

---

**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 to 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  $\mu$ , 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:

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

$$x_{0max} = \max \{x_0(t)\} \quad (2)$$

$$x_{0min} = \min \{x_0(t)\} \quad (3)$$

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

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

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

429 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 25 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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 non-stationary random process of the first kind defined as

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

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 of  $x_0(t)$  given in Eq.(6), and

$\phi$  = a realization of random phase angle  $\Phi$  whose distribution can be selected, according to the control parameter ICON, as follows:

If ICON=3,  $\Phi$  is Gaussian.

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

Otherwise,  $\Phi$  is uniform between  $-\pi$  and  $\pi$ .

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) in between 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  $N=2 N_Y$  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

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

431 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 27 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

where  $x_i(t)$  = simulated component process

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

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

$\Phi_i$ '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  $\mu_i$  and  $\sigma_i$  = mean and standard deviation of  $\Phi_i$ , respectively, and

$\rho$  = coefficient of correlation between  $\Phi_1$  and  $\Phi_2$ .

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  $R_{x_1 x_2}(t_1, t_2)$  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  $\Phi_1$  and  $\Phi_2$  being jointly Gaussian as in Eq. (9).

Specific representation of  $R_{x_1 x_2}(t_1, t_2)$  is as follows:

$$\begin{aligned} R_{x_1 x_2}(t_1, t_2) = & x_{01}(t_1) x_{02}(t_2) \cdot E\{\cos \Phi_1 \cos \Phi_2\} \\ & + \hat{x}_{01}(t_1) \hat{x}_{02}(t_2) \cdot E\{\sin \Phi_1 \sin \Phi_2\} \\ & - \hat{x}_{01}(t_1) x_{02}(t_2) \cdot E\{\sin \Phi_1 \cos \Phi_2\} \\ & - x_{01}(t_1) \hat{x}_{02}(t_2) \cdot E\{\cos \Phi_1 \sin \Phi_2\} \end{aligned} \quad (10)$$

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  $\Phi_i$ 's distributed uniformly, intermediate mode of dependence are very much cumbersome to consider since the joint density functions which involve  $\Phi_i$ 's and produce uniform marginal distributions for  $\Phi_i$ 's appear to be extremely difficult to obtain. However, when

all  $\Phi_i$ '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  $\Phi_i$ '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:

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

where

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

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

$\phi$  = a realization of random phase angle  $\Phi$

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

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

in which

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

$$v(t) = U(t) - U(t - T_0) \quad (13)$$

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

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

$$= \lim_{N \rightarrow \infty} \sum_{k=-N}^N y_0(t - 2kT_0) \quad (14)$$

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

$$y_0(t) = x_0(t) + x_0(-t) \quad (15)$$

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

433 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 29 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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:

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

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  $\pi$  of the Fourier transform  $X_0(\omega)$  of the original time history  $x_0(t)$ .

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

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

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

$$\zeta_0(\omega) = \pm \pi / 2 \quad (18)$$

where  $\text{Re}[X_0(\omega)]$  represents real part of  $X_0(\omega)$  and  $\text{Im}[X_0(\omega)]$  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  $\omega$  (rad./sec),

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

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

Required 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 frequency domain as

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

$$= \frac{1}{\pi} \int_0^{\infty} |X_0(\omega)| \sin \{\omega t + \xi_0(\omega)\} d\omega \quad (20)$$

where  $X_0(\omega)$  and  $\xi_0(\omega)$  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

$$Y_0(\omega) = |X_0(\omega)| \cdot \text{sgn}(\omega) \quad (21)$$

$$\begin{aligned} Z_0(\omega) &= X_0(\omega) \cdot \text{sgn}(\omega) \\ &= |X_0(\omega)| \cdot \exp \{i\xi_0(\omega)\} \cdot \text{sgn}(\omega) \\ &= Y_0(\omega) \cdot \exp \{i\xi_0(\omega)\} \end{aligned} \quad (22)$$

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



435 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 31 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

$$\begin{aligned}\hat{x}_0(t) &= \frac{1}{\pi} \int_0^{\infty} |X_0(\omega)| \sin \{\omega t + \xi_0(\omega)\} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} Y(\omega) \sin \{\omega t + \xi_0(\omega)\} d\omega\end{aligned}\quad (23)$$

where the fact has been used that  $\sin \{\omega t + \xi_0(\omega)\}$  is an odd function of  $\omega$ . Further, since  $Y(\omega)$  is odd and  $\cos \{\omega t + \xi_0(\omega)\}$  is even with respect to  $\omega$ , Eq. (23) can be written as

$$\begin{aligned}\hat{x}_0(t) &= \frac{1}{2\pi i} \int_{-\infty}^{\infty} Y_0(\omega) \exp \{i \{\omega t + \xi_0(\omega)\}\} d\omega \\ &= \frac{1}{2\pi i} \int_{-\infty}^{\infty} Y_0(\omega) \cdot \exp \{i \xi_0(\omega)\} \exp (i\omega t) d\omega \\ &= \frac{1}{2\pi i} \int_{-\infty}^{\infty} Z_0(\omega) \exp (i\omega t) d\omega\end{aligned}\quad (24)$$

Eq. (24) now possesses 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  $y_i$  ( $i=1, 2, \dots, N$ ) in a symmetrical fashion so as to produce mirror image either around the origin or around a folding point  $(i = \frac{N}{2} + 1)$  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 display 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

437 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 33 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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, Columbia 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

```

0001 FIN4
0002 C -----(MAIN PROGRAM SIM10)-----
0003 PROGRAM SIM10
0004 C.... THIS PROGRAM GETS BASIC STATISTICS OF TIME HISTORY X0(T)
0005 C WITH DURATION TT (SEC) SUCH AS TEMPORAL MEAN, MAXIMUM AND
0006 C MINIMUM VALUES DURING DURATION TT, FOURIER AMPLITUDE SPECTRUM
0007 C AND PHASE ANGLE OF THE FOURIER TRANSFORM OF X0(T) AS WELL AS
0008 C ITS HILBERT TRANSFORM.
0009 C THIS PROGRAM ALSO PLOTS TIME HISTORY X0(T), ITS FOURIER
0010 C AMPLITUDE SPECTRUM, PHASE ANGLE AND ITS HILBERT TRANSFORM.
0011 C
0012 C.... DESCRIPTION OF MAIN PARAMETERS:
0013 C X0(I) -- EQUIDISTANT TIME SERIES DATA TO BE ANALYZED.
0014 C WHEN THE ORIGINAL DATA ARE NOT EQUIDISTANT,
0015 C PROGRAM AUTOMATICALLY INTERPOLATES TO GET
0016 C EQUIDISTANT DATA.
0017 C T(I) -- TIME ARRAY (SEC).
0018 C W(I) -- FREQUENCY ARRAY (RADIAN/SEC).
0019 C PSD(I) -- FOURIER AMPLITUDE SPECTRUM.
0020 C PHS(I) -- PHASE ANGLE BETWEEN -PI AND PI.
0021 C XHIL(I) -- HILBERT TRANSFORM OF X0(T).
0022 C.... REQUIRED SUBROUTINES: STUKO,STATC,SPECT,PHASE,HLBRT,FFT,INTPL
0023 C.... REMARKS:
0024 C FOR PLOTTING PURPOSE IS USED SUBROUTINE STUKO WHICH
0025 C UTILIZES TEKTRONIXS TERMINAL CONTROL SYSTEM IN HP
0026 C MINI-COMPUTER. THEREFORE SLIGHT MODIFICATION IS NEEDED
0027 C FOR IBM COMPUTER.
0028 C
0029 C DIMENSION A(1024),X0(513),T(513),W(513),PSD(513),PHS(513)
0030 C DIMENSION NAME(35),XHIL(513)
0031 C
0032 C.... TO ASSIGN LOGICAL UNIT NUMBER OF INPUT/OUTPUT PERIPHERALS.
0033 C LOGICAL UNIT NO. = 1 -- TERMINAL
0034 C LOGICAL UNIT NO. = 6 -- LINE PRINTER
0035 C LOGICAL UNIT NO. = 8 -- MAG-TAPE UNIT
0036 C LU=1
0037 C IIN=8
0038 C IOUT=6
0039 C
0040 C.... TO SPECIFY TIME SERIES DATA TO BE ANALYZED (UP TO 70 CHAR.).
0041 C WRITE(LU,600)
0042 C 600 FORMAT(" INPUT -- PROBLEM IDENTIFICATION CODE")
0043 C READ(LU,550)(NAME(I),I=1,35)
0044 C 550 FORMAT(35A2)
0045 C WRITE(IOUT,660)(NAME(I),I=1,35)
0046 C 660 FORMAT(1H1////5X,35A2)
0047 C
0048 C.... TO ASSIGN NECESSARY PARAMETER VALUES.
0049 C NORG -- NUMBER OF ORIGINAL DATA.
0050 C NDATA -- NUMBER OF INTERPOLATED DATA.
0051 C FOR FFT COMPUTATION, NDATA MUST BE A POWER OF 2.
0052 C WRITE(LU,601)
0053 C 601 FORMAT(" INPUT --- NORG, NDATA")
0054 C READ(LU,*) NORG,NDATA
0055 C
0056 C.... TO READ IN THE ORIGINAL TIME SERIES FROM MAG-TAPE.
0057 C REWIND IIN
0058 C READ(IIN,300)(A(I),W(I),I=1,NORG)
0059 C 300 FORMAT(6F10.3)

```

439 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 35 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0060      REWIND IIN
0061 C
0062 C.... TO COMPUTE AND PRINT OUT APPROPRIATE QUANTITIES.
0063      NFOLD=NDATA/2+1
0064      NDATA1=NDATA+1
0065      TT=A(NORG)-A(1)
0066      DT=TT/FLOAT(NDATA)
0067      PI1=3.1415926
0068      PI2=2.0*PI1
0069      DF=1.0/TT
0070      DW=PI2*DF
0071      FOLDF=DF*FLOAT(NFOLD)
0072      FOLDW=PI2*FOLDF
0073      WRITE(IOUT,630)NORG,NDATA,NFOLD,TT,DT,DF,DW,FOLDF,FOLDW
0074      630 FORMAT( ////
0075      *5X,"NUMBER OF ORIGINAL DATA -- NORG =",I4/
0076      *5X,"NO. OF INTERPOLATED DATA-- NDATA =",I4/
0077      *5X,"NYQUIST FOLDING NUMBER -- NFOLD =",I4/
0078      *5X,"DURATION (PERIOD)      -- TT  =",E12.5," (SEC)"/
0079      *5X,"TIME INCREMENT          -- DT  =",E12.5," (SEC)"/
0080      *5X,"FREQUENCY INCREMENT     -- DF  =",E12.5," (HZ)"/
0081      *5X,"                        -- DW  =",E12.5," (RAD./SEC)"/
0082      *5X,"NYQUIST FOLDING FREQ.   -- FOLDF=",E12.5," (HZ)"/
0083      *5X,"                        -- FOLDW=",E12.5," (RAD./SEC)"/)
0084 C
0085 C.... TO ADJUST TIME AXIS FOR DATA TO START FROM ZERO (SEC).
0086      TFST=A(1)
0087      DO 310 I=1,NORG
0088      A(I)=A(I)-TFST
0089      310 CONTINUE
0090      W(1)=0.0
0091 C
0092 C.... TO INTERPOLATE GIVEN DATA TO GET EQUIDISTANT ARRANGEMENT
0093 C      FOR FFT COMPUTATION.
0094      CALL INTPL(A,W,NORG,T,X0,NDATA1,DT)
0095 C
0096 C.... TO COMPUTE AND PRINT OUT BASIC STATISTICS OF X0(T).
0097      CALL STATC(X0,NDATA,EX,VAR,STDV,DMAX,DMIN)
0098      WRITE(IOUT,610)EX,VAR,STDV,DMAX,DMIN
0099      610 FORMAT(//5X,"STATISTICS OF ORIGINAL RECORD"/
0100      *5X,"TEMPORAL MEAN      EX=",E12.5/
0101      *5X,"VARIANCE           VAR=",E12.5/
0102      *5X,"STAND. DEVIATION STDV=",E12.5/
0103      *5X,"MAXIMUM VALUE      MAX=",E12.5/
0104      *5X,"MINIMUM VALUE      MIN=",E12.5/)
0105 C
0106 C.... TO COMPUTE PHASE ANGLE AND FOURIER (AMPLITUDE) SPECTRUM.
0107      CALL PHASE(A,X0,NDATA,PHS)
0108      CALL SPECT(A,X0,NDATA,TT,1)
0109      DO 200 I=1,NDATA
0110      PSD(I)=A(I)
0111      W(I)=DW*FLOAT(I-1)
0112      200 CONTINUE
0113 C
0114 C.... TO PLOT TIME HISTORY, FOURIER (AMPLITUDE) SPECTRUM AND
0115 C      PHASE ANGLE.
0116 C..... FIRST, ASSIGN DEFAULT PARAMETER VALUES FOR PLOTTING.
0117      XMIN1=T(1)
0118      XMAX1=T(NDATA1)
0119      XMIN2=W(1)
0120      XMAX2=W(NFOLD)

```

```

0121      NX=1
0122      NY=1
0123      YMIN1=-200.
0124      YMIN2=-1.0E4
0125      YMIN3=-PI1
0126      YMAX1=200.
0127      YMAX2=1.0E4
0128      YMAX3=PI1
0129      NP=3
0130      MO=1
0131      MAXX0=950
0132      MINX0=150
0133      MINY0=150
0134      MAXY0=700
0135      JDC=22
0136 C
0137 C..... THEN, INPUT NECESSARY PARAMETER VALUES TO GET THE BEST
0138 C      FIGURE.
0139      WRITE(LU,220)
0140      220 FORMAT(" INPUT -- NY,YMIN1,YMAX1,YMIN2,YMAX2,YMIN3,YMAX3"/
0141      *7X,"--NX,XMIN1,XMAX1,XMIN2,XMAX2,NP,MO"/
0142      *7X,"--JDC,MINX0,MAXX0,MINY0,MAXY0")
0143      10 READ(LU,*)NY,YMIN1,YMAX1,YMIN2,YMAX2,YMIN3,YMAX3,
0144      *NX,XMIN1,XMAX1,XMIN2,XMAX2,NP,MO,JDC,MINX0,MAXX0,MINY0,MAXY0
0145      IF(NY.EQ.100) STOP
0146      IF(NY.LT.0) GO TO 20
0147 C
0148 C..... FINALLY, PLOT THE FIGURE.
0149      CALL STUKU(T,X0,NDATA,NX,XMIN1,XMAX1,
0150      1NY,YMIN1,YMAX1,NP,MO,1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0151      CALL STUKU(W,PSD,NFOLD,NX,XMIN2,XMAX2,
0152      1NY,YMIN2,YMAX2,NP,MO,2,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0153      CALL STUKU(W,PHS,NFOLD,NX,XMIN2,XMAX2,
0154      1NY,YMIN3,YMAX3,NP,MO,3,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0155      GO TO 10
0156 C
0157 C.... TO COMPUTE AND PLOT HILBERT TRANSFORM OF X0(T).
0158      20 CALL HLART(A,X0,XHIL,NDATA,0)
0159      WRITE(LU,230)
0160      230 FORMAT(" INPUT--NY,YMIN1,YMAX1,NX,XMIN1,XMAX1,NP,MO"
0161      */
0162      *7X,"--JDC,MINX0,MAXX0,MINY0,MAXY0")
0163      YMIN1=-200.
0164      YMAX1=200.
0165      NX=1
0166      NY=1
0167      NP=2
0168      30 READ(LU,*)NY,YMIN1,YMAX1,NX,XMIN1,XMAX1,NP,MO,JDC,
0169      *MINX0,MAXX0,MINY0,MAXY0
0170      IF(NY.EQ.100) STOP
0171      IF(NY.LT.0) GO TO 40
0172      CALL STUKU(T,X0,NDATA,NX,XMIN1,XMAX1,NY,YMIN1,YMAX1,
0173      *NP,MO,1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0174      CALL STUKU(T,XHIL,NDATA,NX,XMIN1,XMAX1,NY,YMIN1,YMAX1,
0175      *NP,MO,2,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0176      GO TO 30
0177      40 STOP
0178      END
0179      ENDS

```

441 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 37 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

HSM20 T=00004 IS ON CR00002 USING 00029 BLKS R=0218

```

0001 FTN4
0002 C -----(MAIN PROGRAM SIM20)-----
0003 PROGRAM SIM20
0004 C.... THIS PROGRAM GENERATES AND PLOTS SAMPLE FUNCTIONS OF THE
0005 C NONSTATIONARY RANDOM PROCESS OF THE FIRST KIND AS
0006 C  $X(T)=X_0(T)*\cos(RAN)-XHIL(T)*\sin(RAN)$ 
0007 C WHERE  $X(T)$  = SIMULATED SAMPLE FUNCTION OF THE FIRST KIND.
0008 C  $X_0(T)$  = ORIGINAL RECORD.
0009 C  $XHIL(T)$ = HILBERT TRANSFORM OF  $X_0(T)$ .
0010 C  $RAN$  = RANDOM PHASE ANGLE.
0011 C IF ICON=3,  $RAN$ = GAUSSIAN.
0012 C IF ICON=2,  $RAN$ = UNIFORM BETWEEN  $-\pi/2$  AND  $\pi/2$ .
0013 C OTHERWISE,  $RAN$ = UNIFORM BETWEEN  $\pi$  AND  $\pi$ .
0014 C
0015 C.... REQUIRED SUBROUTINES:
0016 C STUKO, RANDU, SPECT, FFT, INTPL, JNORM, AND HLBRT
0017 C.... REMARKS:
0018 C IF ISSW(14) IS ON, STORING PROCESS ON MAG-TAPE IS SKIPPED.
0019 C DATA STORING FORMAT ON MAG-TAPE:
0020 C  $NDATA, NSMPL, DT, DW, TT$  -- FORMAT(2I5,3E12.5)
0021 C  $T(I)$  -- CALL EXEC(2,1108,T,NDATA2)
0022 C  $X_0(I)$  -- CALL EXEC(2,1108,X_0,NDATA2)
0023 C  $X(I)$  -- CALL EXEC(2,1108,X,NDATA2)
0024 C IF ISSW(15) IS ON, FOURIER SPECTRUM PLOTTING IS SKIPPED.
0025 C FOR PLOTTING PURPOSE IS USED SUBROUTINE STUKO WHICH
0026 C UTILIZES TEKTRONIX TERMINAL CONTROL SYSTEM IN HP
0027 C MINI-COMPUTER. THEREFORE SLIGHT MODIFICATION IS NEEDED
0028 C FOR IBM COMPUTER.
0029 C
0030 DIMENSION A(1024),PSD(513),T(513),X_0(513),XHIL(513),X(513,5)
0031 DIMENSION W(257)
0032 C
0033 C.... TO ASSIGN LOGICAL UNIT NO. OF INPUT/OUTPUT PERIPHERALS.
0034 C LOGICAL UNIT NO. = 1 -- TERMINAL
0035 C LOGICAL UNIT NO. = 6 -- LINE PRINTER
0036 C LOGICAL UNIT NO. = 8 -- MAG-TAPE UNIT
0037 LU=1
0038 IIN=8
0039 IOUT=6
0040 C
0041 C.... TO ASSIGN NECESSARY PARAMETER VALUES.
0042 C NORG -- NUMBER OF ORIGINAL DATA.
0043 C NDATA -- NUMBER OF INTERPOLATED DATA. FOR FFT COMPUTATION
0044 C NDATA MUST BE A POWER OF 2.
0045 WRITE(LU,601)
0046 601 FORMAT(" INPUT --- NORG, NDATA")
0047 READ(LU,*) NORG,NDATA
0048 C
0049 C.... TO READ IN THE ORIGINAL RECORD FROM MAG-TAPE.
0050 REWIND IIN
0051 READ(IIN,300)(A(I),PSD(I),I=1,NORG)
0052 300 FORMAT(6F10.3)
0053 REWIND IIN
0054 C
0055 C.... TO COMPUTE APPROPRIATE QUANTITIES.
0056 NFOLD=NDATA/2+1
0057 NDATA1=NDATA+1
0058 NDATA2=NDATA*2
0059 TT=A(NORG)-A(1)

```

```

0060      DT=TT/FLD*7(NDATA)
0061      PI =3.1415926
0062      PI2=PI*2.0
0063      DW=PI2/TT
0064      DO 310 I=1,NFOLD
0065      310 W(I)=DW*FLOAT(I-1)
0066      C
0067      C.... TO ADJUST TIME AXIS FOR DATA TO START FROM T=0 (SEC).
0068      TFST=A(1)
0069      DO 311 I=1,NORG
0070      A(I)=A(I)-TFST
0071      311 CONTINUE
0072      PSD(1)=0.0
0073      C
0074      C.... TO INTERPOLATE GIVEN DATA TO GET EQUIDISTANT ARRANGEMENT
0075      C      FOR FFT COMPUTATION.
0076      CALL INTPL(A,PSD,NORG,T,X0,NDATA1,DT)
0077      C
0078      C.... TO COMPUTE HILBERT TRANSFORM OF ORIGINAL RECORD.
0079      CALL HLBRT(A,X0,XHIL,NDATA,0)
0080      C
0081      C.... TO COMPUTE FOURIER SPECTRUM OF ORIGINAL RECORD.
0082      CALL SPECT(A,X0,NDATA,TT,1)
0083      DO 101 I=1,NFOLD
0084      101 PSD(I)=A(I)
0085      C
0086      C.... GENERATION OF SAMPLE FUNCTIONS
0087      C..... FIRST, SET INITIAL VALUES FOR SUBROUTINE RANDU.
0088      IX=12347
0089      IY=0
0090      C
0091      C..... THEN, SPECIFY SAMPLE SIZE AND NUMBER OF PLOT ON ONE FRAME.
0092      WRITE(LU,620)
0093      620 FORMAT(" HOW MANY FIGS ON ONE FRAME -- NP"/
0094      * " HOW MANY SAMPLES ? -- NSMPL"/
0095      * " NSMPL MUST BE A MULTIPLE OF NP.")
0096      READ(LU,*)NP,NSMPL
0097      C
0098      C..... TO PREPARE FOR STORING PROCESS ON MAG-TAPE.
0099      C      IF ISSW(14) IS ON, THIS PROCESS IS SKIPPED.
0100      REWIND IIN
0101      IF(ISSW(14).LT.0) GO TO 161
0102      WRITE(IIN,162) NDATA,NSMPL,DT,DW,TT
0103      162 FORMAT(2I5,3E12.5)
0104      CALL EXEC(2,110B,T,NDATA2)
0105      CALL EXEC(2,110B,X0,NDATA2)
0106      C
0107      C..... TO SPECIFY TYPE OF RANDOM PHASE RAN.
0108      C      FOR GAUSSIAN CASE, SUPPLY MEAN AND STANDARD DEVIATION.
0109      C      (DEFAULT VALUES: MEAN=0.0, STDV=50*PI)
0110      161 AMEAN=0.0
0111      SFCTR=50.0
0112      WRITE(LU,625)
0113      625 FORMAT(" INPUT --- ICON,AMEAN,SFCTR")
0114      READ(LU,*) ICON,AMEAN,SFCTR
0115      STDV1=SFCTR*PI
0116      ISHEET=NSMPL/NP
0117      C
0118      C..... TO ASSIGN DEFAULT PARAMETER VALUES FOR PLOTTING.
0119      NX=1
0120      XMIN=T(1)

```



443 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 39 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0121      XMAX=TT
0122      NY=1
0123      YMIN=-200.
0124      YMAX=200.
0125      MO=-1
0126      JDC=22
0127      NW=1
0128      WMIN=0.0
0129      WMAX=W(NFOLD)
0130      NS=1
0131      SMAX=1.0E4
0132      SMIN=0.0
0133      MINX0=150
0134      MAXX0=950
0135      MINY0=150
0136      MAXY0=700
0137      DO 120 I=1,ISHEET
0138      DO 130 J=1,NP
0139      C
0140      C..... TO GENERATE RANDOM PHASE ANGLE RAN.
0141      IF(ICON.NE.3) GO TO 410
0142      C
0143      C..... FOR GAUSSIAN DISTRIBUTION.
0144      CALL JNORM(IX,IY,IX2,IY2,STDV1,STDV2,AMEAN,CORRL,
0145      *RAN,RAN2,-1,1)
0146      GO TO 420
0147      410 CALL RANDU(IX,IY,RAN)
0148      IF(ICON.NE.2) GO TO 430
0149      C
0150      C..... FOR UNIFORM DISTRIBUTION BETWEEN -PI/2 AND +PI/2.
0151      RAN=(RAN-0.5)*PI
0152      GO TO 420
0153      C
0154      C..... FOR UNIFORM DISTRIBUTION BETWEEN -PI AND +PI.
0155      430 RAN=(RAN-0.5)*PI2
0156      C
0157      C..... TO GENERATE SAMPLE FUNCTION OF THE FIRST KIND.
0158      420 CCOS=COS(RAN)
0159      SSIN=SIN(RAN)
0160      DO 140 K=1,NDATA
0161      140 X(K,J)=X0(K)*CCOS-XHIL(K)*SSIN
0162      IF(ISSW(14).LT.0) GO TO 130
0163      C
0164      C..... TO STORE SIMULATED PROCESSES ONTO MAGTAPE.
0165      CALL EXEC(2,110B,X(1,J),NDATA2)
0166      130 CONTINUE
0167      C
0168      C..... TO SKIP PLOTTING ROUTINE IF ISSW(15) IS ON.
0169      IF(ISSW(15).LT.0) GO TO 120
0170      C
0171      C..... TO PLOT SIMULATED PROCESSES.
0172      WRITE(LU,640)
0173      640 FORMAT(" TIME HISTORY PLOTTING"/
0174      *" INPUT---NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC"/
0175      *" ---MINX0,MAXX0,MINY0,MAXY0")
0176      10 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC,
0177      *MINX0,MAXX0,MINY0,MAXY0
0178      IF(NY.EQ.100) STOP
0179      IF(NY.LT.0) GO TO 11
0180      NP1=NP+1
0181      CALL STUKO(T,X0,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NP1,MO,

```

```

0182      *1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0183      DO 12 J=1,NP
0184      J1=J+1
0185      CALL STUKO(T,X(1,J),NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
0186      *NP1,MO,J1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0187      12 CONTINUE
0188      GO TO 10
0189      C
0190      C..... TO COMPUTE CORRESPONDING FOURIER SPECTRUM.
0191      11 DO 13 J=1,NP
0192      CALL SPECT(A,X(1,J),NDATA,TT,1)
0193      DO 13 K=1,NFOLD
0194      X(K,J)=A(K)
0195      13 CONTINUE
0196      C
0197      C.... TO PLOT FOURIER SPECTRUM.
0198      WRITE(LU,650)
0199      650 FORMAT(" FOURIER SPECTRUM PLOTTING"/
0200      *" INPUT---NS,SMIN,SMAX,NW,WMIN,WMAX,MO,JDC"/
0201      *" ---MINX0,MAXX0,MINY0,MAXY0")
0202      15 READ(LU,*) NS,SMIN,SMAX,NW,WMIN,WMAX,MO,JDC,
0203      *MINX0,MAXX0,MINY0,MAXY0
0204      IF(NS.EQ.100) STOP
0205      IF(NS.LT.0) GO TO 120
0206      CALL STUKO(W,PSD,NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,NP1,MO,
0207      *1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0208      DO 17 J=1,NP
0209      J1=J+1
0210      CALL STUKO(W,X(1,J),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,NP1,MO,
0211      *J1,LU,JDC,MINX0,MAXX0,MINY0,MAXY0)
0212      17 CONTINUE
0213      GO TO 15
0214      120 CONTINUE
0215      REWIND IIN
0216      STOP
0217      END
0218      END$

```

445 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 41 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

HSM30 T=00004 IS ON CR00002 USING 00038 BLKS R=0280

```

0001 FTN4
0002 C -----(MAIN PROGRAM SIM30)-----
0003 PROGRAM SIM30
0004 C.... THIS PROGRAM GETS AND PLOTS ENSEMBLE AS WELL AS TEMPORAL
0005 C STATISTICS OF THE DATA-BASED NONSTATIONARY RANDOM PROCESS
0006 C SUCH AS MAX, MIN, MEAN, STANDARD DEVIATION, PROBABILITY
0007 C DENSITY FUNCTION AT T=T1 (FOR ENSEMBLE) OR K=K1 (FOR
0008 C TEMPORAL).
0009 C
0010 C.... REQUIRED SUBROUTINES:
0011 C STUKO,STATC,HLBRT,FFT
0012 C.... REMARKS:
0013 C ISSW(13)=1 --- DENSITY PLOTTING AT T=T1 ONLY.
0014 C ISSW(14)=1 --- TEMPORAL STATISTIC PLOTTING IS SKIPPED.
0015 C ISSW(15)=1 --- THEORETICAL PDF (FIRST KIND) IS PLOTTED.
0016 C.... DATA FORMAT ON MAG-TAPE SHOULD BE:
0017 C NDATA,NSMPL,DT,DW,TT --- FORMAT(2I5,3E12.5)
0018 C T(I) ---CALL EXEC(2,110B,T,NDATA2)
0019 C X0(I) --- CALL EXEC(2,110B,X0,NDATA2)
0020 C X(I) --- CALL EXEC(2,110B,X,NDATA2)
0021 C
0022 DIMENSION FMAX(512),FMIN(512),FSTDV(512),FMEAN(512)
0023 DIMENSION EMAX(512),EMIN(512),ESTDV(512),EMEAN(512)
0024 DIMENSION T(512),X0(512),XSIM(512),T1(5),FFT0(1024)
0025 DIMENSION HIST(51,5),AMP(51,5),IPOS(5),XHIL(512),A0(5)
0026 EQUIVALENCE (EMAX(1),HIST),(EMIN(1),AMP),(XHIL,EMEAN)
0027 EQUIVALENCE (FFT0(1),FMAX),(FFT0(513),FMIN)
0028 C
0029 LU=1
0030 IIN=8
0031 PI=3.141593
0032 PI2=2.0*PI
0033 C
0034 C.... TO READ IN NECESSARY INFORMATION FROM MAGTAPE.
0035 REWIND IIN
0036 READ(IIN,600) NDATA,NSMPL,DT,DW,TT
0037 600 FORMAT(2I5,3E12.5)
0038 NDATA2=NDATA*2
0039 CALL EXEC(1,110B,T,NDATA2)
0040 CALL EXEC(1,110B,X0,NDATA2)
0041 NSMPL2=NSMPL*2
0042 ANDATA=FLOAT(NDATA)
0043 ANDAT1=ANDATA-1.0
0044 ANSMPL=FLOAT(NSMPL)
0045 ANSMPL1=ANSMPL-1.0
0046 NFOLD=NDATA/2+1
0047 DF=1.0/TT
0048 FNIQU1=DW*FLOAT(NFOLD)
0049 FNIQU2=DF*FLOAT(NFOLD)
0050 WRITE(6,610) TT,DT,DW,DF,FNIQU1,FNIQU2,NDATA,NFOLD
0051 610 FORMAT(1H1///3X,"SIMULATION OF NONSTATIONARY PROCESS"/
0052 1" WITH PHASE ANGLE HAVING UNIFORM DISTRIBUTION."///
0053 23X," TOTAL TIME PERIOD T=",E11.4/
0054 33X," TIME INCREMENT DT=",E11.4/
0055 43X," FREQUENCY INCREMENT DW=",E11.4/
0056 53X," DF=",E11.4/
0057 63X," NYQUIST FREQUENCY FNIQU1=",E11.4," (RAD./SEC)"/
0058 73X," FNIQU2=",E11.4," (1/SEC)"/
0059 83X," NUMBER OF DATA NDATA=",I5/

```

```

0060      93X," NYQUIST FOLD. NO. NFOLD=",J5///)
0061      C
0062      C.... TO COMPUTE AND PLOT STATISTICS (TEMPORAL AND OR ENSEMBLE).
0063      C      IF ISSW(13) IS ON, SKIP THIS PROCESS AND GO DIRECTLY TO
0064      C      THE COMPUTATION OF PROBABILITY DENSITY FUNCTION.
0065      C      FMAX,FMIN,FSTDV,FMEAN --- FOR ENSEMBLE.
0066      C      EMAX,EMIN,ESTDV,EMEAN --- FOR TEMPORAL.
0067      IF(ISSW(13).LT.0) GO TO 42
0068      DO 200 J=1,NDATA
0069      FMAX(J)=-1.0E20
0070      FMIN(J)=1.0E20
0071      FSTDV(J)=0.
0072      FMEAN(J)=0.
0073      200 CONTINUE
0074      DO 210 I=1,NSMPL
0075      CALL EXEC(1,110B,XSIM,NDATA2)
0076      CALL STATC(XSIM,NDATA,EMEAN(I),EVR,ESTDV(I),EMAX(I),EMIN(I))
0077      DO 220 J=1,NDATA
0078      DDJ=XSIM(J)
0079      FMEAN(J)=FMEAN(J)+DDJ
0080      FSTDV(J)=FSTDV(J)+DDJ*DDJ
0081      IF(DDJ.GT.FMAX(J)) FMAX(J)=DDJ
0082      IF(DDJ.LT.FMIN(J)) FMIN(J)=DDJ
0083      220 CONTINUE
0084      210 CONTINUE
0085      DO 230 J=1,NDATA
0086      FMEAN(J)=FMEAN(J)/ANSMPL
0087      FJD=FSTDV(J)/ANSMP1-FMEAN(J)**2
0088      IF(FJD.LT.0.) FJD=0.
0089      FSTDV(J)=SQRT(FJD)
0090      230 CONTINUE
0091      C
0092      C.... TO PLOT THE TEMPORAL INFORMATION.
0093      C      IF ISSW(14) IS ON, SKIP THIS PROCESS.
0094      IF(ISSW(14)) 32,2,2
0095      2 MO=-1
0096      NX=1
0097      XMIN=0.0
0098      XMAX=ANSMPL
0099      NY=1
0100      YMAX=200.
0101      YMIN=-200.
0102      NP=4
0103      N=NSMPL
0104      JDC=22
0105      MINX=150
0106      MAXX=950
0107      MINY=150
0108      MAXY=700
0109      DO 240 I=1,NSMPL
0110      240 XSIM(1)=FLOAT(I)
0111      WRITE(LU,630)
0112      630 FORMAT(" INPUT=-NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC"
0113      */"      --MINX,MAXX,MINY,MAXY")
0114      30 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC,
0115      *MINX,MAXX,MINY,MAXY
0116      IF(NY.EQ.100) STOP
0117      IF(NY.LT.0) GO TO 32
0118      CALL STUKO(XSIM,EMEAN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0119      INP,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
0120      CALL STUKO(XSIM,ESTDV,N,NX,XMIN,XMAX,NY,YMIN,YMAX,

```

447 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 43 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0121      2NP,MO,2,LU,JDC,MINX,MAXX,MINY,MAXY)
0122      CALL STUKO(XSIM,EMAX,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0123      INP,MO,3,LU,JDC,MINX,MAXX,MINY,MAXY)
0124      CALL STUKO(XSIM,EMIN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0125      INP,MO,4,LU,JDC,MINX,MAXX,MINY,MAXY)
0126      GO TO 30
0127  C
0128  C.... TO PLOT ENSEMBLE STATISTICS AT T=T1.
0129  C      T1 CAN BE SPECIFIED ARBITRARILY (UP TO 5 CASES).
0130      32 MO=-1
0131      NX=1
0132      XMIN=0.
0133      XMAX=40.
0134      NY=1
0135      YMIN=-200.
0136      YMAX=200.
0137      NP=5
0138      N=NDATA
0139      JDC=22
0140      MAXX=950
0141      MAXY=700
0142      MINX=150
0143      MINY=150
0144      WRITE(LU,630)
0145      40 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDC
0146      *,MINX,MAXX,MINY,MAXY
0147      IF(NY.EQ.100) STOP
0148      IF(NY.LT.0) GO TO 42
0149      CALL STUKO(T,X0,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0150      INP,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
0151      CALL STUKO(T,FMEAN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0152      INP,MO,2,LU,JDC,MINX,MAXX,MINY,MAXY)
0153      CALL STUKO(T,FSTDV,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0154      INP,MO,3,LU,JDC,MINX,MAXX,MINY,MAXY)
0155      CALL STUKO(T,FMAX,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0156      2NP,MO,4,LU,JDC,MINX,MAXX,MINY,MAXY)
0157      CALL STUKO(T,FMIN,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0158      INP,MO,5,LU,JDC,MINX,MAXX,MINY,MAXY)
0159      GO TO 40
0160  C
0161  C.... TO PLOT PROBABILITY DENSITY AT T=T1.
0162  C      USER MUST SPECIFY T1 VALUE THROUGH THE TERMINAL.
0163      42*REWIND IIN
0164      JDC=22
0165      MINX=150
0166      MAXX=950
0167      MINY=150
0168      MAXY=700
0169      WRITE(LU,640)
0170      640 FORMAT(" INPUT THE VALUES RELATED TO HISTOGRAMS."/
0171      1" NSTEP -- NUMBER OF STEPS (UP TO 51)"/
0172      2"          MUST BE ODD INTEGER VALUE."/
0173      3" NT1   -- NUMBER OF SPECIFIC TIMES REQUIRED (UP TO 10)")
0174      READ(LU,*) NSTEP,NT1
0175      WRITE(LU,650)
0176      650 FORMAT(" INPUT -- T1 = SPECIFIC TIME TO BE CONSIDERED"/
0177      1"          MUST BE NT1 NUMBERS."/)
0178      READ(LU,*)(T1(I),I=1,NT1)
0179      DO 250 I=1,NT1
0180      IPOS1=IFIX(T1(I)/DT)
0181      IPOS2=IPOS1+1

```

```

0182      IF((T1(I)-T(IPOS1)).GT.(T(IPOS2)-T1(I))) IPOS1=IPOS2
0183      IPOS(I)=IPOS1
0184      WRITE(6,660) T1(I),IPOS1,T(IPOS1)
0185      660 FORMAT(3X," T1=",F7,3,2X," I=",I4,2X," TRUE T=",F7,3)
0186      250 CONTINUE
0187  C
0188  C.... DUMMY READING OF MAGTAPE TO GET THE CORRECT DATA POSITION.
0189      REWIND IIN
0190      READ(IIN,600)NDMY1,NDMY2,DY1,DY2,DY3
0191      DO 270 I=1,2
0192      270 CALL EXEC(1,110B,XSIM,NDATA2)
0193  C
0194  C.... TO COMPUTE HILBERT TRANSFORM XHIL(T) OF X0(T).
0195      CALL HLBRT(FFTO,X0,XHIL,NDATA,0)
0196  C
0197  C.... TO SPECIFY AMPLITUDE OF EACH STEP
0198      NSTEP2=NSTEP/2
0199      DO 280 J=1,NT1
0200          IPOSJ=IPOS(J)
0201          A0(J)=SQRT(X0(IPOSJ)**2+XHIL(IPOSJ)**2)
0202          DLTJ=A0(J)*2.0/FLOAT(NSTEP-1)
0203          DO 280 I=1,NSTEP
0204              AMP(I,J)=DLTJ*FLOAT(I-NSTEP2-1)
0205      280 CONTINUE
0206          DO 282 J=1,NT1
0207              DO 282 I=1,NSTEP
0208                  282 HIST(I,J)=0.0
0209  C
0210  C.... TO COUNT THE FREQUENCY FOR EACH STEP.
0211      DO 290 I=1,NSMPL
0212          CALL EXEC(1,110B,XSIM,NDATA2)
0213          DO 300 J=1,NT1
0214              IPOSJ=IPOS(J)
0215              DDJ=XSIM(IPOSJ)
0216              DLTJ2=(AMP(NSTEP,J)-AMP(1,J))/(2.0*FLOAT(NSTEP-1))
0217              DFREQ=1.0/(ANSMPL*2.0*DLTJ2)
0218              IF(DDJ.GT.AMP(1,J)) GO TO 301
0219              HIST(1,J)=HIST(1,J)+DFREQ
0220              GO TO 300
0221      301 IF(DDJ.LT.AMP(NSTEP,J)) GO TO 302
0222              HIST(NSTEP,J)=HIST(NSTEP,J)+DFREQ
0223              GO TO 300
0224      302 DO 310 K=1,NSTEP
0225              IF(ABS(DDJ-AMP(K,J)).GT.DLTJ2) GO TO 310
0226              HIST(K,J)=HIST(K,J)+DFREQ
0227              GO TO 300
0228      310 CONTINUE
0229      300 CONTINUE
0230      290 CONTINUE
0231      MO=0
0232      NX=1
0233      XMIN=-200.0
0234      XMAX=200.0
0235      NY=1
0236      YMIN=0.
0237      YMAX=0.5
0238      N=NSTEP
0239      NP=NT1
0240      IF(ISSW(15))320,321,321
0241      321 WRITE(LU,630)
0242      60 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO

```

449 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 45 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0243      *,JDC,MINX,MAXX,MINY,MAXY
0244      IF(NY.EQ.100) STOP
0245      IF(NY.LT.0) GO TO 320
0246      DO 350 I=1,NP
0247      CALL STUKO(AMP(1,I),HIST(1,I),N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0248      1NP,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
0249      350 CONTINUE
0250      GO TO 60
0251      320 DO 750 I=1,NP
0252      WRITE(LU,630)
0253      MO=0
0254      NP0=NP
0255      61 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP0,MO,JDC,
0256      *MINX,MAXX,MINY,MAXY
0257      IF(NY.EQ.100) STOP
0258      IF(NY.LT.0) GO TO 750
0259      DDJ=2.0*A0(I)/ANDATA
0260      N1=NDATA-1
0261      DO 330 J=1,N1
0262      FMAX(J)=-A0(I)+DDJ*FLOAT(J)
0263      ESTDV(J)=1.0/(PI*SQRT(A0(I)**2-FMAX(J)**2))
0264      330 CONTINUE
0265      CALL STUKO(FMAX,ESTDV,N1,NX,XMIN,XMAX,NY,YMIN,YMAX,
0266      *2,MO,1,LU,JDC,MINX,MAXX,MINY,MAXY)
0267      CALL STUKO(AMP(1,I),HIST(1,I),N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0268      *2,MO,2,LU,JDC,MINX,MAXX,MINY,MAXY)
0269      GO TO 61
0270      750 CONTINUE
0271      62 REWIND IIN
0272      WRITE(LU,680)
0273      680 FORMAT(" NEED ANOTHER TRIAL FOR T1 ?"/
0274      1" IF YES ----- TYPE 1"/
0275      2" IF NO ----- -1"/)
0276      READ(LU,*) IJDG
0277      IF(IJDG) 64,42,42
0278      64 STOP
0279      END
0280      ENDS

```

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

```

0001 FTN4
0002 C -----(MAIN PROGRAM SIM40)-----
0003 C PROGRAM SIM40
0004 C.... THIS PROGRAM SIMULATES AND PLOTS DATA-BASED BIVARIATE
0005 C NONSTATIONARY RANDOM PROCESS OF THE FIRST KIND AS WELL AS
0006 C THEIR FOURIER AMPLITUDE SPECTRA WITH RANDOM PHASE ANGLE
0007 C BEING JOINT NORMAL DISTRIBUTION.
0008 C
0009 C.... REQUIRED SUBROUTINES:
0010 C STUKO,SPECT,RANDU,INTPL,JNORM,FFT,HLBRT
0011 C.... REMARKS:
0012 C BIVARIATE ORIGINAL RECORD MUST BE USED.
0013 C
0014 C DIMENSION A(1024),X01(513),X02(513),XHIL1(513),XHIL2(513)
0015 C *,T(513),XSIM2(514),XNS(513,2),XEW(513,2),W(257),SPD(257,2)
0016 C EQUIVALENCE (XSIM2,SPD),(W,A(1)),(T,A(511))
0017 C LU=1
0018 C WRITE(LU,601)
0019 C 601 FORMAT(/" INPUT THE FOLLOWING VALUES."/
0020 C *" NORG1 -- NUMBER OF NS DATA"/
0021 C *" NORG2 -- NUMBER OF EW DATA"/
0022 C *" NDATA -- NUMBER OF INTERPOLATED DATA"/)
0023 C READ(1,*) NORG1,NORG2,NDATA
0024 C
0025 C TO READ THE ORIGINAL TIME SERIES
0026 C REWIND 8
0027 C READ(8,300)(A(I),XHIL1(I),I=1,NORG1)
0028 C 300 FORMAT(6F10.3)
0029 C READ(8,300)(XSIM2(I),XHIL2(I),I=1,NORG2)
0030 C REWIND 8
0031 C
0032 C NFOLD=NDATA/2+1
0033 C NDATA1=NDATA+1
0034 C TT=A(NORG1)-A(1)
0035 C DT=TT/FLOAT(NDATA)
0036 C PI =3.1415926
0037 C PI2=PI*2.0
0038 C DW=PI2/TT
0039 C
0040 C.... INTERPOLATION OF THE ORIGINAL TIME SERIES
0041 C CALL INTPL(A,XHIL1,NORG1,T,X01,NDATA1,DT)
0042 C CALL INTPL(XSIM2,XHIL2,NORG2,T,X02,NDATA1,DT)
0043 C
0044 C.... TO COMPUTE HILBERT TRANSFORM OF ORIGINAL TIME HISTORY.
0045 C CALL HLBRT(A,X01,XHIL1,NDATA,0)
0046 C CALL HLBRT(A,X02,XHIL2,NDATA,0)
0047 C
0048 C.... TO COMPUTE FOURIER AMPLITUDE SPECTRUM OF X0(T).
0049 C CALL SPECT(A,X01,NDATA,TT,1)
0050 C DO 700 I=1,NFOLD
0051 C 700 SPD(I,1)=A(I)
0052 C CALL SPECT(A,X02,NDATA,TT,1)
0053 C DO 710 I=1,NFOLD
0054 C 710 SPD(I,2)=A(I)
0055 C
0056 C.... TO GENERATE JOINT GAUSSIAN RANDOM PHASE ANGLE.
0057 C SFCTR1=50.0
0058 C CORR1=0.999
0059 C AM1=0.

```



## 451 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES — 47 —

```

0060      NSMPL=500
0061      WRITE(LU,880)
0062  880 FORMAT(" INPUT--- NSMPL,CORRL,SFCTR1,AM1")
0063      READ(LU,*)NSMPL,CORRL,SFCTR1,AM1
0064      STDV1=SFCTR1*PI
0065      STDV2=STDV1
0066      NSMPL2=NSMPL/2
0067      MO=-1
0068      NX=1
0069      XMAX=TT
0070      XMIN=T(1)
0071      NY=1
0072      YMIN=-200.
0073      YMAX=200.
0074      NP=20303
0075      MINX0=50
0076      MAXX0=1020
0077      MINY0=150
0078      MAXY0=680
0079      JDCXY=0
0080      NS=1
0081      SMIN=0.0
0082      SMAX=500.
0083      NW=1
0084      WMIN=0.
0085      WMAX=DW*FLOAT(NFOLD=1)
0086      IX1=12347
0087      IY1=0
0088      IX2=30011
0089      IY2=0
0090      DO 250 IJK=1,NSMPL2
0091      DO 255 KK=1,2
0092      CALL JNORM(IX1,IY1,IX2,IY2,STDV1,STDV2,AM1,CORRL ,
0093      *RAN1,RAN2,-1)
0094      CCOS1=COS(RAN1)
0095      SSIN1=SIN(RAN1)
0096      CCOS2=COS(RAN2)
0097      SSIN2=SIN(RAN2)
0098      DO 220 J=1,NDATA
0099      XNS(J,KK)=X01(J)*CCOS1-XHIL1(J)*SSIN1
0100      XEW(J,KK)=X02(J)*CCOS2-XHIL2(J)*SSIN2
0101  220 CONTINUE
0102  255 CONTINUE
0103      JREP=-1
0104      WRITE(LU,555)
0105  555 FORMAT(" INPUT--- NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO"/
0106      *" INPUT--- JDCXY,MINX0,MAXX0,MINY0,MAXY0")
0107      10 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,
0108      *JDCXY,MINX0,MAXX0,MINY0,MAXY0
0109      IF(NY.EQ.100) STOP
0110      IF(NY.LT.0) GO TO 249
0111      IF(JREP.GT.0) GO TO 887
0112      DO 888 I=1,NDATA
0113      888 T(I)=DT*FLOAT(I-1)
0114      887 CALL STUKO(T,X01,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
0115      *NP,MO,1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0116      DO 333 J=1,2
0117      J1=J+1
0118      CALL STUKO(T,XNS(1,J),NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
0119      *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0120      333 CONTINUE

```

```

0121      CALL STUKO(T,X02,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
0122      *NP,MO,4,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0123      DO 334 J=1,2
0124      J1=J+4
0125      CALL STUKO(T,XEW(1,J),NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
0126      *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0127 334 CONTINUE
0128      JREP=1
0129      GO TO 10
0130 249 WRITE(LU,555)
0131      JREP=-1
0132      20 READ(LU,*)NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,JDCXY,
0133      *MINX0,MAXX0,MINY0,MAXY0
0134      IF(NS.EQ.100) STOP
0135      IF(NS.LT.0) GO TO 250
0136      IF(JREP.GT.0) GO TO 891
0137      DO 890 I=1,NFOLD
0138      890 W(I)=DW*FLOAT(I-1)
0139      891 CALL STUKO(W,SPD(1,1),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
0140      *NP,MO,1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0141      DO 400 J=1,2
0142      IF(JREP.GT.0) GO TO 892
0143      CALL SPECT(A,XNS(1,J),NDATA,TT,1)
0144      DO 401 KK=1,NFOLD
0145      401 XNS(KK,J)=A(KK)
0146      DO 900 I=1,NFOLD
0147      900 W(I)=DW*FLOAT(I-1)
0148      892 J1=J+1
0149      CALL STUKO(W,XNS(1,J),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
0150      *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0151      400 CONTINUE
0152      CALL STUKO(W,SPD(1,2),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
0153      *NP,MO,4,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0154      DO 410 J=1,2
0155      IF(JREP.GT.0) GO TO 893
0156      CALL SPECT(A,XEW(1,J),NDATA,TT,1)
0157      DO 402 KK=1,NFOLD
0158      402 XEW(KK,J)=A(KK)
0159      DO 910 I=1,NFOLD
0160      910 W(I)=DW*FLOAT(I-1)
0161      893 J1=J+4
0162      CALL STUKO(W,XEW(1,J),NFOLD,NW,WMIN,WMAX,NS,SMIN,SMAX,
0163      *NP,MO,J1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0164      410 CONTINUE
0165      JREP=1
0166      GO TO 20
0167 250 CONTINUE
0168      STOP
0169      END
0170      END$

```

453 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 49 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

```

0001 FTN4
0002 C -----(MAIN PROGRAM SIM50)-----
0003 PROGRAM SIM50
0004 C.... THIS PROGRAM COMPUTES AND PLOTS CROSS CORRELATION FUNCTION
0005 C BETWEEN X1(T) AND X2(T) SIMULATED BY DATA-BASED BIVARIATE
0006 C NONSTATIONARY PROCESS SIMULATION METHOD OF THE FIRST KIND
0007 C WITH RANDOM PHASE ANGLE BEING JOINT GAUSSIAN.
0008 C
0009 C.... REQUIRED SUBROUTINES:
0010 C STUKO,RANDU,HLBRT,FFT,INTPL,JNORM
0011 C
0012 DIMENSION A(1024),X01(512),X02(512),XHIL1(512),XHIL2(512),
0013 *T(512),XSIM2(512),XYREAL(512,2),XYIMAG(512,2),IPOS(2),TPOS(2)
0014 EQUIVALENCE (A(513),T),(A(1),XSIM2)
0015 C
0016 LU=1
0017 WRITE(LU,601)
0018 601 FORMAT("/" INPUT THE FOLLOWING VALUES,"/
0019 *" NORG1 -- NUMBER OF NS DATA"/
0020 *" NORG2 -- NUMBER OF EW DATA"/
0021 *" NDATA -- NUMBER OF INTERPOLATED DATA"/)
0022 READ(LU,*) NORG1,NORG2,NDATA
0023 C
0024 C.... TO READ THE ORIGINAL TIME SERIES
0025 REWIND 8
0026 READ(8,300)(A(I),XHIL1(I),I=1,NORG1)
0027 300 FORMAT(6F10,3)
0028 READ(8,300)(XSIM2(I),XHIL2(I),I=1,NORG2)
0029 REWIND 8
0030 C
0031 NFOLD=NDATA/2+1
0032 NDATA1=NDATA+1
0033 TT=A(NORG1)-A(1)
0034 DT=TT/FLOAT(NDATA)
0035 PI =3.1415926
0036 PI2=PI*2.0
0037 DW=PI2/TT
0038 C
0039 C.... INTERPOLATION OF THE ORIGINAL TIME SERIES
0040 CALL INTPL(A,XHIL1,NORG1,XYREAL(1,1),X01,NDATA,DT)
0041 CALL INTPL(XSIM2,XHIL2,NORG2,XYREAL(1,1),X02,NDATA,DT)
0042 C
0043 C.... TO COMPUTE HILBERT TRANSFORM XHIL(T) OF X0(T).
0044 CALL HLBRT(A,X01,XHIL1,NDATA,0)
0045 CALL HLBRT(A,X02,XHIL2,NDATA,0)
0046 C
0047 C.... TO CALCULATE EXPECTED VALUE SUCH AS
0048 C.... EXP(I*(PHAI1-PHAI2)),ETC.
0049 30 NSMPL=500
0050 IX1=12347
0051 IY1=0
0052 IX2=30011
0053 IY2=0
0054 CORRL=0.
0055 AM1=0.
0056 SFCTR1=50.0
0057 WRITE(LU,650)
0058 650 FORMAT(" INPUT-- NSMPL,CORRL,SFCTR1,AM1")
0059 READ(LU,*)NSMPL,CORRL,SFCTR1,AM1

```

```

0060      STDV1=PI*SFCTR1
0061      STDV2=STDV1
0062      CCOSP=0.
0063      CCOSM=0.
0064      SSINP=0.
0065      SSINM=0.
0066      DO 200 I=1,NSMPL
0067      CALL JNORM(IX1,IY1,IX2,IY2,STDV1,STDV2,AM1,CORRL,
0068      *RAN1,RAN2,1)
0069      RANP=RAN1+RAN2
0070      RANM=RAN1-RAN2
0071      CCOSP=CCOSP+COS(RANP)
0072      CCOSM=CCOSM+COS(RANM)
0073      SSINP=SSINP+SIN(RANP)
0074      SSINM=SSINM+SIN(RANM)
0075      200 CONTINUE
0076      ANSMPL=FLOAT(NSMPL)
0077      CCOSP=CCOSP/ANSMPL
0078      CCOSM=CCOSM/ANSMPL
0079      SSINP=SSINP/ANSMPL
0080      SSINM=SSINM/ANSMPL
0081      WRITE(6,670)CCOSP,CCOSM,SSINP,SSINM
0082      670 FORMAT(1H1////
0083      *"   EXPECTED VALUE CCOSP=",E12.5/
0084      *"   EXPECTED VALUE CCOSM=",E12.5/
0085      *"   EXPECTED VALUE SSINP=",E12.5/
0086      *"   EXPECTED VALUE SSINM=",E12.5/)
0087      C
0088      C.... CALCULATION OF CROSS CORRELATION.
0089      WRITE(LU,680)
0090      680 FORMAT(" INPUT--- NT1 (UP TO 2)")
0091      READ(LU,*) NT1
0092      WRITE(LU,681)
0093      681 FORMAT(" INPUT--- TIME REQUIRED (UP TO NT1)")
0094      READ(LU,*)(TPOS(I),I=1,NT1)
0095      DO 222 I=1,NT1
0096      IIPOS=IFIX(TPOS(I)/DT)
0097      IF(T(IIPOS+1)-TPOS(I).LT,TPOS(I)-T(IIPOS)) IIPOS=IIPOS+1
0098      IPOS(I)=IIPOS
0099      WRITE(6,690)IIPOS,T(IIPOS)
0100      690 FORMAT(" IPOS=",I4,"   TIME=",E12.5)
0101      222 CONTINUE
0102      EXPCON=0.5*EXP(-2.0*STDV1**2)
0103      DO 220 I=1,NT1
0104      INT=TPOS(I)
0105      FCTR1=X02(INT)
0106      FCTR2=XHIL2(INT)
0107      DO 230 J=1,NDATA
0108      FC1=X01(J)
0109      FC2=XHIL1(J)
0110      XYREAL(J,I)=FNCTR(CCOSM,SSINM,FC1,FC2,FCTR1,-FCTR2)
0111      **FNCTR(CCOSP,-SSINP,FC1,-FC2,FCTR1,-FCTR2)
0112      **FNCTR(CCOSM,-SSINM,FC1,-FC2,FCTR1,FCTR2)
0113      **FNCTR(CCOSP,SSINP,FC1,FC2,FCTR1,FCTR2)
0114      XYINAG(J,I)=0.5*(FC1*FCTR1+FC2*FCTR2)+
0115      *EXPCON*(FC1*FCTR1-FC2*FCTR2)
0116      XYREAL(J,I)=XYREAL(J,I)/4.0
0117      230 CONTINUE
0118      220 CONTINUE
0119      NX=1
0120      XMIN=0.0

```

455 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 51 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0121      XMAX=DT*FLOAT(NDATA)
0122      NY=1
0123      YMAX=1.E6
0124      YMIN=-1.E6
0125      MO=-1
0126      NP=2*NT1
0127      JDC=22
0128      MINX=50
0129      MAXX=1020
0130      MINY=150
0131      MAXY=700
0132      WRITE(LU,555)
0133 555 FORMAT(" INPUT--- NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO"/
0134      *" INPUT--- JDC,MINX,MAXX,MINY,MAXY")
0135      10 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO
0136      *,JDC,MINX,MAXX,MINY,MAXY
0137      IF(NY.EQ.100) STOP
0138      IF(NY.LT.0) GO TO 20
0139      DO 350 K=1,NDATA
0140 350 T(I)=DT*FLOAT(I-1)
0141      DO 333 K=1,NT1
0142 333 CALL STUKO(T,XYIMAG(1,K),NDATA,NX,XMIN,XMAX,
0143      *NY,YMIN,YMAX,NP,MO,K,LU,JDC,MINX,MAXX,MINY,MAXY)
0144      DO 334 K=1,NT1
0145      K1=K+NT1
0146 334 CALL STUKO(T,XYREAL(1,K),NDATA,NX,XMIN,XMAX,
0147      *NY,YMIN,YMAX,NP,MO,K1,LU,JDC,MINX,MAXX,MINY,MAXY)
0148      GO TO 10
0149      20 WRITE(LU,888)
0150 888 FORMAT(" ANOTHER TRIAL?")
0151      READ(LU,*)IJD
0152      IF(IJD.GT.0) GO TO 30
0153      STOP
0154      END
0155      FUNCTION FNCTR(A,B,C,D,E,F)
0156      FNCTR=E*(A*C-B*D)-F*(A*D+B*C)
0157      RETURN
0158      END
0159      ENDS

```

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

```

0001 FTN4
0002 C -----(MAIN PROGRAM SIM60)-----
0003 C PROGRAM SIM60
0004 C.... THIS PROGRAM GENERATES AND PLOTS SAMPLE FUNCTIONS OF
0005 C DATA-BASED NONSTATIONARY RANDOM PROCESS OF THE SECOND KIND
0006 C WITH UNIFORM RANDOM PHASE ANGLE BETWEEN  $\pi/2$  AND  $\pi/2$ .
0007 C CORRESPONDING FOURIER SPECTRA AND STATISTICAL QUANTITIES
0008 C ARE ALSO COMPUTED AND PLOTTED.
0009 C -----
0010 C DATA-BASED NONSTATIONARY RANDOM PROCESS OF THE SECOND KIND
0011 C -----
0012 C
0013 C.... REQUIRED SUBROUTINES:
0014 C STUKO, RANDU, SPECT, ARNGE, HLBRT, INTPL, FFT
0015 C.... REMARKS:
0016 C THIS PROGRAM IS CODED FOR HP MINI-COMPUTER WITH TEKTRONIX
0017 C TERMINAL CONTROL SYSTEM FOR PLOTTING. THEREFORE, FOR IBM
0018 C COMPUTER SLIGHT MODIFICATION IS NEEDED.
0019 C BECAUSE OF THE LIMITED MEMORY SIZE IN MINI-COMPUTER,
0020 C "NDATA" IS ASSIGNED TO BE AT MOST 512.
0021 C.... REQUIREMENT OF ARRAY SIZE:
0022 C A(N+1),F0(2*N+2),T(N+1),X0(N+1),XHIL(N+1),SW(N+1)
0023 C W(N/2+1),RV(NSTEP),WA(N/2+1),WM(N/2+1)
0024 C WHERE N=NDATA=A POWER OF TWO FOR FFT COMPUTATION,
0025 C NSTEP=# OF AMPLITUDE LEVEL FOR PDF COMPUTATION.
0026 C
0027 C DIMENSION A(515),F0(1030),T(515),X0(515),XHIL(515)
0028 C DIMENSION SW(515),W(257),RV(81),WA(257),WM(257)
0029 C
0030 C.... TO ASSIGN NUMBER OF INTERPOLATED DATA FOR FFT COMPUTATION.
0031 C DATA NDATA/512/
0032 C
0033 C.... TO ASSIGN SELECT CODE OF INPUT/OUTPUT PERIPHERAL.
0034 C LU=1
0035 C IIN=8
0036 C IO=6
0037 C
0038 C.... TO READ IN OBSERVED RECORD FROM MAG-TAPE.
0039 C NORG -- NUMBER OF DATA
0040 C F0(I) -- TIME
0041 C A(I) -- OBSERVED VALUE
0042 C REWIND IIN
0043 C WRITE(LU,600)
0044 C 600 FORMAT(" INPUT -- NORG= NO. OF ORIGINAL DATA")
0045 C READ(LU,*) NORG
0046 C READ(IIN,100) (F0(I),A(I),I=1,NORG)
0047 C 100 FORMAT(6F10,3)
0048 C REWIND IIN
0049 C
0050 C.... TO PREPARE APPROPRIATE QUANTITIES FOR LATER COMPUTATION.
0051 C NDATA1=NDATA+1
0052 C ND2=NDATA/2
0053 C ND4=ND2/2
0054 C NFOLD=ND2+1
0055 C TFST=F0(1)
0056 C TTOTAL=F0(NORG)-TFST
0057 C DT=TTOTAL/FLOAT(ND2)
0058 C PI=3.141593
0059 C DW=2.0*PI/TTOTAL

```

457 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 53 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0060      DO 350 JJ=1,NFOLD
0061      350 W(JJ)=DW*(JJ-1)
0062      C
0063      C.... TO ADJUST TIME AXIS FOR DATA TO START FROM T=0.0 (SEC).
0064      DO 352 I=1,NORG
0065      352 F0(I)=F0(I)-TFST
0066      C
0067      C.... TO INTERPOLATE ORIGINAL RECORD TO BE ARRANGED EQUIDISTANTLY.
0068      CALL INTPL(F0,A,NORG,T,X0,ND2,DT)
0069      X0(NFOLD)=0.0
0070      C
0071      C.... TO GET SYMMETRIC-PERIODIC EXTENSION OF THE ORIGINAL RECORD
0072      C AND ALSO COMPUTE TIME ARRAY.
0073      CALL ARNGE(X0,NDATA,10,1)
0074      DO 330 I=1,NDATA1
0075      330 T(I)=DT*FLOAT(I-NFOLD)
0076      C
0077      C.... TO COMPUTE HILBERT TRANSFORM OF SYMMETRIC-PERIODIC
0078      C EXTENSION OF THE ORIGINAL RECORD.
0079      CALL HLBRT(F0,X0,XHIL,NDATA,0)
0080      C
0081      C.... TO PLOT ORIGINAL RECORD AS WELL AS HILBERT TRANSFORM OF
0082      C ITS SYMMETRIC-PERIODIC EXTENSION.
0083      WRITE(LU,610)
0084      610 FORMAT(" NEED TO PLOT HILBERT TRANSFORM ?"/
0085      * " IF YES, TYPE IN +1")
0086      READ(LU,*) IHIL
0087      NX=1
0088      XMIN=-TTOTAL
0089      XMAX=TTOTAL
0090      NY=1
0091      YMIN=-200.0
0092      YMAX=200.0
0093      MINX0=150
0094      MAXX0=950
0095      MINY0=150
0096      MAXY0=700
0097      JDCXY=22
0098      NP=2
0099      MO=-1
0100      IF(IHIL.NE. 1) GO TO 1300
0101      WRITE(LU,620)
0102      620 FORMAT(" INPUT-- NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY"/
0103      * " -- MINX0,MAXX0,MINY0,MAXY0")
0104      1301 READ(LU,*) NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY,
0105      *MINX0,MAXX0,MINY0,MAXY0
0106      IF(NY.EQ. 100) STOP
0107      IF(NY.LT. 0) GO TO 1300
0108      CALL STUKO(T,X0,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,1,
0109      *LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0110      CALL STUKO(T,XHIL,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,
0111      *NP,MO,2,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0112      GO TO 1301
0113      C
0114      C.... TO COMPUTE AND PLOT FOURIER AMPLITUDE OF THE ORIGINAL
0115      C RECORD, SQRT(AMPL(P0(W))**2+AMPL(Q0(W))**2), AMPL(P0(W)),
0116      C AND AMPL(Q0(W)), WHERE SYMBOL AMPL(.) REPRESENTS AMPLITUDE
0117      C OF A COMPLEX VALUE.
0118      1300 NW=1
0119      WMIN=W(1)
0120      WMAX=W(NFOLD)

```

```

0121      NS=1
0122      SMAX=1000.
0123      SMIN=0.0
0124      JDCWS=11
0125      NP=4
0126      MO=-1
0127      MINW0=350
0128      MAXW0=800
0129      MINS0=40
0130      MAXS0=770
0131      JPS=1
0132      NDF=ND2+1
0133      IF(IHIL,NE.1) GO TO 1600
0134      WRITE(LU,630)
0135      630 FORMAT(* INPUT-- NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,JDCWS"/
0136      *      -- MINW0,MAXW0,MINS0,MAXS0")
0137      1650 READ(LU,*) NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,
0138      *JDCWS,MINW0,MAXW0,MINS0,MAXS0
0139      IF(NS,EO.100) STOP
0140      IF(NS,LT.0) GO TO 1600
0141      C
0142      C..... TO COMPUTE FOURIER SPECTRUM OF THE ORIGINAL RECORD.
0143      DO 280 I=1,ND2
0144      F0(2*I-1)=X0(I+ND2)
0145      280 F0(2*I)=0.0
0146      CALL FFT0(F0,A,ND2,-1,1)
0147      DO 290 I=1,ND2
0148      290 SW(I)=SQRT(F0(I*2-1)**2+F0(I*2)**2)*TTOTAL
0149      C
0150      C..... TO CHECK THE AREA.
0151      CALL AREA(AW,SW,ND4,DW)
0152      RV(1)=AW/PI
0153      1400 CALL STUKO(W,SW,NDF,NW,WMIN,WMAX,NS,SMIN,SMAX,NP,MO,
0154      *      JPS,LU,JDCWS,MINW0,MAXW0,MINS0,MAXS0)
0155      IF(JPS,EO.2) GO TO 1630
0156      IF(JPS,EO.3) GO TO 1631
0157      IF(JPS,EO.4) WRITE(IO,66) (RV(LK),LK=1,4)
0158      66 FORMAT(///1H,4E12,5)
0159      IF(JPS,EO.4) GO TO 1650
0160      C
0161      C.... PRELIMINARY ANALYSIS TO GET P0(W) AND Q0(W).
0162      C.... FIRST COMPUTE P0(W).
0163      DO 250 I=1,ND2
0164      F0(2*I-1)=X0(I+ND2)*0.5
0165      250 F0(2*I)=-XHIL(I+ND2)*0.5
0166      CALL FFT0(F0,A,ND2,-1,1)
0167      C..... TO COMPUTE AMP(P0(W))**2.
0168      DO 230 I=1,ND2
0169      230 WA(I)=(F0(2*I-1)**2+F0(2*I)**2)*TTOTAL*TTOTAL
0170      C
0171      C.... SECOND, COMPUTE Q0(W).
0172      DO 260 I=1,ND2
0173      F0(2*I-1)=X0(I+ND2)*0.5
0174      260 F0(2*I)=XHIL(I+ND2)*0.5
0175      CALL FFT0(F0,A,ND2,-1,1)
0176      C..... TO COMPUTE AMPL(Q0(W))**2.
0177      DO 270 I=1,ND2
0178      270 WM(I)=(F0(2*I-1)**2+F0(2*I)**2)*TTOTAL*TTOTAL
0179      C
0180      C.... TO COMPUTE SQRT(AMPL(P0(W))**2+AMPL(Q0(W))**2)
0181      C      AND ALSO CHECK THE AREA.

```



459 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 55 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0182      DO 271 I=1,ND2
0183 271    SW(I)=SQRT(WA(I)+WM(I))
0184      CALL AREA(AW,SW,ND4,DW)
0185      RV(2)=AW/PI
0186      JPS=JPS+1
0187      GO TO 1400
0188 C
0189 C.... TO COMPUTE AMPL(P0(W)) AND CHECK THE AREA.
0190 1630    DO 272 I=1,ND2
0191 272    SW(I)=SQRT(WA(I))
0192      CALL AREA(AW,SW,ND4,DW)
0193      RV(3)=AW/PI
0194      JPS=JPS+1
0195      GO TO 1400
0196 C
0197 C.... TO COMPUTE AMPL(Q0(W)) AND CHECK THE AREA.
0198 1631    DO 273 I=1,ND2
0199 273    SW(I)=SQRT(WM(I))
0200      CALL AREA(AW,SW,ND4,DW)
0201      RV(4)=AW/PI
0202      JPS=JPS+1
0203      GO TO 1400
0204 C
0205 C.... TO PRINT OUT COMPUTED AREA FOR CHECK PURPOSE.
0206 1600 WRITE(IO,66) (RV(L3),L3=1,4)
0207 C
0208 C -----
0209 C.... TO COMPUTE AND PLOT SIMULATED DATA-BASED NONSTATIONARY
0210 C RANDOM PROCESS OF THE SECOND KIND AND CORRESPONDING FOURIER
0211 C SPECTRUM.
0212      NP=1
0213      WRITE(LU,650)
0214 650 FORMAT(" INPUT -- NP      = # OF PLOTS ON ONE PAGE"/
0215      *"      -- NSMPL = # OF SAMPLES TO BE GENERATED"/
0216      *"      --      NSMPL MUST BE A MULTIPLE OF NP.")
0217      READ(LU,*) NP, NSMPL
0218      ISHEET=NSMPL/NP
0219 C
0220 C.... TO SET INITIAL VALUE OF INTERNAL PARAMETER IN SUBROUTINE
0221 C RANDU (UNIFORM RANDOM NUMBER GENERATING SUBROUTINE).
0222      IX=12347
0223      IY=0
0224      IX1=IX
0225      IY1=IY
0226      IX2=IX
0227      IY2=IY
0228 C
0229      WRITE(LU,652)
0230 652 FORMAT(" DO YOU WANT TO PLOT X(T) AND FOURIER SPECTRA ?"/
0231      *" IF YES, TYPE IN +1")
0232      READ(LU,*) INS
0233      IF(INS.NE.1) GO TO 38
0234      DO 333 I=1,ISHEET
0235      IX0=IX1
0236      IY0=IY1
0237      WRITE(LU,654) I
0238 654 FORMAT(" THIS IS (" ,I3," )-TH PAGE")
0239      WRITE(LU,620)
0240 1010 READ(LU,*)NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY,
0241      *MINX0,MAXX0,MINY0,MAXY0
0242      IF(NY.EQ.100) STOP

```

```

0243      IF(NY.EQ.10) GO TO 38
0244      IF(NY.LT.0) GO TO 334
0245      NNP1=NP+1
0246      CALL STUKO(T,X0,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NNP1,MO,
0247      *1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0248      IX1=IX0
0249      IY1=IY0
0250      DO 320 JM=1,NP
0251      JM1=JM+1
0252      CALL SAMPL(A,X0,XHIL,NDATA,IX1,IY1,RAN1)
0253      CALL STUKO(T,A,NDATA,NX,XMIN,XMAX,NY,YMIN,YMAX,NNP1,MO,
0254      *JM1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0255      320 CONTINUE
0256      GO TO 1010
0257      334 IX0=IX2
0258      IY0=IY2
0259      WRITE(LU,630)
0260      1011 READ(LU,*) NS,SMIN,SMAX,NW,WMIN,WMAX,NP,MO,JDCWS,
0261      *MINW0,MAXW0,MINSO,MAXSO
0262      IF(NS.EQ.100) STOP
0263      IF(NS.EQ.10) GO TO 38
0264      IF(NS.LT.0) GO TO 333
0265      DO 381 IS=1,ND2
0266      381 F0(IS)=X0(IS+ND2)
0267      CALL SPECT(SW,F0,ND2,TTOTAL,1)
0268      CALL STUKO(W,SW,ND4,NW,WMIN,WMAX,NS,SMIN,SMAX,NNP1,MO,
0269      *1,LU,JDCWS,MINW0,MAXW0,MINSO,MAXSO)
0270      IX2=IX0
0271      IY2=IY0
0272      DO 310 J=1,NP
0273      J1=J+1
0274      CALL SAMPL(A,X0,XHIL,NDATA,IX2,IY2,RAN2)
0275      DO 382 JI=1,ND2
0276      382 F0(JI)=A(JI+ND2)
0277      CALL SPECT(SW,F0,ND2,TTOTAL,1)
0278      CALL STUKO(W,SW,ND4,NW,WMIN,WMAX,NS,SMIN,SMAX,NNP1,MO,
0279      *J1,LU,JDCWS,MINW0,MAXW0,MINSO,MAXSO)
0280      310 CONTINUE
0281      GO TO 1011
0282      333 CONTINUE
0283      C
0284      C -----
0285      C.... TO COMPUTE ENSEMBLE STATISTICS OF THE SIMULATED PROCESS.
0286      C.... ASSIGNMENT OF ARRAYS:
0287      C      W(I) --- MEAN VALUE
0288      C      WA(I) --- MAXIMUM VALUE
0289      C      WM(I) --- MINIMUM VALUE
0290      C      SW(I) --- STANDARD DEVIATION
0291      C      F0(I) --- FOR COMPUTATION OF EMPIRICAL DENSITY FUNCTION
0292      C      IPOS --- POSITION NUMBER CORRESPONDING TO T=T1 FOR
0293      C      EMPIRICAL DENSITY COMPUTATION.
0294      38 CONTINUE
0295      C.... FIRST, SET INITIAL PARAMETER VALUES IN SUBROUTINE RANDU.
0296      IX1=IX
0297      IY1=IY
0298      NN=NDATA*2
0299      C
0300      C.... TO ASSIGN INITIAL VALUES IN ARRAY.
0301      DO 32 I=1,NN
0302      32 F0(I)=0.0
0303      IF(ISSW(15),LT.0) GO TO 1501

```

461 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 57 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0304      DO 71 I=1,NFOLD
0305      SW(I)=0.0
0306      W(I)=0.0
0307      WA(I)=-1.0E10
0308      WM(I)=1.0E10
0309      71 CONTINUE
0310  C
0311  C.... TO ASSIGN SPECIFIC TIME FOR EMPIRICAL DENSITY TO BE CONSIDERED.
0312  1501 WRITE(LU,88)
0313      88 FORMAT(" INPUT TIME T1 (SEC) FOR PDF TO BE CONSIDERED."/
0314      &" ALSO INPUT NSMPL, IF NECESSARY.")
0315      READ(LU,*) T1,NSMPL
0316      IPOS=IFIX(T1/DT)
0317      TLOW=DT*FLOAT(IPOS)
0318      TUP=TLOW+DT
0319      IF(ABS(T1-TUP),LT,ABS(T1-TLOW)) IPOS=IPOS+1
0320  C
0321  C.... TO COMPUTE ENSEMBLE STATISTICS.
0322  C      NSMPL --- SAMPLE SIZE OF MONTE CARLO SIMULATION.
0323      DO 30 J=1,NSMPL
0324      CALL SAMPL(A,X0,XMIL,NDATA,IX1,IY1,RAN1)
0325      F0(J)=A(ND2+IPOS)
0326      IF(ISSW(15),LT,0) GO TO 30
0327      DO 31 I=1,ND2
0328      IND2=I+ND2
0329      AIND2=A(IND2)
0330      IF(AIND2.GT,WA(I)) WA(I)=AIND2
0331      IF(AIND2.LT,WM(I)) WM(I)=AIND2
0332      W(I)=W(I)+AIND2
0333      SW(I)=SW(I)+AIND2*AIND2
0334      31 CONTINUE
0335      30 CONTINUE
0336      IF(ISSW(15),LT,0) GO TO 85
0337      DO 72 I=1,ND2
0338      W(I)=W(I)/FLOAT(NSMPL)
0339      SW(I)=SW(I)/FLOAT(NSMPL)
0340      72 CONTINUE
0341      DO 73 I=1,ND2
0342      SW(I)=SQRT(SW(I)-W(I)*W(I))
0343      73 CONTINUE
0344  C
0345  C.... TO PLOT COMPUTED ENSEMBLE STATISTICS EXCEPT EMPIRICAL PDF.
0346      NP=5
0347      MO=-1
0348      WRITE(LU,620)
0349      222 READ(LU,*) NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,
0350      *JDCXY,MINX0,MAXX0,MINY0,MAYX0
0351      IF(NY.EQ.100) STOP
0352      IF(NY.LT.0) GO TO 85
0353      CALL STUKO(T(NFOLD),X0(NFOLD),ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,
0354      *NP,MO,1,LU,JDCXY,MINX0,MAXX0,MINY0,MAYX0)
0355      CALL STUKO(T(NFOLD),W,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
0356      *2,LU,JDCXY,MINX0,MAXX0,MINY0,MAYX0)
0357      CALL STUKO(T(NFOLD),SW,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
0358      *3,LU,JDCXY,MINX0,MAXX0,MINY0,MAYX0)
0359      CALL STUKO(T(NFOLD),WA,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
0360      *4,LU,JDCXY,MINX0,MAXX0,MINY0,MAYX0)
0361      CALL STUKO(T(NFOLD),WM,ND2,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
0362      *5,LU,JDCXY,MINX0,MAXX0,MINY0,MAYX0)
0363      GO TO 222
0364  C

```

```

0365 C -----
0366 C.... TO COMPUTE AND PLOT EMPIRICAL AS WELL AS THEORETICAL PROBABILITY
0367 C DENSITY AT T=T1.
0368 C.... ASSIGNMENT OF ARRAY:
0369 C W(I) --- AMPLITUDE LEVEL
0370 C RV(I) --- EMPIRICAL PROBABILITY DENSITY
0371 C SW(I) --- THEORETICAL ASYMMETRIC PROBABILITY DENSITY.
0372 85 CONTINUE
0373 NSTEP=81
0374 WRITE(LU,223)
0375 223 FORMAT(" INPUT-- NSTEP= # OF AMPLITUDE STEPS (UP TO 81)"/
0376 * " -- PREFERABLY NSTEP IS ODD INTEGER.")
0377 READ(LU,*) NSTEP
0378 C
0379 C.... TO FIND OUT MAXIMUM AND MINIMUM VALUE ON F0(I).
0380 XMAX=-1.0E10
0381 XMIN=1.0E10
0382 DO 521 I=1,NSMPL
0383 IF(F0(I).GT.XMAX) XMAX=F0(I)
0384 IF(F0(I).LT.XMIN) XMIN=F0(I)
0385 521 CONTINUE
0386 C
0387 C.... TO FIND OUT MAX. AND MIN. ON SW(I). ALSO COMPUTE W(I).
0388 XZ=X0(ND2+IPOS)
0389 XZT=XHIL(ND2+IPOS)
0390 CALL ASYME(XZ,XZT,W,SW,NSTEP,XMAX,XMIN,DX)
0391 C
0392 C.... TO COMPUTE EMPIRICAL DENSITY.
0393 DX2=DX/2.0
0394 DFREQ=1.0/(FLOAT(NSMPL)*DX)
0395 DO 530 I=1,NSTEP
0396 530 RV(I)=0.0
0397 DO 516 J=1,NSMPL
0398 F0J=F0(J)
0399 DO 517 I=1,NSTEP
0400 IF(W(I).LT.F0J) GO TO 517
0401 I1=I
0402 IF(ABS(F0J-W(I-1)).LT.DX2) I1=I1-1
0403 RV(I1)=RV(I1)+DFREQ
0404 GO TO 516
0405 517 CONTINUE
0406 516 CONTINUE
0407 C
0408 C.... TO PRINT OUT DENSITY FUNCTIONS.
0409 WRITE(6,540) (W(I),SW(I),RV(I),I=1,NSTEP)
0410 540 FORMAT(5X,"AMPE=",F10.3," THEO.PDF=",F7.4," EMP.PDF=",F7.4)
0411 C
0412 C.... TO PLOT EMPIRICAL AND THEORETICAL PROBABILITY DENSITY.
0413 NX=1
0414 NY=1
0415 YMIN=0.0
0416 YMAX=1.0
0417 NP=2
0418 MO=0
0419 JDCXY=23
0420 MINX0=150
0421 MAXX0=950
0422 MINY0=150
0423 MAXY0=700
0424 WRITE(LU,620)
0425 455 READ(LU,*) NY,YMIN,YMAX,NX,XMIN,XMAX,NP,MO,JDCXY,

```

463 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 59 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0426      *MINX0,MAXX0,MINY0,MAXY0
0427      IF(NY.EQ.100) STOP
0428      IF(NY.LT.0) GO TO 450
0429      CALL STUKO(W,SW,NSTEP,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
0430      *1,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0431      CALL STUKO(W,RV,NSTEP,NX,XMIN,XMAX,NY,YMIN,YMAX,NP,MO,
0432      *2,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0433      GO TO 455
0434      450 WRITE(LU,690)
0435      690 FORMAT(" NEED PDF PLOTTING AT DIFERRENT T=T1 ?"/
0436      *" IF YES, TYPE IN +1"/
0437      *" IF NO NEED OF STATISTICS PLOTTING, SET ISSW(15) ON.")
0438      READ(LU,*) IJD
0439      IF(IJD.EQ.1) GO TO 38
0440      STOP
0441      END
0442  C
0443  C -----(SUBROUTINE AREA)-----
0444      SUBROUTINE AREA(SUM,X2,N,DA)
0445      DIMENSION X2(1)
0446      SUM=0.0
0447      DO 10 I=2,N
0448      10 SUM=SUM+X2(I-1)+X2(I)
0449      SUM=SUM*DA/2.0
0450      RETURN
0451      END
0452  C
0453  C -----(SUBROUTINE SAMPL)-----
0454      SUBROUTINE SAMPL(A,X0,XHIL,NDATA,IX,IY,RAN)
0455      DIMENSION A(1),X0(1),XHIL(1)
0456  C
0457  C.... TO GENERATE NONSTATIONARY RANDOM PROCESS WITH UNIFORM
0458  C      RANDOM PHASE ANGLE BETWEEN -PI/2 AND +PI/2.
0459  C
0460  C.... DESCRIPTION OF PARAMETERS:
0461  C      X0(I) --- SAMPLE OBSERVATION TO BE USED FOR SIMULATION.
0462  C      XHIL(I)--- HILBERT TRANSFORM OF X0(I)
0463  C      A(I) --- GENERATED NONSTATIONARY RANDOM PROCESS
0464  C      NDATA --- NUMBER OF DATA POINTS
0465  C      IX, IY --- INTERNAL PARAMETERS IN UNIFORM RANDOM VARIATE
0466  C                  GENERATING SUBROUTINE RANDU(IX,IY,RAN)
0467  C
0468      PI=3.1415926
0469      CALL RANDU(IX,IY,RAN)
0470      RAN=(RAN-0.5)*PI
0471      CCOS=COS(RAN)
0472      SSIN=SIN(RAN)
0473      DO 100 I=1,NDATA
0474      A(I)=CCOS*X0(I)-SSIN*XHIL(I)
0475      100 CONTINUE
0476      RETURN
0477      END
0478  C
0479  C -----(SUBROUTINE FFT0)-----
0480      SUBROUTINE FFT0(Y,DATA,N,ISIGN,MW)
0481      DIMENSION Y(1),DATA(1)
0482  C.... TO TAKE ONE DIMENSIONAL FOURIER (INVERSE) TRANSFORM
0483  C      BY FFT ALGORISM
0484  C.... DESCRIPTION OF PARAMETERS:
0485  C      DATA= ORIGINAL TIME SERIES
0486  C      N= NUMBER OF TIME SERIES

```

```

0487 C      "N" MUST BE A POWER OF TWO
0488 C      ISIGN ----- IF FOURIER TRANSFORM, ISIGN= -1
0489 C      ----- IF INVERSE FOURIER, ISIGN= 1
0490 C      Y= FOURIER COMPLEX COEFFICIENT,
0491 C      OR COMPLEX TIME SERIES.
0492 C      ARRAY MUST BE IN ORDER OF (REAL), (IMAGINARY)
0493 C
0494 C      AN=FLOAT(N)
0495 C      AISIGN=FLOAT(ISIGN)
0496 C      PI=3.141593
0497 C      IF(MW.EQ.1) GO TO 10
0498 C      IF(ISIGN.EQ.1) GO TO 10
0499 C      DO 100 I=1,N
0500 C      II=I+1
0501 C      IR=II-1
0502 C      Y(IR)=DATA(I)
0503 C      Y(II)=0.0
0504 C      100 CONTINUE
0505 C      10 L=1
0506 C      DO 200 I=1,N
0507 C      IF(1.GE.L) GO TO 210
0508 C      LI=L+L
0509 C      LR=LI-1
0510 C      II=I+I
0511 C      IR=II-1
0512 C      AR=Y(LR)
0513 C      AI=Y(LI)
0514 C      Y(LR)=Y(IR)
0515 C      Y(LI)=Y(II)
0516 C      Y(IR)=AR
0517 C      Y(II)=AI
0518 C      210 N2=N/2
0519 C      220 IF(L.LE,N2) GO TO 230
0520 C      L=L-N2
0521 C      N2=N2/2
0522 C      IF(N2.GE.2) GO TO 220
0523 C      230 L=L+N2
0524 C      200 CONTINUE
0525 C      MAX=1
0526 C      240 IF(MAX.GE.N) GO TO 400
0527 C      ISTEP=MAX*2
0528 C      AMAX=FLOAT(MAX)
0529 C      AK=-1.0
0530 C      DO 300 K=1,MAX
0531 C      AK=AK+1.0
0532 C      WT=PI*AISIGN*AK/AMAX
0533 C      DO 310 I=K,N,ISTEP
0534 C      L=I+MAX
0535 C      LI=L+L
0536 C      LR=LI-1
0537 C      II=I+I
0538 C      IR=II-1
0539 C      CCOS=COS(WT)
0540 C      SSIN=SIN(WT)
0541 C      AR=Y(LR)*CCOS-Y(LI)*SSIN
0542 C      AI=Y(LR)*SSIN+Y(LI)*CCOS
0543 C      Y(LR)=Y(IR)-AR
0544 C      Y(LI)=Y(II)-AI
0545 C      Y(IR)=Y(IR)+AR
0546 C      Y(II)=Y(II)+AI
0547 C      310 CONTINUE

```

465 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 61 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0548      300 CONTINUE
0549      MAX=ISTEP
0550      GO TO 240
0551      400 IF(ISIGN,EQ,1) RETURN
0552      NM2=N*2
0553      DO 450 I=1,NM2
0554      Y(I)=Y(I)/AN
0555      450 CONTINUE
0556      RETURN
0557      END
0558 C
0559 C -----(SUBROUTINE ASYME)-----
0560 SUBROUTINE ASYME(XZ,XZT,W,SW,NSTEP,XMAX,XMIN,DX)
0561 DIMENSION W(1),SW(1)
0562 PI=3.141593
0563 PI2=PI/2.0
0564 XZ2=XZ**2
0565 XZT2=XZT**2
0566 AZ=SQRT(XZ2+XZT2)
0567 CALL PHS11(XZ,XZT,TA)
0568 ARF=AZ*COS(PI/2.0-ABS(TA))
0569 BET=AZ*COS(PI/2.0+ABS(TA))
0570 IF(ABS(TA).GT,PI2) GO TO 400
0571 AMIN=BET
0572 AMP=ARF
0573 AMAX=AZ
0574 GO TO 401
0575 400 AMIN=-AZ
0576 AMP=-ARF
0577 AMAX=-BET
0578 401 IF(AMAX.GT,XMAX) XMAX=AMAX
0579 IF(AMIN.LT,XMIN) XMIN=AMIN
0580 DDX=0.0001
0581 XMAX=XMAX+DDX
0582 XMIN=XMIN-DDX
0583 DX=(XMAX-XMIN)/FLOAT(NSTEP-1)
0584 DO 30 I=1,NSTEP
0585 30 W(I)=DX*FLOAT(I-1)+XMIN
0586 DO 55 I=1,NSTEP
0587 WI=W(I)
0588 R1=PI*SQRT(ABS(AZ**2-WI**2))
0589 IF(R1 .LT. 0.01) R1=0.01
0590 SW(I)=1./R1
0591 IF(WI.GT,AMP) SW(I)=2.0*SW(I)
0592 55 CONTINUE
0593 RETURN
0594 END
0595 C
0596 C -----(SUBROUTINE PHS11)-----
0597 SUBROUTINE PHS11(XZ,XZT,B)
0598 PI1=3.141593
0599 PI2=PI1*2.0
0600 PI12=PI1/2.0
0601 AR=XZ
0602 AI=XZT
0603 IF(ABS(AR).LT,E-20) GO TO 30
0604 FAI0=AI/AR
0605 FAI=ATAN(FAI0)
0606 IF(FAI,LT,0.) GO TO 40
0607 IF(AR,LT,0.) FAI=FAI-PI1
0608 GO TO 50

```

```
0609      40 IF(AR,LT.0.) FAI=FAI+PI1
0610      GO TO 50
0611      30 IF(AI,GT.0.) FAI=PI12
0612      IF(AI,LT.0.) FAI=-PI12
0613      50 B=FAI
0614      RETURN
0615      END
0616      ENDS
```



## 467 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES — 63 —

IINTPL T=00004 IS ON CR00002 USING 00005 BLKS R=0033

```

0001 FTN4
0002 C -----(SUBROUTINE INTPL)-----
0003 SUBROUTINE INTPL(X0,Y0,NORG,X,Y,NDATA,DX)
0004 DIMENSION X0(1),Y0(1),X(1),Y(1)
0005 C.... THIS SUBROUTINE INTERPOLATES DATA SET (X0,Y0) TO PRODUCE
0006 C EQUIDISTANT DATA SET (X,Y) WITH ABSCISSA INCREMENT DX.
0007 C.... DESCRIPTION OF PARAMETERS:
0008 C X0 --- ORIGINAL ABSCISSA ARRAY
0009 C Y0 --- ORIGINAL ORDINATE ARRAY
0010 C NORG --- NUMBER OF ORIGINAL DATA
0011 C X --- INTERPOLATED ABSCISSA ARRAY
0012 C Y --- INTERPOLATED ORDINATE ARRAY
0013 C NDATA -- NUMBER OF INTERPOLATED DATA
0014 C DX --- EQUIDISTANT INCREMENT OF ABSCISSA
0015 C
0016 Y(1)=Y0(1)
0017 X(1)=X0(1)
0018 JJ=2
0019 SLOPE=(Y0(2)-Y0(1))/(X0(2)-X0(1))
0020 DO 10 I=2,NDATA
0021 X(I)=X(I-1)+DX
0022 30 IF(X(I).GT.X0(JJ)) GO TO 20
0023 Y(I)=Y0(JJ-1)+SLOPE*(X(I)-X0(JJ-1))
0024 GO TO 10
0025 20 JJ=JJ+1
0026 IF(JJ.LE.NORG) GO TO 25
0027 Y(I)=0.0
0028 GO TO 10
0029 25 SLOPE=(Y0(JJ)-Y0(JJ-1))/(X0(JJ)-X0(JJ-1))
0030 GO TO 30
0031 10 CONTINUE
0032 RETURN
0033 END
0034 ENDS

```

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

```

0001 FTN4
0002 C -----(SUBROUTINE FFT)-----
0003 SUBROUTINE FFT(Y,DATA,N,ISIGN)
0004 DIMENSION Y(1),DATA(1)
0005 C.... THIS SUBROUTINE COMPUTES ONE-DIMENSIONAL FOURIER OR INVERSE
0006 C FOURIER TRANSFORM BY FFT (FAST FOURIER TRANSFORM) ALGORITHM.
0007 C.... DESCRIPTION OF PARAMETERS:
0008 C DATA(I) --- ORIGINAL REAL TIME DOMAIN DATA.
0009 C N --- NUMBER OF DATA POINTS (= A POWER OF 2).
0010 C ISIGN --- CONTROL PARAMETER.
0011 C ISIGN=-1 FOR FOURIER TRANSFORM.
0012 C ISIGN=+1 FOR INVERSE FOURIER TRANSFORM.
0013 C Y(I) --- FOURIER COMPLEX COEFFICIENT (FREQUENCY DOMAIN)
0014 C OR COMPLEX TIME SERIES (TIME DOMAIN).
0015 C THIS ARRAY MUST BE IN THE ORDER OF
0016 C (REAL-1),(IMAG-1),(REAL-2),(IMAG-2),.....
0017 C.... REMARKS:
0018 C WHEN ISIGN=+1, DATA(I) BECOMES DUMMY.
0019 C WHEN ISIGN=-1, PROGRAM AUTOMATICALLY TRANSFERS REAL TIME
0020 C DOMAIN DATA (DATA(I)) INTO COMPLEX ARRAY Y(I).
0021 C ARRAY SIZE REQUIREMENT --- Y(2*N), DATA(N)
0022 C
0023 AN=FLOAT(N)
0024 AISIGN=FLOAT(ISIGN)
0025 PI=3.141593
0026 IF(ISIGN.EQ.1) GO TO 10
0027 DO 100 I=1,N
0028 II=I+1
0029 IR=II-1
0030 Y(IR)=DATA(I)
0031 Y(II)=0.0
0032 100 CONTINUE
0033 10 L=1
0034 DO 200 I=1,N
0035 IF(I.GE.L) GO TO 210
0036 LI=L+L
0037 LR=LI-1
0038 II=I+I
0039 IR=II-1
0040 AR=Y(LR)
0041 AI=Y(LI)
0042 Y(LR)=Y(IR)
0043 Y(LI)=Y(II)
0044 Y(IR)=AR
0045 Y(II)=AI
0046 210 N2=N/2
0047 220 IF(L.LE.N2) GO TO 230
0048 L=L-N2
0049 N2=N2/2
0050 IF(N2.GE.2) GO TO 220
0051 230 L=L+N2
0052 200 CONTINUE
0053 MAX=1
0054 240 IF(MAX.GE.N) GO TO 400
0055 ISTEP=MAX*2
0056 AMAX=FLOAT(MAX)
0057 AK=-1.0
0058 DO 300 K=1,MAX
0059 AK=AK+1.0

```

469 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 65 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```
0060      WT=PI*AISIGN*AK/AMAX
0061      DO 310 I=K,N,ISTEP
0062      L=I+MAX
0063      LI=L+L
0064      LR=LI-1
0065      II=I+I
0066      IR=II-1
0067      CCOS=COS(WT)
0068      SSIN=SIN(WT)
0069      AR=Y(LR)*CCOS-Y(LI)*SSIN
0070      AI=Y(LR)*SSIN+Y(LI)*CCOS
0071      Y(LR)=Y(IR)-AR
0072      Y(LI)=Y(II)-AI
0073      Y(IR)=Y(IR)+AR
0074      Y(II)=Y(II)+AI
0075      310 CONTINUE
0076      300 CONTINUE
0077      MAX=ISTEP
0078      GO TO 240
0079      400 IF(ISIGN.EQ.1) RETURN
0080      NM2=N*2
0081      DO 450 I=1,NM2
0082      Y(I)=Y(I)/AN
0083      450 CONTINUE
0084      RETURN
0085      END
0086      ENDS
```

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

```

0001 FYN4
0002 C -----(SUBROUTINE PHASE)-----
0003 SUBROUTINE PHASE(A,C,NDATA,B)
0004 C.... THIS SUBROUTINE COMPUTES PHASE ANGLE B(I) OF FOURIER TRANSFORM
0005 C OF TIME HISTORY C(I). PHASE ANGLES RANGE BETWEEN -PI AND PI.
0006 C.... DESCRIPTION OF PARAMETERS:
0007 C A(I) --- COMPLEX FOURIER COEFFICIENT.
0008 C C(I) --- TIME SERIES FOR PHASE ANGLE CALC.(INPUT).
0009 C NDATA --- NUMBER OF TIME SERIES DATA.
0010 C NDATA MUST BE A POWER OF TWO FOR FFT COMPUTATION.
0011 C B(I) --- CALCULATED PHASE ANGLE (OUTPUT).
0012 C
0013 DIMENSION A(1),B(1),C(1)
0014 CALL FFT(A,C,NDATA,-1)
0015 PI1=3.141593
0016 PI2=PI1*2.0
0017 PI12=PI1/2.0
0018 DO 100 I=1,NDATA
0019 II=1+I
0020 IR=II-1
0021 AR=A(IR)
0022 AI=A(II)
0023 IF(ABS(AR).LT.E-20) GO TO 30
0024 FAI0=AI/AR
0025 FAI=ATAN(FAI0)
0026 IF(FAI.LT.0.) GO TO 40
0027 IF(AR.LT.0.) FAI=FAI-PI1
0028 GO TO 50
0029 40 IF(AR.LT.0.) FAI=FAI+PI1
0030 GO TO 50
0031 30 IF(AI.GT.0.) FAI=PI12
0032 IF(AI.LT.0.) FAI=-PI12
0033 50 B(I)=FAI
0034 100 CONTINUE
0035 RETURN
0036 END
0037 ENDS

```

471 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 67 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

ISPECT I=00004 IS ON CR00002 USING 00009 BLKS R=0069

```

0001 FTN4
0002 C -----(SUBROUTINE SPECT)-----
0003 SUBROUTINE SPECT(A,Y,NDATA,TT,MO)
0004 DIMENSION A(1),Y(1)
0005 C.... THIS SUBROUTINE COMPUTES POWER SPECTRUM, FOURIER (AMPLITUDE)
0006 C SPECTRUM OR ABSOLUTE VALUE OF FOURIER COMPLEX COEFFICIENT.
0007 C.... DESCRIPTION OF PARAMETERS:
0008 C Y(1) --- ORIGINAL TIME HISTORY.
0009 C NDATA --- NUMBER OF DATA IN Y(I).
0010 C TT --- DURATION (PERIOD) OF TIME HISTORY (SEC).
0011 C NFOLD --- NYQUIST FOLDING NUMBER
0012 C A(1) --- COMPUTED RESULT.
0013 C POWER OR FOURIER SPECTRUM, OR ABSOLUTE
0014 C VALUE OF FOURIER COMPLEX COEFFICIENT.
0015 C IF MO= 10, A(1)= POWER SPECTRUM FOR F (HZ).
0016 C = 11, A(1)= POWER SPECTRUM FOR W (RAD. SEC).
0017 C IF MO= 1, A(1)= FOURIER SPECTRUM
0018 C IF MO= 2, A(1)= ABS. VALUE OF COMPLEX FOURIER COEF.
0019 C FROM 1 TO NDATA
0020 C
0021 NFOLD=NDATA/2+1
0022 PI2=3.141593*2.0
0023 CALL FFT(A,Y,NDATA,-1)
0024 ND1=NDATA+1
0025 ND2=NDATA+2
0026 IF(MO.EQ.1) GO TO 50
0027 IF(MO.EQ.2) GO TO 55
0028 C
0029 C.... TO COMPUTE POWER SPECTRUM.
0030 AR=A(1)**2+A(2)**2
0031 AR=AR*TT
0032 A(1)=AR
0033 IF(MO.GT.10) A(1)=AR/PI2
0034 DO 10 I=2,NFOLD-1
0035 II=I+1
0036 IR=II-1
0037 AR=A(IR)**2+A(II)**2
0038 AR=AR*2.0*TT
0039 A(I)=AR
0040 IF(MO.GT.10) A(I)=AR/PI2
0041 10 CONTINUE
0042 AR=A(ND1)**2+A(ND2)**2
0043 AR=AR*TT
0044 A(NFOLD)=AR
0045 IF(MO.GT.10) A(NFOLD)=AR/PI2
0046 GO TO 60
0047 C
0048 C.... TO COMPUTE FOURIER (AMPLITUDE) SPECTRUM.
0049 50 DO 40 I=1,NFOLD
0050 II=I+1
0051 IR=II-1
0052 AR=A(IR)**2+A(II)**2
0053 A(I)=TT*SQRT(AR)
0054 40 CONTINUE
0055 GO TO 60
0056 C
0057 C.... TO COMPUTE ABS. VALUE OF COMPLEX FOURIER COEFFICIENT.
0058 55 DO 45 I=1,NFOLD
0059 II=I+1

```

```
0060      IR=II-1
0061      AR=A(IR)**2+A(II)**2
0062      A(I)=SQRT(AR)
0063      45 CONTINUE
0064 C
0065      60 DO 20 I=NFOLD+1,NDATA
0066          II=I-NFOLD
0067          IN=NFOLD-II
0068          A(I)=A(IN)
0069      20 CONTINUE
0070      RETURN
0071      END
0072      ENDS
```

473 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 69 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

```

0001 FTN4
0002 C -----(SUBROUTINE STATC)-----
0003 C SUBROUTINE STATC(Y,N,EY,VAR,STDV,YMAX,YMIN)
0004 C DIMENSION Y(1)
0005 C.... THIS SUBROUTINE COMPUTES BASIC STATISTICS ON DATA Y(I)
0006 C SUCH AS OVERALL (TEMPORAL) MEAN, VARIANCE, STANDARD
0007 C DEVIATION, MAXIMUM AND MINIMUM VALUES.
0008 C.... DESCRIPTION OF PARAMETERS:
0009 C Y(I) --- ORIGINAL DATA ARRAY.
0010 C N --- NUMBER OF DATA.
0011 C EY --- OVERALL (TEMPORAL) MEAN OF Y(I).
0012 C VAR --- VARIANCE OF Y(I).
0013 C STDV --- STANDARD DEVIATION OF Y(I).
0014 C YMAX --- MAXIMUM VALUE IN Y(I).
0015 C YMIN --- MINIMUM VALUE IN Y(I).
0016 C
0017 EY=0.0
0018 VAR=0.0
0019 YMAX=-E30
0020 YMIN=E30
0021 DO 10 I=1,N
0022 YI=Y(I)
0023 EY=EY+YI
0024 IF(YI.GT.YMAX) YMAX=YI
0025 IF(YI.LT.YMIN) YMIN=YI
0026 10 CONTINUE
0027 EY=EY/FLOAT(N)
0028 DO 20 I=1,N
0029 YI=Y(I)-EY
0030 VAR=VAR+YI*YI
0031 20 CONTINUE
0032 VAR=VAR/FLOAT(N-1)
0033 STDV=SQRT(VAR)
0034 RETURN
0035 END
0036 ENDS

```

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

```

0001 FTN4
0002 C -----(SUBROUTINE HLBRT)-----
0003 SUBROUTINE HLBRT(A,X0,XHIL,N,MO)
0004 DIMENSION A(1),X0(1),XHIL(1)
0005 C.... THIS SUBROUTINE COMPUTES HILBERT TRANSFORM OF TIME HISTORY
0006 C X0(T) FOR SIMULATION OF DATA-BASED NONSTATIONARY RANDOM
0007 C PROCESSES.
0008 C.... DESCRIPTION OF PARAMETERS:
0009 C X0 --- ORIGINAL TIME HISTORY DATA (REAL ARRAY).
0010 C N --- NUMBER OF DATA ON X0(I).
0011 C A --- ARRAY FOR FFT COMPUTATION.
0012 C ARRAY SIZE MUST BE 2*N.
0013 C XHIL -- HILBERT TRANSFORM OF X0(T).
0014 C MO --- CONTROL PARAMETER.
0015 C IF MO=0 --- FFT COMPUTATION ON X0(T) IS NEEDED.
0016 C IF MO=1 --- NO NEED ON FFT COMPUTATION ON X0(T).
0017 C
0018 IF(MO.EQ.1) GO TO 10
0019 CALL FFT(A,X0,N,-1)
0020 10 NFOLD=N/2+1
0021 DO 100 I=NFOLD+1,N
0022 II=I+1
0023 IR=II-1
0024 A(IR)=-A(IR)
0025 A(II)=-A(II)
0026 100 CONTINUE
0027 CALL FFT(A,X0,N,1)
0028 DO 200 I=1,N
0029 II=I+1
0030 XHIL(I)=A(II)
0031 200 CONTINUE
0032 RETURN
0033 END
0034 ENDS

```



475 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 71 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

```

0001 FTN4
0002 C -----(SUBROUTINE JNORM)-----
0003 SUBROUTINE JNORM(IX1,IY1,IX2,IY2,STDV1,STDV2,AM1,CORRL,
0004 *RAN1,RAN2,MO,MDIM)
0005 C.... THIS SUBROUTINE GENERATES EITHER UNIVARIATE OR JOINTLY
0006 C CORRELATED BIVARIATE GAUSSIAN RANDOM NUMBER.
0007 C.... DESCRIPTION OF PARAMETERS:
0008 C RAN1 -- GENERATED GAUSSIAN RANDOM NUMBER WITH
0009 C MEAN AM1 AND STANDARD DEVIATION STDV1.
0010 C RAN2 -- GENERATED THE OTHER GAUSSIAN RANDOM NUMBER
0011 C WITH STANDARD DEVIATION STDV2. RAN2 IS
0012 C CORRELATED TO RAN1 THROUGH CORRELATION
0013 C COEFFICIENT CORRL.
0014 C CORRL -- CORRELATION COEFFICIENT BETWEEN RAN1 AND RAN2.
0015 C IF ABS(CORRL)=1 -- RAN2=CORRL*RAN1
0016 C IF CORRL=0 ----- RAN1 AND RAN2 ARE INDEPENDENT.
0017 C MO --- CONTROL PARAMETER ON THE OUTPUT RANDOM NUMBERS.
0018 C IF MO > 0 -- -PI< RAN1,RAN2 < PI
0019 C IF MO < 0 -- -INFINITY < RAN1,RAN2 < +INFINITY
0020 C MDIM --- DIMENSION CONTROL PARAMETER.
0021 C IF MDIM=1 -- UNIVARIATE RANDOM NUMBER. IN THIS
0022 C CASE, ONLY RAN1 IS TO BE GENERATED.
0023 C IF MDIM=2 -- CORRELATED BIVARIATE RANDOM NUMBER.
0024 C IX1,IY1,IX2,IY2 -- ENTRY FOR SUBROUTINE RANDU(IX,IY,RAN)
0025 C
0026 IF(NDIM,EQ,0) NDIM=1
0027 C
0028 C.... GENERATION OF FIRST GAUSSIAN RANDOM NUMBER RAN1.
0029 CALL GAUSS(IX1,IY1,STDV1,AM1,RAN1)
0030 IF(MDIM,EQ,1) RETURN
0031 IF(ABS(CORRL).NE.1,0) GO TO 5
0032 RAN2=RAN1*CORRL*STDV2/STDV1
0033 GO TO 6
0034 C
0035 C.... GENERATION OF SECOND GAUSSIAN RANDOM NUMBER RAN2.
0036 5 AM2=CORRL*STDV2*RAN1/STDV1
0037 C
0038 C.... FOR THE CASE OF CORRL BEING VERY CLOSE TO UNITY.
0039 DELTA=1,0-ABS(CORRL)
0040 IF(DELTA,GE,0.001) GO TO 7
0041 STDV=STDV2*SQRT(2,0*DELTA)
0042 GO TO 8
0043 7 STDV=STDV2*SQRT(1,0-CORRL*CORRL)
0044 8 CALL GAUSS(IX2,IY2,STDV,AM2,RAN2)
0045 6 IF(MO,LT,0) RETURN
0046 C
0047 C.... REARRANGEMENT OF GENERATED GAUSSIAN RANDOM NUMBERS SO AS
0048 C TO RANGE BETWEEN -PI AND PI.
0049 CALL REARR(RAN1)
0050 CALL REARR(RAN2)
0051 RETURN
0052 END
0053 C
0054 C -----(SUBROUTINE GAUSS)-----
0055 SUBROUTINE GAUSS(IX,IY,STDV,AMEAN,RAN)
0056 C.... THIS SUBROUTINE GENERATES A GAUSSIAN RANDOM NUMBER WITH
0057 C MEAN AMEAN AND STANDARD DEVIATION STDV.
0058 C
0059 A=0,0

```

```
0060      DO 10 I=1,12
0061      CALL RANDU(IX,IY,Y)
0062      10 A=A+Y
0063      RAN=(A-6.0)*STDV+AMEAN
0064      RETURN
0065      END
0066  C
0067  C -----(SUBROUTINE REARR)-----
0068      SUBROUTINE REARR(RAN)
0069  C.... THIS SUBROUTINE REPLACE RAN SO AS TO RANGE BETWEEN -PI
0070  C      AND PI WITH THE CONSIDERATION OF RAN BEING A PERIODIC
0071  C      FUNCTION OF PERIOD 2*PI.
0072  C
0073      PI=3.141593
0074      PIM=-PI
0075      PI2=2.0*PI
0076      IF(RAN,LT,0.0) GO TO 10
0077      20 IF(RAN,LE,PI) GO TO 30
0078      RAN=RAN-PI2
0079      GO TO 20
0080      10 IF(RAN,GE,PIM) GO TO 30
0081      RAN=RAN+PI2
0082      GO TO 10
0083      30 RETURN
0084      END
0085      ENDS
```

477 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 73 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

```

0001 FTN4
0002 C -----(SUBROUTINE ARNGE)-----
0003 SUBROUTINE ARNGE(Y,N,MO,MJD)
0004 DIMENSION Y(1)
0005 C.... THIS SUBROUTINE ARRANGES ARRAY DATA Y(I) SYMMETRICALLY
0006 C TO PRODUCE MIRROR IMAGE AROUND ORIGIN OR AROUND NFOLD(=N/2+1).
0007 C.... DESCRIPTION OF PARAMETERS:
0008 C Y -- INPUT ARRAY TO BE ARRANGED SYMMETRICALLY
0009 C AT LEAST NFOLD DATA MUST BE DEFINED
0010 C BEFORE CALLING.
0011 C Y COULD BE EITHER REAL OR COMPLEX.
0012 C COMPLEX DATA SHOULD BE IN ORDER OF
0013 C (REAL1),(IMAG1),(REAL2),(IMAG2),...
0014 C N -- ARRAY SIZE OF Y
0015 C MJD -- CLASSIFICATION PARAMETER OF DATA Y
0016 C IF MJD>0, FOR REAL DATA ONLY
0017 C MJD<0, FOR COMPLEX DATA
0018 C FOR COMPLEX DATA, ONLY MIRROR SYMMETRY
0019 C AROUND NFOLD IS CONSIDERED.
0020 C FOR N > NFOLD, SIGN OF IMAGINARY
0021 C PART IS REVERSED.
0022 C "MO" SHALL BE DISREGARDED IN THIS CASE.
0023 C MO -- CONTROL PARAMETER FOR ARRANGEMENT
0024 C IF MO= 10, SYMMETRY AROUND ORIGIN WITH SAME SIGN
0025 C =-10, SYMMETRY AROUND ORIGIN WITH REVERSE SIGN
0026 C IF MO= 11, SYMMETRY AROUND NFOLD WITH SAME SIGN
0027 C =-11, SYMMETRY AROUND NFOLD WITH REVERSED SIGN
0028 C
0029 N1=N+1
0030 N2=N+2
0031 ND2=N/2
0032 NFOLD=ND2+1
0033 NFOLD1=NFOLD+1
0034 IF(MJD,LT.0) GO TO 50
0035 C -----
0036 C.... FOR REAL DATA ARRAY ONLY
0037 C.... SYMMETRY AROUND NFOLD WITH SAME SIGN
0038 IF(ABS(MO).EQ.10) GO TO 10
0039 DO 100 I=NFOLD1,N1
0040 IN=I-NFOLD
0041 INR=NFOLD-IN
0042 Y(I)=Y(INR)
0043 C
0044 C.... SYMMETRY AROUND NFOLD WITH REVERSED SIGN
0045 IF(MO.EQ.-11) Y(I)=-Y(INR)
0046 100 CONTINUE
0047 RETURN
0048 C
0049 C.... SYMMETRY AROUND ORIGIN WITH SAME SIGN
0050 10 YNFOLD=Y(NFOLD)
0051 DO 200 I=1,ND2
0052 IP=I+ND2
0053 Y(IP)=Y(I)
0054 200 CONTINUE
0055 Y(N1)=YNFOLD
0056 DO 205 I=1,ND2
0057 IN=N2-I
0058 Y(I)=Y(IN)
0059 C

```

```
0060 C.... SYMMETRY AROUND ORIGIN WITH REVERSED SIGN
0061       IF(MO,EQ,-10) Y(I)=-Y(IN)
0062       205 CONTINUE
0063       RETURN
0064 C -----
0065 C.... FOR COMPLEX DATA ARRAY
0066       50 NFOLD4=4*NFOLD
0067       DO 300 I=NFOLD1,N
0068         II=I+1
0069         IR=II-1
0070         INI=NFOLD4-II
0071         INR=INI-1
0072         Y(IR)=Y(INR)
0073         Y(II)=-Y(INI)
0074       300 CONTINUE
0075       RETURN
0076       END
0077       ENDS
```

479 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 75 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

```

0001 FTN4
0002 SUBROUTINE STUKO(X,Y,NPT,NX,XMIN,XMAX,NY,YMIN,YMAX,
0003 * NP,MO,IP,LU,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0004 DIMENSION X(1),Y(1),NAME(3),NA(20),AA(5)
0005 EQUIVALENCE (AA,NA(11))
0006 C
0007 C.... THIS SUBROUTINE IS USED TO PLOT Y(I)-X(I) RELATIONSHIP
0008 C IN ANY LOCATION AND SIZE AS DESIRED ON A SHEET.
0009 C THIS IS A MULTI-FUNCTIONAL SUBROUTINE.
0010 C
0011 C.... DATE 06/27/1978
0012 C.... BY DR. H. ISHIKAWA AND D. LI, COLUMBIA UNIVERSITY.
0013 C
0014 C.... DESCRIPTION OF PARAMETERS:
0015 C X -- COORDINATE ARRAY.
0016 C Y -- ABSCISSA ARRAY.
0017 C NPT -- NUMBER OF POINTS TO BE PLOTTED.
0018 C YMAX -- MAXIMUM VALUE ON Y-AXIS.
0019 C YMIN -- MINIMUM VALUE ON Y-AXIS.
0020 C YMIN AND YMAX ARE EITHER SUPPLIED BY USER
0021 C OR SEARCHED BY THE PROGRAM (SEE ALSO NY).
0022 C NY -- CONTROL CODE ON YMAX AND YMIN.
0023 C IF NY> 0 --- USER SUPPLY FOR YMAX, YMIN.
0024 C NY= 0 --- DEFAULT OPTION.
0025 C NY< 0 --- RETURN TO THE MAIN PROGRAM.
0026 C PROGRAM SEARCHES YMAX AND YMIN.
0027 C XMAX -- MAXIMUM ORDINATE VALUE ON X-AXIS.
0028 C XMIN -- MINIMUM ORDINATE VALUE ON X-AXIS.
0029 C X-AXIS IS BOUNDED FROM XMIN TO XMAX.
0030 C NX -- CONTROL CODE ON XMIN AND XMAX.
0031 C IF NX= 0 --- PROGRAM SETS AUTOMATICALLY
0032 C XMIN=X(1) AND XMAX=X(N).
0033 C NX> 0 --- USER SUPPLY FOR XMIN AND XMAX.
0034 C NX< 0 --- ACCORDING TO THE USER SUPPLIED
0035 C XMIN AND XMAX, PROGRAM AUTOMATICALLY
0036 C COMPUTES ORDINATE ARRAY X(I) WITH
0037 C DX=(XMAX-XMIN)/N.
0038 C NP -- CONTROL CODE ON NO. OF FRAMES OR FIGURES.
0039 C THIS CONSISTS OF 5 DIGITS (IJKLM) AS FOLLOWS.
0040 C I (=NDIM) -- NO. OF COLUMNS (UP TO 2).
0041 C JK (=NP1) -- NO. OF FRAMES OR GRAPHS IN LEFT COLUMN.
0042 C LM (=NP2) -- NO. OF FRAMES OR GRAPHS IN RIGHT COLUMN.
0043 C NP1TL=NP1+NP2 -- TOTAL NO. OF GRAPHS TO BE DRAWN.
0044 C DEFAULT OPTION -- IF ONE COLUMN, NP CAN BE ONLY
0045 C TWO DIGITS (LM). IN THIS CASE,
0046 C NP1TL=(LM),
0047 C MO -- CONTROL CODE ON THE WAY OF PLOTTING NP GRAPHS.
0048 C IF MO= 0 --- NP1TL PLOTS ON ONE FRAME.
0049 C MO= 1 --- NP1TL PLOTS ON NP1TL FRAMES,
0050 C WITH NP1TL LABELS.
0051 C MO=-1 --- NP1TL PLOTS ON NP1TL FRAMES,
0052 C WITH ONLY ONE LABEL.
0053 C JDCXY -- LABEL DIGITS CONTROL CODE ON X- AND Y-AXIS.
0054 C THIS CONSISTS OF 3 DIGITS (IJK) AS FOLLOWS.
0055 C I -- LABEL CONTROL CODE.
0056 C IF I= 0 -- LABEL IS PRINTED OUT.
0057 C IF OTHERWISE -- NO LABEL IS PRINTED OUT.
0058 C J -- NO. OF DECIMAL POINTS ON X-AXIS.
0059 C FORMAT NOTATION -- F7,I (I=0,1,....,5)

```

```

0060 C      K -- NO. OF DECIMAL POINTS ON Y-AXIS.
0061 C      MINX0,MAXX0,MINY0,MAXY0
0062 C      -- PARAMETER TO DEFINE THE LOCATION OF THE FIGURE.
0063 C      POSSIBLE LOCATION OF X-AXIS --- 1 TO 1024.
0064 C      POSSIBLE LOCATION OF Y-AXIS --- 1 TO 780.
0065 C      DEFAULT VALUES: MINX0= 40  -- MAXX0=1000
0066 C                        MINY0= 50  -- MAXY0= 750
0067 C      LU  -- TEXTRONIX TERMINAL LOGICAL UNIT NUMBER.
0068 C      IP  -- FRAME NUMBER ON THE PICTURE OR
0069 C      GRAPH LINE NUMBER ON THE PICTURE FRAME.
0070 C      FIRST TIME CALL IP=1, LAST TIME CALL IP=NP TTL.
0071 C
0072 C      DATA NAME/2HCH,2HIE,2H
0073 C      NDIM=NP/10000
0074 C      NP1=NP/100-100*NDIM
0075 C      NP2=NP-100*NP1-10000*NDIM
0076 C      NP TTL=NP1+NP2
0077 C      IF(NDIM.GT.1) GO TO 5
0078 C      IF(NP1.EQ.0) NP1=NP2
0079 C      IF(NP1.EQ.0) NP1=1
0080 C      NP TTL=NP1
0081 C      5 IF(NP TTL.LE.0) NP TTL=1
0082 C      IF(IP.EQ.1) CALL EXEC(4,NP TTL,ISTRK,IDISC,ISECT)
0083 C      NSTRK=ISTRK+IP-1
0084 C      NA(1)=NPT
0085 C      NA(2)=NX
0086 C      NA(3)=NY
0087 C      NA(4)=MO
0088 C      NA(5)=JDCXY
0089 C      NA(6)=MINX0
0090 C      NA(7)=MAXX0
0091 C      NA(8)=MINY0
0092 C      NA(9)=MAXY0
0093 C      AA(1)=XMIN
0094 C      AA(2)=XMAX
0095 C      AA(3)=YMIN
0096 C      AA(4)=YMAX
0097 C      NPT2=2*NPT
0098 C      CALL EXEC (2,1028,NA,20,NSTRK,0)
0099 C      CALL EXEC (2,1028,Y,NPT2,NSTRK,1)
0100 C      IF (NX .LT. 0) GO TO 100
0101 C      CALL EXEC (2,1028,X,NPT2,NSTRK,24)
0102 C      100 IF (IP .LT. NP TTL) GO TO 200
0103 C      CALL EXEC (9,NAME,ISTRK,IDISC,ISECT,LU,NP)
0104 C      CALL EXEC (5,-1)
0105 C      200 RETURN
0106 C      END
0107 C      ENDS

```

## 481 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES — 77 —

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

```

0001 FTN4
0002 SUBROUTINE YOKO(X,Y,N,NX,XMIN,XMAX,NY,YMIN,YMAX,
0003 * NP,MO,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0004 DIMENSION X(1),Y(1)
0005 C
0006 C.... THIS SUBROUTINE IS USED TO PLOT Y(I)-X(I) RELATIONSHIP
0007 C IN ANY LOCATION AND SIZE AS DESIRED ON A SHEET.
0008 C THIS IS A MULTI-FUNCTIONAL SUBROUTINE.
0009 C
0010 C.... DATE 06/27/1978
0011 C.... BY DR. H. ISHIKAWA AND D. LI, COLUMBIA UNIVERSITY.
0012 C
0013 C.... DESCRIPTION OF PARAMETERS:
0014 C X -- COORDINATE ARRAY.
0015 C Y -- ABSCISSA ARRAY.
0016 C N -- NUMBER OF POINTS TO BE PLOTTED.
0017 C YMAX -- MAXIMUM VALUE ON Y-AXIS.
0018 C YMIN -- MINIMUM VALUE ON Y-AXIS.
0019 C YMIN AND YMAX ARE EITHER SUPPLIED BY USER
0020 C OR SEARCHED BY THE PROGRAM (SEE ALSO NY).
0021 C NY -- CONTROL CODE ON YMAX AND YMIN.
0022 C IF NY= 1 --- USER SUPPLY FOR YMAX, YMIN.
0023 C NY= 0 --- DEFAULT OPTION.
0024 C PROGRAM SEARCHES YMAX AND YMIN.
0025 C XMAX -- MAXIMUM ORDINATE VALUE ON X-AXIS.
0026 C XMIN -- MINIMUM ORDINATE VALUE ON X-AXIS.
0027 C X-AXIS IS BOUNDED FROM XMIN TO XMAX.
0028 C NX -- CONTROL CODE ON XMIN AND XMAX.
0029 C IF NX= 0 --- PROGRAM SETS AUTOMATICALLY
0030 C XMIN=X(1) AND XMAX=X(N).
0031 C NX= 1 --- USER SUPPLY FOR XMIN AND XMAX.
0032 C NX=-1 --- ACCORDING TO THE USER SUPPLIED
0033 C XMIN AND XMAX, PROGRAM AUTOMATICALLY
0034 C COMPUTES ORDINATE ARRAY X(I) WITH
0035 C DX=(XMAX-XMIN)/N.
0036 C NP -- CONTROL CODE ON NO. OF FRAMES OR FIGURES.
0037 C THIS CONSISTS OF 5 DIGITS (IJKLM) AS FOLLOWS.
0038 C I (=NDIM) -- NO. OF COLUMNS (UP TO 2).
0039 C JK (=NP1) -- NO. OF FRAMES OR GRAPHS IN LEFT COLUMN.
0040 C LM (=NP2) -- NO. OF FRAMES OR GRAPHS IN RIGHT COLUMN.
0041 C NP1TL=NP1+NP2 -- TOTAL NO. OF GRAPHS TO BE DRAWN.
0042 C DEFAULT OPTION -- IF ONE COLUMN, NP CAN BE ONLY
0043 C TWO DIGITS (LM). IN THIS CASE,
0044 C NP1TL=(LM).
0045 C MO -- CONTROL CODE ON THE WAY OF PLOTTING NP GRAPHS.
0046 C IF MO= 0 --- NP1TL PLOTS ON ONE FRAME.
0047 C MO= 1 --- NP1TL PLOTS ON NP1TL FRAMES,
0048 C WITH NP1TL LABELS.
0049 C MO=-1 --- NP1TL PLOTS ON NP1TL FRAMES,
0050 C WITH ONLY ONE LABEL.
0051 C JDCXY -- LABEL DIGITS CONTROL CODE ON X- AND Y-AXIS.
0052 C THIS CONSISTS OF 3 DIGITS (IJK) AS FOLLOWS.
0053 C I -- LABEL CONTROL CODE.
0054 C IF I= 0 -- LABEL IS PRINTED OUT.
0055 C IF OTHERWISE -- NO LABEL IS PRINTED OUT.
0056 C J -- NO. OF DECIMAL POINTS ON X-AXIS.
0057 C FORMAT NOTATION -- F7,J (J=0,1,...,5)
0058 C K -- NO. OF DECIMAL POINTS ON Y-AXIS.
0059 C MINX0,MAXX0,MINY0,MAXY0

```

```

0060 C      -- PARAMETER TO DEFINE THE LOCATION OF THE FIGURE.
0061 C      POSSIBLE LOCATION OF X-AXIS --- 1 TO 1024.
0062 C      POSSIBLE LOCATION OF Y-AXIS --- 1 TO 780.
0063 C      DEFAULT VALUES: MINX0= 10 -- MAXX0=1020
0064 C                        MINY0= 40 -- MAXY0= 750
0065 C      LU      -- TEXTRONIX TERMINAL LOGICAL UNIT NUMBER.
0066 C      IPLOT -- FRAME NUMBER ON THE PICTURE OR
0067 C      GRAPH LINE NUMBER ON THE PICTURE FRAME.
0068 C
0069 C      DATA IPLOT/1/
0070 C      DATA IPLT1/1/
0071 C.... TO SET UP APPROPRIATE VALUE FOR CONTROL.
0072 C      IF (LU .NE. 7) LU=1
0073 C      NDIM=NP/10000
0074 C      NP1=NP/100-NDIM*100
0075 C      NP2=NP-100*NP1-10000*NDIM
0076 C      NP1TL=NP1+NP2
0077 C      IF(NDIM.GT.1) GO TO 5
0078 C      NDIM=1
0079 C      IF(NP1.EQ.0) NP1=NP2
0080 C      IF(NP1.EQ.0) NP1=1
0081 C      NP1TL=NP1
0082 C      5 IF(NP1TL.LE.0) NP1TL=1
0083 C      JDC=JDCXY/100
0084 C      JDCX=JDCXY/10-10*JDC
0085 C      JDCY=JDCXY-10*JDCX-100*JDC
0086 C
0087 C      TEST IF IT IS THE 1ST GRAPH ON THE SAME PICTURE, IF NOT
0088 C      GO TO PLOT DATA IMMEDIATELY
0089 C      IF(IPLOT.NE.1) GO TO 190
0090 C      CALL INITT(960)
0091 C      190 IF(MO .NE. 0) GO TO 14
0092 C      IF(IPLT1.NE.1) GO TO 85
0093 C      14 IF(NY.GT.0) GO TO 20
0094 C      YMAX=-1.0E20
0095 C      YMIN=1.0E+20
0096 C      DO 15 I=1,N
0097 C      IF(Y(I).GT.YMAX) YMAX=Y(I)
0098 C      IF(Y(I).LT.YMIN) YMIN=Y(I)
0099 C      15 CONTINUE
0100 C      IF(YMIN .GE. 0. .OR. YMAX .LE. 0.) GO TO 20
0101 C      IF(YMAX.GT.ABS(YMIN)) YMIN=-YMAX
0102 C      YMAX=ABS(YMIN)
0103 C      20 IF(NX .NE. 0) GO TO 25
0104 C      XMIN=X(1)
0105 C      XMAX=X(N)
0106 C      25 DX=(XMAX-XMIN)/10.
0107 C      DEX=(XMAX-XMIN)/FLOAT(N-1)
0108 C      DY=(YMAX-YMIN)/10.
0109 C      IF(MAXX0.NE.0 .AND. MAXY0.NE.0) GO TO 27
0110 C      MINX0=40
0111 C      MAXX0=1000
0112 C      MINY0=50
0113 C      MAXY0=750
0114 C      27 MLNGX=(MAXX0-MINX0)/NDIM
0115 C      MLNGY=MAXY0-MINY0
0116 C      NLR=1
0117 C      NDIV=NP1
0118 C      IF(IPLOT.LE.NP1) GO TO 28
0119 C      NLR=2
0120 C      NDIV=NP2

```



483 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION -- 79 --  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0121      28 IF (MO.EQ.0) NDIV=1
0122          MCHRX=13
0123          MCHRY=20
0124          MSEG=5
0125          MFRMT=7
0126          MSPCXF=(MFRMT+1)*MCHRX
0127          MSPCXL=(MFRMT-4)*MCHRX
0128          MLINE1=1
0129          MLINE2=5
0130          MLINEY=MLINE2
0131          IF (MO .GT. 0) GO TO 10
0132          IF (NDIV .GT. 3) GO TO 11
0133          MLINE1=2
0134      11 MLINEY=MLINE1
0135      C  MLNGY=MAXY0-MINY0-(MLINE2-MLINE1)
0136      10 MSPCYF=MLINEY*MCHRY
0137          MSPCYL=0
0138          MDELT=MLNGY/NDIV
0139          MINX=MINX0+(NLR-1)*MLNGX
0140          MAXX=MINX+MLNGX
0141          MAXY=MAXY0-(IPLT1-1)*MDELT
0142          MINY=MAXY-MDELT
0143          CALL TWIND(MINX,MAXX,MINY,MAXY)
0144          XINCR=(XMAX-XMIN)/FLOAT(MLNGX-MSPCXF-MSPCXL)
0145          XFST=XMIN-XINCR*MSPCXF
0146          XLST=XMAX+XINCR*MSPCXL
0147          YINCR=(YMAX-YMIN)/FLOAT(MDELT-MSPCYF-MSPCYL)
0148          YFST=YMIN-YINCR*MSPCYF
0149          IF (MO.LT.0 .AND. NDIV.GT.3) YFST=YMIN-1.5*YINCR*MSPCYF
0150          YLST=YMAX+YINCR*MSPCYL
0151          YSEG=YINCR*MSEG
0152          XSEG=XINCR*MSEG
0153          CALL DWIND (XFST,XLST,YFST,YLST)
0154          CALL MOVEA (XMIN,YMAX)
0155          CALL DRAWA (XMIN,YMIN)
0156          CALL DRAWA (XMAX,YMIN)
0157          CALL DRAWA (XMAX,YMAX)
0158          CALL DRAWA (XMIN,YMAX)
0159          IF (YMIN .GE. 0. .OR. YMAX .LE. 0.) GO TO 59
0160          CALL MOVEA (XMIN,0.)
0161          CALL DRAWA (XMAX,0.)
0162      59 IF (ISSW(10).GE.0) GO TO 60
0163          CALL MOVEA(0.,YMIN)
0164          CALL DRAWA(0.,YMAX)
0165      C  DRAW TIC MARKS ON X-AXIS
0166      60 DO 30 I=1,9
0167          30 CALL SEGMT (0,1,XMIN+I*DX,YMIN+YSEG)
0168      C  DRAW TIC MARKS ON Y-AXIS
0169          DO 40 I=1,9
0170          40 CALL SEGMT (1,1,XMIN+XSEG,YMIN+I*DY)
0171      C  LABEL X AXES
0172          IF (JDC.NE.0) GO TO 85
0173          SDY=1.5*YINCR*MCHRY
0174          IF (MO .GT. 0 .OR. NDIV .EQ. 1) GO TO 150
0175          IF (IPLT1 .NE. NDIV) GO TO 82
0176      150 DO 70 I=1,6
0177          XX=XMIN+FLOAT(I-1)*2.0*DX
0178          XDX=XX-XINCR*FLOAT(MCHRX)*(FLOAT(MFRMT)/2.0-FLOAT(JDCX-3))
0179          IF (ISSW(11).GE.0) GO TO 400
0180          CALL MOVEA(XX,YMIN)
0181          CALL DRAWA(XX,YMAX)

```

```

0182 400 CALL MOVEA (XXD,YMIN-SDY)
0183 CALL ANMOD
0184 CALL JWRTE(JDCX,XX,LU)
0185 70 CONTINUE
0186 82 IF(MO.EQ.0 .OR. NDIV.EQ.1) GO TO 131
0187 IF (ABS(YMIN) .EQ. ABS(YMAX)) GO TO 132
0188 C WRITE 5 LABELS IF YMAX AND YMIN ARE NOT EQUAL.
0189 C IF NDIV > 4 --- WRITE 3 LABELS FOR EASY DISPLAY.
0190 IF(NDIV .GT. 4) GO TO 132
0191 DO 140 I=1,6
0192 YY=YMIN+FLOAT(I-1)*2.*DY
0193 YYD=YY-YINCR*FLOAT(MCHRY)/2.0
0194 IF(ISSW(12).GE.0) GO TO 410
0195 CALL MOVEA(XMIN,YY)
0196 CALL DRAWA(XMAX,YY)
0197 410 CALL MOVEA(XFST,YYD)
0198 CALL ANMOD
0199 CALL JWRTE(JDCY,YY,LU)
0200 140 CONTINUE
0201 GO TO 85
0202 C WRITE 3 LABELS IF YMAX=-YMIN
0203 C HOWEVER, WHEN ONLY ONE FRAME, WRITE 11 LABELS.
0204 C 130 IF(MO.EQ.0 .OR. NDIV.EQ.1) GO TO 131
0205 132 DO 80 I=1,3
0206 YY=YMIN+(I-1)*5*DY
0207 YYD=YY-YINCR*FLOAT(MCHRY)/2.0
0208 CALL MOVEA (XFST,YYD)
0209 CALL ANMOD
0210 CALL JWRTE(JDCY,YY,LU)
0211 80 CONTINUE
0212 GO TO 85
0213 C WRITE 11 LABELS.
0214 131 DO 81 I=1,11
0215 YY=YMIN+(I-1)*DY
0216 YYD=YY-YINCR*FLOAT(MCHRY)/2.0
0217 IF(ISSW(12).GE.0) GO TO 420
0218 CALL MOVFA(XMIN,YY)
0219 CALL DRAWA(XMAX,YY)
0220 420 CALL MOVEA(XFST,YYD)
0221 CALL ANMOD
0222 CALL JWRTE(JDCY,YY,LU)
0223 81 CONTINUE
0224 C DRAW VECTORS BETWEEN TIME POINTS
0225 85 IF (NX .GE. 0) GO TO 100.
0226 CALL MOVEA(XMIN,Y(1))
0227 XM=XMIN-DEX
0228 DO 50 I=1,N
0229 XMIDX=XM+I*DEX
0230 CALL DRAWA (XMIDX,Y(I))
0231 50 CONTINUE
0232 GO TO 110
0233 100 CALL MOVEA(X(1),Y(1))
0234 DO 120 I=2,N
0235 IF(X(I).LT.XMIN .OR. Y(I).LT.YMIN) GO TO 121
0236 IF(X(I).GT.XMAX .OR. Y(I).GT.YMAX) GO TO 121
0237 CALL DRAWA(X(I),Y(I))
0238 GO TO 120
0239 121 DIVX=(X(I)-X(I-1))/10.0
0240 DIVY=(Y(I)-Y(I-1))/10.0
0241 DO 122 J=1,10
0242 DXJ=X(I-1)+DIVX*FLOAT(J)

```

## 485 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES — 81 —

```

0243      DYJ=Y(I-1)+DIVY*FLOAT(J)
0244      IF(DXJ,LT,XMIN ,OR. DYJ,LT,YMIN) GO TO 123
0245      IF(DXJ,GT,XMAX ,OR. DYJ,GT,YMAX) GO TO 123
0246      CALL DRAWA(DXJ,DYJ)
0247      GO TO 122
0248      123 CALL MOVEA(DXJ,DYJ)
0249      122 CONTINUE
0250      120 CONTINUE
0251      110 CONTINUE
0252      CALL FINIT (0,767)
0253      IF (IPL0T .EQ. NP TTL) IPLT1=0
0254      IF (IPL0T .EQ. NP TTL) IPL0T=0
0255      IPL0T=IPL0T+1
0256      IF(NDIM,EQ.2 .AND. IPLT1,EQ.NP1) IPLT1=0
0257      IPLT1=IPLT1+1
0258      RETURN
0259      END
0260      SUBROUTINE JW RTE(JDCXY,XY,LU)
0261      C
0262      C.... TO SELECT NUMBER OF DIGITS BELOW DECIMAL POINT.
0263      C
0264      IL=2H+
0265      IF(JDCXY,NE.0) GO TO 10
0266      IXY=IFIX(XY)
0267      WRITE(LU,100) IXY,IL
0268      RETURN
0269      10 IF(JDCXY,EQ.1) WRITE(LU,101) XY,IL
0270      IF(JDCXY,EQ.2) WRITE(LU,102) XY,IL
0271      IF(JDCXY,EQ.3) WRITE(LU,103) XY,IL
0272      IF(JDCXY,EQ.4) WRITE(LU,104) XY,IL
0273      IF(JDCXY,EQ.5) WRITE(LU,105) XY,IL
0274      100 FORMAT(I7,A1)
0275      101 FORMAT(F7.1,A1)
0276      102 FORMAT(F7.2,A1)
0277      103 FORMAT(F7.3,A1)
0278      104 FORMAT(F7.4,A1)
0279      105 FORMAT(F7.5,A1)
0280      RETURN
0281      END
0282      SUBROUTINE SEG MT (NDIR,NSIZE,X,Y)
0283      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0284      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0285      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0286      3 KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBL SZ,
0287      4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0288      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0289      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0290      C THIS ROUTINE WILL DRAW TIC MARK AT LOCATION (X,Y),
0291      C (X,Y) BEING VIRTUAL CO-ORDINATES ON SCREEN
0292      C NDIR --- 0 DRAW VERTICAL TIC MARK
0293      C --- 1 DRAW HORIZONTAL TIC MARK
0294      C NSIZE --- 1 SIZE OF 10 RASTER UNITS
0295      C --- 2 SIZE OF 20 RASTER UNITS
0296      C
0297      C CONVERT TO SCREEN CO-ORDINATES
0298      CALL LVLCH
0299      CALL V2ST (1,X,Y,IX,IY)
0300      C SKIP IF POINT COMPLETELY OUTSIDE WINDOW
0301      IF(KGNFLG,EQ.1) GO TO 60
0302      CALL VECMD
0303      IF(NDIR,EQ.1) GO TO 30

```

```
0304      CALL XYCNT (IX,IY-NSIZE*5)
0305      GO TO 40
0306 30 CALL XYCNT (IX-NSIZE*5,IY)
0307 40 IF(KKMODE,NE,1) CALL VECMD
0308     IF(KMOVEF,EQ,1) CALL XYCNT(KBEAMX,KBEAMY)
0309     IF(NDIR,EQ,1) GO TO 50
0310     CALL XYCNT (IX,IY+NSIZE*5)
0311     GO TO 60
0312 50 CALL XYCNT (IX+NSIZE*5,IY)
0313 60 RETURN
0314     END
0315     ENDS
```

487 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 83 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

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

```

0001 FTN4
0002     PROGRAM CHIE,3,10
0003     DIMENSION X(1400),Y(1400),NA(20),AA(5),IPARM(5)
0004     COMMON LARY(100)
0005 C     EQUIVALENCE (AA,NA(11)),(X,Y(1401))
0006     EQUIVALENCE (AA,NA(11))
0007 C     THIS PROGRAM SCHEDULED BY SUBROUTINE STUKO,
0008 C     THE MAXIMUM DATA POINT TO BE PLOTTED IS 1400 POINTS.
0009 C     WHEN NX = -1 MAXIMUM NPT=2800 POINTS FOR Y ARRAY ONLY.
0010 C     PARAMETER TRANSFER FROM SUBROUTINE CHIE AS FOLLOWS
0011 C     ISTRK = STARTING DISC TRACK FOR DATA STORAGE.
0012 C     IDISC = DISC LU NUMBER.
0013 C     ISECT = NUMBER OF SECTOR FOR ONE TRACK.
0014 C     LU = LOGIC NUMBER FOR TEKTRONIX TERMINAL.
0015 C     NP = CONSISTS OF 5 DIGITS (IJKLM) AS FOLLOWS.
0016 C     I (=NDIM) -- NO. OF COLUMNS (UP TO 2).
0017 C     JK (=NP1) -- NO. OF FRAMES OR GRAPHS IN LEFT COLUMN.
0018 C     LM (=NP2) -- NO. OF FRAMES OR GRAPHS IN RIGHT COLUMN.
0019 C     NP TTL=NP1+NP2 -- TOTAL NO. OF GRAPHS TO BE DRAWN.
0020 C
0021 C.... DATE 06/26/1978
0022 C.... BY DR. H. ISHIKAWA AND D. LI, COLUMBIA UNIVERSITY.
0023 C
0024     CALL RMPAR(IPARM)
0025     ISTRK=IPARM(1)
0026     IDISC=IPARM(2)
0027     ISECT=IPARM(3)
0028     LU =IPARM(4)
0029     NP =IPARM(5)
0030     NDIM=NP/10000
0031     NP1=NP/100-100*NDIM
0032     NP2=NP-100*NP1-10000*NDIM
0033     NP TTL=NP1+NP2
0034     IF(NDIM.GT.1) GO TO 5
0035     IF(NP1.EQ.0) NP1=NP2
0036     IF(NP1.EQ.0) NP1=1
0037     NP TTL=NP1
0038 S IF(NP TTL.LE.0) NP TTL=1
0039     DO 200 I=1,NP TTL
0040         NSTRK=ISTRK+I-1
0041         CALL EXEC (1,1028,NA,20,NSTRK,0)
0042         NPT=NA(1)
0043         NX=NA(2)
0044         NY=NA(3)
0045         MO=NA(4)
0046         JDCXY=NA(5)
0047         MINX0=NA(6)
0048         MAXX0=NA(7)
0049         MINY0=NA(8)
0050         MAXY0=NA(9)
0051         XMIN=AA(1)
0052         XMAX=AA(2)
0053         YMIN=AA(3)
0054         YMAX=AA(4)
0055         NPT2=2*NPT
0056         CALL EXEC (1,1028,Y,NPT2,NSTRK,1)
0057         IF (NX .LT. 0) GO TO 100
0058         CALL EXEC (1,1028,X,NPT2,NSTRK,24)
0059 100 CALL YOKO(X,Y,NPT,NX,XMIN,XMAX,NY,YMIN,YMAX,

```

```
0060      * NP,MO,JDCXY,MINX0,MAXX0,MINY0,MAXY0)
0061      200 CONTINUE
0062      CALL EXEC (6)
0063      END
0064      ENDS
```

489 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 85 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

TTTLB T=00004 IS ON CR00002 USING 00111 BLKS R=0800

```

0001 FTN4
0002     SUBROUTINE LVLCH
0003     COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0004     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0005     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0006     3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0007     4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0008     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0009     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0010     IF(KGRAFL.NE.0) GO TO 10
0011     CALL REVCO (KBEAMX,KBEAMY,TREALX,TREALY)
0012     TIMAGX=TREALX
0013     TIMAGY=TREALY
0014     KGRAFL=1
0015     10 RETURN
0016     END
0017     SUBROUTINE MOVEA (X,Y)
0018     COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0019     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0020     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0021     3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0022     4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0023     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0024     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0025     CALL LVLCH
0026     C    CONVERT TO SCREEN CO-ORDINATES
0027     CALL V2ST (1,X,Y,IX,IY)
0028     C    SKIP IF LINE IS COMPLETELY OUTSIDE WINDOW
0029     IF(KGNFLG.EQ.1) GO TO 10
0030     CALL VECMD
0031     CALL XYZNT (IX,IY)
0032     10 RETURN
0033     END
0034     SUBROUTINE PARCL (RL1,RL2,RM1,RM2,RN1,RN2)
0035     IF(RL1.LT.RM1) GO TO 10
0036     IF(RL1.GT.RM2) GO TO 20
0037     RN1=RL1
0038     IF(RL2-RM1) 30,40,40
0039     10 RN1=RM1
0040     IF(RL2.LE.RM2) GO TO 50
0041     RN2=RM2
0042     GO TO 60
0043     50 RN2=RL2
0044     GO TO 60
0045     20 RN1=RM2
0046     IF(RL2.GE.RM1) GO TO 50
0047     30 RN2=RM1
0048     60 RETURN
0049     END
0050     SUBROUTINE DRAWA (X,Y)
0051     COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0052     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0053     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0054     3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0055     4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0056     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0057     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0058     C    SET TERMINAL TO DRAW SOLID LINES IF NEEDED
0059     CALL LVLCH

```

```

0060 C   CONVERT TO SCREEN CO-ORDINATES
0061     CALL V2ST (1,X,Y,IX,IY)
0062 C   SKIP IF LINE COMPLETELY OUTSIDE WINDOW
0063     IF(KGNFLG.EQ.1) GO TO 10
0064     IF(KKMODE.NE.1) CALL VECMD
0065     IF(KMOVEF.EQ.1) CALL XYZNT (KBEAMX,KBEAMY)
0066     CALL XYZNT (IX,IY)
0067   10 RETURN
0068     END
0069     SUBROUTINE DWIND (XMIN,XMAX,YMIN,YMAX)
0070     COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0071     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0072     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0073     3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0074     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0075     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0076     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0077 C   DEFINE DATA WINDOW IN TERMINAL COMMON AREA
0078     TMINVX=XMIN
0079     TMAXVX=XMAX
0080     TMINVY=YMIN
0081     TMAXVY=YMAX
0082     CALL RESCL
0083     RETURN
0084     END
0085     SUBROUTINE WINCO (X,Y,IX,IY)
0086     COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0087     1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0088     2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0089     3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0090     4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0091     5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0092     6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0093 C   DATA DE2RAD/0.01745/
0094 C   CHECK FOR PERMITTED VALUE OF CONVERSION KEY
0095 C   DEFAULT IS LINEAR, ERROR IS NONE
0096     DX=X-TMINVX
0097     DY=Y-TMINVY
0098     IF (KEYCON.EQ. 1) GO TO 500
0099 C   KEY=KEYCON
0100 C   IF(KEYCON.LT.1) KEY=5
0101 C   IF(KEYCON.GT.4) KEY=4
0102 C   BRANCH TO PROPER SECTION
0103 C   LINEAR LOG POLAR USER ERROR
0104 C   GO TO (500,300,600,700,100),KEY
0105 C   ERROR
0106   100 IX=X
0107     IY=Y
0108     GO TO 800
0109 C   LOG TRANSFORM
0110 C   300 KEYL=TRPAR1+0.001
0111 C   IF(KEYL.EQ.2) GO TO 400
0112 C   SETUP X LOG TRANSFORM
0113 C   DX=ALOG(X)-TRPAR2
0114 C   400 IF(KEYL.EQ.1) GO TO 500
0115 C   SETUP Y LOG TRANSFORM
0116 C   DY=ALOG(Y)-TRPAR3
0117 C   CONVERT LINEAR
0118   500 IX=IFIX(DX*TRFACX)+KMINSX
0119     IY=IFIX(DY*TRFACY)+KMINSY
0120 C   GO TO EXIT

```



491 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 87 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0121 C      GO TO 800
0122 C      POLAR TRANSFORM
0123 C 600  A=(Y-TRPAR6)*TRFACY
0124 C      R=(X-TRPAR5)*TRFACX
0125 C      IX=R*COS(A*DE2RAD)+TRPAR3
0126 C      IY=R*SIN(A*DE2RAD)+TRPAR4
0127 C      GO TO EXIT
0128 C      GO TO 800
0129 C      USER TRANSFORM IN USE
0130 C 700  CONTINUE
0131 C      CALL USECO (X,Y,IX,IY)
0132 C      EXIT POINT
0133 C      800  RETURN
0134 C      END
0135 C      SUBROUTINE V2ST (I,X,Y,IX,IY)
0136 C      DIMENSION BUFIN(4),BFOUT(4)
0137 C      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0138 C      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0139 C      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0140 C      3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0141 C      4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0142 C      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0143 C      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0144 C      EQUIVALENCE (BUFIN(1),XS),(BUFIN(2),YS),(BUFIN(3),XE),
0145 C      1 (BUFIN(4),YE)
0146 C      EQUIVALENCE (BFOUT(1),CX),(BFOUT(2),CY),(BFOUT(3),CXE),
0147 C      1 (BFOUT(4),CYE)
0148 C      XE=X
0149 C      YE=Y
0150 C      POINT OR MOVE
0151 C      IF(I.EQ.0) GO TO 10
0152 C      BRIGHT VECTOR
0153 C      XS=TIMAGX
0154 C      YS=TIMAGY
0155 C      CLIP VECTOR
0156 C      CALL CLIPT (BUFIN,BFOUT)
0157 C      ON SCREEN
0158 C      IF(KGNFLG.EQ.1) GO TO 110
0159 C      ARE WE AT START POINT
0160 C      IF(CX.EQ.TREALX.AND.CY.EQ.TREALY) GO TO 120
0161 C      MOVE BEAM TO START POINT
0162 C      MODE=KKMODE
0163 C      CALL VECMD
0164 C      CALL WINCO (CX,CY,IX,IY)
0165 C      CALL XYCNT (IX,IY)
0166 C      KKMODE=MODE
0167 C      GO TO 120
0168 C      POINT OR MOVE
0169 C      10  CALL PCLIP (XE,YE)
0170 C      OFF SCREEN
0171 C      IF(KGNFLG.EQ.1) GO TO 110
0172 C      CXE=XE
0173 C      CYE=YE
0174 C      CONVERT TO SCREEN CO-ORDINATES
0175 C      120  CALL WINCO (CXE,CYE,IX,IY)
0176 C      SAVE POSITION ABS AND IMAGINARY
0177 C      TREALX=CXE
0178 C      TREALY=CYE
0179 C      110  TIMAGX=X
0180 C      TIMAGY=Y
0181 C      RETURN

```

```

0182      END
0183      SUBROUTINE REVCO (IX,IY,X,Y)
0184      LOGICAL DEC
0185      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0186      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0187      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0188      3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVRSZ,KTBSZ,
0189      4 KSIZF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0190      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0191      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0192      C      E=2.718281828
0193      DX=FLOAT(IX-KMINSX)/TRFACX
0194      DY=FLOAT(IY-KMINSY)/TRFACY
0195      IF (KEYCON.EQ.1) GO TO 300
0196      C      KEY=KEYCON
0197      C      IF (KEYCON.LT.1) KEY=5
0198      C      IF (KEYCON.GT.4) KEY=4
0199      C      LINEAR LOG POLAR USER ERROR
0200      C      GO TO (300,400,500,600,100),KEY
0201      C      ERROR
0202      100 X=IX
0203      Y=IY
0204      GO TO 700
0205      C      LINEAR
0206      300 X=DX+TMINVX
0207      Y=DY+TMINVY
0208      C      GO TO 700
0209      C      LOG SCALES
0210      C      400 KEYL=TRPAR1
0211      C      X=DX+TMINVX
0212      C      Y=DY+TMINVY
0213      C      IF (KEYL.NE.2) X=E**(DX+TRPAR2)
0214      C      IF (KEYL.NE.1) Y=E**(DY+TRPAR3)
0215      C      GO TO 700
0216      C      POLAR
0217      C      500 DX=FLOAT(IX)-TRPAR3
0218      C      DY=FLOAT(IY)-TRPAR4
0219      C      Y=ATAN2(DX,DY)*57.295780
0220      C      X=SQRT(DY*DY+DX*DX)/TRFACX+TRPAR5
0221      C      ADJUST ANGLE MOD 2 PI TO VALUE WITHIN WINDOW
0222      C      DEC=.FALSE.
0223      C      510 IF (Y.GT.TRPAR1) GO TO 530
0224      C      INCREMENT ANGLE
0225      C      Y=Y+360.
0226      C      GO TO 510
0227      C      530 IF (Y.LE.TRPAR2) GO TO 550
0228      C      DECREMENT ANGLE
0229      C      Y=Y-360.
0230      C      DEC=.TRUE.
0231      C      GO TO 530
0232      C      550 IF (DEC.AND.Y.LT.TRPAR1) Y=Y+360.0
0233      C      IF (TMINVX.GE.0) GO TO 560
0234      C      TR1A=AMOD(TRPAR1+180.,360.)
0235      C      TR2A=AMOD(TRPAR2+180.,360.)
0236      C      IF (Y.GT.AMAX1(TR1A,TR2A).OR.Y.LT.AMIN1(TR1A,TR2A)) GO TO 560
0237      C      Y=AMOD(Y+180.,360.)
0238      C      X=-X
0239      C      560 Y=Y/TRFACY+TRPAR6
0240      C      GO TO 700
0241      C      USER CONVERSION
0242      C      600 CONTINUE

```

493 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 89 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0243 C   CALL UREVC (IX,IT,X,Y)
0244 C   EXIT POINT
0245     700 CALL PCLIP (X,Y)
0246       RETURN
0247       END
0248     SUBROUTINE CLIPT (BUFIN,OUTBF)
0249       COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0250       1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0251       2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPA02,
0252       3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0253       4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0254       5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0255       6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0256       DIMENSION BUFIN(1),OUTBF(1)
0257       GSTAX=BUFIN(1)
0258       GSTAY=BUFIN(2)
0259       GENDX=BUFIN(3)
0260       GENDY=BUFIN(4)
0261       IF(GSTAX.GE.TMINVX) GO TO 10
0262       IF(GENDX.GE.TMINVX) GO TO 20
0263       GO TO 110
0264     10 IF(GSTAX.LE.TMAXVX) GO TO 20
0265       IF(GENDX.LE.TMAXVX) GO TO 20
0266       GO TO 110
0267     20 IF(GSTAY.GE.TMINVY) GO TO 21
0268       IF(GENDY.GE.TMINVY) GO TO 30
0269       GO TO 110
0270     21 IF(GSTAY.LE.TMAXVY) GO TO 30
0271       IF(GENDY.LE.TMAXVY) GO TO 30
0272       GO TO 110
0273     30 IF(GSTAX.NE.GENDX) GO TO 31
0274       DSTAX=GSTAX
0275       DENDX=GSTAX
0276       CALL PARCL (GSTAY,GENDY,TMINVY,TMAXVY,DSTAY,DENDY)
0277       GO TO 120
0278     31 IF(GSTAY.NE.GENDY) GO TO 40
0279       DSTAY=GSTAY
0280       DENDY=GSTAY
0281       CALL PARCL (GSTAX,GENDX,TMINVX,TMAXVX,DSTAX,DENDX)
0282       GO TO 120
0283     40 A=GENDX-GSTAX
0284       B=GENDY-GSTAY
0285       IF(GSTAX.LT.TMINVX) GO TO 41
0286       IF(GSTAX.LE.TMAXVX) GO TO 43
0287       Q=TMAXVX
0288       GO TO 42
0289     43 IF(GSTAY.GT.TMAXVY) GO TO 140
0290       IF(GSTAY.LT.TMINVY) GO TO 44
0291       DSTAX=GSTAX
0292       DSTAY=GSTAY
0293       GO TO 150
0294     41 Q=TMINVX
0295     42 DSTAY=GSTAY+((Q-GSTAX)*B/A)
0296       IF(DSTAY.GT.TMAXVY) GO TO 140
0297       IF(DSTAY.LT.TMINVY) GO TO 44
0298       DSTAX=Q
0299       GO TO 150
0300     44 R=TMINVY
0301       GO TO 45
0302     140 R=TMAXVY
0303     45 DSTAX=GSTAX+((R-GSTAY)*A/B)

```

```

0304      IF(DSTAX.GT.TMAXVX) GO TO 110
0305      IF(DSTAX.LT.TMINVX) GO TO 110
0306      DSTAY=R
0307  150  IF(GENDX.LT.TMINVX) GO TO 50
0308      IF(GENDX.GT.TMAXVX) GO TO 51
0309      IF(GENDY.GT.TMAXVY) GO TO 160
0310      IF(GENDY.LT.TMINVY) GO TO 52
0311      DENDX=GENDX
0312      DENDY=GENDY
0313      GO TO 120
0314  51  Q=TMAXVX
0315      GO TO 53
0316  50  Q=TMINVX
0317  53  DENDY=GSTAY+((Q-GSTAX)*B/A)
0318      IF(DENDY.GT.TMAXVY) GO TO 160
0319      IF(DENDY.LT.TMINVY) GO TO 52
0320      DENDX=Q
0321      GO TO 120
0322  52  R=TMINVY
0323      GO TO 60
0324  160 R=TMAXVY
0325  60  DENDX=GSTAX+((R-GSTAY)*A/B)
0326      DENDY=R
0327  120 OUTBF(1)=DSTAX
0328      OUTBF(2)=DSTAY
0329      OUTBF(3)=DENDX
0330      OUTBF(4)=DENDY
0331      KGNFLG=0
0332      GO TO 70
0333  C   SET FLAG IF LINE OUTSIDE WINDOW
0334  110 KGNFLG=1
0335  70  RETURN
0336      END
0337      SUBROUTINE RESET
0338      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0339      1  TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0340      2  TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0341      3  KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0342      4  KSIZEF,KLMRGN,KRMRGN,KFACTOR,KTERM,KLINE,KZAXIS,KBEAMX,
0343      5  KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0344      6  KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0345      KEYCON=1
0346      TRFACX=1.
0347      TRFACY=1.
0348      KBEAMX=0
0349      KHOMEX=3068/KFACTOR
0350      KBEAMY=KHOMEX
0351      KMINSX=0
0352      KMAXSX=4096/KFACTOR
0353      KMINSY=0
0354      KMAXSY=3120/KFACTOR
0355      KHORSZ=56
0356      KLINE=0
0357      KZAXIS=0
0358      KLMRGN=0
0359      KRMRGN=4040/KFACTOR
0360      KSIZEF=1
0361      KTBSZ=10
0362      KVERSZ=88
0363      TMINVX=0.
0364      TMAXVX=4095./FLOAT(KFACTOR)

```

## 495 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION OF DATA-BASED NONSTATIONARY RANDOM PROCESSES — 91 —

```

0365      TMINVY=0.
0366      TMAXVY=3120./FLOAT(KFACTR)
0367      TRCOSF=1.
0368      TRSINF=0.
0369      TRSCAL=1.
0370 C      MOVE TO THE HOME POSITION
0371      CALL MOVAB (KLMRGN,KHOMY)
0372 C      PLACE TERMINAL IN A/N MODE
0373      CALL ALFMD
0374      RETURN
0375      END
0376      SUBROUTINE NWPAG
0377      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0378      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0379      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0380      3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0381      4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0382      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0383      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0384      DIMENSION ICODE(2)
0385      DATA ICODE(1),ICODE(2)/27,12/
0386      IF(KKMODE,NE.0) CALL ALFMD
0387 C      OUTPUT (ESC) (FF) FOR NEW PAGE
0388      CALL TOUTS (2,ICODE)
0389      CALL IOWAI (10)
0390      IF(KLMRGN,EQ.0) GO TO 10
0391      CALL MOVAB (KLMRGN,KHOMY)
0392      CALL ALFMD
0393      GO TO 20
0394      10 KBEAMX=0
0395      KBEAMY=KHOMY
0396      20 RETURN
0397      END
0398      SUBROUTINE IOWAI (ITIME)
0399 C      THIS ROUTINE IS USED TO GENERATE DELAYS FOR REMOTE TERMINALS
0400      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0401      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0402      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0403      3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0404      4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0405      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0406      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0407      IF(KBAUDR,LE.0) GO TO 20
0408      KOUNT=ITIME*KBAUDR/10
0409      DO 10 J=1,KOUNT
0410 C      OUTPUT (SYN) TO INSURE AGAINST LOSS OF OUTPUT WHILE
0411 C      TERMINAL IS BUSY. (SYN) DOES NOT AFFECT THE TERMINAL
0412      10 CALL TOUTP (22)
0413      20 RETURN
0414      END
0415      SUBROUTINE XYCNT (IX,IY)
0416      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0417      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0418      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0419      3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0420      4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0421      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0422      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0423      DIMENSION IPLT(5),IOPT(5)
0424      DATA IDREW,ISYN/0,22/
0425 C      RECEIVE THE PLOT CHARACTERS

```

```

0426      CALL PLCHR (IX,IY,IPLT)
0427 C      OPTIMIZE THE OUTPUT
0428      LEN=0
0429 C      CHECK IF HIGH Y IS NEEDED
0430      IF(KPCHAR(1).EQ.IPLT(1)) GO TO 10
0431 C      INCLUDE HIGH Y IF NEEDED
0432      LEN=1
0433      KPCHAR(1)=IPLT(1)
0434      IOPT(1)=IPLT(1)
0435 C      CHECK IF LSBXY IS NEEDED
0436      10 IF(KTERM.LE.2) GO TO 20
0437      IF(KPCHAR(2).EQ.IPLT(2)) GO TO 20
0438 C      INCLUDE LSBXY IF NEEDED
0439      LEN=LEN+1
0440      KPCHAR(2)=IPLT(2)
0441      IOPT(LEN)=IPLT(2)
0442      GO TO 30
0443 C      CHECK IF LOW Y IS NEEDED
0444      20 IF(KPCHAR(3).NE.IPLT(3)) GO TO 30
0445      IF(KPCHAR(4).EQ.IPLT(4)) GO TO 40
0446 C      INCLUDE LOW Y IF NEEDED
0447      30 LEN=LEN+1
0448      KPCHAR(3)=IPLT(3)
0449      IOPT(LEN)=IPLT(3)
0450 C      CHECK IF HIGH X IS NEEDED
0451      IF(KPCHAR(4).EQ.IPLT(4)) GO TO 50
0452 C      INCLUDE HIGH X IF NEEDED
0453      LEN=LEN+1
0454      KPCHAR(4)=IPLT(4)
0455      IOPT(LEN)=IPLT(4)
0456 C      CHECK IF LOW X IF NEEDED
0457      40 IF(KPCHAR(5).NE.IPLT(5)) GO TO 50
0458 C      CHECK IF ALL THE CHARACTERS ARE THE SAME
0459      IF(LEN.NE.0) GO TO 50
0460 C      CHECK IF (GS) FOR DARK VECTOR ALREADY SENT
0461      IF(KMOVEF.EQ.1) GO TO 50
0462 C      CHECK IF VECTOR IS ALREADY DRAWN TO SPOT
0463      IF(IDREW.EQ.1) GO TO 80
0464 C      INCLUDE THE LOW X
0465      50 LEN=LEN+1
0466      KPCHAR(5)=IPLT(5)
0467      IOPT(LEN)=IPLT(5)
0468 C      CHECK FOR POSSIBLE SPEED PROBLEM
0469      60 IF(LEN.GE.KPAD2) GO TO 70
0470      LEN=LEN+1
0471      IOPT(LEN)=ISYN
0472      GO TO 60
0473 C      SEND THE ARRAY TO THE OUTPUT BUFFER
0474      70 CALL TOUTS (LEN,IOPT)
0475 C      SET COMMON AND HISTORY VARIABLES
0476 C      SET THE DREW HERE FLAG
0477      IDREW=1
0478 C      REMOVE THE DREW HERE FLAG IF DIDN'T DRAW
0479      IF(KMOVEF.EQ.1) IDREW=0
0480 C      REMOVE THE MOVE FLAG
0481      KMOVEF=0
0482      80 KBEAMX=IX
0483      KBEAMY=IY
0484      RETURN
0485      END
0486      SUBROUTINE ALFMD

```

497 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 93 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0487      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0488      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0489      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0490      3 KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0491      4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0492      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0493      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0494      SET ALPHA MODE OUTPUT (US)
0495      CALL TOUTP (31)
0496      KKMODE=0
0497      IF(KBEAMY.GT.KHOMEX) KBEAMY=KHOMEX
0498      RETURN
0499      END
0500      SUBROUTINE TOUTP (KKOUT)
0501      DIMENSION KOUT(1)
0502      KOUT(1)=KKOUT
0503      CALL TOUTS (1,KOUT)
0504      RETURN
0505      END
0506      SUBROUTINE TSEND
0507      DIMENSION ITEMP(1)
0508      CALL BUFFK (0,ITEMP)
0509      RETURN
0510      END
0511      SUBROUTINE RESCL
0512      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0513      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0514      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0515      3 KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0516      4 KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0517      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0518      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0519      IF (KEYCON.NE. 1) GO TO 500
0520      C      KEY=KEYCON
0521      C      IF(KEYCON.LT.1) KEY=5
0522      C      IF(KEYCON.GT.4) KEY=4
0523      C      BRANCH TO PROPER SECTION AND RETURN
0524      C      LINEAR LOG POLAR USER ERROR
0525      C      GO TO (100,200,300,400,500),KEY
0526      C      BOTH AXES LINEAR
0527      C100  TRPAR1=0.
0528      C      SEMI LOG OR LOG-LOG
0529      C200  KEYL=TRPAR1+1.0001
0530      C      X-AXIS LINEAR OR LOG
0531      C      GO TO (210,220,210,220),KEYL
0532      C      LINEAR
0533      C210  TRFACX=FLOAT(KMAXSX-KMINSX)/(TMAXVX-TMINVX)
0534      C      GO TO 250
0535      C      SEMI-LOG X-AXIS
0536      C220  TRPAR2=ALOG(TMINVX)
0537      C      TRFACX=FLOAT(KMAXSX-KMINSX)/(ALOG(TMAXVX)-TRPAR2)
0538      C      Y-AXIS LINEAR OR LOG
0539      C250  GO TO (260,260,280,280),KEYL
0540      C      LINEAR
0541      C260  TRFACY=FLOAT(KMAXSY-KMINSY)/(TMAXVY-TMINVY)
0542      C      GO TO 600
0543      C      SEMI-LOG Y-AXIS
0544      C280  TRPAR3=ALOG(TMINVY)
0545      C      TRFACY=FLOAT(KMAXSY-KMINSY)/(ALOG(TMAXVY)-TRPAR3)
0546      C      GO TO 600
0547      C      POLAR SCALING

```

```

0548 C300 CONTINUE
0549 C CALL PSCAL
0550 C GO TO 600
0551 C USER DEFINE FUNCTION
0552 C400 CONTINUE
0553 C CALL URSCL
0554 C GO TO 600
0555 C NO SCALE
0556 500 TRFACX=1.
0557 TRFACY=1.
0558 600 RETURN
0559 END
0560 SUBROUTINE MOVAB (IX,IY)
0561 COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0562 1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0563 2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0564 3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0565 4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0566 5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0567 6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0568 CALL VECMD
0569 CALL XYCNT (IX,IY)
0570 KGRAFL=0
0571 RETURN
0572 END
0573 SUBROUTINE VECMD
0574 COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0575 1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0576 2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0577 3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0578 4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0579 5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0580 6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0581 IF(KKMODE.EQ.1) GO TO 10
0582 C OUTPUT (US) TO ENTER A/N MODE AND RESET FOR VECTOR MODE
0583 CALL TOUTP (31)
0584 DO 112 II=1,5
0585 112 KPCHAR(II)=1
0586 KKMODE=1
0587 C OUTPUT (GS) TO ENTER VECTOR MODE
0588 10 CALL TOUTP (29)
0589 KMOVEF=1
0590 RETURN
0591 END
0592 SUBROUTINE PLCHR (IX,IY,ICHAR)
0593 COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0594 1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0595 2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0596 3 KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0597 4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0598 5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0599 6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0600 DIMENSION ICHAR(1)
0601 C CALCULATE THE PLOT CHARACTERS TO ARRIVE AT IX,IY
0602 C ORDER IS HIY,LSBYX,LOY,HIX,LOX
0603 KX=IX*KFACTR
0604 KY=IY*KFACTR
0605 ICHAR(1)=MOD(KY/128,32)+32
0606 ICHAR(2)=MOD(KY,4)*4+MOD(KX,4)+96
0607 ICHAR(3)=MOD(KY/4,32)+96
0608 ICHAR(4)=MOD(KX/128,32)+32

```



499 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 95 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0609      ICHAR(5)=MOD(KX/4,32)+64
0610      RETURN
0611      END
0612      SUBROUTINE INITT (IBAUD)
0613      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0614      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0615      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0616      3 KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0617      4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0618      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0619      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0620      KBAUDR=IBAUD
0621      KPAD2=KBAUDR/220+1
0622      KTERM=1
0623      KFACTR=4
0624      C      SET THE OUTPUT BUFFER FORMAT
0625      KUNIT=3
0626      KINLFT=0
0627      KOTLFT=1
0628      CALL RESET
0629      CALL NWPAG
0630      RETURN
0631      END
0632      SUBROUTINE FINIT (IX,IY)
0633      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0634      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0635      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0636      3 KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0637      4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0638      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0639      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0640      CALL MOVAB (IX,IY)
0641      CALL ALFMD
0642      CALL TEND
0643      RETURN
0644      END
0645      SUBROUTINE TOUTS (LEN,IADE)
0646      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0647      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0648      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0649      3 KBAUDR,KGNFLG,KGRAFL,KHOMEX,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0650      4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0651      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0652      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0653      DIMENSION IADE(1),IUSE(144)
0654      DATA ISYNC,MAXLEN/22,144/
0655      IF(LEN.LE.0) GO TO 40
0656      LENOUT=0
0657      ITEST=KPAD2-1
0658      DO 30 I=1,LEN
0659      ITEMP=IADE(I)
0660      C      INSERT CODE EXPANSION CHARACTER HERE WHEN NEEDED
0661      IF(LENOUT.GE.MAXLEN) GO TO 40
0662      LENOUT=LENOUT+1
0663      30 IUSE(LENOUT)=ITEMP
0664      40 CALL BUFFK (LENOUT,IUSE)
0665      RETURN
0666      END
0667      SUBROUTINE BUFFK (NCHAR,IOUT)
0668      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0669      1 TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,

```

```

0670      2 TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0671      3 KBAUDR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
0672      4 KSIZEF,KLMRGN,KMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0673      5 KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0674      6 KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0675      DIMENSION IDATA(144),IOUT(1)
0676      DATA MAXLEN,LENOUT,NODATA,ITRAIL/144,0,1,1/
0677 C      DUMP THE BUFFER WHEN REQUESTED BY LEN=0
0678      IF(NCHAR,LE,0) GO TO 10
0679 C      DON'T DUMP THE BUFFER IF THE NEW STRING WILL FIT
0680      IF(NCHAR,LE,KOTLFT) GO TO 70
0681 C      DETERMINE IF THERE IS DATA IN BUFFER
0682      10 IF(NODATA,EQ,1) GO TO 50
0683 C      DETERMINE THE FORMAT THE USER WANTS BUFFER DUMPED IN
0684      GO TO (20,30,40),KUNIT
0685 C      OUTPUT BUFFER FORMAT IS (GS),PLTCHRS,DATA,(US)
0686      20 LENOUT=LENOUT+1
0687 C      APPEND (US) TO END OF BUFFER
0688      IDATA(LENOUT)=31
0689      CALL ADOUT (LENOUT,IDATA)
0690      CALL PLCHR (KBEAMX,KBEAMY,IDATA)
0691 C      RESORE THE BEAM POSITION AT FIRST OF THE NEXT BUFFER
0692      IDATA(2)=IDATA(1)
0693      IDATA(1)=29
0694 C      AND NOW THE MODE BEFORE THE OUTPUT IS ASKED FOR
0695      LENOUT=6
0696      KEY=KKMODE+1
0697      IF(KEY,LT,1) KEY=1
0698      IF(KEY,GT,5) KEY=1
0699 C      MODE IS A/N,VECTOR,PNT,INC,DSH
0700      GO TO (21,22,23,24,22),KEY
0701 C      ENTER A/N MODE
0702      21 IDATA(LENOUT)=31
0703      GO TO 50
0704 C      ENTER VECTOR MODE
0705      22 IDATA(LENOUT)=29
0706      IF(KMOVEF,NE,1) LENOUT=LENOUT-1
0707      GO TO 50
0708 C      ENTER POINT MODE
0709      23 IF(KTERM,LT,3) GO TO 22
0710      IDATA(LENOUT)=28
0711      LENOUT=LENOUT+1
0712      GO TO 22
0713 C      ENTER INCREMENT PLOT MODE
0714      24 IDATA(LENOUT)=30
0715      GO TO 50
0716 C      OUTPUT BUFFER FORMAT IS (SYN),DATA,(ESC)
0717      30 LENOUT=LENOUT+1
0718 C      APPEND (ESC) TO END THE BUFFER
0719      IDATA(LENOUT)=27
0720      CALL ADOUT (LENOUT,IDATA)
0721      IDATA(1)=22
0722      LENOUT=1
0723      GO TO 50
0724 C      OUTPUT BUFFER FORMAT IS DATA ONLY
0725      40 CALL ADOUT (LENOUT,IDATA)
0726      LENOUT=0
0727      50 KOTLFT=MAXLEN-LENOUT-ITRAIL
0728 C      TEKTRONIX BUG FIXED IN NEXT LINE
0729      IF(NCHAR,LE,0) GO TO 90
0730      70 NODATA=0

```

501 COMPUTER PROGRAM DEVELOPMENT FOR THE SIMULATION — 97 —  
OF DATA-BASED NONSTATIONARY RANDOM PROCESSES

```

0731      LEN=NCHAR
0732      IF(LEN.GT.KOTLFT) LEN=KOTLFT
0733      DO 80 I=1,LEN
0734      LENOUT=LENOUT+1
0735      80  IDATA(LENOUT)=IOUT(I)
0736      90  KOTLFT=MAXLEN=LENOUT-ITRIL
0737      RETURN
0738      END
0739      SUBROUTINE ADOUT (NCHAR,IARAY)
0740      DIMENSION IARAY(1),KARAY(73)
0741      COMMON LARRY(99),LU
0742      IF (LU .NE. 7) LU=1
0743      C   TEST FOR NCHAR=0
0744      IF(NCHAR.LE.0) RETURN
0745      C   OUTPUT ASCII CODE (95) TO SUPPRESS (CR) AND (LF)
0746      C   PACK AOE ARRAY INTO A2 ARRAY
0747      KK=(NCHAR+1)/2
0748      DO 10 I=1,KK
0749      II=I+I-1
0750      KARAY(I)=IARAY(II)*256
0751      IF (II .EQ. NCHAR) GO TO 20
0752      10  KARAY(I)=KARAY(I)+IARAY(I+I)
0753      KARAY(KK+1)=95*256
0754      GO TO 30
0755      20  KARAY(I)=KARAY(I)+95
0756      C   SEND THE ARRAY,SUPPRESSING CR AND LF.
0757      30  K=-NCHAR-1
0758      CALL EXEC (2,LU,KARAY,K)
0759      RETURN
0760      END
0761      SUBROUTINE ANMOD
0762      C   ENTER ALPHA-NUMERIC MODE
0763      CALL ALFMD
0764      C   DUMP THE OUTPUT BUFFER
0765      CALL TSEND
0766      RETURN
0767      END
0768      SUBROUTINE TWIND (MINX,MAXX,MINY,MAYY)
0769      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0770      1  TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0771      2  TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0772      3  KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0773      4  KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0774      5  KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0775      6  KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT
0776      C   DEFINE TERMINAL WINDOW IN TERMINAL COMMON AREA
0777      KMINSX=MINX
0778      KMAXSX=MAXX
0779      KMINSY=MINY
0780      KMAXSY=MAYY
0781      CALL RESCL
0782      RETURN
0783      END
0784      SUBROUTINE PCLIP (X,Y)
0785      COMMON LARY(20),TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
0786      1  TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,TRPAR1,
0787      2  TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(8),KPAD2,
0788      3  KBAUDR,KGNFLG,KGRAFL,KHOMY,KKMODE,KHORSZ,KVERSZ,KTBSZ,
0789      4  KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,
0790      5  KBEAMY,KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,
0791      6  KMAXSY,KEYCON,KINLFT,KOTLFT,KUNIT

```

```
0792      KGNFLG#0
0793      IF(X,LT,TMINVX) GO TO 10
0794      IF(X,GT,TMAXVX) GO TO 10
0795      IF(Y,LT,TMINVY) GO TO 10
0796      IF(Y,GT,TMAXVY) GO TO 20
0797      10 KGNFLG#1
0798      20 RETURN
0799      END
0800      ENDS
```