FRA-72-11
REPORT NO. FRA-RT-73-28

PULSE TRANSMISSION OVER DISPERSIVE WAVEGUIDES IN RAILROAD COMMUNICATIONS: SOFTWARE FOR COMPUTER SIMULATION

R. E. Eaves

JULY 1973
INTERIM REPORT

Prepared for
DEPARTMENT OF TRANSPORTATION
FEDERAL RAILROAD ADMINISTRATION
Office of Research Development and Demonstrations
Washington, DC 20591

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FRA-RT-73-28		

4. Title and Subtitle	5. Report Date
PULSE TRANSMISSION OVER DISPERSIVE WAVEGUIDES IN RAILROAD COMMUNICATIONS: SOFTWARE FOR COMPUTER SIMULATION	July 1973
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
R.E. Eaves	DOT-TSC-FRA-72-11
9. Performing Organization Name and Address	10. Work Unit No.
Department of Transportation Transportation Systems Center Kendall Square Cambridge, MA 02142	R3303/RR3304 11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Department of Transportation Federal Railroad Administration Office of Research, Development & Demonstrations Washington, DC 20591	Interim Report 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Waveguides and transmission lines employed in train communications exhibit dispersion, which is caused by (a) their inherent properties and (b) the cumulative effect of discontinuities at joints. To provide the means to evaluate such waveguides, several computer programs have been developed to analyze and simulate the effect of dispersion on pulse transmission.

17. Key Words	18. Distribution Statement
Railroad trains, communications Electromagnetic, electromagnetic wave transmission Distortion, signal Wave dispersion, transmission	APPROVED FOR U.S. GOVERNMENT. THIS DOCUMENT IS EXEMPT FROM PUBLIC AVAILABILITY BECAUSE IT CONTAINS INCOMPLETE INFORMATION. TRANSMITTAL OF THIS DOCUMENT OUTSIDE OF THE U.S. GOVERNMENT MUST HAVE THE PRIOR APPROVAL OF THE FEDERAL RAILROAD ADMINSTRATION.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	60	

Form DOT F 1700.7 (8-69)

PREFACE

The work described herein has been performed as part of a continuing program to evaluate and develop communication systems for trains. This program is sponsored by the Department of Transportation, through the Federal Railroad Administration, Office of Research, Development, and Demonstrations.

The Office of Research, Development, and Demonstrations has supported previous studies at the Physical Science Laboratory, New Mexico State University. At Las Cruces, the use of frequency modulation with wayside waveguides and transmission lines has been considered. The purpose of the work described by this report is to develop an analysis and computer software to simulate pulse transmission over wayside waveguides and transmittion lines, and thereby provide a means to evaluate their use with pulse-code modulation.

Programming support from Dr. John Royal and other personnel at Kentron Hawaii, Ltd., is gratefully acknowledged.

CONTENTS

1. INTRODUCTION

Waveguides and transmission lines which couple energy across an air gap by means of leaky waves or surfaces waves offer a promising means of communicating with trains or other vehicles confined to a guideway. However, the dispersion which such structures exhibit can be a limiting factor in signal transmission over long distances. Since pulse-code modulation (PCM) is the most likely candidate in these applications, the pulse-transmission properties of such lines become an important question.

The effect of noise on the performance of PCM communication systems has been thoroughly treated in the literature. However, the effect of a distorting transmission medium has received considerably less attention. Thus, it is desirable to develop an analysis to serve as a basis for computer models of PCM transmission.

The term PCM refers not to a particular type of modulation, but to a diverse class of modulations. In maintaining the generality needed to treat this broad topic, the aim will be to provide the means for computing received signals or pulses. The judgment of numerical results will depend upon the type of PCM and the means of detection employed.

Four separate Fortran programs have been constructed for analyzing signal transmission. They compute the following:

a. An upper bound for distortion in the received signal,
b. Ratio of the energy in the distorted component of the received signal to the energy in the undistorted component of the received signal,
c. Detailed description of the output signal as a function of time (direct integration method), and
d. Detailed description of the output signal as a function of time (fast Fourier method).

Programs (a) and (b) each compute a number which is useful in judging the degree of distortion, while (c) and (d) are alternate programs for computing the distorted output signal as a function of time. These latter two programs have the advantage of providing more detailed information, but they require more computing time. Therefore, the suggested approach is first to use the former two programs so that a definite conclusion may be reached from their results. If still more information is required, then the other programs can be used to provide a detailed description of the output as a function of time.

2. SYSTEM AND SIGNAL DESCRIPTIONS

The properties of a transmission medium are described by the transfer function $H(\omega)$, which relates the transforms of input and output signals by $S_o(\omega)=H(\omega)S_i(\omega)$ (see appendix A.1). For common uniform transmission media, $H(\omega)$ is usually available in analytic form, or is easily determined experimentally with modest laboratory apparatus. However, its determination is more involved for the case of long transmission lines in which discontinuities or other deviations from uniformity are present; the problem is further complicated by the uncertainties of construction which accompany these discontinuities. Experimental determination first requires major construction, and provides data on just one set of parameters. A computer simulation of such a line, while by no means a minor undertaking, may be more expedient, and allows parameters of the line to be easily changed. The analysis presented here incorporates a transfer function $H(\omega)$ given by a computer simulation, such as that constructed by NMSU,[1-3] or determined by experiment. A transmitted input signal $s_i(t)$ emerges from the system as the received output signal $s_o(t)$, given by

$$s_o(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)S_i(\omega)\exp(+j\omega t)d\omega \; . \qquad (2\text{-}1)$$

The input signal can be written in the form $s_i(t)=p(t)(\exp+j\omega_o t)$, which is still general but suggests a pulse-modulated carrier. The transforms of $s_i(t)$ and $p(t)$ are related by $S_i(\omega)=P(\omega-\omega_o)$. It is convenient to introduce the translated transfer function $K(\omega)$ defined by $H(\omega)=K(\omega-\omega_o)$. Then for transmission limited to the band $\omega_o-\Delta\omega<\omega<\omega_o+\Delta\omega$, equation (2-1) can be written in the form:

$$s_o(t) = \frac{1}{2\pi}\int_{-\Delta\omega}^{\Delta\omega}\left\{K(\omega)P(\omega)\exp(+j\omega t)d\omega\right\}\exp(+j\omega_o t), \qquad (2\text{-}2)$$

which, like the input, is in the form of a pulse-modulated carrier. The integration in (eq. 2-2) can be performed directly only for some special cases of $K(\omega)$, and it is evident that, in general, numerical methods are needed. However, before proceeding in that direction, cases of $K(\omega)$ will be considered to provide insight, and to suggest a more refined form of equation (2-2).

3. SPECIAL TRANSFER FUNCTIONS

3.1 A $K(\omega)$ WHICH CAUSES NO ENVELOPE DISTORTION

If $K(\omega)$ is of the form

$$K(\omega) = \exp(-a_o)\exp\left[-j(b_o+b_1\omega)\right], \qquad (3\text{-}1)$$

the integral in (eq. 2-2) can be evaluated exactly to yield

$$s_o(t) = p(t-b_1)\exp(-a_o)\exp\left[+j(t-b_o)\right] .$$

Therefore, the roles of the constants a_o, b_o, and b_i are identified as follows:

a_o = uniform attenuation without distortion of signal shape,

b_o = displacement of the envelope relative to the r-f signal, and

b_1 = envelope delay; i.e., the time for the pulse to travel the length of line.

These alterations are insignificant for commonly used detection techniques which recognize only pulse-envelope shape.

3.2 A SMOOTH $K(\omega)$ WHICH CAUSES DISTORTION

Now to generalize $K(\omega)$ further to the form,

$$K(\omega) = \exp(a_o-a_1\omega-a_2\omega^2)\ \exp(-jb_o-jb_1\omega-jb_2\omega^2). \qquad (3\text{-}2)$$

While equation (2-2) cannot be integrated exactly for arbitrary pulses, it can be integrated exactly if the output pulse is of gaussian shape and transmission is not band-limited; i.e., $\Delta\omega\approx\infty$. Then the gaussian input pulse can be written in the form

$$p(t) = (\tau_i\sqrt{\pi})^{-1/2}\ \exp(-t^2/2\tau_i^{\,2}),$$

so that τ_i is a measure of the input pulse width. The resultant output, obtained through equation (2-2), is[4]

$$s_o(t) = r(t-b_1)\exp(-a_o)\exp[+j(\omega_o t-b_o)],$$

where the pulse envelope, $|r(t)|$, is also of gaussian shape. If the output pulse width is designated by τ_o, then the similar parameters, τ_i and τ_o, are conveniently related to give pulse broadening.

$$\tau_o = \left[\tau_i^2 + 2a_2 + 4b_2^2/(\tau_i^2 + 2a_2)\right]^{1/2} \quad .$$

This result admittedly has limited application. The form given by equation (3-2) assumes that the logarithm of $K(\omega)$ is closely approximated by a three-term power-series expansion. This is likely to be a good approximation for a uniform transmission line, but does not adequately describe the erratic behavior of $K(\omega)$ which accompanies discontinuities with some randomness in location. Furthermore, only pulses of gaussian shape have been considered.

In spite of these limitations, the results of this section provide useful formulas for hand calculations of a uniform transmission line. In some cases, the numbers obtained may render unnecessary a computer study of the more severe discontinuity case.

3.3 A $K(\omega)$ WITH A SPIKE WHICH CAUSES DISTORTION

Studies by NMSU have shown that for long transmission lines with some randomness in the placement of discontinuities, the transfer function may exhibit sharp spikes. As an extreme case, let such a spike at $\omega=\omega_s$ be represented by a delta function, separated from the principal part of the transfer function, $K_p(\omega)$:

$$K(\omega) = K_p(\omega) + A\delta(\omega-\omega_s),$$

where A is the area under the spike. Then, equation (2-2) is easily integrated to give

$$s_o(t) = s_p(t) + AP(\omega_s)\exp\left[j(\omega_s+\omega_o)t\right] \, ,$$

where
$$s_p(t) = \left\{ \frac{1}{2\pi} \int_{-\Delta\omega}^{\Delta\omega} K_p(\omega)\exp(+j\omega t)\,d\omega \right\} \exp(+j\omega_o t) \quad .$$

Therefore, a delta function spike results in a constant sinusoid at $\omega=\omega_o+\omega_s$. Of course, spikes encountered in practice have finite width and height, and consequently, do not result in signals unbounded in the time domain. Nevertheless, this result does demonstrate that abrupt features of the frequency domain correspond to signals which are spread out in the time domain.

Despite the almost trivial nature of this example, several conclusions can be drawn. First, it is not the height of the spikes which is of concern, but rather the area under them; and second, the effect of a spike can be minimized through a judicious choice of pulse shape. If $P(\omega)$ has a zero at ω_s, the spike will have no effect.

4. GENERAL TRANSFER FUNCTIONS

Section 3.1 has revealed that translated transfer functions of the form $\exp(-a_o)\exp[-j(b_o+b_1\omega)]$ can be handled analytically for arbitrary inputs. This suggests that the treatment of general transfer functions may be facilitated by first removing a part which is of that form. That is, general $K(\omega)$ is decomposed as follows:

$$K(\omega) = \exp\left\{-\psi(\omega)\right\}\exp\left\{-j\phi(\omega)\right\} = \exp\left\{-\left[a_o+\gamma(\omega)\right]\right\}\exp\left\{-j\left[b_o+b_1\omega+\mu(\omega)\right]\right\}.$$

This decomposition is by no means unique. For the present, the choice of a_o, b_o, and b_1 is considered to be arbitrary, and $\gamma(\omega)$ and $\mu(\omega)$ are defined by

$$\gamma(\omega)=\psi(\omega)-a_o,$$

$$\mu(\omega)=\phi(\omega)-b_o-b_1\omega.$$

For any decomposition of this form, the output as given by equation (2-2) can be written

$$s_o(t) = \alpha(t-b_1)\exp(-a_o)\exp\left\{j\left[\omega_o t-b_o+\theta(t-b_1)\right]\right\}, \qquad (4\text{-}1)$$

where $\alpha(t)$ and $\theta(t)$ are real and defined by

$$\alpha(t)\exp[+j\theta(t)] = \frac{1}{2\pi}\int_{-\Delta\omega}^{\Delta\omega} \exp[-\gamma(\omega)-j\mu(\omega)]P(\omega)\exp[+j\omega(t-b_1)]d\omega .$$

The constants a_o, b_o, and b_1 have well-defined effects, whereas the effects of $\gamma(\omega)$ and $\mu(\omega)$ must be numerically computed. Therefore it is desirable to express as much of $K(\omega)$ as possible in the form (3-1). To this end, the values of a_o, b_o, and b_1 are specified to have values which will minimize the functionals

$$I_1 = \int_{-\Delta\omega}^{\Delta\omega} \gamma^2(\omega)d\omega,$$

$$I_2 = \int_{-\Delta\omega}^{\Delta\omega} \mu^2(\omega)d\omega,$$

The values of a_o, b_o, and b_1 which result from this minimization, along with the corresponding $\gamma(\omega)$ and $\mu(\omega)$, are to be used in computing equation (4-1).

The output $s_o(t)$ can be expressed in a form which explicitly separates the distortion. This is accomplished through the identity

$$K(\omega) = K_1(\omega) + K_2(\omega),$$

where

$$K_1(\omega) = \exp(-a_o)\exp[-j(b_o+b_1\omega)] ,$$

$$K_2(\omega) = K_1(\omega)\{\exp[-\gamma(\omega)]\exp[-j\mu(\omega)] - 1\} .$$

Then according to (eq. 2-2), the output is

$$s_o(t) = s_1(t) + s_2(t),$$

where

$$s_1(t) = p(t-b_1)\exp(-a_o)\exp[+j(\omega_o t-b_o)] ,$$

and

$$s_2(t) = \left\{\frac{1}{2\pi}\int_{-\Delta\omega}^{\Delta\omega} K_2(\omega)P(\omega)\exp(+j\omega t)d\omega\right\}\exp(+j\omega_o t). \qquad (4-2)$$

The component $s_1(t)$ is recognized as being the undistorted input signal, except for those insignificant differences noted in section 3.1. The component $s_2(t)$ is identified as the distortion, essentially a background noise. The energy in the undistorted component $s_1(t)$ and the energy in the distorted component $s_2(t)$ are given by

$$E_1 = \int_{-\infty}^{\infty} |s_1(t)|^2 dt = \frac{1}{2\pi}\int_{-\Delta\omega}^{\Delta\omega} |K_1(\omega)|^2|P(\omega)|^2 d\omega ,$$

$$E_2 = \int_{-\infty}^{\infty} |s_2(t)|^2 dt = \frac{1}{2\pi} \int_{-\Delta\omega}^{\Delta\omega} |K_2(\omega)|^2 |P(\omega)|^2 d\omega ,$$

where Parseval's theorem has been applied. The ratio E_2/E_1 provides a measure of the signal distortion and is given by

$$\frac{E_2}{E_1} = \frac{\int_{-\Delta\omega}^{\Delta\omega} \left\{\exp[-2\gamma(\omega)]+1-2\exp[-\gamma(\omega)]\cos\mu(\omega)\right\} |P(\omega)|^2 d\omega}{\int_{-\Delta\omega}^{\Delta\omega} |P(\omega)|^2 d\omega} . \quad (4\text{-}3)$$

Another figure which may be useful in judging the distortion is the following upper bound for $|s_2(t)|$, which is obtained by replacing the integrand in (4-2) with its absolute value.

$$|s_2(t)| \leq \frac{\exp(-2a_o)}{2\pi} \int_{-\Delta\omega}^{\Delta\omega} \left\{\exp[-2\gamma(\omega)]+1-2\exp[-\gamma(\omega)]\cos\mu(\omega)\right\}^{1/2} |P(\omega)| d\omega . \quad (4\text{-}4)$$

The usefulness of this information is limited, in that it can show distortion to be acceptable but cannot show distortion to be unacceptable.

5. NUMERICAL ANALYSIS AND PROGRAMMING

The computations which will be considered are those indicated by (eqs. 4-1, 4-2, and 4-3). The main programs given in appendix sections B.1.1 and B.1.2 have been written to evaluate (eqs. 4-2 and 4-3). The one or two numerical integrations involved are performed by a trapezoidal rule subroutine given in B.3.1. Alternately, this subroutine can be replaced by the Simpson's rule subroutine given in B.3.2. However, the potential improvement in accuracy or computation time offered by Simpson's rule is realized only for dense sample points. For either subroutine, the time required for integral evaluation is roughly proportional to N, the number of points.

The Fourier inversion embodied in equation (4-1) will be a more lengthy process. Direct methods require that the integration be re-performed for each point desired in the time domain. Such a numerical integration is achieved by dividing the frequency interval into N subintervals, so that if N points in the time domain are computed, the total computing time required is proportional to N^2. Such an inversion by direct integration is accomplished through the main program in B.1.3, which may be used with either the integration subroutine based on the trapezoidal rule (B.3.1) or the one based on Simpson's rule (B.3.2).

The time required to complete the numerical inversion can be reduced considerably through the fast Fourier transform (appendix A.1), which does not treat points in time independently, and thereby, performs the inversion in a time proportional to N log N. However, it does lack the flexibility of direct integration because of two additional constraints: (a) the number of sample points must be 2^M, where M is an integer, and (b) the sample points must be equidistantly spaced. The main program in B.1.4 is based on the fast Fourier transform and is used in conjunction with the subroutine in B.3.2, which has been adapted from the IBM scientific subroutine version.

The analysis which has been presented presumes that the transfer function $H(\omega)$ has been given, or that a means has been provided for calculating it. For demonstration purposes, an existing computer program written by NMSU for the GASL line[2] has been used to generate $H(\omega)$. That program is embodied in subroutines given in B.2.1, B.2.2, and B.2.3.

The computations indicated by (eqs. 4-1, 4-2, and 4-3) must be preceded by determination of a_o, b_o, b_1, $\gamma(\omega)$, and $\mu(\omega)$. Therefore, all programs share the same subroutine (B.2.4) to find the a_o, b_o, and b_1 which will minimize

$$\int_{-\Delta\omega}^{\Delta\omega} \gamma^2(\omega)d\omega,$$

and

$$\int_{-\Delta\omega}^{\Delta\omega} \mu^2(\omega)d\omega \ .$$

Appendix A.2 describes the minimization technique on which this subroutine is based. In implementing this approach, it has been necessary to use double precision arithmetic to achieve acceptable accuracy.

Input pulses with gaussian spectrum, and hence gaussian envelope, were used in test cases, and it was verified that all programs are operational. Output pulses computed by the trapezoidal rule (B.1.3) and the fast Fourier transform (B.1.4) were in agreement, and these results reduced properly to gaussian pulses as distortion approached zero. The results of the programs given in (B.1.1) and (B.1.2) were consistent with those of the programs given in (B.1.3) and (B.1.4).

6. REFERENCES

1. Anon., Requirements and methodology for evaluation of the wayside communication link, Rep. No. PR00651, Phys. Sci. Lab., NMSU, Las Cruces, N.M., Sep. 1969.

2. Anon., Evaluation of FDM-FM modulation for use on wayside communication systems, Rep. No. PE00651, Phys. Sci. Lab., NMSU, Las Cruces, N.M., Mar. 1970.

3. Hu, A.S., Analysis of transmission lines with couplers for use on wayside communication systems, Rep. No. PA00762, Phys. Sci. Lab., NMSU, Las Cruces, N.M., July 1972.

4. Kapron, F.P. and Keck, D.B., Pulse transmission through a dielectric optical waveguide, Appl. Opt. 10 (7), 1519-1523 (July 1971).

5. Cooley, J.W. and Tukey, J.W., [An] Algorithm for the machine computation of complex Fourier series, Math. Comput. 19, 297-301 (April 1965).

6. Cooley, J.W. and Lewis, P.A.W., [The] Fast Fourier transform algorithm and its applications, IBM Res. Paper RC 1743, 1967.

APPENDIXES A AND B

APPENDIX A - MATHEMATICAL BACKGROUND

APPENDIX B - PROGRAMS AND SUBROUTINES

APPENDIX A: MATHEMATICAL BACKGROUND

A.1 FOURIER TRANSFORMS

The Fourier transform of f(t) is defined by

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\exp(-j\omega t)dt,$$

so that the inversion formula is

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\exp(+j\omega t)d\omega.$$

An important result which follows from this transform pair is Parseval's theorem:

$$\int_{-\infty}^{\infty} |f(t)|^2\, dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2 d\omega.$$

Although transforms or inverse transforms have been analytically determined for many well-known functions, most generally they must be evaluated numerically. This can be accomplished to any degree of accuracy (at least for integrals defined in the Riemann sense) by approximating the integral with a sum over sufficiently small subintervals. For band-limited spectra ($F(\omega)=o;\ \omega<\omega_1, \omega>\omega_2$), the inverse transform can be written

$$f(t) = \frac{1}{2\pi}\int_{\omega_1}^{\omega_2} F(\omega)\exp(+j\omega t)d\omega$$

$$= \exp(+j\omega_1 t)\,.\int_{o}^{\lambda} S(\nu)\exp(+j2\pi\nu t)d\nu,$$

where

$$\lambda = \frac{\omega_2-\omega_1}{2\pi},$$

$$S(\omega) = F\left(\frac{\omega+\omega_1}{2\pi}\right) .$$

Then for sufficiently large N,

$$\int_o^{\lambda} S(\nu)\exp(+j2\pi\nu t)d\nu \approx \Delta\nu. \sum_{n=o}^{N-1} S(n.\Delta\nu)\exp(+j2\pi n\ \Delta\nu.t).$$

Evaluation at the values of t given by $t_k=k.\Delta t$, with Δt chosen so that $\Delta t\cdot\Delta\nu=1/N$, yields

$$\int_o^{\lambda} S(\nu)\exp(+j2\pi\nu t_k)d\nu \approx \Delta\nu. \sum_{n=o}^{N-1} S(n\cdot\Delta\nu)\exp(+j2\pi nk/N).$$

This is part of the following discrete Fourier transform pair,

$$S(n) = \Delta t \sum_{k=o}^{N-1} x(k)\exp(-j2\pi nk/N) \qquad n = o,1,\cdots,N-1$$

$$x(k) = \Delta\nu \sum_{n=o}^{N-1} S(n)\exp(+j2\pi nk/N) \qquad k = o,1,\cdots,N-1$$

which can be evaluated by the fast Fourier transform algorithm.[5-6]

A.2 FUNCTIONAL MINIMIZATION

A functional of the form

$$I = \int_a^b \varepsilon^2(x)dx,$$

where

$$\varepsilon(x) = f(x) - \sum_{n=1}^{N} c_n x^{n-1} ,$$

can be minimized over $\{c_n\}$ by requiring that $\partial I/\partial c_m = o$ for $m=1,2,\ldots,N$. This results in the following set of simultaneous equations which can be solved for the c_n:

$$\sum_{n=1}^{N} \gamma_{mn} c_n = q_m, \qquad m = 1,2,\ldots,n,$$

where

$$\gamma_{mn} = x^{m+n-1}/m+n-1 \Big|_a^b$$

$$q_m = \int_a^b x^{m-1} f(x)dx.$$

This procedure is applied to the minimization of

$$I_1 = \int_{-\Delta\omega}^{\Delta\omega} \gamma^2(\omega)d\omega,$$

where $\gamma(\omega)=\psi(\omega)-a_o$, and gives

$$a_o = \frac{1}{2\Delta\omega}\int_{-\Delta\omega}^{\Delta\omega} \psi(\omega)d\omega.$$

Similarly, minimization of

$$I_2 = \int_{-\Delta\omega}^{\Delta\omega} \mu^2(\omega)d\omega,$$

where $\mu(\omega)=\phi(\omega)-b_o-b_1\omega$, gives the following values for b_o and b_1,

$$b_o = \frac{1}{2\Delta\omega}\int_{-\Delta\omega}^{\Delta\omega} \phi(\omega)d\omega,$$

$$b_1 = \frac{3}{2}\frac{1}{(\Delta\omega)3}\int_{-\Delta\omega}^{\Delta\omega} \omega\phi(\omega)d\omega.$$

APPENDIX B

PROGRAMS AND SUBROUTINES

B.1 MAIN PROGRAMS

B.1.1 PROGRAM FOR DISTORTION BOUND

```
C     GASL TRANSMISSION LINE .
C
      DIMENSION M(3)
C
      COMMON /XDEV/PSI(3,513),PHI(3,513),W(513),GAM(3,513),XMU(3,513),
     1 NVAL
      COMMON /BLK1/ A0(3),B0(3),B1(3)
      COMMON /BLK2/ DELF,PI,TWOPI,F0,F2,FMEAN
      COMMON /BLK3/ NPTS
      COMMON /INOUT/ IREAD,IRITE
      COMMON /INPUT/ LZERO,VSWR1,VSWR2,STD,K6
C
      DOUBLE PRECISION PSI,PHI,W,GAM,XMU
      DOUBLE PRECISION A0,B0,B1
      DOUBLE PRECISION DELF,PI,TWOPI,F0,F2,FMEAN
      DOUBLE PRECISION LZERO,VSWR1,VSWR2
      DOUBLE PRECISION S,X,TWPISQ,REC2PI,XINT,DELW,SIGMA,SIGMSQ,COEFF,
     1 POWR,K2WSQ,SWI,Y1(513),VALUE,K2W
C
      EQUIVALENCE (PSI(1,1),Y1(1)),(PHI(1,1),Y2(1))
C     S(X) IS THE TRANSFORM OF THE PULSE ENVELOPE. GAUSSIAN S(X) TAKE
C     THE FORM COEFF * DEXP(-POWR * X**2).
C
      S(X) = COEFF * DEXP(-POWR * X**2)
C
      IREAD = 5
      IRITE = 6
C
      PI=3.14159265D0
      TWOPI=2.D0*PI
      TWPISQ = TWOPI**2
      REC2PI = 1.D0/TWOPI
C
      M(1) = 5
      M(2) = 0
      M(3) = 0
C
C
C     NN = THE NUMBER OF INTERVALS    BE SURE THAT NN IS EVEN
C
      NN = 64
C
C
C
      NINT = NN
      NPTS = NINT+1
      XINT = NINT
C
      READ (IREAD,11) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
   11 FORMAT (4F10.3,I10,D20.7)
C
      SIGMSQ = SIGMA**2
      COEFF = DSQRT(TWOPI) * SIGMA
      POWR = 0.5D0 * SIGMSQ
C
      IREPT = 1
C
```

```
    1 CONTINUE
C
C
      READ (IREAD,33) F0,F2
   33 FORMAT(2D20.7)
C
      FMEAN = (F0 + F2)/2.D0
      DELF = (F2 - F0)/XINT
      DELW = TWOPI*DELF
C
      CALL XX1
      CALL DEV
C
      WRITE(IRITE,111)
  111 FORMAT(1H1,5X,20HDEVIATIONS ARE BELOW //)
      WRITE(IRITE,220)
  220 FORMAT(9X,4H W   ,13X,3HPSI,15X,3HPHI,15X,3HGAM,15X,2HMU)
      DO 500  K=1,3
      WRITE(IRITE,200)
  200 FORMAT(1H0)
      DO 500 I=1,NVAL
      WRITE(IRITE,222) W(I),PSI(K,I),PHI(K,I),GAM(K,I),XMU(K,I)
  222 FORMAT(1X,5(2X,D15.7))
  500 CONTINUE
C
C
      DO 800 K = 1,3
C
      DO 600 I = 1,NVAL
      K2WSQ = DEXP (-2.D0 * GAM(K,I)) + 1.D0 - 2.D0 * DEXP (-GAM(K,I)) *
     1 DCOS (XMU(K,I))
      K2W = DSQRT(K2WSQ)
      SWI = S(W(I))
      Y1(I) = K2WSQ * SWI
  600 CONTINUE
C
      CALL DTRAP (DELW,Y1,VALUE,NVAL)
      IF (K .GT. 1) GO TO 1000
C
      WRITE(IRITE,850) IREPT,NPTS
  850 FORMAT(1H1,3X,9HFOR CASE  I2,3X,29HTHE NUMBER OF POINTS USED IS
     1 I4 //)
C
      WRITE (IRITE,22) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
   22 FORMAT (1H1,3X,5HSTD = F10.5,3X,7HLZERO = F10.4,3X,7HVSWR1 = F10.5
     1,3X,7HVSWR2 = F10.5 / 4X,4HK6 = I5,3X,7HSIGMA = D15.7//)
C
      WRITE(IRITE,44) F0,F2,FMEAN,DELF,DELW
   44 FORMAT (4X,4HF0 = D15.7,3X,4HF2 = D15.7,3X,7HFMEAN = D15.7 /
     1 4X,6HDELF = D15.7,3X,6HDELW = D15.7)
C
 1000 CONTINUE
C
      WRITE (6,900) K, VALUE
  900 FORMAT      (1X,4HK = ,I1,13H, INTEGRAL = ,D15.7)
C
  800 CONTINUE
C
C
      IF (IREPT.EQ.4) STOP
      IREPT = IREPT + 1
      GO TO 1
C
      END
```

B.1.2 PROGRAM FOR RATIO OF DISTORTION ENERGY TO SIGNAL ENERGY

```
C     GASL TRANSMISSION LINE .
C
      DIMENSION M(3)
C
      COMMON /XDEV/PSI(3,513),PHI(3,513),W(513),GAM(3,513),XMU(3,513),
     1 NVAL
      COMMON /BLK1/ A0(3),B0(3),B1(3)
      COMMON /BLK2/ DELF,PI,TWOPI,F0,F2,FMEAN
      COMMON /BLK3/ NPTS
      COMMON /INOUT/ IREAD,IRITE
      COMMON /INPUT/ LZERO,VSWR1,VSWR2,STD,K6
C
      DOUBLE PRECISION PSI,PHI,W,GAM,XMU
      DOUBLE PRECISION A0,B0,B1
      DOUBLE PRECISION DELF,PI,TWOPI,F0,F2,FMEAN
      DOUBLE PRECISION LZERO,VSWR1,VSWR2
      DOUBLE PRECISION S,X,TWPISQ,REC2PI,XINT,DELW,SIGMA,SIGMSQ,COEFF,
     1 POWR,K2WSQ,SWISQ,Y1(513),Y2(513),YNUM,YDENOM,VALUE
C
      EQUIVALENCE (PSI(1,1),Y1(1)),(PHI(1,1),Y2(1))
C     S(X) IS THE TRANSFORM OF THE PULSE ENVELOPE. GAUSSIAN S(X) TAKE
C     THE FORM COEFF * DEXP(-POWR * X**2).
C
      S(X) = COEFF * DEXP(-POWR * X**2)
C
      IREAD = 5
      IRITE = 6
C
      PI=3.14159265D0
      TWOPI=2.D0*PI
      TWPISQ = TWOPI**2
      REC2PI = 1.D0/TWOPI
C
      M(1) = 5
      M(2) = 0
      M(3) = 0
C
C
C     NN = THE NUMBER OF INTERVALS     BE SURE THAT NN IS EVEN
C
      NN = 2**M(1)
C
C
C
      READ (IREAD,11) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
   11 FORMAT (4F10.3,I10,D20.7)
C
      WRITE (IRITE,22) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
   22 FORMAT (1H1,3X,5HSTD = F10.5,3X,7HLZERO = F10.4,3X,7HVSWR1 = F10.5
     1,3X,7HVSWR2 = F10.5 / 4X,4HK6 = I5,3X,7HSIGMA = D15.7//)
C
      SIGMSQ = SIGMA**2
      COEFF = DSQRT(TWOPI) * SIGMA
      POWR = 0.5D0 * SIGMSQ
C
C
      READ (IREAD,33) F0,F2
   33 FORMAT(2D20.7)
C
      FMEAN = (F0 + F2)/2.D0
```

```
      IREPT = 1
C
    1 CONTINUE
      NINT = NN
      NPTS = NINT+1
      XINT = NINT
C
      DELF = (F2 - F0)/XINT
      DELW = TWOPI*DELF
C
      WRITE(IRITE,44) F0,F2,FMEAN,DELF,DELW
   44 FORMAT (4X,4HF0 = D15.7,3X,4HF2 = D15.7,3X,7HFMEAN = D15.7 /
     1 4X,6HDELF = D15.7,3X,6HDELW = D15.7)
C
      CALL XX1
      CALL DEV
C
      WRITE(IRITE,111)
  111 FORMAT(1H1,5X,20HDEVIATIONS ARE BELOW //)
      WRITE(IRITE,220)
  220 FORMAT(9X,4H W  ,13X,3HPSI,15X,3HPHI,15X,3HGAM,15X,2HMU)
      DO 500  K=1,3
      WRITE(IRITE,200)
  200 FORMAT(1H0)
      DO 500 I=1,NVAL
      WRITE(IRITE,222) W(I),PSI(K,I),PHI(K,I),GAM(K,I),XMU(K,I)
  222 FORMAT(1X,5(2X,D15.7))
  500 CONTINUE
C
C
      DO 800 K = 1,3
C
      DO 600 I = 1,NVAL
      K2WSQ = DEXP (-2.D0 * GAM(K,I)) + 1.D0 - 2.D0 * DEXP (-GAM(K,I)) *
     1 DCOS (XMU(K,I))
      SWISQ = S(W(I))**2
      Y1(I) = K2WSQ * SWISQ
      Y2(I) = SWISQ
  600 CONTINUE
C
      CALL DTRAP (DELW,Y1,YNUM,NVAL)
      CALL DTRAP (DELW,Y2,YDENOM,NVAL)
C
      VALUE = YNUM / YDENOM
      IF (K .EQ. 1) WRITE(IRITE,850) IREPT,NPTS
  850 FORMAT(1H1,3X,9HFOR CASE  I2,3X,29HTHE NUMBER OF POINTS USED IS
     1 I4 //)
      WRITE (6,900) K, VALUE
  900 FORMAT     (1X,4HK = ,I1,13H, INTEGRAL = ,D15.7)
C
  800 CONTINUE
C
      NN = 2 * NN
C
      IF (IREPT.EQ.2) STOP
      IREPT = IREPT + 1
      GO TO 1
C
      END
```

B.1.3 PROGRAM FOR ENVELOPE OF OUTPUT USING TRAPEZOIDAL RULE

```
C      GASL TRANSMISSION LINE .
C
       DIMENSION M(3)
C
       COMMON /XDEV/PSI(3,513),PHI(3,513),W(513),GAM(3,513),XMU(3,513),
      1 NVAL
       COMMON /BLK1/ A0(3),B0(3),B1(3)
       COMMON /BLK2/ DELF,PI,TWOPI,F0,F2,FMEAN
       COMMON /BLK3/ NPTS
       COMMON /INOUT/ IREAD,IRITE
       COMMON /INPUT/ LZERO,VSWR1,VSWR2,STD,K6
C
       DOUBLE PRECISION PSI,PHI,W,GAM,XMU
       DOUBLE PRECISION A0,B0,B1
       DOUBLE PRECISION DELF,PI,TWOPI,F0,F2,FMEAN
       DOUBLE PRECISION LZERO,VSWR1,VSWR2
       DOUBLE PRECISION S,X,TWPISQ,REC2PI,XINT,DELW,SIGMA,SIGMSQ,COEFF,
      1 POWR,K2WSQ,SWISQ,Y1(513),Y2(513),YNUM,YDENOM,VALUE
       DOUBLE PRECISION DELT,T,WT,SWI,WTMMUW,FAC,FAC1,FAC2,VALR,VALI,
      1 AMPLI,XN,PHASE
C
       EQUIVALENCE (PSI(1,1),Y1(1)),(PHI(1,1),Y2(1))
C      S(X) IS THE TRANSFORM OF THE PULSE ENVELOPE. GAUSSIAN S(X) TAKE
C      THE FORM COEFF * DEXP(-POWR * X**2).
C
       S(X) = COEFF * DEXP(-POWR * X**2)
C
       IREAD = 5
       IRITE = 6
C
       PI=3.14159265D0
       TWOPI=2.D0*PI
       TWPISQ = TWOPI**2
       REC2PI = 1.D0/TWOPI
C
       M(1) = 6
       M(2) = 0
       M(3) = 0
C
C
C      NN = THE NUMBER OF INTERVALS     BE SURE THAT NN IS EVEN
C
       NN = 2**M(1)
C
C
       READ (IREAD,11) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
    11 FORMAT (4F10.3,I10,D20.7)
C
       SIGMSQ = SIGMA**2
       COEFF = DSQRT(TWOPI) * SIGMA
       POWR = 0.5D0 * SIGMSQ
C
       READ (IREAD,33) F0,F2
```

```
   33 FORMAT(2D20.7)
C
      FMEAN = (F0 + F2)/2.D0
C
      IREPT = 1
C
    1 CONTINUE
      XN = NN
C
      NINT = NN
      NPTS = NINT+1
      XINT = NINT
C
      DELF = (F2 - F0)/XINT
      DELW = TWOPI*DELF
C
      DELT = 0.3D-04/20.D0
C
C     TIME STEP DELT THAT IS ORDINARILY USED IS GIVEN BELOW
C
C     DELT = 1.D0/(XN*DELF)
C
C
C
      CALL XX1
      CALL DEV
C
      WRITE(IRITE,111)
  111 FORMAT(1H1,5X,20HDEVIATIONS ARE BELOW //)
      WRITE(IRITE,220)
  220 FORMAT(9X,4H W   ,13X,3HPSI,15X,3HPHI,15X,3HGAM,15X,2HMU)
      DO 500  K=1,3
      WRITE(IRITE,200)
  200 FORMAT(1H0)
      DO 500 I=1,NVAL
      WRITE(IRITE,222) W(I),PSI(K,I),PHI(K,I),GAM(K,I),XMU(K,I)
  222 FORMAT(1X,5(2X,D15.7))
  500 CONTINUE
C
      MM = NVAL - 1
C
      WRITE (IRITE,22) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
   22 FORMAT (1H1,3X,5HSTD = F10.5,3X,7HLZERO = F10.4,3X,7HVSWR1 = F10.5
     1,3X,7HVSWR2 = F10.5 / 4X,4HK6 = I5,3X,7HSIGMA = D15.7//)
C
      WRITE(IRITE,44) F0,F2,FMEAN,DELF,DELW
   44 FORMAT (4X,4HF0 = D15.7,3X,4HF2 = D15.7,3X,7HFMEAN = D15.7 /
     1 4X,6HDELF = D15.7,3X,6HDELW = D15.7)
C
      DO 800 K = 1,3
C
      WRITE(IRITE,444)
  444 FORMAT(1H1)
      WRITE(IRITE,882)
  882 FORMAT(1H0,31X,15HI N T E G R A L//)
      WRITE(IRITE,884)K
  884 FORMAT(37X,4HK = I1 //)
      WRITE(IRITE,883)
  883 FORMAT(10X,1HT,15X,4HREAL,14X,4HIMAG,15X,9HAMPLITUDE,11X,5HPHASE)
C
      T = 0.D0
C
      DO 825 L = 1,21
C
C     ORDINARILY THE DO STATEMENT IS AS GIVEN BELOW
```

```
C
C     DO 825 L = 1,NN
C
C
C
      DO 600 I = 1,NVAL
      WT = W(I) * T
      SWI = S(W(I))
      WTMMUW = WT-XMU(K,I)
      FAC = DEXP(-GAM(K,I)) * SWI
      FAC1 = DCOS(WTMMUW)
      FAC2 = DSIN(WTMMUW)
      Y1(I) = FAC * FAC1
      Y2(I) = FAC * FAC2
  600 CONTINUE
C
      CALL DTRAP (DELW,Y1,VALR,NVAL)
      CALL DTRAP (DELW,Y2,VALI,NVAL)
C
      VALR = REC2PI * VALR
      VALI = REC2PI * VALI
      AMPLI = DSQRT(VALR**2 + VALI**2)
      PHASE = DATAN2(VALI,VALR)
C
      WRITE(IRITE,885)T,VALR,VALI,AMPLI,PHASE
  885 FORMAT( 3X,D15.7, 3X,D15.7,3H + ,D15.7,3H J , 3X,D15.7,3X,D15.7)
      T = T + DELT
  825 CONTINUE
C
  800 CONTINUE
C
C
      IF (IREPT.EQ.2) STOP
      IREPT = IREPT + 1
      NN = 2 * NN
      GO TO 1
C
      END
```

B.1.4 PROGRAM FOR ENVELOPE OF OUTPUT USING FFT

```
$JOB  D2451B 2 STC     ROYAL-F(T) USING FFT         010 005
$EXECUTE        IBJOB
$IBJOB          FIOCS
$IBFTC MAIN
C
C
C      GASL TRANSMISSION LINE .
C
       DIMENSION INV(128),M(3)
C
       COMMON /XDEV/PSI(3,513),PHI(3,513),W(513),GAM(3,513),XMU(3,513),
      1 NVAL
       COMMON /BLK1/ A0(3),B0(3),B1(3)
       COMMON /BLK2/DELF,PI,TWOPI,F0,F2,FMEAN
       COMMON /BLK3/ NPTS
       COMMON /INOUT/ IREAD,IRITE
       COMMON /INPUT/ LZERO,VSWR1,VSWR2,STD,K6
C
       DOUBLE PRECISION PSI,PHI,W,GAM,XMU
       DOUBLE PRECISION A0,B0,B1
       DOUBLE PRECISION DELF,PI,TWOPI,F0,F2,FMEAN
       DOUBLE PRECISION LZERO,VSWR1,VSWR2
       DOUBLE PRECISION S,X,TWPISQ,REC2PI,XN,XINT,SIGMA,SIGMSQ,COEFF,
      1  POWR,DELT,CMUW,SMUW,FAC,SS(128),A(1024),T,AMPLI,PHASE
C
       EQUIVALENCE (PSI(1,1),A(1))
C
C      S(X) IS THE TRANSFORM OF THE PULSE ENVELOPE. GAUSSIAN S(X) TAKE
C      THE FORM COEFF * DEXP(-POWR * X**2).
C
       S(X) = COEFF * DEXP(-POWR * X**2)
C
       IREAD = 5
       IRITE = 6
       PI=3.14159265D0
       TWOPI=2.D0*PI
       TWPISQ = TWOPI**2
       REC2PI = 1.D0/TWOPI
C
C
C   PREPARE DATA FOR DHARM
C
       M(1) = 6
       M(2) = 0
       M(3) = 0
C
       READ (IREAD,11) STD,LZERO,VSWR1,VSWR2,K6,SIGMA
    11 FORMAT (4F10.3,I10,D20.7)
C
       SIGMSQ = SIGMA**2
       COEFF = DSQRT(TWOPI) * SIGMA
       POWR = 0.5D0 * SIGMSQ
C
       READ (IREAD,33) F0,F2
    33 FORMAT(2D20.7)
C
       FMEAN = (F0 + F2)/2.D0
C
       IREPT = 1
```

```
C
    1 CONTINUE
C
C      NN = THE NUMBER OF INTERVALS   BE SURE THAT NN IS EVEN
C
       NN = 2**M(1)
       XN = NN
       NINT = NN
       NPTS = NINT + 1
       XINT = NINT
C
       DELF = (F2 - F0)/XINT
       DELT = 1.0D0/(XN*DELF)
C
       CALL XX1
       CALL DEV
C
       WRITE(IRITE,111)
  111 FORMAT(1H1,5X,20HDEVIATIONS ARE BELOW //)
       WRITE(IRITE,220)
  220 FORMAT(9X,4H W   ,13X,3HPSI,15X,3HPHI,15X,3HGAM,15X,2HMU)
       DO 500  K=1,3
       WRITE(IRITE,200)
  200 FORMAT(1H0)
       DO 500 I=1,NVAL
       WRITE(IRITE,222) w(I),PSI(K,I),PHI(K,I),GAM(K,I),XMU(K,I)
  222 FORMAT(1X,5(2X,D15.7))
  500 CONTINUE
       WRITE(IRITE,22) SID,LZERO,VSWR1,VSWR2,K6,SIGMA
   22 FORMAT(1H1,3X,5HSID = F10.5,3X,7HLZERO = F10.4,3X,7HVSWR1 = F10.5,
     1 3X,7HVSWR2 = F10.5 / 4X,4HK6 = I5,3X,7HSIGMA = D15.7//)
       WRITE(IRITE,44) F0,F2,FMEAN,DELF
   44 FORMAT (4X,4HF0 = D15.7,3X,4HF2 = D15.7,3X,7HFMEAN = D15.7 /
     1 4X,6HDELF = D15.7)
C
C  SET UP VALUES FOR F(w)
C
       DO 800 K=1,3
       WRITE(IRITE,880)
  880 FORMAT(1H1)
       WRITE(IRITE,882)
  882 FORMAT(1H0,31X,15HI N T E G R A L//)
       WRITE(IRITE,884)K
  884 FORMAT(37X,4HK = I1 //)
       WRITE(IRITE,883)
  883 FORMAT(10X,1HT,15X,4HREAL,14X,4HIMAG,15X,9HAMPLITUDE,11X,5HPHASE)
C
C
       DO 900 I = 1,NN
       CMUw = DCOS(XMU(K,I))
       SMUw = DSIN(XMU(K,I))
       FAC = DEXP(-GAM(K,I)) * S(w(I))
       A(2*I - 1) = FAC * CMUw
       A(2*I) = - FAC * SMUw
  900 CONTINUE
C
C
       CALL DHARM (A,M,INV,SS, 1,IFERR)
C
       IF (IFERR.NE.0) WRITE (IRITE,999) IFERR,K
  999 FORMAT(//3X,12HERROR IN FFT,5X,8HIFERR = ,I3,3X,4HK = ,I3 //)
C
C
       NNDBLE = 2 * NN
       T = 0.0D0
       KNT = 0
```

```
C
      DO 700 L=1,NNDBLE,2
      KNT = KNT+1
C
      IF(MOD(KNT,2).EQ.0) GO TO 1500
      A(L) = DELF * A(L)
      A(L+1) = DELF * A(L+1)
C
      GO TO 1600
C
 1500 CONTINUE
C
      A(L) = -DELF*A(L)
      A(L+1) = -DELF*A(L+1)
 1600 CONTINUE
      AMPLI = DSQRT (A(L)**2 + A(L+1)**2)
      PHASE = DATAN2(A(L+1),A(L))
      WRITE(IRITE,885)T,A(L),A(L+1),AMPLI,PHASE
  885 FORMAT( 3X,D15.7, 3X,D15.7,3H + ,D15.7,3H J , 3X,D15.7,3X,D15.7)
      T = T + DELT
  700 CONTINUE
C
  800 CONTINUE
C
C
      IF (IREPT . EQ. 1) STOP
      M(1) = M(1)+1
      IREPT = IREPT + 1
      GO TO 1
C
      END
```

B.2 SUBROUTINES COMMON TO ALL MAIN PROGRAMS

B.2.1 XXI (written by NMSU)

```
      SUBROUTINE XXI
      COMMON  /NETYCM/BETAL,COSHAL,SINHAL,L,AR(3),AI(3),Z0,BETA,
     1 ALPHA,YI,CUML,YR1,YI1,YR,RAN(301),K3
      COMMON/MATXCM/ GR(4),GI(4),RR(4),RI(4),ER(4),EI(4)
      COMMON /XDEV/PSI(3,513),PHI(3,513),W(513),GAM(3,513),XMU(3,513),
     1 NVAL
      COMMON /BLK2/DELF,PI,TWOPI,F0,F2,FMEAN
      COMMON /BLK3/ NPTS
      COMMON /INOUT/ IREAD,IRITE
      COMMON /INPUT/ LZERO,VSWR1,VSWR2,STD,K6
C     PN IS AN ARRAY OF 15 RANDOM NUMBERS USED AS INPUT TO A RANDOM
C  NUMBER GENERATOR   ( REF: EMPIRICAL TESTS OF AN ADDITIVE RANDOM NUMBER
C  GENERATOR -BERT F. GREEN ET AL ,JOURNAL OF THE ASSOCIATION FOR COM-
C  PUTING MACHINERY, OCT. 1959, VOL. 5 ,NO. 4).
      INTEGER  PN(17)
      REAL MAXD(4),MIND(4),MAXT(4),MINT(4)
      DIMENSION DBS(3,513),TAS(3,513)
      DOUBLE PRECISION AR,AI,ER,EI,GR,GI,RR,RI,FR,FI,HR,
     1 HI,ALPHA,COSHAL,SINHAL,BETAL,L,Z0,YI,FREQ,BETA,
     2 PI,TWOPI,YR1,YR2,VSWR1,VSWR2,Y0,F0,F1,F2,DELF,DBMI,YI1,YI2,
     3 CUML,YR,W,YRR,ZS,DENOM,EREGR,EREGI,EREGA,XPSI,PSI,DB,EREGP,
     4 DIFPH,PHI,TAU,TAU1,TAU0,TOTL,DBRF,STPH(4),COMPH(4),
     5 GAM,XMU,FMEAN,LZERO
      DATA PN/49319,88786,84866,11849,54966,10959,22784,86037,
     1 72751,79241,43593,29522,88836,65905,98552/
      DATA STPH/4*0.0D0/,COMPH/4*0.0D0/
C       K6 IS THE NUMBER OF LINE SECTIONS.
C       THE NEXT GROUP OF STATEMENTS GENERATES A RANDOM NUMBER ,UNIFORM-
C        LY DISTRIBUTED IN THE INTERVAL FROM  0 TO 99999.
      DO 101  I = 1,K6
      DO 102  J = 1,15
      MJP18 = 18-J
      MJP16 = 16-J
  102 PN(MJP18) = PN(MJP16)
      PN(2) =  PN(3) + PN(17)
      IF (PN(2).GT.99999) PN(2) = PN(2)- 100000
      PN(1) =  PN(2) + PN(16)
      IF (PN(1).GT.99999)  PN(1) = PN(1) - 100000
C       THE NEXT GROUP OF STATEMENTS USES AN ANALYTICAL APPROXIMATION TO
C  CONVERT THE UNIFORMLY DISTRIBUTED NUMBER TO A GAUSSIAN DISTRIBUTED
C  RANDOM NUMBER WITH MEAN ZERO (REF. HANDBOOK OF MATHEMATICAL FUNCT-
C  IONS - ABRAMOWITZ , DOVER 1965 , P. 933 ).
      IF (PN(1).GE.50000)  GO TO 300
      AN = FLOAT(PN(1))/1.E5
      ANS = 1.
      GO TO 301
  300 AN = FLOAT(PN(1) - 49999)/1.E5
      ANS = -1.
  301 TNS = ALOG(1./(AN*AN))
      TN = SQRT(TNS)
C   USE STD    AS    STANDARD DEVIATION FOR THE GAUSSIAN DISTRIBUTION.
  101 RAN(I)=STD   *ANS*(TN-(2.30753 +  0.27061*TN)/(1. + 0.99229*TN +
     1 0.04481*TNS))
C
      WRITE(IRITE,11)
   11 FORMAT(1H1,5X,14HRANDOM NUMBERS //)
C
```

```
      WRITE (IRITE,200) (RAN(I),I = 1,K6)
  200 FORMAT(6X,6E12.4)
C        YRR IS LINE TERMINATION ADMITTANCE.
C        ZS IS GENERATOR SOURCE RESISTANCE .
C        Z0 IS LINE CHARACTERISTIC IMPEDANCE .
C        Y0 IS LINE CHARACTERISTIC ADMITTANCE (1 IF Z0 IS 1) .
C        YR1 IS REAL PART OF TYPE 1 DISCONTINUITY ADMITTANCE.
C        YR2 IS REAL PART OF TYPE 2 DISCONTINUITY ADMITTANCE.
C        K5 IS NUMBER OF TIMES FORMULATED LINE IS DOUBLED, PLUS 1.
C        VSWR1 IS THE UNIT VSWR ASSOCIATED WITH THE TYPE 1 REFLECTION
C          DISCONTINUITY.
C        VSWR2 IS THE UNIT VSWR ASSOCIATED WITH THE TYPE 2 REFLECTION
C          DISCONTINUITY .
C        F0 IS THE INITIAL VALUE FOR THE FREQUENCY ITERATION.
C        F1 IS A REFERENCE FREQUENCY FOR CALCULATING THE REFLECTION
C          ADMITTANCE.
C        F2 IS THE FINAL VALUE FOR THE FREQUENCY ITERATION.
C        FREQ IS THE RADIO FREQUENCY VARIABLE.
C        DELF IS THE RADIO FREQUENCY INCREMENT .
C        DBMI IS THE LINE ATTENUATION IN DB PER MILE .
C        K2 IS THE NUMBER OF LINE ELEMENTS ASSEMBLED .
C        YR IS THE REAL PART OF THE DISCONTINUITY ADMITTANCE.
C        YI IS THE IMAGINARY PART OF THE DISCONTINUITY ADMITTANCE .
C        YI1 IS THE IMAGINARY PART OF THE TYPE 1 DISCONTINUITY ADMIT-
C          TANCE.
C        YI2 IS THE IMAGINARY PART OF THE TYPE 2 DISCONTINUITY ADMIT-
C          TANCE .
C        ALPHA IS THE ATTENUATION CONSTANT IN NEPERS PER FOOT.
C        BETA IS THE PHASE CONSTANT IN RADIANS PER FOOT.
C        CUML IS THE CUMULATIVE LENGTH OF THE FORMULATED LINE .
      YR2=0.D0
      K5=3
      F1=3.6D9
      K1=0
      Y0=1.D0
      YRR = 1.0D0
      ZS = 1.0D0
      Z0 = 1.D0
      YR1 = 0.D0
      YR = 0.D0
      YI  = 0.D0
      FREQ = F0 - DELF
  302 FREQ = FREQ + DELF
      K1 = K1 + 1
      W(K1) = TWOPI * (FREQ - FMEAN)
      YI1 = FREQ*DSQRT(Y0*(YR1+Y0)*(VSWR1-2.0D0+1.0D0/VSWR1)-YR1*YR1)/F1
      YI2 = FREQ*DSQRT(Y0*(YR2+Y0)*(VSWR2-2.0D0+1.0D0/VSWR2)-YR2*YR2)/F1
C       ALPHA CURVE IS FROM GASL DATA AND BETA CURVE IS FROM GASL DATA
C         FOR FREQUENCY RANGE 3.60 TO 3.75 GHZ.
      ALPHA = 8.77287D-4 * (1.0D-9*FREQ) - 2.930142D-3
      BETA = (3.5136D-1 * (1.0D-9 * FREQ) + 5.19065)*(1.0D-9 * FREQ)
      DBMI = 4.02338D-8 * FREQ - 1.34381D2
      S1 = 0
      WRITE (IRITE,201)  FREQ,VSWR1,VSWR2,YI1,YI2,DBMI,ALPHA,BETA
  201 FORMAT(1H0,D17.9,7D12.4/1X)
      K3 =0
      K4 =0
      CUML = 0.D0
  303 K3 = K3 + 1
      K4 = K4 + 1
      CALL NETY
      IF (S1.EQ.1.) GO TO 304
      DO 104 I = 1,4
      GR(I) =ER(I)
  104 GI(I) =EI(I)
      S1 = 1.0
```

```
      GO TO 303
  304 CALL MATX
      DO 105 I = 1,4
      GR(I) = RR(I)
  105 GI(I) = RI(I)
C       EVERY 6TH FLANGE JOINT IS AN EXPANSION JOINT.
      IF (K4.LT.6) GO TO 106
      K4 = 0
      YI = YI2
      YR = YR2
  106 IF  (K3.LT.K6)  GO TO 303
      K = 1
      K2 = 0
      TAU0 = 0.D0
C   THE NEXT 4 STATEMENTS FORM THE TRANSFER FUNCTION FOR THE LINE TERMI-
C     NATION CONDITIONS WHERE FR AND FI ARE THE REAL AND IMAGINARY PARTS
C     OF THE NUMERATOR AND HR AND HI ARE THE REAL AND IMAGINARY PARTS
C     OF THE DENOMINATOR .
  305 FR = RR(1)*RR(4) - RI(1)*RI(4) - RR(2)*RR(3) + RI(2)*RI(3)
      FI = RR(1)*RI(4) + RI(1)*RR(4) - RR(2)*RI(3) - RI(2)*RR(3)
      HR = ZS*RR(3) + RR(1) + ZS*YRR*RR(4) + YRR*RR(2)
      HI = ZS*RI(3) + RI(1) + ZS*YRR*RI(4) + YRR*RI(2)
      DENOM = HR*HR + HI*HI
C   EREGR AND EREGI ARE THE REAL AND IMAGINARY PARTS RESPECTIVELY OF THE
C     TRANSFER FUNCTION , AND DB IS THE RELATIVE AMPLITUDE IN DB .
      EREGR = (FR*HR + FI*HI)/DENOM
      EREGI = (-FR*HI + FI*HR)/DENOM
      EREGA = DSQRT(EREGR*EREGR + EREGI*EREGI)
      XPSI=DLOG(EREGA)
      PSI(K,K1)=-XPSI
      DB = 8.68588D0*DLOG(0.5D0/EREGA)
C     EREGP IS THE PHASE OF THE TRANSFER FUNCTION , AND DIFPH IS THE
C      PHASE CHANGE IN RADIANS SINCE THE PREVIOUS ITERATION FOR FREQUEN-
C       CY .
      EREGP = DATAN2(EREGI,EREGR)
      DIFPH = EREGP - STPH(K)
      STPH(K) = EREGP
C      THE NEXT STATEMENT CORRECTS FOR PHASE CYCLE AMBIGUITY ON SHORT
C        LINE SEGMENTS .
      IF (DIFPH.GT.0.D0)  DIFPH = DIFPH - TWOPI
      COMPH(K) = COMPH(K) + DIFPH
      PHI(K,K1)=-COMPH(K)
C       TAU IS THE ENVELOPE DELAY IN SECONDS .
      TAU = -DIFPH/(DELF*TWOPI)
C       THE NEXT 5 STATEMENTS CORRECT FOR PHASE CYCLE AMBIGUITY FOR
C         LINE DOUBLING .
      TWOK2 = FLOAT(2**K2)
      TAU1 = TAU + DBLE(TWOK2)/DELF
      IF ((DABS(TAU1-2.D0*TAU0)) .GE. (DABS(TAU - 2.D0*TAU0))) GO TO 306
      TAU = TAU1
      K2 = K2 + 1
  306 TAU0 = TAU
C       THE NEXT 3 STATEMENTS PLACE THE DATA IN ARRAY STORAGE .
      DBS(K,K1) = DB
      TAS(K,K1) = TAU*1.0D6
C  NSEC IS THE NUMBER OF FORMULATED LINE SEGMENTS OBTAINED BY DOUBLING,
C     AND TOTL IS THE TOTAL LENGTH OF LINE UNDER LINE DOUBLING .
      NSEC = 2**(K - 1)
      FNSEC = FLOAT(NSEC)
      TOTL = CUML * DBLE(FNSEC)
C   THE NEXT STATEMENT IS A TEST OF ADEQUATE PRECISION FOR THE LENGTH
C     OF LINE AND THE PARTICULAR INPUT CONDITIONS . FR IS NORMALLY
C     UNITY . IF PRECISION IS INADEQUATE, THE PRINT FORMAT IS ALTERED .
      IF (DABS(FR -1.D0).LE.1.D-3) GO TO 307
      WRITE (IRITE,202) FREQ,NSEC
  202 FORMAT(1H0,1X,D13.6,76X,I9)
```

```
      GO TO 308
  307 WRITE(IRITE,203) EREGA,EREGP,DB,EREGR,EREGI,COMPH(K),TAU,K3,NSEC
  203 FORMAT(2X,D12.4,F9.3,F12.5,4D14.6,I6,1X,I3)
  308 IF (K.GE.K5)  GO TO 309
      K = K + 1
C       THE FOLLOWING LOOP SETS INITIAL VALUES FOR LINE DOUBLING .
      DO 108 I = 1,4
      GR(I) = RR(I)
      GI(I) = RI(I)
      ER(I) = RR(I)
  108 EI(I) = RI(I)
      CALL MATX
      GO TO 305
C       THE NEXT STATEMENT IS THE BRANCH POINT FOR FREQUENCY ITERATIONS.
  309 IF (K1.LT.NPTS ) GO TO 302
      DBRF = TOTL*DBMI/5280.D0
      WRITE (IRITE,405)  CUML,TOTL,DBRF,DELF
  405 FORMAT (1H0,11X,4D12.4)
C       WE NEXT USE 2 DO LOOPS TO FIND THE MAXIMA AND MINIMA OF THE
C    DB AND ENVELOPE DELAY, COMPH DECREASES MONOTONICALLY.
      DO 107 I = 1,K5
      MAXD(I) = DBS(I,1)
      MIND(I) = MAXD(I)
      MAXT(I) = TAS(I,2)
      MINT(I) = MAXT(I)
      DO 107  J = 2,K1
      IF (MAXD(I).LT.DBS(I,J)) MAXD(I) = DBS(I,J)
      IF (MIND(I).GT.DBS(I,J)) MIND(I) = DBS(I,J)
      IF (MAXT(I).LT.TAS(I,J)) MAXT(I) = TAS(I,J)
      IF (MINT(I).GT.TAS(I,J)) MINT(I) = TAS(I,J)
  107 CONTINUE
      NVAL=K1
      WRITE (IRITE,205) (MAXD(J),J = 1,K5)
  205 FORMAT (1H0,11X,4E12.4)
      WRITE (IRITE,205) (MIND(J),J = 1,K5)
      WRITE (IRITE,205) (MAXT(J),J= 1,K5)
      WRITE (IRITE,205) (MINT(J),J = 1,K5)
      RETURN
      END
```

B.2.2 NETY (written by NMSU)

```
      SUBROUTINE NETY
      COMMON /INPUT/ LZERO,VSWR1,VSWR2,STD,K6
      COMMON  /NETYCM/BETAL,COSHAL,SINHAL,L,AR(3),AI(3),Z0,BETA,
     1 ALPHA,YI,CUML,YR1,YI1,YR,RAN(301),K3
      COMMON/MATXCM/ GR(4),GI(4),RR(4),RI(4),ER(4),EI(4)
      DOUBLE PRECISION BETAL, COSHAL,SINHAL,L,AR,AI,ER,EI,Z0,BETA,ALPHA,
     1  YI,SNBETL,CSBETL,GR,GI,RR,RI,CUML,YR,YR1,YI1,ARG,XPOPOS,XPONEG
      DOUBLE PRECISION LZERO,VSWR1,VSWR2
C       WE FORM A NEW VALUE OF THE LINE LENGTH CONSISTING OF 30 FEET
C          PLUS  THE RANDOM ELEMENT .
      L = LZERO + DBLE (RAN(K3))
      CUML =  CUML + L
      BETAL = BETA*L
      ARG = ALPHA*L
      XPOPOS = DEXP(ARG)
      XPONEG = DEXP(-ARG)
      COSHAL = 0.5D0 * (XPOPOS + XPONEG)
      SINHAL = 0.5D0 * (XPOPOS - XPONEG)
      SNBETL = DSIN(BETAL)
      CSBETL = DCOS(BETAL)
C        THE NEXT GROUP OF STATEMENTS CALCULATES THE REAL AND IMAGINARY
C          GENERAL CIRCUIT CONSTANTS FOR THE LENGTH OF TRANSMISSION LINE,L
      AR(1) =  COSHAL*CSBETL
      AI(1) =  SINHAL*SNBETL
      AR(2) = SINHAL*CSBETL*Z0
      AI(2) = COSHAL*SNBETL*Z0
      AR(3) = AR(2)/(Z0*Z0)
      AI(3) = AI(2)/(Z0*Z0)
C       THE NEXT GROUP OF STATEMENTS CALCULATES THE REAL AND IMAGINARY
C          GENERAL CIRCUIT CONSTANTS FOR THE LINE ELEMENT PLUS DISCONTI-
C          NUITY ADMITTANCE .
      ER(1) = AR(1)  + AR(2)*YR -AI(2)*YI
      EI(1) = AI(1)  + AI(2)*YR +AR(2)*YI
      ER(2) = AR(2)
      EI(2) = AI(2)
      ER(3) = AR(3)  + AR(1)*YR - AI(1)*YI
      EI(3) = AI(3)  + AI(1)*YR + AR(1)*YI
      ER(4) = AR(1)
      EI(4) = AI(1)
      YI = YI1
      YR = YR1
      RETURN
      END
```

B.2.3 MATX (written by NMSU)

```
      SUBROUTINE MATX
C     THIS SUBROUTINE PERFORMS THE MATRIX MULTIPLICATION ,R = E X G,WHICH
C       CASCADES THE E AND G NETWORK ELEMENTS .
      COMMON/MATXCM/ GR(4),GI(4),RR(4),RI(4),ER(4),EI(4)
      DOUBLE PRECISION ER,EI,GR,GI,RR,RI
      RR(1) = ER(1)*GR(1) - EI(1)*GI(1) + ER(2)*GR(3) - EI(2)*GI(3)
      RR(2) = ER(1)*GR(2) - EI(1)*GI(2) + ER(2)*GR(4) - EI(2)*GI(4)
      RR(3) = ER(3)*GR(1) - EI(3)*GI(1) + ER(4)*GR(3) - EI(4)*GI(3)
      RR(4) = ER(3)*GR(2) - EI(3)*GI(2) + ER(4)*GR(4) - EI(4)*GI(4)
      RI(1) = EI(1)*GR(1) + ER(1)*GI(1) + EI(2)*GR(3) + ER(2)*GI(3)
      RI(2) = EI(1)*GR(2) + ER(1)*GI(2) + EI(2)*GR(4) + ER(2)*GI(4)
      RI(3) = EI(3)*GR(1) + ER(3)*GI(1) + EI(4)*GR(3) + ER(4)*GI(3)
      RI(4) = EI(3)*GR(2) + ER(3)*GI(2) + EI(4)*GR(4) + ER(4)*GI(4)
      RETURN
      END
```

B.2.4 DEV

```
      SUBROUTINE DEV
      COMMON /XDEV/PSI(3,513),PHI(3,513),W(513),GAM(3,513),XMU(3,513),
     1 NVAL
      COMMON /BLK1/ A0(3),B0(3),B1(3)
      DIMENSION PSISUM(3),PHISUM(3),PHIWSM(3)
      DIMENSION PHIBAR(3)
      DOUBLE PRECISION PSI,PHI,W,GAM,XMU,PSISUM,PHIWSM,A0,B0,B1,PHIBAR,
     1 WSUM,WSQSUM,XVAL,WBAR,PSIBAR,PHISUM
      WSUM=0.0D0
      WSQSUM=0.0D0
      DO 50 K=1,3
      PSISUM(K)=0.0D0
      PHIWSM(K)=0.0D0
      PHISUM(K)=0.0D0
   50 CONTINUE
      DO 75 I=1,NVAL
      WSUM=WSUM+W(I)
      WSQSUM=WSQSUM+W(I)*W(I)
   75 CONTINUE
      DO 150 K=1,3
      DO 100 I=1,NVAL
      PSISUM(K)=PSISUM(K)+PSI(K,I)
      PHIWSM(K)=PHIWSM(K)+PHI(K,I)*W(I)
      PHISUM(K)=PHISUM(K)+PHI(K,I)
  100 CONTINUE
  150 CONTINUE
      XVAL=NVAL
      WBAR=WSUM/XVAL
      DO 200 K=1,3
      PSIBAR=PSISUM(K)/XVAL
      A0(K)=PSIBAR
      PHIBAR(K)=PHISUM(K)/XVAL
      B1(K)=(XVAL*PHIWSM(K)-PHISUM(K)*WSUM)/(XVAL*WSQSUM-WSUM*WSUM)
      B0(K)=PHIBAR(K)-B1(K)*WBAR
  200 CONTINUE
      DO 350 K=1,3
      DO 300 I=1,NVAL
      GAM(K,I)=PSI(K,I)-A0(K)
      XMU(K,I)=PHI(K,I)-B0(K)-B1(K)*W(I)
  300 CONTINUE
  350 CONTINUE
      RETURN
      END
```

B.3 SUBROUTINES FOR INTEGRATION AND FOURIER INVERSION

B.3.1 DTRAP

```
C     TRAPEZOIDAL RULE
C
C
      DIMENSION Y(1)
C
      DOUBLE PRECISION H,Y,VALINT,SUM
C
      SUM = 0.5 * (Y(1) + Y(N))
C
      NM1 = N - 1
C
C
      DO 10 I = 2,NM1
      SUM = SUM + Y(I)
   10 CONTINUE
C
C
      VALINT = H * SUM
C
      RETURN
      END
```

B.3.2 DSIMP

```
      SUBROUTINE DSIMP(H,Y,VALINT,N)
C
C     MODIFIED SIMPSON'S RULE FOR N INTERVALS
C             IF N IS EVEN    REGULAR SIMPSON'S RULE
C             IF N IS ODD     MODIFIED SIMPSON'S RULE
C
      DIMENSION Y(1)
      DOUBLE PRECISION H,Y,SUM1,SUM2,SUM3,VALINT
C
      IREM = N - 2 * (N/2)
      IF (IREM.NE.0) GO TO 5
      M = N
      GO TO 6
    5 M = N - 1
    6 CONTINUE
C
      SUM1 = Y(1) + Y(M+1)
C
      SUM2 = 0.0D0
C
C
      DO 10 I = 2,M,2
      SUM2 = SUM2 + Y(I)
   10 CONTINUE
C
C
      SUM3 = 0.0D0
C
      MM1 = M-1
C
C
      DO 20 I= 3,MM1,2
      SUM3 = SUM3+Y(I)
   20 CONTINUE
C
C
      VALINT = (H/3.D0)*(SUM1+4.D0*SUM2+2.D0*SUM3)
C
      IF (IREM.EQ.0) RETURN
      VALINT = VALINT + (H/2.D0) * (Y(N) +Y(N+1))
C
      RETURN
      END
```

B.3.3 DHARM (ADAPTED FROM IBM SCIENTIFIC SUBROUTINE PACKAGE)

```
C        PURPOSE
C           PERFORMS DISCRETE COMPLEX FOURIER TRANSFORMS ON A COMPLEX
C           DOUBLE PRECISION,THREE DIMENSIONAL ARRAY
C        DESCRIPTION OF PARAMETERS
C           A      - A DOUBLE PRECISION VECTOR
C                    AS INPUT, A CONTAINS THE COMPLEX, 3-DIMENSIONAL
C                    ARRAY TO BE TRANSFORMED.  THE REAL PART OF
C                    A(I1,I2,I3) IS STORED IN VECTOR FASHION IN A CELL
C                    WITH INDEX 2*(I3*N1*N2 + I2*N1 + I1) + 1 WHERE
C                    NI = 2**M(I), I=1,2,3 AND I1 = 0,1,...,N1-1 ETC.
C                    THE IMAGINARY PART IS IN THE CELL IMMEDIATELY
C                    FOLLOWING.
C           M      - A THREE CELL VECTOR WHICH DETERMINES THE SIZES
C                    OF THE 3 DIMENSIONS OF THE ARRAY A.   THE SIZE,
C                    NI, OF THE I DIMENSION OF A IS 2**M(I), I = 1,2,3
C        REMARKS
C           THIS SUBROUTINE IS TO BE USED FOR COMPLEX, DOUBLE PRECISION,
C           3-DIMENSIONAL ARRAYS IN WHICH EACH DIMENSION IS A POWER OF
C           2. THE MAXIMUM M(I) MUST NOT BE LESS THAN 3 OR GREATER THAN
C           20, I = 1,2,3.
      SUBROUTINE DHARM(A,M,INV,S,IFSET,IFERR)
      DIMENSION A(1),INV(1),S(1),N(3),M(3),NP(3),W(2),W2(2),W3(2)
      DOUBLE PRECISION A,R,W3,AWI,THETA,ROOT2,S,T,W,W2,FN,AWR
      EQUIVALENCE (N1,N(1)),(N2,N(2)),(N3,N(3))
   10 IF( IABS(IFSET) - 1) 900,900,12
   12 MTT=MAX0(M(1),M(2),M(3)) -2
      ROOT2=DSQRT(2.0D0)
      IF (MTT-MT ) 14,14,13
   13 IFERR=1
      RETURN
   14 IFERR=0
      M1=M(1)
      M2=M(2)
      M3=M(3)
      N1=2**M1
      N2=2**M2
      N3=2**M3
   16 IF(IFSET) 18,18,20
   18 NX= N1*N2*N3
      FN = NX
      DO 19 I = 1,NX
      A(2*I-1) = A(2*I-1)/FN
   19 A(2*I) = -A(2*I)/FN
   20 NP(1)=N1*2
      NP(2)= NP(1)*N2
      NP(3)=NP(2)*N3
      DO 250 ID=1,3
      IL = NP(3)-NP(ID)
      IL1 = IL+1
      MI = M(ID)
      IF (MI)250,250,30
   30 IDIF=NP(ID)
      KBIT=NP(ID)
      MEV = 2*(MI/2)
      IF (MI - MEV )60,60,40
C
C     M IS ODD. DO L=1 CASE
   40 KBIT=KBIT/2
      KL=KBIT-2
      DO 50 I=1,IL1,IDIF
      KLAST=KL+I
```

```
      DO 50 K=1,KLAST,2
      KD=K+KBIT
C
C     DO ONE STEP WITH L=1,J=0
C     A(K)=A(K)+A(KD)
C     A(KD)=A(K)-A(KD)
C
      T=A(KD)
      A(KD)=A(K)-T
      A(K)=A(K)+T
      T=A(KD+1)
      A(KD+1)=A(K+1)-T
   50 A(K+1)=A(K+1)+T
      IF (MI - 1)250,250,52
   52 LFIRST =3
C
C     DEF - JLAST = 2**(L-2) -1
      JLAST=1
      GO TO 70
C
C     M IS EVEN
   60 LFIRST = 2
      JLAST=0
   70 DO 240 L=LFIRST,MI,2
      JJDIF=KBIT
      KBIT=KBIT/4
      KL=KBIT-2
C
C     DO FOR J=0
      DO 80 I=1,IL1,IDIF
      KLAST=I+KL
      DO 80 K=I,KLAST,2
      K1=K+KBIT
      K2=K1+KBIT
      K3=K2+KBIT
C
C     DO TWO STEPS WITH J=0
C     A(K)=A(K)+A(K2)
C     A(K2)=A(K)-A(K2)
C     A(K1)=A(K1)+A(K3)
C     A(K3)=A(K1)-A(K3)
C
C     A(K)=A(K)+A(K1)
C     A(K1)=A(K)-A(K1)
C     A(K2)=A(K2)+A(K3)*I
C     A(K3)=A(K2)-A(K3)*I
C
      T=A(K2)
      A(K2)=A(K)-T
      A(K)=A(K)+T
      T=A(K2+1)
      A(K2+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
C
      T=A(K3)
      A(K3)=A(K1)-T
      A(K1)=A(K1)+T
      T=A(K3+1)
      A(K3+1)=A(K1+1)-T
      A(K1+1)=A(K1+1)+T
C
      T=A(K1)
      A(K1)=A(K)-T
      A(K)=A(K)+T
      T=A(K1+1)
      A(K1+1)=A(K+1)-T
```

```
      A(K+1)=A(K+1)+T
C
      R=-A(K3+1)
      T = A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
   80 A(K2+1)=A(K2+1)+T
      IF (JLAST) 235,235,82
   82 JJ=JJDIF    +1
C
C     DO FOR J=1
      ILAST= IL +JJ
      DO 85 I = JJ,ILAST,IDIF
      KLAST = KL+I
      DO 85 K=I,KLAST,2
      K1 = K+KBIT
      K2 = K1+KBIT
      K3 = K2+KBIT
C
C     LETTING W=(1+I)/ROOT2,W3=(-1+I)/ROOT2,W2=I,
C     A(K)=A(K)+A(K2)*I
C     A(K2)=A(K)-A(K2)*I
C     A(K1)=A(K1)*W+A(K3)*W3
C     A(K3)=A(K1)*W-A(K3)*W3
C
C     A(K)=A(K)+A(K1)
C     A(K1)=A(K)-A(K1)
C     A(K2)=A(K2)+A(K3)*I
C     A(K3)=A(K2)-A(K3)*I
C
      R =-A(K2+1)
      T = A(K2)
      A(K2) = A(K)-R
      A(K) = A(K)+R
      A(K2+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
C
      AWR=A(K1)-A(K1+1)
      AWI = A(K1+1)+A(K1)
      R=-A(K3)-A(K3+1)
      T=A(K3)-A(K3+1)
      A(K3)=(AWR-R)/ROOT2
      A(K3+1)=(AWI-T)/ROOT2
      A(K1)=(AWR+R)/ROOT2
      A(K1+1)=(AWI+T)/ROOT2
      T= A(K1)
      A(K1)=A(K)-T
      A(K)=A(K)+T
      T=A(K1+1)
      A(K1+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
      R=-A(K3+1)
      T=A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
   85 A(K2+1)=A(K2+1)+T
      IF(JLAST-1) 235,235,90
   90 JJ= JJ + JJDIF
C
C     NOW DO THE REMAINING J S
      DO 230 J=2,JLAST
C
C     FETCH W S
C     DEF- W=W**INV(J), W2=W**2, W3=W**3
```

```
   96 I=INV(J+1)
   98 IC=NT-I
      W(1)=S(IC)
      W(2)=S(I)
      I2=2*I
      I2C=NT-I2
      IF(I2C)120,110,100
C
C     2*I IS IN FIRST QUADRANT
  100 W2(1)=S(I2C)
      W2(2)=S(I2)
      GO TO 130
  110 W2(1)=0.
      W2(2)=1.
      GO TO 130
C
C     2*I IS IN SECOND QUADRANT
  120 I2CC = I2C+NT
      I2C=-I2C
      W2(1)=-S(I2C)
      W2(2)=S(I2CC)
  130 I3=I+I2
      I3C=NT-I3
      IF(I3C)160,150,140
C
C     I3 IN FIRST QUADRANT
  140 W3(1)=S(I3C)
      W3(2)=S(I3)
      GO TO 200
  150 W3(1)=0.
      W3(2)=1.
      GO TO 200
C
  160 I3CC=I3C+NT
      IF(I3CC)190,180,170
C
C     I3 IN SECOND QUADRANT
  170 I3C=-I3C
      W3(1)=-S(I3C)
      W3(2)=S(I3CC)
      GO TO 200
  180 W3(1)=-1.
      W3(2)=0.
      GO TO 200
C
C     3*I IN THIRD QUADRANT
  190 I3CCC=NT+I3CC
      I3CC = -I3CC
      W3(1)=-S(I3CCC)
      W3(2)=-S(I3CC)
  200 ILAST=IL+JJ
      DO 220 I=JJ,ILAST,IDIF
      KLAST=KL+I
      DO 220 K=I,KLAST,2
      K1=K+KBIT
      K2=K1+KBIT
      K3=K2+KBIT
C
C     DO TWO STEPS WITH J NOT 0
C     A(K)=A(K)+A(K2)*W2
C     A(K2)=A(K)-A(K2)*W2
C     A(K1)=A(K1)*W+A(K3)*W3
C     A(K3)=A(K1)*W-A(K3)*W3
C
C     A(K)=A(K)+A(K1)
C     A(K1)=A(K)-A(K1)
```

```
C      A(K2)=A(K2)+A(K3)*I
C      A(K3)=A(K2)-A(K3)*I
C
       R=A(K2)*W2(1)-A(K2+1)*W2(2)
       T=A(K2)*W2(2)+A(K2+1)*W2(1)
       A(K2)=A(K)-R
       A(K)=A(K)+R
       A(K2+1)=A(K+1)-T
       A(K+1)=A(K+1)+T
C
       R=A(K3)*W3(1)-A(K3+1)*W3(2)
       T=A(K3)*W3(2)+A(K3+1)*W3(1)
       AWR=A(K1)*W(1)-A(K1+1)*W(2)
       AWI=A(K1)*W(2)+A(K1+1)*W(1)
       A(K3)=AWR-R
       A(K3+1)=AWI-T
       A(K1)=AWR+R
       A(K1+1)=AWI+T
       T=A(K1)
       A(K1)=A(K)-T
       A(K)=A(K)+T
       T=A(K1+1)
       A(K1+1)=A(K+1)-T
       A(K+1)=A(K+1)+T
       R=-A(K3+1)
       T=A(K3)
       A(K3)=A(K2)-R
       A(K2)=A(K2)+R
       A(K3+1)=A(K2+1)-T
   220 A(K2+1)=A(K2+1)+T
C      END OF I AND K LOOPS
C
   230 JJ=JJDIF+JJ
C      END OF J-LOOP
C
   235 JLAST=4*JLAST+3
   240 CONTINUE
C      END OF  L  LOOP
C
   250 CONTINUE
C      END OF  ID  LOOP
C
C      WE NOW HAVE THE COMPLEX FOURIER SUMS BUT THEIR ADDRESSES ARE
C      BIT-REVERSED.  THE FOLLOWING ROUTINE PUTS THEM IN ORDER
       NTSQ=NT*NT
       M3MT=M3-MT
   350 IF(M3MT) 370,360,360
C
C      M3 GR. OR EQ. MT
   360 IGO3=1
       N3VNT=N3/NT
       MINN3=NT
       GO TO 380
C
C      M3 LESS THAN MT
   370 IGO3=2
       N3VNT=1
       NTVN3=NT/N3
       MINN3=N3
   380 JJD3 = NTSQ/N3
       M2MT=M2-MT
   450 IF (M2MT)470,460,460
C
C      M2 GR. OR EQ. MT
   460 IGO2=1
       N2VNT=N2/NT
```

```
      MINN2=NT
      GO TO 480
C
C     M2 LESS THAN MT
  470 IGO2 = 2
      N2VNT=1
      NTVN2=NT/N2
      MINN2=N2
  480 JJD2=NTSQ/N2
      M1MT=M1-MT
  550 IF(M1MT)570,560,560
C
C     M1 GR. OR EQ. MT
  560 IGO1=1
      N1VNT=N1/NT
      MINN1=NT
      GO TO 580
C
C     M1 LESS THAN MT
  570 IGO1=2
      N1VNT=1
      NTVN1=NT/N1
      MINN1=N1
  580 JJD1=NTSQ/N1
  600 JJ3=1
      J=1
      DO 880 JPP3=1,N3VNT
      IPP3=INV(JJ3)
      DO 870 JP3=1,MINN3
      GO TO (610,620),IGO3
  610 IP3=INV(JP3)*N3VNT
      GO TO 630
  620 IP3=INV(JP3)/NTVN3
  630 I3=(IPP3+IP3)*N2
  700 JJ2=1
      DO 870 JPP2=1,N2VNT
      IPP2=INV(JJ2)+I3
      DO 860 JP2=1,MINN2
      GO TO (710,720),IGO2
  710 IP2=INV(JP2)*N2VNT
      GO TO 730
  720 IP2=INV(JP2)/NTVN2
  730 I2=(IPP2+IP2)*N1
  800 JJ1=1
      DO 860 JPP1=1,N1VNT
      IPP1=INV(JJ1)+I2
      DO 850 JP1=1,MINN1
      GO TO (810,820),IGO1
  810 IP1=INV(JP1)*N1VNT
      GO TO 830
  820 IP1=INV(JP1)/NTVN1
  830 I=2*(IPP1+IP1)+1
      IF (J-I) 840,850,850
  840 T=A(I)
      A(I)=A(J)
      A(J)=T
      T=A(I+1)
      A(I+1)=A(J+1)
      A(J+1)=T
  850 J=J+2
  860 JJ1=JJ1+JJD1
C     END OF JPP1 AND JP2
C
  870 JJ2=JJ2+JJD2
C     END OF JPP2 AND JP3 LOOPS
C
```

```
  880 JJ3 = JJ3+JJD3
C     END OF JPP3 LOOP
C
  890 IF(IFSET)891,895,895
  891 DO 892 I = 1,NX
  892 A(2*I) = -A(2*I)
  895 RETURN
C
C     THE FOLLOWING PROGRAM COMPUTES THE SIN AND INV TABLES.
C
  900 MT=MAX0(M(1),M(2),M(3)) -2
      MT = MAX0(2,MT)
  904 IF (MT-18) 906,906,13
  906 IFERR=0
      NT=2**MT
      NTV2=NT/2
C
C     SET UP SIN TABLE
C     THETA=PIE/2**(L+1) FOR L=1
  910 THETA=.7853981633974483
C
C     JSTEP=2**(MT-L+1) FOR L=1
      JSTEP=NT
C
C     JDIF=2**(MT-L) FOR L=1
      JDIF=NTV2
      S(JDIF)=DSIN(THETA)
      DO 950 L=2,MT
      THETA=THETA/2.0D0
      JSTEP2=JSTEP
      JSTEP=JDIF
      JDIF=JSTEP/2
      S(JDIF)=DSIN(THETA)
      JC1=NT-JDIF
      S(JC1)=DCOS(THETA)
      JLAST=NT-JSTEP2
      IF(JLAST - JSTEP) 950,920,920
  920 DO 940 J=JSTEP,JLAST,JSTEP
      JC=NT-J
      JD=J+JDIF
  940 S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)
  950 CONTINUE
C
C     SET UP INV(J) TABLE
C
  960 MTLEXP=NTV2
C
C     MTLEXP=2**(MT-L). FOR L=1
      LM1EXP=1
C
C     LM1EXP=2**(L-1). FOR L=1
      INV(1)=0
      DO 980 L=1,MT
      INV(LM1EXP+1) = MTLEXP
      DO 970 J=2,LM1EXP
      JJ=J+LM1EXP
  970 INV(JJ)=INV(J)+MTLEXP
      MTLEXP=MTLEXP/2
  980 LM1EXP=LM1EXP*2
  982 IF(IFSET)12,895,12
      END
```