'NN' IN THE SUBROUTINES 'STATS' AND
C   'FASTNT'.
C
C   THE INPUT DATA VALUES, INCLUDING FILE HEADERS, ARE ASSUMED
C   TO BE INTEGERS.
C
C   THE ORDER (NU) OF THE IMPULSE FUNCTION IS THE NUMBER OF POINTS
C   IN THE FUNCTION, AND THE DELAY (NULM) IS THE NUMBER OF THESE
C   POINTS WHICH ARE ACAUSAL I.E. BEFORE THE ZERO IMPULSE TIME.
C
C   THE PROGRAM ASKS FOR THE NUMBER OF DATA HEADERS AND THEN THE
C   NUMBER OF DATA VALUES IN THE DATA FILES. THESE ARE ASSUMED
C   CONSTANT THROUGHOUT A PROGRAM RUN. THE PROGRAM CONTINUOUSLY
C   LOOPS TO STATEMENT 50 AFTER EVALUATING THE IMPULSE FUNCTION.
C   THE PROGRAM ACCEPTS A NEW ORDER (NU) AND DELAY (NULM) EACH LOOP.
C   TO EXIT FROM THE PROGRAM IT IS NECESSARY TO HIT A 'CONTOL C' OR
C   SYSTEM EQUIVALENT. IT AINT PRETTY BUT IT SAVES TIME WHEN YOUR
C   USING THE PROGRAM REGULARLY!
C
C   THE MATHMATICAL IDEAS CONTAINED IN THIS PROGRAM, AND THE
C   ORIGINAL PROGRAM, ORIGINATED FROM.....
C         DR. J. S. REID
C         REID RESEARCH
C         21 SERVICE ST
C         GLEBE, TASMANIA
C         AUSTRALIA, 7000
C
C   THE PROGRAM HAS BEEN TESTED AND RUN BY ....
C         DR. GARY BURNS
C         UPPER ATMOSPHERE PHYSICS SECTION
C         ANTARCTIC DIVISION
C         DEPARTMENT OF SCIENCE AND TECHNOLOGY
C         CHANNEL HWY
C         KINGSTON, TASMANIA
C         AUSTRALIA, 7150
C
```

```
      CHARACTER*14 FILE1,FILE2,NPLT
      COMMON XMEAN,YMEAN,XV(3000),YV(3000)
      DOUBLE PRECISION XFILE,YFILE
      LOGICAL ERROR
      DOUBLE PRECISION A(100),W(100),B(100),S(4950),ALPHA(100),
     1T(100)
      EPS=1.E-6
      CON=0.95
      TYPE 121
121   FORMAT(' NUMBER OF DATA HEADERS IN INPUT FILES: ',$)
      ACCEPT 122,IHEAD
122   FORMAT(I)
      TYPE 123
123   FORMAT(' NUMBER OF DATA POINTS IN INPUT FILES: ',$)
      ACCEPT 122,IJK
50    TYPE 120
120   FORMAT(' TYPE PLOT FILE SPECS: '$)
      ACCEPT 102,NPLT
102   FORMAT(A14)
      OPEN(UNIT=4,TYPE='NEW',NAME=NPLT)
      TYPE 201
201   FORMAT(' DRIVING FILE: ',$)
      ACCEPT 202,FILE1
202   FORMAT(A14)
      TYPE 203
203   FORMAT(' DRIVEN FILE: ',$)
      ACCEPT 202,FILE2
      OPEN(UNIT=1,TYPE='OLD',FILE=FILE1)
      OPEN(UNIT=2,TYPE='OLD',FILE=FILE2)
C
C  THIS LOOP IS INCLUDED TO ACCOUNT FOR THE FILE HEADERS COMMON
C  ON ANTARCTIC DIVISION DATA FILES.
C
      IF(IHEAD.EQ.0)GO TO 223
      DO 222 J=1,IHEAD
      READ(1,204)ITIM
      READ(2,204)ITIM
204   FORMAT(I)
222   CONTINUE
223   CONTINUE
      DO 205 J=1,IJK
      READ(1,204,END=211)IXVAL
      READ(2,204,END=211)IYVAL
      XV(J)=FLOAT(IXVAL)
      YV(J)=FLOAT(IYVAL)
205   CONTINUE
      GO TO 221
211   TYPE 215
215   FORMAT(' READ ERROR')
      STOP
221   CONTINUE
1     CONTINUE
2     TYPE 105
105   FORMAT(' INPUT ORDER NU AND DELAY ',$)
      ACCEPT 106,NU,NULM
```

```
106     FORMAT(2I)
        IF(NU.LE.100)GO TO 3
        TYPE 107
107     FORMAT(' NU.GT.100 ')
        GO TO 2
3       CONTINUE
        TYPE 108,NU,NULM
108     FORMAT(' MAX LAG ',I5,' DELAY ',I5)
        CALL STATS(SDX,SDY,NU,NULM,IJK)
        TYPE 104,XMEAN,SDX
104     FORMAT(' DRIVING FILE MEAN ',F7.2,' S.D. ',F7.2)
        TYPE 101,YMEAN,SDY
101     FORMAT(' DRIVEN FILE MEAN ',F7.2,' S.D. ',F7.2)
4       CONTINUE
C
C  SOLVE FOR NON-TOEPLITZ ALPHA VALUES AND S-MATRIX
C
        NSX=0
        NSY=0
        N1=3000
        N2=3000
C
C  INCORPORATE DELAY (CAN BE +VE OR -VE)
C
        IF(NULM)5,9,7
5       NSY=-NULM
        N2=N2-NSY
        GO TO 9
7       NSX=NULM
        N1=N1-NULM
9       CONTINUE
C
C  COMPILE X'X MATRIX AND X'Y VECTOR
C
        NNN=MIN0(N1,N2)
        NU2=NNN-2*NU
        NUNU=NU*(NU+1)/2
        CALL FASTNT(S,NSX,NSY,NU,NUNU,NNN,A,W,B,T,RHOY)
        CALL REV(S,NU,NUNU)
        RHO1=S(1)
        CALL INVSYM(S,NU,NUNU,EPS,IER)
        IF(IER.NE.0)GO TO 11
C
C  CHECK FOR STABLE SOLUTION
C  TYPE 110,RHO1
110     FORMAT(//' RHOX= ',E20.8)
        SPM1=1./S(1)
        TYPE 111,SPM1
111     FORMAT(' 1/S11= ',E20.8)
        TOL=0.2*RHO1*SQRT(FLOAT(NU))*EPS
        TYPE 112,TOL
112     FORMAT(' TOL = ',E20.8)
        IF(SPM1.LT.TOL)GO TO 11
C
C  FIND ALPHA VALUES
```

```
C
        CALL PEF(S,NU,NUNU,T,ALPHA)
        TYPE 113,RHOY
113     FORMAT(' RHOY= ',E20.8)
        TYPE 114
114     FORMAT(///' THESE ARE THE H VALUES ')
C
C  OUTPUT ALPHA VALUES
C
        TYPE 115,(I,ALPHA(I),I=1,NU)
115     FORMAT(1X,I5,1PE13.4)
        DO 12 I=1,NU
        XX=FLOAT(I)
        WRITE(4,116)XX,ALPHA(I)
116     FORMAT(1X,F4.1,3X,F9.5)
12      CONTINUE
        CLOSE(UNIT=1)
        CLOSE(UNIT=2)
        CLOSE(UNIT=4)
        GO TO 50
10      TYPE 117
117     FORMAT(///' OUTPUT FILE ON DX1:IMPPLT.DAT ')
        STOP
C
C  ERROR CONDITIONS FOLLOW
C
11      TYPE 118,IER
118     FORMAT(' COVARIANCE MATRIX S IS ILL-CONDITIONED '/
       1' FACSYM ERROR CODE = ',I5/
       2' TRY SMALLER M! ')
        GO TO 1
13      TYPE 119
119     FORMAT(' SEQUENCE IS ALMOST DETERMINISTIC '//
       1' DECREASE M BY 1.')
        GO TO 1
        END
```

```
C
C          STATS.FOR
C
C   THIS SUBROUTINE CALCULATES THE MEAN AND STANDARD DEVIATION
C   OF THE DRIVING , XV(3000), AND DRIVEN, YV(3000), DATA FILES
C   USED IN THE IMPULSE ANALYSIS. THE MEAN AND STANDARD DEVIATION
C   ARE CALCULATED USING ONLY THOSE DATA POINTS FROM EACH FILE
C   WHICH ARE COMMON TO THE ANALYSIS. THE RESULTS OF AN IMPULSE
C   ANALYSIS ARE DUBIOUS IF THESE VALUES DIFFER SIGNIFICANTLY
C   FROM THE VALUES OBTAINED FOR THE MEAN AND STANDARD DEVIATION
C   WHEN THE ENTIRE DATA FILES ARE USED.
C
C   IJK = NUMBER OF DATA PTS IN EACH FILE. (MUST BE EQUAL).
C   NU = NUMBER OF PTS IN IMPULSE FUNCTION TO BE CALCULATED.
C   NULM = TIME DISPLACEMENT. THE NUMBER OF PTS TO BE EVALUATED
C          PRIOR TO ZERO IMPULSE TIME.
C   XMEAN = MEAN OF DRIVING FILE
C   YMEAN = MEAN OF DRIVEN FILE.
C   SDX = STANDARD DEVIATION OF DRIVING FILE.
C   SDY = STANDARD DEVIATION OF DRIVEN FILE.
C
        SUBROUTINE STATS(SDX,SDY,NU,NULM,IJK)
        COMMON XMEAN,YMEAN,XV(3000),YV(3000)
        IN=IJK-NU-NULM+1
        XBAR=0.
        YBAR=0.
        XSUM=0.
        YSUM=0.
        SUMX2=0.
        SUMY2=0.
        DO 1 I=NU,IJK-NULM
        XSUM=XSUM+XV(I)
        YSUM=YSUM+YV(I)
        SUMX2=SUMX2+XV(I)**2.
        SUMY2=SUMY2+YV(I)**2.
1       CONTINUE
        XMEAN=XSUM/FLOAT(IN)
        YMEAN=YSUM/FLOAT(IN)
        VARX=(SUMX2-XSUM*XMEAN)/FLOAT(IN-1)
        VARY=(SUMY2-YSUM*YMEAN)/FLOAT(IN-1)
        SDX=SQRT(VARX)
        SDY=SQRT(VARY)
        RETURN
        END
```

```
      SUBROUTINE FASTNT(S,NSX,NSY,L,LL,N,A,W,B,T,RHO)
C
C  I/O FREE SUBROUTINE TO COMPUTE NON-TOEPLITZ MATRIX
C  OF SECOND MOMENTS OR COVARIANCES OF A LIST OF NUMBERS
C
C  S:      OUTPUT MATRIX ... ONE TRIANGULE ONLY STORED AS
C          LINEAR ARRAY USING IPP(I,J) TO DETERMINE INDEX
C  L:      (=NU) I.E. DIMENSIONS OF ARRAYS, W AND A = MAXLAG
C          IN TRIANGULAR MATRIX OF DIMENSION NU.
C  N:      NUMBER OF USABLE NUMBERS IN LIST,ALLOWS FOR LAG AND DELAY
C  A,W     WORK ARRAYS OF DIMENSION L.
C  T:      X'Y VECTOR
C  RHO:    SUM Y**2 VALUE
C
      COMMON XMEAN,YMEAN,XV(3000),YV(3000)
      DOUBLE PRECISION S(LL),A(L),W(L),B(L),T(L)
      ND=N-L+1
C
C  CLEAR MATRIX AND WORK VECTORS
C
      JX=0
      JY=0
      RHO=0.
      DO 1 I=1,L
      A(I)=0.
      T(I)=0.
      B(I)=0.
      DO 1 J=1,I
      IP=J+(I*I-I)/2
      S(IP)=0.
1     CONTINUE
C
C  LOOP FOR EACH INPUT NUMBER,X.
C  SLIDE VALUES ALONG WORK VECTOR W.
C
      DO 10 IRR=1,N
      IR=IRR
      IF(IR.EQ.1)GO TO 3
      IM=MIN0(IR,L)
      DO 2 I=IM,2,-1
2     W(I)=W(I-1)
3     CONTINUE
      JX=JX+1
      X=XV(JX+NSX)
      X=X-XMEAN
      W(1)=X
      Y=YV(JX+NSY)
      Y=Y-YMEAN
      IF(IR.GE.L)GO TO 5
C
C  ADD WORK VECTOR MOMENTS INTO MATRIX IF
C  X IS ALONG FIRST L VALUES.
C
      DO 4 I=1,IR
      DO 4 J=1,I
```

```
      IP=J+(I*I-I)/2
4     S(IP)=S(IP)+W(1)*W(I-J+1)
      GO TO 10
C
C  COMPUTE COLUMN SUMS AND X'Y VECTOR (T)
C
5     RHO=RHO+Y*Y
      DO 6 I=1,L
      T(I)=T(I)+Y*W(I)
6     B(I)=B(I)+W(L-I+1)
C
C  ADD MOMENTS INTO VECTOR A FOR TOEPLITZ PART.
C
      IF(IR.GT.ND)GO TO 8
      DO 7 I=1,L
7     A(I)=A(I)+W(1)*W(I)
      GO TO 10
C
C  ADD MOMENTS INTO MATRIX IF X ALONG LAST L-1 NUMBERS
C
8     LND =IR-ND+1
      DO 9 I=LND,L
      DO 9 J=1,I
      IP=J+(I*I-I)/2
9     S(IP)=S(IP)+W(1)*W(I-J+1)
10    CONTINUE
C
C  FINALLY ADD TOEPLITZ PART FROM VECTOR A.
C
      DO 11 I=1,L
      DO 11 J=1,I
      IP=J+(I*I-I)/2
11    S(IP)=S(IP)+A(I-J+1)-B(I)*B(J)/FLOAT(ND)
      RETURN
      END
```

```
      SUBROUTINE REV(S,L,LL)
C
C  REVERSES MATRIX S ABOUT SINISTER DIAGONAL
C  THIS ENSURES THAT FORWARD RATHER THAN BACKWARD
C  PEF IS FOUND
C
      DOUBLE PRECISION S(LL)
      DO 2 I=1,L
      L1=L-I+1
      L2=L1/2
      DO 1 J=1,L2
      I1=I+J-1
      I2=J
      I3=L-I2+1
      I4=L-I1+1
      IP1=I2+(I1*I1-I1)/2
      IP2=I4+(I3*I3-I3)/2
      W=S(IP1)
      S(IP1)=S(IP2)
      S(IP2)=W
1     CONTINUE
2     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVSYM(S,NU,NUNU,EPS,IER)
C
C  INVERTS POSITIVE DEFINITE SYMMETRIC MATRIX S
C  BY TRIANGULAR FACTORIZATION AS IN MOST OFF-THE-SHELF
C  SOFTWARE PACKAGES BUT WITH ADDED FACILITY THAT
C  VECTOR B IS RETURNED FOR LATER USEIN OBTAINING PARTIAL
C  SUMS OF RESIDUALS IN R-ALPHA TEST FOR ORDER
C  ESTIMATION.
C
      DOUBLE PRECISION S(NUNU)
      DOUBLE PRECISION DNN,SCRTCH
C
C  FOACTORIZE MATRIX INTO U"U
C
      CALL FACSYM(S,NU,NUNU,EPS,IER)
      IF(IER.LT.0)GO TO 8
C
C  INVERT U
C
1     IPVT=NUNU
      IND=IPVT
      DO 5 I=1,NU
      DNN=1.D0/DBLE(S(IPVT))
      S(IPVT)=DNN
      MIN=NU
      KND=I-1
      KLNF=NU-KND
      IF(KND.LE.0)GO TO 4
      J=IND
      DO 3 K=1,KND
      SCRTCH=0.D0
      MIN=MIN-1
      LOHR=IPVT
      LVER=J
      DO 2 L=KLNF,MIN
      LVER=LVER+1
      LOHR=LOHR+L
2     SCRTCH=SCRTCH+DBLE(S(LVER)*S(LOHR))
      S(J)=-SCRTCH*DNN
3     J=J-MIN
4     IPVT=IPVT-MIN
5     IND=IND-1
C
C  END OF INVERSION LOOPS
C
      DO 7 I=1,NU
      IPVT=IPVT+I
      J=IPVT
      DO 7 K=I,NU
      SCRTCH=0.D0
      LOHR=J
      DO 6 L=K,NU
      LVER=LOHR+K-I
      SCRTCH=SCRTCH+DBLE(S(LOHR)*S(LVER))
6     LOHR=LOHR+L
```

```
      S(J)=SCRTCH
7     J=J+K
8     RETURN
      END
```

```
      SUBROUTINE FACSYM(S,NU,NUNU,EPS,IER)
C
C  FACTORIZE SYMMETRIC POSITIVE DEFINITE MATRIX INTO
C  UPPER TRIANGULAR MATRIX (AND ITS TRANSPOSE)
C
      DOUBLE PRECISION S(NUNU)
      DOUBLE PRECISION PIVT,SUMM
      IF(NU.LT.1)GO TO 6
      IER=0
      KPVT=0
      DO 5 K=1,NU
      KPVT=KPVT+K
      IND=KPVT
      LOND=K-1
      TOL=ABS(EPS*S(KPVT))
      DO 5 I=K,NU
      SUMM=0.D0
      IF(LOND.EQ.0)GO TO 2
      DO 1 L=1,LOND
      KLNF=KPVT-L
      LIND=IND-L
1     SUMM=SUMM+DBLE(S(KLNF)*S(LIND))
2     SUMM=DBLE(S(IND))-SUMM
      IF(I.NE.K)GO TO 4
      IF(SNGL(SUMM).GT.TOL)GO TO 3
      IF(SUMM.LE.0.D0)GO TO 6
      IF(IER.GT.0)GO TO 3
      IER=K-1
3     PIVT=DSQRT(SUMM)
      S(KPVT)=PIVT
      PIVT=1.D0/PIVT
      GO TO 5
4     S(IND)=SUMM*PIVT
5     IND=IND+I
      RETURN
6     IER=-1
      RETURN
      END
```

```
      FUNCTION IPP(I,J)
      IF(I-J)1,2,2
1     IPP=I+(J*J-J)/2
      RETURN
2     IPP=J+(I*I-I)/2
      RETURN
      END
```

```
      SUBROUTINE PEF(S,NU,NUNU,T,ALPHA)
C
C  RETURNS PREDICTION ERROR COEFFICIENTS ALPHA(I)
C  AND RESIDUAL ERROR POWER,PN.
C
C  NOTE: ALPHA(0)  [=1] IS NOT RETURNED
C
      DOUBLE PRECISION S(NUNU),T(NU),ALPHA(NU)
      DO 2 I=1,NU
      SUM=0.
      IS=I
      DO 1 J=1,NU
      JS=J
      IP=IPP(IS,JS)
      SUM=SUM+S(IP)*T(J)
1     CONTINUE
      ALPHA(I)=SUM
2     CONTINUE
      RETURN
      END
```

```
C
C         PROGRAM IMPHOT
C
C  TO BE MADE OPERATIONAL THIS MAIN PROGRAM MUST BE 'LINKED'
C  WITH THE SUBROUTINES.....
C         STATS, FASTNT, REV, INVSYM, FACSYM, IPP, AND PEF.
C
C  THIS IMPULSE RESPONSE PROGRAM HAS BEEN MODIFIED SPECIFICALLY
C  TO CALCULATE THE 55 POINT (5.5 SECONDS), 10 POINT DELAY
C  (1.0 SECONDS) IMPULSE FUNCTIONS FROM THE 3000 POINT (5 MINUTE)
C  427.8 NM (DRIVING) AND 557.7 NM (DRIVEN) DATA FILES.
C
C  THE ORDER (NU) OF THE IMPULSE FUNCTION IS SET AT 55, THE DELAY
C  (NULM) AT 10 AND THE NUMBER OF DATA POINTS (IJK) IN THE DRIVING
C  (XV) AND DRIVEN (YV) ARRAYS IS FIXED AT 3000.
C
C  NOTE: THE PROGRAM AS FIXED IS VALID ONLY FOR PHOTOMETER FILES
C  WHICH HAVE ONLY ONE FILE INFORMATION RECORD, THE FIRST. IN
C  1983 THE DATA COLLECTION PROGRAM WAS UPGRADED TO INCLUDE SIX
C  INFORMATION RECORDS AT THE BEGINNING OF THE FILE. TO DEAL WITH
C  DATA COLLECTED AFTER THIS TIME, THE TWO 'IF' STATEMENTS JUST
C  PRIOR TO STATEMENT 205 MUST BE APPROPRIATELY MODIFIED!!!
C
C
C
C  THE MATHMATICAL IDEAS CONTAINED IN THIS PROGRAM, AND THE
C  ORIGINAL PROGRAM, ORIGINATED FROM....
C         DR. J. S. REID
C         REID RESEARCH
C         21 SERVICE ST
C         GLEBE, TASMANIA
C         AUSTRALIA, 7000
C
C  THE PROGRAM HAS BEEN TESTED AND RUN BY.....
C         DR. GARY BURNS
C         UPPER ATMOSPHERE PHYSICS SECTION
C         ANTARCTIC DIVISION
C         DEPARTMENT OF SCIENCE AND TECHNOLOGY
C         CHANNEL HWY
C         KINGSTON, TASMANIA
C         AUSTRALIA, 7150
C
      DIMENSION IN1(256),IN2(256)
      CHARACTER*15 NAME1,NAME2,NPLT
      COMMON XMEAN,YMEAN,XV(3000),YV(3000)
      DOUBLE PRECISION XFILE,YFILE
      LOGICAL ERROR
      INTEGER*2 IN1,IN2
      DOUBLE PRECISION A(100),W(100),B(100),S(4950),ALPHA(100),
     1T(100)
      EPS=1.E-6
      CON=0.95
50    TYPE 120
120   FORMAT(' TYPE PLOT FILE SPECS: '$)
      ACCEPT 102,NPLT
```

```
102       FORMAT(A15)
          OPEN(UNIT=4,TYPE='NEW',NAME=NPLT)
C
C  THIS SECTION READS IN NUMBERS FOR DRIVING AND
C  DRIVEN FUNCTIONS THAT HAVE BEEN WRITTEN BY EYES.
C
C  THE FOLLOWING OPEN STATEMENTS ARE MORE COMPLICATED
C  THAN NORMAL BECAUSE OF THE WAY THE DATA HAS BEEN
C  PACKED (IN BINARY FORM, 1 DATA POINT PER WORD).
C  THIS WAS DONE IN ORDER TO SAVE SPACE ON THE INITIAL
C  DATA STORAGE MEDIUM WHICH WAS SINGLE DENSITY, 8 INCH
C  FLOPPY DISKS.
C
          TYPE 201
201       FORMAT(' INPUT DRIVING FILE NAME: ',$)
          ACCEPT 202,NAME1
202       FORMAT(A15)
          TYPE 203
203       FORMAT(' INPUT DRIVEN FILE NAME: ',$)
          ACCEPT 202,NAME2
          OPEN(UNIT=1,STATUS='OLD',FILE=NAME1,ACCESS='SEQUENTIAL',
          1FORM='UNFORMATTED',RECORDTYPE='FIXED',RECL=128)
          OPEN(UNIT=2,STATUS='OLD',FILE=NAME2,ACCESS='SEQUENTIAL',
          1FORM='UNFORMATTED',RECORDTYPE='FIXED',RECL=128)
          K=0
          DO 204 I=1,120
          READ(1)IN1
          READ(2)IN2
          DO 205 J=1,256
          K=K+1
C
C  THE FOLLOWING 'IF' STATEMENT IS INCLUDED TO IGNORE
C  THE FIRST DATA POINT IN THE INPUT FILES. THE FIRST
C  DATA POINT IN THE PHOTOMETER FILES IS THE NUMBER OF
C  UT MINUTES SINCE THE START OF THE UT DAY AT WHICH
C  THE DATA COLLECTION FOR THE FILE IN QUESTION
C  COMMENCED!   COMPLICATED HUH!
C
          IF(K.EQ.1)GO TO 205
          XV(K-1)=FLOAT(IN1(J))
          YV(K-1)=FLOAT(IN2(J))
          IF(K.EQ.3001)GO TO 206
205       CONTINUE
204       CONTINUE
206       CONTINUE
          CLOSE(UNIT=1)
          CLOSE(UNIT=2)
1         CONTINUE
2         CONTINUE
          IJK=3000
          NU=55
          NULM=10
106       FORMAT(2I)
          IF(NU.LE.100)GO TO 3
          TYPE 107
```

```
107      FORMAT(' NU.GT.100 ')
         GO TO 2
3        CONTINUE
         TYPE 108,NU,NULM
108      FORMAT(' MAX LAG ',I5,' DELAY ',I5)
         CALL STATS(SDX,SDY,NU,NULM,IJK)
         TYPE 104,XMEAN,SDX
104      FORMAT(' DRIVING FILE MEAN ',F7.2,' S.D. ',F7.2)
         TYPE 101,YMEAN,SDY
101      FORMAT(' DRIVEN FILE MEAN ',F7.2,' S.D. ',F7.2)
4        CONTINUE
C
C   SOLVE FOR NON-TOEPLITZ ALPHA VALUES AND S-MATRIX
C
         NSX=0
         NSY=0
         N1=3000
         N2=3000
C
C   INCORPORATE DELAY (CAN BE +VE OR -VE)
C
         IF(NULM)5,9,7
5        NSY=-NULM
         N2=N2-NSY
         GO TO 9
7        NSX=NULM
         N1=N1-NULM
9        CONTINUE
C
C   COMPILE X'X MATRIX AND X'Y VECTOR
C
         NNN=MIN0(N1,N2)
         NU2=NNN-2*NU
         NUNU=NU*(NU+1)/2
         CALL FASTNT(S,NSX,NSY,NU,NUNU,NNN,A,W,B,T,RHOY)
         CALL REV(S,NU,NUNU)
         RHO1=S(1)
         CALL INVSYM(S,NU,NUNU,EPS,IER)
         IF(IER.NE.0)GO TO 11
C
C   CHECK FOR STABLE SOLUTION
C   TYPE 110,RHO1
110      FORMAT(//' RHOX= ',E20.8)
         SPM1=1./S(1)
         TYPE 111,SPM1
111      FORMAT(' 1/S11= ',E20.8)
         TOL=0.2*RHO1*SQRT(FLOAT(NU))*EPS
         TYPE 112,TOL
112      FORMAT(' TOL = ',E20.8)
         IF(SPM1.LT.TOL)GO TO 11
C
C   FIND ALPHA VALUES
C
         CALL PEF(S,NU,NUNU,T,ALPHA)
         TYPE 113,RHOY
```

```
113     FORMAT(' RHOY= ',E20.8)
        TYPE 114
114     FORMAT(///' THESE ARE THE H VALUES ')
C
C  OUTPUT ALPHA VALUES
C
        TYPE 115,(I,ALPHA(I),I=1,NU)
115     FORMAT(1X,I5,1PE13.4)
        DO 12 I=1,NU
        XX=FLOAT(I)
        WRITE(4,116)XX,ALPHA(I)
116     FORMAT(1X,F4.1,3X,F9.5)
12      CONTINUE
        WRITE(4,121)NPLT
121     FORMAT(1X,A15)
        CLOSE(UNIT=1)
        CLOSE(UNIT=2)
        CLOSE(UNIT=4)
        GO TO 50
10      TYPE 117
117     FORMAT(///' OUTPUT FILE ON DX1:IMPPLT.DAT ')
        STOP
C
C  ERROR CONDITIONS FOLLOW
C
11      TYPE 118,IER
118     FORMAT(' COVARIANCE MATRIX S IS ILL-CONDITIONED '/
        1' FACSYM ERROR CODE = ',I5/
        2' TRY SMALLER M! ')
        GO TO 1
13      TYPE 119
119     FORMAT(' SEQUENCE IS ALMOST DETERMINISTIC '//
        1' DECREASE M BY 1.')
        GO TO 1
        END
```

```
C
C           BINCON
C
C  THIS PROGRAM READS BINARY FILES, AS GENERATED
C  BY THE 'EYES' AND 'EARS' SAMPLING PROGRAMS
C  OF ANTARCTIC DIVISION, AND CONVERTS THEM TO
C  FORTRAN INTERNAL FORMAT.
C
        DIMENSION IN(256),IOUT(3001)
        CHARACTER*14 NAME1
        CHARACTER*15 NAME2
        INTEGER*2 IN
        TYPE 101
101     FORMAT(' INPUT BINARY FILE NAME: ',$)
        ACCEPT 102,NAME1
102     FORMAT(A14)
        TYPE 103
103     FORMAT(' OUTPUT ASCII FILE NAME: ',$)
        ACCEPT 104,NAME2
104     FORMAT(A15)
        OPEN(UNIT=1,STATUS='OLD',FILE=NAME1,
        1ACCESS='SEQUENTIAL',FORM='UNFORMATTED',
        2RECORDTYPE='FIXED',RECL=128)
        OPEN(UNIT=2,STATUS='NEW',FILE=NAME2)
        K=0
        DO 1 I=1,300
        READ(1)IN
        DO 2 J=1,256
        K=K+1
        IOUT(K)=IN(J)
        IF(K.EQ.3001)GO TO 900
100     FORMAT(I5)
2       CONTINUE
1       CONTINUE
900     CONTINUE
        DO 3 K=1,3001
        WRITE(2,100)IOUT(K)
3       CONTINUE
        CLOSE(UNIT=1)
        CLOSE(UNIT=2)
        STOP
        END
```

```
C
C                    FAKE.FOR
C
C  THIS PROGRAM TAKES A 'REAL' 3000 DATA POINT FILE AND
C  GENERATES 3000 POINT TEST ARRAYS OF DRIVING (AA) AND
C  DRIVEN (BB) DATA FOR THE IMPULZ.FOR PROGRAM.
C
C  NOISE, AS A PERCENTAGE OF THE RANGE, CAN BE ADDED TO THE
C  DRIVING DATA ARRAY. THIS IS VIA THE INPUT VARIABLE 'FN'.
C  A 'FN' VALUE OF 0.20 CORRESPONDS TO 20% OF RANGE ADDED
C  NOISE. THE DECAY CONSTANT 'DT' IS ALSO VARIABLE. IN ORDER
C  TO ADD A LINEAR INCREASE OR DECREASE TO THE TEST ARRAYS,
C  IT IS NECESSARY TO INCORPORATE THE MARKED CODE AT THE
C  NOTED POINTS.
C
C  THE 'REAL' DATA THAT FORMS THE STARTING POINT IS
C  GENERALLY THE 5 MIN, 3000 POINT, 427.8 NM FILE
C  COMMENCING AT 1530 UT ON 18/8/80. (B1808.021 IN THE
C  NOMENCLATURE OF MACCA 1980!!!)
C
       DIMENSION IN(256),AA(3000),BB(3000),F(100)
       CHARACTER*15 NAME1,NAME2,INFILE
       INTEGER*2 IN
       TYPE 200
200    FORMAT(' INPUT 4278A FILE: ',$)
       ACCEPT 101,INFILE
101    FORMAT(A15)
       TYPE 100
100    FORMAT(' NAME OF DRIVING FILE: ',$)
       ACCEPT 101,NAME1
       TYPE 102
102    FORMAT(' NAME OF DRIVEN FILE: ',$)
       ACCEPT 101,NAME2
C
C  THE FOLLOWING OPEN STATEMENT IS MORE COMPLICATED THAN
C  NORMAL BECAUSE OF THE WAY THE DATA HAS BEEN PACKED. IT
C  IS PACKED IN BINARY FORM WITH ONE DATA POINT PER WORD.
C
       OPEN(UNIT=1,STATUS='OLD',FILE=INFILE,ACCESS='SEQUENTIAL',
       1FORM='UNFORMATTED',RECORDTYPE='FIXED',RECL=128)
       K=0
       DO 204 I=1,120
       READ(1)IN
       DO 205 J=1,256
       K=K+1
C
C  THE FOLLOWING 'IF' STATEMENT IS INCLUDED TO DISPOSE OF
C  THE FILE HEADER WHICH IS INCLUDED IN 1980 DATA FILES.
C
       IF(K.EQ.1)GO TO 205
       AA(K-1)=FLOAT(IN(J))
       IF(K.EQ.3001)GO TO 206
205    CONTINUE
204    CONTINUE
206    CONTINUE
```

```
      LL=0
      KK=0
      ITIME=1111
      PI=3.1415926
      OPEN(UNIT=3,TYPE='NEW',FILE=NAME1)
      OPEN(UNIT=2,TYPE='NEW',FILE=NAME2)
C
C  A DUMMY FILE HEADER IS WRITTEN TO MAKE THE OUTPUT DATA
C  CONSISTENT WITH TRUE DATA FILES. THE 1111 HEADER IS
C  UNIQUE TO 'FAKED' DATA!
C
      WRITE(3,103)ITIME
      WRITE(2,103)ITIME
103   FORMAT(I5)
      TYPE 106
106   FORMAT(' INPUT DECAY TIME IN TENTHS OF SECS: ',$)
      ACCEPT 107,DT
107   FORMAT(F)
      TYPE 150
150   FORMAT(' INPUT FRACTION OF NOISE: ',$)
      ACCEPT 105,FN
105   FORMAT(F)
      DO 5 I=1,25
      E=FLOAT(I)/DT
      F(I)=EXP(-E)
      TYPE 157,F(I)
157   FORMAT(1X,F8.4)
5     CONTINUE
      AMAX=AA(1)
      AMIN=AA(1)
      DO 7 I=1,3000
      IF(AA(I).GT.AMAX)AMAX=AA(I)
      IF(AA(I).LT.AMIN)AMIN=AA(I)
7     CONTINUE
      RANGE=AMAX-AMIN
      DO 3 J=1,3000
      BB(J)=AA(J)
      IF(J.LT.26)GO TO 3
      DO 6 L=1,25
6     BB(J)=BB(J)+AA(J-L)*F(L)
3     CONTINUE
      BMAX=BB(1)
      BMIN=BB(1)
      DO 4 I=1,3000
      IF(BB(I).GT.BMAX)BMAX=BB(I)
      IF(BB(I).LT.BMIN)BMIN=BB(I)
      AA(I)=AA(I)+FN*(RAN(LL,KK)-0.5)*RANGE
C
C  THE FOLLOWING LINE IS INSERTED AT THIS POINT TO ADD A
C  20% OF RANGE POSITIVE LINEAR TREND TO THE DRIVING DATA
C  AA(I)=AA(I)+RANGE*0.2*FLOAT(I)/3000.
C
C  THE FOLLOWING LINE IS INSERTED IF IT IS DESIRED TO ADD A
C  NEGATIVE 20% OF RANGE TREND TO THE DRIVING DATA.
C  AA(I)=AA(I)-RANGE*0.2*FLOAT(I)/3000.
```

```
4       CONTINUE
        RA1=BMAX-BMIN
        DO 44 I=1,3000
C
C  THE FOLLOWING LINE IS INSERTED TO ADD A 20% OF RANGE
C  POSITIVE LINEAR TREND TO THE DRIVEN FILE.
C  BB(I)=BB(I)+RA1*0.2*FLOAT(I)/3000.
C
C  THE FOLLOWING LINE IS INSERTED TO ADD A 20% OF RANGE
C  NEGATIVE LINEAR TREND TO THE DRIVEN FILE.
C  BB(I)=BB(I)-RA1*0.2*FLOAT(I)/3000.
        LEAD=INT(AA(I))
        IFOLL=INT(BB(I))
        WRITE(3,103)LEAD
        WRITE(2,103)IFOLL
44      CONTINUE
        CLOSE(UNIT=3)
        CLOSE(UNIT=2)
        STOP
        END
```

```
C
C          EXPFIT.FOR
C
C  THIS PROGRAM DETERMINES 'BEST-FIT' EXPONENTIAL DECAYS TO
C  CONSECUTIVE 15 POINTS OF A CALCULATED IMPULSE FUNCTION.
C
C  IT DETERMINES THE MAXIMUM POSITIVE VALUE OF THE CALCULATED
C  IMPULSE FUNCTION, EXCLUDING THE FIRST AND LAST POINTS.
C  IT THEN DETERMINES THE BEST FIT EXPONENTIAL TO 15 CONSECUTIVE
C  POINTS FROM THE PEAK AND THE RMS VALUE OF THAT FIT. IT THEN
C  SHIFTS ONE POINT ALONG FROM THE PEAK AND RECALCULATES THESE
C  VALUES. THE PROGRAM REPEATS THIS PROCESS UNTIL A NEGATIVE
C  VALUE FOR THE IMPILSE FUNCTION IS ENCOUNTERED. THE BEST FIT
C  DECAY TIMES AND THE RMS VALUE OF THE BEST FIT ARE OUTPUT TO
C  THE TERMINAL AND TO THE FILE 'EXP15.DAT'.
C
C  THE PROGRAM LOOPS TO ACCEPT A NEW FILE TO ANALYZE. IT REQUIRES
C  A 'CONTROL C', OR SYSTEM EQUIVALENT, TO EXIT FROM THIS PROGRAM.
C
C  IM = INDICATES POINT AT WHICH MAXIMUM VALUE OF CALCULATED
C       IMPULSE IS REACHED. EXCLUDES FIRST AND LAST POINTS.
C  IZ = POINT AT WHICH CALCULATED IMPULSE FIRST TURNS NEGATIVE
C       AFTER PEAK.
C  J-1 = NUMBER OF POINTS ALONG FROM PEAK THAT 15 CONSECUTIVE
C       POINTS BEING FITTED STARTS.
C  IS = START POINT FOR 15 CONSECUTIVE POINTS.
C  IE = END POINT FOR 15 CONSECUTIVE POINTS.
C
        DIMENSION X(100),Y(100)
        CHARACTER*16 NAME1,NAME2
3       CONTINUE
        TYPE 100
100     FORMAT(' INPUT FILENAME: ',$)
        ACCEPT 101,NAME1
101     FORMAT(A16)
        OPEN(UNIT=2,TYPE='OLD',FILE=NAME1)
        OPEN(UNIT=3,TYPE='NEW',FILE='EXP15.DAT')
        WRITE(3,77)NAME1
77      FORMAT(1X,A15)
        INUM=55
103     FORMAT(I)
        DO 1 I=1,INUM-1
        READ(2,104)X(I),Y(I)
104     FORMAT(1X,F4.1,3X,F9.5)
        IF(I.EQ.1)GO TO 1
        IF(I.EQ.2)YMAX=Y(2)
        IF(I.EQ.2)II=2
        IF(Y(I).GT.YMAX)IM=I
        IF(Y(I).GT.YMAX)YMAX=Y(I)
        III=I
1       CONTINUE
        IZ=0
        DO 4 I=IM,INUM
        IF(IZ.NE.0)GO TO 5
        IF(Y(I).LE.0.)IZ=I
```

```
4         CONTINUE
          IF(IZ.EQ.0)IZ=INUM
5         CONTINUE
          IF((IZ-IM).LT.15)GO TO 999
          DO 6 J=1,IZ-IM-14
          YLN=0.
          SLNY2=0.
          SLNY=0.
          SXLNY=0.
          SX2=0.
          SX=0.
          IS=IM+J-1
          IE=IM+J+13
          DO 2 I=IS,IE
          YLOG=ALOG(Y(I))
          SLNY2=SLNY2+YLOG**2
          SLNY=SLNY+YLOG
          SXLNY=FLOAT(I)*YLOG+SXLNY
          SX2=SX2+(FLOAT(I))**2
          SX=SX+FLOAT(I)
2         CONTINUE
          B1=SXLNY-SX*SLNY/15.
          B2=SX2-SX*SX/15.
          B=B1/B2
          A=(SLNY-B*SX)/15.
          A=EXP(A)
          A=A*EXP(B*11.)
          R1=SLNY2-SLNY*SLNY/15.
          R2=B1*B1/(B2*R1)
          R=SQRT(R2)
          T=1./(10.*B)
          TYPE 105,J,B,T,R2,A
          WRITE(3,105),J,B,T,R2,A
105       FORMAT(1X,I4,2X,F7.3,2X,F7.2,2X,F7.3,2X,F7.4)
6         CONTINUE
          CLOSE(UNIT=2)
C   CLOSE(UNIT=3)
          GO TO 3
999       TYPE 106
106       FORMAT(' ERROR:')
          CLOSE(UNIT=2)
          GO TO 3
900       STOP
          END
```

REFERENCES

Breiman, L. (1973). Statistics: with a view towards applications. Boston, Houghton Mifflin.

Carnahan, B., Luther, H.A. and Wilkes, J.O. (1969). Applied numerical methods. John Wiley and Sons.

Jenkins, G.M. and Watts, D.G. (1968). Spectral analysis and its applications. San Francisco Holden-Day.

ACKNOWLEDGMENTS

The testing of the programs presented in this report was carried out on the LaTrobe University VAX-780. Access to this computer was provided by the Department of Theoretical and Space Physics, LaTrobe University. I am grateful to Professor K.D. Cole of that Department for helpful discussions on this work.