Dr-180

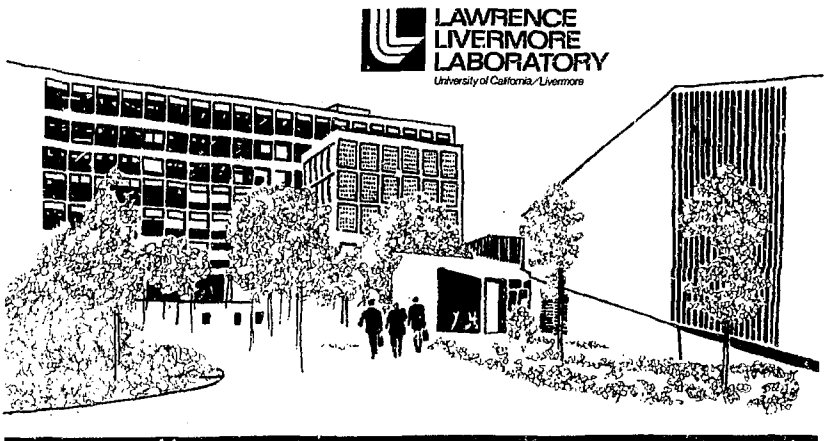
UCRL-51887

MAXIMIZATION OF ENERGY IN THE OUTPUT OF A LINEAR SYSTEM

D. G. Dudley

April 15, 1976

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48



Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy $; Microfiche $2.25

Page Range	Domestic Price	Page Range	Domestic Price
001–025	$ 3.50	326–350	10.00
026–050	4.00	351–375	10.50
051–075	4.50	376–400	10.75
076–100	5.00	401–425	11.00
101–125	5.25	426–450	11.75
126–150	5.50	451–475	12.00
151–175	6.00	476–500	12.50
176–200	7.50	501–525	12.75
201–225	7.75	526–550	13.00
226–250	8.00	551–575	13.50
251–275	9.00	576–600	13.75
276–300	9.25	601–up	*
301–325	9.75		

*Add $2.50 for each additional 100 page increment from 601 to 1,000 pages: add $4.50 for each additional 100 page increment over 1,000 pages.

Distribution Category
UC-32

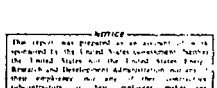
LAWRENCE LIVERMORE LABORATORY
University of California Livermore, California 94550

UCRL-51887

MAXIMIZATION OF ENERGY IN THE OUTPUT OF A LINEAR SYSTEM

D. G. Dudley

MS. date: April 15, 1976



Foreword

D. G. Dudley prepared this report while working as a summer (197?) employee with the Electromagnetic and Systems Research Group of the Electronics Engineering Department. The work was supported by the Defense Nuclear Agency. Dudley is a professor of electrical engineering at the University of Arizona, Tucson.

MAXIMIZATION OF ENERGY IN THE OUTPUT OF A LINEAR SYSTEM

Abstract

In this report, we consider a time-limited signal which, when passed through a linear system, maximizes the total output energy. Previous work has shown that the solution is given by the eigenfunction associated with the maximum eigenvalue in a Hilbert-Schmidt integral equation. Analytical results are available for the case where the transfer function is a low-pass filter. This work is extended by obtaining a numerical solution to the integral equation which allows results for reasonably general transfer functions.

Introduction

Consider a linear system with input f(t) and output g(t). For f(t) time-limited [-T,T] and of specified energy, Chalk[1] has shown that the total energy in the output is maximized through a consideration of the linear integral equation

$$\lambda f(t) = \int_{-T}^{T} f(\tau)\, K(t - \tau)\, d\tau \ . \qquad (1)$$

The "optimal" f(t) is that particular eigenfunction f(t) associated with the maximum eigenvalue λ_1 obtained in the solution to Eq. (1). Chalk has obtained an analytical solution to Eq. (1) for the case where the transfer function H(ω) is the first-order Butterworth low-pass filter

$$H(\omega) = \frac{\alpha}{\alpha + i\omega} \ . \qquad (2)$$

Chalk also gives approximate solutions for the case where the transfer function is an ideal low-pass filter

$$H(\omega) = \begin{cases} 1 & |\omega| < \omega_0 \\ 0 & |\omega| > \omega_0 \end{cases} . \qquad (3)$$

Slepian and Pollak[2] have extended the results by noting that for the case of the ideal low-pass filter, Eq. (1) defines certain of the angular prolate spheroidal functions,[3] a fact which allows for expression of the ideal low-pass filter results in terms of known functions.

In this report, we take a different point of view, abandoning the search for further analytical solutions in favor of obtaining an efficient numerical solution to Eq. (1). Our reason is that our application is in electromagnetics where the transfer function is often known only in terms of discrete experimental data or discrete data from numerical solutions to the electromagnetic boundary value problem. The results herein are reasonably complete in one sense and only preliminary in another. We have produced an efficient numerical technique for the reduction of Eq. (1) and have compared the results favorably both with Chalk and with Slepian and Pollak. In addition we have produced optimum pulses for the important case of a single-resonance transfer function. On the other hand, the numerical technique affords access to a large range of transfer functions which we have not begun to consider and in this sense the results are only preliminary. We make some specific comments concerning this later in the report.

Background

Let the energy in the input f(t) to a linear system be given by

$$E_I = \int_{-T}^{T} f^2(t)\, dt . \qquad (4)$$

Let the energy in the output g(t) be given by

$$E_0 = \int_{-\infty}^{\infty} g^2(t)\, dt . \qquad (5)$$

By Parseval's theorem,[4] this result may be given in terms of the frequency spectrum of the output $G(\omega)$, viz:

$$E_0 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\omega)|^2 \, d\omega, \tag{6}$$

If $F(\omega)$ is the frequency spectrum of the input and $H(\omega)$ the system transfer function,

$$E_0 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)\ H(\omega)|^2 \, d\omega, \tag{7}$$

or, in terms of the Fourier transforms of F and F^*,

$$E_0 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(\omega)|^2 \int_{-T}^{T} f(\tau)\ e^{-i\omega\tau} \, d\tau \int_{-T}^{T} f(s) e^{i\omega x} dx \, d\omega. \tag{8}$$

An interchange of integration produces the result

$$E_0 = \int_{-T}^{T} \int_{-T}^{T} f(\tau)\ f(x)\ K(x - \tau) \, dx \, d\tau, \tag{9}$$

where

$$K(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(\omega)|^2 \, e^{i\omega t} \, d\omega. \tag{10}$$

We wish to find an $f_1(t)$, optimum in the sense that it maximizes the double integral in Eq. (9), subject to the constraint in Eq. (4). This problem is classic in the calculus of variations (see Appendix A) and has the following solution. The function $f_1(t)$ must satisfy the integral equation given by Eq. (1). Specifically,

$$\lambda_1 f_1(t) = \int_{-T}^{T} f_1(\tau)\ K(t - \tau) \, d\tau, \tag{11}$$

where f_1 is the eigenfunction corresponding to the maximum eigenvalue λ_1. Substitution of Eq. (11) into Eq. (9) yields

$$E_0 = \lambda_1 E_I. \tag{12}$$

A final comment is in order concerning the properties of the solutions to Eq. (1). We note that the expression for the kernal function in Eq. (10) may be written

$$K(t) = \frac{1}{\pi} \int_0^\infty |H(\omega)|^2 \cos \omega t \, d\omega, \tag{13}$$

where we have used the even property of $|H(\omega)|^2$. We therefore conclude that the kernal is symmetric, viz:

$$K(t) = K(-t). \tag{14}$$

The result is that the eigenfunctions obtained in the solution to Eq. (1) are orthogonal and corresponding eigenvalues are real.[5] In addition,

$$\lambda_1 > \lambda_2 > \lambda_3 > \ldots \tag{15}$$

so that we may legitimately seek the maximum eigenvalue.

Normalization of Results

In order to produce a convenient normalization of the results to follow, we consider the all-pass transfer function

$$|H(\omega)| = 1. \tag{16}$$

From the Fourier transform pair

$$\delta(t) \longleftrightarrow 1, \tag{17}$$

we obtain in Eq. (1)

$$\lambda f(t) = \int_{-T}^{T} f(\tau) \, \delta(t - \tau) \, d\tau \tag{18}$$

with the solution

$$\lambda = 1, \tag{19}$$

We shall therefore adopt the normalization in general that

$$\max|H(\omega)| = 1. \tag{20}$$

This means that

$$E_0 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)\ H(\omega)|^2\ d\omega \ \leq\ \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2\ d\omega = E_I \tag{21}$$

so that, with our normalization,

$$\lambda_1 \leq 1. \tag{22}$$

Numerical Reduction of Integral Equation

To obtain a numerical solution to the integral equation in Eq. (1), we utilize the method of moments.[6] We expand f(t) in a sequence of pulse functions

$$f(t) = \sum_{n=-N}^{N} a_n p_n(t), \tag{23}$$

where

$$p_n(t) = \begin{cases} 1 & n\Delta - \frac{1}{2} < t < n\Delta + \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

and

$$\Delta = \frac{T}{N + \frac{1}{2}}. \tag{25}$$

Substitution into Eq. (1) yields

$$\lambda \sum a_n p_n(t) = \sum a_n q_n(t), \tag{26}$$

where

$$q_n(t) = \int_{n\Delta - \frac{\Delta}{2}}^{n\Delta + \frac{\Delta}{2}} K(t - \tau)\, d\tau \ . \tag{27}$$

We next multiply both sides by $\delta(t - m\Delta)$ and integrate $(-T,T)$ to get

$$\lambda \sum a_n \delta_{mn} = \sum a_n q_n(m\Delta). \tag{28}$$

We recognize Eq. (28) as a matrix equation

$$[\ell_{mn}][a_n] = \lambda[a_n], \tag{29}$$

where $[\ell_{mn}]$ is a square symmetric matrix with elements given by

$$\ell_{mn} = \int_{(|n-m| - \frac{1}{2})\Delta}^{(|n-m| + \frac{1}{2})\Delta} K(x)dx \quad . \tag{30}$$

The matrix given by Eq. (29) can be solved for its eigenvalues and associated eigenfunctions by utilizing a series of subroutines contained in a package called Eispack.[7] The appropriate subroutines and their function for this problem are as follows:

- TRED1 - transforms the real symmetric matrix in Eq. (30) to tridiagonal symmetric.
- BSTURM - calculates an interval containing a specified number of largest eigenvalues.
- TSTURM - finds the eigenvalues in the interval calculated by BSTURM and the associated eigenfunctions of the tridiagonal matrix.
- TRBAK1 - back transforms the eigenfunctions of the tridiagonal matrix to find the eigenfunctions of the real symmetric matrix.

A computer program has been written to perform the numerical reduction of the integral equation in Eq. (1). The program can operate in one of two modes. In the first mode, the program accepts numerical data representin a transfer function $H(\omega)$ and performs the inverse Fourier transform indicated in Eq. (13) to produce the kernal function $K(t)$. The cosine form of the transform in Eq. (13) differs only by a multiplicative factor of two from the inverse transform applicable to a causal real function. We therefore use subroutine LFILON[8] to obtain this result. Once $K(t)$ has been calculated, the square matrix $[\ell_{mn}]$ is obtained by numerical quadrature. The above subroutines from EISPACK are then employed to give the required maximum eigenvalue and associated eigenfunction. In the second mode, the program depends on the user supplying analytical expressions for $H(\omega)$ and for $K(t)$ in two function subprograms. In this case the necessity of performing a numerical inverse transform is eliminated. Beyond this difference, the remainder of the program functions as in the first mode. A listing of the program is included in Appendix B.

Numerical Results

The program described above was first used to perform checks against the results of Chalk (Ref. 1, p. 89) and Slepian and Pollak (Ref. 2, pp. 47-51). In all cases, the results duplicated theirs and will not be repeated here. Slepian's and Pollak's results are reproduced in a well-known textbook by Papoulis (Ref. 4, p. 69)]. After these checks were performed, we considered the case of a single resonance transfer function, given by

$$H(\omega) = \frac{2\alpha i\omega}{(\alpha + i\omega)^2 + \beta^2} \quad . \tag{31}$$

From Eq. (31), the kernal function $K(t)$ may be found analytically by closing the integral given in Eq. (10) along a semicircle through the top half of the complex ω-plane and using the calculus of residues. This procedure gives the result for $t > 0$ only; since $K(t)$ has been shown to be symmetric, however, this is sufficient. The result is

$$K(t) = \alpha e^{-\alpha|t|}\left(\cos\beta|t| - \frac{\alpha}{\beta}\sin\beta|t|\right) , \tag{32}$$

where

$$\beta = (\omega_0^2 - \alpha^2)^{\frac{1}{2}} \quad , \quad \omega_0 > \alpha \quad . \tag{33}$$

Since the analytical expressions are available in this case, the program operates in the second mode.

In Fig. 1, we show the magnitude of the transfer function in decibels for the single resonant case. We selected parameter values of $\alpha = 9.16 \times 10^6$ and $f_0 = 48.6$ MHz.

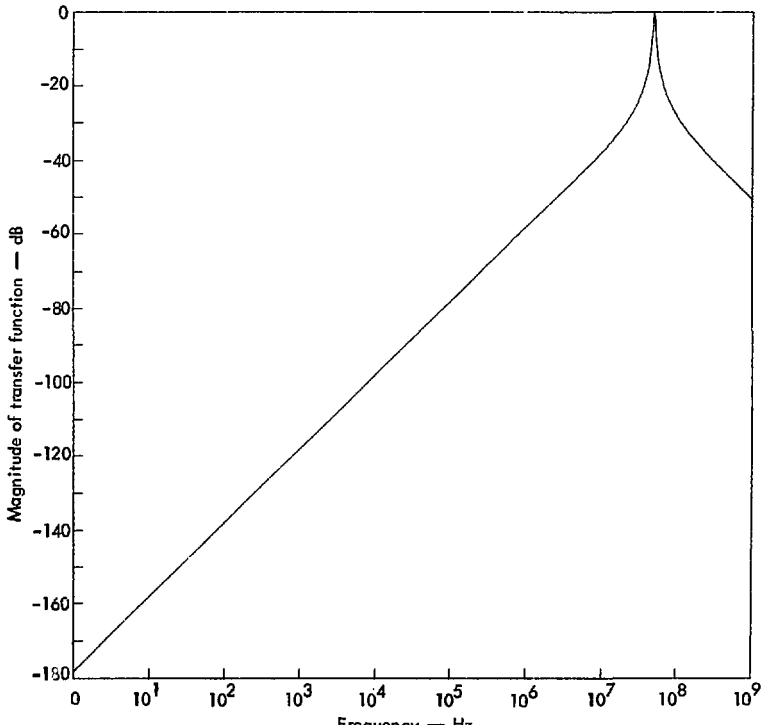


Fig. 1. Magnitude of the transfer function in decibels for the single resonant case.

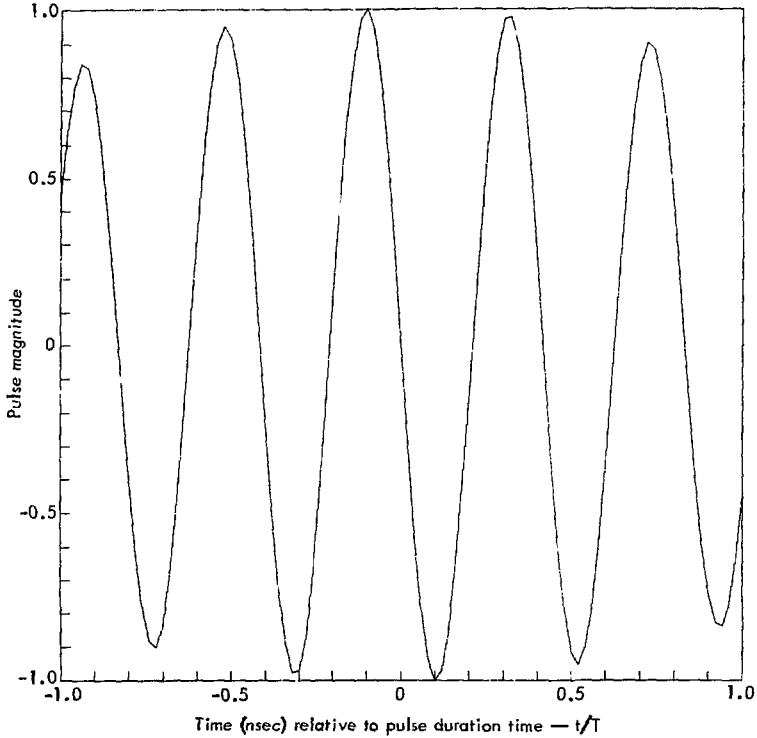


Fig. 2. Optimum pulse for the case where T = 50 nsec. Maximum eigenvalue = 0.35.

The reason for the choice was to attempt to match some numerical transfer function data for a long, narrow slot in an infinite ground plane. The resulting optimum pulse for the case T = 50 nsec (Fig. 2) is a modulated signal whose period corresponds to the frequency of the resonance of the transfer function. When T = 100 nsec (Fig. 3), the period of the ripple stays the same but the decay of the envelope is more pronounced. In

addition the eigenvalue has increased from 0.35 to 0.54 which indicates a strong increase in energy transfer efficiency. When T is increased to 200 nsec (Fig. 4) and then to 300 nsec (Fig. 5), the modulation frequency remains the same but the envelope decay continues to increase. In addition, the eigenvalue increases to 0.68 and then to 0.74.

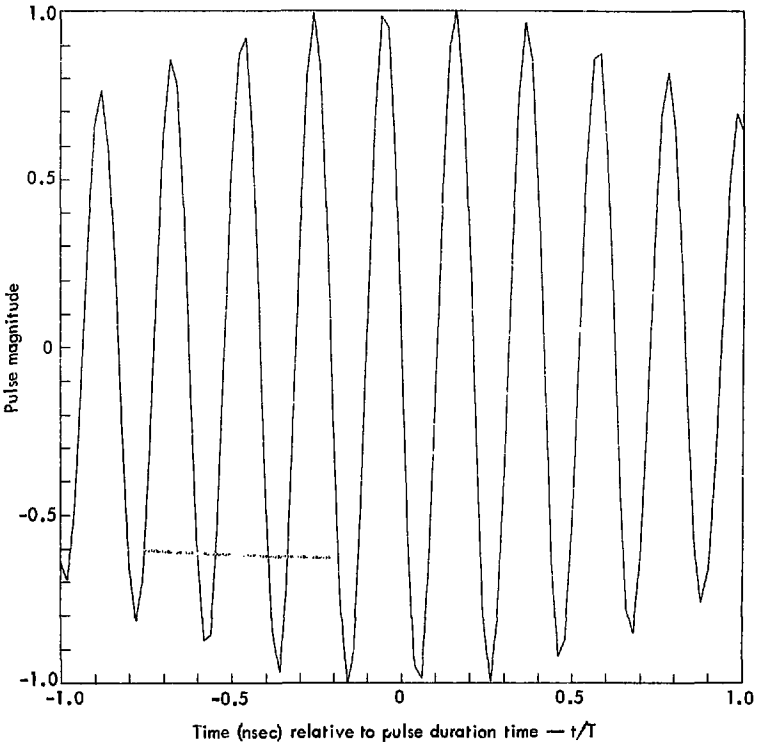


Fig. 3. Optimum pulse for the case where T = 100 nsec. Maximum eigenvalue = 0.54.

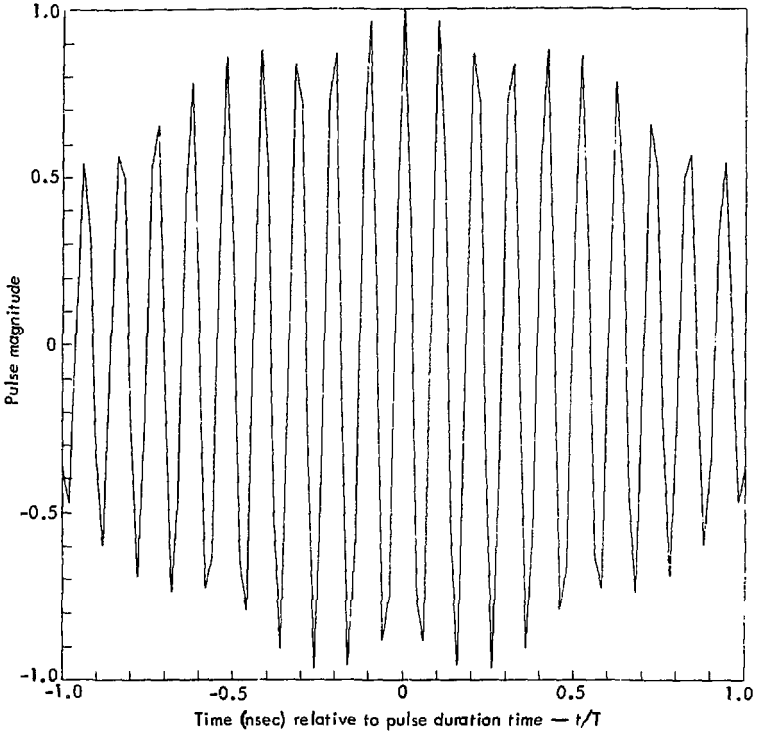


Fig. 4. Optimum pulse for the case where T = 200 nsec. Maximum eigenvalue = 0.68

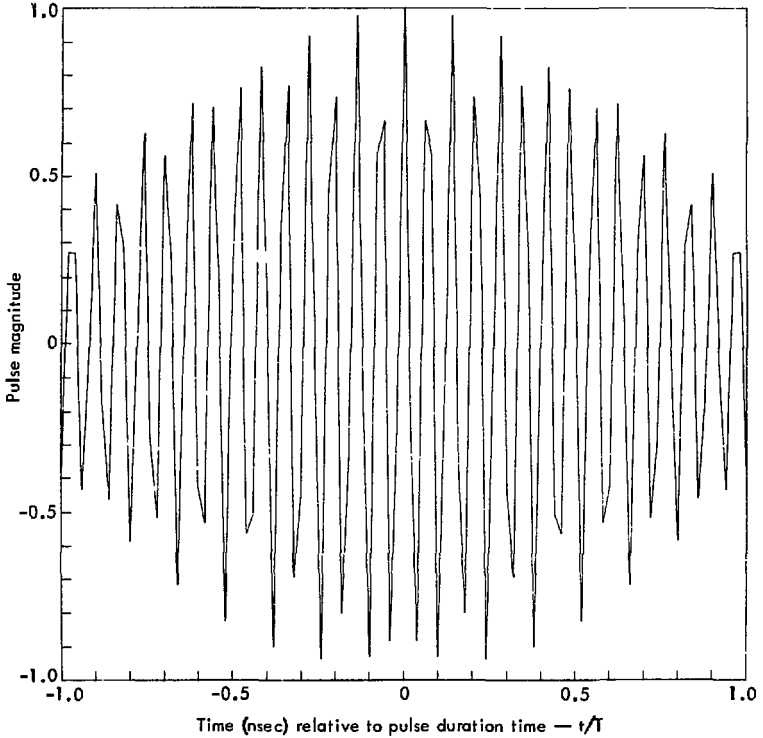


Fig. 5. Optimum pulse for the case where T = 300 nsec. Maximum eigenvalue = 0.74.

Conclusions and Recommendations for Further Work

There are few surprises in the numerical results for the single resonance case. The optimum pulse contains a modulation rate which favors the frequency at which the transfer function has its peak. In addition the pulse becomes more efficient as the length of time (-T,T) of application

of the incident pulse is increased. The shape of the envelope is qualitatively similar to the shape of the entire pulse for the low-pass filter investigated by Chalk and by Slepian and Pollak. This is in keeping with the ideas common in equivalent low-pass filter analysis of band-pass filters (Ref. 4, p. 122). (Here, we regard the single-resonant model as a band-pass filter.)

We noted earlier that the results are only preliminary. However, now that the computer program has been written, a large range of transfer functions can be investigated. Such an investigation should produce some trends in the transfer of energy which should make possible some generalizations concerning the process.

In addition, there seems to be no fundamental limitation on the infinite limits in the integral of Eq. (6). If the limits were finite, the analysis would proceed as before with the exception that the kernal function in Eq. (10) would be calculated from an integral over finite limits. This is an important observation because it raises the possibility of maximizing the energy in a finite bandwidth in the output to the linear system. Fortunately, if we agree to add the energy contribution in the finite bandwidth (ω_1, ω_2) and the energy in the same band over the mirror negative frequencies, the symmetry of the kernal function is not destroyed and we may proceed with the numerical reduction as before.

References

1. J. H. H. Chalk, "The Optimum Pulse-Shape for Pulse Communication," *Proc. Inst. Elec. Engrs. (London)* 87, 88 (1950).
2. D. Slepian and H. O. Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty – I," *Bell System Tech. J.* 40 (1), 43 (1961).
3. C. Flammer, *Spheroidal Wave Functions*, (Stanford University Press, Stanford, 1957).
4. A. Papoulis, *The Fourier Integral and its Applications* (McGraw-Hill Co., New York, 1962), p. 27.
5. G. Arfken, *Mathematical Methods for Physicists* (Academic Press, New York, 1966), pp. 593-597.
6. R. F. Harrington, *Field Computations by Moment Methods* (Macmillan, New York, 1968).
7. R. P. Dickinson, F. N. Fritsch, R. F. Hausman, and R. F. Sincovec, *EISPACK User's Guide*, Lawrence Livermore Laboratory, Rept. UCID-30077 (1973).
8. D. G. Dudley, *Numerical Inversion of the Fourier Transform, A Combination Trapezoidal and Filon Technique*, Lawrence Livermore Laboratory, Rept. UCRL-51878 (1975).

Appendix A – Maximization with a Constraint

We wish to maximize the double integral

$$E_0 = \int_{-T}^{T} f(\tau) \int_{-T}^{T} f(x)\, K(|x - \tau|)\, dx\, d\tau \tag{A-1}$$

subject to the constraint

$$E_I = \int_{-T}^{T} f^2(t)\, dt\ . \tag{A-2}$$

We form the functional E(f)

$$E = E_0 - \lambda E_I\ , \tag{A-3}$$

where λ is a Lagrange multiplier. We therefore have

$$E = \int_{-T}^{T} f(\tau) \int_{-T}^{T} f(x)\, K(|x - \tau|)\, dx d\tau - \lambda \int_{-T}^{T} f^2(t)\, dt. \tag{A-4}$$

Let f take on a variation δ so that

$$E(f + \alpha\delta) = \int_{-T}^{T} [f(\tau) + \alpha\delta(\tau)] \int_{-T}^{T} [f(x) + \alpha\delta(x)]\, K(|x - \tau|)\, dx\, d\tau$$

$$- \alpha \int_{-T}^{T} [f(x) + \alpha\delta(x)]^2\, dx\ . \tag{A-5}$$

Expanding and collecting terms, we obtain

$$E(f + \alpha\delta) = E(f) + \partial\alpha G + \alpha^2 H - \partial\alpha\lambda I - \alpha^2\lambda J\ , \tag{A-6}$$

where

$$G = \int_{-T}^{T} \delta(\tau) \int_{-T}^{T} f(x)\, K(|x - \tau|)\, dx\, d\tau \tag{A-7}$$

$$H = \int_{-T}^{T} \delta(\tau) \int_{-T}^{T} \delta(x)\, K(|x - \tau|)\, dx\, d\tau \qquad \text{(A-8)}$$

$$I = \int_{-T}^{T} f(x)\, \delta(x)\, dx \qquad \text{(A-9)}$$

$$J = \int_{-T}^{T} \delta^2(x)\, dx. \qquad \text{(A-10)}$$

To maximize (or minimize), take

$$\frac{\partial}{\partial \alpha} [E(f + \alpha\delta)]\Big|_{\alpha = 0} = 0 \qquad \text{(A-11)}$$

with the result

$$G = \lambda I\ , \qquad \text{(A-12)}$$

or,

$$\int_{-T}^{T} \delta(\tau) \int_{-T}^{T} f(x)\, K(|x - \tau|)\, dx\, d\tau = \lambda \int_{-T}^{T} \delta(\tau)\ . \qquad \text{(A-13)}$$

This may be written

$$\int_{-T}^{T} \delta(\tau) \left\{ \left[\int_{-T}^{T} f(x)\, K(|x - \tau|)\, dx \right] - \lambda f(\tau) \right\} d\tau = 0 \qquad \text{(A-14)}$$

and since δ is arbitrary

$$\lambda f(\tau) = \int_{-T}^{T} f(x)\, K(|x - \tau|)\, dx\ , \qquad \text{(A-15)}$$

which is a verification of Eq. (11).

Appendix B - Computer Program Listing

```
      PROGRAM LOPAS (INPUT,DUDPUT,TAPE5=INPUT,TAPE6=DUDPUT)
C   THE PURPOSE OF PROGRAM LOPAS IS TO CALCULATE A TIME - LIMITED INPUT
C   PULSE TO A GIVEN TRANSFER FUNCTION WHICH IS OPTIMUM IN THE SENSE
C   THAT FOR A SPECIFIED ENERGY IN THE INPUT IT MAXIMIZES THE TOTAL
C   ENERGY IN THE OUTPUT.  THE PROPGRAM OPERATES IN TWO MODE OPTIONS
C   DEPENDING ON WHETHER THE TRANSFER FUNCTION IS SUPPLIED ANALYTICALLY
C   OR NUMERICALLY BY THE USER.  THE READ-IN SEQUENCE IS AS FOLLOWS.
C   FOR BOTH MODES OF OPERATION, THE FIRST DATA CARD IS READ IN BY
C
C        READ (5,60) MAX,IREAD,WT
C     60 FORMAT (2I10,F10.0)
C
C   WHERE
C
C        MAX - THE SIZE OF THE MATRIX WHOSE EIGENFUNCTIONS AND EIGEN-
C              VALUES ARE TO BE DETERMINED.  THE PROGRAM CALCULATES THE
C              TWO HIGHEST EIGENVALUES AND CORRESPONDING EIGENVECTORS.
C              THE HIGHEST EIGENVALUE IS THE EFFICIENCY OF ENERGY TRANS-
C              FER AND THE CORRESPONDING EIGENVECTOR IS THE OPTIMUM
C              PULSE
C
C       IREAD - THE MODE NUMBER.  IF IREAD = 0, TRANSFER FUNCTION DATA
C              IS SUPPLIED ANALYTICALLY IN TWO FUNCTION SUBPROGRAMS
C              WHICH MUST BE SUPPLIED BY THE USER.  FUNCTION FCTF(X)
C              MUST CALCULATE AND RETURN THE MAGNITUDE-SQUARED OF THE
C              TRANSFER FUNCTION AS A FUNCTION OF THE RADIAN FREQUENCY.
C              FUNCTION FCTT(X) MUST CALCULATE AND RETURN THE INVERSE
C              FOURIER TRANSFORM OF THE TRANSFER FUNCTION MAGNITUDE
C              SQUARED.  IF IREAD = 1, TRANSFER FUNCTION DATA IS SUP-
C              PLIED NUMERICALLY BY THE USER AND THE INVERSE TRANSFORM
C              OF THE MAGNITUDE SQUARED IS SUBSEQUENTLY CALCULATED BY
C              THE PROGRAM THROUGH SUBROUTINE LFILON.
C
C          WT - THE EXTENT OF THE TIME-LIMITED PULSE IS PLUS OR MINUS WT.
C
C   THE ABOVE DATA CARD IS COMMON TO BOTH MODE 0 AND MODE 1.  THE
C   REMAINING CARDS FOR MODE 0 ARE AS FOLLOWS.
C
C        READ (5,50) NDEC
C        READ (5,260) (THS(KK),KK=1,NDEC)
C     50 FORMAT (I10)
C    260 FORMAT (15I5)
C
C   WHERE THE FREQUENCIES ARE CALCULATED IN DECADE BLOCKS AND
C
C       NDEC - THE NUMBER OF FREQUENCY DECADES
C
C     THS(KK) - AN ARRAY CONTAINING THE NUMBER OF POINTS IN EACH DECADE.
C
C   FOR MODE 1, THE REMAINING DATA CARDS ARE READ IN AS FOLLOWS.
C
C        READ (5,240) NFP,H2
C        READ (5,210) (H(K),F(K),K=1,NFP)
C    240 FORMAT (I10,5X,2E15.6)
C    210 FORMAT (2E15.6,10X,E15.6)
C
C   WHERE
C
C        NFP - THE NUMBER OF FREQUENCY POINTS AT WHICH DATA IS SUPPLIED
C
C         H2 - THE TRANSFER FUNCTION (COMPLEX) AT ZERO FREQUENCY
C
C       H(K) - THE TRANSFER FUNCTION (COMPLEX) AT EACH FREQUENCY POINT
C
```

```
C       F(K) - THE FREQUENCY POINTS
C
C   ------------------------------------------------------------------
C
      CALL DEVICE (6HCREATE,6HOUTPUT,50000)
      INTEGER THS
      COMPLEX I,B,H,HZ
      COMMON    A(101,101)    ,E(101) W(2),Z(101,2),R1(101),R2(101),
     1 R3(101),R4(101),R5(101),R6(101),E2(101),EMIN(2),EMAX(2),D(101)
     2 ,U(101),V(101),VOFF(100),T( 700),FT( 700),FTA( 700),IND ( 700),
     3 F(500),H (500),B(500),HMD(500),HT( 700),THS(15),PLD(5,101)
      READ (5,60) MAX,IREAD,WT
      NPTS = 5
      MXL = (MAX - 1) / 2
      DEL = WT / (MXL + .5)
      WRITE (6,20) MAX,WT,NPTS
      XINC = DEL / (2. * (NPTS - 1))
      ITF = 0
      NT = (2 * MAX - 1) * (NPTS - 1) + 1
      DO 131 N = 1,NT
  131 T(N) = (N - 1) * XINC
      WRITE (6,110) T(NT),XINC
      TZ = 0.
      IF (IREAD .EQ. 1) GO TO 201
      WRITE (6,140)
      READ (5,50) NDEC
      READ (5,260) (THS(KK),KK=1,NDEC)
      PI = 3.1415926
      K = 0
      HZ = CMPLX(FCTF(0.),0.)
      DO 211 KK = 1,NDEC
      KL = KK - 1
      COU = 10 ** KL
      XI = 9. / THS(KK)
      NMX = THS(KK)
      DO 241 L = 1,NMX
      NL = L - 1
      K = K + 1
      F(K) = COU * (1. + NL * XI)
      W = 2. * PI * F(K)
      H(K) = CMPLX(FCTF(W),0.)
      HZ = CMPLX(FCTF(0.),0.)
      HMD(K) = 10. * ALOG10(CABS(H(K)))
  241 CONTINUE
  211 CONTINUE
      NFP = K
      DO 261 K = 1,NT
      FT (K) = FCTT(T(K))
  261 CONTINUE
      GO TO 202
  201 CONTINUE
      WRITE (6,150)
      READ (5,240) NFP,HZ
      READ (5,210) (H(K),F(K),K=1,NFP)
      HZ = HZ * CONJG(HZ)
      DO 221 K = 1,NFP
      H(K) = H(K) * CONJG(H(K))
      HMD(K) = 10. * ALOG10(CABS(H(K)))
  221 CONTINUE
      NTM = NT - 1
      DO 281 N = 1,NTM
      NP = N + 1
  281 HT(N) = T(NP)
      TZ = 0.
```

```
      CALL LFILON (TZ,HT,FT,FTA,NTM,F,HZ,H,NFP,8,IND,ITF)
      DO 231 K = 1,NTM
      KR = NT - K + 1
      KRL = KR - 1
  231 FT(KR) = F(KRL) / 2.
      NFPM = NFP - 1
      FT(1) = HZ * F(1) + H(1) * F(2) + H(NFP) * (F(NFP) - F(NFPM))
      DO 271 K = 2,NFPM
      KP = K + 1
      KM = K - 1
      FT(1) = FT(1) + H(K) * (F(KP) - F(KM))
  271 CONTINUE
  202 CONTINUE
      WRITE (6,250) HZ
      DO 61 K = 1,NFP
   61 WRITE (6,230) F(K),H(K)
      WRITE (6,120)
      WRITE (6,130) (T(K),FT(K),K=1,NT)
      CALL QSF (XINC,FT,V,NPTS)
      VZ = V(NPTS)
      NPTS2 = 2 * NPTS - 1
      MAXL = MAX - 1
      DO 111 M = 1,MAXL
      MS = (2 * M - 1) * NPTS - 2 * M + 2
      DO 121 N = 1,NPTS2
      NA = N - 1 + MS
  121 U(N) = FT(NA)
      CALL QSF (XINC,U,V,NPTS2)
      VOFF(M) = V(NPTS2)
  111 CONTINUE
      DO 81 M = 1,MAX
   81 A(M,M) = 2. * VZ
      DO 11 M = 1,MAXL
      MP = M + 1
      DO 21 N = MP,MAX
      NL = N - M
   21 A(N,NL) = VOFF(M)
   11 CONTINUE
      CALL TRED1 (101,MAX,A,D,E,E2)
      IFLAG = 3
      CALL BSTURM (MAX,2,IC1,IFLAG,XL,XU,D,E,E2
      IF (IFLAG .EQ. -1) WRITE (6,10)
      XXQ = -1.
      CALL TSTURM (101,MAX,XXQ,D,E,E2,XL,XU,IC1,ME,W,Z,IERR,R1,R2,R3,R4,
     1 R5,R6)
      IF (IERR .NE. 0) WRITE (6,30) IERR
      CALL TRBAK1 (101,MAX,A,E,ME,Z)
      DO 51 N = 1,MAX
      R1(N) = Z(N,1)
   51 R2(N) = Z(N,2)
      CALL AMNMX (MAX,R1,EMIN(1),EMAX(1))
      CALL AMNMX (MAX,R2,EMIN(2),EMAX(2))
      WRITE (6,80)
      DO 31 M = 1,ME
      IF (ABS(EMIN(M)) .GT. ABS(EMAX(M))) GO TO 2
      GO TO 1
    2 STO = EMAX(M)
      EMAX(M) = EMIN(M)
      EMIN(M) = STO
    1 CONTINUE
      WRITE (6,90) W(M)
      DO 41 N = 1,MAX
      Z(N,M) = Z(N,M) / EMAX(M)
   41 WRITE (6,40) Z(N,M)
```

```
193     31 CONTINUE
194        DO 71 N = 1,MAX
195     71 PLO(1,N) = Z(N,2)
196        WRITE (6,100) W(2)
197        CALL PLOTS (PLO,1,MAX,999)
198     10 FORMAT (//* WARNING -- ERRORS IN BSTURM INPUT*//)
199     20 FORMAT (1H1,* ENERGY MAXIMIZATION*//* MATRIX SIZE =*
200       1I6/* TIME WINDOW (NS) =*E14.3/* NO. OF INTEGRATION PTS = *
201       2I5//)
202     30 FORMAT (//* WARNING -- IERR = *I5//)
203     40 FORMAT (8X,E15.5)
204     50 FORMAT (1I0)
205     60 FORMAT (2I10,F10.0)
206     80 FORMAT (,/* TWO HIGHEST EIGENVALIES WITH ASSOCIATED EIGENVECTORS*
207       1)
208     90 FORMAT (//* EIGENVALUE = *E15.5//12X*EIGENVECTOR*//)
209    100 FORMAT (////////* THE FOLLOWING PLOT IS OF THE EIGENVECTOR ASSOCIA
210       1TED WITH THE MAXIMUM EIGENVALUE LAMBDA = *E11.3//)
211    110 FORMAT (* TIME GOES FROM 0. TO *E10.3* SECONDS IN *E10.3* SECOND
212       1INCREMENTS*//)
213    120 FORMAT (//* INVERSE TRANSFORM OF H(F) MAG SQUARED*//5X*T(K)*13X
214       1*FT(K)*//)
215    130 FORMAT (2E15.5)
216    140 FORMAT (* INVERSE TRANSFORM IS PERFORMED ANALYTICALLY BY USER*//)
217    150 FORMAT (* INVERSE TRANSFORM IS SUPPLIED NUMERICALLY BY USER*//)
218    210 FORMAT (2E15.6,10X,E15.5)
219    230 FORMAT (6E15.5)
220    240 FORMAT (1I0,5X,2E15.6)
221    250 FORMAT (* TRANSFER FUNCTION MAGNITUDE SQUARED*//4X*FREQUENCY*16X
222       1*H(F)*//4X*0.*9X,2E15.5)
223    260 FORMAT (1515)
224        CALL EXIT
225        END
```

```
1       SUBROUTINE AMNMX (MAX,A,AMN,AMX)
2       DIMENSION A(1)
3       AMN = A(1)
4       AMX = AMN
5       DO 11 M = 2,MAX
6       IF (A(M) .GT. AMX) AMX = A(M)
7       IF (A(M) .LT. AMN) AMN = A(M)
8    11 CONTINUE
9       RETURN
10      END
```

```
1       FUNCTION FCTT(X)
2       F = 4.857E7
3       W = 2. * 3.1415926 * F
4       A = .03 * W
5       B = SQRT(W * W - A * A)
6       FCTT = A * EXP(-A * X) * (COS(B * X) - A * SIN(B * X) / B)
7       RETURN
8       END
```

```
      FUNCTION FCTF(X)
      COMPLEX I,Z
      F = 4.857E7
      W = 2. * 3.1415926 * F
      A = .03 * W
      I = CMPLX(0.,1.)
      B = SQRT(W * W - A * A)
      Z = 2. * A * I * X / ((A * I * X) * (A * I * X) + B * B)
      FCTF = REAL(Z * CONJG(Z))
      RETURN
      END
```

```
C                                                                      93215001
C     ------------------------------------------------------------------93215002
C                                                                      93215003
      SUBROUTINE TSTURM(NM,N,EPS1,D,E,E2,LB,UB,MM,M,W,Z,               93215004
     X                  IERR,RV1,RV2,RV3,RV4,RV5,RV6)                  93215005
C*NMS:MATHLIB:NATS:EISPACK:SOURCE:TSTURM          TSTURM E2.V2 11/28/73 LLL2
C                                                                      93215006
      INTEGER I,J,K,M,N,P,Q,R,S,II,IP,JJ,MM,M1,M2,NM,ITS,              93215007
     X        IERR,GROUP,ISTURM                                        93215008
      REAL D(N),E(N),E2(N),W(MM),Z(NM,MM),                             93215009
     X     RV1(N),RV2(N),RV3(N),RV4(N),RV5(N),RV6(N)                   93215010
      REAL U,V,LB,T1,T2,UB,UK,XU,X0,X1,EPS1,EPS2,EPS3,EPS4,            93215011
     X     NORM,MACHEP                                                 93215012
      REAL SQRT,ABS,AMAX1,AMIN1,FLOAT                                  LLL25013
C                                                                      93215014
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TRISTURM  93215015
C     BY PETERS AND WILKINSON.                                         93215016
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 418-439(1971).  93215017
C                                                                      93215018
C     THIS SUBROUTINE FINDS THOSE EIGENVALUES OF A TRIDIAGONAL         93215019
C     SYMMETRIC MATRIX WHICH LIE IN A SPECIFIED INTERVAL AND THEIR     93215020
C     ASSOCIATED EIGENVECTORS, USING BISECTION AND INVERSE ITERATION.  93215021
C                                                                      93215022
C     ON INPUT-                                                        93215023
C                                                                      93215024
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL        93215025
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM         93215026
C          DIMENSION STATEMENT,                                        93215027
C                                                                      93215028
C        N IS THE ORDER OF THE MATRIX,                                 93215029
C                                                                      93215030
C        EPS1 IS AN ABSOLUTE ERROR TOLERANCE FOR THE COMPUTED          93215031
C          EIGENVALUES.  IT SHOULD BE CHOSEN COMMENSURATE WITH         93215032
C          RELATIVE PERTURBATIONS IN THE MATRIX ELEMENTS OF THE        93215033
C          ORDER OF THE RELATIVE MACHINE PRECISION.  IF THE            93215034
C          INPUT EPS1 IS NON-POSITIVE, IT IS RESET FOR EACH            93215035
C          SUBMATRIX TO A DEFAULT VALUE, NAMELY, MINUS THE             93215036
C          PRODUCT OF THE RELATIVE MACHINE PRECISION AND THE           93215037
C          1-NORM OF THE SUBMATRIX,                                    93215038
C                                                                      93215039
C        D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX,         93215040
C                                                                      93215041
C        E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE INPUT MATRIX       93215042
C          IN ITS LAST N-1 POSITIONS.  E(1) IS ARBITRARY,              93215043
C                                                                      93215044
C        E2 CONTAINS THE SQUARES OF THE CORRESPONDING ELEMENTS OF E.   93215045
C          E2(1) IS ARBITRARY,                                         93215046
C                                                                      93215047
C        LB AND UB DEFINE THE INTERVAL TO BE SEARCHED FOR EIGENVALUES. 93215048
C          IF LB IS NOT LESS THAN UB, NO EIGENVALUES WILL BE FOUND,    93215049
C                                                                      93215050
C        MM SHOULD BE SET TO AN UPPER BOUND FOR THE NUMBER OF          93215051
C          EIGENVALUES IN THE INTERVAL.  WARNING- IF MORE THAN         93215052
C          MM EIGENVALUES ARE DETERMINED TO LIE IN THE INTERVAL,       93215053
C          AN ERROR RETURN IS MADE WITH NO VALUES OR VECTORS FOUND.    93215054
C                                                                      93215055
C     ON OUTPUT-                                                       93215056
C                                                                      93215057
C        EPS1 IS UNALTERED UNLESS IT HAS BEEN RESET TO ITS             93215058
C          (LAST) DEFAULT VALUE,                                       93215059
C                                                                      93215060
C        D AND E ARE UNALTERED,                                        93215061
C                                                                      93215062
C        ELEMENTS OF E2, CORRESPONDING TO ELEMENTS OF E REGARDED       93215063
```

```
C          AS NEGLIGIBLE, HAVE BEEN REPLACED BY ZERO CAUSING THE          93215064
C          MATRIX TO SPLIT INTO A DIRECT SUM OF SUBMATRICES.              93215065
C          E2(1) IS ALSO SET TO ZERO.                                     93215066
C                                                                         93215067
C        M IS THE NUMBER OF EIGENVALUES DETERMINED TO LIE IN (LB,UB).     93215068
C                                                                         93215069
C        W CONTAINS THE M EIGENVALUES IN ASCENDING ORDER IF THE MATRIX    93215070
C          DOES NOT SPLIT.  IF THE MATRIX SPLITS, THE EIGENVALUES ARE     93215071
C          IN ASCENDING ORDER FOR EACH SUBMATRIX.  IF A VECTOR ERROR      93215072
C          EXIT IS MADE, W CONTAINS THOSE VALUES ALREADY FOUND.           93215073
C                                                                         93215074
C        Z CONTAINS THE ASSOCIATED SET OF ORTHONORMAL EIGENVECTORS.       93215075
C          IF AN ERROR EXIT IS MADE, Z CONTAINS THOSE VECTORS             93215076
C          ALREADY FOUND.                                                 93215077
C                                                                         93215078
C        IERR IS SET TO                                                   93215079
C          ZERO       FOR NORMAL RETURN.                                 93215080
C          3*N+1      IF M EXCEEDS MM.                                   93215081
C          4*N+R      IF THE EIGENVECTOR CORRESPONDING TO THE R-TH       93215082
C                     EIGENVALUE FAILS TO CONVERGE IN 5 ITERATIONS.      93215083
C                                                                         93215084
C        RV1, RV2, RV3, RV4, RV5, AND RV6 ARE TEMPORARY STORAGE ARRAYS.   93215085
C                                                                         93215086
C     THE ALGOL PROCEDURE STURMCNT CONTAINED IN TRISTURM                  93215087
C     APPEARS IN TSTURM IN-LINE.                                          93215088
C                                                                         93215089
C     NOTE THAT SUBROUTINE TQL2 OR IMTQL2 IS GENERALLY FASTER THAN        93215090
C     TSTURM, IF MORE THAN N/4 EIGENVALUES AND VECTORS ARE TO BE FOUND.   93215091
C                                                                         93215092
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,          93215093
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY           93215094
C                                                                         LLL1
C     LLL CONSULTANT...  R. P. DICKINSON                                  LLL1
C     VERSION 2 - CORRECT DECLARATIONS FOR PUTT                           LLL2
C                                                                         93215095
C     ------------------------------------------------------------------93215096
C                                                                         93215097
C     ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING       93215098
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.     93215099
C                                                                         93215100
C                **********                                               93215101
      MACHEP = 2.**(-47)                                                  93215102
C                                                                         93215103
      IERR = 0                                                            93215104
      T1 = LB                                                             93215105
      T2 = UB                                                             93215106
C     ********** LOOK FOR SMALL SUB-DIAGONAL ENTRIES **********           93215107
      DO 40 I = 1, N                                                      93215108
         IF (I .EQ. 1) GO TO 20                                           93215109
         IF (ABS(E(I)) .GT. MACHEP * (ABS(D(I)) + ABS(D(I-1))))           93215110
     X      GO TO 40                                                     93215111
   20    E2(I) = 0.0                                                     93215112
   40 CONTINUE                                                            93215113
C     ********** DETERMINE THE NUMBER OF EIGENVALUES                     93215114
C                IN THE INTERVAL **********                              93215115
      P = 1                                                               93215116
      Q = N                                                               93215117
      X1 = UB                                                             93215118
      ISTURM = 1                                                          93215119
      GO TO 320                                                           93215120
   60 M = S                                                               93215121
      X1 = LB                                                             93215122
      ISTURM = 2                                                          93215123
      GO TO 320                                                           93215124
```

```
   80 M = M - S                                                          93215125
      IF (M .GT. MM) GO TO 980                                           93215126
      Q = 0                                                              93215127
      R = 0                                                              93215128
C     ********** ESTABLISH AND PROCESS NEXT SUBMATRIX, REFINING          93215129
C                INTERVAL BY THE GERSCHGORIN BOUNDS **********          93215130
  100 IF (R .EQ. M) GO TO 1001                                           93215131
      P = Q + 1                                                          93215132
      XU = D(P)                                                          93215133
      X0 = D(P)                                                          93215134
      U = 0.0                                                            93215135
C                                                                        93215136
      DO 120 Q = P, N                                                    93215137
         X1 = U                                                          93215138
         U = 0.0                                                         93215139
         V = 0.0                                                         93215140
         IF (Q .EQ. N) GO TO 110                                         93215141
         U = ABS(E(Q+1))                                                 93215142
         V = E2(Q+1)                                                     93215143
  110    XU = AMIN1(D(Q)-(X1+U),XU)                                      93215144
         X0 = AMAX1(D(Q)+(X1+U),X0)                                      93215145
         IF (V .EQ. 0.0) GO TO 140                                       93215146
  120 CONTINUE                                                           93215147
C                                                                        93215148
  140 X1 = AMAX1(ABS(XU),ABS(X0)) * MACHEP                               93215149
      IF (EPS1 .LE. 0.0) EPS1 = -X1                                      93215150
      IF (P .NE. Q) GO TO 180                                            93215151
C     ********** CHECK FOR ISOLATED ROOT WITHIN INTERVAL **********      93215152
      IF (T1 .GT. D(P) .OR. D(P) .GE. T2) GO TO 940                      93215153
      R = R + 1                                                          93215154
C                                                                        93215155
      DO 160 I = 1, N                                                    93215156
  160 Z(I,R) = 0.0                                                       93215157
C                                                                        93215158
      W(R) = D(P)                                                        93215159
      Z(P,R) = 1.0                                                       93215160
      GO TO 940                                                          93215161
  180 X1 = X1 * FLOAT(Q-P+1)                                             93215162
      LB = AMAX1(T1,XU-X1)                                               93215163
      UB = AMIN1(T2,X0+X1)                                               93215164
      X1 = LB                                                            93215165
      ISTURM = 3                                                         93215166
      GO TO 320                                                          93215167
  200 M1 = S + 1                                                         93215168
      X1 = UB                                                            93215169
      ISTURM = 4                                                         93215170
      GO TO 320                                                          93215171
  220 M2 = S                                                             93215172
      IF (M1 .GT. M2) GO TO 940                                          93215173
C     ********** FIND ROOTS BY BISECTION **********                      93215174
      X0 = UB                                                            93215175
      ISTURM = 5                                                         93215176
C                                                                        93215177
      DO 240 I = M1, M2                                                  93215178
         RV5(I) = UB                                                     93215179
         RV4(I) = LB                                                     93215180
  240 CONTINUE                                                           93215181
C     ********** LOOP FOR K-TH EIGENVALUE                               93215182
C                FOR K=M2 STEP -1 UNTIL M1 DO --                         93215183
C                (-DO- NOT USED TO LEGALIZE COMPUTEE-GO-TO) **********   93215184
      K = M2                                                             93215185
  250    XU = LB                                                         93215186
C     ********** FOR I=K STEP -1 UNTIL M1 DO -- **********               93215187
         DO 260 II = M1, K                                               93215188
```

```
                I = NI + K - II                                          93215189
                IF (XU .GE. RV4(I)) GO TO 260                            93215190
                XU = RV4(I)                                              93215191
                GO TO 280                                                93215192
  260       CONTINUE                                                     93215193
C                                                                        93215194
  280       IF (XO .GT. RV5(K)) XO = RV5(K)                              93215195
C     ********** NEXT BISECTION STEP **********                         93215196
  300       XI = (XU + XO) * 0.5                                         93215197
            IF ((XO - XU) .LE. (2.0 * MACHEP *                           93215198
     X          (ABS(XU) + ABS(XO)) + ABS(EPS1))) GO TO 420              93215199
C     ********** IN-LINE PROCEDURE FOR STURM SEQUENCE **********         93215200
  320       S = P - 1                                                    93215201
            U = 1.0                                                      93215202
C                                                                        93215203
            DO 340 I = P, Q                                              93215204
               IF (U .NE. 0.0) GO TO 325                                 93215205
               V = ABS(E2(I)) / MACHEP                                   93215206
               GO TO 330                                                 93215207
  325          V = E2(I) / U                                             93215208
  330          U = D(I) - XI - V                                         93215209
               IF (U .LT. 0.0) S = S + 1                                 93215210
  340       CONTINUE                                                     93215211
C                                                                        93215212
            GO TO (60,80,200,220,360), ISTURM                            93215213
C     ********** REFINE INTERVALS **********                            93215214
  360       IF (S .GE. K) GO TO 400                                      93215215
            XU = XI                                                      93215216
            IF (S .GE. M1) GO TO 380                                     93215217
            RV4(M1) = XI                                                 93215218
            GO TO 300                                                    93215219
  380       RV4(S+1) = XI                                                93215220
            IF (RV5(S) .GT. XI) RV5(S) = XI                              93215221
            GO TO 300                                                    93215222
  400       XO = XI                                                      93215223
            GO TO 300                                                    93215224
C     ********** K-TH EIGENVALUE FOUND **********                       93215225
  420       RV5(K) = XI                                                  93215226
        K = K - 1                                                        93215227
        IF (K .GE. M1) GO TO 250                                         93215228
C     ********** FIND VECTORS BY INVERSE ITERATION **********           93215229
        NORM = ABS(D(P))                                                 93215230
        IP = P + 1                                                       93215231
C                                                                        93215232
        DO 500 I = IP, Q                                                 93215233
  500   NORM = NORM + ABS(D(I)) + ABS(E(I))                              93215234
C     ********** EPS2 IS THE CRITERION FOR GROUPING,                    93215235
C                EPS3 REPLACES ZERO PIVOTS AND EQUAL                   93215236
C                ROOTS ARE MODIFIED BY EPS3,                           93215237
C                EPS4 IS TAKEN VERY SMALL TO AVOID OVERFLOW **********  93215238
        EPS2 = 1.0E-3 * NORM                                             93215239
        EPS3 = MACHEP * NORM                                             93215240
        UK = FLOAT(Q-P+1)                                                93215241
        EPS4 = UK * EPS3                                                 93215242
        UK = EPS4 / SQRT(UK)                                             93215243
        GROUP = 0                                                        93215244
        S = P                                                            93215245
C                                                                        93215246
        DO 920 K = M1, M2                                                93215247
           R = R + 1                                                     93215248
           ITS = 1                                                       93215249
           W(R) = RV5(K)                                                 93215250
           XI = RV5(K)                                                   93215251
C     ********** LOOK FOR CLOSE OR COINCIDENT ROOTS **********          93215252
```

```
257          IF (K .EQ. M1) GO TO 520                                   93215253
258          IF (X1 - X0 .GE. EPS2) GROUP = -1                          93215254
259          GROUP = GROUP + 1                                          93215255
260          IF (X1 .LE. X0) X1 = X0 + EPS3                             93215256
261 C     ********** ELIMINATION WITH INTERCHANGES AND                  93215257
262 C                INITIALIZATION OF VECTOR **********               93215258
263   520    V = D(1)                                                   93215259
264 C                                                                   93215260
265          DO 580 I = P, Q                                            93215261
266             RV6(I) = UK                                             93215262
267             IF (I .EQ. P) GO TO 560                                 93215263
268             IF (ABS(E(I)) .LT. ABS(U)) GO TO 540                    93215264
269             XU = U / E(I)                                           93215265
270             RV4(I) = XU                                             93215266
271             RV1(I-1) = E(I)                                         93215267
272             RV2(I-1) = D(I) - XI                                    93215268
273             RV3(I-1) = 0.0                                          93215269
274             IF (I .NE. Q) RV3(I-1) = E(I+1)                         93215270
275             U = V - XU * RV2(I-1)                                   93215271
276             V = -XU * RV3(I-1)                                      93215272
277             GO TO 580                                               93215273
278   540       XU = E(I) / U                                           93215274
279             RV4(I) = XU                                             93215275
280             RV1(I-1) = U                                            93215276
281             RV2(I-1) = V                                            93215277
282             RV3(I-1) = 0.0                                          93215278
283   560       U = D(I) - XI - XU * V                                  93215279
284             IF (I .NE. Q) V = E(I+1)                                93215280
285   580    CONTINUE                                                   93215281
286 C                                                                   93215282
287          IF (U .EQ. 0.0) U = EPS3                                   93215283
288          RV1(Q) = U                                                 93215284
289          RV2(Q) = 0.0                                               93215285
290          RV3(Q) = 0.0                                               93215286
291 C     ********** BACK SUBSTITUTION                                  93215287
292 C                FOR I=Q STEP -1 UNTIL P DO -- **********          93215288
293   600    DO 620 II = P, Q                                           93215289
294             I = P + Q - II                                          93215290
295             RV6(I) = (RV6(I) - U * RV2(I) - V * RV3(I)) / RV1(I)    93215291
296             V = U                                                   93215292
297             U = RV6(I)                                              93215293
298   620    CONTINUE                                                   93215294
299 C     ********** ORTHOGONALIZE WITH RESPECT TO PREVIOUS             93215295
300 C                MEMBERS OF GROUP **********                      93215296
301          IF (GROUP .EQ. 0) GO TO 700                                93215297
302 C                                                                   93215298
303          DO 680 JJ = 1, GROUP                                       93215299
304             J = R - GROUP - 1 + JJ                                  93215300
305             XU = 0.0                                                93215301
306 C                                                                   93215302
307             DO 640 I = P, Q                                         93215303
308   640       XU = XU + RV6(I) * Z(I,J)                               93215304
309 C                                                                   93215305
310             DO 660 I = P, Q                                         93215306
311   660       RV6(I) = RV6(I) - XU * Z(I,J)                           93215307
312 C                                                                   93215308
313   680    CONTINUE                                                   93215309
314 C                                                                   93215310
315   700    NORM = 0.0                                                 93215311
316 C                                                                   93215312
317          DO 720 I = P, Q                                            93215313
318   720    NORM = NORM + ABS(RV6(I))                                  93215314
319 C                                                                   93215315
320          IF (NORM .GE. 1.0) GO TO 840                               93215316
```

```
      ********** FORWARD SUBSTITUTION **********                    93215317
         IF (ITS .EQ. 5) GO TO 960                                  93215318
         IF (NORM .NE. 0.0) GO TO 740                               93215319
         RV6(S) = EPS4                                              93215320
         S = S + 1                                                  93215321
         IF (S .GT. Q) S = P                                        93215322
         GO TO 780                                                  93215323
  740    XU = EPS4 / NORM                                           93215324
C                                                                   93215325
         DO 760 I = P, Q                                            93215326
  760    RV6(I) = RV6(I) * XU                                       93215327
C     ********** ELIMINATION OPERATIONS ON NEXT VECTOR              93215328
C                    ITERATE **********                             93215329
  780    DO 820 I = IP, Q                                           93215330
            U = RV6(I)                                              93215331
C     ********** IF RV1(I-1) .EQ. E(I), A ROW INTERCHANGE           93215332
C                    WAS PERFORMED EARLIER IN THE                   93215333
C                    TRIANGULARIZATION PROCESS **********           93215334
            IF (RV1(I-1) .NE. E(I)) GO TO 800                       93215335
            U = RV6(I-1)                                            93215336
            RV6(I-1) = RV6(I)                                       93215337
  800       RV6(I) = U - RV4(I) * RV6(I-1)                          93215338
  820    CONTINUE                                                   93215339
C                                                                   93215340
         ITS = ITS + 1                                              93215341
         GO TO 600                                                  93215342
C     ********** NORMALIZE SO THAT SUM OF SQUARES IS                93215343
C                    1 AND EXPAND TO FULL ORDER **********          93215344
  840    U = 0.0                                                    93215345
C                                                                   93215346
         DO 860 I = P, Q                                            93215347
  860    U = U + RV6(I)**2                                          93215348
C                                                                   93215349
         XU = 1.0 / SQRT(U)                                         93215350
C                                                                   93215351
         DO 880 I = 1, N                                            93215352
  880    Z(I,R) = 0.0                                               93215353
C                                                                   93215354
         DO 900 I = P, Q                                            93215355
  900    Z(I,R) = RV6(I) * XU                                       93215356
C                                                                   93215357
         XO = X1                                                    93215358
  920 CONTINUE                                                      93215359
C                                                                   93215360
  940 IF (Q .LT. N) GO TO 100                                       93215361
      GO TO 1001                                                    93215362
C     ********** SET ERROR -- NON-CONVERGED EIGENVECTOR **********  93215363
  960 IERR = 4 * N + R                                              93215364
      GO TO 1001                                                    93215365
C     ********** SET ERROR -- UNDERESTIMATE OF NUMBER OF            93215366
C                    EIGENVALUES IN INTERVAL **********             93215367
  980 IERR = 3 * N + 1                                              93215368
 1001 LB = T1                                                       93215369
      UB = T2                                                       93215370
      RETURN                                                        93215371
C     ********** LAST CARD OF TSTURM **********                     93215372
      END                                                           93215373
```

```
C                                                                       77215001
C     ------------------------------------------------------------------77215002
C                                                                       77215003
      SUBROUTINE TRED1(NM,N,A,D,E,E2)                                   77215004
C*NMS:MATHLIB:NATS:E!SPACK:SOURCE:TRED1              TRED1 E2.V2 11:28/73 LLL2
C                                                                       77215005
      INTEGER I,J,K,L,N,II,NM,JP1                                       77215006
      REAL A(NM,N),D(N),E(N),E2(N)                                      77215007
      REAL F,G,H,SCALE                                                  77215008
      REAL SQRT,ABS,SIGN                                                LLL25009
C                                                                       77215010
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TRED1,    77215011
C     NUM. MATH. 11, 181-195(1968) BY MARTIN, REINSCH, AND WILKINSON.   77215012
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 212-226(1971).   77215013
C                                                                       77215014
C     THIS SUBROUTINE REDUCES A REAL SYMMETRIC MATRIX                   77215015
C     TO A SYMMETRIC TRIDIAGONAL MATRIX USING                           77215016
C     ORTHOGONAL SIMILARITY TRANSFORMATIONS.                            77215017
C                                                                       77215018
C     ON INPUT-                                                         77215019
C                                                                       77215020
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL         77215021
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM          77215022
C          DIMENSION STATEMENT,                                         77215023
C                                                                       77215024
C        N IS THE ORDER OF THE MATRIX,                                  77215025
C                                                                       77215026
C        A CONTAINS THE REAL SYMMETRIC INPUT MATRIX.  ONLY THE          77215027
C          LOWER TRIANGLE OF THE MATRIX NEED BE SUPPLIED.               77215028
C                                                                       77215029
C     ON OUTPUT-                                                        77215030
C                                                                       77215031
C        A CONTAINS INFORMATION ABOUT THE ORTHOGONAL TRANS-             77215032
C          FORMATIONS USED IN THE REDUCTION IN ITS STRICT LOWER         77215033
C          TRIANGLE.  THE FULL UPPER TRIANGLE OF A IS UNALTERED,        77215034
C                                                                       77215035
C        D CONTAINS THE DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX,    77215036
C                                                                       77215037
C        E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE TRIDIAGONAL         77215038
C          MATRIX IN ITS LAST N-1 POSITIONS.  E(1) IS SET TO ZERO,      77215039
C                                                                       77215040
C        E2 CONTAINS THE SQUARES OF THE CORRESPONDING ELEMENTS OF E.    77215041
C          E2 MAY COINCIDE WITH E IF THE SQUARES ARE NOT NEEDED.        77215042
C                                                                       77215043
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,        77215044
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY         77215045
C                                                                       LLL1
C     LLL CONSULTANT...  R. P. DICKINSON                                LLL1
C     VERSION 2 - CORRECT DECLARATIONS FOR PUTT                         LLL2
C                                                                       77215046
C     ------------------------------------------------------------------77215047
C                                                                       77215048
      DO 100 I = 1, N                                                   77215049
  100 D(I) = A(I,I)                                                     77215050
C     ********** FOR I=N STEP -1 UNTIL 1 DO -- **********               77215051
      DO  300 II = 1, N                                                 77215052
         I = N + 1 - II                                                 77215053
         L = I - 1                                                      77215054
         H = 0.0                                                        77215055
         SCALE = 0.0                                                    77215056
         IF (L .LT. 1) GO TO 130                                        77215057
C     ********** SCALE ROW (ALGOL TOL THEN NOT NEEDED) **********       77215058
         DO 120 K = 1, L                                                77215059
  120    SCALE = SCALE + ABS(A(I,K))                                    77215060
```

```
C                                                                       77215061
      IF (SCALE .NE. 0.0) GO TO 140                                     77215062
  130 E(I) = 0.0                                                        77215063
      E2(I) = 0.0                                                       77215064
      GO TO 290                                                         77215065
C                                                                       77215066
  140 DO 150 K = I, L                                                   77215067
        A(I,K) = A(I,K) / SCALE                                         77215068
        H = H + A(I,K) * A(I,K)                                         77215069
  150 CONTINUE                                                          77215070
C                                                                       77215071
      E2(I) = SCALE * SCALE * H                                         77215072
      F = A(I,L)                                                        77215073
      G = -SIGN(SQRT(H),F)                                              77215074
      E(I) = SCALE * G                                                  77215075
      H = H - F * G                                                     77215076
      A(I,L) = F - G                                                    77215077
      IF (L .EQ. 1) GO TO 270                                           77215078
      F = 0.0                                                           77215079
C                                                                       77215080
      DO 240 J = 1, L                                                   77215081
        G = 0.0                                                         77215082
C     ********** FORM ELEMENT OF A*U **********                        77215083
        DO 180 K = 1, J                                                 77215084
  180   G = G + A(J,K) * A(I,K)                                         77215085
C                                                                       77215086
        JP1 = J + 1                                                     77215087
        IF (L .LT. JP1) GO TO 220                                       77215088
C                                                                       77215089
        DO 200 K = JP1, L                                               77215090
  200   G = G + A(K,J) * A(I,K)                                         77215091
C     ********** FORM ELEMENT OF P **********                          77215092
  220   E(J) = G / H                                                    77215093
        F = F + E(J) * A(I,J)                                           77215094
  240 CONTINUE                                                          77215095
C                                                                       77215096
      H = F / (H + H)                                                   77215097
C     ********** FORM REDUCED A **********                             77215098
      DO 260 J = 1, L                                                   77215099
        F = A(I,J)                                                      77215100
        G = E(J) - H * F                                                77215101
        E(J) = G                                                        77215102
C                                                                       77215103
        DO 260 K = 1, J                                                 77215104
          A(J,K) = A(J,K) - F * E(K) - G * A(I,K)                       77215105
  260 CONTINUE                                                          77215106
C                                                                       77215107
  270 DO 280 K = 1, L                                                   77215108
  280 A(I,K) = SCALE * A(I,K)                                           77215109
C                                                                       77215110
  290 H = D(I)                                                          77215111
      D(I) = A(I,I)                                                     77215112
      A(I,I) = H                                                        77215113
  300 CONTINUE                                                          77215114
C                                                                       77215115
      RETURN                                                            77215116
C     ********** LAST CARD OF TRED1 **********                         77215117
      END                                                               77215118
```

```
      SUBROUTINE BSTURM(N1,K1,IC1,IFLAG,XL,XU,D,E,E2)                    BSTU   0
C*NMS:MATHL!B:NATS:EISPACK:SOURCE:BSTURM          BSTURM E1.V2 10/09/73 LLL2  10
C                                                                        BSTU  20
      INTEGER  N1,K1,IC1,IFLAG                                           LLL1  30
      REAL  XL,XU, MACHEP                                                LLL1  40
      REAL  D(N1),E(N1),E2(N1)                                           LLL1  50
C                                                                        BSTU  60
C     ----------------------------------------------------------------BSTU  70
C                                                                        BSTU  80
C     BSTURM CALCULATES AN INTERVAL FOR A REAL SYMMETRIC TRIDIAGONAL     BSTU  90
C     MATRIX.  THIS INTERVAL WILL CONTAIN AT LEAST A SPECIFIED NUMBER OFBSTU 100
C     SMALLEST OR LARGEST EIGENVALUES, WHICHEVER IS DESIRED.             BSTU 110
C     THE POSSIBILITY OF GAINING MORE EIGENVALUES THAN DESIRED IN THE    BSTU 120
C     INTERVAL IS CAUSED BY CERTAIN EIGENVALUES HAVING MULTIPLICITIES    BSTU 130
C     GREATER THAN ONE.                                                  BSTU 140
C                                                                        BSTU 150
C     THIS SUBROUTINE CAN ALSO BE USED TO EVALUATE THE STURM COUNT FOR   BSTU 160
C     ANY REAL NUMBER, WHERE THE STURM COUNT IS THE NUMBER OF            BSTU 170
C     EIGENVALUES, COUNTING MULTIPLICITIES, STRICTLY TO THE LEFT OF THE BSTU 180
C     GIVEN INPUT NUMBER.                                                BSTU 190
C                                                                        BSTU 200
C     BSTURM WAS DESIGNED TO SUPPLEMENT THE EISPACK PACKAGE.             BSTU 210
C     THIS SUBROUTINE WAS WRITTEN AT LAWRENCE LIVERMORE LABORATORY       BSTU 220
C     DURING AUGUST 1973 BY ROBERT DICKINSON OF N.M.S.                   BSTU 230
C                                                                        BSTU 240
C     ----------------------------------------------------------------BSTU 250
C                                                                        BSTU 260
C     ON INPUT:                                                          BSTU 270
C                                                                        BSTU 290
C        N1 IS THE ORDER OF THE MATRIX A.                                BSTU 290
C                                                                        BSTU 300
C        K1 IS THE NUMBER OF EXTREME EIGENVALUES DESIRED. IF IFLAG IS 2 BSTU 310
C          OR 3.                                                         BSTU 320
C                                                                        BSTU 330
C        IFLAG MAY HAVE THREE VALUES.                                    BSTU 340
C          IF IFLAG IS 1 THE USER WANTS THE STURM COUNT AT THE INPUT      BSTU 350
C            VALUE XL.                                                   BSTU 360
C          IF IFLAG IS 2 THE USER  WANTS AN INTERVAL WHICH CONTAINS      BSTU 370
C            THE K1 SMALLEST EIGENVALUES OF A.                           BSTU 380
C          IF IFLAG IS 3 THE USER WANTS AN INTERVAL WHICH CONTAINS       BSTU 390
C            THE K1 LARGEST EIGENVALUES OF A.                            BSTU 400
C                                                                        BSTU 410
C        XL IS AN INPUT NUMBER IF IFLAG IS 1, AND IN THIS CASE THE       BSTU 420
C          NUMBER OF EIGENVALUES, COUNTING MULTIPLICITIES, STRICTLY TO   BSTU 430
C          THE LEFT OF XL IS TO BE COMPUTED.                             BSTU 440
C                                                                        BSTU 450
C        D IS A LINEAR ARRAY OF DIMENSION N1 WHICH CONTAINS THE          BSTU 460
C          DIAGONAL ELEMENTS OF A.  D(I)=A(I,I) FOR I=1 THROUGH N1.      BSTU 470
C                                                                        BSTU 480
C        E IS A LINEAR ARRAY OF DIMENSION N1 WHICH CONTAINS THE          BSTU 490
C          SUBDIAGONAL ELEMENTS OF A.  E(1)=0, AND E(I)=A(I,I-1) FOR     BSTU 500
C          I=2 THROUGH N1.                                               BSTU 510
C                                                                        BSTU 520
C        E2 IS A LINEAR ARRAY OF DIMENSION N1 WHICH CONTAINS THE SQUARESBSTU 530
C          OF ELEMENTS OF THE E ARRAY.  THAT IS E2(I)=E(I)*E(I) FOR I=1 BSTU 540
C          THROUGH N1.                                                   BSTU 550
C                                                                        BSTU 560
C     ON OUTPUT:                                                         BSTU 570
C                                                                        BSTU 580
C        XL AND XU ARE THE LOWER AND UPPER ENDPOINTS, RESPECTIVELY, OF   BSTU 590
C          THE DESIRED INTERVAL IF THE INPUT VALUE OF IFLAG IS 2 OR 3.   BSTU 600
C                                                                        BSTU 610
C        IC1 IS THE STURM COUNT IF THE INPUT VALUE OF IFLAG IS 1.        BSTU 620
C        IC1 IS THE NUMBER OF EIGENVALUES IN THE INTERVAL IF THE INPUT   BSTU 630
```

```
C          VALUE OF IFLAG IS 2 OR 3.                                    BSTU 640
C                                                                       BSTU 650
C       IFLAG IS SET TO 0 IF THERE WERE NO ERRORS IN INPUT PARAMETERS;  BSTU 660
C         OTHERWISE IFLAG IS SET TO -1.                                 BSTU 670
C         NOTE.. IFLAG IS ALWAYS AN INTEGER.                            BSTU 680
C                                                                       BSTU 690
C    LLL CONSULTANT...  R. P. DICKINSON                                 BSTU 700
C                                                                       BSTU 710
C    ------------------------------------------------------------------BSTU 720
C    LATEST REVISION   - OCTOBER 9, 1973                                LLL2 730
C VERSION 2 = CHANGES FOR ANSI AND NMS STANDARDS -- F.N.F.              LLL2 740
C                                                                       BSTU 750
C    ********** FPER GUARANTEES THAT BSTURM WILL RETURN AN             BSTU 751
C                        INTERVAL OF NONZERO LENGTH.                    BSTU 752
      DATA  ISET/0/, FPER/.01/                                          LLL2 760
      KFLAG=IFLAG                                                       BSTU 770
      IF((KFLAG.LT.1).OR.(KFLAG.GT.3)) GO TO 30                         LLL2 780
      IF(ISET.NE.0) GO TO 2                                             LLL2 790
C                                                                       BSTU 800
C    ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING      BSTU 810
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.   BSTU 820
C                                                                       BSTU 830
C                **********                                             BSTU 840
      MACHEP=2.**(-47)                                                  BSTU 850
C     **********  RMACHEP = 1./MACHEP  **********                       BSTU 860
      RMCHEP=2.**(47)                                                   BSTU 870
      ISET=1                                                            BSTU 880
C                                                                       BSTU 890
    2 N=NI                                                              BSTU 900
      K=K1                                                              BSTU 910
      IF(N.LE.1) GO TO 30                                               LLL2 920
      E2(1)=E(1)=0.                                                     LLL2 930
      DO 3 I=2,N                                                        LLL2 940
        IF(ABS(E(I)).GT.MACHEP*(ABS(D(I))+ABS(D(I-1)))) GO TO 3         LLL2 950
        E2(I)=0.                                                        LLL2 960
    3 CONTINUE                                                          LLL2 970
      IF(KFLAG.NE.1) GO TO 12                                           BSTU1030
      IPAS=1                                                            BSTU1040
      XC=XL                                                             BSTU1050
    5 IC=0                                                              LLL21060
      DO 10 I=1,N                                                       BSTU1070
        IF(E2(I).NE.0.) GO TO 7                                         LLL21080
        U=D(I)-XC                                                       LLL21090
        IF(U.LT.0.) IC=IC+1                                             LLL21100
        GO TO 10                                                        LLL21110
    7   IF(U.NE.0.) GO TO 8                                             LLL21120
        V=ABS(E(I))*RMCHEP                                              LLL21130
        GO TO 9                                                         LLL21140
    8   V=E2(I)/U                                                       LLL21150
    9   U=D(I)-XC-V                                                     LLL21160
        IF(U.LT.0.) IC=IC+1                                             LLL21170
   10 CONTINUE                                                          BSTU1180
      GO TO (11,14,20) ,IPAS                                            BSTU1190
   11 IFLAG=1                                                           BSTU1200
      IC1=IC                                                            BSTU1210
      RETURN                                                            BSTU1220
C                                                                       BSTU1230
   12 IF((K.LT.1).OR.(K.GT.N)) GO TO 30                                 LLL21240
C     **********  CALCULATE GERSHGORIN BOUNDS **********                BSTU1250
      XT=ABS(E(N))                                                      BSTU1260
      XL1=D(N)-XT                                                       BSTU1270
      XU1=D(N)+XT                                                       BSTU1280
      NM1=N-1                                                           BSTU1285
      DO 13 I=1,NM1                                                     BSTU1290
```

```
      XT=ABS(E(I))+ABS(E(I+1))                                          LLL21300
      XL2=D(I)-XT                                                       LLL21310
      XU2=D(I)+XT                                                       LLL21320
      IF(XL2.LT.XL1) XL1=XL2                                            LLL21330
      IF(XU2.GT.XU1) XU1=XU2                                            LLL21340
   13 CONTINUE                                                          BSTU1350
C     ********** EVALUATE UPPER COUNT **********                       BSTU1360
      XC=XU1                                                            BSTU1370
      IPAS=2                                                            BSTU1380
      GO TO 5                                                           LLL21390
   14 IPAS=3                                                            BSTU1400
      ICU=IC                                                            BSTU1410
      ICL=0                                                             BSTU1420
      IFLAG=0                                                           BSTU1430
      IF(KFLAG.NE.2) GO TO 17                                           LLL21440
      XL=AMIN1(XL1-FPER*ABS(XL1),-FPER)                                 LLL21450
      EPS1=AMAX1(ABS(XL),ABS(XU1))*MACHEP                               BSTU1455
      ITEST=K                                                           BSTU1460
      IF(K.LT.IC) GO TO 19                                              LLL21470
      XU=AMAX1(XU1+FPER*ABS(XU1),FPER)                                  LLL21480
      IC1=N                                                             BSTU1490
      RETURN                                                            BSTU1500
C                                                                       BSTU1510
   17 XL=AMAX1(XU1+FPER*ABS(XU1),FPER)                                  BSTU1520
      EPS1=AMAX1(ABS(XL1),ABS(XU1))*MACHEP                              BSTU1525
      ITEST=N-K                                                         BSTU1530
      IF((N-IC).LT.K) GO TO 19                                          LLL21540
      XL=XU1                                                            LLL21550
      IC1=N-IC                                                          BSTU1550
      RETURN                                                            BSTU1570
C                                                                       BSTU1580
   19 XC=(XL1+XU1)*.5                                                   BSTU1590
      GO TO 5                                                           BSTU1600
   20 IF((XU1-XL1).LE.(2.*MACHEP*(ABS(XU1)+ABS(XL1))+EPS1)) GO TO 24    BSTU1610
      IF(IC.LE.ITEST) GO TO 22                                          LLL21620
      XU1=XC                                                            LLL21630
      ICU=IC                                                            BSTU1640
      GO TO 19                                                          BSTU1650
   22 IF(IC.GE.ITEST) GO TO 27                                          LLL21660
      XL1=XC                                                            LLL21670
      ICL=IC                                                            BSTU1680
      GO TO 19                                                          BSTU1690
   24 IF(KFLAG.NE.2) GO TO 26                                           LLL21700
      XU=XU1                                                            LLL21710
      IC1=ICU                                                           BSTU1720
      RETURN                                                            BSTU1730
C                                                                       BSTU1740
   26 XL=XL1                                                            BSTU1750
      IC1=N-ICL                                                         BSTU1760
      RETURN                                                            BSTU1770
C                                                                       BSTU1780
   27 IF(KFLAG.NE.2) GO TO 29                                           LLL21790
      XU=XC                                                             LLL21800
      IC1=IC                                                            BSTU1810
      RETURN                                                            BSTU1820
C                                                                       BSTU1830
   29 XL=XC                                                             BSTU1840
      IC1=N-IC                                                          BSTU1850
      RETURN                                                            BSTU1860
C                                                                       BSTU1870
C     ********** ERROR RETURN **********                               BSTU1880
   30 IFLAG=-1                                                          BSTU1890
      RETURN                                                            BSTU1900
C     ********** LAST CARD OF BSTURM **********                        BSTU1910
```

```
                                                                BSTV1920
      END
```

```
C                                                                      79215001
C     ------------------------------------------------------------------79215002
C                                                                      79215003
      SUBROUTINE TRBAK1(NM,N,A,E,M,Z)                                  79215004
C*NMS:MATHLIB:NATS:EISPACK:SOURCE:TRBAK1          TRBAK1 E2.V1 09/19/73 LLLI
C                                                                      79215005
      INTEGER I,J,K,L,M,N,NM                                           79215006
      REAL A(NM,N),E(N),Z(NM,M)                                        79215007
      REAL H,S                                                         79215008
C                                                                      79215009
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TRBAK1,  79215010
C     NUM. MATH. 11, 181-195(1968) BY MARTIN, REINSCH, AND WILKINSON.  79215011
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 212-226(1971).  79215012
C                                                                      79215013
C     THIS SUBROUTINE FORMS THE EIGENVECTORS OF A REAL SYMMETRIC       79215014
C     MATRIX BY BACK TRANSFORMING THOSE OF THE CORRESPONDING           79215015
C     SYMMETRIC TRIDIAGONAL MATRIX DETERMINED BY  TRED1.               79215016
C                                                                      79215017
C     ON INPUT-                                                        79215018
C                                                                      79215019
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL        79215020
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM         79215021
C          DIMENSION STATEMENT,                                        79215022
C                                                                      79215023
C        N IS THE ORDER OF THE MATRIX,                                 79215024
C                                                                      79215025
C        A CONTAINS INFORMATION ABOUT THE ORTHOGONAL TRANS-            79215026
C          FORMATIONS USED IN THE REDUCTION BY  TRED1                  79215027
C          IN ITS STRICT LOWER TRIANGLE,                               79215028
C                                                                      79215029
C        E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE TRIDIAGONAL        79215030
C          MATRIX IN ITS LAST N-1 POSITIONS.  E(1) IS ARBITRARY,       79215031
C                                                                      79215032
C        M IS THE NUMBER OF EIGENVECTORS TO BE BACK TRANSFORMED,       79215033
C                                                                      79215034
C        Z CONTAINS THE EIGENVECTORS TO BE BACK TRANSFORMED            79215035
C          IN ITS FIRST M COLUMNS.                                     79215036
C                                                                      79215037
C     ON OUTPUT-                                                       79215038
C                                                                      79215039
C        Z CONTAINS THE TRANSFORMED EIGENVECTORS                       79215040
C          IN ITS FIRST M COLUMNS.                                     79215041
C                                                                      79215042
C     NOTE THAT TRBAK1 PRESERVES VECTOR EUCLIDEAN NORMS.               79215043
C                                                                      79215044
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,       79215045
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY        79215046
C                                                                      LLLI
C     LLL CONSULTANT...  R. P. DICKINSON                               LLLI
C                                                                      79215047
C     ------------------------------------------------------------------79215048
C                                                                      79215049
      IF (N .EQ. 1) GO TO 200                                          79215050
C                                                                      79215051
      DO 140 I = 2, N                                                  79215052
         L = I - 1                                                     79215053
C     ********** H BELOW IS NEGATIVE OF H FORMED IN TRED1 **********  79215054
         H = E(I) * A(I,L)                                             79215055
         IF (H .EQ. 0.0) GO TO 140                                     79215056
C                                                                      79215057
         DO 130 J = 1, M                                               79215058
            S = 0.0                                                    79215059
C                                                                      79215060
            DO 110 K = 1, L                                            79215061
```

```
      110         S = S + A(I,K) * Z(K,J)                                   79215062
C                                                                           79215063
            S = S / H                                                       79215064
C                                                                           79215065
            DO 120 K = I, L                                                 79215066
      120   Z(K,J) = Z(K,J) + S * A(I,K)                                    79215067
C                                                                           79215068
      130   CONTINUE                                                        79215069
C                                                                           79215070
      140 CONTINUE                                                          79215071
C                                                                           79215072
      200 RETURN                                                            79215073
C     ********** LAST CARD OF TRBAK **********                             79215074
      END                                                                   79215075
```

```
      SUBROUTINE QSF(H,Y,Z,NDIM)
C
C     .....................................................................
C
C        SUBROUTINE QSF
C
C        PURPOSE
C           TO COMPUTE THE VECTOR OF INTEGRAL VALUES FOR A GIVEN
C           EQUIDISTANT TABLE OF FUNCTION VALUES.
C
C        USAGE
C           CALL QSF (H,Y,Z,NDIM)
C
C        DESCRIPTION OF PARAMETERS
C           H      - THE INCREMENT OF ARGUMENT VALUES.
C           Y      - THE INPUT VECTOR OF FUNCTION VALUES.
C           Z      - THE RESULTING VECTOR OF INTEGRAL VALUES. Z MAY BE
C                    IDENTICAL WITH Y.
C           NDIM   - THE DIMENSION OF VECTORS Y AND Z.
C
C        REMARKS
C           NO ACTION IN CASE NDIM LESS THAN 3.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           BEGINNING WITH Z(1)=0, EVALUATION OF VECTOR Z IS DONE BY
C           MEANS OF SIMPSONS RULE TOGETHER WITH NEWTONS 3/8 RULE OR A
C           COMBINATION OF THESE TWO RULES. TRUNCATION ERROR IS OF
C           ORDER H**5 (I.E. FOURTH ORDER METHOD). ONLY IN CASE NDIM=3
C           TRUNCATION ERROR OF Z(2) IS OF ORDER H**4.
C           FOR REFERENCE, SEE
C           (1) F.B.HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS,
C               MCGRAW-HILL, NEW YORK/TORONTO/LONDON, 1956, PP.71-76.
C           (2) R.ZURMUEHL, PRAKTISCHE MATHEMATIK FUER INGENIEURE UND
C               PHYSIKER, SPRINGER, BERLIN/GOETTINGEN/HEIDELBERG, 1963,
C               PP.214-221.
C
C     .....................................................................
C
C
C
      DIMENSION Y(1),Z(1)
C
      HT=.3333333*H
      IF(NDIM-5)7,8,1
C
C     NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION LOOP
    1 SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+Y(6))
      SUM2=Y(5)+Y(5)
      SUM2=SUM2+SUM2
      SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))
      Z(1)=0.
      AUX=Y(3)+Y(3)
      AUX=AUX+AUX
      Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
      Z(3)=SUM1
```

```
      Z(4)=SUM2
      IF(NDIM-6)5,5,2
C
C     INTEGRATION LOOP
    2 DO 4 I=7,NDIM,2
      SUM1=AUX1
      SUM2=AUX2
      AUX1=Y(I-1)+Y(I-1)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
      Z(I-2)=SUM1
      IF(I-NDIM)3,6,6
    3 AUX2=Y(I)+Y(I)
      AUX2=AUX2+AUX2
      AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
    4 Z(I-1)=SUM2
    5 Z(NDIM-1)=AUX1
      Z(NDIM)=AUX2
      RETURN
    6 Z(NDIM-1)=SUM2
      Z(NDIM)=AUX1
      RETURN
C     END OF INTEGRATION LOOP
C
    7 IF(NDIM-3)12,11,8
C
C     NDIM IS EQUAL TO 4 OR 5
    8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))
      SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      Z(1)=0.
      AUX1=Y(3)+Y(3)
      AUX1=AUX1+AUX1
      Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
      IF(NDIM-5)10,9,9
    9 AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
   10 Z(3)=SUM1
      Z(4)=SUM2
      RETURN
C
C     NDIM IS EQUAL TO 3
   11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
      SUM2=Y(2)+Y(2)
      SUM2=SUM2+SUM2
      Z(3)=HT*(Y(1)+SUM2+Y(3))
      Z(1)=0.
      Z(2)=SUM1
   12 RETURN
      END
```

```
      SUBROUTINE PLOT5(Y,M,NF,NSX)                                      HUELSMAN
      DIMENSION Y(5,100), LINE(101),L(11),JL(5)                         HUELSMAN
      DATA JL(1), JL(2), JL(3), JL(4), JL(5)
     |                  /1HA,1HB,1HC,1HD,1HE/,JN,JP,JI,JBLANK,JZ/
     1H-,1H+,1H|,1H ,1H$/                                               HUELSMAN
      NS=NSX                                                            HUELSMAN
      WRITE(6, 45)
   45 FORMAT (1H1)
      IF (NS.NE.999) GO TO 95                                           HUELSMAN
      NS=100
      DO 70 J=1,M
      YMAX=-1.E+50
      YMIN=1.E+50
      DO 55 I=1,NF
      IF (Y(J,I).GT.YMAX) YMAX=Y(J,I)
      IF (Y(J,I).LT.YMIN) YMIN=Y(J,I)
   55 CONTINUE
      RANGE=YMAX-YMIN
      TEMP=100./RANGE
      DO 60 I=1,NF
   60 Y(J,I)=(Y(J,I)-YMIN)*TEMP
      WRITE (6, 65) JL(J), YMIN, YMAX, RANGE
   65 FORMAT (1X,5HPLOT ,A1,5H FROM,E10.3,3H TO,
     1E10.3,9H, RANGE =,E10.3)
   70 CONTINUE
      WRITE (6, 75)
   75 FORMAT ( )
   95 DO 99 I=1,101
      LINE(I)=JBLANK                                                    HUELSMAN
   99 CONTINUE                                                          HUELSMAN
      N=0                                                               HUELSMAN
      DO 101 I=1,11                                                     HUELSMAN
      L(I)=10*I-110+NS                                                  HUELSMAN
  101 CONTINUE                                                          HUELSMAN
      WRITE (6, 105) (L(I), I = 1, 11)
  105 FORMAT (3X,11(I4,6X),6HY(1,1))                                    HUELSMAN
      GO TO 115                                                         HUELSMAN
  110 IF (N/10-(N-1)/10) 125,125,115                                    HUELSMAN
  115 ND=0                                                              HUELSMAN
      DO 120 I=1,10                                                     HUELSMAN
      ND=ND+1                                                           HUELSMAN
      LINE(ND)=JP                                                       HUELSMAN
      DO 120 J=1,9                                                      HUELSMAN
      ND=ND+1                                                           HUELSMAN
  120 LINE(ND)=JN                                                       HUELSMAN
      LINE(101)=JP                                                      HUELSMAN
      IF (N) 135,121,135                                                HUELSMAN
  121 WRITE (6, 170) N, LINE
      GO TO 185                                                         HUELSMAN
  125 DO 130 I=1,101,10                                                 HUELSMAN
      LINE(I)=JI                                                        HUELSMAN
  130 CONTINUE                                                          HUELSMAN
  135 DO 160 I=1,M                                                      HUELSMAN
      XNS=NS                                                            HUELSMAN
      JA=Y(I,N)+101.49999-XNS                                           HUELSMAN
      IF (JA-101) 140,155,145                                           HUELSMAN
  140 IF (JA) 150,150,155                                               HUELSMAN
  145 LINE(101)=JZ                                                      HUELSMAN
      GO TO 160                                                         HUELSMAN
  150 LINE(1)=JZ                                                        HUELSMAN
      GO TO 160                                                         HUELSMAN
  155 LINE(JA)=JL(I)                                                    HUELSMAN
  160 CONTINUE                                                          HUELSMAN
      IF (N/10-(N-1)/10) 175,175,165                                    HUELSMAN
```

```
      165 WRITE (6, 170) N, LINE, Y(1, N)
      170 FORMAT (1X,I4,10(A1,1X,  E12.5)                    HUELSMAN
          GO TO 185                                          HUELSMAN
      175 WRITE (6, 180) LINE, Y(1, N)
      180 FORMAT (5X,10(A1,1X,E12.5)                         HUELSMAN
      185 DO 190 I=1,10                                      HUELSMAN
          LINE(I)=JBLANK                                     HUELSMAN
      190 CONTINUE                                           HUELSMAN
      195 N=N+1                                              HUELSMAN
          IF (N-NF) 110,110,200                              HUELSMAN
      200 RETURN                                             HUELSMAN
          END                                                HUELSMAN
```

```
      SUBROUTINE LFILON (TZERO,T,FT,FTA,NT,F,FFZERO,FF,NF,B,IND,ITF)
C
C
C  THIS SUBROUTINE PERFORMS THE FOURIER TRANSFORM EITHER FROM TIME TO FREQUENCY
C  OR FREQUENCY TO TIME.  LIMITATIONS ARE THAT THE TIME FUNCTION MUST BE REAL
C  AND THE TIME FUNCTION MUST BE CAUSAL.
C
C  DESCRIPTION OF ARGUMENTS
C
C       TZERO  - INITIAL VALUE OF TIME.  THE CORRESPONDING FUNCTION OF TIME IS
C                SET TO ZERO BY THE SUBROUTINE
C
C       T      - ARRAY CONTAINING TIME VALUES IN ASCENDING ORDER
C
C       FT     - ARRAY CONTAINING FUNCTION OF TIME VALUES (REAL)
C
C       FTA    - ADDITIONAL ARRAY CONTAINING FUNCTION OF TIME VALUES.  USED
C                ONLY IN FREQUENCY TO TIME MODE
C
C       NT     - NUMBER OF TIME POINTS, EXCLUDING TZERO
C
C       F      - ARRAY CONTAINING FREQUENCY VALUES IN ASCENDING ORDER
C
C       FFZERO - FUNCTION OF FREQUENCY VALUE (COMPLEX) AT ZERO FREQUENCY
C
C       FF     - ARRAY (COMPLEX) CONTAINING FUNCTION OF FREQUENCY VALUES
C
C       NF     - NUMBER OF FREQUENCY POINTS, EXCLUSIVE OF ZERO FREQUENCY POINT
C                ASSOCIATED WITH FFZERO
C
C       B      - AUXILIARY ARRAY USED INTERNALLY ONLY BUT DIMENSIONED IN MAIN
C                PROGRAM
C
C       IND    - AUXILIARY ARRAY USED INTERNALLY ONLY BUT DIMENSIONED IN MAIN
C                PROGRAM
C
C       ITF    - INTEGER INDICATOR SELECTING EITHER TIME TO FREQUENCY OR
C                FREQUENCY TO TIME TRANSFORM
C
C
C  TO USE SUBROUTINE FOR TIME TO FREQUENCY TRANSFORM, PERFORM THE FOLLOWING
C  IN THE MAIN PROGRAM --
C
C       1)  SET ITF TO 1
C       2)  SET NT AND NF TO NUMBER OF TIME POINTS AND NUMBER OF FREQUENCY
C           POINTS RESPECTIVELY
C       3)  DIMENSION T, FT, AND B WITH VALUE OF NT (FTA IS NOT USED AND MAY
C           BE DIMENSIONED 1
C       4)  NAME FFZERO, FF, AND B, AS COMPLEX
C       5)  DIMENSION F, FF, AND IND WITH VALUE OF NF
C       6)  SUPPLY VALUES FOR TZERO, T, FT, NT, F, NF, AND ITF.  SUBROUTINE WILL
C           RETURN FFZERO AND FF VALUES
C
C  TO USE SUBROUTINE FOR FREQUENCY TO TIME TRANSFORM, PERFORM FOLLOWING IN
C  MAIN PROGRAM --
C
C       1)  SET ITF TO 0
C       2)  SET NT AND NF TO NUMBER OF TIME POINTS AND NUMBER OF FREQUENCY
C           POINTS, RESPECTIVELY
C       3)  DIMENSION T, FT, FTA, AND INDM WITH VALUE OF NT
C       4)  NAME FFZERO, FF, AND B COMPLEX
C       5)  DIMENSION F, FF, AND B WITH VALUE OF NF
C       6)  SUPPLY VALUES FOR T, NT, F, FFZERO, FF, NF, AND ITF.  SUBROUTINE
C           WILL RETURN FT AND FTA VALUES.  THE TRANSFORM IS COMPUTED TWO WAYS.
```

```
C          FT CONTAINS THE RESULT OF THE TRANSFORM USING THE REAL PART OF FF
C          MULTIPLIED BY COSINE OF RADIAN FREQUENCY TIMES TIME.  FTA CONTAINS
C          THE RESULT OF THE TRANSFORM USING THE IMAGINARY PART OF FF MULTIPLIED
C          BY SINE OF RADIAN FREQUENCY TIMES TIME.  USER SHOULD CHECK FOR
C          CORRESPONDENCE OF THE TWO ANSWERS
C
C
      DIMENSION T(1),FT(1),F(1),FF(1),B(1),IND(1  ),FTA(1)
      COMPLEX FF,I,FSTORE,B,FFZERO,TSTORE
      TOPI = 6.283185308
      I = CMPLX(0.,1.)
      IF (ITF .EQ. 0) GO TO 303
C     TIME TO FREQUENCY TRANSFORM
C     DO LINEAR SEARCH FOR NUMBER TO DIVIDE TRAPAZOIDAL FROM FILONS METHOD
      NTP = NT - 1
      DO 51 M = 1,NF
      TAU = .1 / F(M)
      N = 1
      IF (TAU .LT. TZERO) GO TO 1
      DO 41 N = 1,NT
      IF (TAU .LT. T(N)) GO TO 1
   41 CONTINUE
      IND(M) = NT
      GO TO 51
    1 IND(M) = N - 1
   51 CONTINUE
   61 CONTINUE
   10 FORMAT (E15.5,I10)
C     COMPUTE COEFFICIENTS
      B(1) = (FT(2) - FT(1)) / (T(2) - T(1)) - FT(1) / (T(1) - TZERO)
      B(NT) = (FT(NTP) - FT(NT)) / (T(NT) - T(NTP))
      DO 11 N = 2,NTP
      NP = N + 1
      NL = N - 1
      B(N) = (FT(NP) - FT(N)) / (T(NP) - T(N)) - (FT(N) - FT(NL)) / (T(N
     1 ) - T(NL))
   11 CONTINUE
C     COMPUTE TRANSFORM
      FFZERO = FT(1) * (T(2) - TZERO)
      DO 81 N = 2,NTP
      NP = N + 1
      NL = N - 1
   81 FFZERO = FFZERO + FT(N) * (T(NP) - T(NL))
      FFZERO = FFZERO + FT(NT  ) * (T(NT  ) - T(NTP  ))
      FFZERO = FFZERO / 2.
      DO 21 M = 1,NF
      W = TOPI * F(M)
C     COMPUTE TRAPAZOIDAL PORTION
      INDM = IND(M)
      INDMM = INDM - 1
      INDMP = INDM + 1
      FSTORE = CMPLX(0.,0.)
      IF (INDMM) 5,2,6
    6 FF(M) = FT(1) * CEXP(-1 * W * T(1)) * (T(2) - TZERO)
      IF (INDM .EQ. 2) GO TO 3
      DO 71 N = 2,INDMM
      NP = N + 1
      NL = N - 1
      FF(M) = FF(M) + FT(N) * CEXP(-1 * W * T(N)) * (T(NP) - T(NL))
   71 CONTINUE
      FF(M) = FF(M) + FT(INDM) * CEXP(-1 * W * T(INDM)) * (T(INDM) -
     1 T(INDMM))
      GO TO 4
    3 FF(M) = FF(M) + FT(2) * CEXP(-1 * W * T(2)) * (T(2) - T(1))
```

```
      GO TO 4
    2 FF(M) = FT(1) * CEXP(-I * W * T(1)) * (T(1) - TZERO)
    4 FF(M) = FF(M) / 2.
C     COMPUTE FILON PORTION
      IF (INDI(M) .EQ. NT) GO TO 21
      FSTORE = FF(M)
      FF(M) =          FT(INDM) * CEXP(-I * W * (T(INDM) - TZERO)) * I *
     1 W - FT(NT) * CEXP(-I * W * (T(NT) - TZERO)) * I * W + ((FT((NDMP)
     2 - FT((NDM)) / (T((NDMP) - T(INDM))) * CEXP(-I * W * (T((INDM) -
     3 TZERO))
      GO TO 7
    5 FF(M) = FT(1) / (T(1) - TZERO) - FT(NT) * CEXP(-I * W * (T(NT) -
     1 TZERO)) * I * W
    7 CONTINUE
      DO 31 N = INDMP,NT
   31 FF(M) = FF(M) + B(N) * CEXP(-I * W * (T(N) - TZERO))
      FF(M) = -FF(M) * CEXP(-I * W * TZERO) / (W * W) + FSTORE
   21 CONTINUE
      RETURN
C     FREQUENCY TO TIME TRANSFORM
  303 CONTINUE
C     DO LINEAR SEARCH FOR NUMBER TO DIVIDE TRAPAZOIDAL FROM FILON
      DO 151 N = 1,NT
      TAU = .1 * T(N)
      M = 1
      IF (TAU .LT. 0.) GO TO 101
      DO 141 M = 1,NF
      IF (TAU .LT. F(M)) GO TO 101
  141 CONTINUE
      IND(N) = NF
      GO TO 151
  101 IND(N) = M - 1
  151 CONTINUE
C     COMPUTE COEFFICIENTS
      NTP = NF - 1
      B(1) = (FF(2) - FF(1)) / (F(2) - F(1)) - (FF(1) -FFZERO) / F(1)
      B(NF) = (FF(NTP) - FF(NF)) / (F(NF) - F(NTP))
      DO 111 M = 2,NTP
      NP = M + 1
      NL = M - 1
      B(M) = (FF(NP) - FF(M)) / (F(NP) - F(M)) - (FF(M) - FF(NL)) / (F(M
     1 ) - F(NL))
  111 CONTINUE
C     COMPUTE TRANSFORM
      DO 121 N = 1,NT
C     COMPUTE TRAPAZOIDAL PORTION
      INDM = IND(N)
      INDMM = INDM - 1
      INDMP = INDM + 1
      TSTORE = CMPLX(0.,0.)
      IF (INDMM) 105,102,106
  106 FT(N) = REAL(FFZERO) * F(1) + REAL(FF(1)) * COS(TOPI * F(1) * T(N)
     1 ) * F(2)
      FTA(N) = AIMAG(FF(1)) * SIN(TOPI * F(1) * T(N)) * F(2)
      IF (INDM .EQ. 2) GO TO 103
      DO 171 M = 2,INDMM
      NP = M + 1
      NL = M - 1
      FT(N) = FT(N) + REAL(FF(M)) * COS(TOPI * F(M) * T(N)) * (F(NP) -
     1 F(NL))
      FTA(N) = FTA(N) + AIMAG(FF(M)) * SIN(TOPI * F(M) * T(N)) * (F(NP)
     1 - F(NL))
  171 CONTINUE
      FT(N) = FT(N) + REAL(FF(INDM)) * COS(TOPI * F(INDM) * T(N)) *
```

```
      1 (F(INDM) - F(INDMM))
       FTA(N) = FTA(N) + AIMAG(FF(INDM)) * SIN(TOPI * F(INDM) * T(N)) *
      1 (F(INDM) - F(INDMM))
       GO TO 104
  103 FT(N) = FT(N) + REAL(FF(2)) * COS(TOPI * F(2) * T(N)) * (F(2) - F(
      1 1))
       FTA(N) = FTA(N) + AIMAG(FF(2)) * SIN(TOPI * F(2) * T(N)) * (F(2) -
      1 F(1))
       GO TO 104
  102 FT(N) = (REAL(FFZERO) + REAL(FF(1)) * COS(TOPI * F(1) * T(N)))*
      1 F(1)
       FTA(N) = AIMAG(FF(1)) * SIN(TOPI * F(1) * T(N)) * F(1)
  104 FT(N) = FT(N) * 2.
       FTA(N) = -FTA(N) * 2.
C      COMPUTE FILON PORTION
       IF (IND(N) .EQ. NF) GO TO 121
       TSTORE = CMPLX(FT(N),FTA(N))
       FT(N) = (REAL(FF(INDMP) - FF(INDM)) / (F(INDMP) - F(INDM))) * COS
      1 (TOPI * F(INDM) * T(N)) - TOPI * T(N) * (REAL(FF(NF)) * SIN(TOPI
      2 * F(NF) * T(N)) - REAL(FF(INDM)) * SIN(TOPI * F(INDM) * T(N)))
       FTA(N) = (AIMAG(FF(INDMP) - FF(INDM)) / (F(INDMP) - F(INDM))) *
      1 SIN(TOPI * F(INDM) * T(N)) + TOPI * T(N) * (AIMAG(FF(NF)) * COS
      2 (TOPI * F(NF) * T(N)) - AIMAG(FF(INDM)) * COS(TOPI * F(INDM) *
      3 T(N)))
       GO TO 107
  105 FT(N) = REAL(FF(1) - FFZERO) / F(1) - TOPI * T(N) * REAL(FF(NF)) *
      1 SIN(TOPI * F(NF) * T(N))
       FTA(N) = TOPI * T(N) * (AIMAG(FF(NF)) * COS(TOPI * F(NF) * T(N))
      1 - AIMAG(FFZERO))
  107 CONTINUE
       DO 131 M = INDMP,NF
       FT(N) = FT(N) + REAL(B(M)) * COS(TOPI * F(M) * T(N))
  131 FTA(N) = FTA(N) + AIMAG(B(M)) * SIN(TOPI * F(M) * T(N))
       FT(N) = -4. * FT(N) / (TOPI * TOPI * T(N) * T(N)) + REAL(TSTORE)
       FTA(N) = 4. * FTA(N) / (TOPI * TOPI * T(N) * T(N)) + AIMAG(TSTORE)
  121 CONTINUE
       RETURN
       END
```