TECHNICAL NOTE

COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

By

HIROSHI ISHIKAWA*

1. General Remarks

In connection with the paper entitled "Digital Simulation of Nonstationary Random Processes and Its Applications," which has appeared in the Kagawa University Economic Review, Vol. 52, No. 3・4 (1979—10), pp. 308~373, the present computer programs have been developed and are provided here for reader's reference to his better understanding of the paper and/or for his practical facility in the application of the data-based nonstationary random process models proposed in the paper.

These programs have been coded in FORTRAN language to meet the requirement of a particular Hewlett-Packard minicomputer (Model 2100A) of memory size 32 KB operated by the RTE-II system with an additional Tektronix terminal control system for graphic display purpose. However, the current RTE-II system for a HP minicomputer is deemed fo be so well designed, with special attention to the compatibility requirement, that it is undoubtedly believed a slight modification does suffice the feasibility of the present programs to any other type of com-

* Ph. D., Associate professor, Junior College of Commerce, Kagawa University, 2—1 Saiwai-cho, Takamatsu-shi 760 Japan.

puter system. One example for this follows: For plotting purpose in any of the present programs is used SUBROUTINE STUKO which, as mentioned earlier, utilizes a Tektronix terminal control system. Therefore, if, unfortunately, a computer system to be used has no graphic display option, then an appropriate action to be taken by the user will be either simply to take out such statements as are related to SUBROUTINE STUKO, or to replace them by other effective alternatives.

EXEC CALL's as well as other specific symbols germane to the present system are to be discussed in some detail whenever they are encountered.

2. Brief Explanation of the Programs Developed

The present programs consist of the following main or subroutine programs to each of which is given a brief explanation of the contexture. Readers are urged to refer to the aforementioned paper for symbolic notations in what follows.

2. 1 PROGRAM SIM 10 (listed on page 438)

This program gets such basic statistics of the given time history $x_0(t)$ of duration T_0 (sec) as temporal mean μ, maximum and minimum values within the duration, x_{0max} and x_{0min}, Fourier (amplitude) spectrum $|X_0(\omega)|$ and phase angle $\zeta_0(\omega)$ of the Fourier transform $X_0(\omega)$ of $x_0(t)$ as well as its Hilbert transform $\hat{x}_0(t)$. The following relations are intrinsic in this program:

$$\mu=\frac{1}{T_0}\int_0^{T_0} x_0(t)dt \tag{1}$$

$$x_{0max}=\max\,[x_0(t)] \tag{2}$$

$$x_{0min}=\min\,[x_0(t)] \tag{3}$$

$$x_0(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} X_0(\omega)\exp(i\omega t)\,d\omega \tag{4}$$

$$\begin{aligned} X_0(\omega)&=\int_{-\infty}^{\infty} x_0(t)\exp(-i\omega t)\,dt \\ &=|X_0(\omega)|\exp\{i\zeta_0(\omega)\} \end{aligned} \tag{5}$$

$$\begin{aligned} \hat{x}_0(t)&=\frac{1}{\pi}\int_0^{\infty}|X_0(\omega)|\sin\{\omega t+\zeta_0(\omega)\}\,d\omega \\ &=\frac{1}{\pi}\int_{-\infty}^{\infty} x_0(\tau)/(t-\tau)\,d\tau \\ &=x_0(t)*[1/(\pi t)] \end{aligned} \tag{6}$$

where an asterisk "*" indicates a convolution integral.

This program also plots, with the aid of SUBROUTINE STUKO, the original time history, its Fourier spectrum, phase angle and Hilbert transform.

For the computation of the Fourier transform is used FFT (Fast Fourier Transform) technique whose details are implicitly expressed in SUBROUTINE FFT discussed later. Hence, if the time series data are those sampled with inequality of time increment, then SUBROUTINE INTPL is applied to interpolate given data so as to produce such time series data as are placed equidistantly.

Required subroutines are STUKO, STATC, SPECT, PHASE, HLBRT, FFT and INTPL. Details of these subroutines are described in the corresponding subsections of this section.

2.2 **PROGRAM SIM 20** (listed on page 441)

This program generates and plots sample functions of the data-based nonstationary random process of the first kind defined as

$$x(t)=x_0(t)\cos\phi-\hat{x}_0(t)\sin\phi \qquad (7)$$

where $x(t)$=simulated sample function of the process of the first kind,

$x_0(t)$=the original time history,

$\hat{x}_0(t)$=the Hilbert transform ot $x_0(t)$ given in Eq.(6), and

ϕ=a realization of random phase angle Φ whose distribution can be selected, according to the control parameter ICON, as follows:

If ICON=3, Φ is Gaussian.

If ICON=2, Φ is uniform between $-\pi/2$ and $\pi/2$.

Otherwise, Φ is uniform between $-\pi$ and π.

There appear, in this program, specific symbols ISSW(14) and ISSW(15), the meaning of which is the following: The symbol ISSW followed by parentheses and a positive integer J (0 to 15) inbetween represents the content of the switch register designated by the numeral J in such a way that ISSW(J) holds a negative integer value when the J-th switch register button is set on and a positive or zero otherwise. As a result, these specific symbols, if used in IF statements, will give those functions to be controlled externally by the user.

Parameters in EXEC CALL, EXEC (I, 1 JKB, Y, N), have the meaning:

I = READ/WRITE control parameter such that it means READ when $I=1$, and WRITE when $I=2$, from and onto the peripheral assigned by the octal number "JK", respectively.

JK = octal representation of the select code of READ/WRITE peripheral. For example, $JK=10$ means that READ/WRITE peripheral has select code of 10 in octal, that is, 8 in decimal, which corresponds to the mag-tape unit in the present system.

B as in JKB = data format in binary form.

Y = READ/WRITE buffer area, and

N = buffer length in terms of one word length. Hence, $N=N_Y$ if Y is an integer array of size N_Y, and $N=2N_Y$ if Y a real array.

Required subroutines are STUKO, RANDU, SPECT, FFT, INTPL JNORM, and HLBRT, where RANDU is a uniform random number (0 to unity) generating subroutine.

2. 3 PROGRAM SIM 30 (listed on page 445)

This program gets and plots such ensemble and/or temporal statistics of the simulated data-based nonstationary random process as mean, maximum and minimum values, standard deviation and probability density function. Digitized values of sample functions of the simulated process and related parameters necessary for this statistical computation should be read out exactly in the same sequence and format from the peripheral on which they have been stored through EXEC CALL's in PROGRAM SIM 20.

Required subroutines are STUKO, STATC, HLBRT and FFT.

2. 4 PROGRAM SIM 40 (listed on page 450)

This program generates and plots sample functions of the data-based bivariate nonstationary random process of the first kind with random phase angles being jointly Gaussian. More specifically, data-based bivariate processes are simulated as

$$x_i(t)=x_{0i}(t)\cos\Phi_i-\hat{x}_{0i}(t)\sin\Phi_i \quad (i=1,\ 2) \qquad (8)$$

where $x_i(t)$ = simulated component process

$x_{0i}(t)$ = original record of each component

$\hat{x}_{0i}(t)$ = Hilbert transform of $x_{0i}(t)$

Φ_i's = random phase angles which are Gaussian with the joint density function

$$f_{\Phi_1\Phi_2}(\phi_1,\phi_2)=\frac{1}{2\pi\sqrt{1-\rho^2}\sigma_1\sigma_2}\exp\left[-\frac{1}{2(1-\rho^2)}\left\{\left(\frac{\phi_1-\mu_1}{\sigma_1}\right)^2 -2\rho\left(\frac{\phi_1-\mu_1}{\sigma_1}\right)\left(\frac{\phi_2-\mu_2}{\sigma_2}\right)+\left(\frac{\phi_2-\mu_2}{\sigma_2}\right)^2\right\}\right] \quad (9)$$

in which μ_i and σ_i = mean and standard deviation of Φ_i, respectively, and ρ = coefficient of correlation between Φ_1 and Φ_2.

This program also computes and plots Fourier spectra of the original record and simulated sample functions of each component process.

Required subroutines are STUKO, SPECT, RANDU, INTPL, JNORM, FFT and HLBRT.

2. 5 PROGRAM SIM 50 (listed on page 453)

This program computes and plots crosscorrelation function $R_{x_1x_2}(t_1, t_2)$ between component processes $x_1(t)$ and $x_2(t)$ of the data-based bivariate nonstationary random process of the first kind simulated with random phase angles Φ_1 and Φ_2 being jointly Gaussian as in Eq. (9).

Specific representation of $R_{x_1x_2}(t_1, t_2)$ is as follows:

$$\begin{aligned} R_{x_1x_2}(t_1, t_2) &= x_{01}(t_1)\,x_{02}(t_2)\cdot E\{\cos\Phi_1\cos\Phi_2\} \\ &+\hat{x}_{01}(t_1)\hat{x}_{02}(t_2)\cdot E\{\sin\Phi_1\sin\Phi_2\} \\ &-\hat{x}_{01}(t_1)x_{02}(t_2)\cdot E\{\sin\Phi_1\cos\Phi_2\} \\ &-x_{01}(t_1)\hat{x}_{02}(t_2)\cdot E\{\cos\Phi_1\sin\Phi_2\} \end{aligned} \quad (10)$$

where $x_{0i}(t)$ and $\hat{x}_{0i}(t)(i=1, 2)$ are the original record and its Hilbert transform, respectively.

It is noted in the multivariate simulation that, for those Φ_i's distributed uniformly, intermediate mode of dependence are very much cumbersome to consider since the joint density functions which involve Φ_i's and produce uniform marginal distributions for Φ_i's appear to be extremely difficult to obtain. However, when all Φ_i's are Gaussian, arbitrary degrees of dependence (including those cases of complete independence and total dependence) can easily be introduced through the well known Gaussian joint density functions involving Φ_i's. This fact is one definite advantage the Gaussian assumption can enjoy over the assumption of uniform distribution; hence, the use of random phase angles of jointly Gaussian distribution in this program.

Required subroutines are STUKO, RANDU, HLBRT, FFT, INTPL and JNORM.

2.6 **PROGRAM SIM 60** (listed on page 456)

This program generates and plots sample functions of the data-based non-stationary random process of the second kind with random phase angle being distributed uniformly between $-\pi/2$ and $\pi/2$.

Generation of a sample function $x(t)$ of the process of the second kind is specified as:

$$x(t)=x_0(t)\cos\phi-\tilde{x}_0(t)\sin\phi \tag{11}$$

where

$x(t)=$ a sample function of the process of the second kind,

$x_0(t)=$ roiginal record of duration T_0

ϕ = a realization of random phase angle Φ

(distributed uniformly between $-\pi/2$ and $\pi/2$ in this program), and

$$\tilde{x}_0(t)=\hat{z}_0(t)v(t) \tag{12}$$

in which

$v(t)=$ a temporal filter of the form

$$v(t)=U(t)-U(t-T_0) \tag{13}$$

with $U(t)$ indicating the Heavyside unit step function, and

$z_0(t)=$ the symmetric-periodic extension of the original record $x_0(t)$

$$=\lim\sum_{k=-N}^{N}y_0(t-2kT_0) \tag{14}$$

with $y_0(t)$ indicating the symmetric-extension of $x_0(t)$ given as

$$y_0(t)=x_0(t)+x_0(-t) \tag{15}$$

Fourier spectra of the original record and of generated sample functions as well as other statistical quantities are also computed and plotted.

This program includes in itself the following subroutines: AREA, SAMPL, FFT 0, ASYME and PHS 11.

Besides those subroutines stated above, this program also requires subroutines STUKO, RANDU, SPECT, ARNGE, HLBRT, INTPL and FFT.

2.7 SUBROUTINE INTPL (listed on page 467)

This subroutine program interpolates time series data with inequality of time increment so as to produce a series of data with the same time increment of a given constant value.

2.8 SUBROUTINE FFT (listed on page 468)

This subroutine takes one-dimensional Fourier transform $X(\omega)$ of the given time history $x(t)$ or inverse Fourier transform $x(t)$ of $X(\omega)$ with the aid of FFT (Fast Fourier Transform) technique.

The following two equations form a Fourier transform pair:

$$\left.\begin{aligned} X(\omega) &= \int_{-\infty}^{\infty} x(t) \exp(-i\omega t)\, dt \\ x(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega) \exp(i\omega t)\, d\omega \end{aligned}\right\} \tag{16}$$

It should be noted that FFT computation requires that array size, or the number of data, be a power of 2. Hence, if the number of data is given otherwise, an appropriate number of trailing zeroes must be added to meet the requirement.

Comment statements in the subroutine program help for better understanding of FFT algorithm.

2.9 SUBROUTINE PHASE (listed on page 470)

This subroutine computes phase angle $\zeta_0(\omega)$ ranging between $-\pi$ and π of the Fourier transform $X_0(\omega)$ of the original time history $x_0(t)$.

The phase angle $\zeta_0(\omega)$ is given, if $\mathrm{Re}[X_0(\omega)]\neq 0$, by

$$\zeta_0(\omega) = \arctan\{\mathrm{Im}[X_0(\omega)]/\mathrm{Re}[X_0(\omega)]\} \tag{17}$$

while, if $\mathrm{Re}[X_0(\omega)]=0$, it is given by

$$\zeta_0(\omega) = \pm\pi/2 \tag{18}$$

where $\mathrm{Re}[X_0(\omega)]$ represents real part of $X_0(\omega)$ and $\mathrm{Im}[X_0(\omega)]$ imaginary part.

2.10 SUBROUTINE SPECT (listed on page 471)

This subroutine computes power spectrum, Fourier (amplitude) spectrum or absolute value of Fourier complex coefficient of the given time history according to the assigned value of the control parameter MO as follows:

If MO=10, power spectrum in terms of frequency f (Hz),

If MO=11, power spectrum in terms of circular frequency ω (rad./sec),

If MO=1, Fourier (amplitude) spectrum, and

If MO=2, absolute value of complex Fourier coefficient.

Reguired subroutine is FFT.

2.11 **SUBROUTINE STATC** (listed on page 473)

This is a subroutine to compute such basic statistics of the given data set as overall (temporal or ensemble) mean, unbiased variance, standard deviation, and maximum and minimum values.

2.12 **SUBROUTINE HLBRT** (listed on page 474)

This subroutine program takes the Hilbert transform $\hat{x}_0(t)$ of the given time history $x_0(t)$ by utilizing FFT technique mentioned earlier.

Originally the Hilbert transform of $x_0(t)$ is defined either in the time domain or in the freguency domain as

$$\hat{x}_0(t)=\frac{1}{\pi}\int_{-\infty}^{\infty} x_0(\tau)/(t-\tau)\, d\tau \tag{19}$$

$$=\frac{1}{\pi}\int_{0}^{\infty} | X_0(\omega) | \sin \{\omega t+\zeta_0(\omega)\}\, d\omega \tag{20}$$

where $X_0(\omega)$ and $\zeta_0(\omega)$ are the Fourier transform of $x_0(t)$ and its phase angle. However, the direct performance of the integrals given in the above two equations is rather time-consuming. Therefore an effective practical method has to be developed. To this end, we first define the functions $Y_0(\omega)$ and $Z_0(\omega)$ as

$$Y_0(\omega)= | X_0(\omega) | \cdot \mathrm{sgn}\ (\omega) \tag{21}$$

$$\begin{aligned} Z_0(\omega)&=X_0(\omega) \cdot \mathrm{sgn}\ (\omega) \\ &= | X_0(\omega) | \cdot \exp\ \{i\zeta_0(\omega)\} \cdot \mathrm{sgn}\ (\omega) \\ &=Y_0(\omega) \cdot \exp\ \{i\zeta_0(\omega)\} \end{aligned} \tag{22}$$

where $\mathrm{sgn}(\omega)$ is a sign function of ω. $Y_0(\omega)$ thus defined is obviously an odd function since $| X_0(\omega) |$ is even and $\mathrm{sgn}(\omega)$ is odd. Then, we try to reduce Eq.

(20) into the form such that the FFT technique is applicable:

$$\hat{x}_0(t)=\frac{1}{\pi}\int_0^{\infty} | X_0(\omega) | \sin \{\omega t+\zeta_0(\omega)\} d\omega$$

$$=\frac{1}{2\pi}\int_{-\infty}^{\infty} Y(\omega) \sin \{\omega t+\zeta_0(\omega)\} d\omega \qquad (23)$$

where the fact has been used that sin $\{\omega t+\zeta_0(\omega)\}$ is an odd function of ω. Further, since $Y(\omega)$ is odd and cos $\{\omega t+\zeta_0(\omega)\}$ is even with respect to ω, Eq. (23) can be written as

$$\hat{x}_0(t)=\frac{1}{2\pi i}\int_{-\infty}^{\infty} Y_0(\omega) \exp [i \{\omega t+\zeta_0(\omega)\}] d\omega$$

$$=\frac{1}{2\pi i}\int_{-\infty}^{\infty} Y_0(\omega) \cdot \exp \{i\zeta_0(\omega)\} \exp (i\omega t) d\omega$$

$$=\frac{1}{2\pi i}\int_{-\infty}^{\infty} Z_0(\omega) \exp (i\omega t) d\omega \qquad (24)$$

Eq. (24) now posseses the form such that FFT can be directly applicable. Hence, for the ready computation of the Hilbert transform $\hat{x}_0(t)$ of $x_0(t)$, we first take the Fourier transform $X_0(\omega)$ of $x_0(t)$ with the aid of FFT technique, then form a function $Z_0(\omega)$ as in Eq. (22), take the inverse Fourier transform of $Z_0(\omega)$ and finally divide the result by the imaginary unit i.

2.13 **SUBROUTINE JNORM** (listed on page 475)

This subroutine generates either univariate Gaussian random number or jointly Gaussian bivariate with arbitrary degrees of dependence including those cases of complete independence and total dependence. Some comments on the bivariate case are also found in the subsection of PROGRAM SIM40 discussed earlier.

2.14 **SUBROUTINE ARNGE** (listed on page 477)

This subroutine arranges given array data y_i ($i=1, 2, \cdots, N$) in a symmetrical fashion so as to produce mirror image either around the origin or around a folding point $\left(i=\frac{N}{2}+1\right)$ for FFT computation. Refer also to the comment statements in the subroutine for detailed discussion.

2.15 **SUBROUTINE STUKO** (listed on page 479)

SUBROUTINE YOKO (listed on page 481)

PROGRAM CHIE (listed on page 487)

The above three are user supply plotting main and subroutine programs for the current Tektronix terminal control system with plotting library subroutines (discussed in the next subsection) in the system. When SUBROUTINE STUKO is to be used, the other two have to be relocated simultaneously. Details are provided through comment statements in each program or subroutine.

At least one week has been consumed with plenty of cumbersome tasks to produce these three of the present forms. Hence, for the celebration of their construction, they are named as follows: STUKO after the author's better half, CHIE and YOKO after his daughters.

Indeed, almost all of the figures in the paper referred to in the preceding section have been drawn automatically through the computer system with graphic dispay option involving these programs, which indicates their extremely wide applicability in the practical drawing.

2.16 **Plotting Library Subroutines** (listed on page 489)

Program listings on page 116 and after provide plotting library subroutines in tern named as follows:

SUBROUTINE LVLCH	SUBROUTINE MOVEA
SUBROUTINE PARCL	SUBROUTINE DRAWA
SUBROUTINE DWIND	SUBROUTINE WINCO
SUBROUTINE V2ST	SUBROUTINE REVCO
SUBROUTINE CLIPT	SUBROUTINE RESET
SUBROUTINE NWPAG	SUBROUTINE IOWAI
SUBROUTINE XYCNT	SUBROUTINE ALFMD
SUBROUTINE RESCL	SUBROUTINE MOVAB
SUBROUTINE VECMD	SUBROUTINE PLCHR
SUBROUTINE INITT	SUBROUTINE FINIT
SUBROUTINE TOUTS	SUBROUTINE BUFFK
SUBROUTINE ADOUT	SUBROUTINE TWIND
SUBROUTINE PCLIP	

3 Computer Program Listings

The whole computer programs developed for the simulation of data-based nonstationary random processes are listed at the end of this paper in such a sequential order as explained in the preceding section. In each program listing an effort has been made to place an effective number of comment statements for better understanding of the program. Together with the brief explanation given in the preceding section, they are believed to help readers understand what is going on there.

Acknowledgement

The author wishes to express his deep gratitude to Renwick Professor Masanobu Shinozuka, Department of Civil Engineering and Engineering Mechanics, Colnmbia University, New York, USA, for his technical advice and encouragement during the course of this research. Thanks are also due to Dr. Hidehiko Mitsuma, National Space Development Agency, Tokyo, Japan, and Dr. Dali Li, Research Associate of Columbia University, for their valuable suggestions and help offered in this program development.

REFERENCES

The following references are of crucial importance to understand the notion of the data-based nonstationary random processes and their statistical properties as well as those of the stationary random processes:

(1) Shinozuka, M., Ishikawa, H., and Mitsuma, H., "Data-Based Nonstationary Random Processes," Proc. of the ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Tucson, Arizona, USA, (1979—1), pp. 39～43.

(2) Ishikawa, H., and Kimura, H., "Application of Stochastic Process Theory to the Fatigue Life Estimation under Random Loading," The Kagawa University Economic Review, Vol. 52, No. 1・2 (1979—6), pp. 96～183.

(3) Ishikawa, H., Mitsuma, H., and Shihozuka, M., "Digital Simulation of Nonstationary Random Processes and Its Applications," The Kagawa University Economic Review, Vol. 52, No. 3・4 (1979—10), pp. 308～373.

```
HSIM10 T=00004 IS ON CR00002 USING 00027 BLKS R=0178

FTN4
C      ------------------------------(MAIN PROGRAM SIM10)------------
       PROGRAM SIM10
C....  THIS PROGRAM GETS BASIC STATISTICS OF TIME HISTORY XO(T)
C      WITH DURATION TT (SEC) SUCH AS TEMPORAL MEAN, MAXIMUM AND
C      MINIMUM VALUES DURING DURATION TT, FOURIER AMPLITUDE SPECTRUM
C      AND PHASE ANGLE OF THE FOURIER TRANSFORM OF XO(T) AS WELL AS
C      ITS HILBERT TRANSFORM.
C      THIS PROGRAM ALSO PLOTS TIME HISTORY XO(T), ITS FOURIER
C      AMPLITUDE SPECTRUM, PHASE ANGLE AND ITS HILBERT TRANSFORM.
C
C....  DESCRIPTION OF MAIN PARAMETERS:
C        XO(I)      -- EQUIDISTANT TIME SERIES DATA TO BE ANALYZED.
C                      WHEN THE ORIGINAL DATA ARE NOT EQUIDISTANT,
C                      PROGRAM AUTOMATICALLY INTERPOLATES TO GET
C                      EQUIDISTANT DATA.
C        T(I)       -- TIME ARRAY (SEC).
C        W(I)       -- FREQUENCY ARRAY (RADIAN/SEC).
C        PSD(I)     -- FOURIER AMPLITUDE SPECTRUM.
C        PHS(I)     -- PHASE ANGLE BETWEEN -PI AND PI.
C        XHIL(I)    -- HILBERT TRANSFORM OF XO(T).
C....  REQUIRED SUBROUTINES: STUKO,STATC,SPECT,PHASE,HLBRT,FFT,INTPL
C....  REMARKS:
C        FOR PLOTTING PURPOSE IS USED SUBROUTINE STUKO WHICH
C        UTILIZES TEKTRONIXS TERMINAL CONTROL SYSTEM IN HP
C        MINI-COMPUTER. THEREFORE SLIGHT MODIFICATION IS NEEDED
C        FOR IBM COMPUTER.
C
       DIMENSION A(1024),XO(513),T(513),W(513),PSD(513),PHS(513)
       DIMENSION NAME(35),XHIL(513)
C
C....  TO ASSIGN LOGICAL UNIT NUMBER OF INPUT/OUTPUT PERIPHERALS.
C        LOGICAL UNIT NO. = 1 -- TERMINAL
C        LOGICAL UNIT NO. = 6 -- LINE PRINTER
C        LOGICAL UNIT NO. = 8 -- MAG-TAPE UNIT
       LU=1
       IIN=8
       IOUT=6
C
C....  TO SPECIFY TIME SERIES DATA TO BE ANALYZED (UP TO 70 CHAR.).
       WRITE(LU,600)
   600 FORMAT(" INPUT -- PROBLEM IDENTIFICATION CODE")
       READ(LU,550)(NAME(I),I=1,35)
   550 FORMAT(35A2)
       WRITE(IOUT,660)(NAME(I),I=1,35)
   660 FORMAT(1H1////5X,35A2)
C
C....  TO ASSIGN NECESSARY PARAMETER VALUES.
C        NORG   -- NUMBER OF ORIGINAL DATA.
C        NDATA  -- NUMBER OF INTERPOLATED DATA.
C                  FOR FFT COMPUTATION, NDATA MUST BE A POWER OF 2.
       WRITE(LU,601)
   601 FORMAT(" INPUT --- NORG, NDATA")
       READ(LU,*) NORG,NDATA
C
C....  TO READ IN THE ORIGINAL TIME SERIES FROM MAG-TAPE.
       REWIND IIN
       READ(IIN,300)(A(I),W(I),I=1,NORG)
   300 FORMAT(6F10.3)
```

```
      REWIND IIN
C
C.... TO COMPUTE AND PRINT OUT APPROPRIATE QUANTITIES.
      NFOLD=NDATA/2+1
      NDATA1=NDATA+1
      TT=A(NORG)-A(1)
      DT=TT/FLOAT(NDATA)
      PI1=3.1415926
      PI2=2.0*PI1
      DF=1.0/TT
      DW=PI2*DF
      FOLDF=DF*FLOAT(NFOLD)
      FOLDW=PI2*FOLDF
      WRITE(IOUT,630)NORG,NDATA,NFOLD,TT,DT,DF,DW,FOLDF,FOLDW
  630 FORMAT( ////
     *5X,"NUMBER OF ORIGINAL DATA -- NORG  =",I4/
     *5X,"NO. OF INTERPOLATED DATA-- NDATA =",I4/
     *5X,"NYQUIST FOLDING NUMBER   -- NFOLD =",I4/
     *5X,"DURATION (PERIOD)        -- TT    =",E12.5," (SEC)"/
     *5X,"TIME INCREMENT           -- DT    =",E12.5," (SEC)"/
     *5X,"FREQUENCY INCREMENT      -- DF    =",E12.5," (HZ)"/
     *5X,"                         -- DW    =",E12.5," (RAD./SEC)"/
     *5X,"NYQUIST FOLDING FREQ.    -- FOLDF =",E12.5," (HZ)"/
     *5X,"                         -- FOLDW =",E12.5," (RAD./SEC)"//)
C
C.... TO ADJUST TIME AXIS FOR DATA TO START FROM ZERO (SEC).
      TFST=A(1)
      DO 310 I=1,NORG
      A(I)=A(I)-TFST
  310 CONTINUE
      W(1)=0.0
C
C.... TO INTERPOLATE GIVEN DATA TO GET EQUIDISTANT ARRANGEMENT
C     FOR FFT COMPUTATION.
      CALL INTPL(A,W,NORG,T,X0,NDATA1,DT)
C
C.... TO COMPUTE AND PRINT OUT BASIC STATISTICS OF X0(T).
      CALL STATC(X0,NDATA,EX,VAR,STDV,DMAX,DMIN)
      WRITE(IOUT,610)EX,VAR,STDV,DMAX,DMIN
  610 FORMAT(//5X,"STATISTICS OF ORIGINAL RECORD"/
     *5X,"TEMPORAL MEAN      EX=",E12.5/
     *5X,"VARIANCE          VAR=",E12.5/
     *5X,"STAND. DEVIATION STDV=",E12.5/
     *5X,"MAXIMUM VALUE     MAX=",E12.5/
     *5X,"MINIMUM VALUE     MIN=",E12.5/)
C
C.... TO COMPUTE PHASE ANGLE AND FOURIER (AMPLITUDE) SPECTRUM.
      CALL PHASE(A,X0,NDATA,PHS)
      CALL SPECT(A,X0,NDATA,TT,1)
      DO 200 I=1,NDATA
      PSD(I)=A(I)
      W(I)=DW*FLOAT(I-1)
  200 CONTINUE
C
C.... TO PLOT TIME HISTORY, FOURIER (AMPLITUDE) SPECTRUM AND
C     PHASE ANGLE.
C....... FIRST, ASSIGN DEFAULT PARAMETER VALUES FOR PLOTTING.
      XMIN1=T(1)
      XMAX1=T(NDATA1)
      XMIN2=W(1)
      XMAX2=W(NFOLD)
```

```
      NX=1
      NY=1
      YMIN1=-200.
      YMIN2=-1.0E4
      YMIN3=-PI1
      YMAX1=200.
      YMAX2=1.0E4
      YMAX3=PI1
      NP=3
      MO=1
      MAXX0=950
      MINX0=150
      MINY0=150
      MAXY0=700
      JDC=22
C
C....... THEN, INPUT NECESSARY PARAMETER VALUES TO GET THE BEST
C           FIGURE.
      WRITE(LU,220)
  220 FORMAT(" INPUT -- NY,YMIN1,YMAX1,YMIN2,YMAX2,YMIN3,YMAX3"/
     *7X,"--NX,XMIN1,XMAX1,XMIN2,XMAX2,NP,MO"/
     *7X,"--JDC,MINX0,MAXX0,MINY0,MAXY0")
   10 READ(LU,*)NY,YMIN1,YMAX1,YMIN2,YMAX2,YMIN3,YMAX3,
     *NX,XMIN1,XMAX1,XMIN2,XMAX2,NP,MO,JDC,MINX0,MAXX0,MINY0,MAXY0
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 20
C
C....... FINALLY, PLOT THE FIGURE.
      CALL STUKO(T,X0,NDATA,NX,XMIN1,XMAX1,
     1NY,YMIN1,YMAX1,NP,MO,1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(W,PSD,NFOLD,NX,XMIN2,XMAX2,
     1NY,YMIN2,YMAX2,NP,MO,2,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(W,PHS,NFOLD,NX,XMIN2,XMAX2,
     1NY,YMIN3,YMAX3,NP,MO,3,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      GO TO 10
C
C.... TO COMPUTE AND PLOT HILBERT TRANSFORM OF X0(T).
   20 CALL HLBRT(A,X0,XHIL,NDATA,0)
      WRITE(LU,230)
  230 FORMAT(" INPUT--NY,YMIN1,YMAX1,NX,XMIN1,XMAX1,NP,MO"
     */"           --JDC,MINX0,MAXX0,MINY0,MAXY0")
      YMIN1=-200.
      YMAX1=200.
      NX=1
      NY=1
      NP=2
   30 READ(LU,*)NY,YMIN1,YMAX1,NX,XMIN1,XMAX1,NP,MO,JDC,
     *MINX0,MAXX0,MINY0,MAXY0
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 40
      CALL STUKO(T,X0,NDATA,NX,XMIN1,XMAX1,NY,YMIN1,YMAX1,
     *NP,MO,1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(T,XHIL,NDATA,NX,XMIN1,XMAX1,NY,YMIN1,YMAX1,
     *NP,MO,2,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      GO TO 30
   40 STOP
      END
      END$
```

```
HSIM20 T=00004 IS ON CR00002 USING 00029 BLKS R=0218

FTN4
C       ------------------------------(MAIN PROGRAM SIM20)-------------
        PROGRAM SIM20
C.... THIS PROGRAM GENERATES AND PLOTS SAMPLE FUNCTIONS OF THE
C       NONSTATIONARY RANDOM PROCESS OF THE FIRST KIND AS
C          X(T)=X0(T)*COS(RAN)-XHIL(T)*SIN(RAN)
C          WHERE X(T)   = SIMULATED SAMPLE FUNCTION OF THE FIRST KIND.
C                X0(T)  = ORIGINAL RECORD.
C                XHIL(T)= HILBERT TRANSFORM OF X0(T).
C                RAN    = RANDOM PHASE ANGLE.
C                   IF ICON=3, RAN= GAUSSIAN.
C                   IF ICON=2, RAN= UNIFORM BETWEEN -PI/2 AND PI/2.
C                   OTHERWISE, RAN= UNIFORM BETWEEN PI AND PI.
C
C.... REQURED SUBROUTINES:
C          STUKO, RANDU, SPECT, FFT, INTPL, JNORM, AND HLBRT
C.... REMARKS:
C          IF ISSW(14) IS ON, STORING PROCESS ON MAG-TAPE IS SKIPPED.
C          DATA STORING FORMAT ON MAG-TAPE:
C             NDATA,NSMPL,DT,DW,TT -- FORMAT(2I5,3E12.5)
C                                 T(I) -- CALL EXEC(2,110B,T,NDATA2)
C                                X0(I) -- CALL EXEC(2,110B,X0,NDATA2)
C                                 X(I) -- CALL EXEC(2,110B,X,NDATA2)
C          IF ISSW(15) IS ON, FOURIER SPECTRUM PLOTTING IS SKIPPED.
C          FOR PLOTTING PURPOSE IS USED SUBROUTINE STUKO WHICH
C          UTILIZES TEKTRONIX TERMINAL CONTROL SYSTEM IN HP
C          MINI-COMPUTER. THEREFORE SLIGHT MODIFICATION IS NEEDED
C          FOR IBM COMPUTER.
C
        DIMENSION A(1024),PSD(513),T(513),X0(513),XHIL(513),X(513,5)
        DIMENSION W(257)
C
C.... TO ASSIGN LOGICAL UNIT NO. OF INPUT/OUTPUT PERIPHERALS.
C          LOGICAL UNIT NO. = 1 -- TERMINAL
C          LOGICAL UNIT NO. = 6 -- LINE PRINTER
C          LOGICAL UNIT NO. = 8 -- MAG-TAPE UNIT
        LU=1
        IIN=8
        IOUT=6
C
C.... TO ASSIGN NECESSARY PARAMETER VALUES.
C          NORG  -- NUMBER OF ORIGINAL DATA.
C          NDATA -- NUMBER OF INTERPOLATED DATA. FOR FFT COMPUTATION
C                   NDATA MUST BE A POWER OF 2.
        WRITE(LU,601)
    601 FORMAT(" INPUT --- NORG, NDATA")
        READ(LU,*) NORG,NDATA
C
C.... TO READ IN THE ORIGINAL RECORD FROM MAG-TAPE.
        REWIND IIN
        READ(IIN,300)(A(I),PSD(I),I=1,NORG)
    300 FORMAT(6F10.3)
        REWIND IIN
C
C.... TO COMPUTE APPROPRIATE QUANTITIES.
        NFOLD=NDATA/2+1
        NDATA1=NDATA+1
        NDATA2=NDATA*2
        TT=A(NORG)-A(1)
```

```
      DT=TT/FLOAT(NDATA)
      PI =3.1415926
      PI2=PI*2.0
      DW=PI2/TT
      DO 310 I=1,NFOLD
  310 W(I)=DW*FLOAT(I-1)
C
C.... TO ADJUST TIME AXIS FOR DATA TO START FROM T=0 (SEC).
      TFST=A(1)
      DO 311 I=1,NORG
      A(I)=A(I)-TFST
  311 CONTINUE
      PSD(1)=0.0
C
C.... TO INTERPOLATE GIVEN DATA TO GET EQUIDISTANT ARRANGEMENT
C     FOR FFT COMPUTATION.
      CALL INTPL(A,PSD,NORG,T,X0,NDATA1,DT)
C
C.... TO COMPUTE HILBERT TRANSFORM OF ORIGINAL RECORD.
      CALL HLBRT(A,X0,XHIL,NDATA,0)
C
C.... TO COMPUTE FOURIER SPECTRUM OF ORIGINAL RECORD.
      CALL SPECT(A,X0,NDATA,TT,1)
      DO 101 I=1,NFOLD
  101 PSD(I)=A(I)
C
C.... GENERATION OF SAMPLE FUNCTIONS
C....... FIRST, SET INITIAL VALUES FOR SUBROUTINE RANDU.
      IX=12347
      IY=0
C
C....... THEN, SPECIFY SAMPLE SIZE AND NUMBER OF PLOT ON ONE FRAME.
      WRITE(LU,620)
  620 FORMAT(" HOW MANY FIGS ON ONE FRAME -- NP"/
     *" HOW MANY SAMPLES ? -- NSMPL"/
     *" NSMPL MUST BE A MULTIPLE OF NP.")
      READ(LU,*)NP,NSMPL
C
C....... TO PREPARE FOR STORING PROCESS ON MAG-TAPE.
C          IF ISSW(14) IS ON, THIS PROCESS IS SKIPPED.
      REWIND IIN
      IF(ISSW(14).LT.0) GO TO 161
      WRITE(IIN,162) NDATA,NSMPL,DT,DW,TT
  162 FORMAT(2I5,3E12.5)
      CALL EXEC(2,110B,T,NDATA2)
      CALL EXEC(2,110B,X0,NDATA2)
C
C....... TO SPECIFY TYPE OF RANDOM PHASE RAN.
C          FOR GAUSSIAN CASE, SUPPLY MEAN AND STANDARD DEVIATION.
C          (DEFAULT VALUES: MEAN=0.0, STDV=50*PI)
  161 AMEAN=0.0
      SFCTR=50.0
      WRITE(LU,625)
  625 FORMAT(" INPUT --- ICON,AMEAN,SFCTR")
      READ(LU,*) ICON,AMEAN,SFCTR
      STDV1=SFCTR*PI
      ISHEET=NSMPL/NP
C
C....... TO ASSIGN DEFAULT PARAMETER VALUES FOR PLOTTING.
      NX=1
      XMIN=T(1)
```

```
      XMAX=TT
      NY=1
      YMIN=-200.
      YMAX=200.
      MO=-1
      JDC=22
      NW=1
      WMIN=0.0
      WMAX=W(NFOLD)
      NS=1
      SMAX=1.0E4
      SMIN=0.0
      MINX0=150
      MAXX0=950
      MINY0=150
      MAXY0=700
      DO 120 I=1,ISHEET
      DO 130 J=1,NP
C
C....... TO GENERATE RANDOM PHASE ANGLE RAN.
      IF(ICON.NE.3) GO TO 410
C
C.......... FOR GAUSSIAN DISTRIBUTION.
      CALL JNORM(IX,IY,IX2,IY2,STDV1,STDV2,AMEAN,CORRL,
     *RAN,RAN2,-1,1)
      GO TO 420
  410 CALL RANDU(IX,IY,RAN)
      IF(ICON.NE.2) GO TO 430
C
C.......... FOR UNIFORM DISTRIBUTION BETWEEN -PI/2 AND +PI/2.
      RAN=(RAN-0.5)*PI
      GO TO 420
C
C.......... FOR UNIFORM DISTRIBUTION BETWEEN -PI AND +PI.
  430 RAN=(RAN-0.5)*PI2
C
C....... TO GENERATE SAMPLE FUNCTION OF THE FIRST KIND.
  420 CCOS=COS(RAN)
      SSIN=SIN(RAN)
      DO 140 K=1,NDATA
  140 X(K,J)=X0(K)*CCOS-XHIL(K)*SSIN
      IF(ISSW(14).LT.0) GO TO 130
C
C....... TO STORE SIMULATED PROCESSES ONTO MAGTAPE.
      CALL EXEC(2,110B,X(1,J),NDATA2)
  130 CONTINUE
C
C....... TO SKIP PLOTTING ROUTINE IF ISSW(15) IS ON.
      IF(ISSW(15).LT.0) GO TO 120
C
C....... TO PLOT SIMULATED PROCESSES.
      WRITE(LU,640)
  640 FORMAT(" TIME HISTORY PLOTTING"/
     *" INPUT---NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC"/
     *"          ---MINX0,MAXX0,MINY0,MAXY0")
   10 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC,
     *MINX0,MAXX0,MINY0,MAXY0
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 11
      NP1=NP+1
      CALL STUKO(T,X0,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NP1,MO,
```

```
     *1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      DO 12 J=1,NP
      J1=J+1
      CALL STUKO(T,X(1,J),NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP1,MO,J1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
   12 CONTINUE
      GO TO 10
C
C....... TO COMPUTE CORRESPONDING FOURIER SPECTRUM.
   11 DO 13 J=1,NP
      CALL SPECT(A,X(1,J),NDATA,TT,1)
      DO 13 K=1,NFOLD
      X(K,J)=A(K)
   13 CONTINUE
C
C.... TO PLOT FOURIER SPECTRUM.
      WRITE(LU,650)
  650 FORMAT(" FOURIER SPECTRUM PLOTTING"/
     *" INPUT---NS,SMIN,SMAX,NW,WMIN,WMAX,MO,JDC"/
     *"         ---MINX0,MAXX0,MINY0,MAXY0")
   15 READ(LU,*) NS,SMIN,SMAX,NW,WMIN,WMAX,MO,JDC,
     *MINX0,MAXX0,MINY0,MAXY0
      IF(NS.EQ.100) STOP
      IF(NS.LT.0) GO TO 120
      CALL STUKO(W,PSD,NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,NP1,MO,
     *1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
      DO 17 J=1,NP
      J1=J+1
      CALL STUKO(W,X(1,J),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,NP1,MO,
     *J1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
   17 CONTINUE
      GO TO 15
  120 CONTINUE
      REWIND IIN
      STOP
      END
      END$
```

```
HSIM30 T=00004 IS ON CR00002 USING 00038 BLKS R=0280

FTN4
C     -------------------------------(MAIN PROGRAM SIM30)-------------
      PROGRAM SIM30
C.... THIS PROGRAM GETS AND PLOTS ENSEMBLE AS WELL AS TEMPORAL
C     STATISTICS OF THE DATA-BASED NONSTATIONARY RANDOM PROCESS
C     SUCH AS MAX, MIN, MEAN, STANDARD DEVIATION, PROBABILITY
C     DENSITY FUNCTION AT T=T1 (FOR ENSEMBLE) OR K=K1 (FOR
C     TEMPORAL).
C
C.... REQUIRED SUBROUTINES:
C       STUKO,STATC,HLBRT,FFT
C.... REMARKS:
C       ISSW(13)=1 --- DENSITY PLOTTING AT T=T1 ONLY.
C       ISSW(14)=1 --- TEMPORAL STATISTIC PLOTTING IS SKIPPED.
C       ISSW(15)=1 --- THEORETICAL PDF (FIRST KIND) IS PLOTTED.
C....   DATA FORMAT ON MAG-TAPE SHOULD BE:
C         NDATA,NSMPL,DT,DW,TT --- FORMAT(2I5,3E12.5)
C                         T(I) ---CALL EXEC(2,110B,T,NDATA2)
C                        X0(I) --- CALL EXEC(2,110B,X0,NDATA2)
C                         X(I) --- CALL EXEC(2,110B,X,NDATA2)
C
      DIMENSION FMAX(512),FMIN(512),FSTDV(512),FMEAN(512)
      DIMENSION EMAX(512),EMIN(512),ESTDV(512),EMEAN(512)
      DIMENSION T(512),X0(512),XSIM(512),T1(5),FFT0(1024)
      DIMENSION HIST(51,5),AMP(51,5),IPOS(5),XHIL(512),A0(5)
      EQUIVALENCE (EMAX(1),HIST),(EMIN(1),AMP),(XHIL,EMEAN)
      EQUIVALENCE (FFT0(1),FMAX),(FFT0(513),FMIN)
C
      LU=1
      IIN=8
      PI=3.141593
      PI2=2.0*PI
C
C.... TO READ IN NECESSARY INFORMATION FROM MAGTAPE.
      REWIND IIN
      READ(IIN,600) NDATA,NSMPL,DT,DW,TT
  600 FORMAT(2I5,3E12.5)
      NDATA2=NDATA*2
      CALL EXEC(1,110B,T,NDATA2)
      CALL EXEC(1,110B,X0,NDATA2)
      NSMPL2=NSMPL*2
      ANDATA=FLOAT(NDATA)
      ANDAT1=ANDATA-1.0
      ANSMPL=FLOAT(NSMPL)
      ANSMP1=ANSMPL-1.0
      NFOLD=NDATA/2+1
      DF=1.0/TT
      FNIQU1=DW*FLOAT(NFOLD)
      FNIQU2=DF*FLOAT(NFOLD)
      WRITE(6,610) TT,DT,DW,DF,FNIQU1,FNIQU2,NDATA,NFOLD
  610 FORMAT(1H1///3X,"SIMULATION OF NONSTATIONARY PROCESS"/
     1" WITH PHASE ANGLE HAVING UNIFORM DISTRIBUTION."///
     23X," TOTAL TIME PERIOD       T=",E11.4/
     33X," TIME INCREMENT         DT=",E11.4/
     43X," FREQUENCY INCREMENT    DW=",E11.4/
     53X,"                        DF=",E11.4/
     63X," NYQUIST FREQUENCY FNIQU1=",E11.4,"  (RAD./SEC)"/
     73X,"                   FNIQU2=",E11.4,"  (1/SEC)"/
     83X," NUMBER OF DATA      NDATA=",I5/
```

```
     93X," NYQUIST FOLD. NO.  NFOLD=",I5///)
C
C.... TO COMPUTE AND PLOT STATISTICS (TEMPORAL AND OR ENSEMBLE).
C     IF ISSW(13) IS ON, SKIP THIS PROCESS AND GO DIRECTLY TO
C     THE COMPUTATION OF PROBABILITY DENSITY FUNCTION.
C       FMAX,FMIN,FSTDV,FMEAN --- FOR ENSEMBLE.
C       EMAX,EMIN,ESTDV,EMEAN --- FOR TEMPORAL.
      IF(ISSW(13).LT.0) GO TO 42
      DO 200 J=1,NDATA
      FMAX(J)=-1.0E20
      FMIN(J)=1.0E20
      FSTDV(J)=0.
      FMEAN(J)=0.
  200 CONTINUE
      DO 210 I=1,NSMPL
      CALL EXEC(1,110B,XSIM,NDATA2)
      CALL STATC(XSIM,NDATA,EMEAN(I),EVR,ESTDV(I),EMAX(I),EMIN(I))
      DO 220 J=1,NDATA
      DDJ=XSIM(J)
      FMEAN(J)=FMEAN(J)+DDJ
      FSTDV(J)=FSTDV(J)+DDJ*DDJ
      IF(DDJ.GT.FMAX(J)) FMAX(J)=DDJ
      IF(DDJ.LT.FMIN(J)) FMIN(J)=DDJ
  220 CONTINUE
  210 CONTINUE
      DO 230 J=1,NDATA
      FMEAN(J)=FMEAN(J)/ANSMPL
      FJD=FSTDV(J)/ANSMP1-FMEAN(J)**2
      IF(FJD.LT.0.) FJD=0.
      FSTDV(J)=SQRT(FJD)
  230 CONTINUE
C
C.... TO PLOT THE TEMPORAL INFORMATION.
C     IF ISSW(14) IS ON, SKIP THIS PROCESS.
      IF(ISSW(14)) 32,2,2
    2 MO=-1
      NX=1
      XMIN=0.0
      XMAX=ANSMPL
      NY=1
      YMAX=200.
      YMIN=-200.
      NP=4
      N=NSMPL
      JDC=22
      MINX=150
      MAXX=950
      MINY=150
      MAXY=700
      DO 240 I=1,NSMPL
  240 XSIM(I)=FLOAT(I)
      WRITE(LU,630)
  630 FORMAT(" INPUT--NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC"
     */"        --MINX,MAXX,MINY,MAXY")
   30 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC,
     *MINX,MAXX,MINY,MAXY
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 32
      CALL STUKO(XSIM,EMEAN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(XSIM,ESTDV,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
```

```
     2NP,MO,2,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(XSIM,EMAX,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,3,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(XSIM,EMIN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,4,LU,JDC,MINX,MAXX,MINY,MAXY)
      GO TO 30
C
C.... TO PLOT ENSEMBLE STATISTICS AT T=T1.
C     T1 CAN BE SPECIFIED ARBITRARILY (UP TO 5 CASES).
   32 MO=-1
      NX=1
      XMIN=0.
      XMAX=40.
      NY=1
      YMIN=-200.
      YMAX=200.
      NP=5
      N=NDATA
      JDC=22
      MAXX=950
      MAXY=700
      MINX=150
      MINY=150
      WRITE(LU,630)
   40 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC
     *,MINX,MAXX,MINY,MAXY
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 42
      CALL STUKO(T,X0,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(T,FMEAN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,2,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(T,FSTDV,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,3,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(T,FMAX,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     2NP,MO,4,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(T,FMIN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,5,LU,JDC,MINX,MAXX,MINY,MAXY)
      GO TO 40
C
C.... TO PLOT PROBABILITY DENSITY AT T=T1.
C     USER MUST SPECIFY T1 VALUE THROUGH THE TERMINAL.
   42 REWIND IIN
      JDC=22
      MINX=150
      MAXX=950
      MINY=150
      MAXY=700
      WRITE(LU,640)
  640 FORMAT(" INPUT THE VALUES RELATED TO HISTOGRAMS."/
     1"  NSTEP -- NUMBER OF STEPS (UP TO 51)"/
     2"            MUST BE ODD INTEGER VALUE."/
     3"  NT1   -- NUMBER OF SPECIFIC TIMES REQUIRED (UP TO 10)")
      READ(LU,*) NSTEP,NT1
      WRITE(LU,650)
  650 FORMAT(" INPUT -- T1 = SPECIFIC TIME TO BE CONSIDERED"/
     1"                MUST BE NT1 NUMBERS."/)
      READ(LU,*)(T1(I),I=1,NT1)
      DO 250 I=1,NT1
      IPOSI=IFIX(T1(I)/DT)
      IPOS2=IPOSI+1
```

```
      IF((T1(I)-T(IPOSI)).GT.(T(IPOS2)-T1(I))) IPOSI=IPOS2
      IPOS(I)=IPOSI
      WRITE(6,660) T1(I),IPOSI,T(IPOSI)
  660 FORMAT(3X,"  T1=",F7.3,2X," I=",I4,2X," TRUE T=",F7.3)
  250 CONTINUE
C
C.... DUMMY READING OF MAGTAPE TO GET THE CORRECT DATA POSITION.
      REWIND IIN
      READ(IIN,600)NDMY1,NDMY2,DMY1,DMY2,DMY3
      DO 270 I=1,2
  270 CALL EXEC(1,110B,XSIM,NDATA2)
C
C.... TO COMPUTE HILBERT TRANSFORM XHIL(T) OF X0(T).
      CALL HLBRT(FFT0,X0,XHIL,NDATA,0)
C
C.... TO SPECIFY AMPLITUDE OF EACH STEP
      NSTEP2=NSTEP/2
      DO 280 J=1,NT1
      IPOSJ=IPOS(J)
      A0(J)=SQRT(X0(IPOSJ)**2+XHIL(IPOSJ)**2)
      DLTJ=A0(J)*2.0/FLOAT(NSTEP-1)
      DO 280 I=1,NSTEP
      AMP(I,J)=DLTJ*FLOAT(I-NSTEP2-1)
  280 CONTINUE
      DO 282 J=1,NT1
      DO 282 I=1,NSTEP
  282 HIST(I,J)=0.0
C
C.... TO COUNT THE FREQUENCY FOR EACH STEP.
      DO 290 I=1,NSMPL
      CALL EXEC(1,110B,XSIM,NDATA2)
      DO 300 J=1,NT1
      IPOSJ=IPOS(J)
      DDJ=XSIM(IPOSJ)
      DLTJ2=(AMP(NSTEP,J)-AMP(1,J))/(2.0*FLOAT(NSTEP-1))
      DFREQ=1.0/(ANSMPL*2.0*DLTJ2)
      IF(DDJ.GT.AMP(1,J)) GO TO 301
      HIST(1,J)=HIST(1,J)+DFREQ
      GO TO 300
  301 IF(DDJ.LT.AMP(NSTEP,J)) GO TO 302
      HIST(NSTEP,J)=HIST(NSTEP,J)+DFREQ
      GO TO 300
  302 DO 310 K=1,NSTEP
      IF(ABS(DDJ-AMP(K,J)).GT.DLTJ2) GO TO 310
      HIST(K,J)=HIST(K,J)+DFREQ
      GO TO 300
  310 CONTINUE
  300 CONTINUE
  290 CONTINUE
      MO=0
      NX=1
      XMIN=-200.0
      XMAX=200.0
      NY=1
      YMIN=0.
      YMAX=0.5
      N=NSTEP
      NP=NT1
      IF(ISSW(15))320,321,321
  321 WRITE(LU,630)
   60 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO
```

```
     *,JDC,MINX,MAXX,MINY,MAXY
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 320
      DO 350 I=1,NP
      CALL STUKO(AMP(1,I),HIST(1,I),N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     1NP,MO,I,LU,JDC,MINX,MAXX,MINY,MAXY)
  350 CONTINUE
      GO TO 60
  320 DO 750 I=1,NP
      WRITE(LU,630)
      MO=0
      NPO=NP
   61 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NPO,MO,JDC,
     *MINX,MAXX,MINY,MAXY
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 750
      DDJ=2.0*A0(I)/ANDATA
      N1=NDATA-1
      DO 330 J=1,N1
      FMAX(J)=-A0(I)+DDJ*FLOAT(J)
      ESTDV(J)=1.0/(PI*SQRT(A0(I)**2-FMAX(J)**2))
  330 CONTINUE
      CALL STUKO(FMAX,ESTDV,N1,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *2,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
      CALL STUKO(AMP(1,I),HIST(1,I),N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *2,MO,2,LU,JDC,MINX,MAXX,MINY,MAXY)
      GO TO 61
  750 CONTINUE
   62 REWIND IIN
      WRITE(LU,680)
  680 FORMAT(" NEED ANOTHER TRIAL FOR T1 ?"/
     1"  IF YES ----- TYPE  1"/
     2"  IF NO  -----      -1"/)
      READ(LU,*) IJDG
      IF(IJDG) 64,42,42
   64 STOP
      END
      END$
```

```
HSIM40 T=00004 IS ON CR00002 USING 00022 BLKS R=0170

FTN4
C     ------------------------------(MAIN PROGRAM SIM40)-------------
      PROGRAM SIM40
C.... THIS PROGRAM SIMULATES AND PLOTS DATA-BASED BIVARIATE
C     NONSTATIONARY RANDOM PROCESS OF THE FIRST KIND AS WELL AS
C     THEIR FOURIER AMPLITUDE SPECTRA WITH RANDOM PHASE ANGLE
C     BEING JOINT NORMAL DISTRIBUTION.
C
C.... REQURED SUBROUTINES:
C       STUKO,SPECT,RANDU,INTPL,JNORM,FFT,HLBRT
C.... REMARKS:
C       BIVARIATE ORIGINAL RECORD MUST BE USED.
C
      DIMENSION A(1024),X01(513),X02(513),XHIL1(513),XHIL2(513)
     *,T(513),XSIM2(514),XNS(513,2),XEW(513,2),W(257),SPD(257,2)
      EQUIVALENCE (XSIM2,SPD),(W,A(1)),(T,A(511))
      LU=1
      WRITE(LU,601)
  601 FORMAT(/" INPUT THE FOLLOWING VALUES,"/
     *" NORG1 -- NUMBER OF NS DATA"/
     *" NORG2 -- NUMBER OF EW DATA"/
     *" NDATA -- NUMBER OF INTERPOLATED DATA"/)
      READ(1,*) NORG1,NORG2,NDATA
C
C     TO READ THE ORIGINAL TIME SERIES
      REWIND 8
      READ(8,300)(A(I),XHIL1(I),I=1,NORG1)
  300 FORMAT(6F10.3)
      READ(8,300)(XSIM2(I),XHIL2(I),I=1,NORG2)
      REWIND 8
C
      NFOLD=NDATA/2+1
      NDATA1=NDATA+1
      TT=A(NORG1)-A(1)
      DT=TT/FLOAT(NDATA)
      PI =3.1415926
      PI2=PI*2.0
      DW=PI2/TT
C
C.... INTERPOLATION OF THE ORIGINAL TIME SERIES
      CALL INTPL(A,XHIL1,NORG1,T,X01,NDATA1,DT)
      CALL INTPL(XSIM2,XHIL2,NORG2,T,X02,NDATA1,DT)
C
C.... TO COMPUTE HILBERT TRANSFORM OF ORIGINAL TIME HISTORY.
      CALL HLBRT(A,X01,XHIL1,NDATA,0)
      CALL HLBRT(A,X02,XHIL2,NDATA,0)
C
C.... TO COMPUTE FOURIER AMPLITUDE SPECTRUM OF X0(T).
      CALL SPECT(A,X01,NDATA,TT,1)
      DO 700 I=1,NFOLD
  700 SPD(I,1)=A(I)
      CALL SPECT(A,X02,NDATA,TT,1)
      DO 710 I=1,NFOLD
  710 SPD(I,2)=A(I)
C
C.... TO GENERATE JOINT GAUSSIAN RANDOM PHASE ANGLE.
      SFCTR1=50.0
      CORRL=0.999
      AM1=0.
```

```
      NSMPL=500
      WRITE(LU,880)
  880 FORMAT(" INPUT--- NSMPL,CORRL,SFCTR1,AM1")
      READ(LU,*)NSMPL,CORRL,SFCTR1,AM1
      STDV1=SFCTR1*PI
      STDV2=STDV1
      NSMPL2=NSMPL/2
      MO=-1
      NX=1
      XMAX=TT
      XMIN=T(1)
      NY=1
      YMIN=-200.
      YMAX=200.
      NP=20303
      MINX0=50
      MAXX0=1020
      MINY0=150
      MAXY0=680
      JDCXY=0
      NS=1
      SMIN=0.0
      SMAX=500.
      NW=1
      WMIN=0.
      WMAX=DW*FLOAT(NFOLD-1)
      IX1=12347
      IY1=0
      IX2=30011
      IY2=0
      DO 250 IJK=1,NSMPL2
      DO 255 KK=1,2
      CALL JNORM(IX1,IY1,IX2,IY2,STDV1,STDV2,AM1,CORRL ,
     *RAN1,RAN2,-1)
      CCOS1=COS(RAN1)
      SSIN1=SIN(RAN1)
      CCOS2=COS(RAN2)
      SSIN2=SIN(RAN2)
      DO 220 J=1,NDATA
      XNS(J,KK)=X01(J)*CCOS1-XHIL1(J)*SSIN1
      XEW(J,KK)=X02(J)*CCOS2-XHIL2(J)*SSIN2
  220 CONTINUE
  255 CONTINUE
      JREP=-1
      WRITE(LU,555)
  555 FORMAT(" INPUT--- NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO"/
     *" INPUT--- JDCXY,MINX0,MAXX0,MINY0,MAXY0")
   10 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,
     *JDCXY,MINX0,MAXX0,MINY0,MAXY0
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 249
      IF(JREP.GT.0) GO TO 887
      DO 888 I=1,NDATA
  888 T(I)=DT*FLOAT(I-1)
  887 CALL STUKO(T,X01,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP,MO,1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      DO 333 J=1,2
      J1=J+1
      CALL STUKO(T,XNS(1,J),NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
  333 CONTINUE
```

```
      CALL STUKO(T,X02,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP,MO,4,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      DO 334 J=1,2
      J1=J+4
      CALL STUKO(T,XEW(1,J),NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
  334 CONTINUE
      JREP=1
      GO TO 10
  249 WRITE(LU,555)
      JREP=-1
   20 READ(LU,*)NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,JDCXY,
     *MINX0,MAXX0,MINY0,MAXY0
      IF(NS.EQ.100) STOP
      IF(NS.LT.0) GO TO 250
      IF(JREP.GT.0) GO TO 891
      DO 890 I=1,NFOLD
  890 W(I)=DW*FLOAT(I-1)
  891 CALL STUKO(W,SPD(1,1),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
     *NP,MO,1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      DO 400 J=1,2
      IF(JREP.GT.0) GO TO 892
      CALL SPECT(A,XNS(1,J),NDATA,TT,1)
      DO 401 KK=1,NFOLD
  401 XNS(KK,J)=A(KK)
      DO 900 I=1,NFOLD
  900 W(I)=DW*FLOAT(I-1)
  892 J1=J+1
      CALL STUKO(W,XNS(1,J),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
     *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
  400 CONTINUE
      CALL STUKO(W,SPD(1,2),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
     *NP,MO,4,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      DO 410 J=1,2
      IF(JREP.GT.0) GO TO 893
      CALL SPECT(A,XEW(1,J),NDATA,TT,1)
      DO 402 KK=1,NFOLD
  402 XEW(KK,J)=A(KK)
      DO 910 I=1,NFOLD
  910 W(I)=DW*FLOAT(I-1)
  893 J1=J+4
      CALL STUKO(W,XEW(1,J),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
     *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
  410 CONTINUE
      JREP=1
      GO TO 20
  250 CONTINUE
      STOP
      END
      END$
```

453 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

HSIM50 T=00004 IS ON CR00002 USING 00020 BLKS R=0159

```
FTN4
C      ------------------------------(MAIN PROGRAM SIM50)------------
       PROGRAM SIM50
C.... THIS PROGRAM COMPUTES AND PLOTS CROSS CORRELATION FUNCTION
C      BETWEEN X1(T) AND X2(T) SIMULATED BY DATA-BASED BIVARIATE
C      NONSTATIONARY PROCESS SIMULATION METHOD OF THE FIRST KIND
C      WITH RANDOM PHASE ANGLE BEING JOINT GAUSSIAN.
C
C.... REQUIRED SUBROUTINES:
C         STUKO,RANDU,HLBRT,FFT,INTPL,JNORM
C
       DIMENSION A(1024),X01(512),X02(512),XHIL1(512),XHIL2(512),
      *T(512),XSIM2(512),XYREAL(512,2),XYIMAG(512,2),IPOS(2),TPOS(2)
       EQUIVALENCE (A(513),T),(A(1),XSIM2)
C
       LU=1
       WRITE(LU,601)
   601 FORMAT(/" INPUT THE FOLLOWING VALUES."/
      *" NORG1 -- NUMBER OF NS DATA"/
      *" NORG2 -- NUMBER OF EW DATA"/
      *" NDATA -- NUMBER OF INTERPOLATED DATA"/)
       READ(LU,*) NORG1,NORG2,NDATA
C
C.... TO READ THE ORIGINAL TIME SERIES
       REWIND 8
       READ(8,300)(A(I),XHIL1(I),I=1,NORG1)
   300 FORMAT(6F10.3)
       READ(8,300)(XSIM2(I),XHIL2(I),I=1,NORG2)
       REWIND 8
C
       NFOLD=NDATA/2+1
       NDATA1=NDATA+1
       TT=A(NORG1)-A(1)
       DT=TT/FLOAT(NDATA)
       PI =3.1415926
       PI2=PI*2.0
       DW=PI2/TT
C
C.... INTERPOLATION OF THE ORIGINAL TIME SERIES
       CALL INTPL(A,XHIL1,NORG1,XYREAL(1,1),X01,NDATA,DT)
       CALL INTPL(XSIM2,XHIL2,NORG2,XYREAL(1,1),X02,NDATA,DT)
C
C.... TO COMPUTE HILBERT TRANSFORM XHIL(T) OF X0(T).
       CALL HLBRT(A,X01,XHIL1,NDATA,0)
       CALL HLBRT(A,X02,XHIL2,NDATA,0)
C
C.... TO CALCULATE EXPECTED VALUE SUCH AS
C.... EXP(I*(PHAI1-PHAI2)),ETC.
    30 NSMPL=500
       IX1=12347
       IY1=0
       IX2=30011
       IY2=0
       CORRL=0.
       AM1=0.
       SFCTR1=50.0
       WRITE(LU,650)
   650 FORMAT(" INPUT-- NSMPL,CORRL,SFCTR1,AM1")
       READ(LU,*)NSMPL,CORRL,SFCTR1,AM1
```

```
      STDV1=PI*SFCTR1
      STDV2=STDV1
      CCOSP=0.
      CCOSM=0.
      SSINP=0.
      SSINM=0.
      DO 200 I=1,NSMPL
      CALL JNORM(IX1,IY1,IX2,IY2,STDV1,STDV2,AM1,CORRL,
     *RAN1,RAN2,1)
      RANP=RAN1+RAN2
      RANM=RAN1-RAN2
      CCOSP=CCOSP+COS(RANP)
      CCOSM=CCOSM+COS(RANM)
      SSINP=SSINP+SIN(RANP)
      SSINM=SSINM+SIN(RANM)
  200 CONTINUE
      ANSMPL=FLOAT(NSMPL)
      CCOSP=CCOSP/ANSMPL
      CCOSM=CCOSM/ANSMPL
      SSINP=SSINP/ANSMPL
      SSINM=SSINM/ANSMPL
      WRITE(6,670)CCOSP,CCOSM,SSINP,SSINM
  670 FORMAT(1H1////
     *"    EXPECTED VALUE CCOSP=",E12.5/
     *"    EXPECTED VALUE CCOSM=",E12.5/
     *"    EXPECTED VALUE SSINP=",E12.5/
     *"    EXPECTED VALUE SSINM=",E12.5/)
C
C.... CALCULATION OF CROSS CORRELATION.
      WRITE(LU,680)
  680 FORMAT(" INPUT--- NT1 (UP TO 2)")
      READ(LU,*) NT1
      WRITE(LU,681)
  681 FORMAT(" INPUT--- TIME REQURED (UP TO NT1)")
      READ(LU,*)(TPOS(I),I=1,NT1)
      DO 222 I=1,NT1
      IIPOS=IFIX(TPOS(I)/DT)
      IF(T(IIPOS+1)-TPOS(I).LT.TPOS(I)-T(IIPOS)) IIPOS=IIPOS+1
      IPOS(I)=IIPOS
      WRITE(6,690)IIPOS,T(IIPOS)
  690 FORMAT("  IPOS=",I4,"   TIME=",E12.5)
  222 CONTINUE
      EXPCON=0.5*EXP(-2.0*STDV1**2)
      DO 220 I=1,NT1
      INT=IPOS(I)
      FCTR1=X02(INT)
      FCTR2=XHIL2(INT)
      DO 230 J=1,NDATA
      FC1=X01(J)
      FC2=XHIL1(J)
      XYREAL(J,I)=FNCTR(CCOSM,SSINM,FC1,FC2,FCTR1,-FCTR2)
     *+FNCTR(CCOSP,-SSINP,FC1,-FC2,FCTR1,-FCTR2)
     *+FNCTR(CCOSM,-SSINM,FC1,-FC2,FCTR1,FCTR2)
     *+FNCTR(CCOSP,SSINP,FC1,FC2,FCTR1,FCTR2)
      XYIMAG(J,I)=0.5*(FC1*FCTR1+FC2*FCTR2)+
     *EXPCON*(FC1*FCTR1-FC2*FCTR2)
      XYREAL(J,I)=XYREAL(J,I)/4.0
  230 CONTINUE
  220 CONTINUE
      NX=1
      XMIN=0.0
```

```
      XMAX=DT*FLOAT(NDATA)
      NY=1
      YMAX=1.E6
      YMIN=-1.E6
      MO=-1
      NP=2*NT1
      JDC=22
      MINX=50
      MAXX=1020
      MINY=150
      MAXY=700
      WRITE(LU,555)
  555 FORMAT(" INPUT--- NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO"/
     *" INPUT--- JDC,MINX,MAXX,MINY,MAXY")
   10 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO
     *,JDC,MINX,MAXX,MINY,MAXY
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 20
      DO 350 K=1,NDATA
  350 T(I)=DT*FLOAT(I-1)
      DO 333 K=1,NT1
  333 CALL STUKO(T,XYIMAG(1,K),NDATA,NX,XMIN,XMAX,
     *NY,YMIN,YMAX,NP,MO,K,LU,JDC,MINX,MAXX,MINY,MAXY)
      DO 334 K=1,NT1
      K1=K+NT1
  334 CALL STUKO(T,XYREAL(1,K),NDATA,NX,XMIN,XMAX,
     *NY,YMIN,YMAX,NP,MO,K1,LU,JDC,MINX,MAXX,MINY,MAXY)
      GO TO 10
   20 WRITE(LU,888)
  888 FORMAT(" ANOTHER TRIAL?")
      READ(LU,*)IJD
      IF(IJD.GT.0) GO TO 30
      STOP
      END
      FUNCTION FNCTR(A,B,C,D,E,F)
      FNCTR=E*(A*C-B*D)-F*(A*D+B*C)
      RETURN
      END
      END$
```

```
HSIM60 T=00004 IS ON CR00002 USING 00077 BLKS R=0610

      FTN4
C     ------------------------------(MAIN PROGRAM SIM60)-------------
      PROGRAM SIM60
C.... THIS PROGRAM GENERATES AND PLOTS SAMPLE FUNCTIONS OF
C     DATA-BASED NONSTATIONARY RANDOM PROCESS OF THE SECOND KIND
C     WITH UNIFORM RANDOM PHASE ANGLE BETWEEN -PI/2 AND PI/2.
C     CORRESPONDING FOURIER SPECTRA AND STATISTICAL QUANTITIES
C     ARE ALSO COMPUTED AND PLOTTED.
C       -------------------------------------------------------------
C       DATA-BASED NONSTATIONARY RANDOM PROCESS OF THE SECOND KIND
C       -------------------------------------------------------------
C
C.... REQUIRED SUBROUTINES:
C       STUKO, RANDU, SPECT, ARNGE, HLBRT, INTPL, FFT
C.... REMARKS:
C       THIS PROGRAM IS CODED FOR HP MINI-COMPUTER WITH TEKTRONIX
C       TERMINAL CONTROL SYSTEM FOR PLOTTING. THEREFORE, FOR IBM
C       COMPUTER SLIGHT MODIFICATION IS NEEDED.
C       BECAUSE OF THE LIMITED MEMORY SIZE IN MINI-COMPUTER,
C       "NDATA" IS ASSIGNED TO BE AT MOST 512.
C.... REQUIREMENT OF ARRAY SIZE:
C       A(N+1),F0(2*N+2),T(N+1),X0(N+1),XHIL(N+1),SW(N+1)
C       W(N/2+1),RV(NSTEP),WA(N/2+1),WM(N/2+1)
C       WHERE N=NDATA=A POWER OF TWO FOR FFT COMPUTATION,
C             NSTEP=# OF AMPLITUDE LEVEL FOR PDF COMPUTATION.
C
      DIMENSION A(515),F0(1030),T(515),X0(515),XHIL(515)
      DIMENSION SW(515),W(257),RV(81),WA(257),WM(257)
C
C.... TO ASSIGN NUMBER OF INTERPOLATED DATA FOR FFT COMPUTATION.
      DATA NDATA/512/
C
C.... TO ASSIGN SELECT CODE OF INPUT/OUTPUT PERIPHERAL.
      LU=1
      IIN=8
      IO=6
C
C.... TO READ IN OBSERVED RECORD FROM MAG-TAPE.
C       NORG  -- NUMBER OF DATA
C       F0(I) -- TIME
C       A(I)  -- OBSERVED VALUE
      REWIND IIN
      WRITE(LU,600)
  600 FORMAT(" INPUT -- NORG= NO. OF ORIGINAL DATA")
      READ(LU,*) NORG
      READ(IIN,100) (F0(I),A(I),I=1,NORG)
  100 FORMAT(6F10.3)
      REWIND IIN
C
C.... TO PREPARE APPROPRIATE QUANTITIES FOR LATER COMPUTATION.
      NDATA1=NDATA+1
      ND2=NDATA/2
      ND4=ND2/2
      NFOLD=ND2+1
      TFST=F0(1)
      TTOTAL=F0(NORG)-TFST
      DT=TTOTAL/FLOAT(ND2)
      PI=3.141593
      DW=2.0*PI/TTOTAL
```

```
      DO 350 JJ=1,NFOLD
  350 W(JJ)=DW*(JJ-1)
C
C.... TO ADJUST TIME AXIS FOR DATA TO START FROM T=0.0 (SEC).
      DO 352 I=1,NORG
  352 F0(I)=F0(I)-TFST
C
C.... TO INTERPOLATE ORIGINAL RECORD TO BE ARRANGED EQUIDISTANTLY.
      CALL INTPL(F0,A,NORG,T,X0,ND2,DT)
      X0(NFOLD)=0.0
C
C.... TO GET SYMMETRIC-PERIODIC EXTENSION OF THE ORIGINAL RECORD
C     AND ALSO COMPUTE TIME ARRAY.
      CALL ARNGE(X0,NDATA,10,1)
      DO 330 I=1,NDATA1
  330 T(I)=DT*FLOAT(I-NFOLD)
C
C.... TO COMPUTE HILBERT TRANSFORM OF SYMMETRIC-PERIODIC
C     EXTENSION OF THE ORIGINAL RECORD.
      CALL HLBRT(F0,X0,XHIL,NDATA,0)
C
C.... TO PLOT ORIGINAL RECORD AS WELL AS HILBERT TRANSFORM OF
C     ITS SYMMETRIC-PERIODIC EXTENSION.
      WRITE(LU,610)
  610 FORMAT(" NEED TO PLOT HILBERT TRANSFORM ?"/
     *" IF YES, TYPE IN +1")
      READ(LU,*) IHIL
      NX=1
      XMIN=-TTOTAL
      XMAX=TTOTAL
      NY=1
      YMIN=-200.0
      YMAX=200.0
      MINX0=150
      MAXX0=950
      MINY0=150
      MAXY0=700
      JDCXY=22
      NP=2
      MO=-1
      IF(IHIL .NE. 1) GO TO 1300
      WRITE(LU,620)
  620 FORMAT(" INPUT-- NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY"/
     *"        -- MINX0,MAXX0,MINY0,MAXY0")
 1301 READ(LU,*) NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY,
     *MINX0,MAXX0,MINY0,MAXY0
      IF(NY .EQ. 100) STOP
      IF(NY .LT. 0) GO TO 1300
      CALL STUKO(T,X0,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,1,
     *LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(T,XHIL,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP,MO,2,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      GO TO 1301
C
C.... TO COMPUTE AND PLOT FOURIER AMPLITUDE OF THE ORIGINAL
C     RECORD, SQRT(AMPL(P0(W))**2+AMPL(Q0(W))**2), AMPL(P0(W)),
C     AND AMPL(Q0(W)), WHERE SYMBOL AMPL(.) REPRESENTS AMPLITUDE
C     OF A COMPLEX VALUE.
 1300 NW=1
      WMIN=W(1)
      WMAX=W(NFOLD)
```

```
      NS=1
      SMAX=1000.
      SMIN=0.0
      JDCWS=11
      NP=4
      MO=-1
      MINWO=350
      MAXWO=800
      MINSO=40
      MAXSO=770
      JPS=1
      NDF=ND2+1
      IF(IHIL .NE. 1) GO TO 1600
      WRITE(LU,630)
  630 FORMAT(" INPUT-- NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,JDCWS"/
     *"       -- MINWO,MAXWO,MINSO,MAXSO")
 1650 READ(LU,*) NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,
     *JDCWS,MINWO,MAXWO,MINSO,MAXSO
      IF(NS.EQ.100) STOP
      IF(NS.LT.0) GO TO 1600
C
C....... TO COMPUTE FOURIER SPECTRUM OF THE ORIGINAL RECORD.
         DO 280 I=1,ND2
         FO(2*I-1)=XO(I+ND2)
  280    FO(2*I)=0.0
         CALL FFTO(FO,A,ND2,-1,1)
         DO 290 I=1,ND2
  290    SW(I)=SQRT(FO(I*2-1)**2+FO(I*2)**2)*TTOTAL
C
C....... TO CHECK THE AREA.
         CALL AREA(AW,SW,ND4,DW)
         RV(1)=AW/PI
 1400    CALL STUKO(W,SW,NDF,NW,WMIN,WMAX,NS,SMIN,SMAX,NP,MO,
     *    JPS,LU,JDCWS,MINWO,MAXWO,MINSO,MAXSO)
      IF(JPS.EQ.2) GO TO 1630
      IF(JPS.EQ.3) GO TO 1631
      IF(JPS.EQ.4) WRITE(IO,66) (RV(LK),LK=1,4)
   66 FORMAT(///1H ,4E12.5)
      IF(JPS.EQ.4) GO TO 1650
C
C.... PRELIMINARY ANALYSYS TO GET PO(W) AND QO(W).
C.... FIRST COMPUTE PO(W).
         DO 250 I=1,ND2
         FO(2*I-1)=XO(I+ND2)*0.5
  250    FO(2*I)=-XHIL(I+ND2)*0.5
         CALL FFTO(FO,A,ND2,-1,1)
C....... TO COMPUTE AMP(PO(W))**2.
         DO 230 I=1,ND2
  230    WA(I)=(FO(2*I-1)**2+FO(2*I)**2)*TTOTAL*TTOTAL
C
C.... SECOND, COMPUTE QO(W).
         DO 260 I=1,ND2
         FO(2*I-1)=XO(I+ND2)*0.5
  260    FO(2*I)=XHIL(I+ND2)*0.5
         CALL FFTO(FO,A,ND2,-1,1)
C....... TO COMPUTE AMPL(QO(W))**2.
         DO 270 I=1,ND2
  270    WM(I)=(FO(2*I-1)**2+FO(2*I)**2)*TTOTAL*TTOTAL
C
C.... TO COMPUTE SQRT(AMPL(PO(W))**2+AMPL(QO(W))**2)
C     AND ALSO CHECK THE AREA.
```

```
            DO 271 I=1,ND2
  271       SW(I)=SQRT(WA(I)+WM(I))
            CALL AREA(AW,SW,ND4,DW)
            RV(2)=AW/PI
         JPS=JPS+1
         GO TO 1400
C
C.... TO COMPUTE AMPL(P0(W)) AND CHECK THE AREA.
 1630       DO 272 I=1,ND2
  272       SW(I)=SQRT(WA(I))
            CALL AREA(AW,SW,ND4,DW)
            RV(3)=AW/PI
         JPS=JPS+1
         GO TO 1400
C
C.... TO COMPUTE AMPL(Q0(W)) AND CHECK THE AREA.
 1631       DO 273 I=1,ND2
  273       SW(I)=SQRT(WM(I))
            CALL AREA(AW,SW,ND4,DW)
            RV(4)=AW/PI
         JPS=JPS+1
         GO TO 1400
C
C.... TO PRINT OUT COMPUTED AREA FOR CHECK PURPOSE.
 1600 WRITE(IO,66) (RV(L3),L3=1,4)
C
C     ------------------------------------------------------------
C.... TO COMPUTE AND PLOT SIMULATED DATA-BASED NONSTATIONARY
C     RANDOM PROCESS OF THE SECOND KIND AND CORRESPONDING FOURIER
C     SPECTRUM.
      NP=1
      WRITE(LU,650)
  650 FORMAT(" INPUT -- NP     = # OF PLOTS ON ONE PAGE"/
     *"        -- NSMPL = # OF SAMPLES TO BE GENERATED"/
     *"        --         NSMPL MUST BE A MULTIPLE OF NP.")
      READ(LU,*) NP, NSMPL
      ISHEET=NSMPL/NP
C
C.... TO SET INITIAL VALUE OF INTERNAL PARAMETER IN SUBROUTINE
C     RANDU (UNIFORM RANDOM NUMBER GENERATING SUBROUTINE).
      IX=12347
      IY=0
      IX1=IX
      IY1=IY
      IX2=IX
      IY2=IY
C
      WRITE(LU,652)
  652 FORMAT(" DO YOU WANT TO PLOT X(T) AND FOURIER SPECTRA ?"/
     *" IF YES, TYPE IN +1")
      READ(LU,*) INS
      IF(INS .NE. 1) GO TO 38
      DO 333 I=1,ISHEET
      IX0=IX1
      IY0=IY1
      WRITE(LU,654) I
  654 FORMAT(" THIS IS (",I3,")-TH PAGE")
      WRITE(LU,620)
 1010 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY,
     *MINX0,MAXX0,MINY0,MAXY0
      IF(NY.EQ.100) STOP
```

```
      IF(NY.EQ.10) GO TO 38
      IF(NY.LT.0) GO TO 334
      NNP1=NP+1
      CALL STUKO(T,X0,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NNP1,MO,
     *1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      IX1=IX0
      IY1=IY0
      DO 320 JM=1,NP
      JM1=JM+1
      CALL SAMPL(A,X0,XHIL,NDATA,IX1,IY1,RAN1)
      CALL STUKO(T,A,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NNP1,MO,
     *JM1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
  320 CONTINUE
      GO TO 1010
  334 IX0=IX2
      IY0=IY2
      WRITE(LU,630)
 1011 READ(LU,*) NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,JDCWS,
     *MINW0,MAXW0,MINS0,MAXS0
      IF(NS.EQ.100) STOP
      IF(NS.EQ.10) GO TO 38
      IF(NS.LT.0) GO TO 333
      DO 381 IS=1,ND2
  381 F0(IS)=X0(IS+ND2)
      CALL SPECT(SW,F0,ND2,TTOTAL,1)
      CALL STUKO(W,SW,ND4,NW,WMIN,WMAX,NS,SMIN,SMAX,NNP1,MO,
     *1,LU,JDCWS,MINW0,MAXW0,MINS0,MAXS0)
      IX2=IX0
      IY2=IY0
      DO 310 J=1,NP
      J1=J+1
      CALL SAMPL(A,X0,XHIL,NDATA,IX2,IY2,RAN2)
      DO 382 JI=1,ND2
  382 F0(JI)=A(JI+ND2)
      CALL SPECT(SW,F0,ND2,TTOTAL,1)
      CALL STUKO(W,SW,ND4,NW,WMIN,WMAX,NS,SMIN,SMAX,NNP1,MO,
     *J1,LU,JDCWS,MINW0,MAXW0,MINS0,MAXS0)
  310 CONTINUE
      GO TO 1011
  333 CONTINUE
C
C     ---------------------------------------------------------------
C.... TO COMPUTE ENSEMBLE STATISTICS OF THE SIMULATED PROCESS.
C.... ASSIGNMENT OF ARRAYS:
C       W(I)  --- MEAN VALUE
C       WA(I) --- MAXIMUM VALUE
C       WM(I) --- MINIMUM VALUE
C       SW(I) --- STANDARD DEVIATION
C       F0(I) --- FOR COMPUTATION OF EMPIRICAL DENSITY FUNCTION
C       IPOS  --- POSITION NUMBER CORRESPONDING TO T=T1 FOR
C                 EMPIRICAL DENSITY COMPUTATION.
   38 CONTINUE
C.... FIRST, SET INITIAL PARAMETER VALUES IN SUBROUTINE RANDU.
      IX1=IX
      IY1=IY
      NN=NDATA*2
C
C.... TO ASSIGN INITIAL VALUES IN ARRAY.
      DO 32 I=1,NN
   32 F0(I)=0.0
      IF(ISSW(15).LT.0) GO TO 1501
```

```
      DO 71 I=1,NFOLD
      SW(I)=0.0
      W(I)=0.0
      WA(I)=-1.0E10
      WM(I)=1.0E10
   71 CONTINUE
C
C.... TO ASSIGN SPECIFIC TIME FOR EMPIRICAL DENSITY TO BE CONSIDERED.
 1501 WRITE(LU,88)
   88 FORMAT(" INPUT TIME T1 (SEC) FOR PDF TO BE CONSIDERED."/
     *" ALSO INPUT NSMPL, IF NECESSARY.")
      READ(LU,*) T1,NSMPL
      IPOS=IFIX(T1/DT)
      TLOW=DT*FLOAT(IPOS)
      TUP=TLOW+DT
      IF(ABS(T1-TUP).LT.ABS(T1-TLOW)) IPOS=IPOS+1
C
C.... TO COMPUTE ENSEMBLE STATISTICS.
C     NSMPL --- SAMPLE SIZE OF MONTE CARLO SIMULATION.
      DO 30 J=1,NSMPL
      CALL SAMPL(A,X0,XHIL,NDATA,IX1,IY1,RAN1)
      F0(J)=A(ND2+IPOS)
      IF(ISSW(15).LT.0) GO TO 30
      DO 31 I=1,ND2
      IND2=I+ND2
      AIND2=A(IND2)
      IF(AIND2.GT.WA(I)) WA(I)=AIND2
      IF(AIND2.LT.WM(I)) WM(I)=AIND2
      W(I)=W(I)+AIND2
      SW(I)=SW(I)+AIND2*AIND2
   31 CONTINUE
   30 CONTINUE
      IF(ISSW(15).LT.0) GO TO 85
      DO 72 I=1,ND2
      W(I)=W(I)/FLOAT(NSMPL)
      SW(I)=SW(I)/FLOAT(NSMPL)
   72 CONTINUE
      DO 73 I=1,ND2
      SW(I)=SQRT(SW(I)-W(I)*W(I))
   73 CONTINUE
C
C.... TO PLOT COMPUTED ENSEMBLE STATISTICS EXCEPT EMPIRICAL PDF.
      NP=5
      MO=-1
      WRITE(LU,620)
  222 READ(LU,*) NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,
     *JDCXY,MINX0,MAXX0,MINY0,MAXY0
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 85
      CALL STUKO(T(NFOLD),X0(NFOLD),ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *NP,MO,1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(T(NFOLD),W,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
     *2,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(T(NFOLD),SW,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
     *3,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(T(NFOLD),WA,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
     *4,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      CALL STUKO(T(NFOLD),WM,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
     *5,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      GO TO 222
C
```

```
C     ------------------------------------------------------------
C.... TO COMPUTE AND PLOT EMPIRICAL AS WELL AS THEORETICAL PROBABILITY
C     DENSITY AT T=T1.
C.... ASSIGNMENT OF ARRAY:
C       W(I)  --- AMPLITUDE LEVEL
C       RV(I) --- EMPIRICAL PROBABILITY DENSITY
C       SW(I) --- THEORETICAL ASYMMETRIC PROBABILITY DENSITY.
   85 CONTINUE
      NSTEP=81
      WRITE(LU,223)
  223 FORMAT(" INPUT-- NSTEP= # OF AMPLITUDE STEPS (UP TO 81)"/
     *       "       -- PREFERABLY NSTEP IS ODD INTEGER.")
      READ(LU,*) NSTEP
C
C.... TO FIND OUT MAXIMUM AND MINIMUM VALUE ON F0(I).
      XMAX=-1.0E10
      XMIN=1.0E10
      DO 521 I=1,NSMPL
      IF(F0(I).GT.XMAX) XMAX=F0(I)
      IF(F0(I).LT.XMIN) XMIN=F0(I)
  521 CONTINUE
C
C.... TO FIND OUT MAX. AND MIN. ON SW(I). ALSO COMPUTE W(I).
      XZ=X0(ND2+IPOS)
      XZT=XHIL(ND2+IPOS)
      CALL ASYME(XZ,XZT,W,SW,NSTEP,XMAX,XMIN,DX)
C
C.... TO COMPUTE EMPIRICAL DENSITY.
      DX2=DX/2.0
      DFREQ=1.0/(FLOAT(NSMPL)*DX)
      DO 530 I=1,NSTEP
  530 RV(I)=0.0
      DO 516 J=1,NSMPL
      F0J=F0(J)
      DO 517 I=1,NSTEP
      IF(W(I).LT.F0J) GO TO 517
      I1=I
      IF(ABS(F0J-W(I-1)).LT.DX2) I1=I1-1
      RV(I1)=RV(I1)+DFREQ
      GO TO 516
  517 CONTINUE
  516 CONTINUE
C
C.... TO PRINT OUT DENSITY FUNCTIONS.
      WRITE(6,540) (W(I),SW(I),RV(I),I=1,NSTEP)
  540 FORMAT(5X,"AMP=",F10.3,"  THEO.PDF=",F7.4,"  EMP.PDF=",F7.4)
C
C.... TO PLOT EMPIRICAL AND THEORETICAL PROBABILITY DENSITY.
      NX=1
      NY=1
      YMIN=0.0
      YMAX=1.0
      NP=2
      MO=0
      JDCXY=23
      MINX0=150
      MAXX0=950
      MINY0=150
      MAXY0=700
      WRITE(LU,620)
  455 READ(LU,*) NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY,
```

```
     *MINXO,MAXXO,MINYO,MAXYO
      IF(NY.EQ.100) STOP
      IF(NY.LT.0) GO TO 450
      CALL STUKO(W,SW,NSTEP,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
     *1,LU,JDCXY,MINXO,MAXXO,MINYO,MAXYO)
      CALL STUKO(W,RV,NSTEP,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
     *2,LU,JDCXY,MINXO,MAXXO,MINYO,MAXYO)
      GO TO 455
  450 WRITE(LU,690)
  690 FORMAT(" NEED PDF PLOTTING AT DIFERRENT T=T1 ?"/
     *" IF YES, TYPE IN +1"/
     *" IF NO NEED OF STATISTICS PLOTTING, SET ISSW(15) ON.")
      READ(LU,*) IJD
      IF(IJD.EQ.1) GO TO 38
      STOP
      END
C
C     --------------------------------(SUBROUTINE AREA)-------------
      SUBROUTINE AREA(SUM,X2,N,DA)
      DIMENSION X2(1)
      SUM=0.0
      DO 10 I=2,N
   10 SUM=SUM+X2(I-1)+X2(I)
      SUM=SUM*DA/2.0
      RETURN
      END
C
C     --------------------------------(SUBROUTINE SAMPL)------------
      SUBROUTINE SAMPL(A,X0,XHIL,NDATA,IX,IY,RAN)
      DIMENSION A(1),X0(1),XHIL(1)
C
C.... TO GENERATE NONSTATIONARY RANDOM PROCESS WITH UNIFORM
C     RANDOM PHASE ANGLE BETWEEN -PI/2 AND +PI/2.
C
C.... DESCRIPTION OF PARAMETERS:
C       X0(I)   --- SAMPLE OBSERVATION TO BE USED FOR SIMULATION.
C       XHIL(I)--- HILBERT TRANSFORM OF X0(I)
C       A(I)    --- GENERATED NONSTATIONARY RANDOM PROCESS
C       NDATA   --- NUMBER OF DATA POINTS
C       IX, IY  --- INTERNAL PARAMETERS IN UNIFORM RANDOM VARIATE
C                   GENERATING SUBROUTINE RANDU(IX,IY,RAN)
C
      PI=3.1415926
      CALL RANDU(IX,IY,RAN)
      RAN=(RAN-0.5)*PI
      CCOS=COS(RAN)
      SSIN=SIN(RAN)
      DO 100 I=1,NDATA
      A(I)=CCOS*X0(I)-SSIN*XHIL(I)
  100 CONTINUE
      RETURN
      END
C
C     --------------------------------(SUBROUTINE FFT0)-------------
      SUBROUTINE FFT0(Y,DATA,N,ISIGN,MW)
      DIMENSION Y(1),DATA(1)
C.... TO TAKE ONE DIMENSIONAL FOURIER (INVERSE) TRANSFORM
C     BY FFT ALGORISM
C.... DESCRIPTION OF PARAMETERS:
C           DATA= ORIGINAL TIME SERIES
C           N= NUMBER OF TIME SERIES
```

```
C                 "N" MUST BE A POWER OF TWO
C             ISIGN ----- IF FOURIER TRANSFORM, ISIGN= -1
C                   ----- IF INVERSE FOURIER,   ISIGN=  1
C             Y= FOURIER COMPLEX COEFFICIENT,
C                OR COMPLEX TIME SERIES.
C                ARRAY MUST BE IN ORDER OF (REAL), (IMAGINARY)
C
      AN=FLOAT(N)
      AISIGN=FLOAT(ISIGN)
      PI=3.141593
      IF(MW.EQ.1) GO TO 10
      IF(ISIGN.EQ.1) GO TO 10
      DO 100 I=1,N
      II=I+I
      IR=II-1
      Y(IR)=DATA(I)
      Y(II)=0.0
  100 CONTINUE
   10 L=1
      DO 200 I=1,N
      IF(I.GE.L) GO TO 210
      LI=L+L
      LR=LI-1
      II=I+I
      IR=II-1
      AR=Y(LR)
      AI=Y(LI)
      Y(LR)=Y(IR)
      Y(LI)=Y(II)
      Y(IR)=AR
      Y(II)=AI
  210 N2=N/2
  220 IF(L.LE.N2) GO TO 230
      L=L-N2
      N2=N2/2
      IF(N2.GE.2) GO TO 220
  230 L=L+N2
  200 CONTINUE
      MAX=1
  240 IF(MAX.GE.N) GO TO 400
      ISTEP=MAX*2
      AMAX=FLOAT(MAX)
      AK=-1.0
      DO 300 K=1,MAX
      AK=AK+1.0
      WT=PI*AISIGN*AK/AMAX
      DO 310 I=K,N,ISTEP
      L=I+MAX
      LI=L+L
      LR=LI-1
      II=I+I
      IR=II-1
      CCOS=COS(WT)
      SSIN=SIN(WT)
      AR=Y(LR)*CCOS-Y(LI)*SSIN
      AI=Y(LR)*SSIN+Y(LI)*CCOS
      Y(LR)=Y(IR)-AR
      Y(LI)=Y(II)-AI
      Y(IR)=Y(IR)+AR
      Y(II)=Y(II)+AI
  310 CONTINUE
```

```
  300 CONTINUE
      MAX=ISTEP
      GO TO 240
  400 IF(ISIGN.EQ.1) RETURN
      NM2=N*2
      DO 450 I=1,NM2
      Y(I)=Y(I)/AN
  450 CONTINUE
      RETURN
      END
C
C     --------------------------------(SUBROUTINE ASYME)-------------
      SUBROUTINE ASYME(XZ,XZT,W,SW,NSTEP,XMAX,XMIN,DX)
      DIMENSION W(1),SW(1)
      PI=3.141593
      PI2=PI/2.0
      XZ2=XZ**2
      XZT2=XZT**2
      AZ=SQRT(XZ2+XZT2)
      CALL PHS11(XZ,XZT,TA)
      ARF=AZ*COS(PI/2.0-ABS(TA))
      BET=AZ*COS(PI/2.0+ABS(TA))
      IF(ABS(TA).GT.PI2) GO TO 400
      AMIN=BET
      AMP=ARF
      AMAX=AZ
      GO TO 401
  400 AMIN=-AZ
      AMP=-ARF
      AMAX=-BET
  401 IF(AMAX.GT.XMAX) XMAX=AMAX
      IF(AMIN.LT.XMIN) XMIN=AMIN
      DDX=0.0001
      XMAX=XMAX+DDX
      XMIN=XMIN-DDX
      DX=(XMAX-XMIN)/FLOAT(NSTEP-1)
      DO 30 I=1,NSTEP
   30 W(I)=DX*FLOAT(I-1)+XMIN
      DO 55 I=1,NSTEP
      WI=W(I)
      R1=PI*SQRT(ABS(AZ**2-WI**2))
      IF(R1 .LT. 0.01) R1=0.01
      SW(I)=1./R1
      IF(WI.GT.AMP) SW(I)=2.0*SW(I)
   55 CONTINUE
      RETURN
      END
C
C     --------------------------------(SUBROUTINE PHS11)------------
      SUBROUTINE PHS11(XZ,XZT,B)
      PI1=3.141593
      PI2=PI1*2.0
      PI12=PI1/2.0
      AR=XZ
      AI=XZT
      IF(ABS(AR).LT.E-20) GO TO 30
      FAI0=AI/AR
      FAI=ATAN(FAI0)
      IF(FAI.LT.0.) GO TO 40
      IF(AR.LT.0.) FAI=FAI-PI1
      GO TO 50
```

```
   40 IF(AR.LT.0.) FAI=FAI+PI1
      GO TO 50
   30 IF(AI.GT.0.) FAI=PI12
      IF(AI.LT.0.) FAI=-PI12
   50 B=FAI
      RETURN
      END
      END$
```

```
IINTPL T*00004 IS ON CR00002 USING 00005 BLKS R*0033

FTN4
C     -------------------------------(SUBROUTINE INTPL)-------------
      SUBROUTINE INTPL(X0,Y0,NORG,X,Y,NDATA,DX)
      DIMENSION X0(1),Y0(1),X(1),Y(1)
C.... THIS SUBROUTINE INTERPOLATES DATA SET (X0,Y0) TO PRODUCE
C     EQUIDISTANT DATA SET (X,Y) WITH ABSCISSA INCREMENT DX.
C.... DESCRIPTION OF PARAMETERS:
C       X0    --- ORIGINAL ABSCISSA ARRAY
C       Y0    --- ORIGINAL ORDINATE ARRAY
C       NORG  --- NUMBER OF ORIGINAL DATA
C       X     --- INTERPOLATED ABSCISSA ARRAY
C       Y     --- INTERPOLATED ORDINATE ARRAY
C       NDATA -- NUMBER OF INTERPOLATED DATA
C       DX    --- EQUIDISTANT INCREMENT OF ABSCISSA
C
      Y(1)=Y0(1)
      X(1)=X0(1)
      JJ=2
      SLOPE=(Y0(2)-Y0(1))/(X0(2)-X0(1))
      DO 10 I=2,NDATA
      X(I)=X(I-1)+DX
   30 IF(X(I).GT.X0(JJ)) GO TO 20
      Y(I)=Y0(JJ-1)+SLOPE*(X(I)-X0(JJ-1))
      GO TO 10
   20 JJ=JJ+1
      IF(JJ.LE.NORG) GO TO 25
      Y(I)=0.0
      GO TO 10
   25 SLOPE=(Y0(JJ)-Y0(JJ-1))/(X0(JJ)-X0(JJ-1))
      GO TO 30
   10 CONTINUE
      RETURN
      END
      END$
```

```
IIFFT  T=00004 IS ON CR00002 USING 00010 BLKS R=0084

      FTN4
C     --------------------------------(SUBROUTINE FFT)---------------
      SUBROUTINE FFT(Y,DATA,N,ISIGN)
      DIMENSION Y(1),DATA(1)
C.... THIS SUBROUTINE COMPUTES ONE-DIMENSIONAL FOURIER OR INVERSE
C     FOURIER TRANSFORM BY FFT (FAST FOURIER TRANSFORM) ALGORITHM.
C.... DESCRIPTION OF PARAMETERS:
C       DATA(I) --- ORIGINAL REAL TIME DOMAIN DATA.
C       N       --- NUMBER OF DATA POINTS (= A POWER OF 2).
C       ISIGN   --- CONTROL PARAMETER.
C                   ISIGN=-1 FOR FOURIER TRANSFORM.
C                   ISIGN=+1 FOR INVERSE FOURIER TRANSFORM.
C       Y(I)    --- FOURIER COMPLEX COEFFICIENT (FREQUENCY DOMAIN)
C                   OR COMPLEX TIME SERIES (TIME DOMAIN).
C                   THIS ARRAY MUST BE IN THE ORDER OF
C                   (REAL-1),(IMAG-1),(REAL-2),(IMAG-2),......
C.... REMARKS:
C       WHEN ISIGN=+1, DATA(I) BECOMES DUMMY.
C       WHEN ISIGN=-1, PROGRAM AUTOMATICALLY TRANSFERS REAL TIME
C       DOMAIN DATA (DATA(I)) INTO COMPLEX ARRAY Y(I).
C       ARRAY SIZE REQUIREMENT --- Y(2*N), DATA(N)
C
      AN=FLOAT(N)
      AISIGN=FLOAT(ISIGN)
      PI=3.141593
      IF(ISIGN.EQ.1) GO TO 10
      DO 100 I=1,N
      II=I+I
      IR=II-1
      Y(IR)=DATA(I)
      Y(II)=0.0
  100 CONTINUE
   10 L=1
      DO 200 I=1,N
      IF(I.GE.L) GO TO 210
      LI=L+L
      LR=LI-1
      II=I+I
      IR=II-1
      AR=Y(LR)
      AI=Y(LI)
      Y(LR)=Y(IR)
      Y(LI)=Y(II)
      Y(IR)=AR
      Y(II)=AI
  210 N2=N/2
  220 IF(L.LE.N2) GO TO 230
      L=L-N2
      N2=N2/2
      IF(N2.GE.2) GO TO 220
  230 L=L+N2
  200 CONTINUE
      MAX=1
  240 IF(MAX.GE.N) GO TO 400
      ISTEP=MAX*2
      AMAX=FLOAT(MAX)
      AK=-1.0
      DO 300 K=1,MAX
      AK=AK+1.0
```

```
      WT=PI*AISIGN*AK/AMAX
      DO 310 I=K,N,ISTEP
      L=I+MAX
      LI=L+L
      LR=LI-1
      II=I+I
      IR=II-1
      CCOS=COS(WT)
      SSIN=SIN(WT)
      AR=Y(LR)*CCOS-Y(LI)*SSIN
      AI=Y(LR)*SSIN+Y(LI)*CCOS
      Y(LR)=Y(IR)-AR
      Y(LI)=Y(II)-AI
      Y(IR)=Y(IR)+AR
      Y(II)=Y(II)+AI
  310 CONTINUE
  300 CONTINUE
      MAX=ISTEP
      GO TO 240
  400 IF(ISIGN.EQ.1) RETURN
      NM2=N*2
      DO 450 I=1,NM2
      Y(I)=Y(I)/AN
  450 CONTINUE
      RETURN
      END
      END$
```

```
IPHASE T=00004 IS ON CR00002 USING 00005 BLKS R=0041

FTN4
C     ----------------------------------(SUBROUTINE PHASE)-------------
      SUBROUTINE PHASE(A,C,NDATA,B)
C.... THIS SUBROUTINE COMPUTES PHASE ANGLE B(I) OF FOURIER TRANSFORM
C     OF TIME HISTORY C(I). PHASE ANGLES RANGE BETWEEN -PI AND PI.
C.... DESCRIPTION OF PARAMETERS:
C       A(I)    --- COMPLEX FOURIER COEFFICIENT.
C       C(I)    --- TIME SERIES FOR PHASE ANGLE CALC.(INPUT).
C       NDATA   --- NUMBER OF TIME SERIES DATA.
C                   NDATA MUST BE A POWER OF TWO FOR FFT COMPUTATION.
C       B(I)    --- CALCULATED PHASE ANGLE (OUTPUT).
C
      DIMENSION A(1),B(1),C(1)
      CALL FFT(A,C,NDATA,-1)
      PI1=3.141593
      PI2=PI1*2.0
      PI12=PI1/2.0
      DO 100 I=1,NDATA
      II=I+I
      IR=II-1
      AR=A(IR)
      AI=A(II)
      IF(ABS(AR).LT.E-20) GO TO 30
      FAI0=AI/AR
      FAI=ATAN(FAI0)
      IF(FAI.LT.0.) GO TO 40
      IF(AR.LT.0.) FAI=FAI-PI1
      GO TO 50
   40 IF(AR.LT.0.) FAI=FAI+PI1
      GO TO 50
   30 IF(AI.GT.0.) FAI=PI12
      IF(AI.LT.0.) FAI=-PI12
   50 B(I)=FAI
  100 CONTINUE
      RETURN
      END
      END$
```

```
ISPECT T=00004 IS ON CR00002 USING 00009 BLKS R=0069

FTN4
C     --------------------------------(SUBROUTINE SPECT)------------
      SUBROUTINE SPECT(A,Y,NDATA,TT,MO)
      DIMENSION A(1),Y(1)
C.... THIS SUBROUTINE COMPUTES POWER SPECTRUM, FOURIER (AMPLITUDE)
C     SPECTRUM OR ABSOLUTE VALUE OF FOURIER COMPLEX COEFFICIENT.
C.... DESCRIPTION OF PARAMETERS:
C        Y(I)  --- ORIGINAL TIME HISTORY.
C        NDATA --- NUMBER OF DATA IN Y(I).
C        TT    --- DURATION (PERIOD) OF TIME HISTORY (SEC).
C        NFOLD --- NYQUIST FOLDING NUMBER
C        A(I)  --- COMPUTED RESULT.
C                  POWER OR FOURIER SPECTRUM, OR ABSOLUTE
C                  VALUE OF FOURIER COMPLEX COEFFICIENT.
C              IF  MO= 10, A(I)= POWER SPECTRUM FOR F (HZ).
C                      = 11, A(I)= POWER SPECTRUM FOR W (RAD. SEC).
C              IF  MO=  1, A(I)= FOURIER SPECTRUM
C              IF  MO=  2, A(I)= ABS. VALUE OF COMPLEX FOURIER COEF.
C                                FROM 1 TO NDATA
C
      NFOLD=NDATA/2+1
      PI2=3.141593*2.0
      CALL FFT(A,Y,NDATA,-1)
      ND1=NDATA+1
      ND2=NDATA+2
      IF(MO.EQ.1) GO TO 50
      IF(MO.EQ.2) GO TO 55
C
C.... TO COMPUTE POWER SPECTRUM.
      AR=A(1)**2+A(2)**2
      AR=AR*TT
      A(1)=AR
      IF(MO.GT.10) A(1)=AR/PI2
      DO 10 I=2,NFOLD-1
      II=I+I
      IR=II-1
      AR=A(IR)**2+A(II)**2
      AR=AR*2.0*TT
      A(I)=AR
      IF(MO.GT.10) A(I)=AR/PI2
   10 CONTINUE
      AR=A(ND1)**2+A(ND2)**2
      AR=AR*TT
      A(NFOLD)=AR
      IF(MO.GT.10) A(NFOLD)=AR/PI2
      GO TO 60
C
C.... TO COMPUTE FOURIER (AMPLITUDE) SPECTRUM.
   50 DO 40 I=1,NFOLD
      II=I+I
      IR=II-1
      AR=A(IR)**2+A(II)**2
      A(I)=TT*SQRT(AR)
   40 CONTINUE
      GO TO 60
C
C.... TO COMPUTE ABS. VALUE OF COMPLEX FOURIER COEFFICIENT.
   55 DO 45 I=1,NFOLD
      II=I+I
```

```
      IR=II-1
      AR=A(IR)**2+A(II)**2
      A(I)=SQRT(AR)
   45 CONTINUE
C
   60 DO 20 I=NFOLD+1,NDATA
      II=I-NFOLD
      IN=NFOLD-II
      A(I)=A(IN)
   20 CONTINUE
      RETURN
      END
      END$
```

```
ISTATC T=00004 IS ON CR00002 USING 00005 BLKS R=0032

FTN4
C     --------------------------------(SUBROUTINE STATC)------------
      SUBROUTINE STATC(Y,N,EY,VAR,STDV,YMAX,YMIN)
      DIMENSION Y(1)
C.... THIS SUBROUTINE COMPUTES BASIC STATISTICS ON DATA Y(I)
C     SUCH AS OVERALL (TEMPORAL) MEAN, VARIANCE, STANDARD
C     DEVIATION, MAXIMUM AND MINIMUM VALUES.
C.... DESCRIPTION OF PARAMETERS:
C       Y(I) --- ORIGINAL DATA ARRAY.
C       N    --- NUMBER OF DATA.
C       EY   --- OVERALL (TEMPORAL) MEAN OF Y(I).
C       VAR  --- VARIANCE OF Y(I).
C       STDV --- STANDARD DEVIATION OF Y(I).
C       YMAX --- MAXIMUM VALUE IN Y(I).
C       YMIN --- MINIMUM VALUE IN Y(I).
C
      EY=0.0
      VAR=0.0
      YMAX=-E30
      YMIN=E30
      DO 10 I=1,N
      YI=Y(I)
      EY=EY+YI
      IF(YI.GT.YMAX) YMAX=YI
      IF(YI.LT.YMIN) YMIN=YI
   10 CONTINUE
      EY=EY/FLOAT(N)
      DO 20 I=1,N
      YI=Y(I)-EY
      VAR=VAR+YI*YI
   20 CONTINUE
      VAR=VAR/FLOAT(N-1)
      STDV=SQRT(VAR)
      RETURN
      END
      END$
```

```
IHLBRT T=00004 IS ON CR00002 USING 00005 BLKS R=0035

FTN4
C     ---------------------------------(SUBROUTINE HLBRT)-------------
      SUBROUTINE HLBRT(A,X0,XHIL,N,MO)
      DIMENSION A(1),X0(1),XHIL(1)
C.... THIS SUBROUTINE COMPUTES HILBERT TRANSFORM OF TIME HISTORY
C     X0(T) FOR SIMULATION OF DATA-BASED NONSTATIONARY RANDOM
C     PROCESSES.
C.... DESCRIPTION OF PARAMETERS:
C       X0   --- ORIGINAL TIME HISTORY DATA (REAL ARRAY).
C       N    --- NUMBER OF DATA ON X0(I).
C       A    --- ARRAY FOR FFT COMPUTATION.
C                ARRAY SIZE MUST BE 2*N.
C       XHIL -- HILBERT TRANSFORM OF X0(T).
C       MO   --- CONTROL PARAMETER.
C                IF MO=0 --- FFT COMPUTATION ON X0(T) IS NEEDED.
C                IF MO=1 --- NO NEED ON FFT COMPUTATION ON X0(T).
C
      IF(MO.EQ.1) GO TO 10
      CALL FFT(A,X0,N,-1)
   10 NFOLD=N/2+1
      DO 100 I=NFOLD+1,N
      II=I+I
      IR=II-1
      A(IR)=-A(IR)
      A(II)=-A(II)
  100 CONTINUE
      CALL FFT(A,X0,N,1)
      DO 200 I=1,N
      II=I+I
      XHIL(I)=A(II)
  200 CONTINUE
      RETURN
      END
      END$
```

```
IJNORM T=00004 IS ON CR00002 USING 00011 BLKS R=0075

FTN4
C     ---------------------------------(SUBROUTINE JNORM)------------
      SUBROUTINE JNORM(IX1,IY1,IX2,IY2,STDV1,STDV2,AM1,CORRL,
     *RAN1,RAN2,MO,MDIM)
C.... THIS SUBROUTINE GENERATES EITHER UNIVARIATE OR JOINTLY
C     CORRELATED BIVARIATE GAUSSIAN RANDOM NUMBER.
C.... DESCRIPTION OF PARAMETERS:
C        RAN1   -- GENERATED GAUSSIAN RANDOM NUMBER WITH
C                  MEAN AM1 AND STANDARD DEVIATION STDV1.
C        RAN2   -- GENERATED THE OTHER GAUSSIAN RANDOM NUMBER
C                  WITH STANDARD DEVIATION STDV2.  RAN2 IS
C                  CORRELATED TO RAN1 THROUGH CORRELATION
C                  COEFFICIENT CORRL.
C        CORRL -- CORRELATION COEFFICIENT BETWEEN RAN1 AND RAN2.
C                  IF ABS(CORRL)=1 -- RAN2=CORRL*RAN1
C                  IF CORRL=0 ------- RAN1 AND RAN2 ARE INDEPENDENT.
C        MO    --- CONTROL PARAMETER ON THE OUTPUT RANDOM NUMBERS.
C                  IF MO > 0 -- -PI< RAN1,RAN2 < PI
C                  IF MO < 0 -- -INFINITY < RAN1,RAN2 < +INFINITY
C        MDIM --- DIMENSION CONTROL PARAMETER.
C                  IF MDIM=1 -- UNIVARIATE RANDOM NUMBER. IN THIS
C                               CASE, ONLY RAN1 IS TO BE GENERATED.
C                  IF MDIM=2 -- CORRELATED BIVARIATE RANDOM NUMBER.
C        IX1,IY1,IX2,IY2 -- ENTRY FOR SUBROUTINE RANDU(IX,IY,RAN)
C
      IF(NDIM.EQ.0) NDIM=1
C
C.... GENERATION OF FIRST GAUSSIAN RANDOM NUMBER RAN1.
      CALL GAUSS(IX1,IY1,STDV1,AM1,RAN1)
      IF(MDIM.EQ.1) RETURN
      IF(ABS(CORRL).NE.1.0) GO TO 5
      RAN2=RAN1*CORRL*STDV2/STDV1
      GO TO 6
C
C.... GENERATION OF SECOND GAUSSIAN RANDOM NUMBER RAN2.
    5 AM2=CORRL*STDV2*RAN1/STDV1
C
C.... FOR THE CASE OF CORRL BEING VERY CLOSE TO UNITY.
      DELTA=1.0-ABS(CORRL)
      IF(DELTA.GE.0.001) GO TO 7
      STDV=STDV2*SQRT(2.0*DELTA)
      GO TO 8
    7 STDV=STDV2*SQRT(1.0-CORRL*CORRL)
    8 CALL GAUSS(IX2,IY2,STDV,AM2,RAN2)
    6 IF(MO.LT.0) RETURN
C
C.... REARRANGEMENT OF GENERATED GAUSSIAN RANDOM NUMBERS SO AS
C     TO RANGE BETWEEN -PI AND PI.
      CALL REARR(RAN1)
      CALL REARR(RAN2)
      RETURN
      END
C
C     ---------------------------------(SUBROUTINE GAUSS)------------
      SUBROUTINE GAUSS(IX,IY,STDV,AMEAN,RAN)
C.... THIS SUBROUTINE GENERATES A GAUSSIAN RANDOM NUMBER WITH
C     MEAN AMEAN AND STANDARD DEVIATION STDV.
C
      A=0.0
```

```
      DO 10 I=1,12
      CALL RANDU(IX,IY,Y)
   10 A=A+Y
      RAN=(A-6.0)*STDV+AMEAN
      RETURN
      END
C
C     -------------------------------(SUBROUTINE REARR)------------
      SUBROUTINE REARR(RAN)
C.... THIS SUBROUTINE REPLACE RAN SO AS TO RANGE BETWEEN -PI
C     AND PI WITH THE CONSIDERATION OF RAN BEING A PERIODIC
C     FUNCTION OF PERIOD 2*PI.
C
      PI=3.141593
      PIM=-PI
      PI2=2.0*PI
      IF(RAN.LT.0.0) GO TO 10
   20 IF(RAN.LE.PI) GO TO 30
      RAN=RAN-PI2
      GO TO 20
   10 IF(RAN.GE.PIM) GO TO 30
      RAN=RAN+PI2
      GO TO 10
   30 RETURN
      END
      END$
```

```
IARNGE T=00004 IS ON CR00002 USING 00011 BLKS R=0075

FTN4
C     --------------------------------(SUBROUTINE ARNGE)-------------
      SUBROUTINE ARNGE(Y,N,MO,MJD)
      DIMENSION Y(1)
C.... THIS SUBROUTINE ARRANGES ARRAY DATA Y(I) SYMMETRICALLY
C     TO PRODUCE MIRROR IMAGE AROUND ORIGIN OR AROUND NFOLD(=N/2+1).
C.... DESCRIPTION OF PARAMETERS:
C       Y    -- INPUT ARRAY TO BE ARRANGED SYMMETRICALLY
C               AT LEAST NFOLD DATA MUST BE DEFINED
C               BEFORE CALLING.
C               Y COULD BE EATHER REAL OR COMPLEX.
C               COMPLEX DATA SHOULD BE IN ORDER OF
C               (REAL1),(IMAG1),(REAL2),(IMAG2),...
C       N    -- ARRAY SIZE OF Y
C       MJD -- CLASSIFICATION PARAMETER OF DATA Y
C         IF MJD>0, FOR REAL DATA ONLY
C            MJD<0, FOR COMPLEX DATA
C                   FOR COMPLEX DATA, ONLY MIRROR SYMMETRY
C                   AROUND NFOLD IS CONSIDERED.
C                   FOR N > NFOLD, SIGN OF IMAGINARY
C                   PART IS REVERSED.
C                   "MO" SHALL BE DISREGARDED IN THIS CASE.
C       MO   -- CONTROL PARAMETER FOR ARRANGEMENT
C         IF MO= 10, SYMMETRY AROUND ORIGIN WITH SAME SIGN
C               =-10, SYMMETRY AROUND ORIGIN WITH REVERSE SIGN
C         IF MO= 11, SYMMETRY AROUND NFOLD WITH SAME SIGN
C               =-11, SYMMETRY AROUND NFOLD WITH REVERSED SIGN
C
      N1=N+1
      N2=N+2
      ND2=N/2
      NFOLD=ND2+1
      NFOLD1=NFOLD+1
      IF(MJD.LT.0) GO TO 50
C     ---------------------------------------------
C.... FOR REAL DATA ARRAY ONLY
C.... SYMMETRY AROUND NFOLD WITH SAME SIGN
      IF(IABS(MO).EQ.10) GO TO 10
      DO 100 I=NFOLD1,N1
      IN=I-NFOLD
      INR=NFOLD-IN
      Y(I)=Y(INR)
C
C.... SYMMETRY AROUND NFOLD WITH REVERSED SIGN
      IF(MO.EQ.-11) Y(I)=-Y(INR)
  100 CONTINUE
      RETURN
C
C.... SYMMETRY AROUND ORIGIN WITH SAME SIGN
   10 YNFOLD=Y(NFOLD)
      DO 200 I=1,ND2
      IP=I+ND2
      Y(IP)=Y(I)
  200 CONTINUE
      Y(N1)=YNFOLD
      DO 205 I=1,ND2
      IN=N2-I
      Y(I)=Y(IN)
C
```

```
C.... SYMMETRY AROUND ORIGIN WITH REVERSED SIGN
      IF(MO.EQ.-10) Y(I)=-Y(IN)
  205 CONTINUE
      RETURN
C     ---------------------------------------------
C.... FOR COMPLEX DATA ARRAY
   50 NFOLD4=4*NFOLD
      DO 300 I=NFOLD1,N
      II=I+I
      IR=II-1
      INI=NFOLD4-II
      INR=INI-1
      Y(IR)=Y(INR)
      Y(II)=-Y(INI)
  300 CONTINUE
      RETURN
      END
      END$
```

ISTUKO T=00004 IS ON CR00002 USING 00019 BLKS R=0106

```
FTN4
          SUBROUTINE STUKO(X,Y,NPT,NX,XMIN,XMAX,NY,YMIN,YMAX,
        *  NP,MO,IP,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
          DIMENSION X(1),Y(1),NAME(3),NA(20),AA(5)
          EQUIVALENCE (AA,NA(11))
C
C.... THIS SUBROUTINE IS USED TO PLOT Y(I)-X(I) RELATIONSHIP
C     IN ANY LOCATION AND SIZE AS DESIRED ON A SHEET.
C     THIS IS A MULTI-FUNCTIONAL SUBROUTINE.
C
C.... DATE 06/27/1978
C.... BY DR. H. ISHIKAWA AND D. LI, COLUMBIA UNIVERSITY.
C
C.... DESCRIPTION OF PARAMETERS:
C     X       -- COORDINATE ARRAY.
C     Y       -- ABSCISSA ARRAY.
C     NPT     -- NUMBER OF POINTS TO BE PLOTTED.
C     YMAX    -- MAXIMUM VALUE ON Y-AXIS.
C     YMIN    -- MINIMUM VALUE ON Y-AXIS.
C                YMIN AND YMAX ARE EITHER SUPPLIED BY USER
C                OR SEARCHED BY THE PROGRAM (SEE ALSO NY).
C     NY      -- CONTROL CODE ON YMAX AND YMIN.
C                IF NY> 0 --- USER SUPPLY FOR YMAX, YMIN.
C                   NY= 0 --- DEFAULT OPTION.
C                   NY< 0 --- RETURN TO THE MAIN PROGRAM.
C                             PROGRAM SEARCHES YMAX AND YMIN.
C     XMAX    -- MAXIMUM ORDINATE VALUE ON X-AXIS.
C     XMIN    -- MINIMUM ORDINATE VALUE ON X-AXIS.
C                X-AXIS IS BOUNDED FROM XMIN TO XMAX.
C     NX      -- CONTROL CODE ON XMIN AND XMAX.
C                IF NX= 0 --- PROGRAM SETS AUTOMATICALLY
C                             XMIN=X(1) AND XMAX=X(N).
C                   NX> 0 --- USER SUPPLY FOR XMIN AND XMAX.
C                   NX< 0 --- ACCORDING TO THE USER SUPPLIED
C                             XMIN AND XMAX, PROGRAM AUTOMATICALLY
C                             COMPUTES ORDINATE ARRAY X(I) WITH
C                             DX=(XMAX-XMIN)/N.
C     NP      -- CONTROL CODE ON NO. OF FRAMES OR FIGURES.
C                THIS CONSISTS OF 5 DIGITS (IJKLM) AS FOLLOWS.
C                I (=NDIM) -- NO. OF COLUMNS (UP TO 2).
C                JK (=NP1) -- NO. OF FRAMES OR GRAPHS IN LEFT COLUMN.
C                LM (=NP2) -- NO. OF FRAMES OR GRAPHS IN RIGHT COLUMN.
C                NPTTL=NP1+NP2 -- TOTAL NO. OF GRAPHS TO BE DRAWN.
C                DEFAULT OPTION -- IF ONE COLUMN, NP CAN BE ONLY
C                                  TWO DIGITS (LM). IN THIS CASE,
C                                  NPTTL=(LM).
C     MO      -- CONTROL CODE ON THE WAY OF PLOTTING NP GRAPHS.
C                IF MO= 0 --- NPTTL PLOTS ON ONE FRAME.
C                   MO= 1 --- NPTTL PLOTS ON NPTTL FRAMES,
C                             WITH NPTTL LABELS.
C                   MO=-1 --- NPTTL PLOTS ON NPTTL FRAMES,
C                             WITH ONLY ONE LABEL.
C     JDCXY -- LABEL DIGITS CONTROL CODE ON X- AND Y-AXIS.
C                THIS CONSISTS OF 3 DIGITS (IJK) AS FOLLOWS.
C                I -- LABEL CONTROL CODE.
C                     IF I= 0         -- LABEL IS PRINTED OUT.
C                     IF OTHERWISE -- NO LABEL IS PRINTED OUT.
C                J -- NO. OF DECIMAL POINTS ON X-AXIS.
C                     FORMAT NOTATION -- F7.I (I=0,1,...,5)
```

```
C            K -- NO. OF DECIMAL POINTS ON Y-AXIS.
C      MINXO,MAXXO,MINYO,MAXYO
C            -- PARAMETER TO DEFINE THE LOCATION OF THE FIGURE.
C               POSSIBLE LOCATION OF X-AXIX --- 1 TO 1024.
C               POSSIBLE LOCATION OF Y-AXIS --- 1 TO  780.
C               DEFAULT VALUES: MINXO= 40   -- MAXXO=1000
C                               MINYO= 50   -- MAXYO= 750
C      LU    -- TEXTRONIX TERMINAL LOGICAL UNIT NUMBER.
C      IP    -- FRAME NUMBER ON THE PICTURE OR
C               GRAPH LINE NUMBER ON THE PICTURE FRAME.
C               FIRST TIME CALL IP=1, LAST TIME CALL IP=NPTTL.
C
      DATA NAME/2HCH,2HIE,2H  /
      NDIM=NP/10000
      NP1=NP/100-100*NDIM
      NP2=NP-100*NP1-10000*NDIM
      NPTTL=NP1+NP2
      IF(NDIM.GT.1) GO TO 5
      IF(NP1.EQ.0) NP1=NP2
      IF(NP1.EQ.0) NP1=1
      NPTTL=NP1
    5 IF(NPTTL.LE.0) NPTTL=1
      IF(IP.EQ.1) CALL EXEC(4,NPTTL,ISTRK,IDISC,ISECT)
      NSTRK=ISTRK+IP-1
      NA(1)=NPT
      NA(2)=NX
      NA(3)=NY
      NA(4)=MO
      NA(5)=JDCXY
      NA(6)=MINXO
      NA(7)=MAXXO
      NA(8)=MINYO
      NA(9)=MAXYO
      AA(1)=XMIN
      AA(2)=XMAX
      AA(3)=YMIN
      AA(4)=YMAX
      NPT2=2*NPT
      CALL EXEC (2,102B,NA,20,NSTRK,0)
      CALL EXEC (2,102B,Y,NPT2,NSTRK,1)
      IF (NX .LT. 0) GO TO 100
      CALL EXEC (2,102B,X,NPT2,NSTRK,24)
  100 IF (IP .LT. NPTTL) GO TO 200
      CALL EXEC (9,NAME,ISTRK,IDISC,ISECT,LU,NP)
      CALL EXEC (5,-1)
  200 RETURN
      END
      END$
```

```
IYOKO   T=00004 IS ON CR00002 USING 00045 BLKS R=0315

FTN4
      SUBROUTINE YOKO(X,Y,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
     *  NP,MO,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
      DIMENSION X(1),Y(1)
C
C.... THIS SUBROUTINE IS USED TO PLOT Y(I)-X(I) RELATIONSHIP
C     IN ANY LOCATION AND SIZE AS DESIRED ON A SHEET.
C     THIS IS A MULTI-FUNCTIONAL SUBROUTINE.
C
C.... DATE 06/27/1978
C.... BY DR. H. ISHIKAWA AND D. LI, COLUMBIA UNIVERSITY.
C
C.... DESCRIPTION OF PARAMETERS:
C     X      -- COORDINATE ARRAY.
C     Y      -- ABSCISSA ARRAY.
C     N      -- NUMBER OF POINTS TO BE PLOTTED.
C     YMAX   -- MAXIMUM VALUE ON Y-AXIS.
C     YMIN   -- MINIMUM VALUE ON Y-AXIS.
C               YMIN AND YMAX ARE EITHER SUPPLIED BY USER
C               OR SEARCHED BY THE PROGRAM (SEE ALSO NY).
C     NY     -- CONTROL CODE ON YMAX AND YMIN.
C               IF NY= 1 --- USER SUPPLY FOR YMAX, YMIN.
C                  NY= 0 --- DEFAULT OPTION.
C                            PROGRAM SEARCHES YMAX AND YMIN.
C     XMAX   -- MAXIMUM ORDINATE VALUE ON X-AXIS.
C     XMIN   -- MINIMUM ORDINATE VALUE ON X-AXIS.
C               X-AXIS IS BOUNDED FROM XMIN TO XMAX.
C     NX     -- CONTROL CODE ON XMIN AND XMAX.
C               IF NX= 0 --- PROGRAM SETS AUTOMATICALLY
C                            XMIN=X(1) AND XMAX=X(N).
C                  NX= 1 --- USER SUPPLY FOR XMIN AND XMAX.
C                  NX=-1 --- ACCORDING TO THE USER SUPPLIED
C                            XMIN AND XMAX, PROGRAM AUTOMATICALLY
C                            COMPUTES ORDINATE ARRAY X(I) WITH
C                            DX=(XMAX-XMIN)/N.
C     NP     -- CONTROL CODE ON NO. OF FRAMES OR FIGURES.
C               THIS CONSISTS OF 5 DIGITS (IJKLM) AS FOLLOWS.
C               I (=NDIM) -- NO. OF COLUMNS (UP TO 2).
C               JK (=NP1) -- NO. OF FRAMES OR GRAPHS IN LEFT COLUMN.
C               LM (=NP2) -- NO. OF FRAMES OR GRAPHS IN RIGHT COLUMN.
C               NPTTL=NP1+NP2 -- TOTAL NO. OF GRAPHS TO BE DRAWN.
C               DEFAULT OPTION -- IF ONE COLUMN, NP CAN BE ONLY
C                                 TWO DIGITS (LM). IN THIS CASE,
C                                 NPTTL=(LM).
C     MO     -- CONTROL CODE ON THE WAY OF PLOTTING NP GRAPHS.
C               IF MO= 0 --- NPTTL PLOTS ON ONE FRAME.
C                  MO= 1 --- NPTTL PLOTS ON NPTTL FRAMES,
C                            WITH NPTTL LABELS.
C                  MO=-1 --- NPTTL PLOTS ON NPTTL FRAMES,
C                            WITH ONLY ONE LABEL.
C     JDCXY -- LABEL DIGITS CONTROL CODE ON X- AND Y-AXIS.
C               THIS CONSISTS OF 3 DIGITS (IJK) AS FOLLOWS.
C               I -- LABEL CONTROL CODE.
C                    IF I= 0        -- LABEL IS PRINTED OUT.
C                    IF OTHERWISE -- NO LABEL IS PRINTED OUT.
C               J -- NO. OF DECIMAL POINTS ON X-AXIS.
C                    FORMAT NOTATION -- F7.J (J=0,1,...,5)
C               K -- NO. OF DECIMAL POINTS ON Y-AXIS.
C     MINX0,MAXX0,MINY0,MAXY0
```

```
C            -- PARAMETER TO DEFINE THE LOCATION OF THE FIGURE.
C               POSSIBLE LOCATION OF X-AXIX --- 1 TO 1024.
C               POSSIBLE LOCATION OF Y-AXIS --- 1 TO  780.
C               DEFAULT VALUES: MINX0= 10   -- MAXX0=1020
C                                MINY0= 40   -- MAXY0= 750
C        LU    -- TEXTRONIX TERMINAL LOGICAL UNIT NUMBER.
C        IPLOT -- FRAME NUMBER ON THE PICTURE OR
C                 GRAPH LINE NUMBER ON THE PICTURE FRAME.
C
      DATA IPLOT/1/
      DATA IPLT1/1/
C.... TO SET UP APPROPRIATE VALUE FOR CONTROL.
      IF (LU .NE. 7) LU=1
      NDIM=NP/10000
      NP1=NP/100-NDIM*100
      NP2=NP-100*NP1-10000*NDIM
      NPTTL=NP1+NP2
      IF(NDIM.GT.1) GO TO 5
      NDIM=1
      IF(NP1.EQ.0) NP1=NP2
      IF(NP1.EQ.0) NP1=1
      NPTTL=NP1
    5 IF(NPTTL.LE.0) NPTTL=1
      JDC=JDCXY/100
      JDCX=JDCXY/10-10*JDC
      JDCY=JDCXY-10*JDCX-100*JDC
C
C     TEST IF IT IS THE 1ST GRAPH ON THE SAME PICTURE,IF NOT
C     GO TO PLOT DATA IMMEDIATELY
      IF(IPLOT.NE.1) GO TO 190
      CALL INITT(960)
  190 IF(MO .NE. 0) GO TO 14
      IF(IPLT1.NE.1) GO TO 85
   14 IF(NY.GT.0) GO TO 20
      YMAX=-1.0E20
      YMIN=1.0E+20
      DO 15 I=1,N
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
   15 CONTINUE
      IF(YMIN .GE. 0. .OR. YMAX .LE. 0.) GO TO 20
      IF(YMAX.GT.ABS(YMIN)) YMIN=-YMAX
      YMAX=ABS(YMIN)
   20 IF(NX .NE. 0) GO TO 25
      XMIN=X(1)
      XMAX=X(N)
   25 DX=(XMAX-XMIN)/10.
      DEX=(XMAX-XMIN)/FLOAT(N-1)
      DY=(YMAX-YMIN)/10.
      IF(MAXX0.NE.0 .AND. MAXY0.NE.0) GO TO 27
      MINX0=40
      MAXX0=1000
      MINY0=50
      MAXY0=750
   27 MLNGX=(MAXX0-MINX0)/NDIM
      MLNGY=MAXY0-MINY0
      NLR=1
      NDIV=NP1
      IF(IPLOT.LE.NP1) GO TO 28
      NLR=2
      NDIV=NP2
```

```
   28 IF(MO.EQ.0) NDIV=1
      MCHRX=13
      MCHRY=20
      MSEG=5
      MFRMT=7
      MSPCXF=(MFRMT+1)*MCHRX
      MSPCXL=(MFRMT-4)*MCHRX
      MLINE1=1
      MLINE2=5
      MLINEY=MLINE2
      IF(MO .GT. 0) GO TO 10
      IF(NDIV .GT. 3) GO TO 11
      MLINE1=2
   11 MLINEY=MLINE1
C     MLNGY=MAXY0-MINY0-(MLINE2-MLINE1)
   10 MSPCYF=MLINEY*MCHRY
      MSPCYL=0
      MDELTY=MLNGY/NDIV
      MINX=MINX0+(NLR-1)*MLNGX
      MAXX=MINX+MLNGX
      MAXY=MAXY0-(IPLT1-1)*MDELTY
      MINY=MAXY-MDELTY
      CALL TWIND(MINX,MAXX,MINY,MAXY)
      XINCR=(XMAX-XMIN)/FLOAT(MLNGX-MSPCXF-MSPCXL)
      XFST=XMIN-XINCR*MSPCXF
      XLST=XMAX+XINCR*MSPCXL
      YINCR=(YMAX-YMIN)/FLOAT(MDELTY-MSPCYF-MSPCYL)
      YFST=YMIN-YINCR*MSPCYF
      IF(MO.LT.0 .AND. NDIV.GT.3) YFST=YMIN-1.5*YINCR*MSPCYF
      YLST=YMAX+YINCR*MSPCYL
      YSEG=YINCR*MSEG
      XSEG=XINCR*MSEG
      CALL DWIND (XFST,XLST,YFST,YLST)
      CALL MOVEA (XMIN,YMAX)
      CALL DRAWA (XMIN,YMIN)
      CALL DRAWA (XMAX,YMIN)
      CALL DRAWA (XMAX,YMAX)
      CALL DRAWA (XMIN,YMAX)
      IF (YMIN .GE. 0. .OR. YMAX .LE. 0.) GO TO 59
      CALL MOVEA (XMIN,0.)
      CALL DRAWA (XMAX,0.)
   59 IF(ISSW(10).GE.0) GO TO 60
      CALL MOVEA(0.,YMIN)
      CALL DRAWA(0.,YMAX)
C     DRAW TIC MARKS ON X-AXIS
   60 DO 30 I=1,9
   30 CALL SEGMT (0,1,XMIN+I*DX,YMIN+YSEG)
C     DRAW TIC MARKS ON Y-AXIS
      DO 40 I=1,9
   40 CALL SEGMT (1,1,XMIN+XSEG,YMIN+I*DY)
C     LABEL X  AXES
      IF(JDC.NE.0) GO TO 85
      SDY=1.5*YINCR*MCHRY
      IF (MO .GT. 0 .OR. NDIV .EQ. 1) GO TO 150
      IF (IPLT1 .NE. NDIV) GO TO 82
  150 DO 70 I=1,6
      XX=XMIN+FLOAT(I-1)*2.0*DX
      XXD=XX-XINCR*FLOAT(MCHRX)*(FLOAT(MFRMT)/2.0-FLOAT(JDCX-3))
      IF(ISSW(11).GE.0) GO TO 400
      CALL MOVEA(XX,YMIN)
      CALL DRAWA(XX,YMAX)
```

```
      400 CALL MOVEA (XXD,YMIN-SDY)
          CALL ANMOD
          CALL JWRTE(JDCX,XX,LU)
       70 CONTINUE
       82 IF(MO.EQ.0 .OR. NDIV.EQ.1) GO TO 131
          IF (ABS(YMIN) .EQ. ABS(YMAX)) GO TO 132
 C        WRITE 5 LABELS IF YMAX AND YMIN ARE NOT EQUAL.
 C        IF NDIV > 4 --- WRITE 3 LABELS FOR EASY DISPLAY.
          IF(NDIV .GT. 4) GO TO 132
          DO 140 I=1,6
          YY=YMIN+FLOAT(I-1)*2.*DY
          YYD=YY-YINCR*FLOAT(MCHRY)/2.0
          IF(ISSW(12).GE.0) GO TO 410
          CALL MOVEA(XMIN,YY)
          CALL DRAWA(XMAX,YY)
      410 CALL MOVEA(XFST,YYD)
          CALL ANMOD
          CALL JWRTE(JDCY,YY,LU)
      140 CONTINUE
          GO TO 85
 C        WRITE 3 LABELS IF YMAX=-YMIN
 C        HOWEVER, WHEN ONLY ONE FRAME, WRITE 11 LABELS.
 C    130 IF(MO.EQ.0 .OR. NDIV.EQ.1) GO TO 131
      132 DO 80 I=1,3
          YY=YMIN+(I-1)*5*DY
          YYD=YY-YINCR*FLOAT(MCHRY)/2.0
          CALL MOVEA (XFST,YYD)
          CALL ANMOD
          CALL JWRTE(JDCY,YY,LU)
       80 CONTINUE
          GO TO 85
 C        WRITE 11 LABELS.
      131 DO 81 I=1,11
          YY=YMIN+(I-1)*DY
          YYD=YY-YINCR*FLOAT(MCHRY)/2.0
          IF(ISSW(12).GE.0) GO TO 420
          CALL MOVEA(XMIN,YY)
          CALL DRAWA(XMAX,YY)
      420 CALL MOVEA(XFST,YYD)
          CALL ANMOD
          CALL JWRTE(JDCY,YY,LU)
       81 CONTINUE
 C        DRAW VECTORS BETWEEN TIME POINTS
       85 IF (NX .GE. 0) GO TO 100
          CALL MOVEA(XMIN,Y(1))
          XM=XMIN-DEX
          DO 50 I=1,N
          XMIDX=XM+I*DEX
          CALL DRAWA (XMIDX,Y(I))
       50 CONTINUE
          GO TO 110
      100 CALL MOVEA(X(1),Y(1))
          DO 120 I=2,N
          IF(X(I).LT.XMIN .OR. Y(I).LT.YMIN) GO TO 121
          IF(X(I).GT.XMAX .OR. Y(I).GT.YMAX) GO TO 121
          CALL DRAWA(X(I),Y(I))
          GO TO 120
      121 DIVX=(X(I)-X(I-1))/10.0
          DIVY=(Y(I)-Y(I-1))/10.0
          DO 122 J=1,10
          DXJ=X(I-1)+DIVX*FLOAT(J)
```

```
      DYJ=Y(I-1)+DIVY*FLOAT(J)
      IF(DXJ.LT.XMIN .OR. DYJ.LT.YMIN) GO TO 123
      IF(DXJ.GT.XMAX .OR. DYJ.GT.YMAX) GO TO 123
      CALL DRAWA(DXJ,DYJ)
      GO TO 122
  123 CALL MOVEA(DXJ,DYJ)
  122 CONTINUE
  120 CONTINUE
  110 CONTINUE
      CALL FINIT (0,767)
      IF (IPLOT .EQ. NPTTL) IPLT1=0
      IF (IPLOT .EQ. NPTTL) IPLOT=0
      IPLOT=IPLOT+1
      IF(NDIM.EQ.2 .AND. IPLT1.EQ.NP1) IPLT1=0
      IPLT1=IPLT1+1
      RETURN
      END
      SUBROUTINE JWRTE(JDCXY,XY,LU)
C
C.... TO SELECT NUMBER OF DIGITS BELOW DECIMAL POINT.
C
      IL=2H+
      IF(JDCXY.NE.0) GO TO 10
      IXY=IFIX(XY)
      WRITE(LU,100) IXY,IL
      RETURN
   10 IF(JDCXY.EQ.1) WRITE(LU,101) XY,IL
      IF(JDCXY.EQ.2) WRITE(LU,102) XY,IL
      IF(JDCXY.EQ.3) WRITE(LU,103) XY,IL
      IF(JDCXY.EQ.4) WRITE(LU,104) XY,IL
      IF(JDCXY.EQ.5) WRITE(LU,105) XY,IL
  100 FORMAT(I7,A1)
  101 FORMAT(F7.1,A1)
  102 FORMAT(F7.2,A1)
  103 FORMAT(F7.3,A1)
  104 FORMAT(F7.4,A1)
  105 FORMAT(F7.5,A1)
      RETURN
      END
      SUBROUTINE SEGMT (NDIR,NSIZE,X,Y)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
C      THIS ROUTINE WILL DRAW TIC MARK AT LOCATION (X,Y),
C      (X,Y) BEING VIRTUAL CO-ORDINATES ON SCREEN
C      NDIR    --- 0 DRAW VERTICAL TIC MARK
C              --- 1 DRAW HORIZONTAL TIC MARK
C      NSIZE   --- 1 SIZE OF 10 RASTER UNITS
C              --- 2 SIZE OF 20 RASTER UNITS
C
C      CONVERT TO SCREEN CO-ORDINATES
      CALL LVLCH
      CALL V2ST (1,X,Y,IX,IY)
C      SKIP IF POINT COMPLETELY OUTSIDE WINDOW
      IF(KGNFLG.EQ.1) GO TO 60
      CALL VECMD
      IF(NDIR.EQ.1) GO TO 30
```

```
      CALL XYCNT (IX,IY-NSIZE*5)
      GO TO 40
   30 CALL XYCNT (IX-NSIZE*5,IY)
   40 IF(KKMODE.NE.1) CALL VECMD
      IF(KMOVEF.EQ.1) CALL XYCNT(KBEAMX,KBEAMY)
      IF(NDIR.EQ.1) GO TO 50
      CALL XYCNT (IX,IY+NSIZE*5)
      GO TO 60
   50 CALL XYCNT (IX+NSIZE*5,IY)
   60 RETURN
      END
      END$
```

```
ICHIE  T=00004 IS ON CR00002 USING 00009 BLKS R=0063

FTN4
            PROGRAM CHIE,3,10
            DIMENSION X(1400),Y(1400),NA(20),AA(5),IPARM(5)
            COMMON LARY(100)
C           EQUIVALENCE (AA,NA(11)),(X,Y(1401))
            EQUIVALENCE (AA,NA(11))
C           THIS PROGRAM SCHEDULED BY SUBROUTINE STUKO,
C           THE MAXIMUM DATA POINT TO BE PLOTED IS 1400 POINTS.
C           WHEN NX = -1 MAXIMUM NPT=2800 POINTS FOR Y ARRAY ONLY.
C           PARAMETER TEANSFER FROM SUBROUTINE CHIE AS FOLLOW
C           ISTRK - STARTING DISC TRACK FOR DATA STORAGE.
C           IDISC - DISC LU NUMBER.
C           ISECT - NUMBER OF SECTOR FOR ONE TRACK.
C           LU    - LOGIC NUMBER FOR TEKTRONIX TERMINAL.
C           NP    - CONSISTS OF 5 DIGITS (IJKLM) AS FOLLOWS.
C                   I (=NDIM) -- NO. OF COLUMNS (UP TO 2).
C                   JK (=NP1) -- NO. OF FRAMES OR GRAPHS IN LEFT COLUMN.
C                   LM (=NP2) -- NO. OF FRAMES OR GRAPHS IN RIGHT COLUMN.
C                   NPTTL=NP1+NP2 -- TOTAL NO. OF GRAPHS TO BE DRAWN.
C
C.... DATE 06/26/1978
C.... BY DR. H. ISHIKAWA AND D. LI, COLUMBIA UNIVERSITY.
C
            CALL RMPAR(IPARM)
            ISTRK=IPARM(1)
            IDISC=IPARM(2)
            ISECT=IPARM(3)
            LU   =IPARM(4)
            NP   =IPARM(5)
            NDIM=NP/10000
            NP1=NP/100-100*NDIM
            NP2=NP-100*NP1-10000*NDIM
            NPTTL=NP1+NP2
            IF(NDIM.GT.1) GO TO 5
            IF(NP1.EQ.0) NP1=NP2
            IF(NP1.EQ.0) NP1=1
            NPTTL=NP1
          5 IF(NPTTL.LE.0) NPTTL=1
            DO 200 IPLOT=1,NPTTL
            NSTRK=ISTRK+IPLOT-1
            CALL EXEC (1,1028,NA,20,NSTRK,0)
            NPT=NA(1)
            NX=NA(2)
            NY=NA(3)
            MO=NA(4)
            JDCXY=NA(5)
            MINX0=NA(6)
            MAXX0=NA(7)
            MINY0=NA(8)
            MAXY0=NA(9)
            XMIN=AA(1)
            XMAX=AA(2)
            YMIN=AA(3)
            YMAX=AA(4)
            NPT2=2*NPT
            CALL EXEC (1,1028,Y,NPT2,NSTRK,1)
            IF (NX .LT. 0) GO TO 100
            CALL EXEC (1,1028,X,NPT2,NSTRK,24)
        100 CALL YOKO(X,Y,NPT,NX,XMIN,XMAX,NY,YMIN,YMAX,
```

```
     *   NP,MO,JDCXY,MINXO,MAXXO,MINYO,MAXYO)
 200 CONTINUE
     CALL EXEC (6)
     END
     END$
```

```
TTYLB  T*00004 IS ON CR00002 USING 00111 BLKS R*0800

FTN4
            SUBROUTINE LVLCH
            COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
          1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
          2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
          3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
          4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
          5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
          6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
            IF(KGRAFL.NE.0) GO TO 10
            CALL REVCO (KBEAMX,KBEAMY,TREALX,TREALY)
            TIMAGX=TREALX
            TIMAGY=TREALY
            KGRAFL=1
         10 RETURN
            END
            SUBROUTINE MOVEA (X,Y)
            COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
          1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
          2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
          3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
          4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
          5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
          6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
            CALL LVLCH
C           CONVERT TO SCREEN CO-ORDINATES
            CALL V2ST (1,X,Y,IX,IY)
C           SKIP IF LINE IS COMPLETELY OUTSIDE WINDOW
            IF(KGNFLG.EQ.1) GO TO 10
            CALL VECMD
            CALL XYCNT (IX,IY)
         10 RETURN
            END
            SUBROUTINE PARCL (RL1,RL2,RM1,RM2,RN1,RN2)
            IF(RL1.LT.RM1) GO TO 10
            IF(RL1.GT.RM2) GO TO 20
            RN1=RL1
            IF(RL2-RM1) 30,40,40
         10 RN1=RM1
         40 IF(RL2.LE.RM2) GO TO 50
            RN2=RM2
            GO TO 60
         50 RN2=RL2
            GO TO 60
         20 RN1=RM2
            IF(RL2.GE.RM1) GO TO 50
         30 RN2=RM1
         60 RETURN
            END
            SUBROUTINE DRAWA (X,Y)
            COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
          1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
          2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
          3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
          4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
          5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
          6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
C           SET TERMINAL TO DRAW SOLID LINES IF NEEDED
            CALL LVLCH
```

```
C     CONVERT TO SCREEN CO-ORDINATES
      CALL V2ST (1,X,Y,IX,IY)
C     SKIP IF LINE COMPLETELY OUTSIDE WINDOW
      IF(KGNFLG.EQ.1) GO TO 10
      IF(KKMODE.NE.1) CALL VECMD
      IF(KMOVEF.EQ.1) CALL XYCNT (KBEAMX,KBEAMY)
      CALL XYCNT (IX,IY)
   10 RETURN
      END
      SUBROUTINE DWIND (XMIN,XMAX,YMIN,YMAX)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
C     DEFINE DATA WINDOW IN TERMINAL COMMON AREA
      TMINVX=XMIN
      TMAXVX=XMAX
      TMINVY=YMIN
      TMAXVY=YMAX
      CALL RESCL
      RETURN
      END
      SUBROUTINE WINCO (X,Y,IX,IY)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
C     DATA DE2RAD/0.01745/
C     CHECK FOR PERMITTED VALUE OF CONVERSION KEY
C     DEFAULT IS LINEAR, ERROR IS NONE
      DX=X-TMINVX
      DY=Y-TMINVY
      IF (KEYCON .EQ. 1) GO TO 500
C     KEY=KEYCON
C     IF(KEYCON.LT.1) KEY=5
C     IF(KEYCON.GT.4) KEY=4
C     BRANCH TO PROPER SECTION
C     LINEAR LOG POLAR USER ERROR
C     GO TO (500,300,600,700,100),KEY
C     ERROR
  100 IX=X
      IY=Y
      GO TO 800
C     LOG TRANSFORM
C 300 KEYL=TRPAR1+0.001
C     IF(KEYL.EQ.2) GO TO 400
C     SETUP X LOG TRANSFORM
C     DX=ALOG(X)-TRPAR2
C 400 IF(KEYL.EQ.1) GO TO 500
C     SETUP Y LOG TRANSFORM
C     DY=ALOG(Y)-TRPAR3
C     CONVERT LINEAR
  500 IX=IFIX(DX*TRFACX)+KMINSX
      IY=IFIX(DY*TRFACY)+KMINSY
C     GO TO EXIT
```

```
C       GO TO 800
C       POLAR TRANSFORM
C 600   A=(Y-TRPAR6)*TRFACY
C       R=(X-TRPAR5)*TRFACX
C       IX=R*COS(A*DE2RAD)+TRPAR3
C       IY=R*SIN(A*DE2RAD)+TRPAR4
C       GO TO EXIT
C       GO TO 800
C       USER TRANSFORM IN USE
C 700   CONTINUE
C       CALL USECO (X,Y,IX,IY)
C       EXIT POINT
  800   RETURN
        END
        SUBROUTINE V2ST (I,X,Y,IX,IY)
        DIMENSION BUFIN(4),BFOUT(4)
        COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
       1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
       2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
       3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
       4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
       5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
       6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
        EQUIVALENCE (BUFIN(1),XS),(BUFIN(2),YS),(BUFIN(3),XE),
       1 (BUFIN(4),YE)
        EQUIVALENCE (BFOUT(1),CXS),(BFOUT(2),CYS),(BFOUT(3),CXE),
       1 (BFOUT(4),CYE)
        XE=X
        YE=Y
C       POINT OR MOVE
        IF(I.EQ.0) GO TO 10
C       BRIGHT VECTOR
        XS=TIMAGX
        YS=TIMAGY
C       CLIP VECTOR
        CALL CLIPT (BUFIN,BFOUT)
C       ON SCREEN
        IF(KGNFLG.EQ.1) GO TO 110
C       ARE WE AT START POINT
        IF(CXS.EQ.TREALX.AND.CYS.EQ.TREALY) GO TO 120
C       MOVE BEAM TO START POINT
        MODE=KKMODE
        CALL VECMD
        CALL WINCO (CXS,CYS,IX,IY)
        CALL XYCNT (IX,IY)
        KKMODE=MODE
        GO TO 120
C       POINT OR MOVE
   10   CALL PCLIP (XE,YE)
C       OFF SCREEN
        IF(KGNFLG.EQ.1) GO TO 110
        CXE=XE
        CYE=YE
C       CONVERT TO SCREEN CO-ORDINATES
  120   CALL WINCO (CXE,CYE,IX,IY)
C       SAVE POSITION ABS AND IMAGINARY
        TREALX=CXE
        TREALY=CYE
  110   TIMAGX=X
        TIMAGY=Y
        RETURN
```

```
      END
      SUBROUTINE REVCO (IX,IY,X,Y)
      LOGICAL DEC
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
C     E=2.718281828
      DX=FLOAT(IX-KMINSX)/TRFACX
      DY=FLOAT(IY-KMINSY)/TRFACY
      IF (KEYCON .EQ.1) GO TO 300
C     KEY=KEYCON
C     IF(KEYCON.LT.1) KEY=5
C     IF(KEYCON.GT.4) KEY=4
C     LINEAR LOG POLAR USER ERROR
C     GO TO (300,400,500,600,100),KEY
C     ERROR
  100 X=IX
      Y=IY
      GO TO 700
C     LINEAR
  300 X=DX+TMINVX
      Y=DY+TMINVY
C     GO TO 700
C     LOG SCALES
C 400 KEYL=TRPAR1
C     X=DX+TMINVX
C     Y=DY+TMINVY
C     IF(KEYL.NE.2) X=E**(DX+TRPAR2)
C     IF(KEYL.NE.1) Y=E**(DY+TRPAR3)
C     GO TO 700
C     POLAR
C 500 DX=FLOAT(IX)-TRPAR3
C     DY=FLOAT(IY)-TRPAR4
C     Y=ATAN2(DX,DY)*57.295780
C     X=SQRT(DY*DY+DX*DX)/TRFACX+TRPAR5
C     ADJUST ANGLE MOD 2 PI TO VALUE WITHIN WINDOW
C     DEC=.FALSE.
C 510 IF(Y.GT.TRPAR1) GO TO 530
C     INCREMENT ANGLE
C     Y=Y+360.
C     GO TO 510
C 530 IF(Y.LE.TRPAR2) GO TO 550
C     DECREMENT ANGLE
C     Y=Y-360.
C     DEC=.TRUE.
C     GO TO 530
C 550 IF(DEC.AND.Y.LT.TRPAR1) Y=Y+360.0
C     IF(TMINVX.GE.0) GO TO 560
C     TR1A=AMOD(TRPAR1+180.,360.)
C     TR2A=AMOD(TRPAR2+180.,360.)
C     IF(Y.GT.AMAX1(TR1A,TR2A).OR.Y.LT.AMIN1(TR1A,TR2A)) GO TO 560
C     Y=AMOD(Y+180.,360.)
C     X=-X
C 560 Y=Y/TRFACY+TRPAR6
C     GO TO 700
C     USER CONVERSION
C 600 CONTINUE
```

```
C     CALL UREVC (IX,IT,X,Y)
C     EXIT POINT
  700 CALL PCLIP (X,Y)
      RETURN
      END
      SUBROUTINE CLIPT (BUFIN,OUTBF)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      DIMENSION BUFIN(1),OUTBF(1)
      GSTAX=BUFIN(1)
      GSTAY=BUFIN(2)
      GENDX=BUFIN(3)
      GENDY=BUFIN(4)
      IF(GSTAX.GE.TMINVX) GO TO 10
      IF(GENDX.GE.TMINVX) GO TO 20
      GO TO 110
   10 IF(GSTAX.LE.TMAXVX) GO TO 20
      IF(GENDX.LE.TMAXVX) GO TO 20
      GO TO 110
   20 IF(GSTAY.GE.TMINVY) GO TO 21
      IF(GENDY.GE.TMINVY) GO TO 30
      GO TO 110
   21 IF(GSTAY.LE.TMAXVY) GO TO 30
      IF(GENDY.LE.TMAXVY) GO TO 30
      GO TO 110
   30 IF(GSTAX.NE.GENDX) GO TO 31
      DSTAX=GSTAX
      DENDX=GSTAX
      CALL PARCL (GSTAY,GENDY,TMINVY,TMAXVY,DSTAY,DENDY)
      GO TO 120
   31 IF(GSTAY.NE.GENDY) GO TO 40
      DSTAY=GSTAY
      DENDY=GSTAY
      CALL PARCL (GSTAX,GENDX,TMINVX,TMAXVX,DSTAX,DENDX)
      GO TO 120
   40 A=GENDX-GSTAX
      B=GENDY-GSTAY
      IF(GSTAX.LT.TMINVX) GO TO 41
      IF(GSTAX.LE.TMAXVX) GO TO 43
      Q=TMAXVX
      GO TO 42
   43 IF(GSTAY.GT.TMAXVY) GO TO 140
      IF(GSTAY.LT.TMINVY) GO TO 44
      DSTAX=GSTAX
      DSTAY=GSTAY
      GO TO 150
   41 Q=TMINVX
   42 DSTAY=GSTAY+((Q-GSTAX)*B/A)
      IF(DSTAY.GT.TMAXVY) GO TO 140
      IF(DSTAY.LT.TMINVY) GO TO 44
      DSTAX=Q
      GO TO 150
   44 R=TMINVY
      GO TO 45
  140 R=TMAXVY
   45 DSTAX=GSTAX+((R-GSTAY)*A/B)
```

```
      IF(DSTAX.GT.TMAXVX) GO TO 110
      IF(DSTAX.LT.TMINVX) GO TO 110
      DSTAY=R
  150 IF(GENDX.LT.TMINVX) GO TO 50
      IF(GENDX.GT.TMAXVX) GO TO 51
      IF(GENDY.GT.TMAXVY) GO TO 160
      IF(GENDY.LT.TMINVY) GO TO 52
      DENDX=GENDX
      DENDY=GENDY
      GO TO 120
   51 Q=TMAXVX
      GO TO 53
   50 Q=TMINVX
   53 DENDY=GSTAY+((Q-GSTAX)*B/A)
      IF(DENDY.GT.TMAXVY) GO TO 160
      IF(DENDY.LT.TMINVY) GO TO 52
      DENDX=Q
      GO TO 120
   52 R=TMINVY
      GO TO 60
  160 R=TMAXVY
   60 DENDX=GSTAX+((R-GSTAY)*A/B)
      DENDY=R
  120 OUTBF(1)=DSTAX
      OUTBF(2)=DSTAY
      OUTBF(3)=DENDX
      OUTBF(4)=DENDY
      KGNFLG=0
      GO TO 70
C     SET FLAG IF LINE OUTSIDE WINDOW
  110 KGNFLG=1
   70 RETURN
      END
      SUBROUTINE RESET
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      KEYCON=1
      TRFACX=1.
      TRFACY=1.
      KBEAMX=0
      KHOMEY=3068/KFACTR
      KBEAMY=KHOMEY
      KMINSX=0
      KMAXSX=4096/KFACTR
      KMINSY=0
      KMAXSY=3120/KFACTR
      KHORSZ=56
      KLINE=0
      KZAXIS=0
      KLMRGN=0
      KRMRGN=4040/KFACTR
      KSIZEF=1
      KTBLSZ=10
      KVERSZ=88
      TMINVX=0.
      TMAXVX=4095./FLOAT(KFACTR)
```

```
      TMINVY=0.
      TMAXVY=3120./FLOAT(KFACTR)
      TRCOSF=1.
      TRSINF=0.
      TRSCAL=1.
C     MOVE TO THE HOME POSITION
      CALL MOVAB (KLMRGN,KHOMEY)
C     PLACE TERMINAL IN A/N MODE
      CALL ALFMD
      RETURN
      END
      SUBROUTINE NWPAG
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      DIMENSION ICODE(2)
      DATA ICODE(1),ICODE(2)/27,12/
      IF(KKMODE.NE.0) CALL ALFMD
C     OUTPUT (ESC) (FF) FOR NEW PAGE
      CALL TOUTS (2,ICODE)
      CALL IOWAI (10)
      IF(KLMRGN.EQ.0) GO TO 10
      CALL MOVAB (KLMRGN,KHOMEY)
      CALL ALFMD
      GO TO 20
   10 KBEAMX=0
      KBEAMY=KHOMEY
   20 RETURN
      END
      SUBROUTINE IOWAI (ITIME)
C     THIS ROUTINE IS USED TO GENERATE DELAYS FOR REMOTE TERMINALS
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      IF(KBAUDR.LE.0) GO TO 20
      KOUNT=ITIME*KBAUDR/10
      DO 10 J=1,KOUNT
C     OUTPUT (SYN) TO INSURE AGAINST LOSS OF OUTPUT WHILE
C     TERMINAL IS BUSY. (SYN) DOES NOT AFFECT THE TERMINAL
   10 CALL TOUTP (22)
   20 RETURN
      END
      SUBROUTINE XYCNT (IX,IY)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      DIMENSION IPLT(5),IOPT(5)
      DATA IDREW,ISYN/0,22/
C     RECEIVE THE PLOT CHARACTERS
```

```
              CALL PLCHR (IX,IY,IPLT)
      C       OPTIMIZE THE OUTPUT
              LEN=0
      C       CHECK IF HIGH Y IS NEEDED
              IF(KPCHAR(1).EQ.IPLT(1)) GO TO 10
      C       INCLUDE HIGH Y IF NEEDED
              LEN=1
              KPCHAR(1)=IPLT(1)
              IOPT(1)=IPLT(1)
      C       CHECK IF LSBXY IS NEEDED
         10   IF(KTERM.LE.2) GO TO 20
              IF(KPCHAR(2).EQ.IPLT(2)) GO TO 20
      C       INCLUDE LSBXY IF NEEDED
              LEN=LEN+1
              KPCHAR(2)=IPLT(2)
              IOPT(LEN)=IPLT(2)
              GO TO 30
      C       CHECK IF LOW Y IS NEEDED
         20   IF(KPCHAR(3).NE.IPLT(3)) GO TO 30
              IF(KPCHAR(4).EQ.IPLT(4)) GO TO 40
      C       INCLUDE LOW Y IF NEEDED
         30   LEN=LEN+1
              KPCHAR(3)=IPLT(3)
              IOPT(LEN)=IPLT(3)
      C       CHECK IF HIGH X IS NEEDED
              IF(KPCHAR(4).EQ.IPLT(4)) GO TO 50
      C       INCLUDE HIGH X IF NEEDED
              LEN=LEN+1
              KPCHAR(4)=IPLT(4)
              IOPT(LEN)=IPLT(4)
      C       CHECK IF LOW X IF NEEDED
         40   IF(KPCHAR(5).NE.IPLT(5)) GO TO 50
      C       CHECK IF ALL THE CHARACTERS ARE THE SAME
              IF(LEN.NE.0) GO TO 50
      C       CHECK IF (GS) FOR DARK VECTOR ALREADY SENT
              IF(KMOVEF.EQ.1) GO TO 50
      C       CHECK IF VECTOR IS ALREADY DRAWN TO SPOT
              IF(IDREW.EQ.1) GO TO 80
      C       INCLUDE THE LOW X
         50   LEN=LEN+1
              KPCHAR(5)=IPLT(5)
              IOPT(LEN)=IPLT(5)
      C       CHECK FOR POSSIBLE SPEED PROBLEM
         60   IF(LEN.GE.KPAD2) GO TO 70
              LEN=LEN+1
              IOPT(LEN)=ISYN
              GO TO 60
      C       SEND THE ARRAY TO THE OUTPUT BUFFER
         70   CALL TOUTS (LEN,IOPT)
      C       SET COMMON AND HISTORY VARIABLES
      C       SET THE DREW HERE FLAG
              IDREW=1
      C       REMOVE THE DREW HERE FLAG IF DIDN'T DRAW
              IF(KMOVEF.EQ.1) IDREW=0
      C       REMOVE THE MOVE FLAG
              KMOVEF=0
         80   KBEAMX=IX
              KBEAMY=IY
              RETURN
              END
              SUBROUTINE ALFMD
```

```
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      SET ALPHA MODE OUTPUT (US)
      CALL TOUTP (31)
      KKMODE=0
      IF(KBEAMY.GT.KHOMEY) KBEAMY=KHOMEY
      RETURN
      END
      SUBROUTINE TOUTP (KKOUT)
      DIMENSION KOUT(1)
      KOUT(1)=KKOUT
      CALL TOUTS (1,KOUT)
      RETURN
      END
      SUBROUTINE TSEND
      DIMENSION ITEMP(1)
      CALL BUFFK (0,ITEMP)
      RETURN
      END
      SUBROUTINE RESCL
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      IF (KEYCON .NE. 1) GO TO 500
C     KEY=KEYCON
C     IF(KEYCON.LT.1) KEY=5
C     IF(KEYCON.GT.4) KEY=4
C     BRANCH TO PROPER SECTION AND RETURN
C     LINEAR LOG POLAR USER ERROR
C     GO TO (100,200,300,400,500),KEY
C     BOTH AXES LINEAR
C100  TRPAR1=0.
C     SEMI LOG OR LOG-LOG
C200  KEYL=TRPAR1+1.0001
C     X-AXIS LINEAR OR LOG
C     GO TO (210,220,210,220),KEYL
C     LINEAR
  210 TRFACX=FLOAT(KMAXSX-KMINSX)/(TMAXVX-TMINVX)
C     GO TO 250
C     SEMI-LOG X-AXIS
C220  TRPAR2=ALOG(TMINVX)
C     TRFACX=FLOAT(KMAXSX-KMINSX)/(ALOG(TMAXVX)-TRPAR2)
C     Y-AXIS LINEAR OR LOG
C250  GO TO (260,260,280,280),KEYL
C     LINEAR
  260 TRFACY=FLOAT(KMAXSY-KMINSY)/(TMAXVY-TMINVY)
      GO TO 600
C     SEMI-LOG Y-AXIS
C280  TRPAR3=ALOG(TMINVY)
C     TRFACY=FLOAT(KMAXSY-KMINSY)/(ALOG(TMAXVY)-TRPAR3)
C     GO TO 600
C     POLAR SCALING
```

```
C300  CONTINUE
C     CALL PSCAL
C     GO TO 600
C     USER DEFINE FUNCTION
C400  CONTINUE
C     CALL URSCL
C     GO TO 600
C     NO SCALE
  500 TRFACX=1.
      TRFACY=1.
  600 RETURN
      END
      SUBROUTINE MOVAB (IX,IY)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      CALL VECMD
      CALL XYCNT (IX,IY)
      KGRAFL=0
      RETURN
      END
      SUBROUTINE VECMD
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      IF(KKMODE.EQ.1) GO TO 10
C     OUTPUT (US) TO ENTER A/N MODE AND RESET FOR VECTOR MODE
      CALL TOUTP (31)
      DO 112 II=1,5
  112 KPCHAR(II)=-1
      KKMODE=1
C     OUTPUT (GS) TO ENTER VECTOR MODE
   10 CALL TOUTP (29)
      KMOVEF=1
      RETURN
      END
      SUBROUTINE PLCHR (IX,IY,ICHAR)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      DIMENSION ICHAR(1)
C     CALCULATE THE PLOT CHARACTERS TO ARRIVE AT IX,IY
C     ORDER IS HIY,LSBYX,LOY,HIX,LOX
      KX=IX*KFACTR
      KY=IY*KFACTR
      ICHAR(1)=MOD(KY/128,32)+32
      ICHAR(2)=MOD(KY,4)*4+MOD(KX,4)+96
      ICHAR(3)=MOD(KY/4,32)+96
      ICHAR(4)=MOD(KX/128,32)+32
```

```
      ICHAR(5)=MOD(KX/4,32)+64
      RETURN
      END
      SUBROUTINE INITT (IBAUD)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      KBAUDR=IBAUD
      KPAD2=KBAUDR/220+1
      KTERM=1
      KFACTR=4
C     SET THE OUTPUT BUFFER FORMAT
      KUNIT=3
      KINLFT=0
      KOTLFT=1
      CALL RESET
      CALL NWPAG
      RETURN
      END
      SUBROUTINE FINIT (IX,IY)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      CALL MOVAB (IX,IY)
      CALL ALFMD
      CALL TSEND
      RETURN
      END
      SUBROUTINE TOUTS (LEN,IADE)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      DIMENSION IADE(1),IUSE(144)
      DATA ISYNC,MAXLEN/22,144/
      IF(LEN.LE.0) GO TO 40
      LENOUT=0
      ITEST=KPAD2-1
      DO 30 I=1,LEN
      ITEMP=IADE(I)
C     INSERT CODE EXPANSION CHARACTER HERE WHEN NEEDED
      IF(LENOUT.GE.MAXLEN) GO TO 40
      LENOUT=LENOUT+1
   30 IUSE(LENOUT)=ITEMP
   40 CALL BUFFK (LENOUT,IUSE)
      RETURN
      END
      SUBROUTINE BUFFK (NCHAR,IOUT)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
```

```
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
      DIMENSION IDATA(144),IOUT(1)
      DATA MAXLEN,LENOUT,NODATA,ITRAIL/144,0,1,1/
C     DUMP THE BUFFER WHEN REQUESTED BY LEN=0
      IF(NCHAR.LE.0) GO TO 10
C     DON'T DUMP THE BUFFER IF THE NEW STRING WILL FIT
      IF(NCHAR.LE.KOTLFT) GO TO 70
C     DETERMINE IF THERE IS DATA IN BUFFER
   10 IF(NODATA.EQ.1) GO TO 50
C     DETERMINE THE FORMAT THE USER WANTS BUFFER DUMPED IN
      GO TO (20,30,40),KUNIT
C     OUTPUT BUFFER FORMAT IS (GS),PLTCHRS,DATA,(US)
   20 LENOUT=LENOUT+1
C     APPEND (US) TO END OF BUFFER
      IDATA(LENOUT)=31
      CALL ADOUT (LENOUT,IDATA)
      CALL PLCHR (KBEAMX,KBEAMY,IDATA)
C     RESORE THE BEAM POSITION AT FIRST OF THE NEXT BUFFER
      IDATA(2)=IDATA(1)
      IDATA(1)=29
C     AND NOW THE MODE BEFORE THE OUTPUT IS ASKED FOR
      LENOUT=6
      KEY=KKMODE+1
      IF(KEY.LT.1) KEY=1
      IF(KEY.GT.5) KEY=1
C     MODE IS A/N,VECTOR,PNT,INC,DSH
      GO TO (21,22,23,24,22),KEY
C     ENTER A/N MODE
   21 IDATA(LENOUT)=31
      GO TO 50
C     ENTER VECTOR MODE
   22 IDATA(LENOUT)=29
      IF(KMOVEF.NE.1) LENOUT=LENOUT-1
      GO TO 50
C     ENTER POINT MODE
   23 IF(KTERM.LT.3) GO TO 22
      IDATA(LENOUT)=28
      LENOUT=LENOUT+1
      GO TO 22
C     ENTER INCREMENT PLOT MODE
   24 IDATA(LENOUT)=30
      GO TO 50
C     OUTPUT BUFFER FORMAT IS (SYN),DATA,(ESC)
   30 LENOUT=LENOUT+1
C     APPEND (ESC) TO END THE BUFFER
      IDATA(LENOUT)=27
      CALL ADOUT (LENOUT,IDATA)
      IDATA(1)=22
      LENOUT=1
      GO TO 50
C     OUTPUT BUFFER FORMAT IS DATA ONLY
   40 CALL ADOUT (LENOUT,IDATA)
      LENOUT=0
   50 KOTLFT=MAXLEN-LENOUT-ITRAIL
C     TEKTRONIX BUG FIXED IN NEXT LINE
      IF(NCHAR.LE.0) GO TO 90
   70 NODATA=0
```

```
      LEN=NCHAR
      IF(LEN.GT.KOTLFT) LEN=KOTLFT
      DO 80 I=1,LEN
      LENOUT=LENOUT+1
   80 IDATA(LENOUT)=IOUT(I)
   90 KOTLFT=MAXLEN-LENOUT-ITRAIL
      RETURN
      END
      SUBROUTINE ADOUT (NCHAR,IARAY)
      DIMENSION IARAY(1),KARAY(73)
      COMMON LARRY(99),LU
      IF (LU .NE. 7) LU=1
C     TEST FOR NCHAR=0
      IF(NCHAR.LE.0) RETURN
C     OUTPUT ASCII CODE (95) TO SUPPRESS (CR) AND (LF)
C     PACK ADE ARRAY INTO A2 ARRAY
      KK=(NCHAR+1)/2
      DO 10 I=1,KK
      II=I+I-1
      KARAY(I)=IARAY(II)*256
      IF (II .EQ. NCHAR) GO TO 20
   10 KARAY(I)=KARAY(I)+IARAY(I+I)
      KARAY(KK+1)=95*256
      GO TO 30
   20 KARAY(I)=KARAY(I)+95
C     SEND THE ARRAY,SUPPRESSING CR AND LF.
   30 K=-NCHAR-1
      CALL EXEC (2,LU,KARAY,K)
      RETURN
      END
      SUBROUTINE ANMOD
C     ENTER ALPHA-NUMERIC MODE
      CALL ALFMD
C     DUMP THE OUTPUT BUFFER
      CALL TSEND
      RETURN
      END
      SUBROUTINE TWIND (MINX,MAXX,MINY,MAXY)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
C     DEFINE TERMINAL WINDOW IN TERMINAL COMMON AREA
      KMINSX=MINX
      KMAXSX=MAXX
      KMINSY=MINY
      KMAXSY=MAXY
      CALL RESCL
      RETURN
      END
      SUBROUTINE PCLIP (X,Y)
      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
     3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
```

```
      KGNFLG=0
      IF(X.LT.TMINVX) GO TO 10
      IF(X.GT.TMAXVX) GO TO 10
      IF(Y.LT.TMINVY) GO TO 10
      IF(Y.GT.TMAXVY) GO TO 20
   10 KGNFLG=1
   20 RETURN
      END
      ENDS
```