BNWL-1124

UC-32

2-
6-69

COMPUTER PROGRAMS
FOR HARMONIC ANALYSIS
AND SYNTHESIS OF TABULAR DATA

June 1969

AEC RESEARCH &
DEVELOPMENT REPORT

ROUTE TO	DATE	LOCATION	ROUTE DATE

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

operated by

BATTELLE MEMORIAL INSTITUTE

for the

UNITED STATES ATOMIC ENERGY COMMISSION UNDER CONTRACT AT(45-1)-1830

3 3679 00061 4448

BNWL-1124

UC-32, Mathematics
and Computers

COMPUTER PROGRAMS
FOR HARMONIC ANALYSIS AND SYNTHESIS
OF TABULAR DATA

By

C. A. Oster

Applied Mathematics Department
Systems and Electronics Division

June 1969

RECEIVED JUL 23 '69

BATTELLE MEMORIAL INSTITUTE
PACIFIC NORTHWEST LABORATORY
RICHLAND, WASHINGTON 99352

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U.S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

TABLE OF CONTENTS

List of Tables	v
Summary	1
Introduction	2
Harmonic Analysis and Synthesis	4
Fourier Series	4
Fourier Transform	7
Inversion of the Fourier Transform	10
Programs for Harmonic Analysis and Synthesis	12
Subroutine FORFIT	14
Subroutine F	14
Subroutine FINV	14
Example Using Harmonic Analysis and Synthesis Routines	17
Main Program	17
Additional Subprograms	21
Output	21
Program Parameter Statements	22
Acknowledgements	23
Bibliography	24
Appendix A	25
A General Case of the Fourier Transform	25

Appendix B	27
Program Listings of FORFIT, F and FINV	27
Appendix C	33
Program Listings of Main Program and Associated Subprograms .	33
Appendix D	47
Sample Input Cases	47
Appendix E	49
Sample Output	49

LIST OF TABLES

1. COMMON block descriptions	13
2. FINV argument list description	15
3. Description of input to Main Program	18
4. Definition of names listed under contents in Table 3	19
5. Minor subprograms and their function	21
6. Order of output	22
7. Parameter for program dimension statements	23

SUMMARY

This report describes three algorithms for the harmonic analysis and synthesis of tabular data. It includes a brief discussion of the underlying theory of Fourier series approximation, the Fourier transform of such a series and a method of inverting such a transform. These procedures have been coded as subprograms in FORTRAN V. A program using these subprograms is also described. The Appendices contain listings of all routines under discussion as well as sample input and the resulting output.

INTRODUCTION

Usually the analysis of natural phenomena leads to the examination of experimental data for recognizable characteristics. One of the most powerful mathematical tools available entails the subsection of the experimental data to a harmonic analysis with the aid of Fourier series [1,3]. The general form of the Fourier series is

$$f(t) = A_0 + \sum_{k=1}^n A_k \cos 2\pi v_k t + B_k \sin 2\pi v_k t.$$

The independent variable t , is treated here as having units of "time" and the parameters v_k have units of "vibrations per time unit". The function $f(t)$ may be operated upon by the Fourier transform to yield

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

where the independent variable ω is referred to as frequency.

There is a second transform which when applied to $F(\omega)$ produces $f(t)$. This transform is known as the inverse Fourier transform [5] and is given by

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

By taking $f(t)$ to be time limited, i.e. $f(t)$ is zero outside some finite interval, the function $F(\omega)$ can be readily determined. However the range of ω is now the entire real axis, so that the inverse Fourier transform is difficult to apply numerically. Herein are described a set of algorithms to generate the Fourier coefficients A_0, A_k, B_k $k = 1, 2, \dots, n$, from experimental data in either the temporal or frequency domains, and to calculate $F(\omega)$ for any value of ω resulting from the Fourier series $f(t)$.

Recently algorithms have been developed which result in considerable time savings in performing some of the same operations as the programs described herein. These algorithms are generally grouped under the name of the fast Fourier transform or FFT [1,3]. The FFT algorithm requires the number of data points to be highly composite in nature. In practice the number of data points is taken to be a power of 2. Output from the FFT is discrete in nature. Some kind of interpolation procedure must be used to obtain information at intermediate points. The algorithms presented here yield an appropriate interpolation function based upon the nature of the original data. Further the only requirement on the number of data points is that it be odd.

The algorithms described here are written as FORTRAN V subprograms. Their listings are given in Appendix B and their use is described below. Also included here is a program which uses these subprograms enabling one to produce Fourier series and Fourier transforms from tabular data. This program and other supporting routines are described below and listings appear in Appendix C.

Appendix D contains sample problems and Appendix E contains the computed results based upon the sample problems.

Many of the quantities involved in these programs are complex. No attempt has been made to utilize the complex arithmetic features of the FORTRAN V system [6]. All arithmetic here is real and the real and imaginary parts of complex numbers are kept distinctly labelled.

HARMONIC ANALYSIS AND SYNTHESIS

Below is a brief discussion concerning the fitting of Fourier series to tabular data, the Fourier transform of such a series, and the inversion of the Fourier transform.

FOURIER SERIES

Suppose we are given $2N+1$ data points $f_j \equiv \phi(T_j)$ obtained from some function $\phi(T)$, where the T_j satisfy the relation $T_{\min} = T_1 < T_2 < \dots < T_{2N+1} = T_{\max}$ and $T_j - T_{j-1}$ is constant for $j = 2, 3, \dots, 2N+1$. By transforming the $(T_{\min}, T_{\max})$ interval to the $(-1, 1)$ interval by $T = \underline{T}t - \bar{T}$ where

$$\underline{T} = \frac{1}{2} (T_{\max} - T_{\min}), \quad \bar{T} = \frac{1}{2} (T_{\max} + T_{\min}) \quad (1)$$

we obtain $-1 = t_1 < t_2 < \dots < t_{2N+1} = 1$. The coefficients of the Fourier series which passes through or interpolates over the data points (t_j, f_j) $j = 1, 2, \dots, 2N+1$ may be determined by the procedure given in Hildebrand [2]. This procedure begins by combining the data points into "even" and "odd" points according to the formulae

$$\left. \begin{aligned} G_j &= \frac{1}{2} [f_{N+j} + f_{N-j+2}] \\ H_j &= \frac{1}{2} [f_{N+j} - f_{N-j+2}] \end{aligned} \right\} j = 1, 2, \dots, N+1.$$

The Fourier coefficients may now be calculated by

$$\left. \begin{aligned}
 A_0 &= \frac{1}{N} \frac{1}{2} (G_1 + G_{N+1}) + \sum_{j=2}^n G_j \\
 A_k &= \frac{2}{N} \left\{ \frac{1}{2} [G_1 + (-1)^k G_{N+1}] \right. \\
 &\quad \left. + \sum_{j=2}^N G_j \cos \left[k \frac{\pi}{N} (j-1) \right] \right\} \quad k = 1, 2, \dots, N-1 \\
 A_N &= \frac{1}{N} \left\{ \frac{1}{2} [G_1 + (-1)^N G_{N+1}] + \sum_{j=2}^n (-1)^{j+1} G_j \right\} \\
 B_k &= \frac{2}{N} \sum_{j=2}^N H_j \sin \left[k \frac{\pi}{N} (j-1) \right] \quad k = 1, 2, \dots, N
 \end{aligned} \right\} \quad (2)$$

The sine and cosine functions are orthogonal under summation when evaluated at the specific points used here. This permits the writing of the Fourier series of $n \leq N$ periodic terms which "best" fits the original data in the sense of least square residue [2]. This series is given by

$$f(T) = A_0 + \sum_{k=1}^n A_k \cos [\theta_k (T - \bar{T})] + B_k \sin [\theta_k (T - \bar{T})] \quad (3)$$

where

$$\theta_k = \frac{k\pi}{\underline{T}} \quad (4)$$

with $\underline{T}$ and $\bar{T}$ as defined by equations (1) and the Fourier coefficients $A_0, A_1, \dots, A_n, B_1, B_2, \dots, B_n$ are precisely the same as those generated by equations (2) for the interpolation series.

The question of how large n must be to explain all of the information in the data (not to be confused with the question of how large N must be to describe the experiment from which the data was obtained) f_j can be answered by considering the root mean square (RMS) deviation between the true function and the calculated Fourier series over the points of observation T_j . Let the subscript k denote the use of the constant term A_0 and k pairs of periodic terms in the Fourier series. And let ϵ_k be the sum of the squared deviations of the approximating Fourier series evaluated at T_j from the f_j values. The orthogonality of the sine and cosine functions when summed over the points T_j permits the development of the following difference equation satisfied by ϵ_k .

$$\epsilon_0 = \sum_{j=1}^{2N+1} f_j^2 - 2NA_0^2$$

$$\epsilon_N = 0$$

$$\epsilon_k = \epsilon_{k-1} - N(A_k^2 + B_k^2) \quad k = 1, 2, \dots, N-1$$

The deviation RMS_k between the true and the k^{th} calculated functions is now defined in terms of ϵ_k as

$$\text{RMS}_k = \sqrt{\epsilon_k / (2N - 2k + 1)} \quad .$$

When k is large enough to explain the information in the f_j then RMS_ℓ is approximately constant for $\ell \geq k$. Thus by taking $n = k$ most of the information in the data will be included in (3).

FOURIER TRANSFORM

Associated with the original function $\phi(T)$ is another function known as the Fourier transform of $\phi(T)$. This transform is defined as the integral

$$\Phi(\omega) = \int_{-\infty}^{\infty} \phi(T) \exp(-i\omega T) dT.$$

This function resolves the given $\phi(T)$ into its harmonic components. The independent variable ω may take on any real value. To facilitate the discussion below we now consider the problem of approximating $\phi(\omega)$ by approximating the function $\phi(T)$ by the Fourier series described in equation (3). Let $F(\omega)$ denote this approximation of $\phi(\omega)$, that is

$$F(\omega) = \int_{-\infty}^{\infty} f(T) \exp(-i\omega T) dT.$$

Now if $\phi(T)$ is a pulse we have $\phi(T) \equiv 0$ when T is outside the interval $(T_{\min}, T_{\max})$. In this case the limits of integration on the Fourier transform become finite and the approximation of $\phi(\omega)$ by $F(\omega)$ becomes

$$F(\omega) = \int_{T_{\min}}^{T_{\max}} f(T) \exp(-i\omega T) dt$$

$$= \sum_{k=0}^n F_k(\omega)$$

Where the $F_k(\omega)$ are as follows

$$F_0(\omega) = \begin{cases} 2A_0 T & \omega = 0 \\ \frac{2A_0}{\omega} \sin T\omega [\cos T\omega - i \sin T\omega] & \omega \neq 0 \end{cases} \quad (5)$$

and for $k > 0$

$$F_k(\pm\theta_k) = \underline{T} \left\{ \left[A_k \cos(\theta_k \bar{T}) - B_k \sin(\theta_k \bar{T}) \right] \right. \\ \left. + i \left[A_k \sin(\theta_k \bar{T}) + B_k \cos(\theta_k \bar{T}) \right] \right\} \quad (6)$$

$$F_k(\omega) = (-1)^k 2 \left[\frac{\sin \omega \bar{T}}{\theta_k^2 - \omega^2} \right] \left\{ \left[A_k \omega \cos \omega \bar{T} - B_k \theta_k \sin \omega \bar{T} \right] \right. \\ \left. - i \left[A_k \omega \sin \omega \bar{T} + B_k \theta_k \cos \omega \bar{T} \right] \right\} \quad (7)$$

if $\omega \neq \pm \theta_k$.

The calculation algorithm may now be described. Begin by calculating the quantities $g_0, g_1, \dots, g_n, h_1, h_2, \dots, h_n$ as follows

$$g_0 = \begin{cases} 2\underline{T} & \omega = 0 \\ \frac{2}{\omega} \sin \underline{T}\omega & \omega \neq 0 \end{cases} \quad (8)$$

And for $k > 0$

$$g_k = \begin{cases} \underline{T} & \omega = \pm \theta_k \\ (-1)^{k+1} 2 \frac{\omega}{\theta_k^2 - \omega^2} \sin \underline{T}\omega & \omega \neq \pm \theta_k \end{cases} \quad (9)$$

$$h_k = \begin{cases} \bar{+} \underline{T} & \omega = \underline{+} \theta_k \\ (-1)^k 2 \frac{\theta_k}{\theta_k^2 - \omega^2} \sin \underline{T}\omega & \omega \neq \underline{+} \theta_k \end{cases} \quad (10)$$

Then

$$S_A = \sum_{k=0}^n A_k g_k, \quad S_B = \sum_{k=1}^n B_k h_k.$$

Finally

$$F(\omega) = \left[S_A \cos \bar{T}\omega + S_B \sin \bar{T}\omega \right] \\ + i \left[-S_A \sin \bar{T}\omega + S_B \cos \bar{T}\omega \right]$$

Notice that if $\omega = \underline{+} \theta_k$, $k > 0$ then $\underline{T}\omega = \underline{+} k\pi$ and the above expressions simplify to

$$S_A = A_{k\underline{-}} \underline{T}, \quad S_B = \bar{+} B_{k\underline{-}} \underline{T}.$$

Similarly if $\omega = 0$

$$S_A = 2A_{0\underline{-}} \underline{T}, \quad S_B = 0.$$

These relationships can be used to make the computation more efficient since equations (8, 9, and 10) need not be used to obtain S_A and S_B used in the calculation of $F(\underline{+} \theta_k)$.

INVERSION OF THE FOURIER TRANSFORM

The Fourier transform $\phi(\omega)$ resolves the given function $\phi(T)$ into its harmonic components. Here we consider the problem of inverting this process. We wish to start with the harmonic components and synthesize them to the original function. This process involves an infinite integral over the range of ω and is not practical here. We shall replace this process with a least square procedure designed to produce a Fourier series approximation to $\phi(T)$.

Suppose we are given N distinct real parameters $\theta_1, \theta_2, \dots, \theta_N$, and $M > N$ pairs $(\omega_j, F(\omega_j))$ where the ω_j are real, and the $F(\omega_j)$ complex numbers. We write $F_j \equiv F(\omega_j)$ and wish to determine the quantities $A_0, A_1, \dots, A_N, B_1, B_2, \dots, B_N$ which are the Fourier coefficients of a finite series whose Fourier Transform is the distribution from which the data $\theta_1, \theta_2, \dots, \theta_N, F_1, F_2, \dots, F_M$ when taken.

Using least square methods we begin with

$$\epsilon^2 = \sum_{j=1}^M \left[F_j - \sum_{k=0}^N F_k(\omega_j) \right] \left[\overline{F_j} - \sum_{k=0}^N \overline{F_k(\omega_j)} \right] .$$

Where $F_k(\omega)$ is defined in equations (4,5,6) and the bar indicates conjugation.

The task of computing the Fourier coefficients is greatly simplified by first transforming the data F_j through multiplication by $e^{i\omega_j \overline{T}}$. From equations (12, 13, and 14) we see that this replaces the original problem by a similar one but now it is centered about the time origin. We shall hence forth assume the aforementioned multiplication has taken place. As indicated above the resulting data may be treated as though $\overline{T} = 0$. By taking $\overline{T} = 0$ the expression for $F_k(\omega)$ equations (12, 13, and 14) reduces to a real part which is independent of B_k and an imaginary part which is independent of A_k . Hence the A_k can be determined by real arithmetic

operations quite independent of B_k . Similarly B_k can be obtained without regard to A_k .

From a theoretical point of view there need be no other requirements on the parameters θ_k . The procedure given below can be readily extended to cover the arbitrary case. By requiring that the θ_k satisfy equation (4), the procedure simplifies considerably. This is the case considered here.

By writing $F_{kj} \equiv F_k(\omega_j)$, $g_{kj} \equiv g_k(\omega_j)$ and $h_{kj} \equiv h_k(\omega_j)$ where g and h are defined by equations (8,9, and 10) we obtain

$$F_{kj} = g_{kj} A_k + i h_{kj} B_k$$

We further define the real and imaginary parts of F_{kj} as

$$F_{rj} \equiv \text{Re}(F(\omega_j)), F_{ij} \equiv \text{Im}(F(\omega_j)).$$

By differentiating ϵ^2 with respect to $A_0, A_1, \dots, A_N, B_1, \dots, B_N$ and setting the derivatives equal to zero we obtain the normal equations

$$\sum_{\ell=0}^N A_{\ell} \left(\sum_{j=1}^M g_{kj} g_{\ell j} \right) = \sum_{j=1}^M F_{rj} g_{kj} \quad k = 0, 1, 2, \dots, N$$

$$\text{and} \quad \sum_{\ell=1}^N B_{\ell} \left(\sum_{j=1}^M h_{kj} h_{\ell j} \right) = \sum_{j=1}^M F_{ij} h_{kj} \quad k = 1, 2, \dots, N.$$

These equations are linear in A_{ℓ} , and B_{ℓ} . Hence the Fourier coefficients can be readily obtained, yielding

$$f(t) = A_0 + \sum_{k=1}^N \left\{ A_k \cos \theta_k t + B_k \sin \theta_k t \right\}$$

for $|t| \leq T$, or

$$f(T) = A_0 + \sum_{k=1}^N \left\{ A_k \cos \theta_k (T - \bar{T}) + B_k \sin \theta_k (T - \bar{T}) \right\}$$

for $T \leq T \leq T$

PROGRAMS FOR HARMONIC ANALYSIS AND SYNTHESIS

The three algorithms for computing Fourier coefficients from temporal domain data, evaluating the Fourier transform of a finite Fourier series, and the computation of Fourier coefficients from frequency domain data are described below. These algorithms are programmed as subprograms having names FORFIT, F and FINV respectively. FORFIT and F are more closely related than FINV is to either FORFIT or F. For this reason the COMMON block labeled BLOCK1 is used to transmit quantities between them. The contents of this COMMON block are given below in Table 1. The dimensions are all defined by the use of the PARAMETER statement feature of FORTRAN V [6]. The single parameter IPTS is used to govern the size of the arrays in BLOCK1. IPTS has the value 250. The dimensions of the arrays can be changed by simply changing this parameter value. See also Table 6. All arithmetic is real with the real and imaginary parts of complex numbers clearly identified.

Table 1. COMMON Block Descriptions

<u>Block Name</u>	<u>Program Names*</u>	<u>Contents</u>
BLOCK1	A(IPTS+1)	A_k $k = 0, 1, \dots, N$
	B(IPTS)	B_k $k = 1, 2, \dots, N$
	AK(IPTS) †	θ_k $k = 1, 2, \dots, N$
	FTI(2*IPTS+1)	f_j $j = 1, 2, \dots, 2N+1$
	EN	N
	N	N
	N2P1	2N+1
	TMIN	$T_{\min}$
	TMAX	$T_{\max}$
	TSUBAR	$\underline{T}$
	TBAR	$\overline{T}$
PY	PI	π
	TW θ PI	2π

* Dimensions of arrays are given in terms of the parameter IPTS and are shown in parentheses.

† This array is labeled AA in FORFIT.

SUBROUTINE FORFIT

This subprogram computes the Fourier coefficients from temporal domain data. Input to FORFIT are the data f_j , $j = 1, 2, \dots, 2N+1$; the number of data points $2N+1$, $T_{\min}$ and $T_{\max}$. The output are the Fourier coefficients $A_0, A_1, \dots, A_N, B_1, B_2, \dots, B_N$; the number N in integer and floating point form, $\underline{T}$ and $\overline{T}$. This program has no call list since all quantities input and output are in COMMON block BLOCK1.

SUBROUTINE F

This subprogram computes the Fourier transform of the Fourier series whose parameters are stored in COMMON block BLOCK1. See Table 1. The form of the call statement is

$$\text{CALL F}(\omega, n, \text{Re}(F(\omega)), \text{Im}(F(\omega))).$$

Input through the argument call list are: The frequency ω , at which the transform is to be evaluated, and the number of terms n to be used in the Fourier series. Output is through the argument call list as two numbers, the real part of $F(\omega)$ and the imaginary part of $F(\omega)$.

SUBROUTINE FINV

This subprogram computes the Fourier coefficients from frequency domain data. FINV is used by including a call statement

$$\text{CALL FINV} (M, \text{BFR}, \text{BFI}, \text{MM}, \text{TMIN}, \text{TMAX}, A, B, \text{TH}, \text{NN}, \text{TSUBAR}, \text{TBAR}, \text{EN}, N)$$

Table 2. FINV Argument List Description

	LIST NAMES	DESCRIPTION
INPUT		
	θM	$\omega_j \quad j = 1, 2, \dots, MM$
	BFR	$F_{rj} \quad j = 1, 2, \dots, MM$
	BFI	$F_{ij} \quad j = 1, 2, \dots, MM$
	MM	Length of the arrays θM , BFR, BFI
	TMIN	$T_{\min}$
	TMAX	$T_{\max}$
	NN	Numbers of coefficients to be computed
OUTPUT		
	A	$A_k \quad k = 0, 1, \dots, N$
	B	$B_k \quad k = 1, 2, \dots, N$
	TH	$k \quad k = 1, 2, \dots, N$
	TSUBAR	$\underline{T}$
	TBAR	$\overline{T}$
	EN	Floating point N
	N	N number of coefficients actually computed.

It should be pointed out that NN need not be the same as N. N is taken as the minimum of NN and ISPC+1, where ISPC is the parameter defining the dimensions of the storage arrays the program FINV. See Table 7. If MM exceeds 2 IPTS then only 2 IPTS+1 points will be selected from the input data. The points omitted will be approximately evenly distributed ("subscript-wise") over the range from 1 to MM. Thus one is protected from exceeding array sizes in FINV.

FINV requires one additional routine for solving a system of simultaneous linear equations. The name of this subroutine is CROUT. It has been included in Appendix C.

EXAMPLE USING HARMONIC ANALYSIS AND SYNTHESIS ROUTINES

Below are described a program which uses the subprograms FORFIT, F and FINV described earlier. This program also uses two other subprograms which are not included in a standard FORTRAN V library. They are briefly noted below.

MAIN PROGRAM

Included in Appendix C is a listing of a program which indicates how to use the three subroutines FORFIT, F and FINV. This program is useful in its own right. It will compute Fourier coefficients from either temporal or frequency domain data. If the data is from the temporal domain the Fourier transform may also be obtained. The resulting Fourier series is plotted as a continuous function of time. If the Fourier transform is requested it too is printer plotted as a continuous function of frequency. All plotted data is also listed so that it can be manually plotted if higher resolution is required.

Input to this program is described in Table 3 below. The contents are further described in Table 4. It must be emphasized that cards of type 1- and 2- cannot appear within the same case.

Cases may be stacked. The two different kinds of data may occur within a given run by stacking two different cases. Any number of cases may be stacked.

Table 3. Description of Input to Main Program

<u>Card Type</u>	<u>Contents</u>	<u>Format</u>
0-1	72 Alphanumeric identification	12A6
0-2	T1 T2, MM, M, KEY	2E10.7, 3I5
IF KEY = 0		
1-1	T _{min} , T _{max} , NN, SF1, SF2	2E10.7, I5, 10 x 2E10.7
1-2	FTI ₁ , FTI ₂ , ..., FTI ₇ , FTI ₈ , ... ⋮ FTI _{NN-1} , FTI _N	} 7E10.7
1-3	FC1, FC2, NQ, NT	
IF KEY ≠ 0		
2-1	NQ, SFACTR	20 X I5, 10 X E10.7
2-2	ΘM ₁ , BRF ₁ , BRT ₁ , M ₂ , BFR ₂ , BRI ₂ ΘM ₃ , ... ⋮ ΘM _{NQ} , BFR _{NQ} , BFI _{NQ}	} 6E10.7
2-3	T _{min} , T _{max} , NN	

Table 4. Definition of Names Listed Under Contents in Table 3.

<u>Card Type</u>	<u>Symbol</u>	<u>Description</u>
0-2	T1, T2	If $KEY \neq 0$ or $MM \leq 0$ these quantities are ignored. If $KEY = 0$ and $MM > 0$ the Fourier series of M pairs of periodic terms plus A_0 obtained from Temporal domain data is evaluated and plotted for MM equally spaced values of T in $T1 \leq T \leq T2$.
	MM	See description under T1, T2 above
	M	See description under T1, T2 above for $M > 0$. If $M \leq 0$ it will be set to the index where the minimum RMS value is obtained. The treatment of M is then as described above under T1, T2.
	KEY	If KEY is blank or zero the Fourier coefficients are obtained from temporal data. If $KEY \neq 0$ the coefficients are obtained from frequency domain data.
1-1	TMIN, TMAX	The range of the independent variable for temporal domain data.
	NN	Total number of data points. Must be odd.
	SF1, SF2	These two parameters are used to determine the scale factor which will be used to multiply each value of FTI before processing begins. If SF1 and SF2 are both zero or absent the scale factor is 1. If SF2 is absent the scale factor is SF1. If SF2 is present then the scale factor

Table 4. (Continued)

<u>Card Type</u>	<u>Symbol</u>	<u>Description</u>
		is calculated from the equation $10^{(SF1-SF2)/20}$.
1-2	FTI	These are the NN data points. FTI_1 is the amplitude at $T = T_{min}$. Similarly FTI_{NN} is the amplitude as $T = T_{max}$.
1-3	FC1, FC2	If NQ and NT > 0 the Fourier transform of the obtained series of NT terms will be evaluated for NQ uniformly spaced frequencies between FC1 and FC2. The end points of this range are included. If NQ > 0 and NT = 0 or is absent NT is set to M.
	NQ, NT	See description under FC1, FC2 above.
2-1	NQ	This the number of frequency domain data points to follow immediately.
	SFACTR	Scale factor - both the BFR and BFI are multiplied by SFACTR before processing begins.
2-2	θM, BFR, BFI	This triple constituted one frequency domain data point. θM is the independent variable and $BFR + i BFI$ is the dependent variable at θM.
2-3	TMIN, TMAX	See explanation under card type 1-1 above.
	NN	

ADDITIONAL SUBPROGRAMS

There are several minor subprograms related to the main program. These are included in the table below and their function is briefly noted.

Table 5. Minor Subprograms and Their Function

<u>Name</u>	<u>Calling Routine</u>	<u>Function</u>
COLUMN	MAIN	Print out input data in columnar form.
CROUT	FINV	Solve the system of linear equations yielding the Fourier coefficients.
PLOT	MAIN	Produces plots of the various tabulated functions of the program, e.g., approximating Fourier series, Fourier transform of this series.

The full potential of these routines has not been described here, but rather only the use made in this program package.

OUTPUT

The output of this program package is labelled in such a manner that there should be little misunderstanding by the user.

The order of the output is described in Table 6 which also gives the units when they differ from the input units. The amount of output yielded depends upon the input. The order of printing follows Table 6 with the exceptions that some groups may not be requested by input data.

Table 6. Order of Output

1. Input data as read from input card type 1-2 or 2-2, before any scaling occurs. The scale factor is printed below this group of output.
2. Fourier coefficients $A_0, A_1, \dots, A_N, B_1, B_2, \dots, B_N$; the angular frequency parameters $\theta_1/2\pi, \dots, \theta_N/2\pi$ in units of cycles per second. If the input data is temporal i.e. card type 1-2 is used, then $RMS_0, RMS_1, \dots, RMS_N$ is also included in this group of output data. If the input data is spectral the RMS values are omitted. This group of output is ended with the two quantities $\underline{T}$ and $\overline{T}$.
3. The evaluation of the temporal Fourier series at the points requested on input card type 0-2.
4. The data printed under 3 above is plotted on a printer plot.
5. If the input data is temporal and $NQ > 0$ (see card type 1-3 in Table 4) the Fourier transform of the temporal Fourier series is evaluated. This group of data includes ω (cycles per second), $Re(F(\omega)), Im(F(\omega))$ and $|F(\omega)|$ for ω over the range specified by FC1 and FC2 on card type 1-3.
6. The data listed in 5 above is printer plotted on three graphs, one for each $Re(F(\omega)), Im(F(\omega)), |F(\omega)|$ as a function of ω . Again the units of ω is cycles per second.

PROGRAM PARAMETER STATEMENTS

The dimensions of arrays which can be modified without altering the program itself i.e. data arrays, are defined by use of the PARAMETER statement of FORTRAN V, see [4]. The following table gives the values currently used by the various programs.

Table 7. Parameters for Program Dimension Statements

<u>PARAMETER NAME</u>	<u>ROUTINES INVOLVED</u>	<u>VALUE</u>
IPTS	MAIN FORFIT F FINV	250
ISPC	MAIN PLOT	101
JSPC	MAIN FINV CROUT	25

ACKNOWLEDGEMENT

Many individuals have contributed to this work. At least two members of the Mathematical and Numerical Analysis Section must have special recognition. Many discussions with Mr. R. L. Richardson provided much of the insight required for the understanding of the theory described herein. Dr. W. R. McSpadden provided similar support in the understanding as well as many helpful comments on the written material. Their help is greatly appreciated.

BIBLIOGRAPHY

1. Brigham, E. O., and R. E. Morrow "The Fast Fourier Transform" IEEE Spectrum December 1967 pp. 63-70.
2. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, New York, 1956.
3. IEEE Transactions on Audio and Electroacoustics, special issue on Fast Fourier Transform and its application to digital filtering and spectral analysis. volume AU-15 No. 2 June 1967.
4. Jackson, Dunham, Fourier Series and Orthogonal Polynomials, Carus Mathematical Monographs, Math. Assoc. of America, Oberlin, Ohio, 1941.
5. Lanczos, Cornelius, Applied Analysis Prentice Hall, Inc., Englewood Cliffs, New Jersey. 1956.
6. Programmers Reference Manual, UNIVAC 1108, FORTRAN V FORM UP.4060.

APPENDIX A

A GENERAL CASE OF THE FOURIER TRANSFORM

APPENDIX A

A GENERAL CASE OF THE FOURIER TRANSFORM

Let $f(T)$ be defined as in equation (3) except permit θ_k to be arbitrary non-zero real numbers. The Fourier transform of $f(T)$ is defined to be

$$F(\omega) = \int_{-\infty}^{\infty} f(T) \exp(-i\omega T) dT. \quad (11)$$

If we are interested in studying pulses $f(T) = 0$ for T outside some finite interval, designated by $(T_{\min}, T_{\max})$ then $F(\omega)$ in equation (11) becomes the finite Fourier transform

$$F(\omega) = \int_{T_{\min}}^{T_{\max}} f(T) \exp(-i\omega T) dt.$$

By interchanging the order of integration and summation, and transforming each term of $f(T)$ we obtain

$$F(\omega) = \sum_{k=0}^n F_k(\omega)$$

Where the terms $F_k(\omega)$ are defined below.

$$F_o(\omega) = \begin{cases} 2A_o T & \omega = 0 \\ \frac{2A_o}{\omega} \sin \omega T e^{-i\omega T} & \omega \neq 0 \end{cases} \quad (12)$$

For $k > 0$ and $\theta = \pm \theta_k$

$$F_k(\pm \theta_k) = [A_k \mp iB_k]T e^{\mp \theta_k T} \quad (13)$$

If $\theta \neq \pm \theta_k$ and $k > 1$

$$F_k(\omega) = \left\{ A_k \left[\frac{\sin(\theta_k - \omega)T}{\theta_k - \omega} + \frac{\sin(\theta_k + \omega)T}{\theta_k + \omega} \right] - iB_k \left[\frac{\sin(\theta_k - \omega)T}{\theta_k - \omega} - \frac{\sin(\theta_k + \omega)T}{\theta_k + \omega} \right] \right\} e^{-i\omega T} \quad (14)$$

A close examination of $F(\omega)$ reveals that it is linear in A_k and B_k . Notice that $\text{Re}(F(\omega))$ is independent of the B_k 's and $\text{Im}(F(\omega))$ is independent of the A_k 's if and only if $T = 0$. This fact is used in the inversion process since the two sequences A_k and B_k of Fourier coefficients can be determined separately.

The above expression can be further simplified if $\theta_k = k\pi/T$ then for $k > 0$ and $\omega \neq \pm \theta_k$ we obtain

$$F_k(\omega) = (-1)^{k+1} 2 \left[\frac{\sin \omega T}{\theta_k^2 - \omega^2} \right] \left[A_k \omega - iB_k \theta_k \right] e^{-i\omega T}$$

Expanding $e^{-i\omega T}$ we obtain equation (6). The form of the equations for $F_0(\omega)$ and $F_k(\pm \theta_k)$ remain unchanged by this requirement on θ_k .

APPENDIX B

PROGRAM LISTINGS OF FORFIT, F AND FINV

```

SUBROUTINE FORFIT                                FORF0000
C----- THIS ROUTINE CALCULATES THE STANDARD FOURIER COEFFICIENTS BASED  FORF0010
C      ON THE DATA IN FTI. FTI IS ASSUMED TO CONTAIN N2P1 (2*N+1)  FORF0020
C      VALUES OF A DEPENDENT VARIABLE OBTAINED AT N2P1 EQUALLY SPACED  FORF0030
C      POINTS BETWEEN TMIN AND TMAX INCLUSIVE. THE COSINE COEF-  FORF0040
C      FICIENTS ARE STORED IN A AND THE SINE COEFFICIENTS ARE STORED  FORF0050
C      IN B. TBAR IS THE MID POINT OF THE INTERVAL (TMIN,TMAX) AND  FORF0060
C      TSUBAR IS THE HALF LENGTH OF THE INTERVAL (TMIN,TMAX).  FORF0070
C      THE ANGULAR FREQUENCIES ENCOUNTERED ARE STORED IN AA. OUTPUT  FORF0080
C      IS N+1 A-S, N B-S, N AA-S, TSUBAR, AND TBAR. THUS ONE MAY  FORF0090
C      USE F(T)= A(1)+A(2)*COS(AA(1)*(T-TBAR))+  FORF0100
C      B(1)*SIN(AA(2)*(T-TBAR))+  FORF0110
C      A(3)*COS(AA(2)*(T-TBAR))+  FORF0120
C      B(2)*SIN(AA(2)*(T-TBAR))+ . . . +  FORF0130
C      A(N+1)*COS(AA(N)*(T-TBAR))+  FORF0140
C      B(N)*SIN(AA(N)*(T-TBAR))  FORF0150
C      FORF0160
C      AA(1)=I*PI/TSUBAR  FORF0170
C      N IS SUCH THAT 2*N+1=N2P1  FORF0180
C      FORF0190
C      PARAMETER IPTS=250,JPTS=IPTS+1,KPTS=2*IPTS,MPTS=KPTS+1  FORF0200
C      COMMON /BLOCK1/A(JPTS),B(IPTS),AA(IPTS),FTI(MPTS),EN,N,N2P1,  FORF0210
1  TMIN,TMAX,TSUBAR,TBAR  FORF0220
C      COMMON /PY/P1,TWOPI  FORF0230
C      DIMENSION F(JPTS),G(JPTS)  FORF0240
C      N=(N2P1-1)/2  FORF0250
C      NP1=N+1  FORF0260
C      NP2=N+2  FORF0270
C      EN=N  FORF0280
C      TSUBAR=(TMAX-TMIN)/2.  FORF0290
C      TBAR=(TMAX+TMIN)/2.  FORF0300
C      TSLOP=EN/TSUBAR  FORF0310
C      AA(N)=PI*TSLOP  FORF0320
C      STEP=PI/EN  FORF0330
C      ENO2=EN/2.  FORF0340
C      DO 1 J=1,NP1  FORF0350
C      K=N+J  FORF0360
C      L=NP2-J  FORF0370
C      F(J)=(FTI(K)+FTI(L))/2.  FORF0380
1  G(J)=(FTI(K)-FTI(L))/2.  FORF0390
C      FNP1=F(NP1)  FORF0400
C      SGN=1.  FORF0410
C      AK=(F(1)+FNP1)/2.  FORF0420
C      BK=0.  FORF0430
C      DO 2 J=2,N  FORF0440
C      SGN=-SGN  FORF0450
C      AK=AK+F(J)  FORF0460
2  BK=BK+SGN*F(J)  FORF0470

```

```
A(1)=AK/EN
A(NP1)=(BK+(F(1)-SGN*FNP1)/2.)/EN
ANG=0.
SGN=FNP1
DO 4 K=2,N
ANG=ANG+STEP
AA(K-1)=ANG*TSLOP
SGN=-SGN
AK=(F(1)+SGN)/2.
ANGLE=0.
DO 3 J=2,N
ANGLE=ANGLE+ANG
3 AK=AK+F(J)*COS(ANGLE)
4 A(K)=AK/ENO2
ANG=0.
DO 6 K=2,NP1
ANG=ANG+STEP
ANGLE=0.
BK=0.
DO 5 J=2,N
ANGLE=ANGLE+ANG
5 BK=BK+G(J)*SIN(ANGLE)
6 B(K-1)=BK/ENO2
RETURN
END
```

```
FORF0480
FORF0490
FORF0500
FORF0510
FORF0520
FORF0530
FORF0540
FORF0550
FORF0560
FORF0570
FORF0580
FORF0590
FORF0600
FORF0610
FORF0620
FORF0630
FORF0640
FORF0650
FORF0660
FORF0670
FORF0680
FORF0690
FORF0700
FORF0710
FORF072-
```

	SUBROUTINE F(OME,M,FREAL,FIMAG)	F	0000
C----	THIS ROUTINE CALCULATES THE REAL (FREAL) AND IMAGINARY (FIMAG)	F	0010
C	PARTS OF THE FOURIER SERIES WHICH HAS BEEN TRANSFORMED BY	F	0020
C	THE COMPLEX FOURIER TRANSFORM. OME IS THE INDEPENDENT	F	0030
C	VARIABLE, M IS THE NUMBER OF TERMS DESIRED IN THE SERIES.	F	0040
C	SEE FORFIT SUBROUTINE FOR REMARKS ON THE FOURIER SERIES.	F	0050
C		F	0060
	PARAMETER IPTS=250,JPTS=IPTS+1,KPTS=2*IPTS,MPTS=KPTS+1	F	0070
	COMMON /PY/PI,TWOPI	F	0080
	COMMON /BLOCK1/A(JPTS),B(IPTS),AK(IPTS),FTI(MPTS),EN,N,N2P1,	F	0090
	ITMIN,TMAX,TSUBAR,TBAR	F	0100
	OMEGA=OME	F	0110
	MM=MINO(M,N)+1	F	0120
	IF(OMEGA) 2,1,2	F	0130
1	FKR= 2.*TSUBAR*A(1)	F	0140
	FKI=0.	F	0150
	GO TO 9	F	0160
2	SINO=-SIN(TSUBAR*OMEGA)	F	0170
	SA=-2.*A(1)/OMEGA*SINO	F	0180
	SB=0.	F	0190
	DO 5 K=2,MM	F	0200
	SINO=-SINO	F	0210
	IF(ABS((ABS(OMEGA)-AK(K-1))*TSUBAR)-7.745E-4) 3,3,4	F	0220
3	SA=SA+TSUBAR*A(K)	F	0230
	SB=SB-TSUBAR*B(K-1)*SIGN(1.,OMEGA)	F	0240
	GO TO 5	F	0250
4	FKP=1./(AK(K-1)+OMEGA)	F	0260
	FKM=1./(AK(K-1)-OMEGA)	F	0270
	SA =SA +A(K)*(FKM-FKP)*SINO	F	0280
	SB =SB -B(K-1)*(FKM+FKP)*SINO	F	0290
5	CONTINUE	F	0300
6	BAW=-TBAR*OMEGA	F	0310
	IF(BAW) 8,7,8	F	0320
7	FKR=SA	F	0330
	FKI=SB	F	0340
	GO TO 9	F	0350
8	SINBAW=SIN(BAW)	F	0360
	COSBAW=COS(BAW)	F	0370
	FKR=SA*COSBAW-SB*SINBAW	F	0380
	FKI=SA*SINBAW+SB*COSBAW	F	0390
9	FREAL=FKR	F	0400
	FIMAG=FKI	F	0410
	RETURN	F	0420
	END	F	043-

```

SUBROUTINE FINV(OM,BFR,BFI,MM,TMIN,TMAX,A,B,TH,NN,TSUBAR,TBAR,
1EN,N)
C----- THIS SUBROUTINE INVERTS THE FOURIER TRANSFORM (SEE SUBROUTINE F)
C      AND GIVES BACK THE COEFFICIENTS A AND B AND THE ANGULAR
C      FREQUENCIES TH OF THE FOURIER SERIES (SEE SUBROUTINE FORFIT).
C      IT IS ASSUMED THAT THERE ARE MM TRIPLES (OM,BFR,BFI), WHERE
C      OM IS THE INDEPENDENT VARIABLE AND BFR+SQRT(-1)*BFI IS THE
C      DEPENDENT VARIABLE. THERE ARE NN PAIRS OF PERIODIC TERMS
C      PLUS A CONSTANT TERM IN THE FOURIER SERIES (SEE SUBROUTINE
C      FORFIT). RANGE OF THE FOURIER SERIES INDEPENDENT VARIABLE IS
C      (TMIN,TMAX). A LEAST SQUARES METHOD IS USED.
PARAMETER IPTS=250,JPTS=IPTS+1,KPTS=2*IPTS,MPTS=KPTS+1
PARAMETER ISPC=100,JSPC=25,KSPC=JSPC+1
PARAMETER LSPC=KSPC+1,MSPC=2*IPTS
DIMENSION AA(KSPC,LSPC),SF(KSPC,ISPC),SG(JSPC,ISPC),SFR(ISPC),
1SFI(ISPC)
C----- OBTAIN LICENSE TO SUBSCRIPT THE FOLLOWING VARIABLES.
DIMENSION OM(1),BFR(1),BFI(1),TH(1),A(1),B(1)
COMMON /PY/PI,TWOPI
MJ=MM
INC=MJ/MSPC
C----- IF MM EXCEEDS 2*IPTS THEN SELECT EVERY INC-TH TRIPLE.
IF(MJ-MSPC*INC) 101,102,101
101 INC=INC+1
102 CONTINUE
M=MIN0(MJ,MSPC)
N=MIN0(NN,JSPC)
MN1=N+1
NN2=N+2
EN=N
TSUBAR=(TMAX-TMIN)/2.
TBAR=(TMAX+TMIN)/2.
C----- SET UP BASIS FREQUENCIES
STEP=PI/TSUBAR
FI=0.
DO 100 I=1,N
FI=FI+1.
100 TH(I)=FI*STEP
IF(TBAR) 300,200,300
200 ASSIGN 500 TO ITBAR
GO TO 400
300 ASSIGN 600 TO ITBAR
400 CONTINUE
C----- CONSTRUCT NORMAL EQUATIONS MATRICES
C----- FIRST WE CONSTRUCT ESSENTIAL VECTORS.
J=0
DO 6 JJ=1,MJ,INC
J=J+1

```

FINV0000
FINV0010
FINV0020
FINV0030
FINV0040
FINV0050
FINV0060
FINV0070
FINV0080
FINV0090
FINV0100
FINV0110
FINV0120
FINV0130
FINV0140
FINV0150
FINV0160
FINV0170
FINV0180
FINV0190
FINV0200
FINV0210
FINV0220
FINV0230
FINV0240
FINV0250
FINV0260
FINV0270
FINV0280
FINV0290
FINV0300
FINV0310
FINV0320
FINV0330
FINV0340
FINV0350
FINV0360
FINV0370
FINV0380
FINV0390
FINV0400
FINV0410
FINV0420
FINV0430
FINV0440
FINV0450
FINV0460
FINV0470

OMJ=OM(JJ)	FINV0480
GO TO ITBAR,(500,600)	FINV0490
500 SFR(J)=BFR(JJ)	FINV0500
SFI(J)=BFI(JJ)	FINV0510
GO TO 700	FINV0520
600 SH=OMJ*TBAR	FINV0530
SINE=SIN(SH)	FINV0540
COSINE=COS(SH)	FINV0550
SFR(J)=BFR(JJ)*COSINE-BFI(JJ)*SINE	FINV0560
SFI(J)=BFR(JJ)*SINE +BFI(JJ)*COSINE	FINV0570
700 CONTINUE	FINV0580
SIM=-SIN(TSUBAR*OMJ)*2.	FINV0590
OMSQ=OMJ**2	FINV0600
IF(OMJ) 2,1,2	FINV0610
C---- OMJ IS THE BASIS FREQUENCY 0.	FINV0620
1 SF(1,J)=TSUBAR*2.	FINV0630
GO TO 3	FINV0640
C---- NORMAL CASE	FINV0650
2 SF(1,J)=-SIM/OMJ	FINV0660
C---- FILL IN REMAINDER OF VECTORS	FINV0670
3 DO 6 K=1,N	FINV0680
THK=TH(K)	FINV0690
SIM=-SIM	FINV0700
IF(ABS((ABS(OMJ)-THK)*TSUBAR)-7.745E-4) 4,4,5	FINV0710
C---- OMJ IS THE BASIS FREQUENCY TH(K)	FINV0720
4 SF(K+1,J)=TSUBAR	FINV0730
SG(K,J)=-SIGN(SF(K+1,J),OMJ)	FINV0740
GO TO 6	FINV0750
C---- NORMAL CASE	FINV0760
5 P=SIM/(THK**2-OMSQ)	FINV0770
SF(K+1,J)= OMJ*P	FINV0780
SG(K,J)=-THK*P	FINV0790
6 CONTINUE	FINV0800
C---- FORM THE ACTUAL ELEMENTS OF THE NORMAL MATRIX FOR A-S	FINV0810
DO 9 K=1,NN1	FINV0820
DO 8 L=1,NN1	FINV0830
AAKL=0.	FINV0840
DO 7 J=1,M	FINV0850
7 AAKL=AAKL+SF(K,J)*SF(L,J)	FINV0860
8 AA(K,L)=AAKL	FINV0870
RHS=0.	FINV0880
DO 108 J=1,M	FINV0890
108 RHS=RHS+SFR(J)*SF(K,J)	FINV0900
9 AA(K,NN2)=RHS	FINV0910
C---- SOLVE FOR A-S	FINV0920
CALL CROUT(AA,A,NN1,8,6)	FINV0930
C---- FORM THE ACTUAL ELEMENTS OF THE NORMAL MATRIX FOR B-S	FINV0940
DO 12 K=1,N	FINV0950
DO 11 L=1,N	FINV0960
AAKL=0.	FINV0970
DO 10 J=1,M	FINV0980
10 AAKL=AAKL+SG(K,J)*SG(L,J)	FINV0990
11 AA(K,L)=AAKL	FINV1000
RHS=0.	FINV1010
DO 111 J=1,M	FINV1020
111 RHS=RHS+SFI(J)*SG(K,J)	FINV1030
12 AA(K,NN1)=RHS	FINV1040
C---- SOLVE FOR B-S	FINV1050
CALL CROUT(AA,B,N,8,6)	FINV1060
RETURN	FINV1070
END	FINV108-

APPENDIX C

PROGRAM LISTINGS OF MAIN PROGRAM
AND ASSOCIATED SUBPROGRAMS

C	MAIN PROGRAM FOR COMPUTING FOURIER COEFFICIENTS AND SPECTRA	MAIN0000
	PARAMETER IPTS=250,JPTS=IPTS+1,KPTS=2*IPTS,MPTS=KPTS+1	MAIN0010
	PARAMETER ISPC=101,JSPC=25,KSPC=JSPC+1	MAIN0020
	PARAMETER IPTS12=IPTS+2,IPTS22=2*IPTS+2,IPTS31=3*IPTS+1	MAIN0030
	PARAMETER NPTS=2*MPTS,MPPTS=MPTS+1	MAIN0040
	COMMON /PY/PI,TWOPI	MAIN0050
	COMMON /BLOCK1/A(JPTS),B(IPTS),AK(IPTS),FTI(MPTS),EN,N,N2P1,	MAIN0060
	1TMIN,TMAX,TSUBAR,TBAR	MAIN0070
	COMMON /BLOCK2/OM(ISPC),RFR(ISPC),BFI(ISPC),TH(JSPC)	MAIN0080
	DIMENSION TITLE(12)	MAIN0090
	DIMENSION PHED(22),FOURCL(12),FCL(12),FORMT(12),FMT(12)	MAIN0100
	DIMENSION T(MPTS),FT(MPTS)	MAIN0110
	DIMENSION FNDX(NPTS),INDX(MPTS)	MAIN0120
	DIMENSION ABAK(IPTS31)	MAIN0130
	DIMENSION Y(100,5),KSIZ(5)	MAIN0140
	EQUIVALENCE (TITLE,PHED(3)),(T,INDX,FNDX),(FNDX(MPPTS),FT)	MAIN0150
	EQUIVALENCE (ABAK,A),(ABAK(IPTS12),B),(ABAK(IPTS22),AK)	MAIN0160
99	FORMAT(1H12A6)	MAIN0170
98	FORMAT(7E10.7)	MAIN0180
97	FORMAT(5H 01PE16.7,32XE16.7)	MAIN0190
96	FORMAT(15,1P4E16.7)	MAIN0200
95	FORMAT(2E10.7,3I5,2E10.7)	MAIN0210
94	FORMAT(23H1SPECTRAL DISTRIBUTION.I5,35H TERMS USED IN THIS APPROX	MAIN0220
	1IMATION.//)	MAIN0230
93	FORMAT(3X1P4E16.7)	MAIN0240
92	FORMAT(17H1FITTED FUNCTION.I5,13H TERMS USED./14X1HT14X4HF(T))	MAIN0250
91	FORMAT(6E10.7)	MAIN0260
90	FORMAT(5H0 I 7X4HA(I)12X4HB(I)11X7HFREQ(I))	MAIN0270
89	FORMAT(12A6)	MAIN0280
88	FORMAT(5H0 I 7X4HA(I)12X4HB(I)11X7HFREQ(I)11X6HRMS(I))	MAIN0290
87	FORMAT(12X4HFREQ11X6HF-REAL10X6HF-IMAG10X5HPOWER)	MAIN0300
86	FORMAT(35H1REAL PART OF SPECTRAL DISTRIBUTION,5X12A6)	MAIN0310
85	FORMAT(40H1IMAGINARY PART OF SPECTRAL DISTRIBUTION,5X12A6)	MAIN0320
84	FORMAT(23H1POWER DENSITY FUNCTION,5X12A6)	MAIN0330
83	FORMAT(24H1PLOT OF FITTED FUNCTION,5X12A6)	MAIN0340
82	FORMAT(8X13H(TMAX-TMIN)/2,1PE16.7,3X13H(TMAX+TMIN)/2,E16.7)	MAIN0350
81	FORMAT(29H SCALE FACTOR FOR INPUT DATA 1PE15.7)	MAIN0360
80	FORMAT(5X1P2E16.7)	MAIN0370
C----	FORMATS USED IN SUBROUTINE COLUMN	MAIN0380
	DATA PHED(1),PHED(2)/6HINPUT ,6HDATA /,(PHED(I),I=15,22)/	MAIN0390
1	8*6H /	MAIN0400
	DATA FOURCL/6H3H I5,6HX9HF(T,6H(I)) ,9*6H /	MAIN0410
	DATA FCL /6H5X,5H0,6HMEGA 9,6HX4HREA,6H18X9HI,6HMAGINA,6HRY	MAIN0420
1	6*6H /	MAIN0430
	DATA FORMT /6H(I4,1P,6HE14.7),10*6H /	MAIN0440
	DATA FMT /6H(1P3E1,6H4.7) ,10*6H /	MAIN0450
	DATA MAXLIN/41/	MAIN0460
	DATA PI/3.141592654/	MAIN0470

TWOPI=2.*PI	MAIN0480
CALL SETEOF(\$9)	MAIN0490
1 READ (5,89) TITLE	MAIN0500
READ (5,95) T1,T2,MM,M,KEY	MAIN0510
MM=MINO(MM,ISPC)	MAIN0520
IF(KEY.NE.0) GO TO 11	MAIN0530
READ (5,95) TMIN,TMAX,NN,NQ,NT,SF1,SF2	MAIN0540
READ (5,98) (FTI(I),I=1,NN)	MAIN0550
READ (5,95) FC1,FC2,NQ,NT	MAIN0560
C---- PRINT OUT INPUT DATA	MAIN0570
DO 601 I=1,NN	MAIN0580
INDX(I)=I	MAIN0590
J=NN+I	MAIN0600
601 FNDX(J)=FTI(I)	MAIN0610
CALL COLUMN(FNDX,NN,2,MAXLIN,5,PHED,FOURCL,FORMT)	MAIN0620
IF(ABS(SF1)+ABS(SF2)) 301,501,301	MAIN0630
301 IF(SF2) 1101,901,1101	MAIN0640
901 IF(SIGN(1.,SF2)) 1001,1101,1101	MAIN0650
1001 SFACTR=SF1	MAIN0660
GO TO 1201	MAIN0670
1101 SFACTR=10.**((SF1-SF2)/20.)	MAIN0680
1201 WRITE (6,81) SFACTR	MAIN0690
DO 401 I=1,NN	MAIN0700
401 FTI(I)=SFACTR*FTI(I)	MAIN0710
501 CONTINUE	MAIN0720
MT=NQ	MAIN0730
OMEG0=TWOPI*FC1	MAIN0740
OMEGN=TWOPI*FC2	MAIN0750
N2P1=NN	MAIN0760
C---- OBTAIN FOURIER SERIES FOR INPUT DATA POINTS.	MAIN0770
CALL FORFIT	MAIN0780
C---- PRINT OUT COEFFICIENTS AND ANGULAR FREQUENCIES	MAIN0790
MTRAN=MINO(NT,N)	MAIN0800
MT=MINO(MT,ISPC)	MAIN0810
M=MINO(M,N)	MAIN0820
ASSIGN 201 TO MSR	MAIN0830
IF(M.LE.0) ASSIGN 701 TO MSR	MAIN0840
FXI=0.	MAIN0850
DO 101 I=1,N2P1	MAIN0860
101 FXI=FXI+FTI(I)**2	MAIN0870
EPS=FXI-2.*EN*A(1)**2	MAIN0880
EPS=AMAX1(EPS,0.)	MAIN0890
TWNMI=N2P1-1	MAIN0900
EPP=SQRT(EPS/TWNMI)	MAIN0910
WRITE (6,99) TITLE	MAIN0920
WRITE (6,88)	MAIN0930
WRITE (6,97) A(1),EPP	MAIN0940
RMSMIN=EPP	MAIN0950

```

MINRMS=1
DO 201 I=1,N
  HERTZ=AK(I)/TWOPI
  EPS=EPS-EN*(A(I+1)**2+B(I)**2)
  EPS=AMAX1(EPS,0.)
  TWNMI=TWNMI-2.
  EPP=SQRT(EPS/TWNMI)
  GO TO MSR,(201,701)
C----- LOCATE MINIMUM RMS VALUE
701 IF(I.EQ.N) GO TO 201
  IF(EPP-RMSMIN) 801,201,201
801 RMSMIN=EPP
  MINRMS=I
201 WRITE (6,96) I,A(I+1),B(I),HERTZ,EPP
  WRITE (6,96)
  WRITE (6,96)
  WRITE (6,82) TSUBAR,TBAR
  IF(M.LE.0) M=MINRMS
  IF(MM.LE.0) GO TO 104
C----- EVALUATE SERIES FOR OUTPUT
  T(1)=T1
  FXI=A(1)
  THE=T(1)-TBAR
  DTT=(T2-T1)/FLOAT(MM-1)
  DO 2 K=1,M
    ANG=AK(K)*THE
2  FXI=FXI+A(K+1)*COS(ANG)+B(K)*SIN(ANG)
  FT(1)=FXI
  DO 4 I=2,MM
    T(I)=T(I-1)+DTT
    THE=THE+DTT
    FXI=A(1)
    DO 3 K=1,M
      ANG=AK(K)*THE
3  FXI=FXI+A(K+1)*COS(ANG)+B(K)*SIN(ANG)
  FT(I)=FXI
4  CONTINUE
  WRITE (6,92) M
  WRITE (6,80) (T(I),FT(I),I=1,MM)
  WRITE (6,83) TITLE
C----- PLOT ABOVE DATA
  KSI(1)=MM
  KSI(2)=0
  CALL PLOT(40,ISPC,T,MM,KSI,FT(1),FT(1),FT(1),FT(1),FT(1))
104 IF(MT.LE.0) GO TO 1
  IF(NT.LE.0) MTRAN=M
C----- OBTAIN TRANSFORM OF DATA
  OD=(OMEGN-OMEG0)/FLOAT(MT-1)

```

```

MAIN0960
MAIN0970
MAIN0980
MAIN0990
MAIN1000
MAIN1010
MAIN1020
MAIN1030
MAIN1040
MAIN1050
MAIN1060
MAIN1070
MAIN1080
MAIN1090
MAIN1100
MAIN1110
MAIN1120
MAIN1130
MAIN1140
MAIN1150
MAIN1160
MAIN1170
MAIN1180
MAIN1190
MAIN1200
MAIN1210
MAIN1220
MAIN1230
MAIN1240
MAIN1250
MAIN1260
MAIN1270
MAIN1280
MAIN1290
MAIN1300
MAIN1310
MAIN1320
MAIN1330
MAIN1340
MAIN1350
MAIN1360
MAIN1370
MAIN1380
MAIN1390
MAIN1400
MAIN1410
MAIN1420
MAIN1430

```

O=OMEGA-OD	MAIN1440
DO 5 I=1,MT	MAIN1450
O=O+OD	MAIN1460
CALL F(O,MTRAN,FREAL,FIMAG)	MAIN1470
Y(I,5)=O	MAIN1480
Y(I,3)=O/TWOPI	MAIN1490
Y(I,1)=FREAL	MAIN1500
Y(I,2)=FIMAG	MAIN1510
5 Y(I,4)=SQRT(FREAL**2+FIMAG**2)	MAIN1520
KSIZ(1)=MT	MAIN1530
WRITE (6,94) MTRAN	MAIN1540
WRITE (6,87)	MAIN1550
WRITE(6,93) (Y(I,3),Y(I,1),Y(I,2),Y(I,4),I=1,MT)	MAIN1560
WRITE (6,86) TITLE	MAIN1570
CALL PLOT(40,ISPC,Y(1,3),MT,KSIZ,Y(1,1),Y(1,2),Y(1,3),Y(1,4),	MAIN1580
1Y(1,5))	MAIN1590
WRITE (6,85) TITLE	MAIN1600
CALL PLOT(40,ISPC,Y(1,3),MT,KSIZ,Y(1,2),Y(1,2),Y(1,3),Y(1,4),	MAIN1610
1Y(1,5))	MAIN1620
WRITE (6,84) TITLE	MAIN1630
CALL PLOT(40,ISPC,Y(1,3),MT,KSIZ,Y(1,4),Y(1,2),Y(1,3),Y(1,4),	MAIN1640
1Y(1,5))	MAIN1650
GO TO 1	MAIN1660
9 CALL EXIT	MAIN1670
C---- READ IN SPECTRAL DATA	MAIN1680
11 READ (5,95) FC1,FC2,NQ,NT,NN,SFACTR	MAIN1690
READ (5,91) (OM(I),BFR(I),BFI(I),I=1,NQ)	MAIN1700
READ (5,95) TMIN,TMAX,NN	MAIN1710
C---- PRINT OUT INPUT DATA	MAIN1720
DO 511 I=1,NQ	MAIN1730
ABAK(I)=OM(I)	MAIN1740
J=NQ+I	MAIN1750
ABAK(J)=BFR(I)	MAIN1760
J=NQ+J	MAIN1770
511 ABAK(J)=BFI(I)	MAIN1780
CALL COLUMN(ABAK,NQ,3,MAXLIN,2,PHED,FCL,FMT)	MAIN1790
IF(SFACTR) 211,411,211	MAIN1800
211 WRITE (6,81) SFACTR	MAIN1810
DO 311 I=1,NQ	MAIN1820
BFR(I)=SFACTR*BFR(I)	MAIN1830
311 BFI(I)=SFACTR*BFI(I)	MAIN1840
411 CONTINUE	MAIN1850
CALL FINV(OM,BFR,BFI,NQ,TMIN,TMAX,A,B,TH,NT,TSUBAR,TBAR,EN,N)	MAIN1860
WRITE (6,99) TITLE	MAIN1870
WRITE (6,90)	MAIN1880
WRITE (6,97) A(1)	MAIN1890
DO 111 I=1,N	MAIN1900
HERTZ=TH(I)/TWOPI	MAIN1910

111	WRITE (6,96) I,A(I+1),B(I),HERTZ	MAIN1920
	WRITE (6,96)	MAIN1930
	WRITE (6,96)	MAIN1940
	WRITE (6,82) TSUBAR,TBAR	MAIN1950
C-----	EVALUATE SERIES FOR OUTPUT	MAIN1960
	T(1)=T1	MAIN1970
	FXI=A(1)	MAIN1980
	THE=T(1)-TBAR	MAIN1990
	DTT=(T2-T1)/FLOAT(MM-1)	MAIN2000
	DO 12 K=1,M	MAIN2010
	ANG=TH(K)*THE	MAIN2020
12	FXI=FXI+A(K+1)*COS(ANG)+B(K)*SIN(ANG)	MAIN2030
	FT(1)=FXI	MAIN2040
	DO 14 I=2,MM	MAIN2050
	T(I)=T(I-1)+DTT	MAIN2060
	THE=THE+DTT	MAIN2070
	FXI=A(1)	MAIN2080
	DO 13 K=1,M	MAIN2090
	ANG=TH(K)*THE	MAIN2100
13	FXI=FXI+A(K+1)*COS(ANG)+B(K)*SIN(ANG)	MAIN2110
	FT(I)=FXI	MAIN2120
14	CONTINUE	MAIN2130
	WRITE (6,92) N	MAIN2140
	WRITE (6,80) (T(I),FT(I),I=1,MM)	MAIN2150
	WRITE (6,83) TITLE	MAIN2160
C-----	PLOT ABOVE DATA	MAIN2170
	KSIZ(1)=MM	MAIN2180
	KSIZ(2)=0	MAIN2190
	CALL PLOT(40,ISPC,T,MM,KSIZ,FT(1),FT(1),FT(1),FT(1),FT(1))	MAIN2200
	GO TO 1	MAIN2210
	END	MAIN222-

```

SUBROUTINE COLUMN(ARRAY,IROW,ICOL,MAXLIN,MAXCOL,PHED,CHED,CFMT)  COLU0000
C---- ARRAY IS A 2 DIM ARRAY TO BE PRINTED IN COLUMNAR FASHION  COLU0010
C---- IROW IS THE NO OF ROWS IN ARRAY  COLU0020
C---- ICOL IS THE NO OF COLUMNS IN ARRAY  COLU0030
C---- MAXLIN IS THE MAXIMUM NO OF LINES PER PAGE  COLU0040
C---- MAXCOL IS THE MAXIMUM NO OF COLUMNS PER PAGE  COLU0050
C---- PHED IS 131 ALPHAMERIC CHARACTERS TO BE PRINTED AT THE TOP OF PAGE  COLU0060
C---- CHED IS 72 ALPHAMERIC CHARACTERS USED FOR COLUMN HEADINGS. THE  COLU0070
C      COLUMN HEADING MUST BE LEFT JUSTIFIED AND THE REMAINDER OF  COLU0080
C      THE 72 CHARACTERS MUST BE BLANKS. COLUMN WILL PLACE ONE  COLU0090
C      BLANK BETWEEN EACH COLUMN HEADING PRINTED. CHED MUST BE  COLU0100
C      FORMAT-LIKE BUT WITHOUT ENCLOSING PARENTHESES.  COLU0110
C---- CFMT IS THE FORMAT FOR EACH COLUMN (INCLUDES ENCLOSING PARENS)  COLU0120
C      COLU0130
C---- ARRAY WILL BE WRITTEN OUT AS FOLLOWS THE FIRST MAXLIN ROWS OF  COLU0140
C      ARRAY WILL MAKE UP THE FIRST COLUMN OF OUTPUT ON THE FIRST  COLU0150
C      PAGE. THE 2-ND MAXLIN ROWS WILL MAKE THE 2-ND COLUMN OF  COLU0160
C      OUTPUT, ETC. UNTIL THE FIRST PAGE HAS MAXLIN*MAXCOL ROWS  COLU0170
C      OF ARRAY ARE PRINTED. THE 2-ND PAGE ETC.  COLU0180
      DIMENSION PHED(22),CHED(12),CFMT(12),ARRAY(IROW,ICOL),COLHED(15),  COLU0190
      1COLFMT(14),HEAD(64)  COLU0200
      DATA COLHED(1),COLHED(2)/12H(1H064(A1, / ,COLHED(15)/6H)) / ,  COLU0210
      1COLFMT(1)/6H(1X,64/,COLFMT(14)/6H) / ,HEAD/64*6H /  COLU0220
99 FORMAT(1H121A6,A5)  COLU0230
98 FORMAT(1H0)  COLU0240
      DO 1 I=1,12  COLU0250
        COLHED(I+2)=CHED(I)  COLU0260
1 COLFMT(I+1)=CFMT(I)  COLU0270
      LINE=MAXLIN  COLU0280
      NPP=MAXCOL*MAXLIN  COLU0290
      NOPGE=(IROW-1)/NPP+1  COLU0300
      J2=0  COLU0310
      IGS=0  COLU0320
      NTOP=MAXLIN  COLU0330
      DO 4 NP=1,NOPGE  COLU0340
        IGS=MIN0(IGS+NPP,IROW)  COLU0350
        J1=J2  COLU0360
        J2=IGS-MOD(IGS,MAXLIN)  COLU0370
        NTOP=MIN0(NTOP,IGS)  COLU0380
        DO 4 NL=1,NTOP  COLU0390
          J1=J1+1  COLU0400
          IF(J2.EQ.IGS) J2=J2-MAXLIN  COLU0410
          J2=J2+1  COLU0420
          JJJ=(J2-J1)/MAXLIN+1  COLU0430
          IF(LINE-MAXLIN) 3,2,2  COLU0440
2 LINE=0  COLU0450
          WRITE (6,99) PHED  COLU0460
          WRITE (6,COLHED) (HEAD(I),I=1,JJJ)  COLU0470
          WRITE (6,98)  COLU0480
3 LINE=LINE+1  COLU0490
          WRITE (6,COLFMT) ((ARRAY(J,I),I=1,ICOL),J=J1,J2,MAXLIN)  COLU0500
4 CONTINUE  COLU0510
      RETURN  COLU0520
      END  COLU053-

```

```

SUBROUTINE CROUT(AA,C,INPUTN,NC,NF)                                CROU0000
C----- SOLVES A SYSTEM OF N LINEAR ALGEBRAIC EQUATIONS IN N UNKNOWN BY CROU0010
C      THE CROUT METHOD. THE PROGRAM WAS ORIGINALLY WRITTEN BY CROU0020
C      RL REYNOLDS AND LATER MODIFIED BY CA OSTER. CROU0030
C      CROU0040
C----- A IS N,N+1 MATRIX OF INPUT COEFFICIENTS CROU0050
C      C IS N,1 COLUMN MATRIX OF UNKNOWNNS AS OUTPUT CROU0060
C      NC IS MAX NUMBER OF ITERATIONS TO BE PERMITTED (6 - 10 IS ADVISED) CROU0070
C      NF IS NUMBER OF SIGNIFICANT FIGURES OF ACCURACY REQUIRED IN CROU0080
C      BALANCE OF TWO SIDES OF EACH EQUATION (NOT TO EXCEED 6) CROU0090
C      CROU0100
C----- (REVISION FOR SPECIFIC APPLICATIONS - A CHANGE IN THE PARAMETER CROU0110
C      STATEMENT DOES NOT REQUIRE ANY ADDITIONAL PROGRAM CHANGES) CROU0120
C----- REF- AIEE TRANS 1941 VOL 60 P1235-1240 CROU0130
C      PARAMETER JPTS=25,NQ=JPTS+1,NQ1=JPTS+2 CROU0140
C      DIMENSION A(NQ,NQ1),B(NQ,NQ1),C(NQ),D(NQ),AA(NQ,NQ1),AL(NQ1) CROU0150
100 FORMAT (84HACCURACY OF SIMULTANEOUS EQUATION SOLUTION HAS NOT REACROU0160
11CHED THE REQUIRED VALUE AFTER I3,13H CORRECTIONS./59H MAXIMUM ERRORCROU0170
12 IN BALANCE OF TWO SIDES OF EACH EQUATION IS 1PE10.3,8H IN THE I1,CROU0180
1322HTH SIGNIFICANT FIGURE.) CROU0190
101 FORMAT (119H0SUBROUTINE CROUT5 REQUIRES THAT AT LEAST ONE INPUT EQCROU0200
11UATION HAVE A NON-ZERO FIRST COEFFICIENT. NO SOLUTION IS POSSIBLE)CROU0210
102 FORMAT (119H0SUBROUTINE CROUT5 IS UNABLE TO FORM AN AUXILIARY MATRCROU0220
11X WITHOUT A ZERO ON THE DIAGONAL. NO VALID SOLUTION IS POSSIBLE.)CROU0230
104 FORMAT (101H0NO SOLUTION IS POSSIBLE FROM CROUT5 BECAUSE OF AN INSCROU0240
11UFFICIENT NUMBER OF EQUATIONS IN THE INPUT SET./4H ROW I3,28H IS ACROU0250
12N EXACT MULTIPLE OF ROW I3,1H.) CROU0260
105 FORMAT(7H0COLUMN I5,21H IS IDENTICALLY ZERO.) CROU0270
106 FORMAT(7H0 ROW I5,21H IS IDENTICALLY ZERO.) CROU0280
107 FORMAT(7H0COLUMN I5,31H IS AN EXACT MULTIPLE OF COMUMNI5,1H.) CROU0290
N=INPUTN CROU0300
ZFAC=10.**(NF-1) CROU0310
KOUNT = 0 CROU0320
N1L=1 CROU0330
NP=N+1 CROU0340
NM=N-1 CROU0350
DO 201 I=1,N CROU0360
DO 201 J=1,NP CROU0370
201 A(I,J)=AA(I,J) CROU0380
C----- TESTS FOR INVALID INPUT DATA CROU0390
DO 401 J=1,N CROU0400
AJ=0. CROU0410
DO 301 I=1,N CROU0420
301 AJ=MAX1(AJ,ABS(A(I,J))) CROU0430
IF(AJ) 83,83,401 CROU0440
401 CONTINUE CROU0450
DO 601 I=1,N CROU0460
AJ=0. CROU0470

```


DO 501 J=1,N	CROU0480
501 AJ=MAX1(AJ,ABS(A(I,J)))	CROU0490
IF(AJ) 84,84,601	CROU0500
601 CONTINUE	CROU0510
IZ=2	CROU0520
1 IF (A(1,1)) 6,2,6	CROU0530
202 DO 302 J=1,N	CROU0540
302 A(J,NP)=AL(J)	CROU0550
NIL=3	CROU0560
2 IF (IZ-N) 3,3,5	CROU0570
3 DO 4 JZ=1,NP	CROU0580
AL(JZ)=A(1,JZ)	CROU0590
A(1,JZ)=A(IZ,JZ)	CROU0600
4 A(IZ,JZ)=AL(JZ)	CROU0610
IZ=IZ+1	CROU0620
GO TO 1	CROU0630
5 GO TO (71,71,72), NIL	CROU0640
6 GO TO (7,7,12), NIL	CROU0650
7 DO 11 I=1,NM	CROU0660
IP=I+1	CROU0670
DO 11 K=IP,N	CROU0680
DO 209 J=1,NP	CROU0690
KKI=J+1	CROU0700
FACTOR = A(I,J)/A(K,J)	CROU0710
IF(FACTOR) 309,1209,309	CROU0720
1209 P=ABS(A(I,J))	CROU0730
Q=ABS(A(K,J))	CROU0740
IF(P+Q)11,209,11	CROU0750
209 CONTINUE	CROU0760
GO TO 11	CROU0770
309 IF(KKI-NP) 409,409,11	CROU0780
409 DO 10 J=KKI,NP	CROU0790
IF(ABS((FACTOR-A(I,J)/A(K,J))/FACTOR)-.5E-7) 10,10,11	CROU0800
10 CONTINUE	CROU0810
GO TO 74	CROU0820
11 CONTINUE	CROU0830
DO 511 J=1,NM	CROU0840
JP=J+1	CROU0850
DO 511 K=JP,N	CROU0860
DO 211 I=1,N	CROU0870
KKI=I+1	CROU0880
FACTOR=A(I,J)/A(I,K)	CROU0890
IF(FACTOR) 311,1211,311	CROU0900
1211 P=ABS(A(I,J))	CROU0910
Q=ABS(A(I,K))	CROU0920
IF(P+Q) 511,211,511	CROU0930
211 CONTINUE	CROU0940
GO TO 511	CROU0950

311 IF(KKI-N) 1311,1311,511	CROU0960
1311 DO 411 I=KKI,N	CROU0970
IF(ABS((FACTOR-A(I,J)/A(I,K))/FACTOR)-.5E-7) 411,411,511	CROU0980
411 CONTINUE	CROU0990
GO TO 85	CROU1000
511 CONTINUE	CROU1010
C---- CONSTRUCTION OF INTERMEDIATE MATRIX	CROU1020
12 DO 13 I=1,N	CROU1030
D(I)=0.0	CROU1040
AL(I)=A(I,NP)	CROU1050
13 B(I,1)=A(I,1)	CROU1060
14 DO 15 I=1,N	CROU1070
15 B(1,I+1)=A(1,I+1)/A(1,1)	CROU1080
DO 17 L=2,N	CROU1090
DO 17 I=L,N	CROU1100
SM=0.	CROU1110
SP=0.	CROU1120
TM=0.	CROU1130
TP=0.	CROU1140
DO 16 J=2,L	CROU1150
S=B(I,J-1)*B(J-1,L)	CROU1160
T=B(L,J-1)*B(J-1,I+1)	CROU1170
IF(S) 215,415,315	CROU1180
215 SM=SM+S	CROU1190
GO TO 415	CROU1200
315 SP=SP+S	CROU1210
415 IF(T) 515,16,615	CROU1220
515 TM=TM+T	CROU1230
GO TO 16	CROU1240
615 TP=TP+T	CROU1250
16 CONTINUE	CROU1260
S=SP+SM	CROU1270
T=TP+TM	CROU1280
B(I,L)=A(I,L)-S	CROU1290
17 B(L,I+1)=(A(L,I+1)-T)/B(L,L)	CROU1300
C---- TEST FOR ZERO DIAGONAL IN AUXILIARY MATRIX	CROU1310
DO 23 I=1,N	CROU1320
IF (B(I,I)) 23,202,23	CROU1330
23 CONTINUE	CROU1340
C---- CALCULATION OF MATRIX OF VARIABLES (BACK SUBSTITUTION)	CROU1350
31 KOUNT=KOUNT+1	CROU1360
C(N)=B(N,NP)	CROU1370
DO 33 I=2,N	CROU1380
K=NP-I	CROU1390
KM1=K-1	CROU1400
SM=0.	CROU1410
SP=0.	CROU1420
DO 32 L=2,I	CROU1430

KL=KM1+L	CROU1440
S=C(KL)*B(K,KL)	CROU1450
IF(S) 231,32,331	CROU1460
231 SM=SM+S	CROU1470
GO TO 32	CROU1480
331 SP=SP+S	CROU1490
32 CONTINUE	CROU1500
S=SP+SM	CROU1510
33 C(K)=B(K,NP)-S	CROU1520
C----- CORRECTION OF ELEMENTS OF VARIABLES MATRIX	CROU1530
DO 41 I=1,N	CROU1540
41 C(I)=C(I)+D(I)	CROU1550
C----- CALCULATION OF NEXT CORRECTION	CROU1560
DO 52 I=1,N	CROU1570
SM=0.	CROU1580
SP=0.	CROU1590
DO 51 J=1,N	CROU1600
S=A(I,J)*C(J)	CROU1610
IF(S) 50,51,250	CROU1620
50 SM=SM+S	CROU1630
GO TO 51	CROU1640
250 SP=SP+S	CROU1650
51 CONTINUE	CROU1660
A(I,NP)=AL(I)-(SP+SM)	CROU1670
52 D(I)=C(I)	CROU1680
C----- CHECK FOR ACCURACY	CROU1690
NIL=1	CROU1700
ZOUNDS=0.0	CROU1710
DO 69 I=1,N	CROU1720
ZZ=0.	CROU1730
DO 61 J=1,N	CROU1740
61 ZZ=ZZ+C(J)*A(I,J)	CROU1750
ZZ=ABS(AL(I)-ZZ)	CROU1760
Z=ABS(AL(I))	CROU1770
IF (ZZ/Z-0.5E-7) 69,69,62	CROU1780
62 IF (1.0-Z) 65,66,63	CROU1790
63 Z=10.0*Z	CROU1800
ZZ=10.0*ZZ	CROU1810
GO TO 62	CROU1820
64 IF (1.0-Z) 65,66,66	CROU1830
65 Z=0.1*Z	CROU1840
ZZ=0.1*ZZ	CROU1850
GO TO 64	CROU1860
66 ZZ=ZZ*ZFAC	CROU1870
IF (1.0-ZZ) 67,67,69	CROU1880
67 IF (KOUNT - NC) 70,70,68	CROU1890
68 NIL=2	CROU1900
ZOUNDS=MAX1(ZOUNDS,ZZ)	CROU1910

69	CONTINUE	CROU1920
	GO TO (81,77,81), NIL	CROU1930
C----	MAKE CORRECTIONS	CROU1940
70	B(1,NP)=A(1,NP)/B(1,1)	CROU1950
	DO 870 I=2,N	CROU1960
	IM=I-1	CROU1970
	IF(A(I,NP)) 270,370,370	CROU1980
270	TM=A(I,NP)	CROU1990
	TP=0.	CROU2000
	GO TO 470	CROU2010
370	TM=0.	CROU2020
	TP=A(I,NP)	CROU2030
470	DO 770 K=1,IM	CROU2040
	T=B(I,K)*B(K,NP)	CROU2050
	IF(T) 570,770,670	CROU2060
570	TP=TP-T	CROU2070
	GO TO 770	CROU2080
670	TM=TM-T	CROU2090
770	CONTINUE	CROU2100
870	B(I,NP)=(TP+TM)/B(I,I)	CROU2110
	GO TO 31	CROU2120
C----	MESSAGES FOR DUBIOUS SOLUTION OR INVALID INPUT	CROU2130
71	WRITE (6,101)	CROU2140
	GO TO 75	CROU2150
72	WRITE (6,102)	CROU2160
	GO TO 75	CROU2170
74	WRITE (6,104) K,I	CROU2180
75	DO 76 I=1,N	CROU2190
76	C(I)=0.0	CROU2200
	GO TO 81	CROU2210
77	WRITE (6,100) NC,ZOUNDS,NF	CROU2220
C----	PREPARATION FOR RETURN	CROU2230
81	RETURN	CROU2240
83	WRITE (6,105) J	CROU2250
	GO TO 75	CROU2260
84	WRITE (6,106) I	CROU2270
	GO TO 75	CROU2280
85	WRITE (6,107) K,I	CROU2290
	GO TO 75	CROU2300
	END	CROU231-

```

SUBROUTINE PLOT(KLIN,KOL,X,IX,KSIZ,A,B,C,D,E)          PLOT0000
PARAMETER ISPC=101                                     PLOT0010
C----- IF ISPC IS CHANGED FORMATS 2,3, AND 4 SHOULD ALSO BE CHANGED PLOT0020
C THIS PROGRAM WAS ORIGNALLY WRITTEN BY JR LILLY AND MODIFIED BY PLOT0030
C BY DR SKEEN AND CA OSTER PLOT0040
C----- KLIN IS NUMBER OF LINES TO BE USED IN PLOT PLOT0050
C KOL IS NJMBER OF COLUMNS USED IN PLOT(GREATER THAN 1). IF GREATER PLOT0060
C THAN ISPC,ONLY DATA IN FIRST ISPC COLUMNS WILL BE PLOTTED. PLOT0070
C NVAR IS NUMBER OF DEPENDENT VARIABLES TO BE PLOTTED(FROM 1 THRU 5) PLOT0080
C X IS INDEPENDENT VARIABLE IN FLOATING POINT(IX LESS THAN 101) PLOT0090
C IX IS NUMBER SUCCESSIVE VALUES OF X IN ARRAY WHICH ARE SPECIFIED. PLOT0100
C KSIZ IS AN ARRAY WHOSE COMPONENTS GIVE THE NUMBERS OF VALUES OF PLOT0110
C THE DEPENDENT VARIABLES A,B,C,D,AND E,,RESPECTIVELY,WHICH ARE PLOT0120
C TO BE PLOTTED. ZERO ENTRIES STOP PLOTTING OF THAT AND PLOT0130
C SUCCEEDING ARRAYS. PLOT0140
C ALL VARIABLES ARE PLOTTED ON A ZERO TO ONE SCALE. NORMALIZING PLOT0150
C NUMBERS ARE GIVEN. PLOT0160
C THIS SUBROUTINE DOES NOT CHANGE PAGES ON THE PRINTER. PLOT0170
C WHEN PLOTTED POINTS FOR TWO CURVES COINCIDE,ONLY THE ONE WHOSE PLOT0180
C SYMBOL COMES LATEST IN THE ALPHABET IS PLOTTED PLOT0190
C----- VAR IS A DUMMY VARIABLE, USED TO GET AN ABSOLUTE LOCATION ONLY. PLOT0200
C THE FACT THAT IT IS LISTED IN COMMON HAS NO EFFECT ON COMMON- PLOT0210
C STORED VARIABLES IN MAIN PROGRAM PLOT0220
COMMON VAR(1) PLOT0230
DIMENSION X(ISPC),KSIZ(5),A(ISPC),B(ISPC),C(ISPC),D(ISPC),E(ISPC), PLOT0240
1 LXC(5),LXC1(5),SYMB(ISPC),MARK1(ISPC) PLOT0250
1 FORMAT(51H DEPENDENT VARIABLE RANGE ON ZERO TO ONE PLOT IS 1PE10 PLOT0260
1.3,4H TO 1PE10.3,16H. PLOT ERROR IS 0PF4.2,23H PERCENT OF FULL SCAPLOT0270
2LT.) PLOT0280
2 FORMAT(9X,1H*101A1,1H*) PLOT0290
3 FORMAT(3X,F5.3,2H-*101A1,2H*-F5.3) PLOT0300
4 FORMAT(4X, 113HC.00-***** PLOT0310
1*****-0.00/ PLOT0320
210X,1H19X,1H19X,1H19X,1H19X,1H19X,1H19X,1H19X,1H19X,1H19X,1H19X, PLOT0330
31H1/9X,3H0.07X,3H0.17X,3H0.27X,3H0.37X,3H0.47X,3H0.57X,3H0.67X, PLOT0340
43HC.77X,3H0.87X,3HC.97X,3H1.0) PLOT0350
5 FORMAT(51H INDEPENDENT VARIABLE RANGE ON ZERO TO ONE PLOT IS 1PE10 PLOT0360
1.3,4H TO 1PE10.3,16H. PLOT ERROR IS 0PF4.2,23H PERCENT OF FULL SCAPLOT0370
2LE.) PLOT0380
DATA BLSY/1H /,ASYM/1HA/,BSYM/1HB/,CSYM/1HC/,DSYM/1HD/,ESYM/1HE/ PLOT0390
20 XMIN=X(1) PLOT0400
XMAX=X(1) PLOT0410
C----- FIND RANGE OF INDEPENDENT VARIABLE PLOT0420
DO 40 I=1,IX PLOT0430
IF(X(I)-XMAX)30,40,25 PLOT0440
25 XMAX=X(I) PLOT0450
GO TO 40 PLOT0460
30 IF(X(I)-XMIN)35,40,40 PLOT0470

```

35 XMIN=X(I)	PLOT0480
40 CONTINUE	PLOT0490
COL=KOL-1	PLOT0500
C---- COMPUTE PRINT WHEEL TO BE USED FOR EACH X(I)	PLOT0510
COLOXM=COL/(XMAX-XMIN)	PLOT0520
DO 45 I=1,IX	PLOT0530
45 MARKI(I)=IFIX((X(I)-XMIN)*COLOXM+.5)+1	PLOT0540
C---- LOCATE DEPENDENT ARRAYS,RELATIVE TO VAR. DETERMINE NVAR	PLOT0550
50 DO 90 L=1,5	PLOT0560
IF(KSIZ(L))55,55,60	PLOT0570
55 NVAR=L-1	PLOT0580
GO TO 95	PLOT0590
60 GO TO(65,70,75,80,85),L	PLOT0600
65 LXC(L) = LOC(A(1))-LOC(VAR(1))+1	PLOT0610
LXC1(L)=LXC(L)+KSIZ(L)-1	PLOT0620
GO TO 90	PLOT0630
70 LXC(L) = LOC(B(1))-LOC(VAR(1))+1	PLOT0640
LXC1(L)=LXC(L)+KSIZ(L)-1	PLOT0650
GO TO 90	PLOT0660
75 LXC(L) = LOC(C(1))-LOC(VAR(1))+1	PLOT0670
LXC1(L)=LXC(L)+KSIZ(L)-1	PLOT0680
GO TO 90	PLOT0690
80 LXC(L) = LOC(D(1))-LOC(VAR(1))+1	PLOT0700
LXC1(L)=LXC(L)+KSIZ(L)-1	PLOT0710
GO TO 90	PLOT0720
85 LXC(L) = LOC(E(1))-LOC(VAR(1))+1	PLOT0730
LXC1(L)=LXC(L)+KSIZ(L)-1	PLOT0740
90 CONTINUE	PLOT0750
C---- WE HAVE FOUND LOCATIONS OF ALL DEPENDENT VARIABLES RELATIVE TO VAR	PLOT0760
NVAR=5	PLOT0770
C---- FIND MAXIMUM AND MINIMUM OF DEPENDENT VARIABLES	PLOT0780
95 AMIN=A(1)	PLOT0790
AMAX=A(1)	PLOT0800
DO 120 L=1,NVAR	PLOT0810
L1=LXC(L)	PLOT0820
L2=LXC1(L)	PLOT0830
DO 115 LL=L1,L2	PLOT0840
IF(VAR(LL)-AMAX)105,115,100	PLOT0850
100 AMAX =VAR(LL)	PLOT0860
GO TO 115	PLOT0870
105 IF(VAR(LL)-AMIN)110,115,115	PLOT0880
110 AMIN =VAR(LL)	PLOT0890
115 CONTINUE	PLOT0900
120 CONTINUE	PLOT0910
C---- NORMALIZE DEPENDENT VARIABLES TO RANGE FROM ONE THRU KLIN	PLOT0920
AB=KLIN-1	PLOT0930
ABOMAX=AB/(AMAX-AMIN)	PLOT0940
C---- NOW FOR THE PLOT	PLOT0950

ZMAX=AMAX-(AB+1.)/ABOMAX	PLOT0960
ERA=50./ (AB+1.)	PLOT0970
WRITE (6,1)ZMAX,AMAX,ERA	PLOT0980
VLINE=KLIN+1	PLOT0990
DO 200 LINE=1,KLIN	PLOT1000
VLINE=VLINE-1.	PLOT1010
DO 135 I=1,ISPC	PLOT1020
135 SYMB(I) = BLSY	PLOT1030
C----- THIS GIVES A BLANK FOR FIRST BCD CHARACTER IN EACH WORD	PLOT1040
DO185 J=1,NVAR	PLOT1050
GO TO (140,145,150,155,160),J	PLOT1060
140 SY = ASYM	PLOT1070
GO TO 165	PLOT1080
145 SY = BSYM	PLOT1090
GO TO 165	PLOT1100
150 SY = CSYM	PLOT1110
GO TO 165	PLOT1120
155 SY = DSYM	PLOT1130
GO TO 165	PLOT1140
160 SY = ESYM	PLOT1150
165 L1=LXC(J)	PLOT1160
L2=LXC1(J)	PLOT1170
L1M1=L1-1	PLOT1180
DO 180 LL=L1,L2	PLOT1190
VARLL=(VAR(LL)-AMIN)*ABOMAX+.5-VLINE	PLOT1200
C----- TEST IF LESS THAN MAX. ALLOWED VALUE FOR LINE	PLOT1210
IF(VARLL) 170,175,180	PLOT1220
C----- TEST IF GREATER THAN MINIMUM VALUE	PLOT1230
170 IF(VARLL+1.) 180,175,175	PLOT1240
175 I=LL-L1M1	PLOT1250
MARK=MARKI(I)	PLOT1260
IF(MARK-ISPC)178,178,180	PLOT1270
178 SYMB(MARK)= SY	PLOT1280
180 CONTINUE	PLOT1290
185 CONTINUE	PLOT1300
IF(LINE-1-5*((LINE-1)/5))195,190,195	PLOT1310
C----- WRITE ORDINATE EVERY 5-TH LINE	PLOT1320
190 A1=1.- FLOAT(LINE-1)/(AB+1.)	PLOT1330
WRITE(6,3)A1,SYMB,A1	PLOT1340
GO TO 200	PLOT1350
195 WRITE(6,2)SYMB	PLOT1360
200 CONTINUE	PLOT1370
C----- PLOT COMPLETED	PLOT1380
ZMAX=XMIN+(COL+1.)/COLOXM	PLOT1390
ERA=50./ (COL+1.)	PLOT1400
WRITE (6,4)	PLOT1410
WRITE(6,5) XMIN,ZMAX,ERA	PLOT1420
RETURN	PLOT1430
END	PLOT144-

APPENDIX D

SAMPLE INPUT CASES

Y=1-2*COS(X-PI)-3*SIN(X-PI)-2*COS(3*(X-PI))-3*SIN(3*(X-PI)) X IN (0,2PI) 0-1-0000
 0.0000000 6.2831853 41 0 0-2-0001
 0.0000000 6.2831853 13 1-1-0002
 5.0000000 7.2320508 2.5980762 1.0000000 4.5980762 3.7679492-3.0000000 1-2-0003
 -5.2320508-.5980762 1.0000000-2.5980762-1.7679492 5.0000000 1-2-0104
 0.0000000 1.0000000 41 6 1-3-0005

Y=1-2*COS(X-PI)-3*SIN(X-PI)-2*COS(3*(X-PI))-3*SIN(3*(X-PI)) X IN (0,2PI) 0-1-0000
 0.0000000 6.2831853 41 6 1 0-2-0001
 0.0000000 1.0000000 41 6 6.2831853 2-1-0002
 .0000000 1.0000000 .0000000 .1570796 1.0892704 .1124153 2-2-0003
 .3141593 1.3152197 .1000768 .4712389 1.5642059 -.1139403 2-2-0104
 .6283185 1.6891501 -.5195919 .7853981 1.5723011-1.0082597 2-2-0205
 .9424778 1.1902408-1.4085165 1.0995574 .6411721-1.5657977 2-2-0306
 1.2566370 .1050255-1.4261307 1.4137166 -.2504399-1.0674781 2-2-0407
 1.5707963 -.3591827 -.6498102 1.7278759 -.2776161 -.3138504 2-2-0508
 1.8849555 -.1194575 -.1025849 2.0420352 .0438059 .0329496 2-2-0609
 2.1991148 .2328779 .1489365 2.3561943 .5142687 .2198946 2-2-0710
 2.5132740 .8934418 .1313684 2.6703536 1.2437814 -.2244523 2-2-0811
 2.8274332 1.3496732 -.8207255 2.9845128 1.0518149-1.4479762 2-2-0912
 3.1415924 .3903687-1.8077936 3.2986720 -.3739295-1.6993736 2-2-1013
 3.4557516 -.8977438-1.1665071 3.6128312 -.9669732 -.4886017 2-2-1114
 3.7699109 -.6325836 -.0094200 3.9269904 -.1690143 .0801624 2-2-1215
 4.0840700 .1171723 -.1490379 4.2411496 .0897298 -.4395530 2-2-1316
 4.3982292 -.1485958 -.5448819 4.5553088 -.3623362 -.3998969 2-2-1417
 4.7123884 -.3736649 -.1418109 4.8694679 -.1884842 .0209929 2-2-1518
 5.0265475 .0310402 -.0206071 5.1836271 .1121365 -.2013079 2-2-1619
 5.3407066 .0107418 -.3514109 5.4977863 -.1652521 -.3436997 2-2-1720
 5.6548658 -.2575599 -.1929460 5.8119453 -.1915414 -.0313848 2-2-1821
 5.9690250 -.0308423 .0101831 6.1261045 .0863255 -.0890457 2-2-1922
 6.2831841 .0681300 -.2306319 2-2-2023
 0.0000000 6.2831853 13 2-3-0024

APPENDIX E
SAMPLE OUTPUT

INPUT DATA Y=1-2*COS(X-PI)-3*SIN(X-PI)-2*COS(3*(X-PI))-3*SIN(3*(X-PI)) X IN (0,2PI)

I F(T(I))

1	5.0000000+00
2	7.2320507+00
3	2.5980762+00
4	1.0000000+00
5	4.5980762+00
6	3.7679492+00
7	-3.0000000+00
8	-5.2320508+00
9	-5.9807621-01
10	1.0000000+00
11	-2.5980762+00
12	-1.7679492+00
13	5.0000000+00

Y=1-2*COS(X-PI)-3*SIN(X-PI)-2*COS(3*(X-PI))-3*SIN(3*(X-PI)) X IN (0,2PI)

I	A(I)	B(I)	FREQ(I)	RMS(I)
0	1.0000000+00			3.8837267+00
1	-2.0000000+00	-2.9999999+00	1.5915494-01	3.2093613+00
2	-2.4635268-08	2.9802322-08	3.1830989-01	3.5881750+00
3	-1.9999999+00	-2.9999999+00	4.7746483-01	2.0412416+00
4	-4.9670537-08	-7.9472859-08	6.3661977-01	2.5000002+00
5	6.9538751-08	2.185036-07	7.9577471-01	3.5355342+00
6	0.0000000	-1.2113140-07	9.5492966-01	-0.0000000
(TMAX-TMIN)/2		3.1415927+00	(TMAX+TMIN)/2	3.1415927+00

FITTED FUNCTION.	3	TERMS USED.
T		F(T)
0.6000000		5.0000002+00
1.5707963	-01	6.5886645+00
3.1415926	-01	7.4317852+00
4.7123889	-01	7.4199183+00
6.2831852	-01	6.6165256+00
7.8539815	-01	5.2426413+00
9.4247778	-01	3.6275606+00
1.0995574	+00	2.1363216+00
1.2566370	+00	1.0898146+00
1.4137166	+00	6.9493344-01
1.5707963	+00	9.9999925-01
1.7278759	+00	1.8851562+00
1.8849555	+00	3.0898120+00
2.0420352	+00	4.2711103+00
2.1991148	+00	5.0806435+00
2.3561943	+00	5.2426407+00
2.5132740	+00	4.6165267+00
2.6703536	+00	3.2301571+00
2.8274332	+00	1.2764217+00
2.9845128	+00	-9.2611134-01
3.1415924	+00	-2.9999969+00
3.2986720	+00	-4.5886624+00
3.4557516	+00	-5.4317845+00
3.6128312	+00	-5.4199190+00
3.7699109	+00	-4.6165270+00
3.9269904	+00	-3.2426433+00
4.0840700	+00	-1.6275621+00
4.2411496	+00	-1.3632287-01
4.3982292	+00	9.1018479-01
4.5553088	+00	1.3050666+00
4.7123884	+00	1.0000013+00
4.8694679	+00	1.1484483-01
5.0265475	+00	-1.0898109+00
5.1836271	+00	-2.2711094+00
5.3407066	+00	-3.0806430+00
5.4977863	+00	-3.2426411+00
5.6548658	+00	-2.6165280+00
5.8119453	+00	-1.2301595+00
5.9690250	+00	7.2357464-01
6.1261045	+00	2.9261073+00
6.2831841	+00	4.9999934+00

```

PLOT OF FITTED FUNCTION      Y=1-2*COS(X-PI)-3*SIN(X-PI)=2*COS(3*(X-PI))-3*SIN(3*(X-PI)) X IN (0,2PI)
DEPENDENT VARIABLE RANGE ON ZERO TO ONE PLOT IS -5.762+00 TO 7.432+00. PLOT ERROR IS 1.25 PERCENT OF FULL SCALE.
1.000--* A A
* * *
* * A
* *
* *
*.875--*
* * A
* *
* *
*.750--*
* * A
* *
* *
*.625--*
* * A
* *
* *
*.500--*
* * A
* *
* *
*.375--*
* * A
* *
* *
*.250--*
* * A
* *
* *
*.125--*
* * A
* *
* *
0.00--*
* * A
* *
INDEPENDENT VARIABLE RANGE ON ZERO TO ONE PLOT IS 0.000 TO 6.346+00. PLOT ERROR IS .50 PERCENT OF FULL SCALE.
0.0 1.0

```

SPECTRAL DISTRIBUTION. 6 TERMS USED IN THIS APPROXIMATION.

FREQ	F-REAL	F-IMAG	POWER
0.000000	6.2831853+00	0.000000	6.2831852+00
2.500000-02	6.8440878+00	7.0632619-01	6.8804385+00
4.999999-02	8.2637697+00	6.2880135-01	8.2876583+00
7.499998-02	9.8281962+00	-7.1590793-01	9.8542358+00
9.999998-02	1.0613243+01	-3.2646921+00	1.1104015+01
1.250000-01	9.8790597+00	-6.3350828+00	1.1735804+01
1.500000-01	7.4785040+00	-8.8499703+00	1.1586630+01
1.750000-01	4.0286035+00	-9.8381974+00	1.0631076+01
2.000000-01	6.5989494-01	-8.9606433+00	8.9849090+00
2.249999-01	-1.5735602+00	-6.7071629+00	6.8892761+00
2.499999-01	-2.2568113+00	-4.0828777+00	4.6650924+00
2.749999-01	-1.7443133+00	-1.9719798+00	2.6327425+00
2.999999-01	-7.5057366-01	-6.4455957-01	9.8935224-01
3.249999-01	2.7524080-01	2.0702855-01	3.4441010-01
3.499998-01	1.4632150+00	9.3579578-01	1.7368684+00
3.749998-01	3.2312457+00	1.3816389+00	3.5142388+00
3.999998-01	5.6136607+00	8.2541223-01	5.6740190+00
4.249997-01	7.8149094+00	-1.4102753+00	7.9411387+00
4.499997-01	8.4802473+00	-5.1567702+00	9.9250628+00
4.749996-01	6.6087480+00	-9.0979027+00	1.1244882+01
4.999996-01	2.4527595+00	-1.1358702+01	1.1620505+01
5.249995-01	-2.3494683+00	-1.0677479+01	1.0932912+01
5.499994-01	-5.6406906+00	-7.3293801+00	9.2486323+00
5.749994-01	-6.0756719+00	-3.0699750+00	6.8072414+00
5.999993-01	-3.9746401+00	-5.9187457-02	3.9750807+00
6.249993-01	-1.0619483+00	5.0367536-01	1.1753395+00
6.499993-01	7.3621512-01	-9.3643236-01	1.1911835+00
6.749992-01	5.6378908-01	-2.7617931+00	2.8187513+00
6.999991-01	-9.3365489-01	-3.4235936+00	3.5486200+00
7.249990-01	-2.2766253+00	-2.5126258+00	3.3906210+00
7.499990-01	-2.3478055+00	-8.9102390-01	2.5111978+00
7.749989-01	-1.1842812+00	1.3190236-01	1.1916041+00
7.999988-01	1.9503134-01	-1.2947802-01	2.3409780-01
8.249987-01	7.0457445-01	-1.2648547+00	1.4478545+00
8.499986-01	6.7492634-02	-2.2079798+00	2.2090110+00
8.749985-01	-1.0383094+00	-2.1595288+00	2.3961743+00
8.999984-01	-1.6182967+00	-1.2121350+00	2.020266+00
9.249983-01	-1.2034902+00	-1.9719604-01	1.2195389+00
9.499982-01	-1.9378792-01	6.3982561-02	2.0407726-01
9.749982-01	5.4239962-01	-5.5949064-01	7.7924778-01
9.999981-01	4.2807342-01	-1.4491028+00	1.5110082+00

[illegible]

INPUT DATA Y=1-2*COS(X-PI)-3*SIN(X-PI)-2*COS(3*(X-PI))-3*SIN(3*(X-PI)) X IN (0,2PI)

OMEGA	REAL	IMAGINARY
0.000000	1.000000+00	0.000000
1.570796-01	1.089270+00	1.124153-01
3.141593-01	1.315219+00	1.000768-01
4.712389-01	1.564205+00	-1.139403-01
6.283185-01	1.689150+00	-5.195919-01
7.853981-01	1.572301+00	-1.008259+00
9.424778-01	1.190240+00	-1.408516+00
1.099557+00	6.411721-01	-1.565797+00
1.256637+00	1.050255-01	-1.426130+00
1.413716+00	-2.504399-01	-1.067478+00
1.570796+00	-3.591827-01	-6.498101-01
1.727875+00	-2.776161-01	-3.138504-01
1.884955+00	-1.194575-01	-1.025849-01
2.042035+00	4.380589-02	3.294960-02
2.199114+00	2.328779-01	1.489365-01
2.356194+00	5.142687-01	2.198946-01
2.513274+00	8.934418-01	1.313684-01
2.670353+00	1.243781+00	-2.244523-01
2.827433+00	1.349673+00	-8.207254-01
2.984512+00	1.051814+00	-1.447976+00
3.141592+00	3.903687-01	-1.807793+00
3.298672+00	-3.739295-01	-1.699373+00
3.455751+00	-8.977438-01	-1.166507+00
3.612831+00	-9.669731-01	-4.886017-01
3.769910+00	-6.325836-01	-9.419999-03
3.926990+00	-1.690143-01	8.016239-02
4.084070+00	1.171723-01	-1.490379-01
4.241149+00	8.972979-02	-4.395530-01
4.398292+00	-1.485958-01	-5.448819-01
4.555308+00	-3.623362-01	-3.998969-01
4.712388+00	-3.736649-01	-1.418109-01
4.869467+00	-1.884842-01	2.099290-02
5.026547+00	3.104020-02	-2.060710-02
5.183627+00	1.121365-01	-2.013079-01
5.340706+00	1.074180-02	-3.514109-01
5.497786+00	-1.652521-01	-3.436997-01
5.654865+00	-2.575599-01	-1.929460-01
5.811945+00	-1.915414-01	-3.138480-02
5.969025+00	-3.084230-02	1.018310-02
6.126104+00	8.632549-02	-8.904570-02
6.283184+00	6.812999-02	-2.306319-01
SCALE FACTOR FOR INPUT DATA		6.283185+00

$Y=1-2*\cos(X-PI)-3*\sin(X-PI)-2*\cos(3*(X-PI))-3*\sin(3*(X-PI))$ X IN (0,2PI)

I	A(I)	B(I)	FREQ(I)
0	1.0000001+00		
1	-1.9999998+00	-2.9999998+00	1.5915494-01
2	-6.1467290-08	7.8580342-08	3.1830989-01
3	-2.0000001+00	-2.9999999+00	4.7746483-01
4	1.3054612-07	5.4976940-08	6.3661977-01
5	6.0086815-08	2.1433591-07	7.9577471-01
6	4.1496802-08	-1.9194536-08	9.5492966-01
	(TMAX-TMIN)/2	3.1415927+00	(TMAX+TMIN)/2 3.1415927+00

FITTED FUNCTION.	6	TERMS USED.
T		F(T)
0.000000		5.0000002+00
1.5707963-01		6.5886644+00
3.1415926-01		7.4317850+00
4.7123889-01		7.4199180+00
6.2831852-01		6.6165255+00
7.8539815-01		5.2426412+00
9.4247778-01		3.6275605+00
1.0995574+00		2.1363216+00
1.2566370+00		1.0898143+00
1.4137166+00		6.9493318-01
1.5707963+00		9.9999920-01
1.7278759+00		1.8851562+00
1.8849555+00		3.0898122+00
2.0420352+00		4.2711106+00
2.1991148+00		5.0806437+00
2.3561943+00		5.2426410+00
2.5132740+00		4.6165267+00
2.6703536+00		3.2301569+00
2.8274332+00		1.2764216+00
2.9845128+00		-9.2611125-01
3.1415924+00		-2.9999966+00
3.2986720+00		-4.5886621+00
3.4557516+00		-5.4317842+00
3.6128312+00		-5.4199186+00
3.7699109+00		-4.6165267+00
3.9269904+00		-3.2426432+00
4.0840700+00		-1.6275620+00
4.2411496+00		-1.3632260-01
4.3982292+00		9.1018524-01
4.5553086+00		1.3050671+00
4.7123884+00		1.0000018+00
4.8694679+00		1.1484521-01
5.0265475+00		-1.0898109+00
5.1836271+00		-2.2711095+00
5.3407066+00		-3.0806433+00
5.4977863+00		-3.2426415+00
5.6548658+00		-2.6165283+00
5.8119453+00		-1.2301595+00
5.9690250+00		7.2357478-01
6.1261045+00		2.9261076+00
6.2831841+00		4.9999934+00

DISTRIBUTION

No. of
Copies

OFFSITE

1	<u>AEC Chicago Patent Group</u> G. H. Lee
242	<u>AEC Division of Technical Information Extension</u>

ONSITE-HANFORD

1	<u>AEC Chicago Patent Group</u> R. K. Sharp (Richland)
2	<u>AEC RDT Site Representative</u> P. G. Holsted
1	<u>AEC Richland Operations Office</u> C. L. Robinson
3	<u>Battelle Memorial Institute</u>
47	<u>Battelle-Northwest</u> L. C. Beck F. L. Becker T. M. Beetle N. E. Dixon J. L. Jaech W. R. McSpadden (2) R. N. Ord C. A. Oster (23) H. N. Pedersen (4) R. L. Richardson W. G. N. Slinn Technical Information (5) Technical Publications (2)