

AD-A241 211



DREDGING RESEARCH PROGRAM

TECHNICAL REPORT DRP-91-2

2

US Army Corps
of Engineers

SIMULATION OF TIME SEQUENCES OF WAVE
HEIGHT, PERIOD, AND DIRECTION

by

Leon E. Borgman

University of Wyoming
Laramie, Wyoming 32070

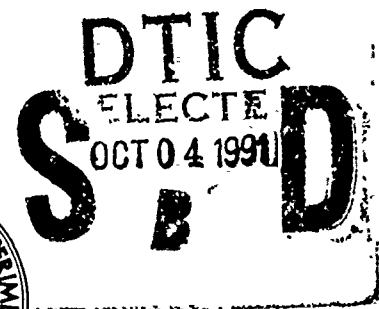
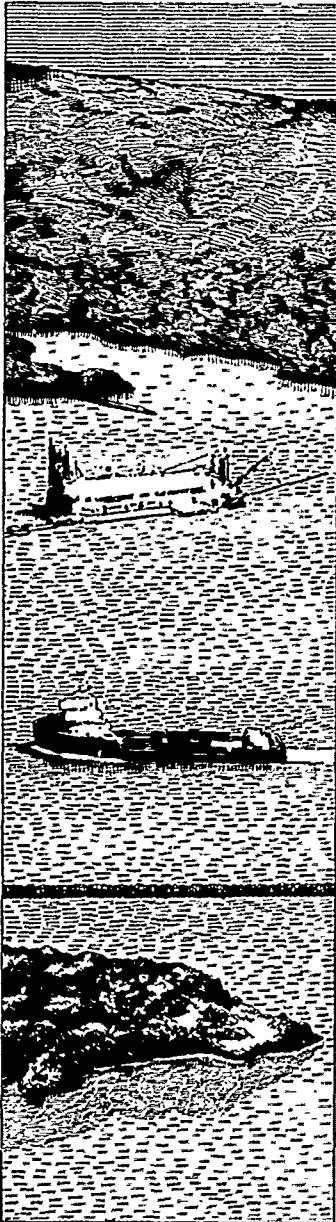
and

Norman W. Scheffner

Coastal Engineering Research Center

DEPARTMENT OF THE ARMY

Waterways Experiment Station, Corps of Engineers
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199



August 1991

Final Report

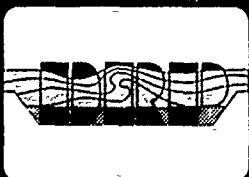
Approved For Public Release; Distribution Unlimited

91-12323



Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit 32466



07 70 2 000

The Dredging Research Program (DRP) is a seven-year program of the US Army Corps of Engineers. DRP research is managed in these five technical areas:

- Area 1 - Analysis of Dredged Material Placed in Open Waters
- Area 2 - Material Properties Related to Navigation and Dredging
- Area 3 - Dredge Plant Equipment and Systems Processes
- Area 4 - Vessel Positioning, Survey Controls, and Dredge Monitoring Systems
- Area 5 - Management of Dredging Projects

Destroy this report when no longer needed. Do not return
it to the originator.

This program is furnished by the Government and is accepted and used by the recipient with the express understanding that the United States Government makes no warranties, expressed or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof. The program belongs to the Government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent this program to anyone as other than a Government program.

The contents of this report are not to be used for
advertising, publication, or promotional purposes.
Citation of trade names does not constitute an
official endorsement or approval of the use of
such commercial products.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE August 1991	3. REPORT TYPE AND DATES COVERED Final Report		
4. TITLE AND SUBTITLE Simulation of Time Sequences of Wave Height, Period, and Direction		5. FUNDING NUMBERS DRP Work Unit 32466		
6. AUTHOR(S) Leon E. Borgman, Norman W. Scheffner				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) See reverse.		8. PERFORMING ORGANIZATION REPORT NUMBER Technical Report DRP-91-2		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Corps of Engineers Washington, DC 20314-1000		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report describes a procedure for generating simulated time sequences of wave height, period, and direction data corresponding to specific locations. The technique uses a finite length wave record to compute a matrix of coefficient multipliers that can be used to generate arbitrarily long time sequences of simulated wave data which preserve the primary statistical properties of the finite data set. The procedure was developed for simulating the 20-year Wave Information Study (WIS) hindcast database; however, it can be applied to any appropriate data sequence. Application of the methodology is demonstrated in this report through comparisons of simulated data to hindcast data corresponding to a Gulf of Mexico WIS station located offshore of the entrance to Mobile Bay, Alabama. Results show that the simulated series exhibits the primary statistical properties of the WIS data, including seasonality and wave sequencing.				
14. SUBJECT TERMS Statistics of waves Waves			15. NUMBER OF PAGES 222	
Wave Information Study Wave simulation			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) (Continued):

University of Wyoming
Laramie, WY 32070

USAE Waterways Experiment Station
Coastal Engineering Research Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

PREFACE

The study described herein was authorized and funded under Work Unit 32466, "Numerical Simulation Techniques for Evaluating Long-Term Fate and Stability of Dredged Material Disposed in Open Water," of Technical Area 1, Analysis of Dredged Material Placed in Open Water, of the Dredging Research Program (DRP), sponsored by Headquarters, US Army Corps of Engineers (HQUSACE). The HQUSACE Technical Monitors for the DRP were Messrs. Glenn Drummond, Rixie Hardy, John Perez, M. K. Miles, Don Ponmer, and John Sanda. Mr. E. Clark McNair, Jr., is Program Manager of the DRP; Dr. Nicholas C. Kraus, CERC, is the Technical Manager of DRP Technical Area 1; and Dr. Norman W. Scheffner is the Principal Investigator of Work Unit 32466.

The simulation methodology was developed by Dr. Leon E. Borgman, and the final report was prepared by Drs. Scheffner and Borgman over the period of 17 June 1988 through 28 February 1990. The project was conducted under the direct supervision of Mr. Bruce A. Ebersole, Chief, Coastal Processes Branch, Research Division (RD), CERC, and Mr. H. Lee Butler, Chief, RD, and under the general supervision of Dr. James R. Houston, Chief, and Mr. Charles C. Calhoun, Jr., Assistant Chief, CERC.

COL Larry B. Fulton, EN, is Commander and Director of WES. Dr. Robert W. Whalin is Technical Director.

For further information on this report or on the Dredging Research Program, please contact Mr. E. Clark McNair, Jr., Program Manager, at (601) 634-2070.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

CONTENTS

	<u>Page</u>
PREFACE	1
LIST OF FIGURES	4
SUMMARY	5
PART I: INTRODUCTION.....	7
Background.....	7
Scope of Report.....	8
PART II: THEORY.....	9
Normal Scores Transformation.....	10
Program Overview.....	13
Piecewise, Month-by-Month, Frequency-Domain Simulation of Normal Score Time Series.....	14
Nonstationary Adjustment for Seasonality by Interpolation.....	22
PART III: USER INSTRUCTIONS.....	24
Introduction.....	24
Overall Structure.....	24
Input WIS Data.....	25
Preprocessor Parameters.....	26
Preprocessor Data Statements.....	26
Preprocessor Read Statements.....	27
Preprocessor Program Run.....	27
Parameters for Simulation.....	28
Starting Seed Number.....	28
Running a Simulation.....	28
Simulation Input and Output.....	28
Computational Details.....	29
Program Listings.....	30
PART IV: EXAMPLE APPLICATIONS.....	31
PART V: CONCLUSIONS.....	52
REFERENCES.....	54
APPENDIX A: SCREEN LISTING DEVELOPED DURING HPDPRE	A1
APPENDIX B: OUTPUT FROM HPDPRE.....	B1
APPENDIX C: SCREEN LISTING DEVELOPED DURING HPDSIM	C1
APPENDIX D: SAMPLE OUTPUT AND TABLE OUTPUT FROM SIMULATIONS BY HPDSIM.....	D1
APPENDIX E: PROGRAM LISTING FOR HPDPRE.FOR.....	E1
APPENDIX F: PROGRAM LISTING FOR HPDSIM.FOR.....	F1
APPENDIX G: A SELECTED YEAR OF WAVE INFORMATION STUDY DATA.....	G1
APPENDIX H: A SIMULATED YEAR OF DATA IMITATING THE STATISTICS OF THE YEAR IN APPENDIX G.....	H1

APPENDIX I: COMPARISONS OF THE PROBABILITY DENSITIES FOR THE ACTUAL WAVE INFORMATION STUDY DATA IN APPENDIX G AND THE SIMULATIONS IN APPENDIX H.....	11
--	----

LIST OF FIGURES

<u>No.</u>		<u>Page</u>
1	Normal scores mapping.....	11
2	Basic organization of sequence of computations.....	22
3	Location map for WIS Station 27.....	32
4	WIS wave height, period, and direction for January 1957.....	33
5	WIS wave height, period, and direction for June 1957.....	34
6	Simulated wave height, period, and direction for January 1957..	36
7	Simulated wave height, period, and direction for June 1957.....	37
8	Simulated wave height, period, and direction for 1957.....	39
9	WIS wave height, period, and direction for 1957.....	40
10	Probability histogram comparison for January 1957 data.....	41
11	Probability histogram comparison for June 1957 data.....	43
12	WIS wave direction histogram plots for June 1956, 1961, and 1966.....	44
13	Three simulated wave direction histogram plots for June 1957...	45
14	WIS wave direction histogram for all Junes 1956 - 1975.....	46
15	Probability histogram comparison for 1957 data.....	47
16	Location map for WIS Station 147.....	48
17	Simulated Station 147 wave height, period, and direction for 1956.....	49
18	WIS Station 147 wave height, period, and direction for 1956....	50
19	Probability histogram comparison of 1956 data for Station 147..	51
G1	WIS Station #47 wave data	G3
H1	Simulated wave data	H3
I1	Probability histogram comparisons	I3

SUMMARY

The program presented herein was developed by the Dredging Research Program (DRP) specifically to provide a means for simulating the 20-year hindcast database of the Wave Information Study (WIS) for use in generating a comprehensive database of wave field boundary conditions. The intent of the DRP is to make the intercorrelation matrix available for all Phase II WIS stations such that personal computers (PC's) can be used to generate and analyze realistic wave field boundary conditions for investigating the long-term fate and stability of dredged material placement sites. The database is, however, applicable to any coastal design application requiring site-specific wave field time series. The general methodology can also be applied to any time and direction correlated data triplet of any duration. Since the program is based on actual data, the generated time series will not contain values or trends that are not contained in the original data. Therefore, this methodology should not be used to investigate extreme conditions, but as an alternative to prototype data.

This technical report describes the theory and implementation of a numerical procedure for simulating arbitrarily long time sequences of wave height, period, and direction that are statistically similar to those of an existing finite length time series. The procedure is based on the calculation of the statistical parameters describing the intercorrelations among the wave field variables of the existing data. Once the intercorrelation matrix has been computed, the original data are no longer required in the wave generation procedure. The matrix is used to compute single or multiple simulated time series that reflect the primary statistical properties of the entire original database, including seasonality and wave sequencing. Since the matrix requires less computer storage than the input data, the simulation technique is well suited to the limited storage capacity of a PC. The overall simulation methodology is analagous to the calculation of harmonic constituents for a tidal series that can be used to generate accurate long-term tidal reproductions.

Numerous examples of simulated waves corresponding to a Phase II WIS data station located offshore of Mobile, AL, are presented. Included are comparisons of probability histograms for both the simulated time series and for a selected year of the WIS hindcast database to demonstrate the capability of the model to simulate the hindcast data. The report contains both program

listings and example input and output listings as a guide to program use and implementation.

The statistical procedures described in this report represent an improved methodology for defining time series boundary conditions for coastal design or analysis applications. The approach provides an alternative to the use of discrete or repeated historical events for design conditions and allows the user the ability either to generate long sequences of data or to analyze the variability of the existing data.

SIMULATION OF TIME SEQUENCES OF WAVE HEIGHT,
PERIOD, AND DIRECTION

PART I: INTRODUCTION

Background

1. The goal of the "Numerical Simulation Techniques for Evaluating Long-Term Fate and Stability of Dredged Material Disposed in Open Water" work unit is to provide a simulation technique for determining how a specified dredged material mound behaves over time. The methodology is intended for site-designation investigations and is based on long-term sediment transport simulations using local hydrodynamic boundary condition input data. The intended use of the program is to provide a systematic and quantifiable approach to analyzing disposal site stability based on the local environmental conditions. If material is eroded from a mound and transported away from the designated disposal site, the site is classified as dispersive; otherwise, it is nondispersive. For locations predominated by strong wave and current regimes, sediment transport calculations based on average wave and current data may easily show the site to be dispersive; however, if the local environmental conditions are not severe, it may take months or years before significant amounts of sediment are transported from the disposal site. The ability to identify these long-term dispersive sites is especially important since the eroded material could be transported into environmentally sensitive areas. These long-term dispersion investigations cannot be accurately made without knowledge of the local wave climate and current field at the site.

2. The approach selected for disposal site analysis is a numerically based sediment transport prediction scheme using long-term local boundary conditions as input. These conditions represent those forcings that entrain and transport sediment, including waves, tides, and storms. In many cases, these data are not available or are incomplete or too short in duration for long-term predictions. To satisfy the ultimate goals of this work unit, the intermediate goal of simulating realistic boundary conditions for specific locations was developed. This report addresses the generation of arbitrarily long time sequences of ocean wave data corresponding to specific locations. Subsequent reports will address additional components of the current field.

3. Unlike astronomical tides that can be determined with great accuracy, waves cannot be precisely predicted. A viable approach for generating realistic wave field boundary conditions is to develop a wave simulation procedure that generates waves statistically similar to those known to occur at the site, i.e., preserving seasonality, directionality, distribution, sequencing, etc. This method requires an existing database from which the statistical parameters describing the intercorrelations of wave field parameters can be computed. This procedure is somewhat analogous to the computation of harmonic constituents for a tidal record. The 20-year Wave Information Study (WIS) database was selected as the source of wave parameter statistics, from which a matrix of multipliers is obtained that can then be used to generate time series of wave data which emulate the WIS 20-year hindcast data.

4. This statistical approach will enable the user to generate data series of any length, for example, 31 days of January or several years, for any WIS station. Although the procedure is based on the WIS database, it is fully applicable to any database. The only limitation of alternate data sources is that they must be of sufficient length to resolve significant time scale trends. For example, multiyear trends cannot be reduced from less than a year of data. The advantage of the procedure is that a finite length of data can be used to generate an infinitely long data series which is statistically similar to the original record but exhibits normal variations about its mean.

Scope of Report

5. This report is divided into four parts. Following the Introduction, Part II presents the theory of the simulation methodology used in the generation of simulated wave sequences. Part III briefly describes the structure of the simulation program and the input parameters necessary for using the program. Part IV presents example applications of the program to several WIS stations and includes the generation of wave field time series and the statistical comparison of results. Conclusions are presented in Part V. Output and program listings are presented in Appendices A through G to provide confirmation that the code is executing correctly and to allow the user to alter or modify the existing code. Finally, time series of WIS data, simulated WIS data, and the statistical comparisons generated in Part III are included in Appendix H.

PART II: THEORY

6. Two criteria were required of the technique selected for simulating time sequences of wave data based on finite length wave records. The synthetic record has to be realistic with respect to the statistical properties of the WIS database, and the methodology has to be computationally feasible with respect to both computer memory and computational speed. Five procedures were investigated as potential methodologies for satisfying these requirements: (a) multivariate autoregressive techniques; (b) nonstationary multivariate simulations; (c) piecewise, month-by-month, multivariate, stationary computations; (d) a conditional simulation version of type (c); and (e) a hybrid combination of the separate months in type (c).

7. Every simulation of random phenomena is artificial; therefore, differences between the simulated record and the actual data will exist. The adopted mathematical procedure must be carefully selected to yield a synthetic data series that preserves those statistical features of the WIS database deemed most important. It is therefore necessary to explicitly identify those features that should be preserved as an aid in choosing between the inevitable compromises involved in the design of the simulation algorithm.

8. The following statistical properties, listed in their order of importance, were judged to be important characteristics that should be preserved in a simulation of the WIS data:

- a. The univariate probability law for the three wave parameters of height, period, and direction.
- b. The temporal and spatial correlations within and between the wave parameters.
- c. Nonstationarity such as seasonal variations.

9. The piecewise, month-by-month, multivariate, stationary simulation approach was selected as the best means of generating realistic time sequences of wave data. This technique seeks to preserve the univariate probability laws and the first- and second-order moment properties that describe the intercorrelations of the data sequences. Higher order univariate moments are maintained to some extent by enforcing the univariate probability laws. Seasonal or nonstationary changes are imposed by simulating each month separately based on the WIS hindcast database for that month and then using a square-root interpolation scheme to force a smooth transition in the time series and intercorrelations from month to month. The statistical properties that may not

be preserved are the third and higher order moments between the variables and between different time and space elements in the data.

10. An empirical transformation of the WIS data sequences to a multivariate normal representation is used to maintain the first-order multivariate and higher order univariate moments of the database. The approach was selected because multivariate normal sequences of data can be conveniently and rapidly generated and returned to the correct univariate distribution through inverse empirical transformations. This technique, referred to as the normal scores transformation, forms the basis of the simulation approach and will be described in the following sections.

Normal Scores Transformation

11. A fundamental representation of the probability law for a random variable is the distribution function, defined as

$$F_X(x) = Pr [X \leq x] \quad (1)$$

where $Pr[]$ represents the probability that the random variable X is less than or equal to some value x . A distribution function can be estimated from an existing data set of the form

$$x_1, x_2, x_3, \dots, x_n \quad (2)$$

by the following procedure. The n values of x are first ranked in order of increasing size such that

$$x_{(1)} \leq x_{(2)} \leq x_{(3)} \leq \dots \leq x_{(n)} \quad (3)$$

where the parentheses surrounding the subscript indicate that the data have been rank-ordered. The value $x_{(1)}$ is the smallest in the series, and $x_{(n)}$ represents the largest. Let r denote the rank of the value $x_{(r)}$ such that rank 1 is the smallest and rank $r = n$ is the largest. An empirical estimate of $F_X[x_{(r)}]$, denoted by $\hat{F}_X[x_{(r)}]$, is given by Gumbel (1954) (see also Borgman and Resio 1982, p 446)

$$\hat{F}_x(x_r) = \frac{r}{(n+1)} \quad (4)$$

for $[x_{(r)}, r = 1, 2, 3, \dots, n]$. This form of estimate allows for future values of x to be less than the smallest observed datum $x_{(1)}$ with probability of $1/(n+1)$ and to be larger than the largest datum $x_{(n)}$ also with probability $1/(n+1)$.

12. The standard normal distribution function is defined as

$$F_z(z) = \int_{-\infty}^z \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy \quad (5)$$

The normal scores transformation is a mapping of the variable x in Equation 4 to the variable z in Equation 5 by equating the two distribution functions

$$\hat{F}_x(x) = F_z(z) \quad (6)$$

If $x = x_{(r)}$ is a random variable corresponding to rank r , the normal scores mapping represents solving for the variable z corresponding to the estimated distribution function of known value, according to the equation

$$F_z(z) = \int_{-\infty}^z \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy = \frac{r}{n+1} \quad (7)$$

An example of the mapping is shown in Figure 1 for the random variables $x_1 = 50$, $x_2 = 3$, and $x_3 = 10$. The ranked variables are $x_{(1)} = x_2 = 3$, $x_{(2)} = x_3 = 10$, and $x_{(3)} = x_1 = 50$. These variables may be taken as representing the random variable of wave height, period, or direction. The approximate distribution function for the first ranked value is $\hat{F}_x[x_{(1)}] = 1/(3+1) = 0.25$. The normal scores transformation represents a solution for z corresponding to a selected value of $F_z(z)$. For the example at $r = 1$, the value of z corresponding to $F_z(z) = 0.25$ is computed.

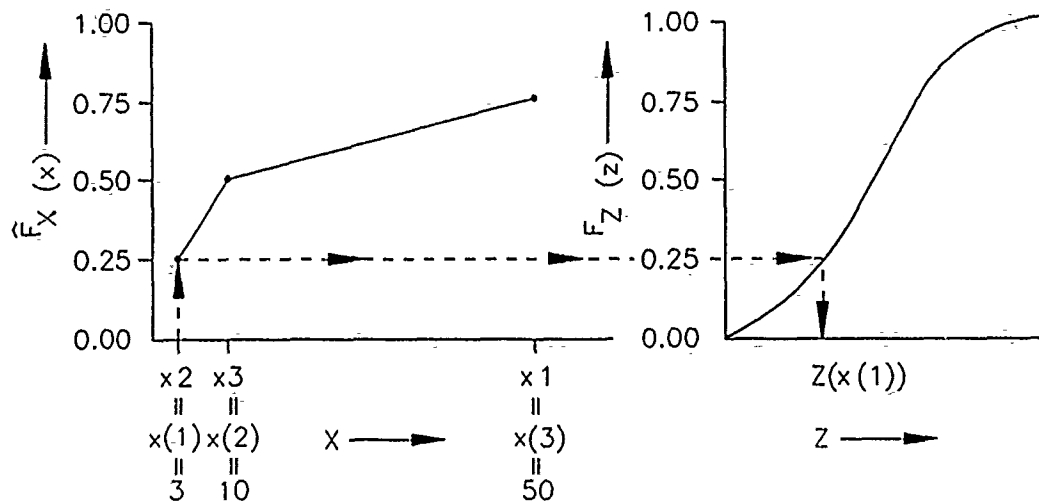


Figure 1. Normal scores mapping

The values of $\hat{F}_X(x)$ corresponding to all possible values of x are now used to compute corresponding values of z . Interpolation of the $\hat{F}_X(x)$ curve is required for $x_{(r)} < x < x_{(r+1)}$, and "tail" function curves are required to define the $\hat{F}_X(x)$ relationship for $x < x_{(1)}$ and $x > x_{(r)}$. The resulting values of z represent the normal scores transformation of x . Since the values of z were determined from the estimated distribution function, an inverse normal scores transform can be used to compute simulated random values of x based on randomly selected values of z .

13. The inverse normal scores transform involves solving for p from known z in the following equation

$$p = \int_{-\infty}^z \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy \quad (8)$$

With known p , the value of x can be interpolated from

$$\hat{F}_X(x) = p \quad (9)$$

If $p < 1/(n+1)$ or $p > n/(n+1)$, the assigned "tail" functions discussed above are used. Accurate, well-documented computer algorithms are available to solve for z given p , or p given z (Zelen and Severo 1965, Equations 26.2.23 and 26.2.17). The basic emphasis in the subsequent exposition will be on the

mechanics of simulating the wave field variables $[(H_i, T_i, \theta_i), i = 0, 1, 2, \dots, N-1]$,

where

H_i = significant wave height

T_i = spectral peak period

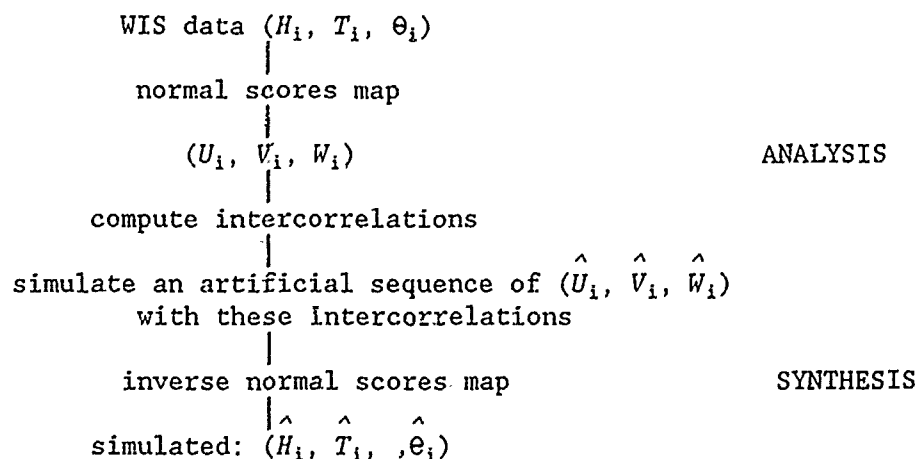
θ_i = wave direction

N is the number of 3-hr segments in the time interval being simulated. Each of these variables are ordered, and the normal scores transformations can be individually applied to each of the variable triple to yield an equivalent three-component sequence of multivariate normal vectors. Let U_i , V_i , and W_i represent these vectors such that

$$(H_i, T_i, \theta_i) \rightarrow \text{normal scores} \rightarrow (U_i, V_i, W_i) \\ \text{mapping}$$

Program Overview

14. The simulation problem becomes that of producing an artificial sequence of data $[(\hat{U}_i, \hat{V}_i, \hat{W}_i), i = 1, 2, 3, \dots, N]$ that has the proper intercorrelations embodied in the normal scores transformed data $[(U_i, V_i, W_i), i = 1, 2, 3, \dots, N]$. Once an acceptable simulation is produced by any means, the inverse normal scores transformation is used to simulate an artificial sequence of wave data $(\hat{H}_i, \hat{T}_i, \hat{\theta}_i)$ that statistically resembles the WIS database variables (H_i, T_i, θ_i) . A diagram of the process is thus



15. How does this procedure match the evaluation criteria? For the particular month selected, the simulated data $(\hat{H}_i, \hat{T}_i, \hat{\theta}_i)$:

- a. Exactly preserve the univariate empirical distribution of the WIS database (H_i, T_i, θ_i) with any bimodality, skewness, etc., present in the data.
- b. Preserve the intercorrelations of the mapped variables (U_i, V_i, W_i) . However, they only approximately preserve the intercorrelations between the original variables (H_i, T_i, θ_i) . The degree of approximation will be evaluated later with real WIS data.
- c. Preserve all univariate moments embodied in the empirical distribution for each variable (H_i, T_i, θ_i) through the inverse normal scores transformation.
- d. Do not necessarily preserve moments higher than second order for relations between variables and from one space-time location to another, although the relations may not be too bad. These will have whatever values are created through the multivariate normal simulation and the inverse normal scores transformation.

16. These general properties match the goals and priorities previously selected fairly well. The following sections will expand on the selected approach for generating WIS database simulations.

Piecewise, Month-by-Month, Frequency-Domain Simulation of Normal Score Time Series

17. It is reasonable to assume that the WIS data are stationary within each month such that all months of January are statistically similar, as are months of February, March, etc. This assumption allows much more stable estimates of the covariances, which describe the intercorrelations between the parameters of a particular data set, since they can be computed over all days of the month and for all 20 years. The covariance functions are written in the following form for the variable U (where all variables have been normed to have a zero mean):

$$\begin{aligned} C_{UU}(k) &= \sum_i \sum_j U_{ij} U_{i+k,j} / \text{NUM} \\ C_{UV}(k) &= \sum_i \sum_j U_{ij} V_{i+k,j} / \text{NUM} \\ C_{UW}(k) &= \sum_i \sum_j U_{ij} W_{i+k,j} / \text{NUM} \end{aligned} \quad (10)$$

These functions describe the statistical relationship between the normal score WIS data transform U and the parameters V and W . Here j is summed from 1 to 20, the indices i and $i+k$ are summed over all observations in the month, and

NUM refers to the number of products. Similar representations for the variables V and W define the following matrix of covariance functions

$$C = \begin{bmatrix} C_{UU}(k) & C_{UV}(k) & C_{UW}(k) \\ C_{VU}(k) & C_{VV}(k) & C_{VW}(k) \\ C_{WU}(k) & C_{WV}(k) & C_{WW}(k) \end{bmatrix} \quad (11)$$

for $k = 1, 2, 3, \dots$. Generally, the covariance function is essentially zero for k values greater than some k_0 . The value of k_0 is called the maximum lag of importance, and k_0 multiplied by the time interval between data points is called the range or zone of influence. It is anticipated that the range of the WIS data will not exceed 2 weeks and will probably be more like 1 week.

18. An algorithm for the simulation of frequency domain multivariate stationary Gaussian sequences has been developed by Borgman (1982) and makes use of the Fast Fourier Transform (FFT) algorithm. The general approach follows. For $m = 0, 1, 2, \dots, N-1$, the discrete Fourier coefficients can be written in terms of the generalized vector $\{U_n, V_n, W_n\}$, $n = 0, 1, 2, \dots, N-1$ as

$$\begin{aligned} A_U(m) &= \frac{1}{N} \sum_{n=0}^{N-1} U_n \exp\left(\frac{-i2\pi mn}{N}\right) \\ A_V(m) &= \frac{1}{N} \sum_{n=0}^{N-1} V_n \exp\left(\frac{-i2\pi mn}{N}\right) \\ A_W(m) &= \frac{1}{N} \sum_{n=0}^{N-1} W_n \exp\left(\frac{-i2\pi mn}{N}\right) \end{aligned} \quad (12)$$

19. Each of these frequency domain coefficients are complex-valued functions of m . The exact inverse of these relations is:

$$\begin{aligned}
U_n &= \sum_{m=0}^{N-1} A_U(m) \exp\left(\frac{i2\pi mn}{N}\right) \\
V_n &= \sum_{m=0}^{N-1} A_V(m) \exp\left(\frac{i2\pi mn}{N}\right) \\
W_n &= \sum_{m=0}^{N-1} A_W(m) \exp\left(\frac{i2\pi mn}{N}\right)
\end{aligned} \tag{13}$$

This inverse yields values of the generalized vector as a function of the discrete Fourier coefficients. The frequency domain simulation approach consists of a random number generation of Fourier coefficients $A_U(m)$, $A_V(m)$, and $A_W(m)$ that have the statistical intercorrelations of the WIS database shown in matrix 11. With the appropriately valued Fourier coefficients, the inverse FFT algorithm can be used to generate a simulated sequence of the generalized parameters (U_n, V_n, W_n) . Since the parameters contain the intercorrelations of the transformed WIS data by way of the Fourier coefficients, the generalized parameters can now be considered a simulated sequence of the normal score transformed WIS data.

20. In order for the vectors $[U_n, V_n, W_n]$, $n = 0, 1, 2, \dots, N-1$ to be stationary and multivariate normal with zero mean and be described by the covariance function matrix previously defined, then the Fourier coefficients are required to have the following statistical properties:

- a. The real and imaginary parts of $A_U(m)$, $A_V(m)$, and $A_W(m)$ are jointly multivariate normal.
- b. For $0 \leq m_1 < m_2 \leq N/2$, the vector $[ReA_U(m_1), ImA_U(m_1), ReA_V(m_1), ImA_V(m_1), ReA_W(m_1), ImA_W(m_1)]$ is statistically independent of the vector $[ReA_U(m_2), ImA_U(m_2), ReA_V(m_2), ImA_V(m_2), ReA_W(m_2), ImA_W(m_2)]$, where Re indicates the "real part of" and Im indicates the "imaginary part of" in each respective vector.
- c. The expected value of the real and imaginary parts of all coefficients is zero.
- d. The covariance matrix for the coefficients at a given m -value is related to the spectra and cross-spectra of the process defined by

$$\begin{aligned}
S_{UU}(m) &= Dt \sum_{k=0}^{N-1} C_{UU}(k) \exp\left(\frac{-i2\pi mk}{N}\right) \\
S_{VV}(m) &= Dt \sum_{k=0}^{N-1} C_{VV}(k) \exp\left(\frac{-i2\pi mk}{N}\right) \\
S_{WW}(m) &= Dt \sum_{k=0}^{N-1} C_{WW}(k) \exp\left(\frac{-i2\pi mk}{N}\right) \\
S_{UV}(m) &= Dt \sum_{k=0}^{N-1} C_{UV}(k) \exp\left(\frac{-i2\pi mk}{N}\right) \\
S_{UW}(m) &= Dt \sum_{k=0}^{N-1} C_{UW}(k) \exp\left(\frac{-i2\pi mk}{N}\right) \\
S_{VW}(m) &= Dt \sum_{k=0}^{N-1} C_{VW}(k) \exp\left(\frac{-i2\pi mk}{N}\right)
\end{aligned} \tag{14}$$

In the above formulas, Dt is the time increment between succeeding values of the covariance functions. Furthermore, the estimates of spectra need to be made only for $0 \leq m \leq N/2$ since the following symmetries are always present for real-valued U_n , V_n , and W_n and can be used to define the values for $m > N/2$:

$$\begin{aligned}
C_{UU}(N-k) &= C_{UU}(k) \\
C_{VV}(N-k) &= C_{VV}(k) \\
C_{WW}(N-k) &= C_{WW}(k) \\
C_{UV}(N-k) &= C_{VU}(k) \\
C_{UW}(N-k) &= C_{WU}(k) \\
C_{VW}(N-k) &= C_{WV}(k)
\end{aligned} \tag{15}$$

It is also assumed that the range is less than $N/2$ in all cases. The spectral functions $S_{UU}(m)$, $S_{VV}(m)$, and $S_{WW}(m)$ are real valued. However, the cross-spectral functions $S_{UV}(m)$, $S_{UW}(m)$, and $S_{VW}(m)$ are in general complex-valued.

e. The covariance matrix of the coefficients at a given m are, for $m = 0$ or $m = N/2$ where always $\text{Im}A_U(m) = \text{Im}A_V(m) = \text{Im}A_W(m) = 0$,

$$\text{COV} \begin{bmatrix} \text{Re}A_U(m) \\ U \\ \text{Re}A_V(m) \\ \text{Re}A_W(m) \end{bmatrix} = \begin{bmatrix} S(m) & \text{Re}S(m) & \text{Re}S(m) \\ UU & UV & UW \\ \text{Re}S_{UV}(m) & S_{VV}(m) & \text{Re}S_{VW}(m) \\ \text{Re}S_{UW}(m) & \text{Re}S_{VW}(m) & S_{WW}(m) \end{bmatrix} Df \quad (16)$$

For $0 < m < N/2$, the covariances are given by

$$\text{Cov} \begin{bmatrix} \text{Re}A_U(m) \\ \text{Im}A_U(m) \\ \text{Re}A_V(m) \\ \text{Im}A_V(m) \\ \text{Re}A_W(m) \\ \text{Im}A_W(m) \end{bmatrix} = Df/2 \begin{bmatrix} S_{UU}(m) & 0 & \text{Re}S_{UV}(m) & -\text{Im}S_{UW}(m) & \text{Re}S_{UW}(m) & -\text{Im}S_{VW}(m) \\ 0 & S_{UU}(m) & \text{Im}S_{UV}(m) & \text{Re}S_{UV}(m) & \text{Im}S_{UW}(m) & \text{Re}S_{UW}(m) \\ \text{Re}S_{UV}(m) & \text{Im}S_{UV}(m) & S_{VV} & 0 & \text{Re}S_{VW}(m) & -\text{Im}S_{VW}(m) \\ -\text{Im}S_{UV}(m) & \text{Re}S_{UV}(m) & 0 & S_{VV}(m) & \text{Im}S_{VW}(m) & \text{Re}S_{VW}(m) \\ \text{Re}S_{UW}(m) & \text{Im}S_{UW}(m) & \text{Re}S_{VW}(m) & \text{Im}S_{VW}(m) & S_{WW}(m) & 0 \\ -\text{Im}S_{UW}(m) & \text{Re}S_{UW}(m) & -\text{Im}S_{VW}(m) & \text{Re}S_{VW}(m) & 0 & S_{WW}(m) \end{bmatrix}$$

where $Df = 1/(NDt)$.

f. For $N/2 < m \leq N - 1$, the Fourier coefficients are defined by

$$\begin{aligned} A_U(N - m) &= \overline{A_U(m)} \\ A_V(N - m) &= \overline{A_V(m)} \\ A_W(N - m) &= \overline{A_W(m)} \end{aligned} \quad (18)$$

where the overbar denotes the complex conjugate. A concise statement of these relations is given in the appendix to Borgman (in press).

21. The simulation of real and imaginary parts of $A_U(m)$, $A_V(m)$, and $A_W(m)$ is most efficiently achieved through an eigenvector-eigenvalue procedure, although there are other ways to proceed. The eigenvector procedure will be outlined here. At a given m , let $\underline{b}_j$ be the eigenvector associated with the eigenvalue λ_j . Then, if C is the relevant covariance matrix,

$$C\underline{b}_j = \lambda_j \underline{b}_j \quad (19)$$

for $j = 1, 2, 3$ (if $m = 0$ or $N/2$) or $j = 1, 2, 3, 4, 5, 6$ (if $0 < m < N/2$). Let B be the matrix whose j^{th} column is $\underline{b}_j$ and let L be the diagonal matrix

whose off-diagonal elements are zero and whose (j,j) element is λ_j . Then the eigenvector-eigenvalue relations can be summarized in a single equation as

$$CB = BL \quad (20)$$

Furthermore, from the properties of eigenvectors, the matrix B can be taken as orthogonal, or

$$B^T = B^{-1} \quad (21)$$

where the superscript "T" denotes the matrix transpose. Then

$$C = BLB^T \quad (22)$$

By properties of diagonal matrices,

$$L = L^{1/2} L^{1/2} \quad (23)$$

where

$$L^{1/2} = \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots) \quad (24)$$

Thus, C can be decomposed into

$$C = (BL^{1/2}) (BL^{1/2})^T \quad (25)$$

22. This form of the matrix C represents the covariance matrix of the Fourier coefficients. Simulation of the Fourier coefficients can be achieved with the equation

$$\hat{A} = (BL^{1/2}) Z^* \quad (26)$$

where Z^* is a vector of independent, standard normal random numbers (Borgman 1982).

23. The covariance matrix for $\hat{A}$ can be shown to be identical to that of the Fourier coefficients,

$$C = E [\hat{A}\hat{A}^T] = BLB^T = C \quad (27)$$

where $E[]$ denotes the expectation operator. Consequently, the parts of the corresponding Fourier coefficients can be simulated by the randomly selected vector $\hat{A}$.

24. The above procedure is applied for all m where the spectra have significantly nonzero values. This generally requires the computation for

$$0 \leq m \leq m_0 \ll N/2 \quad (28)$$

It is generally satisfactory to require that all the Fourier coefficients at $m = 0$ are zero. This is equivalent to forcing the corresponding $(\hat{U}, \hat{V}, \hat{W})$ sequences exactly to have an arithmetic mean equal to zero. Thus, it is usually quite sufficient to set

$$\begin{aligned} \hat{A}_u(0) &= \hat{A}_v(0) = \hat{A}_w(0) = 0 \\ \hat{A}_u(m) &= \hat{A}_v(m) = \hat{A}_w(m) = 0 \quad \text{for } m_0 < m \leq \frac{N}{2} \end{aligned} \quad (29)$$

and then to use the eigenvector simulation procedure only for

$$0 < m \leq m_0 \quad (30)$$

The FFT coefficients for $N/2 < m < N$ are obtained from the complex conjugate relations with $0 < m < N/2$. Finally, the (U_n, V_n, W_n) sequences are obtained by the FFT inverse transformation.

25. This procedure seems much more complex and involved than the straightforward time-domain convolution procedure; however, it is faster in computer calculations by several orders of magnitude. The basic reason for this difference in computational speed is the trade-off between the properties of a series in the time-domain and the frequency-domain. The stationary sequences in time have constant variances but are correlated in time. The Fourier coefficients for these same series have frequency-varying variance but are uncorrelated from frequency to frequency. Thus, correlation in time is traded for independence (for the circularly periodic Gaussian case) from frequency to frequency. A single large problem that is correlated in time is separated into many small independent problems in the frequency domain. Each frequency, for those frequencies making contributions to the variations, can be treated separately.

26. The resulting sequence $\{(\hat{U}, \hat{V}, \hat{W}); n = 0, 1, 2, \dots, N - 1\}$ has one minor flaw. It is circularly periodic, and the values at small n are correlated with the values of n near N . This circular periodicity is usually unimportant. If it has undesirable effects in the application, N can be made larger than needed, and then a "zone of influence" of values is deleted from the end of the simulation. For the current application to the simulation of a month of WIS data, there are $240 = 8 \times 30$ 3-hr intervals in a 30-day month. Each monthly simulation is performed with $N = 480$. The first 5 days (40 values) and last 5 days can be discarded as possibly being contaminated with circular periodicity. The square-root interpolation method discussed below is used to join the remaining 10-day overlap on either side of the month with the preceding and following monthly simulations. This procedure should destroy any possible correlation between the beginning and end of a simulated data sequence.

27. This procedure can be applied to each month of the year to generate many years of WIS sequences. One fault arises at the discontinuity from the end of 1 month and the beginning of the next. The approach adopted to eliminate this discontinuity is discussed in the following section.

Nonstationary Adjustments for Seasonality by Interpolation

28. Let $X(t)$ and $Y(t)$ be two stationary, mean zero, time series and define

$$Z(t) = \sqrt{a}X(t) + \sqrt{1-a} Y(t) \quad (31)$$

where $0 \leq a \leq 1$. Further, let $X(t)$ and $Y(t)$ be incoherent with each other (independent of each other at all frequencies), and let the covariance functions be

$$\begin{aligned} C_{xx}(\tau) &= \text{Cov. fn of } X(t) \\ C_{yy}(\tau) &= \text{Cov. fn of } Y(t) \end{aligned} \quad (32)$$

The covariance function of $Z(t)$ is then

$$\begin{aligned} C_{zz}(\tau) &= E[Z(t)Z(t+\tau)] \\ &= E\{[\sqrt{a}X(t) + \sqrt{1-a}Y(t)][\sqrt{a}X(t+\tau) + \sqrt{1-a}Y(t+\tau)]\} \end{aligned} \quad (33)$$

or

$$\begin{aligned} C_{zz}(\tau) &= aE[X(t)X(t+\tau)] + (1-a)E[Y(t)Y(t+\tau)] \\ &\quad + \sqrt{a(1-a)} E[X(t)Y(t+\tau)] + \sqrt{a(1-a)} E[Y(t)X(t+\tau)] \end{aligned} \quad (34)$$

Since the two series are incoherent with each other (that is, simulated separately with separate starting seed numbers), the last two terms are zero. Thus

$$C_{zz}(\tau) = aC_{xx}(\tau) + (1-a)C_{yy}(\tau) \quad (35)$$

The covariance function for $Z(t)$ is therefore a linear interpolation between the covariance functions of $X(t)$ and $Y(t)$. This property will be exploited in the following procedure for eliminating the discontinuity in the month-by-month stationary simulation of the WIS data. Let

$$[(\hat{U}_{nj}, \hat{V}_{nj}, \hat{W}_{nj}) \quad n = 1, 2, 3, \dots, 240] \quad (36)$$

be the simulation for a month of time intervals centered about the start of the j^{th} month, for $j = 1, 2, 3, \dots, 12$. Suppose these simulations are prepared independently of each other such that they are incoherent with each other. Define the simulation during the month j as

$$\begin{bmatrix} \hat{U}_{nj}^* \\ \hat{V}_{nj}^* \\ \hat{W}_{nj}^* \end{bmatrix} = \sqrt{240 - n} \begin{bmatrix} \hat{U}_{nj} \\ \hat{V}_{nj} \\ \hat{W}_{nj} \end{bmatrix} + \sqrt{n} \begin{bmatrix} \hat{U}_{n,j+1} \\ \hat{V}_{n,j+1} \\ \hat{W}_{n,j+1} \end{bmatrix} \quad (37)$$

29. This simulation will have an approximately nonstationary covariance function that will interpolate between the stationary covariance functions centered at the beginning of the month and the corresponding stationary covariance centered at the end of the month.

30. The above outlined simulation technique has been shown to adequately simulate wave data sequences that are statistically representative of the WIS database. In the following section, an example implementation of the program to a WIS station will be shown and will include statistical comparisons of WIS data and the simulated data. These comparisons demonstrate the general validity of the technique and the powerful implications of a methodology for simulating infinitely long, statistically correct, wave data series based on a finite length of data.

PART III: USER INSTRUCTIONS

Introduction

31. Extremely fast algorithms are used to analyze WIS data on wave height, period, and direction of travel to produce a preprocessed data summary. This data summary, together with a starting seed number, can be used with additional fast algorithms to generate synthetic sequences of wave height, period, and direction (H,T,D) that statistically resemble the WIS data in long-term single-variable probability law and intercorrelation structure. Values of (H,T,D) may be produced for as many years of artificial history as is desired. It is completely feasible to generate several centuries of data. Throughout the following section, standard FORTRAN notation will be used to specify variables and relations.

Overall Structure

32. The basic organization of the sequence of computations is shown in Figure 2. The WIS 20-year data sequence for a given site constitutes the primary input to the system. The flowchart shows rectangular boxes for computer programs.

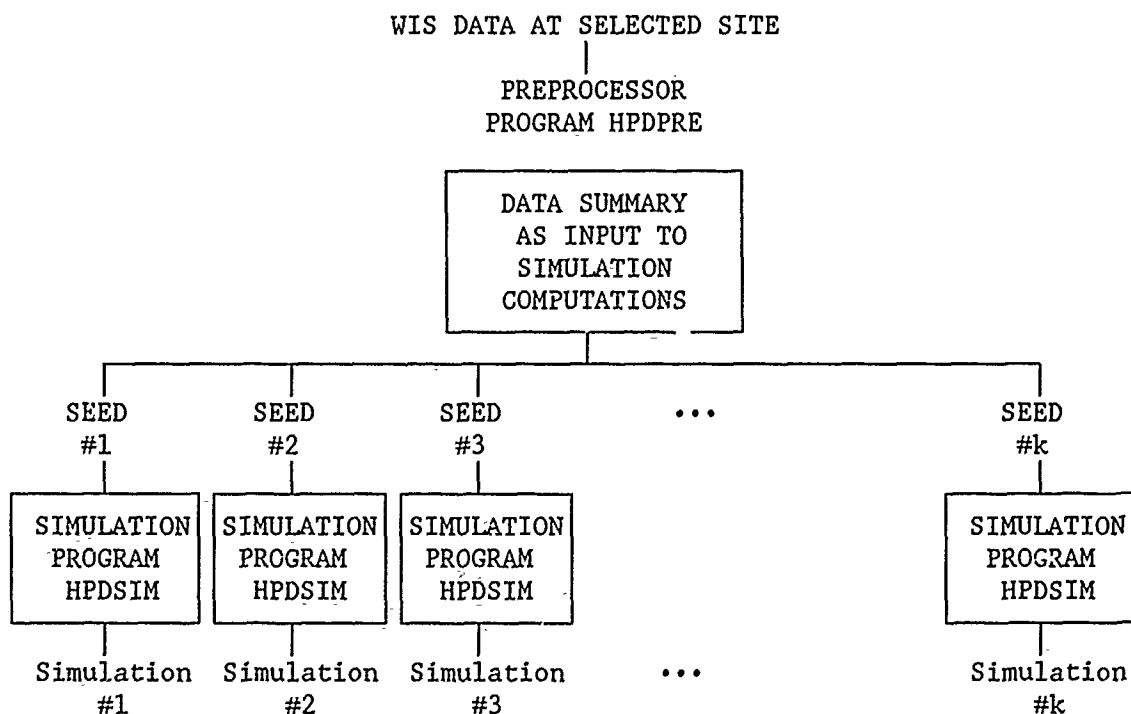


Figure 2. Basic organization of sequence of computations

33. The data summary produced by the preprocessor can be re-used many times, with new seed numbers each time, to obtain many alternate simulations. The program/subroutine package Height, Period, Direction, SIMulation (HPDSIM) contains the simulation software.

Input WIS Data

34. The WIS data tapes give the following variables every 3-hr for the 20-year period 1956-1975.

IY = 2-digit year integer (e.g., 56)
IM = 2-digit month integer (e.g., 01)
ID = 2-digit day integer (e.g., 12)
IH = 2-digit hour integer
ISTA =)
ILOC =) not used
JLOC =)
HS = significant wave height
TP = period at peak of spectral density
WDM = wave direction of movement
HSEA = sea wave height
TSEA = sea wave period
DSEA = sea wave direction
HSWL = swell wave height
TSWL = swell wave period
DSWL = swell wave direction
ALFA =)
VPAR =) not used
IVEL =)
IDIR =

35. These data are read with the FORTRAN format (2X, 4I2, 3I3, 3(F6.1, F6.1, F6.0), F5.3, F7.4, 2I4). The user selects which triple of (H,T,D) is to be used in the analysis and sets equations

H1 =)
T1 =) variable names selected for use
D1 =)

at three well-marked locations in the computer code main program HPDPRE. All code is in FORTRAN, so the user can easily set the lines of code defining H1,

T1, and D1 and then compile, link, and run the program to get the data summary.

Preprocessor Parameters

36. For the WIS data, as based on 20 years of hindcasts and for all 12 months, the parameters in the preprocessor software HPDPRE should be set to

NYR = number of years in database = 20

NNY = number of years to be analyzed = 20

NMO = number of months to be analyzed = 12

N = number of data triples (H,T,D) in an analysis unit of 1 month = 240 (If months have less than 30 days, the additional days are taken from the next month. If the month has more than 30 days, only the first 30 are used in the analysis.)

NSIM = number of data triples in a basic simulation interval (taken here as 2 months) = 480 (This larger set in simulation is used to allow an overlap with preceding and following months so that smooth transitions can be formed by various operations of linear interpolation or "ramping.")

NMAX = maximum value of N for dimensioning = 240 here.

NSMAX = maximum value of NSIM for dimensioning = 480 here.

NTM = $N * NNY = 240(20) = 4,800$

NTMAX = $NMAX * NNY = 4,800$

NTMX05 = an integer equal to or greater than $NTMAX/5 = 970$

MNO2P1 = $(NSMAX/2) + 1 = 241$

DT = time increment = 3.0 hr.

HSCALE = 1.0

) (allows for scaling H and T)

TSCALE = 1.0

Preprocessor Data Statements

37. Three data statements are introduced to specify the months to be analyzed, the frequency cutoff each month, and the years out of the 20 to be analyzed. Since all are being treated here, the data statements are

DATA MONTHS/12*1/

DATA CUTOFF/12*0.0833333/

DATA IYEARS/20*1/

This sets vector (MONTHS(J), J=1, 12) to have all components equal to 1, vector (CUTOFF(J), J=1, 12) to have all components equal to 0.0833333 cycles per hour, and vector (IYEARS(I), I=1, 20) to have all components equal to 1. If some months or years were to be excluded, those components of MONTHS(J) and YEARS(I) would be set to zero. Similarly the cutoff can be made different for each month to reflect seasonal changes by setting CUTOFF(J) to the cutoff frequency in cycles per hour of each month, J.

Preprocessor Read Statements

38. The wave height, period, and direction from the WIS data to be analyzed are selected by setting the FORTRAN code after the READ statements at three well-marked places in the main program HPDPRE. The code recognizes the variables H1, T1, and D1, so these are set equal to the variable names from the WIS data to be analyzed. Currently these are set to

H1 = HS

T1 = TP

D1 = WDM

Another possibility, not used here, is

H1 = HSEA

T1 = TSEA

D1 = DSEA

39. The same three equations should be set at the three designated locations in the code.

Preprocessor Program Run

40. The code is then compiled, linked, and run to produce a data file, HPDPRE.OUT, which serves as the input for the simulation runs with HPDSIM. Another output file, HPDPRE.DOC, provides convenient documentation for program checking and debugging. The screen printout is shown in Appendix A, and a section of the output file HPDPRE.OUT is given in Appendix B.

Parameters for Simulation

41. The parameters NSMAX, MNO2P1, and NMAX used in HPDSIM should not be smaller, but may be larger, than the corresponding parameters set in the preprocessor. The parameter NYR should be the same as in the preprocessor.

Finally, a new parameter sequence for the simulation is introduced as

IYB = starting year for simulation

IMB = starting month for simulation

IYE = ending year for simulation

IME = ending month for simulation

42. These parameters need not have any relation to the 20-year WIS database period. Thus, the simulation could run from (IYB = 1995, IMB = 1) to (IYE = 2094, IME = 12) to develop a century of synthetic wave data.

Starting Seed Number

43. The simulation is controlled and determined by a starting seed integer, ISEED0. This is set in the code as a FORTRAN equation immediately after the open statements. For example,

ISEED0 = 67676767

is shown in the present code. Every new simulation should use a new seed number. The number can be any integer in the interval (1, 2147395599).

Running a Simulation

44. After IYB, IMB, IYE, and IME are set as parameters and ISEED0 is assigned an initial value, the FORTRAN code is compiled, linked, and run. The console screen will show a sequence of messages that describe the current actions in the computer. A sample of the screen listing is given in Appendix C.

Simulation Input and Output

45. The basic input to the simulation program HPDSIM is the file HPDPRE.OUT produced by the preprocessor program HDPPRE. An example file is shown in Appendix B.

46. There are various possible output files. These can be blocked out if they are not wanted by searching for the WRITE (N) expression and "commenting" it out with a "C" in column 1. The N here is the device number for that output file in the open statement.

47. The basic two output files from Program HPDSIM are HPDSIM.OUT and HPDSIM.TAB, illustrated in Appendix D. HPDSIM.OUT is a serial listing of year, month, day, hour, wave height, wave period, and direction. One set of the seven values is printed on each line. The HPDSIM.TAB file gives the output of the simulation in a form that is neatly tabled in compact form with heading and row documentation. The other output files are specialty output useful under some conditions for debugging, preparing graphs, and so forth. The output file HPDSIM.MAT produces a file that can be used as input to a software analysis and graphics package. The file HPDSIM.DOC gives detailed intermediate documentation of the simulation process. File HPDSIM.RAM provides documentation of the "heap" storage and "ramping" process used in making smooth transitions from one month to the next. File HPDSIM.VRS provides documentation details on the "tail" interpolation used in back interpolating from the synthetic Gaussian processes to the empirical probability laws for the WIS H, T, and D values. Of these various files, only the HPDSIM.OUT and HPDSIM.TAB actually provide the simulated values for use in subsequent application.

Computational Details

48. The preprocessor operates on a large WIS tape with perhaps 5 megabytes (MB) of data. The program really requires something like a VAX computer or even a more powerful mainframe computer. The computations on a VAX computer for HPDPRE may take about 30 min or more, depending on whether the VAX is being used in a time-sharing mode. The output from HPDPRE can be broken into three parts and put on diskettes for a top-line personal computer (PC) (perhaps a 386-chip machine).

49. The simulation runs can be performed on a VAX or PC. With the VAX, the program simulates a month of data about every 15 sec, or a year of data every 3 min. A century of data could be developed in 300 min or 5 hr of VAX time. If such long runs are desired, some code adaption should be introduced into the HPDSIM program to dump a decade of data at a time to files. Also, the subsidiary output files HPDSIM.DOC, HPDSIM.MAT, HPDSIM.RAM, and HPDSIM.VRS

should all be canceled out. Otherwise, memory would be used up with unnecessary files. These files can be left in for short runs of up to 5 years without causing too much trouble.

50. Both the preprocessor HPDPRE and the simulator HPDSIM are optimized to work with frequency-domain methods, the fast FFT, and other "fast" algorithms. In particular, substantial effort has been expended in seeking every way to make the simulation program work rapidly since it is the program that will be run and re-run to obtain many simulated sets of wave conditions.

Program Listings

51. A code listing for the HPDPRE program is given in Appendix E. Similarly, the code listing for the HPDSIM program is printed in Appendix F. All code is in FORTRAN 77.

PART IV: EXAMPLE APPLICATIONS

52. The HPDPRE and HPDSIM programs were written for the specific application to the 20-year hindcast WIS database of wave height, period, and direction. The purpose was to provide arbitrarily long sequences of wave field boundary conditions for long-term numerical simulations of sediment transport without having to use historical data. The following examples demonstrate an application of the program to the WIS database. Applications to other data series will require changes to the internal program variables described in Part III.

53. Before comparing WIS time series to simulated wave time series, the intent of the simulation process is re-emphasized. Input parameters of the HPDSIM program include the specification of a starting and ending month and year. These simulated times represent arbitrary temporal delimiters such that the sequencing of months and years are maintained with the proper spacing of leap year. The program generates a time series corresponding to a specified starting seed and beginning month; however, the year of the simulation is arbitrary; it is assigned the user-defined input value. If two independent 1-year simulations are generated, both will be identical if the identical starting month and seed are specified. Therefore, the simulation of June 1957 is not intended to duplicate the June 1957 WIS time series. It is, however, expected to represent the statistical properties of a typical June and reflect the variability shown in all Junes contained in the 20-year hindcast. Different Junes of 1957 can be generated by specifying different random seed numbers (ISEEDO). This multiple generation of the same time period gives the user the capability of investigating the variability of fixed time periods (months, seasons, or years) without having to select historical data that may not be appropriate for the specific application. The WIS station selected for demonstrating the capability of the simulation package is the Phase II Gulf of Mexico Station 27, located just outside the entrance to Mobile Bay, Alabama, shown in Figure 3. The time series of wave height, period, and direction for Station 27 are shown for each month of 1957 in Appendix G. Two examples, January and June, are selected from Appendix G to illustrate the program output. The two selected time series of wave height, period, and direction are shown in Figures 4 and 5.

54. The simulation example used in the screen and output examples of Appendixes A and C generated a synthetic data series for September 1956

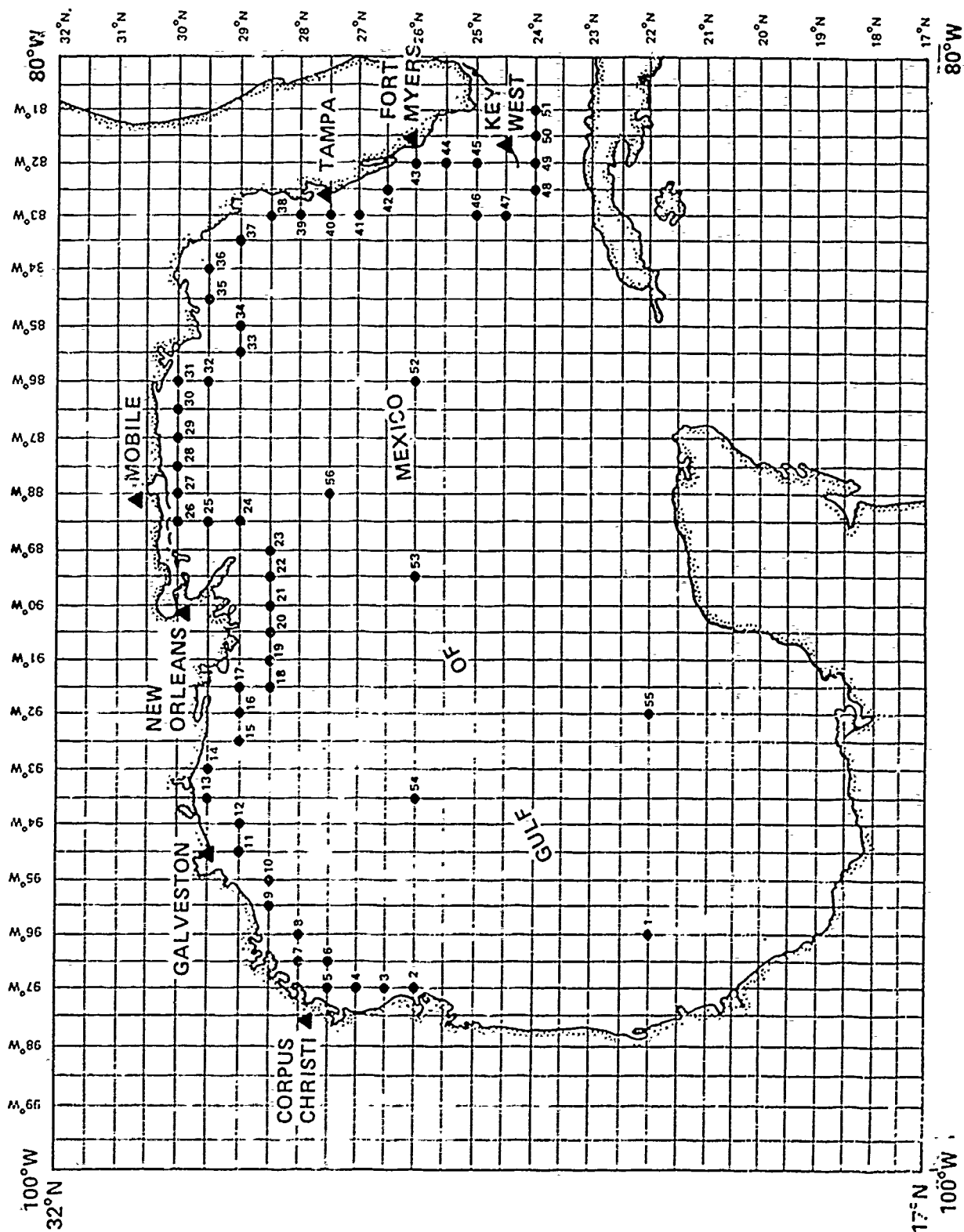


Figure 3. Location map for WIS Station 27

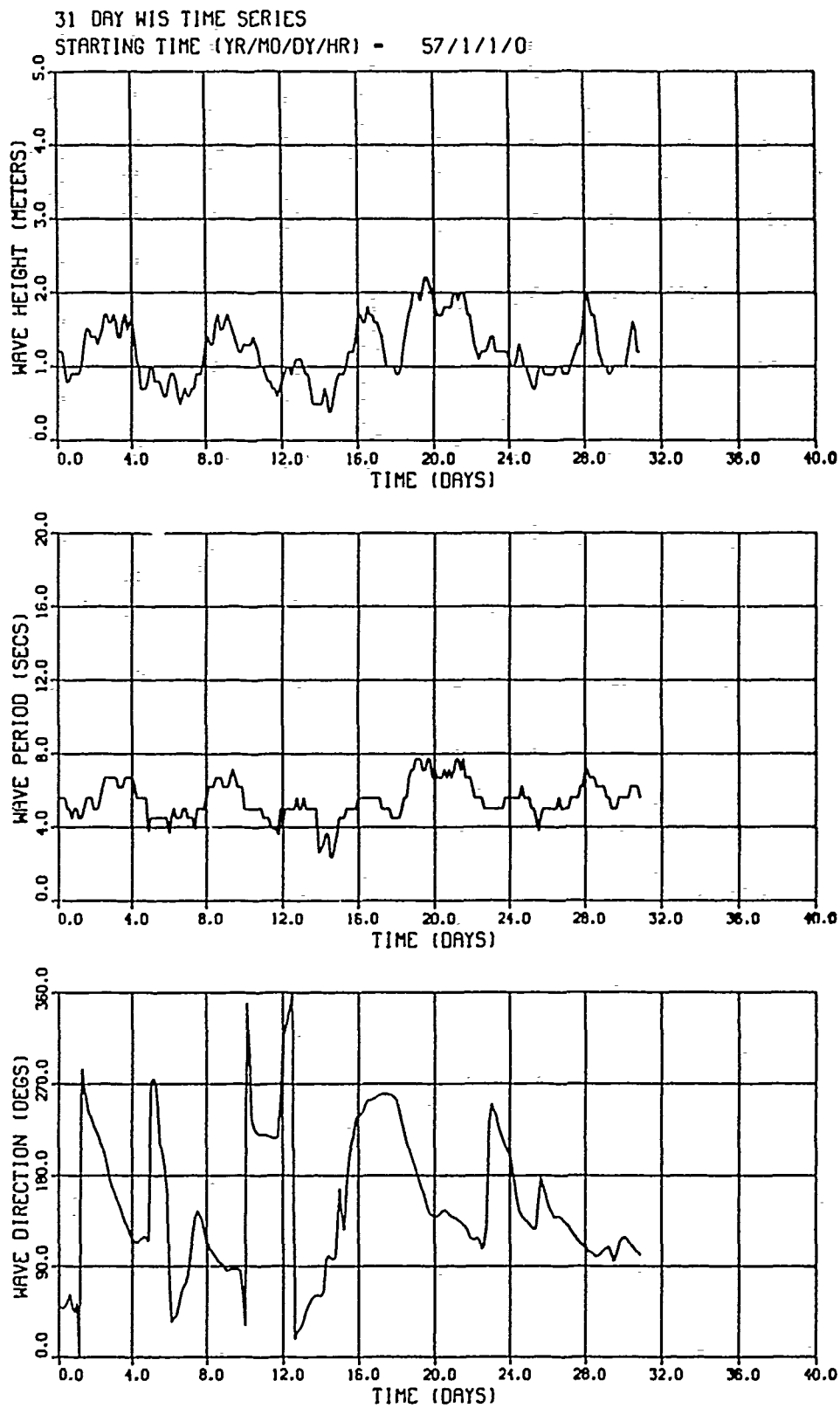


Figure 4. WIS wave height, period, and direction for January 1957

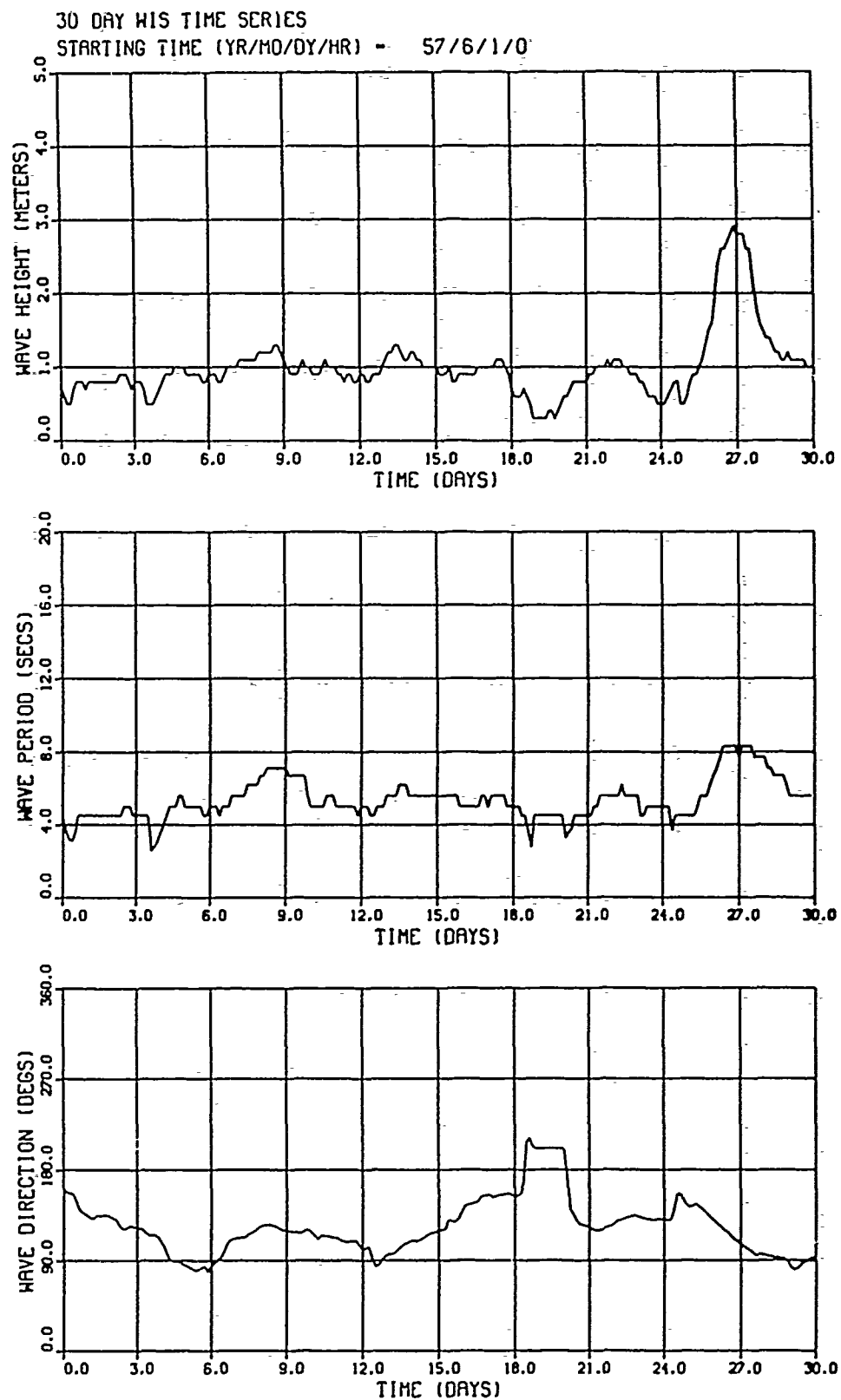


Figure 5. WIS wave height, period, and direction for June 1957

through June 1958, with a starting random seed of 67676767. The simulated time series for each month of the generated year, indexed as 1957 and contained in output file HPDSIM.OUT, are shown in Appendix H. Figures 6 and 7 represent the January and June simulated time series.

55. A comparison of Januarys shown in Figures 4 and 6 show similar patterns of both wave height and period. Both records show cyclic variations on the order of 4 to 6 days. Heights are both generally between 1.0 and 2.0 m with several outliers above and below. Periods show a similar pattern with the majority of values falling between 4.0 and 8.0 sec. Wave directions also exhibit similar trends, as will be subsequently demonstrated. Similar qualitative comparisons can also be seen between Figures 5 and 7, although an extreme event is shown in the WIS data around day 27 and the wave directions appear bimodal for the simulations and not in the WIS data.

56. The computed values of maximum, minimum, average, and standard deviation (SD) for wave heights and periods for each of the compared months shown in Appendixes G and H are given in Tables 1 and 2. Similar figures for directional data are meaningless because of the circularity of the data; however, relevant comparisons will be presented. The comparisons shown below indicate that the simulated and WIS data are similar with respect to both magnitudes and temporal trends, although expected differences between the two can be seen. An example of data variability will be presented subsequently.

57. A summary plot of the entire year of simulated and WIS time series for height, period and direction are shown in Figures 7 and 8. As shown in the comparisons of January and June, the general features of the WIS data are evident in the simulated data.

58. A more descriptive comparison of the data can be seen through percent probability histograms that visually indicate the percentage of observation in the data series that fall within specified limits, for example, the percentage of wave height observations between 1.0 and 1.1 m. Selected bands for the histograms were 0.1-m increments for wave heights, 0.5 sec for period, and 10.0 deg for direction. These can be computed for both the simulated data series and the WIS data series and overlaid to provide a visual comparison of both data sets. Histogram plots for each month of 1957 are shown in Appendix I. The periods of January and June are reproduced below in Figures 9 and 10. The comparison shown in Figure 9 discloses that approximately 30 percent of the January data for both the WIS and simulated time series fall within a period band of 5.0 to 5.5 sec. The plots provide a comparison of the

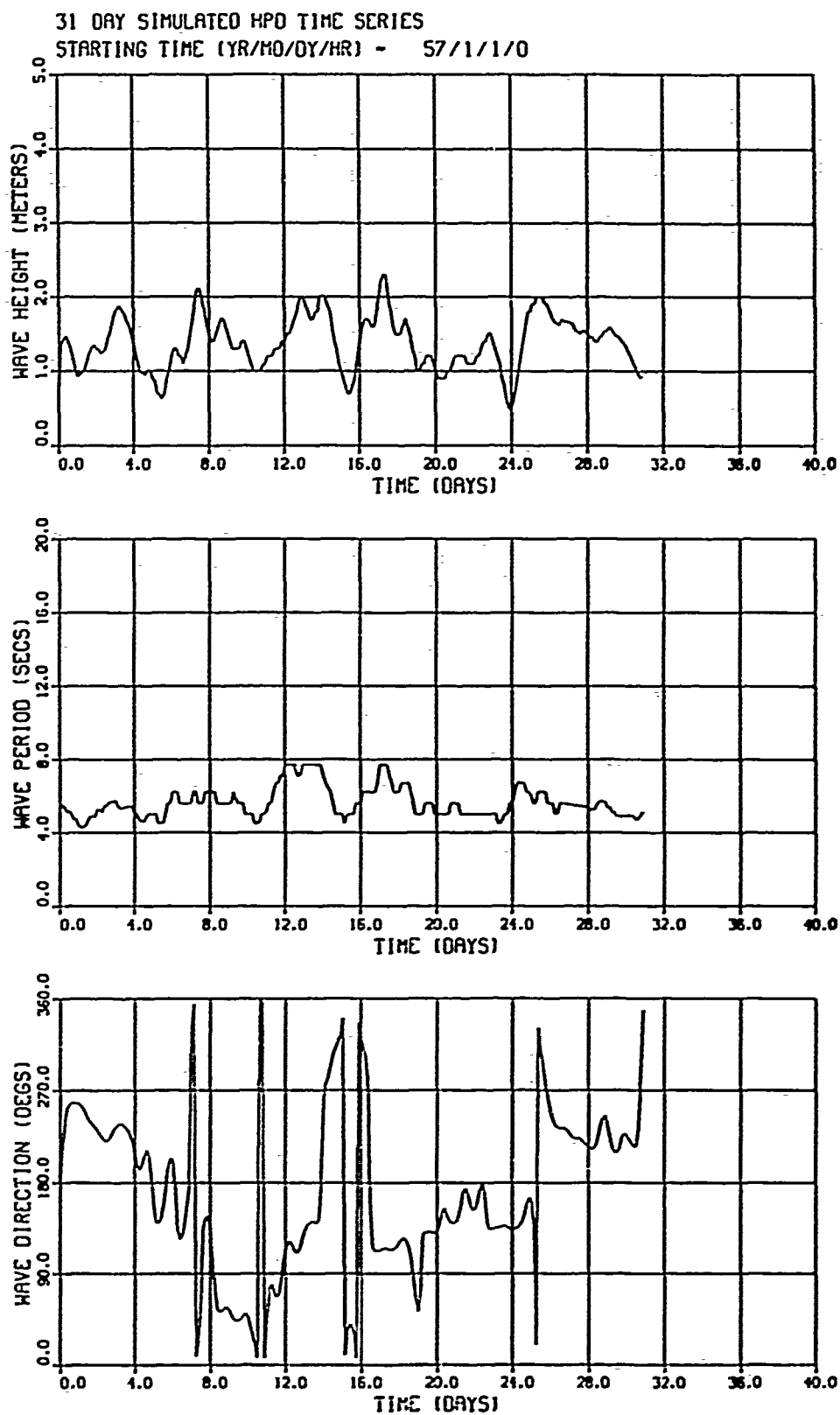


Figure 6. Simulated wave height, period, and direction for January 1957

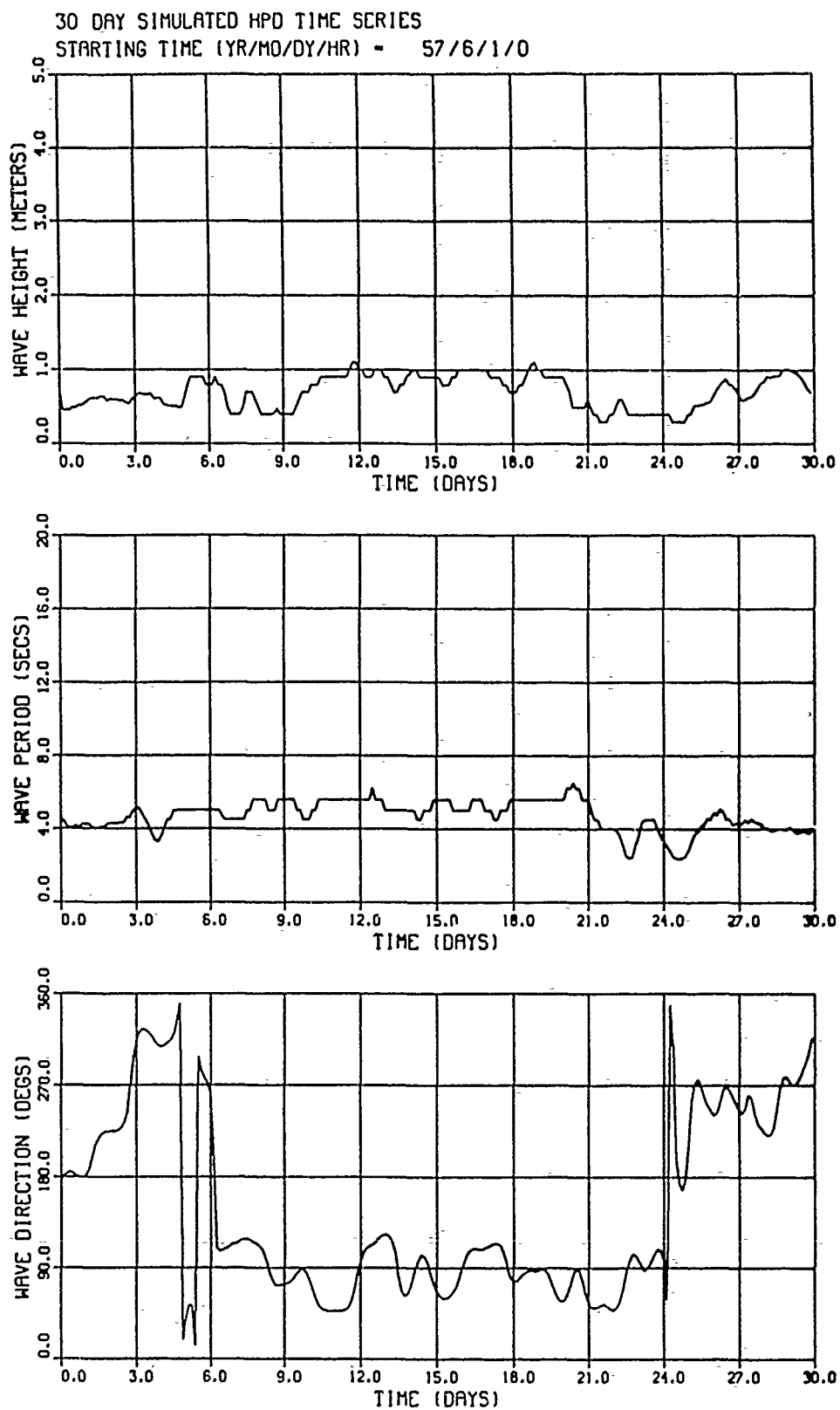


Figure 7. Simulated wave height, period, and direction for June 1957

Table 1
Wave Height Comparisons for 1957

Month	Simulated Wave Height, m				WIS Wave Height, m			
	Max	Min	Ave	SD	Max	Min	Ave	SD
Jan	2.30	0.50	1.38	0.36	2.20	0.40	1.20	0.40
Feb	3.03	0.70	1.37	0.41	2.20	0.40	1.17	0.36
Mar	2.47	0.70	1.27	0.33	2.20	0.50	1.17	0.32
Apr	1.77	0.49	1.03	0.23	3.40	0.80	1.34	0.48
May	2.19	0.40	1.00	0.29	2.00	0.40	0.95	0.33
Jun	1.10	0.30	0.70	0.22	2.90	0.30	1.00	0.46
Jul	1.12	0.40	0.71	0.19	1.10	0.30	0.65	0.23
Aug	1.66	0.22	0.72	0.25	2.40	0.30	0.91	0.36
Sept	3.85	0.40	1.10	0.43	3.30	0.40	1.12	0.46
Oct	2.36	0.50	1.18	0.35	2.70	0.50	1.27	0.50
Nov	2.10	0.70	1.27	0.29	2.90	0.40	1.40	0.59
Dec	2.87	0.29	1.34	0.44	2.90	0.50	1.35	0.45
Jan-Dec	3.85	0.22	1.09	0.41	3.40	0.30	1.12	0.47

Table 2
Wave Period Comparisons for 1957

Month	Simulated Wave Period, sec				WIS Wave Period, sec			
	Max	Min	Ave	SD	Max	Min	Ave	SD
Jan	7.70	4.30	5.63	0.84	7.70	2.40	5.47	1.01
Feb	8.30	4.50	5.72	0.77	7.70	2.00	5.48	1.07
Mar	7.70	3.44	5.67	0.78	7.70	3.20	5.62	0.79
Apr	6.70	3.22	5.16	0.64	9.10	4.50	6.02	1.05
May	7.44	2.72	5.20	0.80	7.70	2.60	5.20	1.02
Jun	6.49	2.35	4.71	0.83	8.30	2.60	5.35	1.09
Jul	5.60	2.80	4.64	0.47	6.20	2.80	4.80	0.59
Aug	6.20	1.70	4.39	0.81	8.30	2.00	4.81	1.03
Sep	7.70	3.60	5.29	0.86	10.00	3.40	5.63	1.03
Oct	7.99	3.87	5.46	0.90	9.10	3.00	5.36	1.27
Nov	7.70	4.14	5.48	0.67	9.10	3.50	5.83	1.22
Dec	7.83	1.24	5.47	1.02	9.10	3.60	5.81	0.92
Jan-Dec	8.30	1.24	5.23	0.90	10.00	2.00	5.45	1.08

365 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) - 57/1/1/0

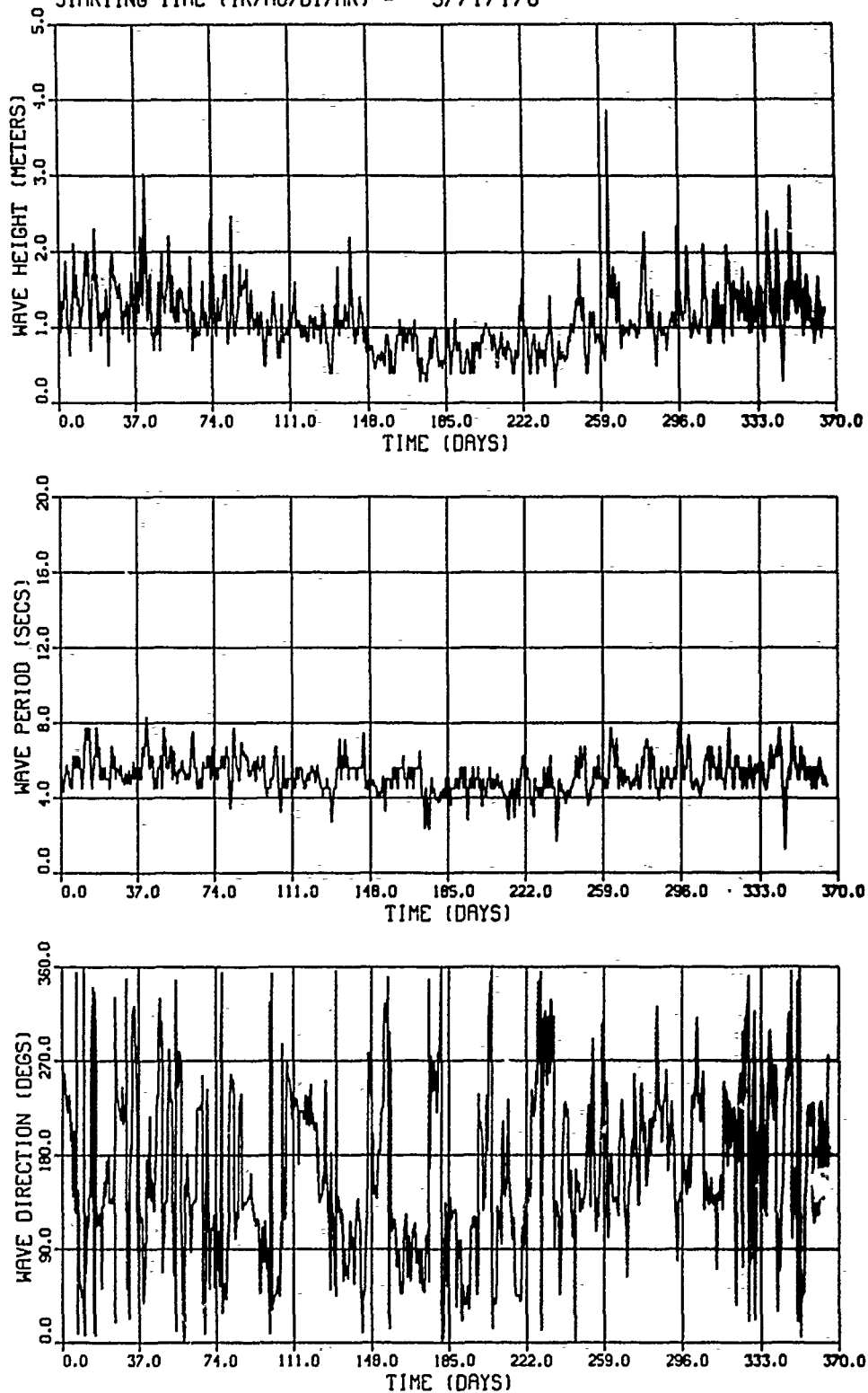


Figure 8. Simulated wave height, period, and direction for 1957

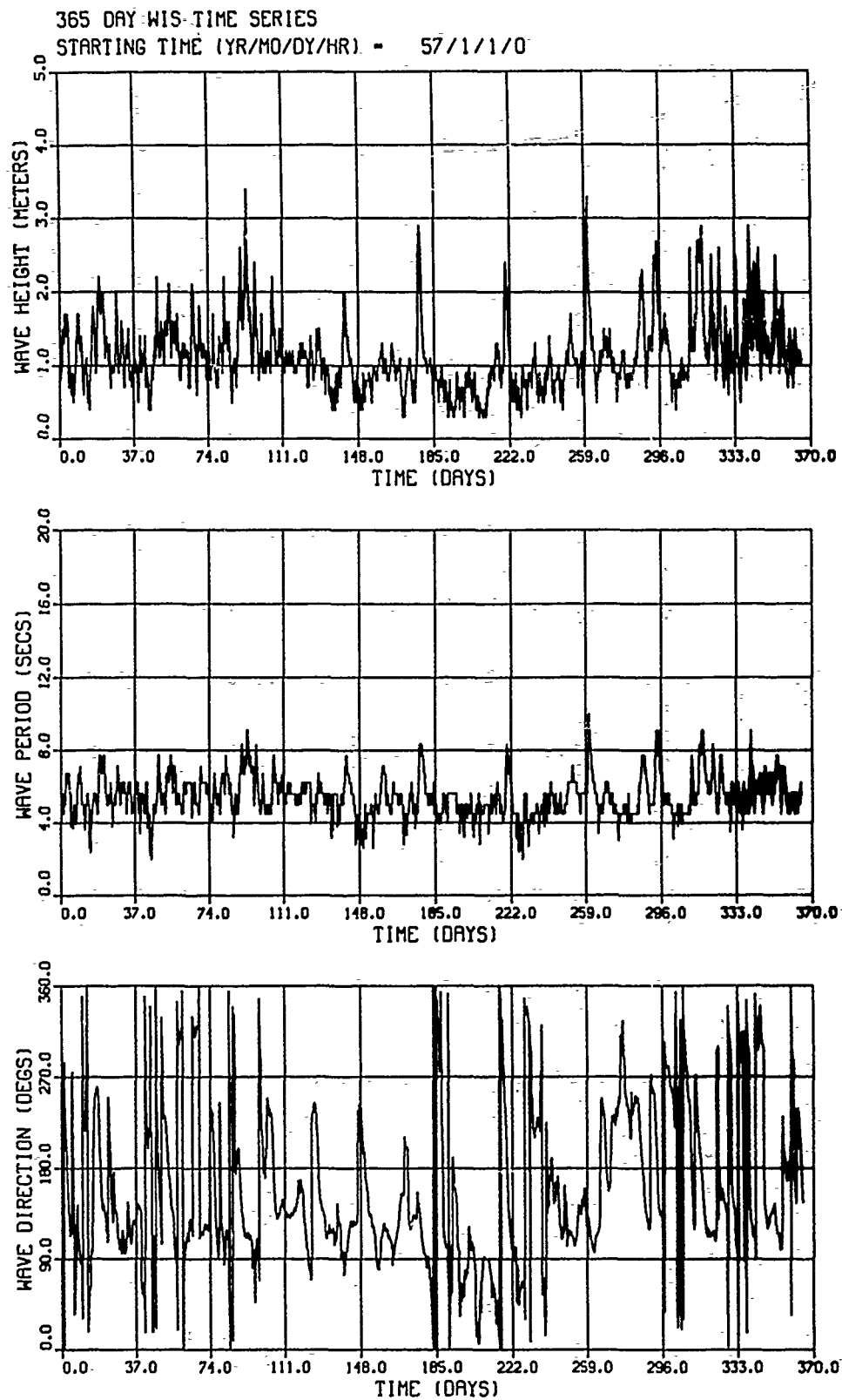


Figure 9. WIS wave height, period, and direction for 1957

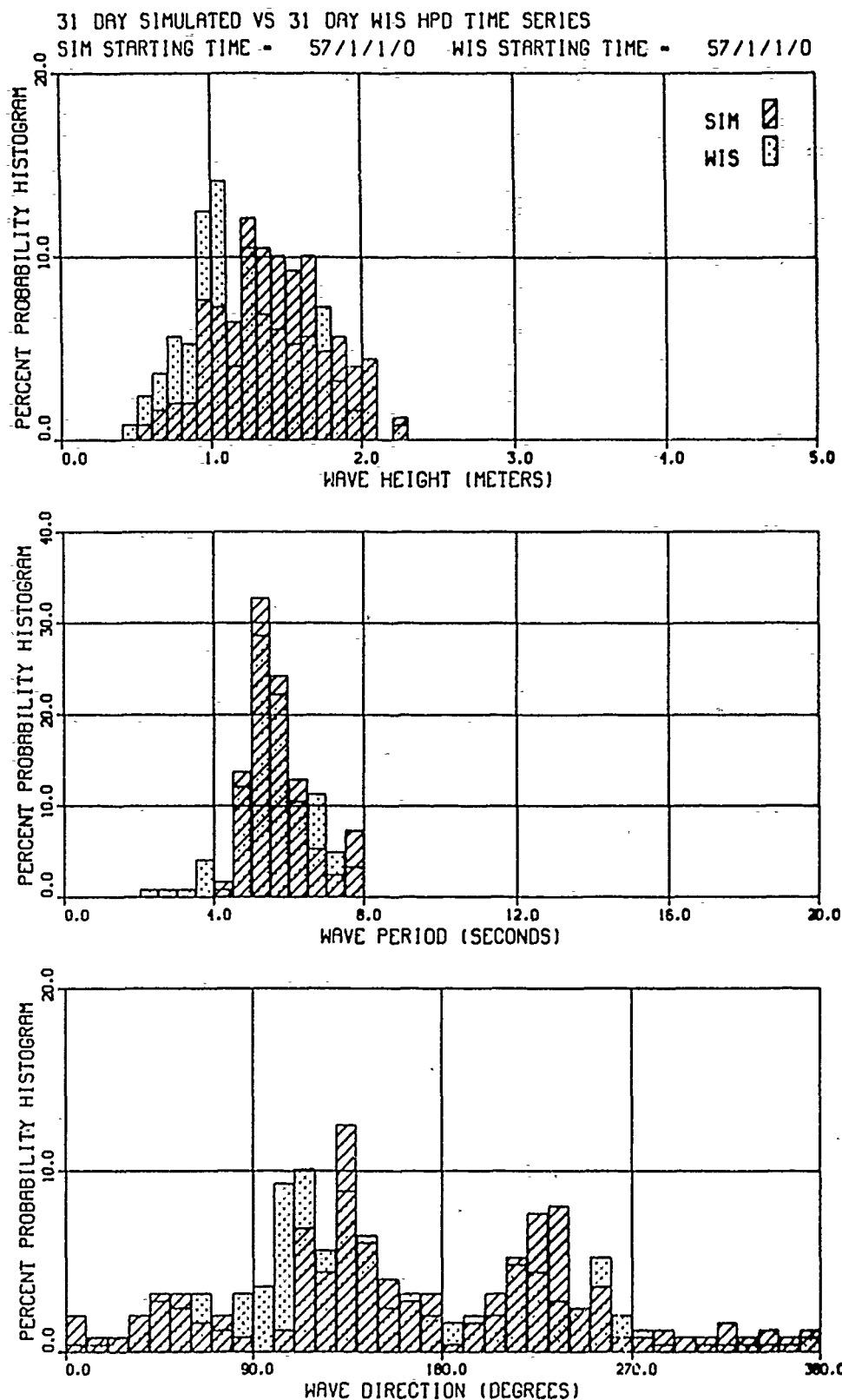


Figure 10. Probability histogram comparison for January 1957 data

directionality of the data that cannot be conveyed by a single mean value. For example, Figure 9 indicates that both the WIS and simulated January data series exhibit a bimodal directionality with a slight predominance in the band from 100 to 120 deg. This bimodal distribution is not shown in the Figure 11 WIS data for June, although it is indicated in the simulated data. This example demonstrates that the simulated data reflect the trends of the entire database, not merely 1 month.

59. The natural variability of the WIS database can be seen in direction histogram plots for June 1956, 1961, and 1966 shown in Figure 12. The year of 1956 shows a bimodal distribution while 1961 and 1966 do not. Also the predominant direction of the 1956 data is centered at approximately 90 deg, whereas it is centered at nearly 200 deg in 1966. A similar variability in simulated data can be demonstrated by selecting different starting random number seeds. Figure 13 presents three different directional histogram plots for June of 1957, each generated with a different seed. As shown, the plots reflect the same degree of variability shown in the WIS data. As the number of observations used in the histogram calculations increases, the plots exhibit less variability and approach the long-term average of the data. Figure 14 represents a directional histogram plot for all 20 Junes contained in the WIS database of 1956 through 1975. Results show the primary direction to be in the 110- to 120-deg band, but that waves from all directions are possible. For comparison purposes, the probability histogram for the year 1957, corresponding to Figures 8 and 9, is shown in Figure 15.

60. An additional simulation was performed for a WIS location exhibiting a different wave climate than that of Gulf of Mexico Station 47 shown in the previous examples. The example demonstrates the ability of the program to generate statistically similar time series for different wave regimes. The example presented is for the Phase III Cape Canaveral data Station 147 shown in Figure 16. Figures 17 and 18 represent the simulation and WIS data series for the year 1956. Figure 19 presents the comparison histograms for both data series. As shown, the wave sequences are similar in both magnitude and temporal distribution.

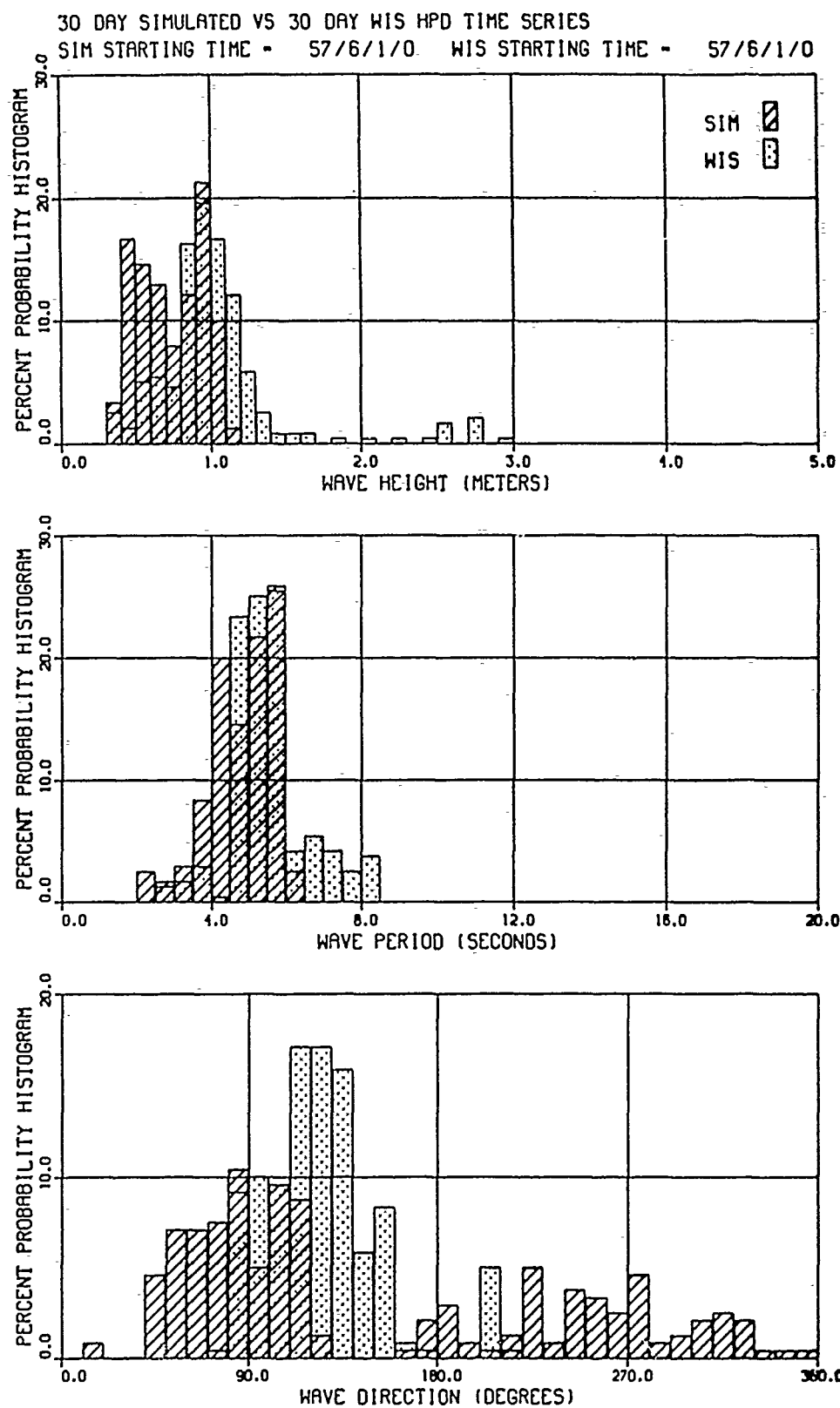


Figure 11. Probability histogram comparison for June 1957 data

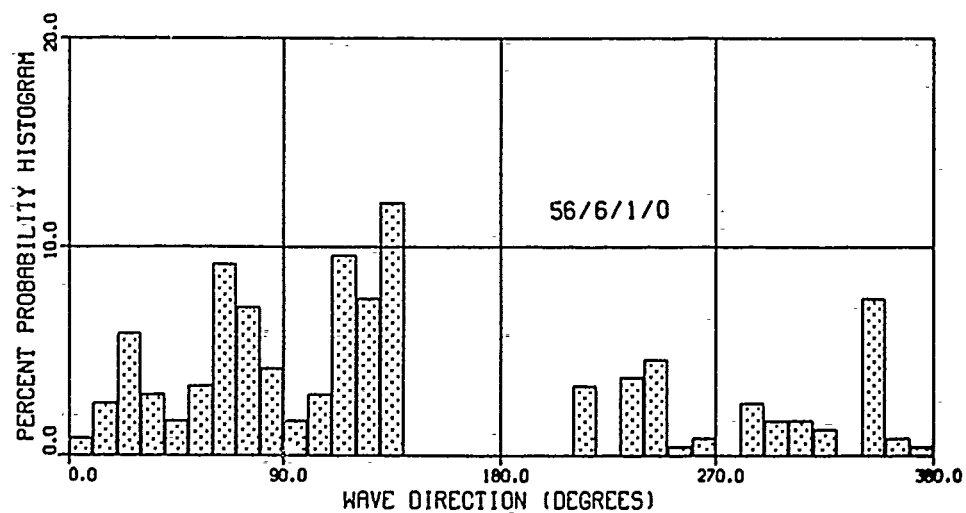
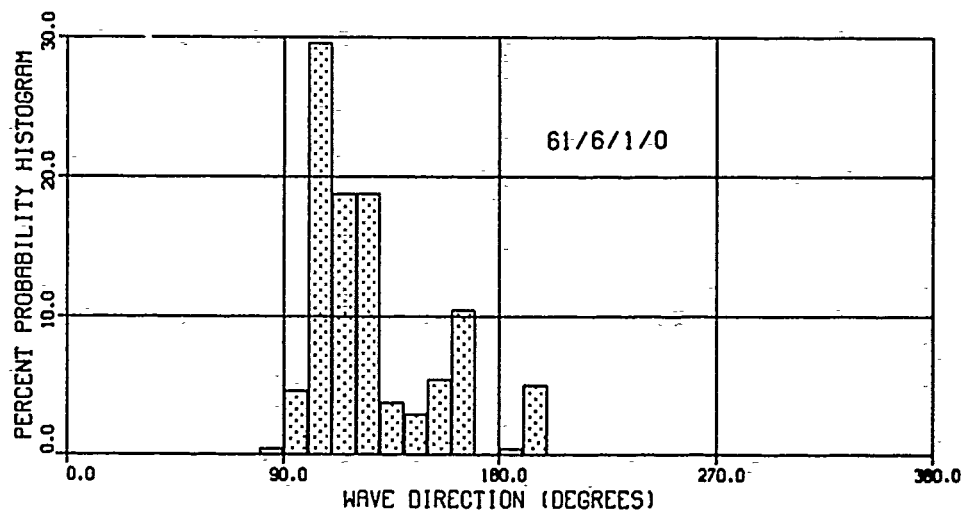
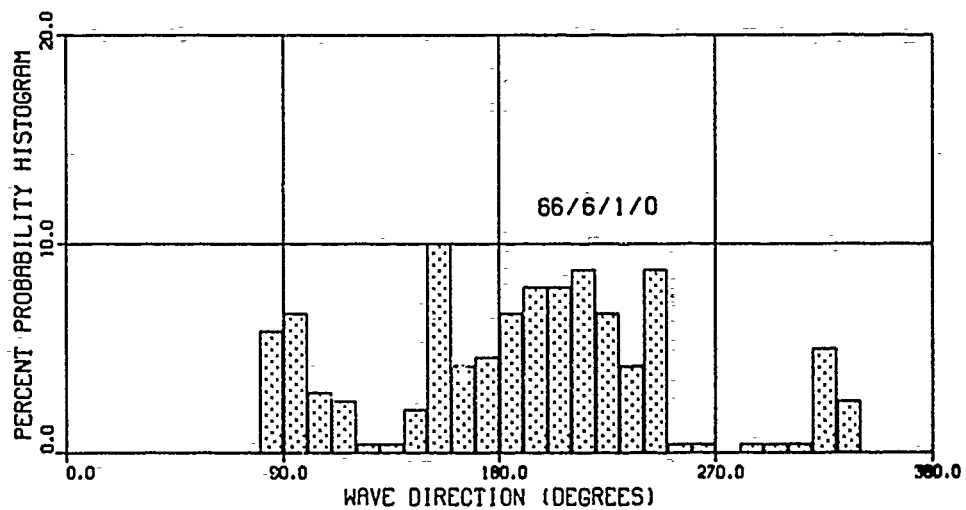


Figure 12. WIS wave direction histogram plots for June 1956, 1961, and 1966

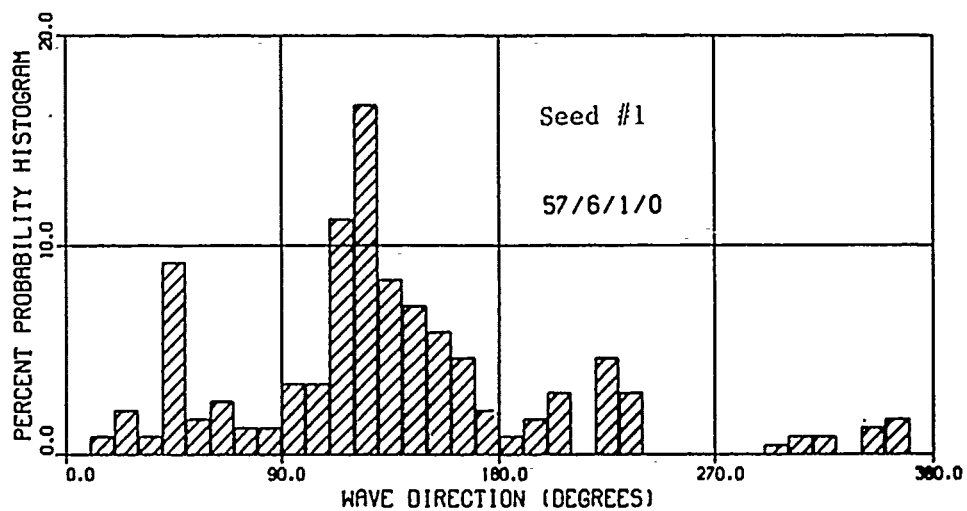
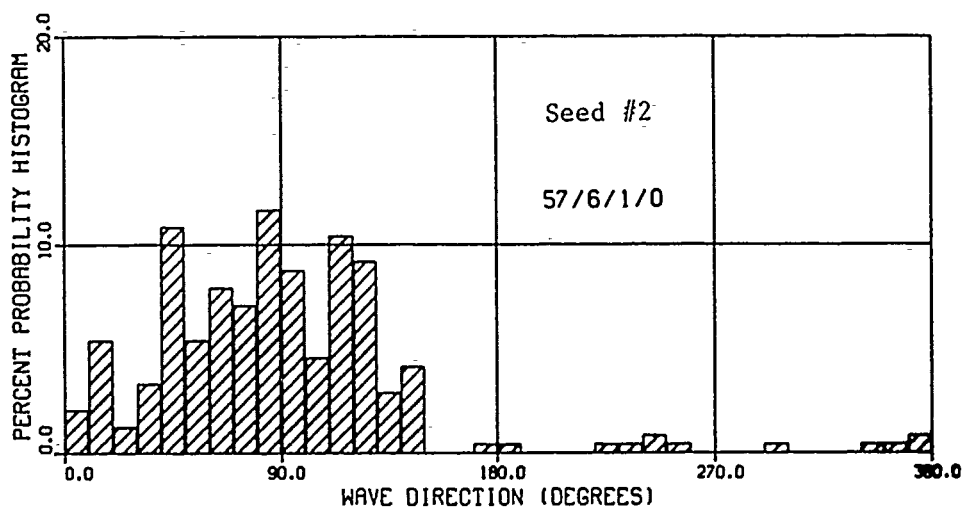
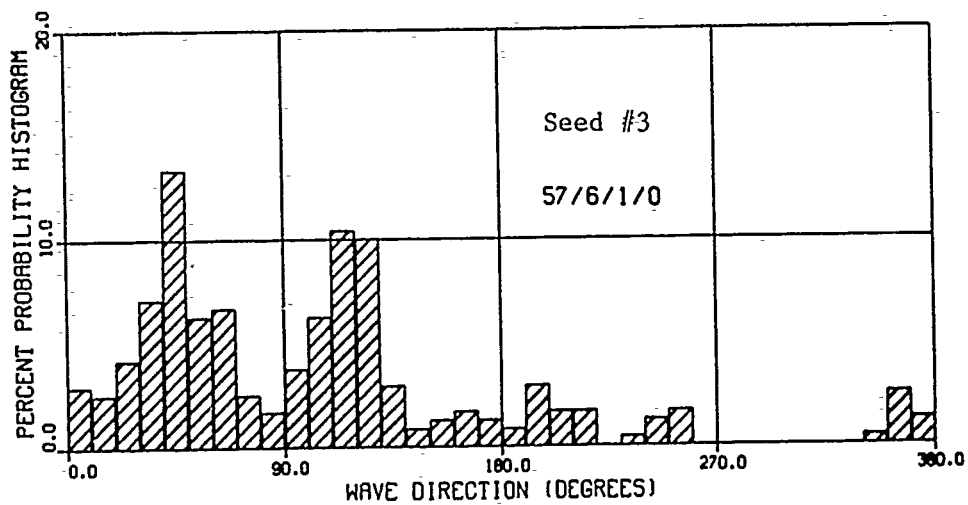


Figure 13. Three simulated wave direction histogram plots for June 1957

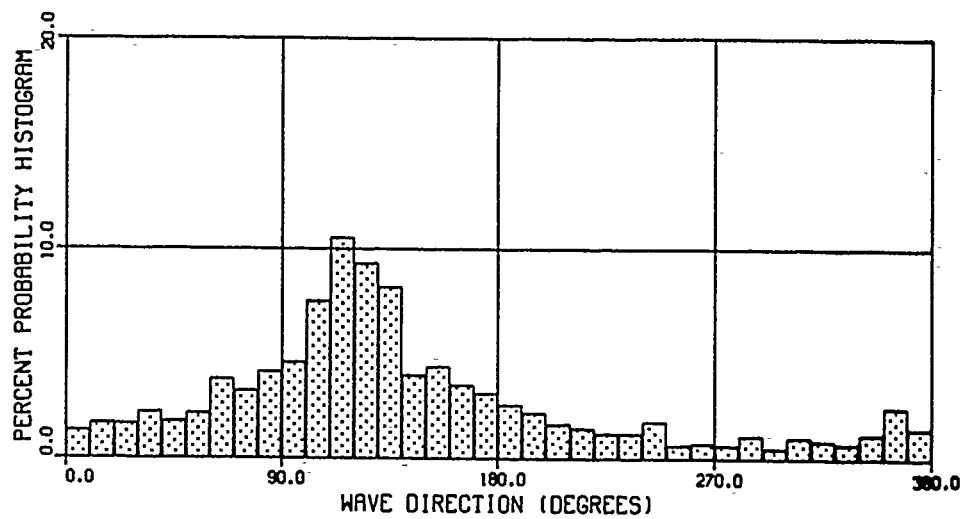


Figure 14. WIS wave direction histogram for all Junes 1956-1975

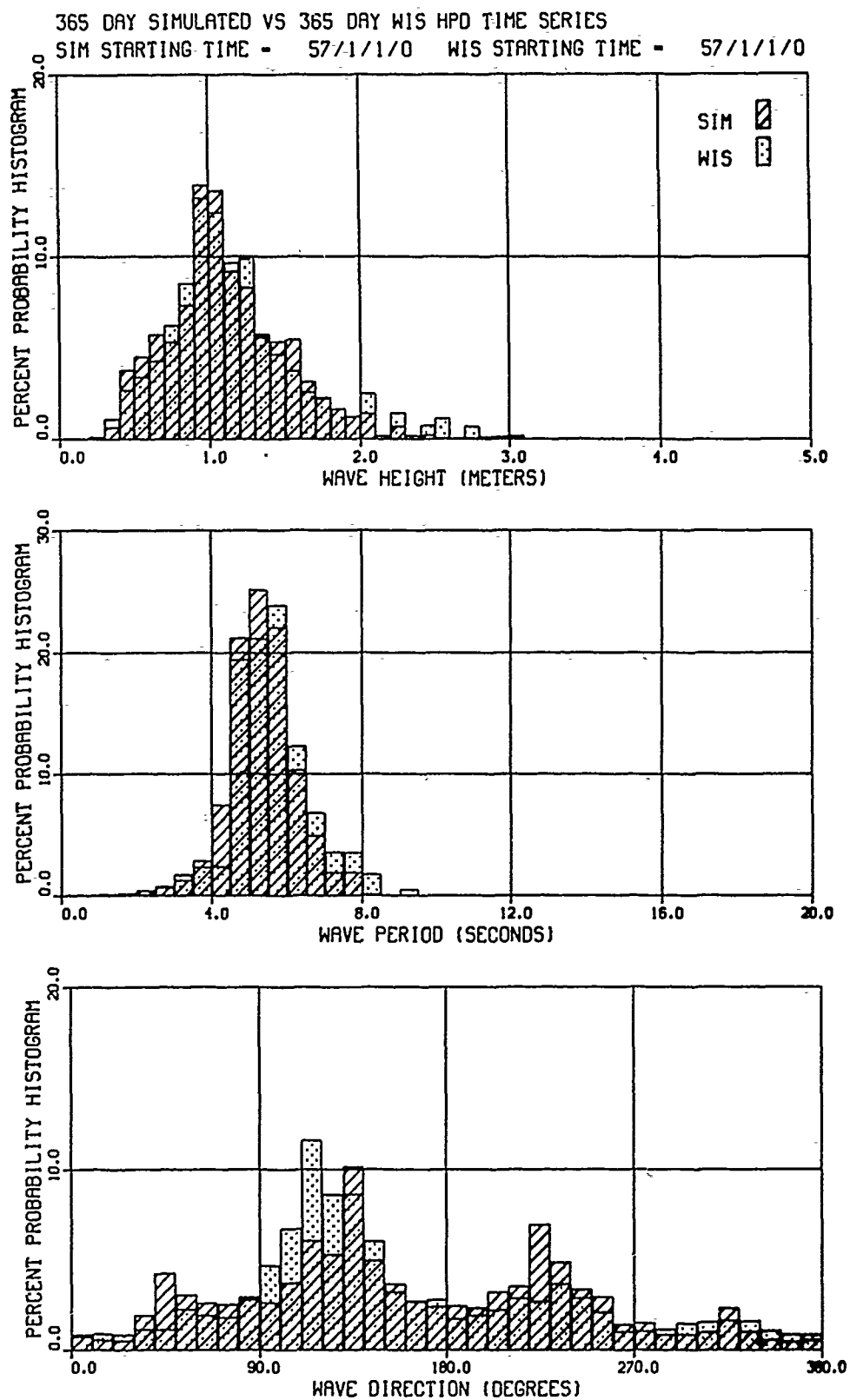


Figure 15. Probability histogram comparison for 1957 data

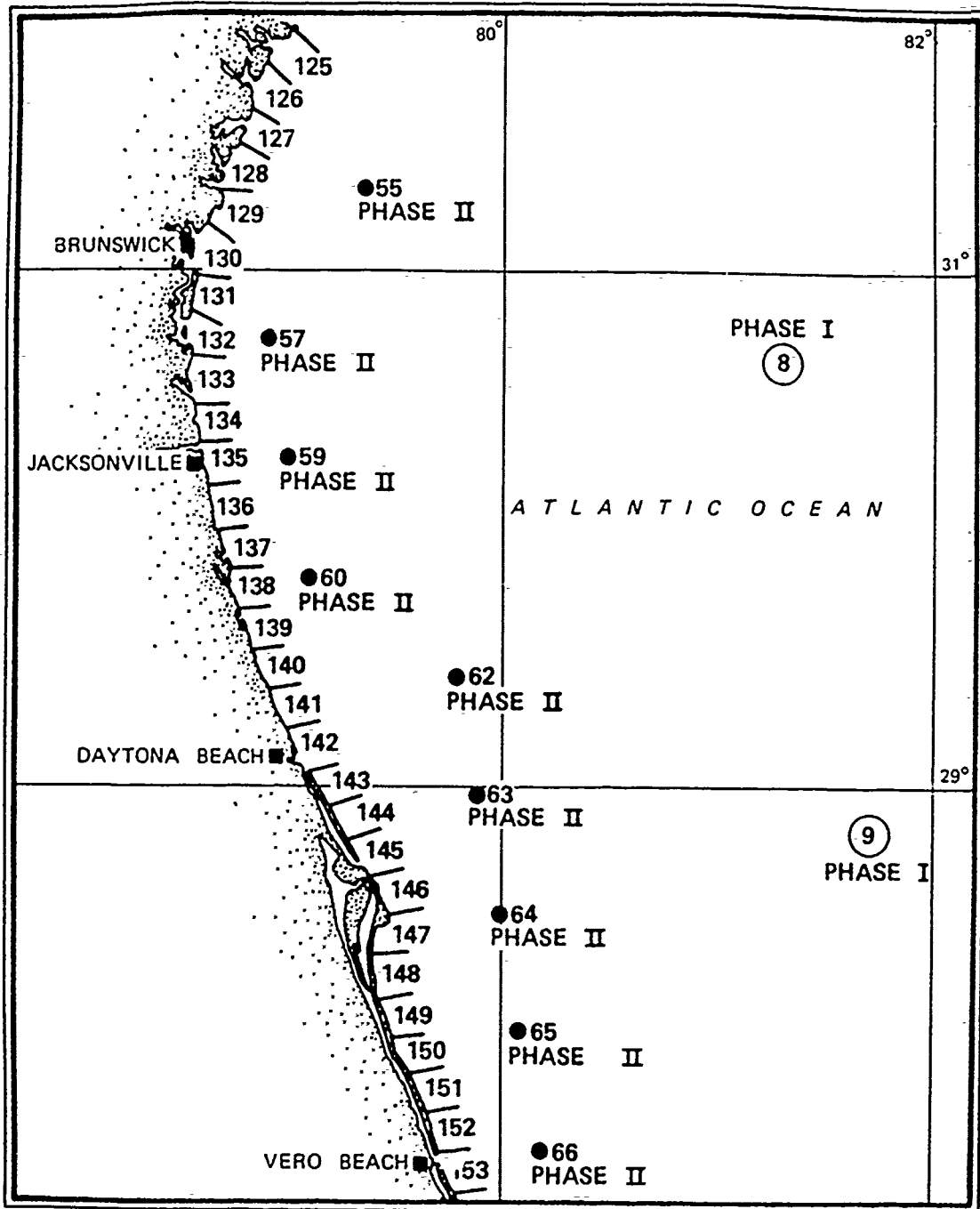


Figure 16. Location map for WIS Station 147

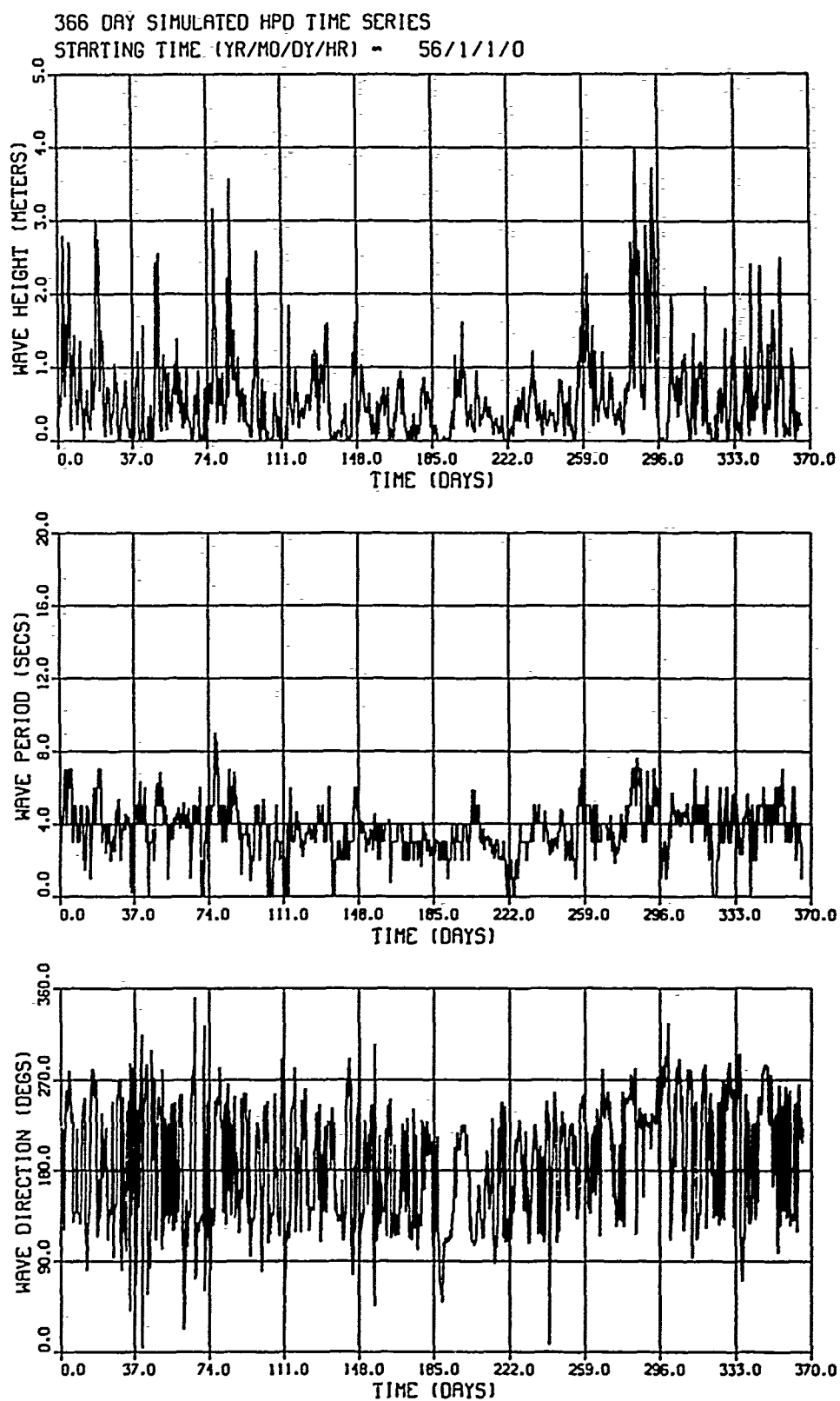


Figure 17. Simulated Station 147 wave height, period, and direction for 1956

366 DAY WIS TIME SERIES

STARTING TIME (YR/MO/DY/HR) - 56/1/1/0

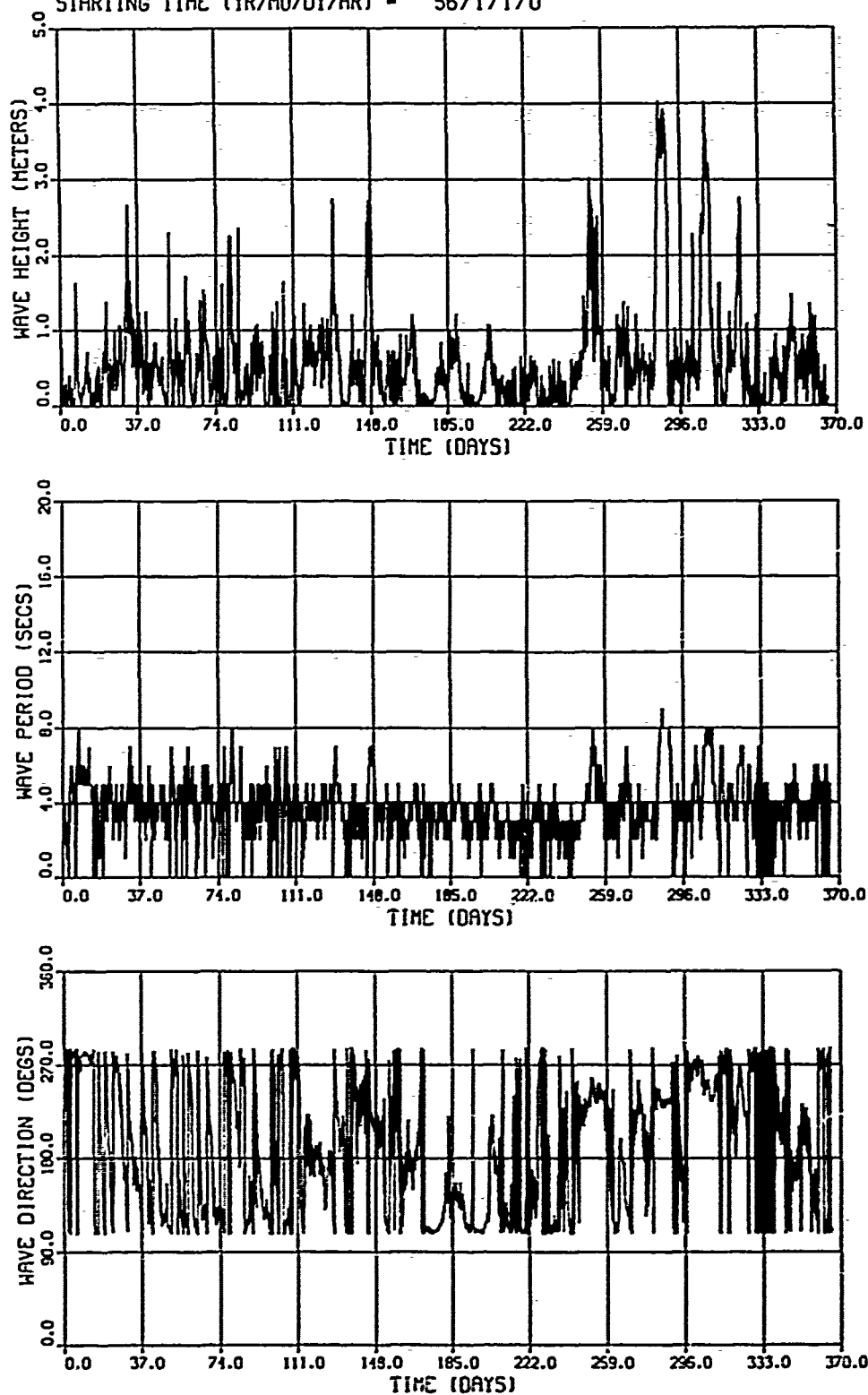


Figure 18. WIS Station 147 wave height, period, and direction for 1956

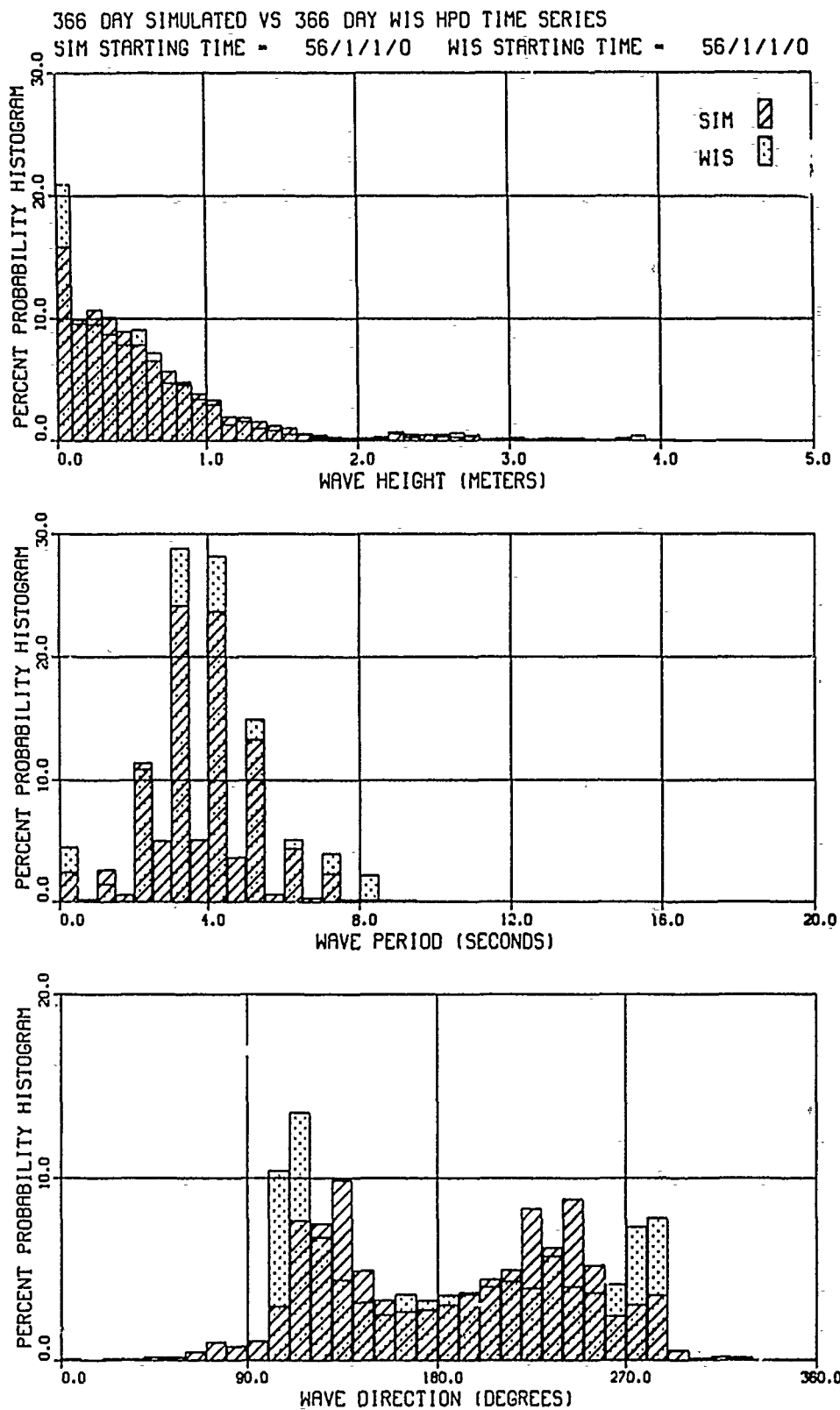


Figure 19. Probability histogram comparison of 1956 data for Station 147

PART V: CONCLUSIONS

61. The simulation program presented herein has been shown to be capable of simulating artificial sequences of two time-correlated variables and their direction. The methodology of the simulation is to use a fixed length of time-correlated data (wave height, period, and direction, for example) to compute the covariance matrix describing the intercorrelations of each of the variables of the data set. This matrix of coefficient multipliers is then used to generate a synthetic data series that preserves the primary statistical properties of the original data set, including seasonality and wave sequencing. The data can be generated for any length of time such that a finite length of data can be used to generate an infinite amount of data which contains the statistical attributes of the original data.

62. The examples presented demonstrate that the procedure is successful in generating synthetic wave time series which are statistically similar to prototype conditions and exhibit the proper distribution of observed data variability shown in the prototype records. This generation capability does not require access to the prototype data once the intercorrelation matrix has been computed. If the user does not have the facilities for generating the matrix, it can be made available by the US Army Engineer Waterways Experiment Station. This matrix availability is an important consideration since the WIS database requires more storage capacity than can be accommodated by most PC's; however, the simulation algorithm is PC based, and the user can easily generate any number or length of time series by merely specifying starting and ending time delimiters.

63. This statistical procedure represents an improved methodology for specifying time series boundary conditions over the use of discrete or repeated historical data series. The approach represents the stochastic equivalent of generating infinitely long tidal series from a finite number of harmonic constituents. The methodology gives the user the capability of analyzing the variability of certain statistical parameters of an entire prototype database such that specific prototype data periods can be analyzed with respect to their variability about mean conditions. The procedure can be used to generate future time series of wave data, thus avoiding the practice of selecting design data from past prototype conditions or of having to repeat fixed data series in order to compose a longer time series of data than is available. The methodology is therefore recommended for any coastal application requiring

either the analysis of past time series or the generation of variable length future time series boundary conditions.

REFERENCES

- Borgman, L. E. 1982. "Techniques for Computer Simulation of Ocean Waves," Topics in Ocean Physics, Italian Physical Society, Bologna, Italy, pp 387-417.
- _____. "Irregular Ocean Waves: Kinematics and Forces," The Sea, Ocean Engineering Science, Vol 9, in press, McGraw-Hill, New York.
- Borgman, L. E., and Resio, D. T. 1982. "External Statistics in Wave Climatology," Topics in Ocean Physics, Italian Physical Society, Bologna, Italy, pp 439-471.
- Gumbel, E. J. 1954. "Statistical Theory of Extreme Value and Some Practical Application," National Bureau of Standards Applied Math. Series 33, US Government Publication, Washington, DC.
- Zelen, M., and Severo, N. C. 1965. "Probability Functions," Handbook of Mathematical Functions, M. Abramowitz and I. Stegun, ed., Dover Publishers Inc., New York, pp 925-995.

APPENDIX A: SCREEN LISTING DEVELOPED DURING HPDPRE

HPDPRE.SCR

READING DATABASE, IY= 56
 READING DATABASE, IY= 57
 READING DATABASE, IY= 58
 READING DATABASE, IY= 59
 READING DATABASE, IY= 60
 READING DATABASE, IY= 61
 READING DATABASE, IY= 62
 READING DATABASE, IY= 63
 READING DATABASE, IY= 64
 READING DATABASE, IY= 65
 READING DATABASE, IY= 66
 READING DATABASE, IY= 67
 READING DATABASE, IY= 68
 READING DATABASE, IY= 69
 READING DATABASE, IY= 70
 READING DATABASE, IY= 71
 READING DATABASE, IY= 72
 READING DATABASE, IY= 73
 READING DATABASE, IY= 74
 READING DATABASE, IY= 75

WHAT TITLE DO YOU WANT ATTACHED TO THIS ANALYSIS?

(UP TO 50 CHARACTERS MAY BE ENTERED)

Example Run. 11/2/89

JM= 1
 MONTHS= 1 1 1 1
 PROCEEDING WITH ANALYSIS OF MONTH 1
 Z1,Z2,Z3= 0.5000000 0.4000000 0.0000000E+00
 T1,T2,T3= 0.0000000E+00 4.1493778E-03 8.2987556E-03
 C: W,WW= 0.2000000 0.5000000
 C= 2.321928
 BB: W,WW= 0.5000000 8.2987556E-03
 Z1,Z2,Z3= 2.800000 3.100000 4.200000
 T1,T2,T3= 2.961819 3.095690 3.312038
 C: W,WW= 0.2142857 0.3822517
 C= 1.601834
 BB: W,WW= 1.400000 0.3502190
 Z1,Z2,Z3= 3.600000 3.300000 0.0000000E+00
 T1,T2,T3= 0.0000000E+00 4.1493778E-03 8.2987556E-03
 C: W,WW= 8.3333321E-02 0.5000000
 C= 3.584963
 BB: W,WW= 3.600000 8.2987556E-03
 Z1,Z2,Z3= 8.300000 9.100000 11.10000
 T1,T2,T3= 2.961819 3.095690 3.312038
 C: W,WW= 0.2857143 0.3822517
 C= 1.302687
 BB: W,WW= 2.800000 0.3502190
 Z1,Z2,Z3= -0.9993908 -0.9998477 -1.000000
 T1,T2,T3= 0.0000000E+00 8.2987556E-03 1.2448133E-02
 C: W,WW= 0.7500000 0.6666667
 C= 0.7095114
 BB: W,WW= 6.0915947E-04 1.2448133E-02

HPDPRE.SCR

```

AT PRINT RESULTS, JM=          1
MONTHS=          1          1          1          1
STORING PREPROCESS OUTPUT FOR MONTH 1
JM=          2
MONTHS=          1          1          1          1
PROCEEDING WITH ANALYSIS OF MONTH 2
Z1,Z2,Z3=    0.5000000    0.4000000    0.0000000E+00
T1,T2,T3=    0.0000000E+00  8.2987556E-03  1.2448133E-02
C: W,WW=    0.2000000    0.6666667
C=    3.969363
BB: W,WW=    0.5000000    1.2448133E-02
Z1,Z2,Z3=    2.900000    3.300000    4.100000
T1,T2,T3=    2.961819    3.095690    3.312038
C: W,WW=    0.3333333    0.3822517
C=    1.142393
BB: W,WW=    1.200000    0.3502190
Z1,Z2,Z3=    3.200000    2.700000    0.0000000E+00
T1,T2,T3=    0.0000000E+00  4.1493778E-03  8.2987556E-03
C: W,WW=    0.1562500    0.5000000
C=    2.678072
BB: W,WW=    3.200000    8.2987556E-03
Z1,Z2,Z3=    8.300000    9.100000    10.00000
T1,T2,T3=    2.863041    3.095690    3.312038
C: W,WW=    0.4705884    0.5181540
C=    1.146450
BB: W,WW=    1.700000    0.4489965
Z1,Z2,Z3=   -0.9993908   -0.9998477   -1.000000
T1,T2,T3=    0.0000000E+00  8.2987556E-03  1.2448133E-02
C: W,WW=    0.7500000    0.6666667
C=    0.7095114
BB: W,WW=    6.0915947E-04  1.2448133E-02
AT PRINT RESULTS, JM=          2
MONTHS=          1          1          1          1
STORING PREPROCESS OUTPUT FOR MONTH 2
JM=          3

```

ETC., FOR REST OF THE 12 MONTHS

APPENDIX B: OUTPUT FROM HPDPRE

HPDPRE.OUT

FOR WES VISIT. 10/30/89

```

NYR,NNY,NMO=    20    20    12
IYEARS=    1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
MONTHS=    1  1  1  1  1  1  1  1  1  1  1  1  1
CUTOFF=    0.083333    0.083333    0.083333    0.083333    0.083333
CUTOFF=    0.083333    0.083333    0.083333    0.083333    0.083333
CUTOFF=    0.083333    0.083333
IJY=    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
IJM=    1  2  3  4  5  6  7  8  9 10 11 12 PREPROCESS OUTPUT FOR MO.    1
SCALE FACTORS:    1.00000    1.00000
N,M,NTM,NTOP,NSIM=    240    20    4800    120    480
DT=    3.0000
BW,FCO,FRACT=    0.16667E-01    0.83333E-01    0.99990
RANKED ORIGINAL {H(J);J=M,NTM,M}
0.40000    0.50000    0.50000    0.50000    0.50000
0.60000    0.60000    0.60000    0.60000    0.70000
0.70000    0.70000    0.70000    0.70000    0.70000
0.70000    0.80000    0.80000    0.80000    0.80000
0.90000    0.90000    0.90000    0.90000    0.90000
0.90000    0.90000    0.90000    0.90000    0.90000
0.90000    0.90000    0.90000    0.90000    0.90000
0.90000    0.90000    1.0000    1.0000    1.0000
1.0000    1.0000    1.0000    1.0000    1.0000
1.0000    1.0000    1.0000    1.0000    1.0000
1.0000    1.0000    1.0000    1.0000    1.0000
1.0000    1.0000    1.0000    1.0000    1.0000
1.1000    1.1000    1.1000    1.1000    1.1000
1.1000    1.1000    1.1000    1.1000    1.1000
1.1000    1.1000    1.1000    1.1000    1.1000
1.1000    1.1000    1.1000    1.1000    1.2000
1.2000    1.2000    1.2000    1.2000    1.2000
1.2000    1.2000    1.2000    1.2000    1.2000
1.2000    1.2000    1.2000    1.2000    1.2000
1.2000    1.2000    1.2000    1.2000    1.2000
1.2000    1.2000    1.2000    1.3000    1.3000
1.3000    1.3000    1.3000    1.3000    1.3000
1.3000    1.3000    1.3000    1.3000    1.3000
1.3000    1.3000    1.3000    1.3000    1.3000
1.3000    1.3000    1.3000    1.4000    1.4000
1.4000    1.4000    1.4000    1.4000    1.4000
1.4000    1.4000    1.4000    1.4000    1.4000
1.4000    1.4000    1.4000    1.4000    1.4000
1.5000    1.5000    1.5000    1.5000    1.5000
1.5000    1.5000    1.5000    1.5000    1.5000
1.5000    1.5000    1.5000    1.5000    1.5000
1.5000    1.5000    1.5000    1.5000    1.5000
1.6000    1.6000    1.6000    1.6000    1.6000
1.6000    1.6000    1.6000    1.6000    1.6000
1.6000    1.6000    1.6000    1.6000    1.6000
1.7000    1.7000    1.7000    1.7000    1.7000
1.7000    1.7000    1.7000    1.7000    1.7000

```

HPDPRE.OUT

RANKED ORIGINAL [T(J);J=M,NTM,M]

HPDPRE.OUT

6.7000	6.7000	6.7000	6.7000	6.7000
6.7000	6.7000	6.7000	6.7000	6.7000
6.7000	6.7000	6.7000	6.7000	6.7000
7.1000	7.1000	7.1000	7.1000	7.1000
7.1000	7.1000	7.1000	7.1000	7.1000
7.7000	7.7000	7.7000	7.7000	7.7000
7.7000	7.7000	7.7000	7.7000	7.7000
8.3000	8.3000	8.3000	9.1000	11.100
RANKED ORIGINAL [DS(J);J=M,NTM,M]				
-0.99985	-0.99985	-0.99939	-0.99756	-0.99756
-0.99619	-0.99452	-0.99452	-0.99255	-0.99027
-0.99027	-0.98769	-0.98481	-0.98163	-0.97815
-0.97437	-0.97030	-0.96593	-0.96593	-0.96126
-0.96126	-0.95630	-0.95106	-0.94552	-0.94552
-0.93969	-0.93969	-0.93358	-0.92718	-0.92050
-0.91355	-0.90631	-0.90631	-0.89879	-0.89101
-0.88295	-0.87462	-0.86603	-0.85717	-0.84805
-0.83867	-0.83867	-0.82904	-0.81915	-0.80902
-0.79864	-0.79864	-0.78801	-0.77715	-0.76604
-0.75471	-0.75471	-0.73135	-0.71934	-0.70711
-0.69466	-0.68200	-0.66913	-0.65606	-0.65606
-0.64279	-0.62932	-0.61566	-0.61566	-0.60182
-0.58779	-0.57358	-0.55919	-0.54464	-0.54464
-0.51504	-0.50000	-0.48481	-0.46947	-0.45399
-0.43837	-0.42262	-0.40674	-0.39073	-0.37461
-0.35837	-0.34202	-0.32557	-0.32557	-0.30902
-0.29237	-0.27564	-0.25882	-0.24192	-0.22495
-0.20791	-0.19081	-0.17365	-0.15643	-0.15643
-0.13917	-0.13917	-0.12187	-0.10453	-0.87156E-01
-0.69756E-01	-0.52336E-01	-0.34900E-01	-0.34899E-01	-0.17452E-01
-0.87423E-07	0.17452E-01	0.34899E-01	0.34899E-01	0.69756E-01
0.87156E-01	0.87156E-01	0.10453	0.12187	0.12187
0.13917	0.15643	0.17365	0.19081	0.20791
0.22495	0.22495	0.25882	0.25882	0.27564
0.29237	0.30902	0.32557	0.32557	0.34202
0.35837	0.35837	0.37461	0.40674	0.40674
0.42262	0.43837	0.45399	0.46947	0.48481
0.48481	0.50000	0.51504	0.52992	0.54464
0.55919	0.57358	0.57358	0.58779	0.58779
0.60182	0.60182	0.61566	0.62932	0.62932
0.64279	0.64279	0.65606	0.65606	0.66913
0.68200	0.69466	0.69466	0.70711	0.70711
0.71934	0.71934	0.73135	0.73135	0.74314
0.75471	0.75471	0.76604	0.76604	0.77715
0.77715	0.78801	0.78801	0.79864	0.79864
0.80902	0.80902	0.80902	0.81915	0.81915
0.81915	0.82904	0.83867	0.83867	0.84805
0.85717	0.85717	0.86603	0.86603	0.86603
0.87462	0.88295	0.88295	0.89101	0.89101
0.89879	0.89879	0.89879	0.90631	0.90631
0.90631	0.91355	0.91355	0.92050	0.92050

HPDPRE.OUT

0.92718	0.92718	0.93358	0.93969	0.93969
0.94552	0.94552	0.95106	0.95106	0.95630
0.95630	0.96126	0.96593	0.97030	0.97030
0.97437	0.97815	0.97815	0.98163	0.98481
0.98769	0.98769	0.99027	0.99255	0.99452
0.99619	0.99756	0.99863	0.99939	1.0000
RANKED ORIGINAL [DC(J);J=M,NTM,M]				
-1.0000	-0.99985	-0.99985	-0.99939	-0.99939
-0.99863	-0.99863	-0.99756	-0.99756	-0.99619
-0.99619	-0.99452	-0.99452	-0.99255	-0.99255
-0.99027	-0.99027	-0.99027	-0.98769	-0.98769
-0.98481	-0.98481	-0.98163	-0.98163	-0.97815
-0.97815	-0.97437	-0.97437	-0.97030	-0.97030
-0.96593	-0.96593	-0.96126	-0.96126	-0.95630
-0.95630	-0.95106	-0.95106	-0.94552	-0.94552
-0.93969	-0.93969	-0.93969	-0.93358	-0.93358
-0.92718	-0.92718	-0.92050	-0.92050	-0.91355
-0.91355	-0.90631	-0.90631	-0.89879	-0.89879
-0.89101	-0.88295	-0.88295	-0.87462	-0.87462
-0.87462	-0.86603	-0.85717	-0.85717	-0.84805
-0.84805	-0.83867	-0.83867	-0.82904	-0.82904
-0.81915	-0.81915	-0.80902	-0.80902	-0.80902
-0.79864	-0.79864	-0.78801	-0.78801	-0.78801
-0.77715	-0.77715	-0.76604	-0.76604	-0.76604
-0.75471	-0.75471	-0.75471	-0.74314	-0.74314
-0.73135	-0.73135	-0.71934	-0.71934	-0.70711
-0.70711	-0.69466	-0.69466	-0.69466	-0.68200
-0.68200	-0.66913	-0.66913	-0.65606	-0.65606
-0.64279	-0.64279	-0.64279	-0.62932	-0.62932
-0.62932	-0.61566	-0.61566	-0.60182	-0.60182
-0.60182	-0.58779	-0.58779	-0.58779	-0.57358
-0.57358	-0.57358	-0.55919	-0.55919	-0.54464
-0.54464	-0.52992	-0.52992	-0.51504	-0.51504
-0.51504	-0.50000	-0.50000	-0.50000	-0.48481
-0.48481	-0.46947	-0.46947	-0.45399	-0.45399
-0.43837	-0.43837	-0.43837	-0.43837	-0.42262
-0.42262	-0.42262	-0.40674	-0.40674	-0.40674
-0.39073	-0.39073	-0.37461	-0.37461	-0.35837
-0.35837	-0.35837	-0.34202	-0.34202	-0.32557
-0.32557	-0.32557	-0.30902	-0.30902	-0.30902
-0.29237	-0.29237	-0.27564	-0.27564	-0.27564
-0.25882	-0.25882	-0.24192	-0.22495	-0.22495
-0.20791	-0.20791	-0.19081	-0.17365	-0.17365
-0.15643	-0.13917	-0.13917	-0.12187	-0.10453
-0.10453	-0.87156E-01	-0.69757E-01	-0.52336E-01	-0.34900E-01
-0.17452E-01	0.11925E-07	0.17452E-01	0.34899E-01	0.52336E-01
0.69757E-01	0.87156E-01	0.10453	0.12187	0.13917
0.15643	0.17365	0.19081	0.20791	0.24192
0.25882	0.27564	0.30902	0.32557	0.34202
0.35837	0.39073	0.40674	0.43837	0.45399
0.48481	0.51504	0.52992	0.55919	0.57358

HPDPRE.OUT

0.57358	0.60182	0.62932	0.64279	0.66913	
0.69466	0.73135	0.75471	0.77715	0.80902	
0.82904	0.85717	0.89879	0.92718	0.95106	
0.97030	0.98481	0.99255	0.99756	1.0000	
J,FREQ,NEV=	1	0.000000	4		
82.1213	165.401	0.000000E+000	0.000000E+00		
172.236	114.175	0.000000E+000	0.000000E+00		
156.656	41.4496	0.000000E+000	0.000000E+00		
-191.237	207.812	0.000000E+000	0.000000E+00		
0.000000E+000	0.000000E+000	0.000000E+000	0.000000E+00		
0.000000E+000	0.000000E+000	0.000000E+000	0.000000E+00		
0.000000E+000	0.000000E+000	0.000000E+000	0.000000E+00		
0.000000E+000	0.000000E+000	0.000000E+000	0.000000E+00		
J,FREQ,NEV=	2	0.000694	8		
-80.6793	-20.8271	58.9524	6.37815	99.5073	1.71672
20.8271	-80.6793	6.37813	-58.9524	-1.71672	99.5073
-84.3646	-25.2291	46.1587	78.5935	54.5131	23.2819
25.2291	-84.3646	78.5935	-46.1587	-23.2819	54.5131
20.0377	-19.7668	-4.02857	161.010	-53.7832	53.5720
19.7668	20.0377	161.010	4.02856	-53.5719	-53.7832
179.649	-23.1725	14.3945	36.5441	83.2911	-18.2884
23.1726	179.649	36.5441	-14.3945	18.2884	83.2911
J,FREQ,NEV=	3	0.001389	8		
-89.9804	4.04406	-21.1232	81.8348	69.6503	-21.1041
4.04406	89.9804	-81.8348	-21.1232	21.1041	69.6503
-70.6850	-4.68033	32.0676	68.3669	64.9075	36.1004
-4.68033	70.6850	-68.3669	32.0676	-36.1004	64.9075
26.2562	26.6207	135.000	-2.68513	1.91987	83.3607
26.6207	-26.2562	2.68513	135.000	-83.3608	1.91987
0.945657E-02	199.711	44.6048	16.2210	-7.83290	-63.9404
199.711	-9.45918E-02	-16.2210	44.6048	63.9404	-7.83290
J,FREQ,NEV=	4	0.002083	8		
14.4138	-51.3341	149.483	-10.6365	16.4418	33.7953
-51.3340	-14.4138	10.6365	149.483	33.7953	-16.4418
21.3918	8.53286	143.108	11.2353	-26.8586	9.71984
8.53289	-21.3917	-11.2353	143.108	9.71984	26.8586
-42.6272	142.657	26.7984	38.1338	-105.184	-21.9194
142.657	42.6273	-38.1338	26.7984	-21.9194	105.184
-149.754	40.5967	-5.46432	-29.3135	25.9251	111.219
40.5967	149.754	29.3135	-5.46436	111.219	-25.9252
J,FREQ,NEV=	5	0.002778	8		
180.103	-11.5904	-35.8411	-3.63007	2.00776	52.4116
11.5904	180.103	3.63007	-35.8411	52.4117	-2.00776
156.216	3.03657	32.1762	20.9054	-23.2681	28.5791
-3.03657	156.216	-20.9054	32.1762	28.5792	23.2681
10.2368	51.4161	184.244	-7.13644	-51.9516	-10.4781
-51.4161	10.2368	7.13644	184.244	-10.4781	51.9515
-83.3741	-23.5440	25.5388	-82.1123	4.10584	131.156
23.5440	-83.3741	82.1123	25.5388	131.156	-4.10585
J,FREQ,NEV=	6	0.003472	8		
-5.48404	160.091	-57.8400	-3.77154	-14.0609	34.0385
					-4.9994

HPDPRE.OUT

160.091	5.48404	-3.77153	57.8400	-34.0385	-14.0609	-20.60`
23.4277	137.476	11.9343	-20.8735	10.8303	38.8174	0.4802`
137.476	-23.4277	-20.8735	-11.9343	-38.8174	10.8303	24.90`
82.4823	23.3708	111.066	-3.51946	90.5736	7.15957	2.417`
23.3708	-82.4823	-3.51946	-111.066	-7.15957	90.5735	-9.299`
-24.8419	-44.3530	-13.5737	147.438	15.4941	79.7330	3.113`
-44.3530	24.8419	147.438	13.5737	-79.7330	15.4941	1.008`
J,FREQ,NEV=	7	0.004167	8			
144.176	-12.5591	-44.2529	-9.14233	-33.7762	37.1908	-23.54`
12.5591	144.176	-9.14234	44.2529	-37.1908	-33.7762	2.789`
131.260	43.6789	-4.92428	-41.2472	16.6703	41.6247	26.02`
-43.6789	131.260	-41.2472	4.92429	-41.6247	16.6703	1.789`
35.2866	128.815	54.0263	-36.0846	70.2409	-14.4290	-14.85`
-128.815	35.2866	-36.0846	-54.0263	14.4290	70.2409	-4.153`
4.93714	9.56558	6.47149	130.344	39.8647	53.7171	-.3245`
-9.56558	4.93714	130.344	-6.47148	-53.7171	39.8647	-1.222`
J,FREQ,NEV=	8	0.004861	8			
143.859	-13.3216	1.29894	-58.4085	-45.6609	25.0165	2.916`
13.3216	143.859	58.4085	1.29894	-25.0165	-45.6608	27.09`
172.648	10.1587	-27.6910	-2.82933	0.661633	35.5563	3.744`
-10.1587	172.648	2.82933	-27.6910	-35.5563	0.661630	-28.62`
95.2954	79.1763	21.0056	34.9377	78.3283	3.66097	-5.823`
-79.1763	95.2954	-84.9377	21.0055	-3.66097	78.3283	14.18`
-37.8281	-20.4111	123.839	9.68228	-8.93960	81.6123	-8.797`
20.4111	-37.8280	-9.68229	123.839	-81.6123	-8.93960	-3.567`
J,FREQ,NEV=	9	0.005556	8			
125.760	-4.80646	-31.1993	-25.1679	-43.3239	17.2289	22.84`
4.80645	125.760	-25.1679	31.1993	-17.2239	-43.3238	-2.661`
144.560	12.8455	4.74319	-8.45751	-.545800	37.9353	-23.38`
-12.8455	144.560	-8.45751	-4.74320	-37.9353	-.545805	-3.332`
69.5360	61.8444	9.91634	83.6979	68.2368	7.93075	10.36`
-61.8444	69.5360	83.6979	-9.91634	-7.93076	68.2368	5.931`
-20.7770	-63.3316	-4.01722	120.848	-53.3714	19.2872	-3.049`
63.3316	-20.7770	120.848	4.01722	-19.2872	-53.3714	0.9666`
J,FREQ,NEV=	10	0.006250	8			
105.413	-23.8628	0.854512	-63.0650	30.4641	6.03178	-2.133`
23.8628	105.413	-63.0650	-.854496	-6.03177	30.4641	-17.93`
119.321	16.6125	21.4644	-25.4528	16.0882	-19.8763	-2.350`
-16.6125	119.321	-25.4528	-21.4644	19.8763	16.0882	22.42`
98.8449	81.4289	-7.37545	54.4003	-17.1749	-46.8148	3.141`
-81.4289	98.8449	54.4004	7.37544	46.8148	-17.1749	-10.02`
4.20515	-77.3907	-56.3917	49.4473	58.6371	-7.83429	0.9263`
77.3907	4.20515	49.4473	56.3916	7.83429	58.6371	2.281`
J,FREQ,NEV=	11	0.006944	8			
-17.1742	114.191	-38.6634	-3.73445	41.0593	-39.8662	1.508`
114.191	17.1742	3.73445	-38.6634	39.8662	41.0593	17.63`
23.7312	121.898	-3.73439	4.92992	3.62539	-53.0218	1.487`
121.898	-23.7311	-4.92994	-3.73439	53.0218	3.62539	-19.07`
78.0499	63.1010	104.026	-12.9498	-41.9367	-13.5866	-2.292`
63.1010	-78.0499	12.9498	104.026	13.5866	-41.9367	7.965`
-111.698	-16.6736	82.9019	-4.63692	71.0027	-2.98699	0.6400`

APPENDIX C: SCREEN LISTING DEVELOPED DURING HPDSIM

HPDSIM.SCR

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 1
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 2
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 3
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 4
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 5
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 6
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 7
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 8
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 9
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 10
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1956, MO: 9

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 11
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1956, MO: 10

YEAR BEING SIMULATED= 1956
MONTH BEING READ IN = 12
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1956, MO: 11

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 1

HPDSIM.SCR

(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1956, MO: 12

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 2
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1957, MO: 1

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 3
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1957, MO: 2

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 4
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1957, MO: 3

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 5
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1957, MO: 4

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 6
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1957, MO: 5

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 7
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1957, MO: 6

YEAR BEING SIMULATED= 1957
MONTH BEING READ IN = 8
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING

HPDSIM.SCR

WRITING SIM. TIME SERIES, YR: 1957, MO: 7

YEAR BEING SIMULATED= 1957
 MONTH BEING READ IN = 9
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1957, MO: 8

YEAR BEING SIMULATED= 1957
 MONTH BEING READ IN = 10
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1957, MO: 9

YEAR BEING SIMULATED= 1957
 MONTH BEING READ IN = 11
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1957, MO: 10

YEAR BEING SIMULATED= 1957
 MONTH BEING READ IN = 12
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1957, MO: 11

YEAR BEING SIMULATED= 1958
 MONTH BEING READ IN = 1
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1957, MO: 12

YEAR BEING SIMULATED= 1958
 MONTH BEING READ IN = 2
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1958, MO: 1

YEAR BEING SIMULATED= 1958
 MONTH BEING READ IN = 3
 (SIMULATION REQUESTED FOR THIS MONTH)
 NORMAL SCORES CREATED
 BACKWARD INTERPOLATION FINISHED
 SIMULATION STORED IN WHEEL ARRAY
 RAMPING

WRITING SIM. TIME SERIES, YR: 1958, MO: 2

YEAR BEING SIMULATED= 1958
 MONTH BEING READ IN = 4
 (SIMULATION REQUESTED FOR THIS MONTH)

HPDSIM.SCR

NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1958, MO: 3

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 5
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1958, MO: 4

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 6
(SIMULATION REQUESTED FOR THIS MONTH)
NORMAL SCORES CREATED
BACKWARD INTERPOLATION FINISHED
SIMULATION STORED IN WHEEL ARRAY
RAMPING
WRITING SIM. TIME SERIES, YR: 1958, MO: 5

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 7
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 8
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 9
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 10
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 11
(NO SIMULATION REQUESTED)

YEAR BEING SIMULATED= 1958
MONTH BEING READ IN = 12
(NO SIMULATION REQUESTED)
WRITING SIM. TIME SERIES, YR: 1958, MO: 6

APPENDIX D: SAMPLE OUTPUT AND TABLE OUTPUT
FROM SIMULATIONS BY HPDSIM

HPDSIM.OUT

FOR WES VISIT. 10/30/89

NYR, NNY, NMO=	20	20	12																
IYEARS=	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MONTHS=	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CUTOFF=		0.083333			0.083333			0.083333			0.083333			0.083333			0.083333		
CUTOFF=		0.083333			0.083333			0.083333			0.083333			0.083333			0.083333		
CUTOFF=		0.083333			0.083333														
IJY=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
IJM=	1	2	3	4	5	6	7	8	9	10	11	12							

195609	100		0.70000		5.00000		113.27737
195609	103		0.70000		5.00000		114.03854
195609	106		0.70000		5.00000		116.59304
195609	109		0.70000		5.00000		120.14225
195609	112		0.70000		5.00000		123.14068
195609	115		0.70000		5.00000		124.64937
195609	118		0.70000		5.00000		121.49031
195609	121		0.80000		5.53105		118.15048
195609	200		0.90000		5.60000		115.40046
195609	203		1.00000		5.60000		112.00829
195609	206		1.20000		5.60000		109.42976
195609	209		1.30000		6.20000		105.58495
195609	215		1.40000		6.20000		95.13261
195609	218		1.30000		5.60000		93.15313
195609	221		1.30000		5.60000		100.19435
195609	300		1.30000		5.60000		117.00001
195609	303		1.30000		5.60000		134.50224
195609	306		1.40000		5.60000		137.47356
195609	309		1.40000		5.60000		129.99557
195609	312		1.45985		5.60000		118.57941
195609	315		1.40000		5.60000		108.89072
195609	318		1.30000		5.60000		98.28101
195609	321		1.20000		5.60000		92.67716
195609	400		1.20000		5.60000		94.01678
195609	403		1.20000		5.60000		100.00198
195609	406		1.20000		5.60000		105.55058
195609	409		1.20000		5.60000		109.40114
195609	412		1.21875		5.60000		111.01221
195609	415		1.30000		6.20000		110.27739
195609	418		1.30000		6.20000		108.05641
195609	421		1.30000		6.70000		106.31438
195609	500		1.30000		6.70000		105.42564
195609	503		1.40000		7.10000		105.84527
195609	506		1.50000		6.70000		106.73637
195609	509		1.70000		6.70000		108.06607
195609	512		1.77409		6.70000		109.76466
195609	515		1.80000		6.20000		111.60548
195609	518		1.70000		6.20000		116.05808
195609	521		1.50000		6.20000		122.40803
195609	600		1.40000		6.20000		126.56181
195609	603		1.30000		6.70000		129.75008
195609	606		1.20000		7.10000		131.62173

HPDSIM.OUT

195609 609	1.20000	7.10000	131.41963
195609 612	1.30000	7.26311	129.01366
195609 615	1.30000	7.10000	123.07030
195609 618	1.30000	6.70000	110.52518
195609 621	1.30000	6.20000	101.51926
195609 700	1.30000	6.20000	98.75435
195609 703	1.20000	5.60000	104.63429
195609 706	1.20000	5.60000	113.87487
195609 709	1.10000	5.60000	124.92681
195609 712	1.10000	5.60000	129.58037
195609 715	1.10000	5.60000	132.02078
195609 718	1.10000	5.00000	133.88715
195609 721	1.10000	5.00000	136.30389
195609 800	1.20000	5.00000	139.26863
195609 803	1.20000	5.00000	150.79036
195609 806	1.20000	5.00000	171.44414
195609 809	1.20000	5.00000	186.96484
195609 812	1.10000	5.00000	195.82111
195609 815	1.00000	5.00000	194.43350
195609 818	0.90000	5.00000	188.20230
195609 821	0.90000	5.00000	181.41382
195609 900	0.90000	5.00000	177.79990
195609 903	0.90000	5.00000	176.04645
195609 906	0.90000	5.00000	176.12930
195609 909	0.90000	5.00000	176.95799
195609 912	0.80000	5.00000	178.99467
195609 915	0.80000	4.50000	180.00000
195609 918	0.70000	4.50000	181.00227
195609 921	0.60000	4.50000	181.49889
1956091000	0.60000	5.00000	182.01271
1956091003	0.70000	5.00000	181.83344
1956091006	0.80000	5.00000	181.01132
1956091009	0.90000	5.00000	180.00000
1956091012	1.00000	4.50000	178.97528
1956091015	1.20000	4.50000	176.94470
1956091018	1.30000	4.50000	175.90515
1956091021	1.30000	4.50000	175.86948
1956091100	1.20000	4.50000	176.86748
1956091103	1.20000	5.00000	173.94572
1956091106	1.20000	5.00000	180.00000
1956091109	1.30000	5.60000	179.41515
1956091112	1.40000	6.20000	176.73889
1956091115	1.50000	6.20000	172.85391
1956091118	1.70000	6.20000	168.01492
1956091121	1.80000	5.60000	164.91008
1956091200	1.80000	5.60000	166.06068
1956091203	1.80000	5.60000	173.33278
1956091206	1.70000	5.60000	181.41985
1956091209	1.50000	5.60000	193.71585
1956091212	1.40000	5.60000	202.98055
1956091215	1.20000	5.60000	208.58812

HPDSIM.OUT

1956091218	1.10000	5.60000	211.83379
1956091221	1.00000	5.00000	213.65858
1956091300	0.90000	5.00000	215.03752
1956091303	0.90000	5.00000	217.22058
1956091306	0.90000	5.00000	221.31921
1956091309	0.90000	4.50000	226.83105
1956091312	0.90000	4.50000	235.47476
1956091315	0.90000	5.00000	243.44914
1956091318	0.90000	5.00000	247.15381
1956091321	1.00000	5.00000	245.13658
1956091400	1.00000	5.00000	235.99905
1956091403	1.10000	5.00000	216.38318
1956091406	1.20000	5.60000	202.98611
1956091409	1.20000	5.60000	194.71094
1956091412	1.30000	5.60000	188.98627
1956091415	1.30000	5.60000	181.14311
1956091418	1.20000	6.20000	171.54427
1956091421	1.20000	6.20000	154.33618
1956091500	1.20000	6.20000	134.92424
1956091503	1.20000	6.20000	119.72636
1956091506	1.20000	6.20000	113.88950
1956091509	1.20000	6.20000	112.50871
1956091512	1.20000	6.20000	113.89182
1956091515	1.10000	5.60000	114.66509
1956091518	1.00000	5.00000	114.10305
1956091521	0.90000	5.00000	112.46371
1956091600	0.90000	4.50000	109.17908
1956091603	0.80000	4.50000	106.67773
1956091606	0.80000	4.50000	107.29001
1956091609	0.80000	4.50000	110.06353
1956091612	0.80000	5.00000	111.98402
1956091615	0.80000	5.00000	112.30267
1956091618	0.70000	4.50000	110.30171
1956091621	0.60000	4.50000	103.63221
1956091700	0.60000	4.50000	92.27570
1956091703	0.50000	4.50000	74.24382
1956091706	0.54146	4.50000	63.27306
1956091709	0.60000	4.50000	59.39151
1956091712	0.70000	4.50000	57.82533
1956091715	0.80000	5.00000	55.95670
1956091718	0.90000	5.00000	54.44638
1956091721	0.90000	5.00000	53.53473
1956091800	0.90000	5.00000	53.68654
1956091803	0.90000	5.00000	55.72710
1956091806	0.90000	5.00000	60.45648
1956091809	0.90000	5.60000	68.80946
1956091812	0.90000	6.20000	83.55972
1956091815	0.90000	6.20000	90.25688
1956091818	0.90000	6.20000	89.52858
1956091821	0.90000	6.20000	81.17751
1956091900	0.90000	6.20000	68.14755

HPDSIM.OUT

1956091903	0.90000	5.60000	61.35139
1956091906	0.90000	5.60000	59.06908
1956091909	0.90000	6.20000	59.91687
1956091912	1.00000	6.70000	65.31298
1956091915	1.10000	7.70000	80.15697
1956091918	1.10000	8.30000	94.48664
1956091921	1.19073	8.46252	102.54816
1956092000	1.10000	8.30000	106.36530
1956092003	1.10000	7.70000	108.98872
1956092006	1.10000	7.10000	110.93769
1956092009	1.10000	6.70000	113.10519
1956092012	1.10000	6.70000	114.68254
1956092015	1.00000	7.10000	117.14014
1956092018	1.00000	7.70000	120.22757
1956092021	1.00000	7.70000	122.49940
1956092100	0.90000	7.70000	124.34167
1956092103	0.90000	7.10000	127.36533
1956092106	1.00000	6.26795	128.95450
1956092109	1.00000	6.20000	130.23665
1956092112	1.10000	5.60000	130.68146
1956092115	1.10000	5.00000	130.18292
1956092118	1.00000	5.00000	128.69323
1956092121	0.90000	5.00000	126.05055
1956092200	0.70000	4.50000	122.49184
1956092203	0.60000	4.50000	117.39726
1956092206	0.50000	4.50000	115.53141
1956092209	0.50000	4.00000	118.21554
1956092212	0.50000	3.89052	123.14030
1956092215	0.60000	3.69868	130.52321
1956092218	0.70000	3.87105	140.77469
1956092221	0.72987	4.00000	153.55513
1956092300	0.80000	4.23322	175.32532
1956092303	0.80000	4.50000	223.72047
1956092306	0.70000	4.50000	248.52458
1956092309	0.70000	4.50000	251.50821
1956092312	0.70000	4.50000	248.34949
1956092315	0.70000	4.00000	243.78043
1956092318	0.70000	3.81241	237.73955
1956092321	0.60000	3.59418	233.61884
1956092400	0.60000	3.60000	232.46693
1956092403	0.60000	3.86975	232.27081
1956092406	0.60000	4.02669	233.08113
1956092409	0.70000	4.50000	234.62604
1956092412	0.70000	4.50000	236.57376
1956092415	0.80000	5.00000	238.62755
1956092418	0.90000	5.00000	241.96198
1956092421	1.00000	5.00000	245.69299
1956092500	1.00000	5.00000	248.98045
1956092503	1.00000	5.00000	251.77727
1956092506	1.00000	5.00000	252.48395
1956092509	1.00000	5.00000	241.64134

HPDSIM.OUT

1956092512	1.00000	5.00000	193.38211
1956092515	1.10000	5.00000	143.91312
1956092518	1.20000	5.00000	128.12947
1956092521	1.40000	5.60000	125.93976
1956092600	1.50000	5.60000	126.31360
1956092603	1.56033	5.57088	129.00328
1956092606	1.52726	5.53980	134.20052
1956092609	1.49996	5.96277	142.96313
1956092612	1.49050	5.97930	154.15851
1956092615	1.47214	5.92535	158.42711
1956092618	1.43658	5.87236	156.98851
1956092621	1.37196	5.82047	154.38594
1956092700	1.28397	5.33613	152.56271
1956092703	1.18660	4.87795	152.49086
1956092706	1.11066	4.85843	152.75754
1956092709	0.97397	4.50324	155.10767
1956092712	0.97087	4.49382	158.44809
1956092715	0.90602	4.79894	161.68088
1956092718	0.88669	4.77925	162.42569
1956092721	0.88484	5.10989	159.43446
1956092800	0.93996	5.08065	154.49979
1956092803	0.93671	5.05233	149.79715
1956092806	0.98666	5.12813	146.78482
1956092809	1.03325	5.10477	145.81468
1956092812	1.02796	5.37916	146.78078
1956092815	1.07062	5.23962	154.08023
1956092818	1.11067	5.44137	171.00379
1956092821	1.14834	5.58527	185.56042
1956092900	1.14123	5.80601	196.99275
1956092903	1.17626	5.37957	207.09972
1956092906	1.16911	5.59659	225.97812
1956092909	1.16239	5.49932	267.19836
1956092912	1.11692	5.37910	301.80780
1956092915	1.07286	5.15954	318.38794
1956092918	1.01802	5.14115	327.49789
1956092921	0.97761	5.28586	317.56952
1956093000	0.93330	5.05198	212.00726
1956093003	0.90000	5.21347	180.25500
1956093006	0.89375	4.99775	181.03079
1956093009	0.89027	4.99994	185.36816
1956093012	0.89415	4.83055	188.35381
1956093015	0.93073	4.83804	189.65738
1956093018	0.93701	4.84634	189.48723
1956093021	0.97310	4.85549	184.88297
195610 100	0.98003	4.86551	165.56885
195610 103	0.95439	4.87643	132.64728
195610 106	0.92649	4.88828	113.21537
195610 109	0.92883	4.69231	110.25206
195610 112	0.90370	4.70209	113.47340
195610 115	0.91069	4.71273	118.15321
195610 118	0.90908	4.64567	122.56877

HPDSIM.TAB

FOR WES VISIT. 10/30/89

NYR,NNY,NMO= 20 20 12
 IYEARS= 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 MONTHS= 1 1 1 1 1 1 1 1 1 1 1 1
 CUTOFF= 0.083333 0.083333 0.083333 0.083333 0.083333
 CUTOFF= 0.083333 0.083333 0.083333 0.083333 0.083333
 CUTOFF= 0.083333 0.083333
 IJY= 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
 IJM= 1 2 3 4 5 6 7 8 9 10 11 12

SIMULATED (H,T,D) TIME SERIES, TITLE: FOR WES VISIT. 10/30/89

(YR/MO)	DA/HR:	100	103	106	109	112	115	118	121	200	203
195609	HEIGHT:	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.80	0.90	1.00
	PERIOD:	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.5	5.6	5.6
	DIRECTION:	113.	114.	117.	120.	123.	125.	121.	118.	115.	112.
	SIN DIRECTION:	0.91	0.87	0.75	0.64	0.55	0.49	0.54	0.64	0.75	0.88
	COS DIRECTION:	-0.39	-0.39	-0.38	-0.37	-0.36	-0.34	-0.33	-0.34	-0.36	-0.36

(YR/MO)	DA/HR:	206	209	212	215	218	221	300	303	306	309
195609	HEIGHT:	1.20	1.30	1.40	1.40	1.30	1.30	1.30	1.30	1.40	1.40
	PERIOD:	5.6	6.2	6.2	6.2	5.6	5.6	5.6	5.6	5.6	5.6
	DIRECTION:	109.	106.	101.	95.	93.	100.	117.	135.	137.	130.
	SIN DIRECTION:	0.92	0.93	0.92	0.85	0.71	0.58	0.43	0.33	0.36	0.48
	COS DIRECTION:	-0.33	-0.26	-0.17	-0.08	-0.04	-0.10	-0.22	-0.33	-0.39	-0.41

(YR/MO)	DA/HR:	312	315	318	321	400	403	406	409	412	415
195609	HEIGHT:	1.46	1.40	1.30	1.20	1.20	1.20	1.20	1.20	1.22	1.30
	PERIOD:	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	6.2
	DIRECTION:	119.	109.	98.	93.	94.	100.	106.	109.	111.	110.
	SIN DIRECTION:	0.63	0.76	0.90	0.95	0.97	0.98	0.99	0.99	1.00	1.00
	COS DIRECTION:	-0.34	-0.26	-0.13	-0.04	-0.07	-0.17	-0.28	-0.35	-0.38	-0.37

(YR/MO)	DA/HR:	418	421	500	503	506	509	512	515	518	521
195609	HEIGHT:	1.30	1.30	1.30	1.40	1.50	1.70	1.77	1.80	1.70	1.50
	PERIOD:	6.2	6.7	6.7	7.1	6.7	6.7	6.7	6.2	6.2	6.2
	DIRECTION:	108.	106.	105.	106.	107.	108.	110.	112.	116.	122.
	SIN DIRECTION:	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99
	COS DIRECTION:	-0.33	-0.29	-0.28	-0.28	-0.30	-0.33	-0.36	-0.39	-0.49	-0.63

(YR/MO)	DA/HR:	600	603	606	609	612	615	618	621	700	703
195609	HEIGHT:	1.40	1.30	1.20	1.20	1.30	1.30	1.30	1.30	1.30	1.20
	PERIOD:	6.2	6.7	7.1	7.1	7.3	7.1	6.7	6.2	6.2	5.6
	DIRECTION:	127.	130.	132.	131.	129.	123.	111.	102.	99.	105.
	SIN DIRECTION:	0.99	0.98	0.97	0.96	0.95	0.93	0.91	0.90	0.90	0.91
	COS DIRECTION:	-0.73	-0.82	-0.87	-0.85	-0.77	-0.60	-0.34	-0.18	-0.14	-0.24

(YR/MO)	DA/HR:	706	709	712	715	718	721	800	803	806	809
195609	HEIGHT:	1.20	1.10	1.10	1.10	1.10	1.10	1.20	1.20	1.20	1.20
	PERIOD:	5.6	5.6	5.6	5.6	5.0	5.0	5.0	5.0	5.0	5.0

HPDSIM.TAB

	DIRECTION:	114.	125.	130.	132.	134.	136.	139.	151.	171.	187.
	SIN DIRECTION:	0.92	0.92	0.91	0.89	0.81	0.69	0.55	0.32	0.08	-0.06
	COS DIRECTION:	-0.41	-0.64	-0.76	-0.80	-0.78	-0.73	-0.64	-0.57	-0.51	-0.52
(YR/MO)	DA/HR:	812	815	818	821	900	903	906	909	912	915
195609	HEIGHT:	1.10	1.00	0.90	0.90	0.90	0.90	0.90	0.90	0.80	0.80
	PERIOD:	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	4.5
	DIRECTION:	196.	194.	188.	181.	178.	176.	176.	177.	179.	180.
	SIN DIRECTION:	-0.15	-0.15	-0.09	-0.02	0.03	0.06	0.06	0.05	0.02	0.00
	COS DIRECTION:	-0.54	-0.59	-0.64	-0.71	-0.79	-0.88	-0.95	-0.93	-0.99	-1.00
(YR/MO)	DA/HR:	918	921	1000	1003	1006	1009	1012	1015	1018	1021
195609	HEIGHT:	0.70	0.60	0.60	0.70	0.80	0.90	1.00	1.20	1.30	1.30
	PERIOD:	4.5	4.5	5.0	5.0	5.0	5.0	4.5	4.5	4.5	4.5
	DIRECTION:	181.	181.	182.	182.	181.	180.	179.	177.	176.	176.
	SIN DIRECTION:	-0.02	-0.03	-0.03	-0.03	-0.02	0.00	0.02	0.05	0.07	0.07
	COS DIRECTION:	-1.00	-1.00	-0.99	-0.99	-0.99	-0.99	-0.98	-0.98	-0.97	-0.97
(YR/MO)	DA/HR:	1100	1103	1106	1109	1112	1115	1118	1121	1200	1203
195609	HEIGHT:	1.20	1.20	1.20	1.30	1.40	1.50	1.70	1.80	1.80	1.80
	PERIOD:	4.5	5.0	5.0	5.6	6.2	6.2	6.2	5.6	5.6	5.6
	DIRECTION:	177.	179.	180.	179.	177.	173.	168.	165.	166.	173.
	SIN DIRECTION:	0.05	0.02	0.00	0.01	0.05	0.11	0.18	0.22	0.20	0.09
	COS DIRECTION:	-0.96	-0.95	-0.94	-0.93	-0.92	-0.90	-0.87	-0.83	-0.81	-0.81
(YR/MO)	DA/HR:	1206	1209	1212	1215	1213	1221	1300	1303	1306	1309
195609	HEIGHT:	1.70	1.50	1.40	1.20	1.10	1.00	0.90	0.90	0.90	0.90
	PERIOD:	5.6	5.6	5.6	5.6	5.6	5.0	5.0	5.0	5.0	4.5
	DIRECTION:	181.	194.	203.	209.	212.	214.	215.	217.	221.	227.
	SIN DIRECTION:	-0.02	-0.21	-0.37	-0.47	-0.53	-0.55	-0.55	-0.57	-0.62	-0.67
	COS DIRECTION:	-0.83	-0.85	-0.87	-0.87	-0.85	-0.82	-0.79	-0.75	-0.70	-0.62
(YR/MO)	DA/HR:	1312	1315	1318	1321	1400	1403	1406	1409	1412	1415
195609	HEIGHT:	0.90	0.90	0.90	1.00	1.00	1.10	1.20	1.20	1.30	1.30
	PERIOD:	4.5	5.0	5.0	5.0	5.0	5.0	5.6	5.6	5.6	5.6
	DIRECTION:	235.	243.	247.	245.	236.	216.	203.	195.	189.	181.
	SIN DIRECTION:	-0.70	-0.72	-0.69	-0.63	-0.53	-0.42	-0.31	-0.22	-0.14	-0.02
	COS DIRECTION:	-0.48	-0.36	-0.29	-0.29	-0.36	-0.57	-0.73	-0.84	-0.88	-0.87
(YR/MO)	DA/HR:	1418	1421	1500	1503	1506	1509	1512	1515	1518	1521
195609	HEIGHT:	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.10	1.00	0.90
	PERIOD:	6.2	6.2	6.2	6.2	6.2	6.2	6.2	5.6	5.0	5.0
	DIRECTION:	172.	154.	135.	120.	114.	113.	114.	115.	114.	112.
	SIN DIRECTION:	0.12	0.33	0.53	0.66	0.74	0.80	0.81	0.78	0.74	0.71
	COS DIRECTION:	-0.81	-0.69	-0.52	-0.37	-0.33	-0.33	-0.36	-0.36	-0.33	-0.29
(YR/MO)	DA/HR:	1600	1603	1606	1609	1612	1615	1618	1621	1700	1703
195609	HEIGHT:	0.90	0.80	0.80	0.80	0.80	0.80	0.70	0.60	0.60	0.50
	PERIOD:	4.5	4.5	4.5	4.5	5.0	5.0	4.5	4.5	4.5	4.5
	DIRECTION:	109.	107.	107.	110.	112.	112.	110.	104.	92.	74.
	SIN DIRECTION:	0.69	0.71	0.73	0.75	0.78	0.79	0.79	0.79	0.81	0.85

HPDSIM.TAB

COS DIRECTION:		-0.24	-0.21	-0.23	-0.28	-0.32	-0.33	-0.29	-0.19	-0.03	0.24
----------------	--	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

(YR/MO)	DA/HR:	1706	1709	1712	1715	1718	1721	1800	1803	1806	1809
195609	HEIGHT:	0.54	0.60	0.70	0.80	0.90	0.90	0.90	0.90	0.90	0.90
	PERIOD:	4.5	4.5	4.5	5.0	5.0	5.0	5.0	5.0	5.0	5.6
	DIRECTION:	63.	59.	58.	56.	54.	54.	54.	56.	60.	69.
	SIN DIRECTION:	0.90	0.94	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	COS DIRECTION:	0.45	0.56	0.61	0.67	0.71	0.74	0.73	0.68	0.57	0.39

(YR/MO)	DA/HR:	1812	1815	1818	1821	1900	1903	1906	1909	1912	1915
195609	HEIGHT:	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	1.00	1.10
	PERIOD:	6.2	6.2	6.2	6.2	6.2	5.6	5.6	6.2	6.7	7.7
	DIRECTION:	84.	90.	90.	81.	68.	61.	59.	60.	65.	80.
	SIN DIRECTION:	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	COS DIRECTION:	0.11	0.00	0.01	0.16	0.40	0.55	0.60	0.58	0.45	0.17

(YR/MO)	DA/HR:	1918	1921	2000	2003	2006	2009	2012	2015	2018	2021
195609	HEIGHT:	1.10	1.19	1.10	1.10	1.10	1.10	1.10	1.00	1.00	1.00
	PERIOD:	3.3	8.5	8.3	7.7	7.1	6.7	6.7	7.1	7.7	7.7
	DIRECTION:	94.	103.	106.	109.	111.	113.	115.	117.	120.	122.
	SIN DIRECTION:	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.98	0.98	0.97
	COS DIRECTION:	-0.08	-0.22	-0.29	-0.34	-0.38	-0.42	-0.45	-0.50	-0.57	-0.62

(YR/MO)	DA/HR:	2100	2103	2106	2109	2112	2115	2118	2121	2200	2203
195609	HEIGHT:	0.90	0.90	1.00	1.00	1.10	1.10	1.00	0.90	0.70	0.60
	PERIOD:	7.7	1	6.3	6.2	5.6	5.0	5.0	5.0	4.5	4.5
	DIRECTION:	125.		129.	130.	131.	130.	129.	125.	122.	117.
	SIN DIRECTION:	0.96		0.96	0.97	0.98	0.98	0.98	0.98	0.97	0.94
	COS DIRECTION:	-0.67	-0.73	-0.78	-0.82	-0.84	-0.83	-0.79	-0.71	-0.62	-0.49

(YR/MO)	DA/HR:	2206	2209	2212	2215	2218	2221	2300	2303	2306	2309
195609	HEIGHT:	0.50	0.50	0.50	0.60	0.70	0.73	0.80	0.80	0.70	0.70
	PERIOD:	4.5	4.0	3.9	3.7	3.9	4.0	4.2	4.5	4.5	4.5
	DIRECTION:	115.	118.	123.	131.	141.	154.	175.	224.	249.	252.
	SIN DIRECTION:	0.86	0.70	0.57	0.43	0.28	0.14	0.02	-0.14	-0.35	-0.57
	COS DIRECTION:	-0.41	-0.38	-0.37	-0.36	-0.34	-0.29	-0.21	-0.15	-0.14	-0.19

(YR/MO)	DA/HR:	2312	2315	2318	2321	2400	2403	2406	2409	2412	2415
195609	HEIGHT:	0.70	0.70	0.70	0.60	0.60	0.60	0.60	0.70	0.70	0.80
	PERIOD:	4.5	4.0	3.8	3.6	3.6	3.9	4.0	4.5	4.5	5.0
	DIRECTION:	248.	244.	238.	234.	232.	232.	233.	235.	237.	239.
	SIN DIRECTION:	-0.73	-0.79	-0.82	-0.84	-0.84	-0.84	-0.85	-0.87	-0.88	-0.89
	COS DIRECTION:	-0.29	-0.39	-0.52	-0.62	-0.64	-0.65	-0.64	-0.62	-0.50	-0.54

(YR/MO)	DA/HR:	2418	2421	2500	2503	2506	2509	2512	2515	2518	2521
195609	HEIGHT:	0.90	1.00	1.00	1.00	1.00	1.00	1.00	1.10	1.20	1.40
	PERIOD:	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.6
	DIRECTION:	242.	246.	249.	252.	252.	242.	193.	144.	128.	126.
	SIN DIRECTION:	-0.88	-0.86	-0.80	-0.74	-0.57	-0.32	-0.05	0.20	0.42	0.54
	COS DIRECTION:	-0.47	-0.39	-0.31	-0.24	-0.18	-0.17	-0.20	-0.28	-0.33	-0.39

HPDSIM.TAB

(YR/MO)	DA/HR:	2600	2603	2606	2609	2612	2615	2618	2621	2700	2703
195609	HEIGHT:	1.50	1.56	1.53	1.50	1.49	1.47	1.44	1.37	1.28	1.19
	PERIOD:	5.6	5.6	5.5	6.0	6.0	5.9	5.9	5.8	5.3	4.9
	DIRECTION:	126.	129.	134.	143.	154.	153.	157.	154.	153.	152.
	SIN DIRECTION:	0.58	0.55	0.48	0.38	0.29	0.28	0.35	0.44	0.51	0.52
	COS DIRECTION:	-0.43	-0.45	-0.47	-0.51	-0.60	-0.71	-0.82	-0.92	-0.98	-0.99

(YR/MO)	DA/HR:	2706	2709	2712	2715	2718	2721	2800	2803	2806	2809
195609	HEIGHT:	1.11	0.97	0.97	0.91	0.89	0.88	0.94	0.94	0.99	1.03
	PERIOD:	4.9	4.5	4.5	4.8	4.8	5.1	5.1	5.1	5.1	5.1
	DIRECTION:	153.	155.	158.	162.	162.	159.	154.	150.	147.	146.
	SIN DIRECTION:	0.51	0.46	0.40	0.34	0.34	0.40	0.48	0.54	0.54	0.50
	COS DIRECTION:	-0.98	-0.99	-1.01	-1.04	-1.06	-1.06	-1.01	-0.93	-0.83	-0.74

(YR/MO)	DA/HR:	2812	2815	2818	2821	2900	2903	2906	2909	2912	2915
195609	HEIGHT:	1.03	1.07	1.11	1.15	1.14	1.18	1.17	1.16	1.12	1.07
	PERIOD:	5.4	5.2	5.4	5.6	5.8	5.9	5.6	5.5	5.4	5.2
	DIRECTION:	147.	154.	171.	186.	197.	207.	226.	267.	302.	310.
	SIN DIRECTION:	0.43	0.30	0.10	-0.06	-0.19	-0.27	-0.31	-0.31	-0.31	-0.29
	COS DIRECTION:	-0.65	-0.61	-0.63	-0.65	-0.63	-0.53	-0.29	-0.02	0.19	0.33

(YR/MO)	DA/HR:	2918	2921	3000	3003	3006	3009	3012	3015	3018	3021
195609	HEIGHT:	1.02	0.98	0.94	0.90	0.89	0.89	0.89	0.93	0.94	0.97
	PERIOD:	5.1	5.3	5.1	5.2	5.0	5.0	4.8	4.8	4.8	4.9
	DIRECTION:	327.	318.	212.	180.	181.	185.	188.	190.	189.	185.
	SIN DIRECTION:	-0.22	-0.14	-0.06	0.00	-0.01	-0.09	-0.13	-0.15	-0.13	-0.06
	COS DIRECTION:	0.34	0.15	-0.10	-0.45	-0.68	-0.83	-0.87	-0.86	-0.80	-0.70

SIMULATED (H,T,D) TIME SERIES, TITLE: FOR WES VISIT. 10/30/82

(YR/MO)	DA/HR:	100	103	106	109	112	115	118	121	200	203
195610	HEIGHT:	0.98	0.95	0.93	0.93	0.90	0.91	0.91	0.97	1.01	1.00
	PERIOD:	4.9	4.9	4.9	4.7	4.7	4.7	4.6	4.7	5.1	5.1
	DIRECTION:	166.	133.	113.	110.	113.	113.	123.	127.	135.	143.
	SIN DIRECTION:	0.14	0.45	0.79	0.96	1.03	1.04	0.99	0.87	0.68	0.51
	COS DIRECTION:	-0.56	-0.41	-0.34	-0.36	-0.45	-0.55	-0.63	0.67	-0.68	-0.67

(YR/MO)	DA/HR:	206	209	212	215	218	221	300	303	306	309
195610	HEIGHT:	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	0.98	1.04
	PERIOD:	5.1	4.9	4.9	4.7	4.6	4.4	4.4	4.4	4.4	4.7
	DIRECTION:	153.	165.	180.	194.	203.	206.	205.	204.	203.	204.
	SIN DIRECTION:	0.34	0.17	0.00	-0.14	-0.22	-0.25	-0.24	-0.23	-0.22	-0.24
	COS DIRECTION:	-0.67	-0.64	-0.60	-0.56	-0.53	-0.51	-0.51	-0.52	-0.52	-0.54

(YR/MO)	DA/HR:	312	315	318	321	400	403	406	409	412	415
195610	HEIGHT:	1.02	1.02	0.97	0.98	0.98	0.98	0.98	0.99	0.99	0.99
	PERIOD:	4.7	4.7	4.7	4.4	4.5	4.5	4.5	4.5	4.5	4.6
	DIRECTION:	204.	204.	205.	206.	208.	211.	218.	225.	231.	236.
	SIN DIRECTION:	-0.25	-0.28	-0.30	-0.34	-0.38	-0.44	-0.55	-0.68	-0.80	-0.90
	COS DIRECTION:	-0.57	-0.61	-0.65	-0.69	-0.72	-0.72	-0.71	-0.68	-0.65	-0.60

APPENDIX E: PROGRAM LISTING FOR HPDPRE.FOR

HPDPRE.FOR

```

PROGRAM HPDPRE
C *****
C PROGRAM TO RUN SUBR HPDPRE AND ASSOCIATED CODE
C FOR DECIMAL FILES OF (H,T,D)
C *****

C ***** SOME DEFINITIONS ***
C NYR = NUMBER OF YEARS IN DATA BASE (SOME OF WHICH YOU MAY
C NOT WANT TO USE.
C NNY = NUMBER OF YEARS ACTUALLY TO BE ANALYZED (MAY BE A
C SUBSET OF THE YEARS IN THE DATABASE
C NMO = NUMBER OF MONTHS OUT OF 12 SELECTED FOR ANALYSIS
C IN THIS COMPUTATIONAL RUN
C IYEARS(JY) = 1, IF JY-TH YEAR IN DATABASE SEQUENCE IS TO
C BE ANALYZED
C IYEARS(JY) = 0, IF JY-TH YEAR IS NOT TO BE ANALYZED
C MONTHS(JM) = 1, IF JM-TH MONTH IS TO BE ANALYZED
C MONTHS(JM) = 0, IF JM-TH MONTH IS NOT TO BE ANALYZED
C (BOTH IYEARS(JY) AND MONTHS(JM) ARE SET IN A DATA STATEMENT
C BELOW
C CUTOFF(JM) = FREQUENCY IN CYCLES PER HOUR FOR HIGHEST FREQ.
C TO BE ANALYZED IN SPECTRAL ESTIMATION FOR THE
C JM-TH MONTH. THIS IS ALSO SET IN A DATA
C STATEMENT BELOW. CUTOFF(JM) MAY BE LEFT ZERO
C FOR MONTHS NOT TO BE ANALYZED.
C N = NUMBER OF (H,T,D) VALUES IN BASIC ANALYSIS UNIT, ORDINARILY
C A MONTH UNIT OF 30 DA AND 8 SETS PER DAY, TO GIVE N=240
C NMAX = A MAXIMUM VALUE FOR N, IF YOU WANT TO SET THE
C DIMENSIONS ONCE AND NOT CHANGE THEM THEREAFTER
C NTM = N*NNY
C NTMAX = NMAX*NNY
C NTMX05 = FIRST INTEGER LARGER THAN OR EQUAL TO NTMAX/5
C NSIM = LENGTH OF BASIC DATA SEQUENCE AFTER PADDING WITH
C ZEROS, SET HERE TO 2*N OR 480.
C NSMAX = MAXIMUM VALUE OF NSIM, FOR DIMENSIONING
C MNO2P1 = NSMAX/2+1, IN FORTRAN INTEGER ARITHMETIC
C DT = BASIC TIME INCREMENT, DT = 3 HRS. FOR WIS DATA.
C HSCALE = FACTOR BY WHICH TO DIVIDE WAVE HEIGHT TO PUT IT
C INTO A CONVENIENT RANGE.
C TSCALE = FACTOR FOR DIVISION OF WAVE PERIOD
C (NO SCALING IS PERMITTED ON DIRECTIONS SEQUENCE.)
C ***** END OF DEFINITION SET ***

PARAMETER (NYR=20,NNY=20,NMO=12,N=240,NSIM=480)
PARAMETER (NSMAX=480,NMAX=240,NTM=4800,NTMAX=4800)
PARAMETER (NTMX05=970,MNO2P1=241)
PARAMETER (DT=3.0)
PARAMETER (HSCALE=1.0,TSCALE=1.0)

C ***** DIMENSION STATEMENTS ***

```

HPDPRE. FOR

```

DIMENSION H(NTMAX),T(NTMAX),D(NTMAX),DS(NTMAX),DC(NTMAX)
DIMENSION HO(NTMAX),TO(NTMAX),DSO(NTMAX),DCO(NTMAX)
DIMENSION HH(10),TT(10),DD(N)
DIMENSION SP(4,4,MNO2P1),EM(8,8,MNO2P1),NEV(MNO2P1)
DIMENSION IPOS(NTMAX),B(5),IB(NTMX05),IT(NTMX05)
DIMENSION XR(NSMAX),XI(NSMAX),YR(NSMAX),YI(NSMAX)
DIMENSION EVAL(36),EIVEC(64)
DIMENSION AA(4,2),BB(4,2),CC(4,2),TSO(4,NMAX)
DIMENSION IBND(4,2),ZTS(4,3,2),UTS(4,3,2)
DIMENSION IBOUND(2),ZAB(3,2),UAB(3,2),AAB(2),BAB(2),CAB(2)
DIMENSION LLL(NTMAX)
DIMENSION MONTHS(12),IYEARS(NYR),CUTOFF(12),IJM(12),IJY(NYR)
DIMENSION HY(2928),TY(2928),DY(2928),MOBEG(12)
DIMENSION HYY(NNY,NMO,240),TYY(NNY,NMO,240),DYY(NNY,NMO,240)
COMPLEX VC(4,NSMAX)
CHARACTER TITLE*50

C      ***** DATA STATEMENTS ***
DATA MONTHS/12*1/
DATA CUTOFF/12*0.0833333/
DATA IYEARS/20*1/

C      ***** OPEN INPUT/OUTPUT ***
OPEN (2,FILE='HPDPRE.OUT',STATUS='UNKNOWN')
C      OPEN (3,FILE='HPDPRE.DAT',STATUS='OLD')
OPEN (3,FILE='SOCAL.DAT',STATUS='OLD')
OPEN (4,FILE='HPDPRE.DOC',STATUS='UNKNOWN')

C      ***** DEFINE CONSTANTS ***
BW=0.01666667
FRACT=0.9999
EPS=0.00001
M=NNY
IDOC=1

C      ***** READ INPUT DATA ***
ISY=0
DO 62 JY=1,NYR
    ISY=ISY+IYEARS(JY)
62 CONTINUE
C      ***** READ FIRST LINE OF DATA ***
READ (3,150,END=99) IY,IM,ID,IH,ISTA,ILOC,JLOC,HS,TP,WDM,
&HSEA,TSEA,DSEA,HSWL,TSWL,DSWL,ALFA,VPAR,IVEL,IDIR
150 FORMAT (2X,4I2,3I3,3(F6.1,F6.1,F6.0),F5.3,F7.4,2I4)

C      ++++++
C      ***** FIRST OF THREE "SET CODE GROUPS" WHICH DETERMINE ***
C      ***** WHICH (H,T,D) ARE TO BE ANALYZED. HERE HS, ***
C      ***** TP, AND WDM ARE SELECTED AND EQUATED TO H1, ***

```

HPDPRE.FOR

```

C      ***** T1, AND D1. IF A DIFFERENT TRIPLE IS DESIRED ***
C      ***** THEY SHOULD BE SET TO H1, T1, AND D1.      ***
      H1=HS
      T1=TP
      D1=WDM
C      ***** END OF FIRST "SET CODE GROUP "      ***
C      ++++++
      IYY=IY
      JY=0
      LY=0

C      ***** START LOOP OVER YEARS ***
      64 CONTINUE

      IF (ISY.EQ.0) GO TO 98
      WRITE (*,127) IY
127    FORMAT (1X,'READING DATABASE, IY= ',I5)
C      ***** NO MORE YEARS IN THOSE REMAINING THAT ARE TO ***
C      ***** BE ANALYZED, SO QUIT LINE READ      ***

      JY=JY+1
      L=1
      HY(L)=H1
      TY(L)=T1
      DY(L)=D1
      MOBEG(L)=1
      IMM=1

C      ***** START LOOP OVER 'LINE BY LINE' READ OF DATA. ***
      63 READ (3,150,END=99) IY,IM,ID,IH,ISTA,ILOC,JLOC,HS,TP,WDM,
      & HSEA,TSEA,DSEA,HSWL,TSWL,DSWL,ALFA,VPAR,IVEL,IDIR

C      ***** IF LAST LINE OF DATABASE IS ENCOUNTERED, QUIT READ ***
      IF (IY.EQ.75.AND.IM.EQ.12.AND.ID.EQ.31.AND.IH.EQ.21) THEN
        ISY=0
        GO TO 99
      END IF

      IF (IY.EQ.IYY) THEN
C      ***** LINE READ IS IN SAME YEAR AS LAST LINE. ***
        L=L+1

C      ++++++
C      ***** START OF SECOND "SET CODE GROUP". THE USER ***
C      ***** SHOULD SET HY(L), TY(L), AND DY(L) TO ***
C      ***** WHATEVER VARIABLES FROM THE ABOVE "READ" ***
C      ***** LIST ARE TO BE ANALYZED. HERE, HS,TP, AND ***
        HY(L)=HS
        TY(L)=TP
        DY(L)=WDM

```

HPDPRE.FOR

```

C      ***** END OF SECOND "SET CODE GROUP". ***
C      ++++++
C      IF (IM.NE.IMM) THEN
C          ***** NEW MONTH ENCOUNTERED.  SET MOBEG(IM) EQUAL ***
C          ***** TO THE L VALUE AT WHICH THIS NEW MO. STARTS ***
C          MOBEG(IM)=L
C          IMM=IM
C          END IF
C          ***** GO BACK TO READ ANOTHER LINE ***
C          GO TO 63
C      ELSE
C          ***** LINE READ IS IN A DIFFERENT YEAR FROM LAST. ***
C          ++++++
C          ***** START OF THIRD, AND LAST, "SET CODE GROUP". ***
C          ***** H1, T1, AND D1 SHOULD BE SET TO EQUAL THE ***
C          ***** VARIABLE SELECTED FROM THE "READ" LIST. ***
C          H1=HS
C          T1=TP
C          D1=WDM
C          ***** END OF THIRD "SET CODE GROUP". ***
C          ++++++
C          IYY=IY
C          ISY=ISY-IYEARS(JY)
C          END IF
C          ***** IF 'END OF FILE' WAS FOUND IN READ, PROGRAM JUMPS TO ***
C          ***** HERE. THIS MEANS THE LAST OF TWENTY YEARS HAS ***
C          ***** FINISHED READING. ***
99  CONTINUE

C      ***** AT END OF EACH YEAR, STORE MONTHLY SETS OF DATA ***
C      LM=0
C      IF (IYEARS(JY).NE.0) THEN
C          ***** YEAR JY WAS DESIGNATED FOR ANALYSIS IN IYEARS(JY) ***
C          LY=LY+1
C          IJY(JY)=LY
C          DO 65 JM=1,12
C              IF (MONTHS(JM).NE.0) THEN
C                  ***** MONTH JM WAS DESIGNATED FOR ANALYSIS IN ***
C                  ***** MONTHS(JM). ***
C                  LM=LM+1
C                  IJM(JM)=LM
C                  DO 66 L=1,240
C                      ***** STORE DATA IN COMPACTED YEAR-MONTH LIST ***
C                      HYY(IJY(JY),IJM(JM),L)=HY(MOBEG(JM)+L-1)
C                      TYY(IJY(JY),IJM(JM),L)=TY(MOBEG(JM)+L-1)
C                      DYY(IJY(JY),IJM(JM),L)=DY(MOBEG(JM)+L-1)
66  CONTINUE

```

HPDPRE.FOR

```

        END IF
65      CONTINUE
        END IF

C      ***** RETURN TO READ NEXT YEAR ***
        GO TO 64
98      CONTINUE

C      ***** INPUT TITLE ***
        WRITE (*,128)
128      FORMAT (1X,'WHAT TITLE DO YOU WANT ATTACHED TO THIS ',
&'ANALYSIS?'/
&' (UP TO 50 CHARACTERS MAY BE ENTERED)')
        READ (*,'(A50)') TITLE

C      ***** WRITE OVERALL DEFINING PARAMETERS ***
        WRITE (2,'(A50)') TITLE
        WRITE (2,120) NYR,NNY,NMO
120      FORMAT (1X,'NYR,NNY,NMO= ',3I5)
        WRITE (2,121) (IYEARS(J),J=1,NYR)
121      FORMAT (1X,'IYEARS= ',20I3)
        WRITE (2,122) (MONTHS(J),J=1,12)
122      FORMAT (1X,'MONTHS= ',12I3)
        WRITE (2,123) (CUTOFF(J),J=1,12)
123      FORMAT (1X,'CUTOFF= ',5F12.6)
        WRITE (2,125) (IJY(J),J=1,NYR)
125      FORMAT (1X,'IJY= ',20I3)
        WRITE (2,126) (IJM(J),J=1,12)
126      FORMAT (1X,'IJM= ',20I3)

C      ***** PROCEED WITH A MONTH-BY-MONTH ANALYSIS ***
        DO 67 JM=1,12
          PRINT*, 'JM= ', JM
          PRINT*, 'MONTHS= ', MONTHS(1), MONTHS(2), MONTHS(3), MONTHS(4)
          IF (MONTHS(JM).NE.0) THEN
            WRITE (*,129) JM
129          FORMAT (1X,'PROCEEDING WITH ANALYSIS OF MONTH ',I5)
            L=0
            FCO=CUTOFF(JM)
            DO 1 JY=1,NYR
              IF (IYEARS(JY).NE.0) THEN
                DO 2 J=1,N
                  L=L+1
                  H(L)=HYY(IJY(JY),IJM(JM),J)/HSCALE
                  T(L)=TYY(IJY(JY),IJM(JM),J)/TSCALE
                  D(L)=DYY(IJY(JY),IJM(JM),J)
                2          CONTINUE
              END IF
            1          CONTINUE
          END IF
        END DO

```

HPDPRE.FOR

```

      IF (IDOC.NE.0) THEN
C      ***** WRITE INPUT TIME SERIES ***
      DO 82 L=1,NTM
        LLL(L)=L
      82      CONTINUE
      WRITE (4,204) JM
204      FORMAT (//1X,'LISTING OF INPUT DATA FOR MD.',I4)
      NS04=NTM/10
      NREM=NTM-NS04*10
      DO 81 I=1,NS04
        LSTART=(I-1)*10+1
        LEND=LSTART+9
        DO 90 L=LSTART,LEND
          HH(L-LSTART+1)=H(L)*HSCALE
          TT(L-LSTART+1)=T(L)*TSCALE
          DD(L-LSTART+1)=D(L)
      90      CONTINUE
          WRITE (4,'(1X)')
          WRITE (4,200) (LLL(L),L=LSTART,LEND)
200      FORMAT (1X,'TIME STEP NO.: ',10I7)
          WRITE (4,201) (HH(L),L=1,10)
201      FORMAT (1X,' WAVE HEIGHT: ',10F7.1)
          WRITE (4,202) (TT(L),L=1,10)
202      FORMAT (1X,' WAVE PERIOD: ',10F7.1)
          WRITE (4,203) (DD(L),L=1,10)
203      FORMAT (1X,' DIRECTION: ',10F7.1)
      81      CONTINUE
      IF (NREM.NE.0) THEN
        LSTART=NTM-NREM+1
        LEND=NTM
        DO 91 L=LSTART,LEND
          HH(L-LSTART+1)=H(L)*HSCALE
          TT(L-LSTART+1)=T(L)*TSCALE
          DD(L-LSTART+1)=D(L)
      91      CONTINUE
          WRITE (4,'(1X)')
          WRITE (4,200) (LLL(L),L=LSTART,LEND)
          WRITE (4,201) (HH(L),L=1,NREM)
          WRITE (4,202) (TT(L),L=1,NREM)
          WRITE (4,201) (DD(L),L=1,NREM)
        END IF
      END IF

      CALL HPDPRE(N,NSIM,NSMAX,M,NTM,NTMAX,MND2P1,H,T,D,
&      DS,DC,HO,TO,DSO,DCO,DT,BW,SP,FCO,VC,EM,NEV,IPOS,B,IB,
&      IT,NTMX05,XR,XI,YR,YI,EVAL,EIVC,FRACT,EPS,LLL)
      LL=0
      DO 60 L=1,NTM
        LDM=L-(L/M)*M

```

HPDPRE.FOR

```

        IF (LDM.EQ.0) THEN
            LL=LL+1
            TSO(1,LL)=HO(L)
            TSO(2,LL)=TO(L)
            TSO(3,LL)=DSO(L)
            TSO(4,LL)=DCO(L)
        END IF
60      CONTINUE

        CALL ABC(N,NMAX,TSO,IBND,ZTS,UTS,AA,BB,CC,IBOUND,ZAB,
&        UAB,AAB,BAB,CAB)

C      ***** PRINT RESULTS ***
        PRINT*, 'AT PRINT RESULTS, JM= ', JM
        PRINT*, 'MONTHS= ', MONTHS(1), MONTHS(2), MONTHS(3), MONTHS(4)
        DF=1.0/(NSIM*DT)
        NO2P1=(NSIM/2)+1
        NTOP=FCO*NSIM*DT+1
        IF (NTOP.GT.NO2P1) NTOP=NO2P1

        WRITE (*,130) JM
130      FORMAT (1X, 'STORING PREPROCESS OUTPUT FOR MONTH ', I5)
        WRITE (2,124) JM
124      FORMAT (//1X, 'PREPROCESS OUTPUT FOR MO. ', I3)
        WRITE (2,310) HSCALE,TSCALE
310      FORMAT (1X, 'SCALE FACTORS: ', 2F15.5)
        WRITE (2,110) N,M,NTM,NTOP,NSIM
110      FORMAT (1X, 'N,M,NTM,NTOP,NSIM=', 5J8)
        WRITE (2,111) DT
111      FORMAT (1X, 'DT= ', G15.5)
        WRITE (2,112) BW,FCO,FRACT
112      FORMAT (1X, 'BW,FCO,FRACT= ', 3G15.5)
        WRITE (2,113)
113      FORMAT (1X, 'RANKED ORIGINAL [H(J); J=M,NTM,M]')
        WRITE (2,114) (HO(J), J=M,NTM,M)
114      FORMAT (1X, 5G15.5)
        WRITE (2,115)
115      FORMAT (1X, 'RANKED ORIGINAL [T(J); J=M,NTM,M]')
        WRITE (2,114) (TO(J), J=M,NTM,M)
        WRITE (2,116)
116      FORMAT (1X, 'RANKED ORIGINAL [DS(J); J=M,NTM,M]')
        WRITE (2,114) (DSO(J), J=M,NTM,M)
        WRITE (2,516)
516      FORMAT (1X, 'RANKED ORIGINAL [DC(J); J=M,NTM,M]')
        WRITE (2,114) (DCO(J), J=M,NTM,M)
        DO 3 J=1,NTOP
            FREQ=(J-1)*DF
            WRITE (2,105) J,FREQ,NEV(J)
105      FORMAT (1X, 'J,FREQ,NEV=', I5,F12.6, I5)
            DO 4 I=1,8
                WRITE (2,106) (EM(I,II,J), II=1,NEV(J))

```

HPDPRE.FOR

```

106          FORMAT (1X,8G12.6)
4            CONTINUE
3            CONTINUE
            DO 70 K=1,4
              WRITE (2,300) K, (IBND(K,J),J=1,2)
300          FORMAT (1X,'K,IBND(K,J)= ',3I5)
              WRITE (2,301) K, (AA(K,J),J=1,2)
301          FORMAT (1X,'K,AA(K,J)= ',15,2G15.5)
              WRITE (2,302) K, (BB(K,J),J=1,2)
302          FORMAT (1X,'K,BB(K,J)= ',15,2G15.5)
              WRITE (2,303) K, (CC(K,J),J=1,2)
303          FORMAT (1X,'K,CC(K,J)= ',15,2G15.5)
              DO 72 I=1,3
                WRITE (2,304) K,I, (ZTS(K,I,J),J=1,2)
304          FORMAT (1X,'K,I,ZTS(K,I,J)= ',2I5,2G15.5)
                WRITE (2,305) K,I, (UTS(K,I,J),J=1,2)
305          FORMAT (1X,'K,I,UTS(K,I,J)= ',2I5,2G15.5)
              72          CONTINUE
            70          CONTINUE
          END IF
67 CONTINUE
      STOP
      END

```

C

SUBROUTINE HPDPRE(N,NSIM,NSMAX,M,NTM,NTMAX,MNDZP1,H,T,D,DS,
&DC,HO,TO,DSO,DCO,DT,BW,SP,FCO,VC,EM,NEV,IPOS,B,IB,IT,NTMX05,
&XR,XI,YR,YI,EVAL,EIVC,FRACT,EPS,LLL)

C *****

C PREPROCESS M CONTINUOUS TIME INTERVALS OF DATA ALL WITH
C N TIME INCREMENTS OF LENGTH, DT, TO PREPARE INPUT
C FOR LONG-TERM SYNTHESIS OR SIMULATION OF SIGNIFICANT
C WAVE HEIGHT, PEAK PERIOD, AND DOMINANT DIRECTION OF
C TRAVEL.

C

C M = NUMBER OF CONTINUOUS PIECES OF DATA, EACH OF LENGTH
C N. THE SUBROUTINE ASSUMES THAT THE "PIECES" HAVE
C BEEN STACKED INTO A VECTOR.

C

C N = LENGTH OF EACH INPUT DATA FOR STATIONARY INTERVAL,
C USUALLY ONE MONTH. IF KK MONTHS OF THE SAME TYPE (SAY,
C A COMPLETE WINTER PERIOD) ARE TO BE USED FOR EACH
C PIECE OR SEQUENCE OF DATA, THEN N = NUMBER OF DATA IN
C THE COMBINED KK-MONTHS.

C NSIM = LENGTH OF TIME SERIES TO BE SIMULATED SUBSEQUENTLY.

C THIS SHOULD BE AN EXACT INTEGER MULTIPLE OF N.

C IT MAY BE EQUAL TO N, TWO TIMES N, THREE TIMES N,

C OR SOME OTHER INTEGER MULTIPLE OF N. IN ANY CASE,

C NSIM SHOULD BE SELECTED TO BE REPRESENTABLE IN THE

C FORM (2**I)*(3**J)*(5**K) FOR POSITIVE INTEGERS

C (I,J,K). THIS IS REQUIRED FOR THE FFT SOFTWARE USED.

HPDPRE.FOR

```

C      NSMAX = MAXIMUM VALUE OF NSIM, FOR DIMENSIONING ONLY
C      MNO2P1 = (NSMAX/2)+1, IN FORTRAN INTEGER ARITHMETIC
C      M = NUMBER OF PIECES OF DATA, EACH OF LENGTH N
C      NTM = N TIMES M
C      NTMAX = MAXIMUM VALUE OF NTM FOR DIMENSIONING
C
C      [H(I);I=1,NTM] = SIGNIFICANT WAVE HEIGHTS
C      [T(I);I=1,NTM] = PERIOD AT PEAK OF WAVE SPECTRUM
C      [D(I);I=1,NTM] = DOMINANT DIRECTION OF TRAVEL IN DEGREES
C      [DS(I);I=1,NTM] = SINE(D(I))
C      [DC(I);I=1,NTM] = COS(D(I))
C      [HO(I);I=1,NTM] = LIST OF ORDERED WAVE HEIGHTS
C      [TO(I);I=1,NTM] = LIST OF ORDERED WAVE PERIODS
C      [DSO(I);I=1,NTM] = LIST OF ORDERED SINES
C      [DCO(I);I=1,NTM] = LIST OF ORDERED COSINES
C      DT = TIME INCREMENT IN HOURS
C      BW = EFFECTIVE WIDTH OF SMOOTHING IN SPECTRA EST., IN HZ.
C      [SP(4,4,I);I=1,MNO2P1] = SPECTRA MATRIX AT EACH FREQUENCY
C      WHERE THE SPECTRA AND CROSS-SPECTRA TABLED ARE ACTUALLY FOR
C      THE NORMAL SCORES OF THE TRANSFORMED DATA
C
C      FCO = HIGH FREQUENCY CUT OFF. FREQUENCIES HIGHER THAN THIS
C      WILL NOT BE COMPUTED IN ANALYSIS OR INCLUDED IN LATER
C      SIMULATIONS
C      [VC(I,J);I=1,4;J=1,NSIM] = COMPLEX-VALUED WORK ARRAY
C
C      [EM(I,II,J);I=1,8;II=1,8;J=1,MNO2P1] = MULTIPLIERS FOR
C      SIMULATION
C      [NEV(J);J=1,MNO2P1] = NUMBER OF NON-NEGLIGIBLE EIGENVALUES AT
C      FREQUENCY, J/(N*DT)
C      FRACT = FRACTION OF TRACE OF COVARIANCE PRESERVED IN
C      EIGENVALUES
C      EPS = EIGENVALUES LESS THAN EPS ARE AUTOMATICALLY DISCARDED
C      IN EIGENVALUE-EIGENVECTOR DECOMPOSITION OF THE
C      COVARIANCE MATRIX
C
C      IPDS,B,IB,IT,NMAX05 = WORK ARRAYS
C      NTMX05 = (NTMAX/5)+1, FORTRAN INTEGER ARITHMETIC
C      XR,XI,YR,YI = WORK VECTORS FOR FFT
C      [LLL(L);L=1,NTM] = WORK VECTOR IN DOCUMENTATION
C
C      WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING
C      *****
C
C      ***** DIMENSION STATEMENTS *****
C      DIMENSION H(NTMAX),T(NTMAX),D(NTMAX),DS(NTMAX),DC(NTMAX)
C      DIMENSION HO(NTMAX),TO(NTMAX),DSO(NTMAX),DCO(NTMAX)
C      DIMENSION SP(4,4,MNO2P1),EM(8,8,MNO2P1),NEV(MNO2P1)
C      DIMENSION IPDS(NTMAX),B(5),IB(NTMX05),IT(NTMX05)
C      DIMENSION XR(NSMAX),XI(NSMAX),YR(NSMAX),YI(NSMAX)

```

HPDPRE. FOR

```

DIMENSION EVAL(36),EIVEC(64),LLL(NTMAX)
COMPLEX VC(4,NSMAX)

```

```

C ***** BASIC CONSTANTS ***

```

```

EPS2=0.000001
NEO=NSIM-(NSIM/2)*2
NO2P1=(NSIM/2)+1
NTOP=FCO*NSIM*DT+1
IF (NTOP.GT.NO2P1) NTOP=NO2P1
IDOC=1
DF=1 0/(NSIM*DT)

```

```

C ***** COMPUTE NORMAL SCORE VERSION OF DATA CROSS-SPECTRAL ***

```

```

C ***** MATRIX ***

```

```

CALL HPDXSP(N,M,NSIM,NSMAX,MNO2P1,NTM,NTMAX,H,T,D,DS,DC,HO,
&TO,DSO,DCO,DT,BW,SP,NTOP,VC,EM,NEV,IPOS,B,IB,IT,NTMX05,XR,XI,
&YR,YI)

```

```

IF (IDOC.NE.0) THEN

```

```

C ***** WRITE NORMAL SCORE INPUT TIME SERIES ***

```

```

DO 82 L=1,NTM

```

```

LLL(L)=L

```

```

82 CONTINUE

```

```

WRITE (4,204)

```

```

204 FORMAT (//1X,'LISTING OF INPUT NORMAL SCORE DATA')

```

```

NS04=NTM/10

```

```

NREM=NTM-NS04*10

```

```

DO 81 I=1,NS04

```

```

LSTART=(I-1)*10+1

```

```

LEND=LSTART+9

```

```

WRITE (4,'(1X)')

```

```

WRITE (4,200) (LLL(L),L=LSTART,LEND)

```

```

200 FORMAT (1X,'TIME STEP NO.: ',10I7)

```

```

WRITE (4,201) (H(L),L=LSTART,LEND)

```

```

201 FORMAT (1X,' WAVE HEIGHT: ',10F7.2)

```

```

WRITE (4,202) (T(L),L=LSTART,LEND)

```

```

202 FORMAT (1X,' WAVE PERIOD: ',10F7.2)

```

```

WRITE (4,203) (DS(L),L=LSTART,LEND)

```

```

203 FORMAT (1X,'SIN DIRECTION: ',10F7.2)

```

```

WRITE (4,303) (DC(L),L=LSTART,LEND)

```

```

303 FORMAT (1X,'COS DIRECTION: ',10F7.2)

```

```

81 CONTINUE

```

```

IF (NREM.NE.0) THEN

```

```

LSTART=NTM-NREM+1

```

```

LEND=NTM

```

```

WRITE (4,'(1X)')

```

```

WRITE (4,200) (LLL(L),L=LSTART,LEND)

```

```

WRITE (4,201) (H(L),L=LSTART,LEND)

```

```

WRITE (4,202) (T(L),L=LSTART,LEND)

```

```

WRITE (4,203) (DS(L),L=LSTART,LEND)

```

```

WRITE (4,303) (DC(L),L=LSTART,LEND)

```

```

END IF

```

HPDPRE. FOR

END IF

IF (IDOC.NE.0) THEN

S1=0.0

S2=0.0

S3=0.0

S4=0.0

DO 1 I=1,NTM

S1=S1+H(I)

S2=S2+T(I)

S3=S3+DS(I)

S4=S4+DC(I)

1 CONTINUE

AV1=S1/NTM

AV2=S2/NTM

AV3=S3/NTM

AV4=S4/NTM

S11=0.0

S12=0.0

S13=0.0

S14=0.0

S22=0.0

S23=0.0

S24=0.0

S33=0.0

S34=0.0

S44=0.0

DO 2 I=1,NTM

S11=S11+(H(I)-AV1)*(H(I)-AV1)

S12=S12+(H(I)-AV1)*(T(I)-AV2)

S13=S13+(H(I)-AV1)*(DS(I)-AV3)

S14=S14+(H(I)-AV1)*(DC(I)-AV4)

S22=S22+(T(I)-AV2)*(T(I)-AV2)

S23=S23+(T(I)-AV2)*(DS(I)-AV3)

S24=S24+(T(I)-AV2)*(DC(I)-AV4)

S33=S33+(DS(I)-AV3)*(DS(I)-AV3)

S34=S34+(DS(I)-AV3)*(DC(I)-AV4)

S44=S44+(DC(I)-AV4)*(DC(I)-AV4)

2 CONTINUE

VAR11=S11/NTM

COV12=S12/NTM

COV13=S13/NTM

COV14=S14/NTM

VAR22=S22/NTM

COV23=S23/NTM

COV24=S24/NTM

VAR33=S33/NTM

COV34=S34/NTM

VAR44=S44/NTM

WRITE (4,100)

HPDPRE.FOR

```

100  FORMAT (//1X,'COVARIANCES COMPUTED FROM DATA:')
      WRITE (4,101) VAR11,VAR22,VAR33,VAR44
101  FORMAT (1X,'VAR11,VAR22,VAR33,VAR44= ',4F12.5)
      WRITE (4,102) COV12,COV13,COV14
102  FORMAT (1X,'COV12,COV13,COV14= ',3F12.5)
      WRITE (4,302) COV23,COV24,COV34
302  FORMAT (1X,'COV23,COV24,COV34= ',3F12.5)

      S11=0.0
      S12=0.0
      S13=0.0
      S14=0.0
      S22=0.0
      S23=0.0
      S24=0.0
      S33=0.0
      S34=0.0
      S44=0.0
      DO 3 I=1,NTOP
          S11=S11+SP(1,1,I)
          S12=S12+SP(1,2,I)
          S13=S13+SP(1,3,I)
          S14=S14+SP(1,4,I)
          S22=S22+SP(2,2,I)
          S23=S23+SP(2,3,I)
          S24=S24+SP(2,4,I)
          S33=S33+SP(3,3,I)
          S34=S34+SP(3,4,I)
          S44=S44+SP(4,4,I)
3      CONTINUE
      VAR11=2.0*S11*DF
      VAR22=2.0*S22*DF
      VAR33=2.0*S33*DF
      VAR44=2.0*S44*DF
      COV12=2.0*S12*DF
      COV13=2.0*S13*DF
      COV14=2.0*S14*DF
      COV23=2.0*S23*DF
      COV24=2.0*S24*DF
      COV34=2.0*S34*DF
      WRITE (4,103)
103  FORMAT (//1X,'COVARIANCES COMPUTED FROM SPECTRA:')
      WRITE (4,104) VAR11,VAR22,VAR33,VAR44
104  FORMAT (1X,'VAR11,VAR22,VAR33,VAR44= ',4F12.5)
      WRITE (4,105) COV12,COV13,COV14
105  FORMAT (1X,'COV12,COV13,COV14= ',3F12.5)
      WRITE (4,305) COV23,COV24,COV34
305  FORMAT (1X,'COV23,COV24,COV34= ',3F12.5)
      END IF

```

C ***** COMPUTE THE MULTIPLIER MATRIX FROM ***

HPDPRE. FOR

```

C      ***** EIGENVECTORS FOR EACH J.          ***
      SMULT=NSIM*DT
      SSMULT=SMULT/2.0
      DO 9 J=1,NTOP
        DO 16 I=1,8
          DO 17 II=1,8
            EM(I,II,J)=0.0
17      CONTINUE
16      CONTINUE
      TRC=SMULT*(SP(1,1,J)+SP(2,2,J)+SP(3,3,J)+SP(4,4,J))
      IF (TRC.GT.EPS2) THEN
        IF ((NEO.EQ.1.AND.J.GT.1).OR.(NEO.EQ.0.AND
&      .J.GT.1.AND.J.LT.NO2P1)) THEN
          EVAL(1)=SP(1,1,J)*SSMULT
          EVAL(2)=0.0
          EVAL(3)=SP(1,1,J)*SSMULT
          EVAL(4)=SP(1,2,J)*SSMULT
          EVAL(5)=-SP(2,1,J)*SSMULT
          EVAL(6)=SP(2,2,J)*SSMULT
          EVAL(7)=SP(2,1,J)*SSMULT
          EVAL(8)=SP(1,2,J)*SSMULT
          EVAL(9)=0.0
          EVAL(10)=SP(2,2,J)*SSMULT
          EVAL(11)=SP(1,3,J)*SSMULT
          EVAL(12)=-SP(3,1,J)*SSMULT
          EVAL(13)=SP(2,3,J)*SSMULT
          EVAL(14)=-SP(3,2,J)*SSMULT
          EVAL(15)=SP(3,3,J)*SSMULT
          EVAL(16)=SP(3,1,J)*SSMULT
          EVAL(17)=SP(1,3,J)*SSMULT
          EVAL(18)=SP(3,2,J)*SSMULT
          EVAL(19)=SP(2,3,J)*SSMULT
          EVAL(20)=0.0
          EVAL(21)=SP(3,3,J)*SSMULT
          EVAL(22)=SP(1,4,J)*SSMULT
          EVAL(23)=-SP(4,1,J)*SSMULT
          EVAL(24)=SP(2,4,J)*SSMULT
          EVAL(25)=-SP(4,2,J)*SSMULT
          EVAL(26)=SP(3,4,J)*SSMULT
          EVAL(27)=-SP(4,3,J)*SSMULT
          EVAL(28)=SP(4,4,J)*SSMULT
          EVAL(29)=SP(4,1,J)*SSMULT
          EVAL(30)=SP(1,4,J)*SSMULT
          EVAL(31)=SP(4,2,J)*SSMULT
          EVAL(32)=SP(2,4,J)*SSMULT
          EVAL(33)=SP(4,3,J)*SSMULT
          EVAL(34)=SP(3,4,J)*SSMULT
          EVAL(35)=0.0
          EVAL(36)=SP(4,4,J)*SSMULT

          CALL EIG(EVAL,EIVC,8,0,36,36,64)

```

HPDPRE.FOR

```

NEV(J)=0
SUM=0.0
DO 10 I=1,8
    III=I*(I+1)/2
    SUM=SUM+EVAL(III)/TRC
    IF (EVAL(III).GT.EPS) NEV(J)=NEV(J)+1
    IF (SUM.GT.FRACT) GO TO 21
10    CONTINUE
21    CONTINUE

IF (IDOC.NE.0) THEN
    WRITE (4,106) J,TRC,NEV(J),SUM
106    FORMAT (/1X,' J,TRC,NEV,SUM= ',I5,G15.5,I5,G15.5)
    WRITE (4,107) EVAL(1),EVAL(3),EVAL(6),EVAL(10)
107    FORMAT (6X,' EV1,EV2,EV3,EV4= ',4G13.5)
    WRITE (4,108) EVAL(15),EVAL(21),EVAL(28),EVAL(36)
108    FORMAT (6X,' EV5,EV6,EV7,EV8= ',4G13.5)
    WRITE (4,110)
110    FORMAT (1X,' EIGENVECTORS:')
    DO 111 KEV=1,8
        EE1=EIVC(KEV)
        EE2=EIVC(KEV+8)
        EE3=EIVC(KEV+16)
        EE4=EIVC(KEV+24)
        EE5=EIVC(KEV+32)
        EE6=EIVC(KEV+40)
        EE7=EIVC(KEV+48)
        EE8=EIVC(KEV+56)
        WRITE (4,112) EE1,EE2,EE3,EE4,EE5,EE6,EE7,EE8
112    FORMAT (1X,8F9.3)
111    CONTINUE
    END IF

    DO 11 I=1,8
        DO 12 II=1,8
            EM(I,II,J)=0.0
12        CONTINUE
11    CONTINUE
    DO 13 II=1,NEV(J)
        IIII=II*(II+1)/2
        DO 14 I=1,8
            III=(II-1)*8+I
            EM(I,II,J)=SQRT(EVAL(IIII))*EIVC(III)
14        CONTINUE
13    CONTINUE
ELSE
    EVAL(1)=SP(1,1,J)*SMULT
    EVAL(2)=SP(1,2,J)*SMULT
    EVAL(3)=SP(2,2,J)*SMULT
    EVAL(4)=SP(1,3,J)*SMULT

```

HPDPRE.FOR

```

EVAL(5)=SP(2,3,J)*SMULT
EVAL(6)=SP(3,3,J)*SMULT
EVAL(7)=SP(1,4,J)*SMULT
EVAL(8)=SP(2,4,J)*SMULT
EVAL(9)=SP(3,4,J)*SMULT
EVAL(10)=SP(4,4,J)*SMULT

CALL EIG(EVAL,EIVC,4,0,10,36,64)

NEV(J)=0
SUM=0.0
DO 15 I=1,4
    III=I*(I+1)/2
    SUM=SUM+EVAL(III)/TRC
    IF (EVAL(III).GT.EPS) NEV(J)=NEV(J)+1
    IF (SUM.GT.FRACT) GO TO 31
15  CONTINUE
31  CONTINUE

IF (IDOC.NE.0) THEN
    WRITE (4,109) J,TRC
109  FORMAT (/1X,'J,TRC= ',I5,G15.5)
    WRITE (4,107) EVAL(1),EVAL(3),EVAL(6),EVAL(10)
    WRITE (4,110)
    DO 114 KEV=1,4
        EE1=EIVC(KEV)
        EE2=EIVC(KEV+8)
        EE3=EIVC(KEV+16)
        EE4=EIVC(KEV+24)
        WRITE (4,112) EE1,EE2,EE3,EE4
114  CONTINUE
    END IF

    DO 18 II=1,NEV(J)
        IIII=II*(II+1)/2
        DO 19 I=1,4
            III=(II-1)*8+I
            EM(I,II,J)=SQRT(EVAL(IIII))*EIVC(III)
19  CONTINUE
18  CONTINUE
    END IF
    END IF
9  CONTINUE
RETURN
END

C
SUBROUTINE HPDXSP(N,M,NSIM,NSMAX,MNO2P1,NTM,NTMAX,H,T,D,DS,
&DC,H0,TO,DSO,DCO,DT,BW,SP,NTOP,VC,EM,NEV,IPOS,B,IB,IT,
&NTMX05,XR,XI,YR,YI)
C *****

```

HPDPRE.FOR

```

C   PREPROCESS M PIECES OF CONTINUOUS TIME INTERVAL DATA, EACH
C   OF LENGTH N, TO PREPARE ARRAYS OF NORMAL SCORED DATA,
C   RANKED DATA VECTORS, AND CROSS-SPECTRAL MATRICES AT EACH
C   FREQUENCY. ALL THIS IS DONE AS SUBSEQUENT INPUT FOR
C   COMPUTATION OF LONG-TERM SYNTHESIS OR SIMULATION OF
C   SIGNIFICANT WAVE HEIGHT, PEAK PERIOD, AND DOMINANT
C   DIRECTION OF TRAVEL.
C
C   N = LENGTH OF INPUT DATA FOR STATIONARY INTERVAL, USUALLY
C   ONE MONTH. IF KK MONTHS OF THE SAME TYPE (SAY, A
C   COMPLETE WINTER PERIOD) ARE TO BE USED TO GET THE
C   STATISTICS FOR THAT TIME INTERVAL THEN N = NUMBER OF
C   DATA IN THE COMBINED KK-MONTHS.
C   NSIM = LENGTH OF TIME SERIES TO BE SIMULATED SUBSEQUENTLY.
C   THIS SHOULD BE AN EXACT INTEGER MULTIPLE OF N.
C   IT MAY BE EQUAL TO N, TWO TIMES N, THREE TIMES N,
C   OR SOME OTHER INTEGER MULTIPLE OF N. IN ANY CASE,
C   NSIM SHOULD BE SELECTED TO BE REPRESENTABLE IN THE
C   FORM  $(2**I)*(3**J)*(5**K)$  FOR POSITIVE INTEGERS
C   (I,J,K). THIS IS REQUIRED FOR THE FFT SOFTWARE USED.
C   NSMAX = MAXIMUM VALUE OF NSIM, FOR DIMENSIONING ONLY
C   NO2P1 = (NSIM/2)+1, IN FORTRAN INTEGER ARITHMETIC
C   MND2P1 = (NSMAX/2)+1, IN FORTRAN INTEGER ARITHMETIC
C   M = NUMBER OF PIECES OF DATA, EACH OF LENGTH N
C   NTM = N TIMES M
C   NTMAX = MAXIMUM VALUE OF NTM FOR DIMENSIONING
C
C   [H(I);I=1,NTM] = SIGNIFICANT WAVE HEIGHTS
C   [T(I);I=1,NTM] = PERIOD AT PEAK OF WAVE SPECTRUM
C   [D(I);I=1,NTM] = DOMINANT DIRECTION OF TRAVEL IN DEGREES
C   [DS(I);I=1,NTM] = SIN(D(I))
C   [DC(I);I=1,NTM] = COS(D(I))
C   [HO(I);I=1,NTM] = LIST OF ORDERED WAVE HEIGHTS
C   [TO(I);I=1,NTM] = LIST OF ORDERED WAVE PERIODS
C   [DSO(I);I=1,NTM] = LIST OF ORDERED SINES OF DIRECTION
C   [DCO(I);I=1,NTM] = LIST OF ORDERED COSINES OF DIRECTION
C   DT = TIME INCREMENT IN HOURS
C   BW = EFFECTIVE WIDTH OF SMOOTHING IN SPECTRA EST., IN HZ.
C   [SP(4,4,I);I=1,NO2P1] = SPECTRA MATRIX AT EACH FREQUENCY IN
C   FORM:
C
C   ! SPEC H          C-SPEC (H,T)      C-SPEC (H,DS)   C-SPEC (H,DC) !
C   ! Q-SPEC (H,T)    SPEC T            C-SPEC (T,DS)   C-SPEC (T,DC) !
C   ! Q-SPEC (H,DS)   Q-SPEC (T,DS)     SPEC DS        C-SPEC (DS,DC) !
C   ! Q-SPEC (H,DC)   Q-SPEC (T,DC)     Q-SPEC (DS,DC)  SPEC DC      !
C
C   WHERE THE SPECTRA AND CROSS-SPECTRA TABLED ARE ACTUALLY FOR
C   THE NORMAL SCORES OF THE TRANSFORMED DATA
C
C   FCC = HIGH FREQUENCY CUT OFF. FREQUENCIES HIGHER THAN THIS
C   WILL NOT BE COMPUTED IN ANALYSIS OR INCLUDED IN LATER

```

HPDPRE.FOR

```

C          SIMULATIONS
C      NTOP = MINIMUM OF (FCO*NSIM*DT+1,NO2P1)
C      [VC(I,J);I=1,4;J=1,NSIM] = COMPLEX-VALUED WORK ARRAY
C
C      IPOS,B,IB,IT,NTMX05 = WORK ARRAYS
C      NTMX05 = (NTMAX/5)+1, FORTRAN INTEGER ARITHMETIC
C      XR,XI,YR,YI = WORK VECTORS FOR FFT
C
C      WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING
C      *****

C      ***** PARAMETER AND DIMENSION STATEMENTS *****
C      DIMENSION H(NTMAX),T(NTMAX),D(NTMAX),DS(NTMAX),DC(NTMAX)
C      DIMENSION HO(NTMAX),TO(NTMAX),DSO(NTMAX),DCO(NTMAX)
C      DIMENSION SP(4,4,MNO2P1),EM(8,8,MNO2P1),NEV(MNO2P1)
C      DIMENSION IPOS(NTMAX),B(5),IB(NTMX05),IT(NTMX05)
C      DIMENSION XR(NSMAX),XI(NSMAX),YR(NSMAX),YI(NSMAX)
C      COMPLEX VC(4,NSMAX)

C      ***** CONSTANTS *****
C      PI=3.14159265
C      CONV=PI/180.0
C      NO2P1=(NSIM/2)+1
C      NMULT=NSIM/N

C      LEFT=NSIM-NMULT*N
C      IF (LEFT.GT.0) THEN
C          WRITE (*,200)
C          WRITE (2,200)
200      FORMAT (1X,'!!!! WARNING, NSIM SHOULD BE AN EXACT',
C          & ' MULTIPLE OF N !!!!!')
C      END IF

C      ***** COMPUTE DS, DC, NORMAL SCORES, DSO, AND DCO *****
C      DO 1 I=1,NTM
C          TH=D(I)*CONV
C          DS(I)=SIN(TH)
C          DC(I)=COS(TH)
1      CONTINUE
C      CALL NSCORE(H,HO,IPOS,NTM,NTMAX,NTMX05,B,IB,IT)
C      CALL NSCORE(T,TO,IPOS,NTM,NTMAX,NTMX05,B,IB,IT)
C      CALL NSCORE(DS,DSO,IPOS,NTM,NTMAX,NTMX05,B,IB,IT)
C      CALL NSCORE(DC,DCO,IPOS,NTM,NTMAX,NTMX05,B,IB,IT)

C      ***** INITIALIZE CROSS-SPECTRAL MATRIX *****
C      DO 11 J=1,NO2P1
C          DO 12 I=1,8
C              DO 13 II=1,8
C                  SP(I,II,J)=0.0
13      CONTINUE
12      CONTINUE

```

HPDPRE.FOR

11 CONTINUE

C ***** COMPUTE CROSS SPECTRA. SUM OF MATRIX OF CROSS-SPECTRA ***
C ***** IS ACCUMULATED IN SUBR. XSPEC *****

L=0

DO 14 I=1,M

DO 3 J=1,N

L=L+1

VC(1,J)=H(L)

VC(2,J)=T(L)

VC(3,J)=DS(L)

VC(4,J)=DC(L)

3 CONTINUE

C ***** PAD WITH ZEROS, IF NEEDED ***

IF (NSIM.GT.N) THEN

DO 40 J=N+1,NSIM

VC(1,J)=0.0

VC(2,J)=0.0

VC(3,J)=0.0

VC(4,J)=0.0

40 CONTINUE

END IF

CALL XSPEC(VC,4,4,N,NSIM,NSMAX,NO2P1,MNO2P1,DT,BW,SP,

& NTOP,XR,XI,YR,YI)

14 CONTINUE

C ***** AVERAGE THE SP MATRIX OVER THE M PIECES ***

C ***** IF THERE IS MORE THAN ONE PIECE *****

IF (M.GT.1) THEN

DO 26 J=1,NTOP

DO 27 I=1,4

DO 28 II=1,4

SP(I,II,J)=SP(I,II,J)/M

28 CONTINUE

27 CONTINUE

26 CONTINUE

END IF

IDOC=1

IF (IDOC.NE.0) THEN

WRITE (4,210)

210 FORMAT(/1X,'OUTPUT FROM SUBR. HPDXSP')

DO 30 J=1,NTOP

WRITE (4,211) J

211 FORMAT (/1X,'J= ',I5)

DO 31 I=1,4

WRITE (4,212) (SP(I,II,J),II=1,4)

212 FORMAT (1X,4G15.5)

31 CONTINUE

30 CONTINUE

END IF

HPDPRE.FOR

RETURN
END

```

C
C      SUBROUTINE FT235(N,SGN,LIST,XR,XI)
C *****
C      (XR(J),J=1,N)=REAL PART OF SEQUENCE TO BE FOURIER TRANSFORMED.
C      (XI(J),J=1,N)=IMAG. PART OF SEQUENCE TO BE FOURIER TRANSFORMED.
C      THE DISCRETE FOURIER TRANSFORM IS OF THE FORM:
C      A(M)=SUM FOR J=1 TO N OF
C              (XR+I*XI)*EXP(I*2*PI*SGN*(M-1)*(J-1)/N),
C              FOR 1.LE.M.LE.N
C      THUS, SGN DEFINES THE SIGN OF THE EXPONENT, AND IS EITHER 1.0
C      OR -1.0.
C      A TABLE OF TRIGONOMETRIC AND BIT REVERSAL FUNCTIONS ARE
C      USED IN THE FFT. IF LIST=0, THE TABLE IS COMPUTED. IF
C      LIST=1, THE TABLE IS NOT COMPUTED AND IS PRESUMED TO BE
C      PRESENT FROM EARLIER CALLS TO THE SUBROUTINE.
C      CONSEQUENTLY, LIST=0 ON FIRST CALL TO SUBR.
C              LIST=1 FOR SUBSEQUENT CALLS AT SAME N VALUE.
C
C      PARAMETER VALUES:
C      MAXN=LARGEST VALUE TO BE USED FOR N (LENGTH OF SEQUENCE)
C      MAXN2=LARGEST POWER OF 2 LESS THAN OR EQUAL TO MAXN
C      MAXN3=LARGEST POWER OF 3 LESS THAN OR EQUAL TO MAXN
C      MAXN5=LARGEST POWER OF 5 LESS THAN OR EQUAL TO MAXN
C      MAXN02=MAXN2/2
C      MAXN03=MAXN3/3
C      MAXN05=MAXN5/5
C
C      COMMON BLOCK *****
C      PARAMETER (MAXN=1200,MAXN2=1024,MAXN3=729,MAXN5=625)
C      PARAMETER (MAXN02=512,MAXN03=243,MAXN05=125)
C      DIMENSION XR(1),XI(1)
C *****
C      DIMENSION W(MAXN),WR(MAXN2),WI(MAXN2),WRD(MAXN3),WID(MAXN3)
C      DIMENSION WRT(MAXN5),WIT(MAXN5)
C      DIMENSION IRB(MAXN02),CO(MAXN02),SI(MAXN02)
C      DIMENSION IRBD(MAXN03),COD(MAXN03),SID(MAXN03)
C      DIMENSION IRBT(MAXN05),COT(MAXN05),SIT(MAXN05)
C      DIMENSION CO1(MAXN05),SI1(MAXN05),CO2(MAXN05),SI2(MAXN05)
C      DIMENSION CO3(MAXN05),SI3(MAXN05)
C      COMMON /FFT/W,WR,WI,WRD,WID,WRT,WIT,IRB,CO,SI,IRBD,COD,SID
C      COMMON /FFT/IRBT,COT,SIT,CO1,SI1,CO2,SI2,CO3,SI3
C *****
C      PARAMETER (MAXN=5000,MAXN2=4096,MAXN3=2187,MAXN5=3125)
C      PARAMETER (MAXN02=2048,MAXN03=729,MAXN05=625)
C      NOTE: THE ABOVE COMMON BLOCK PARAMETERS WOULD ALLOW THE USE OF
C      ANY N.LE.5000 OF THE FORM N=(2**K)*(3**L)*(5**M) FOR
C      INTEGER K, L, AND M WITHOUT CHANGING DIMENSIONS.
C      THE CHOICES OF N WHICH MATCH THESE REQUIREMENTS ARE:

```

HPDPRE.FOR

C	2	3	4	5	6	8	9	10
C	12	15	16	18	20	24	25	27
C	30	32	36	40	45	48	50	54
C	60	64	72	75	80	81	90	96
C	100	108	120	125	128	135	144	150
C	160	162	180	192	200	216	225	240
C	243	250	256	270	288	300	320	324
C	360	375	384	400	405	432	450	480
C	486	500	512	540	576	600	625	640
C	648	675	720	729	750	768	800	810
C	864	900	960	972	1000	1024	1080	1125
C	1152	1200	1215	1250	1280	1296	1350	1440
C	1458	1500	1536	1600	1620	1728	1800	1875
C	1920	1944	2000	2025	2048	2160	2187	2250
C	2304	2400	2430	2500	2560	2592	2700	2880
C	2916	3000	3072	3125	3200	3240	3375	3456
C	3600	3645	3750	3840	3888	4000	4050	4096
C	4320	4374	4500	4608	4800	4860	5000	

C HOWEVER, FOR A FIXED N, LESS COMPUTER STORAGE CAN BE
C REQUIRED IF THE FOLLOWING "EXACT" DIMENSIONING IS
C INTRODUCED:

C N2=2**K
C N3=3**L
C N5=5**M
C N02=N2/2
C N03=N3/3
C N05=N5/5

C DIMENSION W(N),WR(N2),WI(N2),WRD(N3),WID(N3),WRT(N5),
C WIT(N5),IRB(N02),CO(N02),SI(N02),IRBD(N03),COD(N03),
C SID(N03),IRBT(N05),COT(N05),SIT(N05),COI(N05),
C SI1(N05),CO2(N05),SI2(N05),CO3(N05),SI3(N05)

C THE MAXIMAL DIMENSIONING WAS TAKEN FROM THE LARGEST
C POWER OF 2, OF 3, AND OF 5 WHICH IS LESS THAN 5000.

C PROGRAMS WRITTEN BY A. YFANTIS, UNIV. NEVADA, LAS VEGAS,
C AND LEON BORGMAN, UNIV. WYOMING, LARAMIE, WYO.

C *****

```

      NN=N
      K=0
1  ND=NN-(NN/2)*2
      IF (ND.NE.0) GO TO 2
      K=K+1
      NN=NN/2
      IF (NN.EQ.1) GO TO 2
      GO TO 1
2  L=0
3  ND=NN-(NN/3)*3
      IF (ND.NE.0) GO TO 4
      L=L+1
      NN=NN/3

```

HPDPRE.FOR

```

      IF (NN.EQ.1) GO TO 4
      GO TO 3
4 M=0
5 ND=NN-(NN/5)*5
      IF (ND.NE.0) GO TO 6
      M=M+1
      NN=NN/5
      IF (NN.EQ.1) GO TO 6
      GO TO 5
6 CONTINUE
      IF (NN.NE.1) THEN
        WRITE (2,100)
100    FORMAT ('WARNING: THE ASSUMPTIONS OF THE SUBR. ARE VIOLATED' /
&'          N IS NOT EQUAL TO (2**K)*(3**L)*(5**M)')
        RETURN
      ELSE IF (K.NE.0.AND.L.EQ.0.AND.M.EQ.0) THEN
        NO2=N/2
        CALL FFT2(K,N,NO2,SGN,LIST,XR,XI,IRB,CO,SI)
      ELSE IF (K.EQ.0.AND.L.NE.0.AND.M.EQ.0) THEN
        NO3=N/3
        CALL FFT3(L,N,NO3,SGN,LIST,XR,XI,IRBD,COD,SID)
      ELSE IF (K.EQ.0.AND.L.EQ.0.AND.M.NE.0) THEN
        NO5=N/5
        CALL FFT5(M,N,NO5,SGN,LIST,XR,XI,IRBT,COT,SIT,CO1,SI1,
&    CO2,SI2,CO3,SI3)
      ELSE IF (K.NE.0.AND.L.NE.0.AND.M.EQ.0) THEN
        N2=2**K
        N3=3**L
        NO2=N2/2
        NO3=N3/3
        CALL FFT23(K,L,N,N2,N3,NO2,NO3,SGN,LIST,XR,XI,IRB,IRBD,
&    CO,SI,COD,SID,WR,WI,WRD,WID,W)
      ELSE IF (K.NE.0.AND.L.EQ.0.AND.M.NE.0) THEN
        N2=2**K
        N5=5**M
        NO2=N2/2
        NO5=N5/5
        CALL FFT25(K,M,N,N2,N5,NO2,NO5,SGN,LIST,XR,XI,IRB,IRBT,
&    CO,SI,COT,SIT,WR,WI,WRT,WIT,W,CO1,SI1,CO2,SI2,CO3,SI3)
      ELSE IF (K.EQ.0.AND.L.NE.0.AND.M.NE.0) THEN
        N3=3**L
        N5=5**M
        NO3=N3/3
        NO5=N5/5
        CALL FFT35(L,M,N,N3,N5,NO3,NO5,SGN,LIST,XR,XI,IRBD,IRBT,
&    COD,SID,COT,SIT,WRD,WID,WRT,WIT,W,CO1,SI1,CO2,SI2,CO3,SI3)
      ELSE
        N2=2**K
        N3=3**L
        N5=5**M
        NO2=N2/2

```

HPDPRE.FOR

```

      NO3=N3/3
      NO5=N5/5
      CALL FFT235(K,L,M,N,N2,N3,N5,NO2,NO3,NO5,SGN,LIST,XR,XI,
&   IRB,IRBD,IRBT,CO,SI,COD,SID,COT,SIT,WR,WI,WRD,WID,WRT,WIT,
&   CO1,SI1,CO2,SI2,CO3,SI3,W)
      END IF
      RETURN
      END

C
      SUBROUTINE FFT2(K,N,NO2,SGN,LIST,XR,XI,IRB,CO,SI)
C *****
C   N=LENGTH OF DATA SERIES BEING TRANSFORMED.
C   K=LOG2(N).
C   NO2=N/2
C   SGN= +1 OR -1 ACCORDING AS THE EXPONENT IN THE FFT TRANSFORMATION
C   IS POSITIVE OR NEGATIVE.
C   IF LIST=0, THE REVERSED-BIT LIST AND THE ASSOCIATED SINES AND
C   COSINES ARE COMPUTED. IF LIST=1, THE LIST IS NOT COMPUTED.
C   INSTEAD, THE LIST COMPUTED ON THE PREVIOUS CALL IS USED.
C   XR=REAL PART OF THE INPUT DATA SERIES. INITIALLY AND EQUALS REAL
C   PART OF OUTPUT ON RETURN FROM THE SUBR.
C   XI=IMAG. PART OF DATA SERIES ON INPUT AND IS IMAG. PART OF OUTPUT
C   ON RETURN FROM THE SUBR.
C   IRB=NO2-DIMENSIONAL VECTOR OF REVERSED BITS ON OUTPUT.
C   CO=NO2-DIMENSIONAL VECTOR OF COSINES ON OUTPUT.
C   SI=NO2-DIMENSIONAL VECTOR OF SINES ON OUTPUT.
C *****
      DIMENSION IRB(1),CO(1),SI(1),XR(1),XI(1)
      IF (LIST.EQ.1) GO TO 3
C   THE REVERSED-BIT LIST AND TRIG. FUNCTIONS ARE GENERATED BY THE
C   PROCEDURE GIVED IN TABLE 7-VI.
      IRB(1)=0
      DO 1 J=1,K
        ID=2**(J-1)
        DO 1 I=1,ID
          IRB(I)=IRB(I)*2
          IF (J.LT.K) IRB(I+ID)=IRB(I)+1
1      CONTINUE
      FN=N
      W=6.2831853/FN
      DO 2 I=1,ID
        FIR=IRB(I)/2
        A=FIR*W
        CO(I)=COS(A)
        SI(I)=SIN(A)
2      CONTINUE
3      CONTINUE
C   FOR EACH COLUMN OF THE FFT (NC=COLUMN NUMBER), CALCULATE
C   NB=NUMBER OF BLOCKS IN THE COLUMN, LB=LENGTH OF BLOCKS, LB2=HALF
C   OF BLOCK LENGTH.
      DO 4 NC=1,K

```

HPDPRE.FOR

```

NB=2** (NC-1)
LB=N/NB
LB2=LB/2
C THE BLOCKS OF THE COLUMN ARE LOOPED OVER. IS=SEQUENCE NUMBER
C AT THE START OF THE BLOCK, IFF=SEQUENCE NUMBER HALF WAY THROUGH
C BLOCK.
DO 4 IB=1,NB
C=CO(IB)
S=SGN*SI(IB)
IS=(IB-1)*LB+1
IFF=(IB-1)*LB+LB2
C THE VALUES IN THE NEXT COLUMN ARE COMPUTED USING THE TRIG.
C FUNCTIONS FROM THE PRE-GENERATED LIST.
DO 4 I=IS, IFF
I2=I+LB2
QR=XR(I2)*C-XI(I2)*S
QI=XR(I2)*S+XI(I2)*C
XR(I2)=XR(I)-QR
XI(I2)=XI(I)-QI
XR(I)=XR(I)+QR
XI(I)=XI(I)+QI
4 CONTINUE
C THE FFT COEFFICIENTS ARE UNSCRANBLED INTO NORMAL ORDER.
DO 6 I=1,N02
DO 6 L=1,2
IR=IRB(I)+L
II=I+(L-1)*N02
IF (IR.LE.II) GO TO 6
ZR=XR(IR)
ZI=XI(IR)
XR(IR)=XR(II)
XI(IR)=XI(II)
XR(II)=ZR
XI(II)=ZI
6 CONTINUE
RETURN
END
C
SUBROUTINE FFT3(K,N,N03,SGN,LIST,XR,XI,IRB,CO,SI)
DIMENSION IRB(1),CO(1),SI(1),XR(1),XI(1)
C *****
C THE INPUT TO THE SUBROUTINE CONSISTS OF THE VARIABLES
C N, K, N03, SGN, LIST, XR, AND XI. N IS THE NUMBER OF
C MEASUREMENTS WHICH IS OF THE FORM N=3**K, N03 IS EQUAL
C TO N/3. SGN IS -1 IN THE CASE OF THE FOURIER TRANSFORM,
C AND +1 IN THE CASE OF THE INVERSE FOURIER TRANSFORM.
C XR, XI ARE THE REAL AND IMAGINARY PARTS RESPECTIVELY,
C OF THE INPUT COMPLEX SEQUENCE OF NUMBERS. AFTER THE
C FFT3 IS PERFORMED, XR, XI, ARE THE REAL AND IMAGINARY
C PARTS OF THE TRANSFORMED DATA. IF LIST=0, THE REVERSE-
C DIGIT LIST AND THE ASSOCIATED SINES AND COSINES ARE

```

HPDPRE.FOR

```

C      COMPUTED. IF LIST=1 THE LIST IS NOT COMPUTED. INSTEAD
C      THE LIST COMPUTED OF THE PREVIOUS CALL IS USED. IRB,
C      CO, AND SI, ARE WORK VECTORS.
C *****
      T=6.2831853/3.0
      T1=2.*T
      FN=N
      IF(LIST .EQ. 1) GO TO 3
      IRB(1)=0
      DO 1 J=1,K
      ID=3*(J-1)
      DO 1 I=1,ID
      IRB(I)=IRB(I)*3
      IF(J .EQ. K) GO TO 1
      IRB(I+ID)=IRB(I)+1
      IRB(I+2*ID)=IRB(I)+2
1      CONTINUE
      DO 2 I=1,ID
      FIR=IRB(I)/FN
      A=FIR*T
      CO(I)=COS(A)
      SI(I)=SIN(A)
2      CONTINUE
C      FOR EACH COLUMN OF THE FFT (NC=COLUMN NUMBER), NB=NUMBER
C      OF BLOCKS IN THE COLUMN, LB=LENGTH OF BLOCKS, LB3= A THIRD
C      OF BLOCK LENGTH.
3      C1=COS(T)
      S1=SGN*SIN(T)
      C2=COS(T1)
      S2=SGN*SIN(T1)
      DO 4 NC=1,K
      NB=3*(NC-1)
      LB=N/NB
      LB3=LB/3
C      THE BLOCKS OF THE COLUMN ARE LOOPED OVER. IS=SEQUENCE NUMBER
C      AT THE START OF THE BLOCK, IFF=SEQUENCE NUMBER A THIRD WAY
C      THROUGH BLOCK
      DO 4 IB=1,NB
      C=CO(IB)
      S=SGN*SI(IB)
      IS=(IB-1)*LB+1
      IFF=(IB-1)*LB+LB3
C      THE VALUES IN THE NEXT COLUMN ARE COMPUTED USING THE TRIG.
C      FUNCTIONS FROM THE PRE-GENERATED LIST.
      DO 4 I=IS, IFF
      I2=I+LB3
      I3=I+2*LB3
      QR1=XR(I2)*C-XI(I2)*S
      QI1=XR(I2)*S+XI(I2)*C
      QR2=XR(I3)*(C*C-S*S)-2.*XI(I3)*S*C
      QI2=(C*C-S*S)*XI(I3)+2.*S*C*XR(I3)

```

HPDPRE.FOR

```

XR(I2)=XR(I)+C1*QR1-S1*QI1+C2*QR2-S2*QI2
XI(I2)=XI(I)+S1*QR1+C1*QI1+S2*QR2+C2*QI2
XR(I3)=XR(I)+C2*QR1-S2*QI1+C1*QR2-S1*QI2
XI(I3)=XI(I)+S2*QR1+C2*QI1+S1*QR2+C1*QI2
XR(I)=XR(I)+QR1+QR2
XI(I)=XI(I)+QI1+QI2
4 CONTINUE
C THE FFT COEFFICIENTS ARE INSCRAMBLED INTO NORMAL ORDER
DO 6 I=1,N03
DO 6 L=1,3
IR=IRB(I)+L
II=I+(L-1)*N03
IF (IR.LE.II) GO TO 6
ZR=XR(IR)
ZI=XI(IR)
XR(IR)=XR(II)
XI(IR)=XI(II)
XR(II)=ZR
XI(II)=ZI
6 CONTINUE
RETURN
END
C
SUBROUTINE FFT5(K,N,N05,SGN,LIST,XR,XI,IRB,CO,SI,CO1,SI1,
+ CO2,SI2,CO3,SI3)
DIMENSION IRB(1),CO(1),SI(1),XR(1),XI(1),CO1(1),
+ SI1(1),SI2(1),CO2(1),CO3(1),SI3(1)
C *****
C THE INPUT TO THE SUBROUTINE CONSISTS OF THE VARIABLES
C N, K, N05, SGN, LIST, XR, XI. N IS THE NUMBER OF MEAS-
C UREMENTS WHICH IS OF THE FORM N=5**K, N05 IS EQUAL TO
C N/5. SGN IS -1 IN THE CASE OF THE FOURIER TRANSFORM,
C AND +1 IN THE CASE OF THE INVERSE FOURIER TRANSFORM.
C XR AND XI ARE THE REAL AND IMAGINARY PARTS RESPECTIVELY
C OF THE INPUT COMPLEX SEQUENCE OF NUMBERS. AFTER THE FFT5
C IS PERFORMED XR, XI ARE THE REAL AND IMAGINARY PARTS OF
C THE TRANSFORMED DATA. IF LIST=0, THE REVERSE-DIGIT LIST
C AND THE ASSOCIATED SINES AND COSINES ARE COMPUTED. IF
C LIST=1, THE LIST IS NOT COMPUTED. INSTEAD THE LIST COM-
C PUTED ON THE PREVIOUS CALL IS USED. IRB, CO, SI, CO1,
C SI1, CO2, SI2, CO3, AND SI3, ARE WORK VECTORS.
C *****
T=6.2831853/5.
T1=2.*T
T2=3.*T
T3=4.*T
FN=N
IF(LIST.EQ. 1) GO TO 3
IRB(1)=0
DO 1 J=1,K
ID=5** (J-1)

```

HPDPRE.FOR

```

DO 1 I=1,ID
  IRB(I)=IRB(I)*5
  IF(J.EQ.K) GO TO 1
  IRB(I+ID)=IRB(I)+1
  IRB(I+2*ID)=IRB(I)+2
  IRB(I+3*ID)=IRB(I)+3
  IRB(I+4*ID)=IRB(I)+4
1  CONTINUE
  DO 2 I=1,ID
    FIR=IRB(I)/FN
    A=FIR*T
    CO(I)=COS(A)
    SI(I)=SIN(A)
    A1=2.*A
    CO1(I)=COS(A1)
    SI1(I)=SIN(A1)
    A2=3.*A
    CO2(I)=COS(A2)
    SI2(I)=SIN(A2)
    A3=4.*A
    CO3(I)=COS(A3)
    SI3(I)=SIN(A3)
  2  CONTINUE
C   FOR EACH COLUMN OF THE FFT (NC=COLUMN NUMBER), CALCULATE
C   NB=NUMBER OF BLOCKS IN THE COLUMN, LB=LENGTH OF BLOCKS,
C   LB5= A FIFTH OF BLOCK LENGTH
3   S1=SGN*SIN(T)
    C1=COS(T)
    S2=SGN*SIN(T1)
    C2=COS(T1)
    S3=SGN*SIN(T2)
    C3=COS(T2)
    S4=SGN*SIN(T3)
    C4=COS(T3)
    DO 4 NC=1,K
      NB=5**(NC-1)
      LB=N/NB
      LB5=LB/5
C   THE BLOCKS OF THE COLUMN ARE LOOPED OVER, IS=SEQUENCE NUMBER
C   AT THE START OF THE BLOCK, IFF=SEQUENCE NUMBER A FIFTH WAY
C   THROUGH BLOCK.
      DO 4 IB=1,NB
        IS=(IB-1)*LB+1
        IFF=(IB-1)*LB+LB5
C   THE VALUES OF THE COLUMN ARE COMPUTED USING THE TRIG. FUNC-
C   TIONS FROM THE PRE-GENERATED LIST.
      DO 4 I=IS,IFF
        SAA=SGN*SI(IB)
        SAA1=SGN*SI1(IB)
        SAA2=SGN*SI2(IB)
        SAA3=SGN*SI3(IB)

```

HPDPRE.FOR

```

      I2=I+LB5
C   THE FFT COEFFICIENTS ARE UNSCRAMBLD INTO NORMAL ORDER
      I3=I+2*LB5
      I4=I+3*LB5
      I5=I+4*LB5
      QR1=XR(I2)*CO(IB)-XI(I2)*SAA
      QI1=XR(I2)*SAA+XI(I2)*CO(IB)
      QR2=XR(I3)*CO1(IB)-XI(I3)*SA1
      QI2=XR(I3)*SA1+XI(I3)*CO1(IB)
      QR3=XR(I4)*CO2(IB)-XI(I4)*SA2
      QI3=XR(I4)*SA2+XI(I4)*CO2(IB)
      QR4=XR(I5)*CO3(IB)-XI(I5)*SA3
      QI4=XR(I5)*SA3+XI(I5)*CO3(IB)
      XR(I2)=XR(I)+QR1*C1-QI1*S1+QR2*C2-QI2*S2+QR3*C3-QI3*S3+QR4*C4-QI4*
+ S4
      XI(I2)=XI(I)+QR1*S1+QI1*C1+QR2*S2+QI2*C2+QR3*S3+QI3*C3+QR4*S4+QI4*
+ C4
      XR(I3)=XR(I)+QR1*C2-QI1*S2+QR2*C4-QI2*S4+QR3*C1-QI3*S1+QR4*C3-QI4*
+ S3
      XI(I3)=XI(I)+QR1*S2+QI1*C2+QR2*S4+QI2*C4+QR3*S1+QI3*C1+QR4*S3+QI4*
+ C3
      XR(I4)=XR(I)+QR1*C3-QI1*S3+QR2*C1-QI2*S1+QR3*C4-QI3*S4+QR4*C2-QI4*
+ S2
      XI(I4)=XI(I)+QR1*S3+QI1*C3+QR2*S1+QI2*C1+QR3*S4+QI3*C4+QR4*S2+QI4*
+ C2
      XR(I5)=XR(I)+QR1*C4-QI1*S4+QR2*C3-QI2*S3+QR3*C2-QI3*S2+QR4*C1-QI4*
+ S1
      XI(I5)=XI(I)+QR1*S4+QI1*C4+QR2*S3+QI2*C3+QR3*S2+QI3*C2+QR4*S1+QI4*
+ C1
      XR(I)=XR(I)+QR1+QR2+QR3+QR4
      XI(I)=XI(I)+QI1+QI2+QI3+QI4
4   CONTINUE
C   THE FFT COEFFICIENTS ARE UNSCRAMBLD INTO NORMAL ORDER
      DO 6 I=1,N05
      DO 6 L=1,5
      IR=IRB(I)+L
      II=I+(L-1)*N05
      IF (IR.LE.II) GO TO 6
      ZR=XR(IR)
      ZI=XI(IR)
      XR(IR)=XR(II)
      XI(IR)=XI(II)
      XR(II)=ZR
      XI(II)=ZI
6   CONTINUE
      RETURN
      END
C
      SUBROUTINE FFT23(K1,K2,N,N1,N2,N02,N03,SGN,LIST,XR,XI,IRB,IRBD,CO,
+ SI,COD,SID,WR,WI,WRD,WID,W)
      DIMENSION XR(1),XI(1),IRB(1),IRBD(1),CO(1),SI(1),COD(1),

```

HPDPRE.FOR

```

      +SID(1),WR(1),WI(1),WRD(1),WID(1),W(1)
C *****
C   THE SUBROUTINE FFT23 JOINS THE FFT2 AND THE FFT3. WHEN
C   USED, BOTH, FFT2 AND FFT3, SUBROUTINES MUST BE PRESENT.
C   N IS THE NUMBER OF MEASUREMENTS WHICH IS OF THE FORM
C    $N=(2**K1)*(3**K2)$ .  $N1=2**K1$  AND  $N2=3**K2$ ,  $NO2=N1/2$ ,
C    $NO3=N2/3$ .  $SGN=-1$  IF THE FOURIER TRANSFORM IS TO BE PER-
C   FORMED. IF  $LIST=0$ , THE REVERSED-DIGIT LIST AND THE AS-
C   SOCIATED SINES AND COSINES ARE COMPUTED. IF  $LIST=1$ , THE
C   LIST IS NOT COMPUTED. INSTEAD THE LIST COMPUTED ON THE
C   PREVIOUS CALL IS USED. XR IS THE REAL PART AND XI IS THE
C   IMAGINARY PART OF THE COMPLEX SEQUENCE TO BE TRANSFORMED.
C   AFTER THE FFT23 IS PERFORMED XR AND XI ARE THE REAL AND
C   IMAGINARY PART OF THE TRANSFORMED DATA. IRB, CO, AND SI
C   ARE WORK VECTORS.
C *****
      TPI=6.283185307
      XN=N
      DO 30 I=1,N1
        FI=I-1
        TPNF=TPI*FI/XN
        DO 10 J=1,N2
          L=J-1
          WRD(J)=XR(N1*L+I)
          WID(J)=XI(N1*L+I)
10      CONTINUE
        CALL FFT3(K2,N2,NO3,SGN,LIST,WRD,WID,IRBD,COD,SID)
        DO 20 J=1,N2
          L=J-1
          XJ=J-1
          THET=SGN*TPNF*XJ
          C=COS(THET)
          S=SIN(THET)
          XR(N1*L+I)=WRD(J)*C-WID(J)*S
          XI(N1*L+I)=WRD(J)*S+WID(J)*C
20      CONTINUE
30      CONTINUE
        DO 60 I=1,N2
          M=I-1
          DO 40 J=1,N1
            WR(J)=XR(N1*M+J)
            WI(J)=XI(N1*M+J)
40      CONTINUE
        CALL FFT2(K1,N1,NO2,SGN,LIST,WR,WI,IRB,CO,SI)
        DO 50 J=1,N1
          XR(N1*M+J)=WR(J)
          XI(N1*M+J)=WI(J)
50      CONTINUE
60      CONTINUE
        DO 70 I=1,N
          W(I)=XR(I)

```

HPDPRE.FOR

```

70  CONTINUE
    DO 80 I=1,N2
    DO 80 J=1,N1
    XR((J-1)*N2+I)=W(N1*(I-1)+J)
80  CONTINUE
    DO 90 I=1,N
    W(I)=XI(I)
90  CONTINUE
    DO 95 I=1,N2
    DO 95 J=1,N1
    XI((J-1)*N2+I)=W(N1*(I-1)+J)
95  CONTINUE
    RETURN
    END

```

C

```

    SUBROUTINE FFT25(K1,K2,N,N1,N2,NO2,NO5,SGN,LIST,XR,XI,IRB,IRBD,
+CO,SI,COD,SID,WR,WI,WRD,WID,W,CO1,SI1,CO2,SI2,CO3,SI3)
    DIMENSION XR(1),XI(1),IRB(1),IRBD(1)
    DIMENSION CO(1),SI(1),COD(1),SID(1),WR(1),WI(1)
    DIMENSION WRD(1),WID(1),W(1),CO1(1),SI1(1)
    DIMENSION CO2(1),CO3(1),SI2(1),SI3(1)

```

C *****

```

C    THE SUBROUTINE FFT25 JOINS THE FFT2 AND THE FFT5.  HENCE
C    WHEN USED THE SUBROUTINES FFT2 AND FFT5 MUST BE PRESENT.
C    N IS THE NUMBER OF MEASUREMENTS, WHICH IS OF THE FORM
C    N=(2**K1)*(5**K2).  N1=2**K1, N2=5**K2, NO2=N1/2, NO5=N2/5,
C    SGN=-1, IF THE FOURIER TRANSFORM IS TO BE PERFORMED.  IF
C    LIST=0 THE REVERSED-DIGIT LIST AND THE ASSOCIATED SINES AND
C    COSINES ARE COMPUTED.  IF LIST=1, THE LIST IS NOT COMPUTED.
C    INSTEAD THE LIST COMPUTED ON THE PREVIOUS CALL IS USED.
C    XR, XI ARE THE REAL AND THE IMAGINARY PARTS RESPECTIVELY
C    OF THE INPUT COMPLEX SEQUENCE OF NUMBERS.  AFTER THE FFT25
C    IS PERFORMED XR AND XI ARE THE REAL AND IMAGINARY PARTS
C    OF THE TRANSFORMED DATA.  IRB, IRBD, CO, COD, SID, WR, WI,
C    WRD, WID, W, CO1, SI1, CO2, SI2, CO3, AND SI3 ARE WORK
C    VECTORS.

```

C *****

```

    TPI=6.283185307
    XN=N
    DO 30 I=1,N1
    FI=I-1
    TPNF=TPI*FI/XN
    DO 10 J=1,N2
    L=J-1
    WRD(J)=XR(N1*L+I)
    WID(J)=XI(N1*L+I)
10  CONTINUE
    CALL FFT5(K2,N2,NO5,SGN,LIST,WRD,WID,IRBD,COD,SID,CO1,SI1,
+CO2,SI2,CO3,SI3)
    DO 20 J=1,N2
    L=J-1

```

HPDPRE.FOR

```

      XJ=J-1
      THET=SGN*TPNF*XJ
      C=COS(THET)
      S=SIN(THET)
      XR(N1*L+I)=WRD(J)*C-WID(J)*S
      XI(N1*L+I)=WRD(J)*S+WID(J)*C
20  CONTINUE
30  CONTINUE
      DO 60 I=1,N2
      M=I-1
      DO 40 J=1,N1
      WR(J)=XR(N1*M+J)
      WI(J)=XI(N1*M+J)
40  CONTINUE
      CALL FFT2(K1,N1,N02,SGN,LIST,WR,WI,IRB,CO,SI)
      DO 50 J=1,N1
      XR(N1*M+J)=WR(J)
      XI(N1*M+J)=WI(J)
50  CONTINUE
60  CONTINUE
      DO 70 I=1,N
      W(I)=XR(I)
70  CONTINUE
      DO 80 I=1,N2
      DO 80 J=1,N1
      XR((J-1)*N2+I)=W(N1*(I-1)+J)
80  CONTINUE
      DO 90 I=1,N
      W(I)=XI(I)
90  CONTINUE
      DO 95 I=1,N2
      DO 95 J=1,N1
      XI((J-1)*N2+I)=W(N1*(I-1)+J)
95  CONTINUE
      RETURN
      END
C
      SUBROUTINE FFT35(K1,K2,N,N1,N2,N03,N05,SGN,LIST,XR,XI,IRB,IRBD,
+CO,SI,COD,SID,WR,WI,WRD,WID,W,CO1,SI1,CO2,SI2,CO3,SI3)
      DIMENSION XR(1),XI(1),IRB(1),IRBD(1),CO(1),SI(1),COD(1)
      DIMENSION SID(1),WR(1),WI(1),WRD(1),WID(1),W(1),CO1(1)
      DIMENSION SI1(1),CO2(1),SI2(1),CO3(1),SI3(1)
C *****
C   THE SUBROUTINE FFT35 JOINS THE FFT3 AND THE FFT5.  HENCE WHEN
C   IS USED, THE SUBROUTINES FFT3 AND FFT5 MUST BE PRESENT.  N IS
C   THE NUMBER OF MEASUREMENTS, WHICH IS OF THE FORM
C   N=(3**K1)*(5**K2).  N1=3**K1, N2=5**K2, N03=N1/3, N05=N2/5,
C   SGN=-1, IF THE FOURIER TRANSFORM IS TO BE PERFORMED, AND SGN=+1
C   IF THE INVERSE FOURIER TRANSFORM IS TO BE PERFORMED.  IF LIST=0,
C   THE REVERSED-DIGIT LIST AND THE ASSOCIATED SINES AND COSINES.
C   IF LIST=1, THE LIST IS NOT COMPUTED.  INSTEAD THE LIST COMPUTED

```

HPDPRE.FOR

C ON THE PREVIOUS CALL IS USED. XR, XI, ARE THE REAL AND THE
C IMAGINARY PARTS RESPECTIVELY OF THE INPUT COMPLEX SEQUENCE OF
C NUMBERS. AFTER THE FFT35 IS PERFORMED XR AND XI ARE THE REAL
C AND IMAGINARY PARTS OF THE TRANSFORMED DATA. IRB, IRBD, CO, SI,
C COD, SID, WR, WI, WRD, WID, W, CO1, SI1, CO2, SI2, CO3, AND SI3
C ARE WORK VECTORS.

C *****

TPI=6.283185307

XN=N

DO 30 I=1,N1

FI=I-1

TPNF=TPI*FI/XN

DO 10 J=1,N2

L=J-1

WRD(J)=XR(N1*L+I)

WID(J)=XI(N1*L+I)

10 CONTINUE

CALL FFT5(K2,N2,N05,SGN,LIST,WRD,WID,IRBD,COD,SID,CO1,SI1,CO2,
+SI2,CO3,SI3)

DO 20 J=1,N2

L=J-1

XJ=J-1

THET=SGN*TPNF*XJ

C=COS(THET)

S=SIN(THET)

XR(N1*L+I)=WRD(J)*C-WID(J)*S

XI(N1*L+I)=WRD(J)*S+WID(J)*C

20 CONTINUE

30 CONTINUE

DO 60 I=1,N2

M=I-1

DO 40 J=1,N1

WR(J)=XR(N1*M+J)

WI(J)=XI(N1*M+J)

40 CONTINUE

CALL FFT3(K1,N1,M03,SGN,LIST,WR,WI,IRB,CO,SI)

DO 50 J=1,N1

XR(N1*M+J)=WR(J)

XI(N1*M+J)=WI(J)

50 CONTINUE

60 CONTINUE

DO 70 I=1,N

W(I)=XR(I)

70 CONTINUE

DO 80 I=1,N2

DO 80 J=1,N1

XR((J-1)*N2+I)=W(N1*(I-1)+J)

80 CONTINUE

DO 90 I=1,N

W(I)=XI(I)

90 CONTINUE

HPDPRE.FOR

```

      DO 95 I=1,N2
      DO 95 J=1,N1
      XI((J-1)*N2+I)=W(N1*(I-1)+J)
95  CONTINUE
      RETURN
      END

C
      SUBROUTINE FFT235(K1,N,N1,N2,N3,N02,N03,N05,SGN,LIST,XR,XI,
+IRB,IRBD,IRBT,CO,SI,COD,SID,COT,SIT,WR,WI,WRD,WID,WRT,WIT,CO1,SI1,
+CO2,SI2,CO3,SI3,W)
      DIMENSION XR(1),XI(1),IRB(1),IRBD(1),IRBT(1),CO(1),SI(1),
+COD(1),SID(1),COT(1),SIT(1),WR(1),WI(1),WRD(1),WID(1),
+WRT(1),WIT(1),CO1(1),SI1(1),CO2(1),SI2(1),CO3(1),
+SI3(1),W(1)
C *****
C   THIS SUBROUTINE JOINS THE FFT2, THE FFT3, AND THE FFT5. WHEN
C   IT IS USED, THE SUBROUTINES FFT2, FFT3, AND FFT5, MUST BE PRESENT.
C   N IS THE NUMBER OF MEASUREMENTS. N IS OF THE FORM N=(2**K1)*
C   (3**K2)*(5**K3). N1=2**K1, N2=3**K2, N3=5**K3. N02=N1/2,
C   N03=N2/3, N05=N3/5. SGN=-1, IF THE FOURIER TRANSFORM IS TO BE
C   PERFORMED, AND SGN=+1 IF THE INVERSE FOURIER TRANSFORM IS TO BE
C   PERFORMED. IF LIST=0, THE REVERSED-DIGIT LIST AND THE ASSOCIATED
C   SINES AND COSINES ARE COMPUTED. IF LIST=1, THE LIST IS NOT
C   COMPUTED. INSTEAD, THE LIST COMPUTED ON THE PREVIOUS CALL IS
C   USED. XR IS THE REAL PART AND XI IS THE IMAGINARY PART OF THE
C   COMPLEX SEQUENCE TO BE TRANSFORMED. AFTER THE FFT235 IS PERFORMED
C   XR AND XI ARE THE REAL AND IMAGINARY PARTS OF THE TRANSFORMED DATA.
C   IRB, IRBD, IRBT, CO, SI, COD, SID, COT, SIT, WR, WI, WRD, WID, WRT,
C   WIT, CO1, CO2, CO3, SI1, SI2, SI3, AND W ARE WORK VECTORS.
C *****
      TPI=6.283185307
      XN=N
      XN2=N2
      XN3=N3
      DO 65 I=1,N1
      XL=I-1
      TP1=(XL*TPI)/XN
      TP3=TP1*XN3
      DO 30 J=1,N2
      XJ=J-1
      TP2=TPI*XJ/(XN2*XN3)+TP1
      DO 10 K=1,N3
      WRT(K)=XR(N1*N2*(K-1)+N1*(J-1)+I)
      WIT(K)=XI(N1*N2*(K-1)+N1*(J-1)+I)
10  CONTINUE
      CALL FFT5(K3,N3,N05,SGN,LIST,WRT,WIT,IRBT,COT,SIT,CO1,SI1,CO2,SI2,
+CO3,SI3)
      DO 20 K=1,N3
      XK=K-1
      THET=SGN*TP2*XK
      C=COS(THET)

```

HPDPRE.FOR

```

      S=SIN(THET)
      XR(N1*N2*(K-1)+N1*(J-1)+I)=WRT(K)*C-WIT(K)*S
      XI(N1*N2*(K-1)+N1*(J-1)+I)=WIT(K)*C+WRT(K)*S
20  CONTINUE
30  CONTINUE
      DO 60 K=1,N3
      KM=K-1
      DO 40 J=1,N2
      WRD(J)=XR(N1*N2*KM+N1*(J-1)+I)
      WID(J)=XI(N1*N2*KM+N1*(J-1)+I)
40  CONTINUE
      CALL FFT3(K2,N2,N03,SGN,LIST,WRD,WID,IRBD,COD,SID)
      DO 50 J=1,N2
      XJ=J-1
      THET=SGN*TP3*XJ
      C=COS(THET)
      S=SIN(THET)
      XR(N1*N2*KM+N1*(J-1)+I)=WRD(J)*C-WID(J)*S
      XI(N1*N2*KM+N1*(J-1)+I)=WRD(J)*S+WID(J)*C
50  CONTINUE
60  CONTINUE
65  CONTINUE
      DO 90 K=1,N3
      KM=K-1
      DO 80 J=1,N2
      KJ=J-1
      DO 70 I=1,N1
      WR(I)=XR(N1*N2*KM+N1*KJ+I)
      WI(I)=XI(N1*N2*KM+N1*KJ+I)
70  CONTINUE
      CALL FFT2(K1,N1,N02,SGN,LIST,WR,WI,IRB,CO,SI)
      DO 80 I=1,N1
      XR(N1*N2*KM+N1*KJ+I)=WR(I)
      XI(N1*N2*KM+N1*KJ+I)=WI(I)
80  CONTINUE
90  CONTINUE
C   THE FFT COEFFICIENTS ARE UNSCRAMBLD INTO NORMAL ORDER
      DO 100 I=1,N
      W(I)=XR(I)
100  CONTINUE
      DO 110 I=1,N1
      DO 110 J=1,N2
      MJ=J-1
      DO 110 K=1,N3
      MK=K-1
      XR((I-1)*N2*N3+N3*MJ+K)=W(N1*N2*MK+N1*MJ+I)
110  CONTINUE
      DO 120 I=1,N
      W(I)=XI(I)
120  CONTINUE
      DO 130 I=1,N1

```

HPDPRE.FOR

```

      DO 130 J=1,N2
      MJ=J-1
      DO 130 K=1,N3
      MK=K-1
      XI((I-1)*N2*N3+N3*MJ+K)=W(N1*N2*MK+N1*MJ+I)
130  CONTINUE
      RETURN
      END

C
      SUBROUTINE NSCORE(V,N,IPOS,N,NMAX,NMAX05,B,IB,IT)
C *****
C      STANDARD NORMAL EQUIVALENTS OR "NORMAL SCORES" ARE
C      COMPUTED FOR A VECTOR OF DATA.  THE RANKED DATA IS
C      ALSO PREPARED.
C
C      INPUT:
C      V(I)=VECTOR OF DATA, 1<=I<=N
C      N=NUMBER OF ACTUAL COMPONENTS FOR V(I)
C      NMAX=DIMENSION DECLARED FOR V(I), SHOULD BE >= N
C      NMAX05=SMALLEST INTEGER GREATER THAN OR EQUAL TO NMAX/5
C
C      OUTPUT:
C      V(I)=VECTOR OF NORMAL SCORES
C      R(I)=RANKED ORIGINAL DATA
C      IPOS(I)=INTEGER GIVING THE POSITION OF THE DATUM WITH
C              RANK I WITHIN THE ORIGINAL DATA VECTOR.
C
C      WORK VECTORS:
C      B(J)=WORK VECTOR WITH DIMENSION 5
C      IB(J)=WORK VECTOR WITH DIMENSION NMAX/5
C      IT(J)=WORK VECTOR WITH DIMENSION NMAX/5
C
C      SUBROUTINE WRITTEN BY L. E. BORGMAN, UNIVERSITY OF WYOMING
C *****
      DIMENSION V(NMAX),R(NMAX)
      DIMENSION IPOS(NMAX),B(5),IB(NMAX05),IT(NMAX05)
      DO 1 I=1,N
      R(I)=V(I)
      IPOS(I)=I
1  CONTINUE
      CALL FQSORT(R,IPOS,B,N,IB,IT,NI)
      DO 2 I=1,N
      P=I/(N+1.0)
      CALL RNORM(P,VV)
      V(IPOS(I))=VV
2  CONTINUE
      RETURN
      END

C
C
      SUBROUTINE RNORM(U,Z)

```

HPDPRE.FOR

```

C *****
C   SUBROUTINE OUTPUTS Z, GIVEN F(Z)=U FOR THE STANDARD
C   NORMAL
C *****
      P=U
      IF (U.GT.0.5) P=1.0-U
      T=SQRT(ALOG(1.0/(P*P)))
      FN=(T*0.010328+0.802853)*T+2.515517
      FD=((T*0.001308+0.189269)*T+1.432788)*T+1.0
      Z=T-FN/FD
      IF (U.LT.0.5) Z=-Z
      RETURN
      END

C
      SUBROUTINE FOSORT(A,IA,B,N,IB,IT,NI)
C*****
C   SUBROUTINE SORTS THE COMPONENTS OF THE REAL VECTOR A(J),
C       1.LE.J.LE.N, IN ORDER OF INCREASING SIZE.
C
C   A(N) IS THE VECTOR TO BE SORTED, IA IS A VECTOR OF INTEGERS
C       CO-SORTED WITH A(N).
C   B(S) IS A WORK ARRAY.
C   IB(N/S) AND IT(N/S) ARE INTEGER WORK VECTORS OF DIMENSION
C       AT LEAST N/S
C   NI IS A WORK INTEGER
C
C   PROGRAM WRITTEN BY L. E. BORGMAN, LARAMIE, WYOMING
C
C   THE SUBROUTINE USES SUBSIDIARY SUBROUTINES: FMID, FRANK,
C       FSWAP, AND FSORT.
C*****
      DIMENSION IB(1),IT(1),IA(1),A(1),B(1)
      IB(1)=1
      IT(1)=N
      NI=1
10  IF(NI.EQ.0) RETURN
      CALL FOSORT(A,IA,B,IB,IT,NI)
      GO TO 10
      END

C
      SUBROUTINE FSWAP(A,B,IA,IB)
C *****
C   SUBSIDIARY SUBR TO FOSORT
C *****
      T=A
      A=B
      B=T
      I=IA
      IA=IB
      IB=I
      RETURN

```

HPDPRE.FOR

END

C

SUBROUTINE FRANK(NS,NT,A,IA)

C *****

C SUBSIDIARY SUBR TO SUBR FQSORT

C *****

DIMENSION IA(1),A(1)

NTOT=NT-NS+1

IF (NTOT.EQ.1) RETURN

NTM=NT-1

DO 10 I=NS,NTM

IP=I+1

DO 10 J=IP,NT

IF (A(I).GT.A(J)) CALL FSWAP(A(I),A(J),IA(I),IA(J))

10 CONTINUE

RETURN

END

C

SUBROUTINE FMID(NS,NT,A,B,IA)

C *****

C SUBSIDIARY SUBR TO FQSORT

C *****

DIMENSION A(1),B(1),IA(1)

I1=NS

I5=NT

I2=I1+(NT-NS)/4

I3=I1+(NT-NS)/2

I4=I1+(3*(NT-NS))/4

B(1)=A(I1)

B(2)=A(I2)

B(3)=A(I3)

B(4)=A(I4)

B(5)=A(I5)

DO 1 J=1,4

JP=J+1

DO 1 JJ=JP,5

IF (B(J).LE.B(JJ)) GO TO 1

Z=B(JJ)

B(JJ)=B(J)

B(J)=Z

1 CONTINUE

AA=A(I1)

II=IA(I1)

BB=B(3)

IF(BB.EQ.A(I2)) CALL FSWAP(AA,A(I2),II,IA(I2))

IF(BB.EQ.A(I3)) CALL FSWAP(AA,A(I3),II,IA(I3))

IF(BB.EQ.A(I4)) CALL FSWAP(AA,A(I4),II,IA(I4))

IF(BB.EQ.A(I5)) CALL FSWAP(AA,A(I5),II,IA(I5))

A(I1)=AA

IA(I1)=II

RETURN

HPDPRE.FOR

```

      END
C
      SUBROUTINE FSORT(A,IA,B,IB,IT,NI)
C *****
C      SUBSIDIARY SUBR TO FQSORT
C *****
      DIMENSION IB(1),IT(1),IA(1),A(1),B(1)
      NS=IB(1)
      NT=IT(1)
      NTOT=NT-NS+1
      IF(NTOT.GE.10) GO TO 10
      CALL FRANK(NS,NT,A,IA)
      NI=NI-1
      IF(NI.GE.1) GO TO 17
      IB(1)=0
      IT(1)=0
      RETURN
17 DO 18 JK=1,NI
      JP=JK+1
      IB(JK)=IB(JP)
      IT(JK)=IT(JP)
18 CONTINUE
      IB(JP)=0
      IT(JP)=0
      RETURN
10 CONTINUE
      CALL FMID(NS,NT,A,B,IA)
      JS=NS
      JE=NT
      JM=NS
      IGO=1
11 CONTINUE
      IF(IGO) 13,12,12
12 CONTINUE
      IF(A(JE).GE.A(JM).AND.JE.GT.JM) JE=JE-1
      IF(A(JE).GE.A(JM).AND.JE.GT.JM) GO TO 12
      IF(JE.LE.JM) GO TO 14
      CALL FSWAP(A(JE),A(JM),IA(JE),IA(JM))
      JM=JE
14 IGO=-1
      GO TO 15
13 CONTINUE
      IF(A(JS).LE.A(JM).AND.JS.LT.JM) JS=JS+1
      IF(A(JS).LE.A(JM).AND.JS.LT.JM) GO TO 13
      IF(JS.GE.JM) GO TO 16
      CALL FSWAP(A(JS),A(JM),IA(JS),IA(JM))
      JM=JS
16 IGO=1
15 IF(JE.GT.JS) GO TO 11
      NI=NI-1
      IBOT=JM-NS

```

HPDPRE. FOR

```

      IUP=NT-JM
      IF (IBOT.EQ.0.OR.IUP.EQ.0) GO TO 19
      DO 20 JK=1,NI
      J=NI-JK+2
      JJ=J+1
      IB(JJ)=IB(J)
      IT(JJ)=IT(J)
20  CONTINUE
      IB(1)=NS
      IT(1)=JM-1
      IB(2)=JM+1
      IT(2)=NT
      NI=NI+2
      RETURN
19  NI=NI+1
      IF (IUP.EQ.0) GO TO 21
      IB(1)=JM+1
      IT(1)=NT
      RETURN
21  IB(1)=NS
      IT(1)=JM-1
      RETURN
      END

C
      SUBROUTINE XSPEC (VC,M,MMAX,N,NSIM,NSMAX,NO2P1,MNO2P1,DT,BW,
&SPEC,NTOP,XR,XI,YR,YI)
C *****
C      COMPUTE CROSS-SPECTRAL MATRIX OF DENSITIES OF VECTOR TIME
C      SERIES [VC(I,J); I=1,M; J=1,NSIM]
C
C      N = LENGTH OF TIME SERIES
C
C      M = NUMBER OF COMPONENTS IN DATA RANDOM VECTOR AT EACH TIME
C      MMAX = MAXIMUM SIZE OF M, FOR DIMENSIONING ONLY
C      DT = TIME INCREMENT
C
C      NSIM = LENGTH OF TIME SERIES TO BE SIMULATED SUBSEQUENTLY.
C      THIS SHOULD BE AN EXACT INTEGER MULTIPLE OF N.
C      IT MAY BE EQUAL TO N, TWO TIMES N, THREE TIMES N,
C      OR SOME OTHER INTEGER MULTIPLE OF N. IN ANY CASE,
C      NSIM SHOULD BE SELECTED TO BE REPRESENTABLE IN THE
C      FORM (2**I)*(3**J)*(5**K) FOR POSITIVE INTEGERS
C      (I,J,K). THIS IS REQUIRED FOR THE FFT SOFTWARE USED.
C      NSMAX = MAXIMUM VALUE OF NSIM, FOR DIMENSIONING ONLY
C      NO2P1 = (NSIM/2)+1, IN FORTRAN INTEGER ARITHMETIC
C      MNO2P1 = (NSMAX/2)+1, IN FORTRAN INTEGER ARITHMETIC
C
C      [VC(I,J); I=1,M; J=1,NSIM] = COMPLEX-VALUED ARRAY WITH REAL
C      PART OF THE I-TH TIME SERIES STORED IN [VC(I,J); J=1,NI].
C      IMAGINARY PARTS ARE ALL ZERO INITIALLY. LATER, THE
C      DISCRETE FOURIER TRANSFORM OF THE TIME SERIES, FOR

```

HPDPRE.FOR

```

C      J=1,NO2P1 IS STORED IN THE FIRST PART OF VC
C      IN THIS POSITION.
C      [SPEC(I,J,K);I=1,M;J=1,M;K=1,NO2P1] = CROSS-SPECTRAL DENSITIES.
C      ABOVE THE MAIN (I,J)-DIAGONAL, THE CO-SPECTRA AT FREQUENCY,
C      K/(N*DT), ARE STORED. BELOW THE MAIN (I,J)-DIAGONAL, THE
C      QUAD-SPECTRA AT FREQUENCY, K/(N*DT), ARE STORED.
C      NTOP = LARGEST J-VALUE AT WHICH THERE IS SIGNIFICANT VARIANCE.
C      BW = BAND WIDTH OF SMOOTHING FOR SPECTRAL ESTIMATES, IN HZ.
C      XR,XI,YR,YI = WORK VECTORS
C
C      PROGRAM WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING.
C      *****
C      DIMENSION SPEC(MMAX,MMAX,MNO2P1),XR(NSMAX),XI(NSMAX),YR(NSMAX)
C      DIMENSION YI(NSMAX)
C      COMPLEX VC(MMAX,NSMAX),CC
C
C      ***** CONSTANTS ***
C      DF=1.0/(NSIM*DT)
C      IDOC=0
C
C      ***** FFT THE TIME SERIES ***
C      DO 6 I=1,M
C        DO 1 J=1,NSIM
C          XR(J)=REAL(VC(I,J))*DT
C          XI(J)=0.0
C        1 CONTINUE
C        IF (IDOC.NE.0) THEN
C          WRITE (2,100)
C100      FORMAT (/1X,'XR=V')
C          WRITE (2,101) (XR(J),J=1,NSIM)
C101      FORMAT (1X,10E12.5)
C          WRITE (2,102)
C102      FORMAT (/1X,'XI=0.0')
C          WRITE (2,101) (XI(J),J=1,NSIM)
C        END IF
C        CALL FT235(NSIM,-1.0,0,XR,XI)
C        IF (IDOC.NE.0) THEN
C          WRITE (2,103)
C103      FORMAT (/1X,'XR AFTER FFT OF V')
C          WRITE (2,101) (XR(J),J=1,NSIM)
C          WRITE (2,104)
C104      FORMAT (/1X,'XI AFTER FFT OF V')
C          WRITE (2,101) (XI(J),J=1,NSIM)
C        END IF
C        DO 7 J=1,NO2P1
C          IF (J.LE.NTOP) THEN
C            VC(I,J)=CMPLX(XR(J),XI(J))
C          ELSE
C            ***** SCREEN OUT NOISE ABOVE CUTOFF FREQ. ***
C            VC(I,J)=0.0
C          END IF

```

HPDPRE.FOR

7 CONTINUE

6 CONTINUE

```

C      ***** COMPUTE RAW SPECTRAL LINES *****
      DO 8 II=1,M
        DO 9 I=1,II
          DO 3 J=1,NO2P1
C            ***** NOTE: DIVIDE BY N*DT INSTEAD OF NSIM*DT *****
C            ***** TO ADJUST FOR PADDING WITH ZEROS *****
            CC=CONJG(VC(I,J))*VC(II,J)/(N*DT)
            XR(J)=REAL(CC)
            XI(J)=0.0
            IF (I.NE.II) THEN
              YR(J)=-AIMAG(CC)
              YI(J)=0.0
            END IF
            IF (J.GT.1) THEN
              XR(NSIM-J+2)=XR(J)
              XI(NSIM-J+2)=0.0
              IF (I.NE.II) THEN
                YR(NSIM-J+2)=YR(J)
                YI(NSIM-J+2)=0.0
              END IF
            END IF
3          CONTINUE
          IF (IDOC.NE.0) THEN
            WRITE (2,109)
109          FORMAT (/1X,'RAW CD-SPECTRAL LINES BEFORE FFT')
            WRITE (2,101) (XR(J),J=1,NSIM)
            IF (I.NE.II) THEN
              WRITE (2,110)
110          FORMAT (/1X,'RAW QUAD-SPECTRAL LINES BEFORE FFT')
              WRITE (2,101) (YR(J),J=1,NSIM)
            END IF
          END IF
        END IF
      END IF

C      ***** INVERSE FFT SPECTRAL LINES TO LAG DOMAIN *****
      CALL FT235(NSIM,1.0,0,XR,XI)
      IF (I.NE.II) THEN
        CALL FT235(NSIM,1.0,0,YR,YI)
      END IF

C      ***** MULTIPLY BY TRANSFORM OF GAUSSIAN FILTER *****
      DO 2 J=1,NO2P1
        JJ=J-1
        EX=3.14159265*BW*BW*DF*DF*JJ*JJ*DT*DT
        IF (EX.LT.18.0) THEN
          W=EXP(-EX)
        ELSE
          W=0.0
        END IF

```

HPDPRE.FOR

```

C      ***** NOTE: DIVISION BY NSIM HERE TAKES CARE ***
C      ***** OF DF*DT REQUIRED IN THE COMBINED      ***
C      ***** PREVIOUS AND NEXT FFT                  ***
      XR(J)=XR(J)*W/NSIM
      XI(J)=XI(J)*W/NSIM
      IF (I.NE.II) THEN
        YR(J)=YR(J)*W/NSIM
        YI(J)=YI(J)*W/NSIM
      END IF
      IF (J.GT.1) THEN
        XR(NSIM-J+2)=XR(J)
        XI(NSIM-J+2)=-XI(J)
        IF (I.NE.II) THEN
          YR(NSIM-J+2)=YR(J)
          YI(NSIM-J+2)=-YI(J)
        END IF
      END IF

2      CONTINUE
      IF (IDOC.NE.0) THEN
        WRITE (2,105)
105      FORMAT (/1X,'BEFORE FFT, XR PART')
        WRITE (2,101) (XR(J),J=1,NSIM)
        WRITE (2,106)
106      FORMAT (/1X,'BEFORE FFT, XI PART')
        WRITE (2,101) (XI(J),J=1,NSIM)
        IF (I.NE.II) THEN
          WRITE (2,305)
305      FORMAT (/1X,'BEFORE FFT, YR PART')
          WRITE (2,101) (YR(J),J=1,NSIM)
          WRITE (2,306)
306      FORMAT (/1X,'BEFORE FFT, YI PART')
          WRITE (2,101) (YI(J),J=1,NSIM)
        END IF
      END IF

C      ***** FFT BACK TO FREQUENCY TO OBTAIN SPECTRUM ***
      CALL FT235(NSIM,-1.0,0,XR,XI)
      IF (I.NE.II) THEN
        CALL FT235(NSIM,-1.0,0,YR,YI)
      END IF
      IF (IDOC.NE.0) THEN
        WRITE (2,107)
107      FORMAT (/1X,'SPECTRAL DENSITY, XR PART')
        WRITE (2,101) (XR(J),J=1,NSIM)
        WRITE (2,108)
108      FORMAT (/1X,'SPECTRAL DENSITY, XI PART')
        WRITE (2,101) (XI(J),J=1,NSIM)
        IF (I.NE.II) THEN
          WRITE (2,307)

```

HPDPRE.FOR

```

307          FORMAT (/1X,'SPECTRAL DENSITY, YR PART')
             WRITE (2,101) (YR(J),J=1,NSIM)
             WRITE (2,308)
308          FORMAT (/1X,'SPECTRAL DENSITY, VI PART')
             WRITE (2,101) (YI(J),J=1,NSIM)
             END IF
             END IF
             DO 5 J=1,NO2P1
C             ***** ADD IN THE SPECTRA FOR THIS PIECE OF DATA ***
C             ***** TO THE ACCUMULATED SUM FROM PREVIOUS ***
C             ***** CALLS TO SUBR. XSP. A DIVISION TO ***
C             ***** AVERAGE WILL BE MADE AFTER LAST CALL ***
             IF (J.LE.NTOP) THEN
                 SPEC(I,II,J)=SPEC(I,II,J)+XR(J)
                 IF (I.NE.II) THEN
                     SPEC(II,I,J)=SPEC(II,I,J)+YR(J)
                 END IF
             ELSE
C             ***** SCREEN OUT NOISE ABOVE CUTOFF FREQ. ***
                 SPEC(I,II,J)=0.0
                 IF (I.NE.II) THEN
                     SPEC(II,I,J)=0.0
                 END IF
             END IF
5             CONTINUE
9             CONTINUE
8             CONTINUE
             RETURN
             END

C
C          SUBROUTINE EIG(A,EIVC,N,IC,MV,MVMAX,NMAX2)
C          *****
C          COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC
C          MATRIX
C          A = ORIGINAL MATRIX (OF ORDER N) IS DESTROYED IN COMPUTATION.
C          THE ELEMENTS OF A ARE ORDERED INTO A VECTOR OF UPPER
C          TRIANGULAR ELEMENTS AS SHOWN IN THE SEQUENCING PATTERN
C
C          1  2  4  7 11 16      THE EIGENVALUES ARE CONTAINED
C          3  5  8 12 17      IN THE POSITIONS OF THE DESTROYED
C          6  9 13 18 ETC.    MATRIX A CORRESPONDING TO THE
C          10 14 19      MAIN DIAGONAL. THAT IS: 1,3,6,10,
C          15 20      15,21, ETC.
C          21
C          THE J-TH EIGENVALUE IS IN POSITION J*(J+1)/2
C
C          EIVC = MATRIX OF EIGENVECTORS ,STORED IN A SINGLE VECTOR,
C          ONE AFTER THE OTHER, IN SAME SEQUENCE AS EIGENVALUES.
C          THE J-TH EIGENVECTOR WOULD OCCUPY THE SEQUENCE
C          OF COMPONENTS OF EIVC, STARTING AT COMPONENT
C          (J-1)*N+1 AND ENDING AT COMPONENT J*N

```

HPDPRE.FOR

```

C      N = ORDER OF MATRICES A AND EIVEC
C      NMAX = MAXIMUM SIZE OF N, FOR DIMENSIONING ONLY. (NOTE: NMAX
C              IS NOT REQUIRED AS INPUT, BUT DETERMINES MVMAX
C              AND NMAX2.)
C      MV = THE DIMENSION OF THE VECTOR A, =N*(N+1)/2
C      MVMAX = MAXIMUM SIZE OF MV, FOR DIMENSIONING ONLY
C              = NMAX*(NMAX+1)/2
C      NMAX2 = THE DIMENSION OF THE VECTOR EIVEC, = NMAX**2
C      IC = 0  COMPUTE EIGENVALUES AND EIGENVECTORS
C      IC = 1  COMPUTE EIGENVALUES ONLY (EIVEC NEED NOT BE
C              DIMENSIONED BUT MUST STILL APPEAR IN CALLING
C              SEQUENCE)
C
C      DIAGONALIZATION METHOD FROM 'MATHEMATICAL
C      METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND
C      H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7
C
C      FOR A DOUBLE PRECISION VERSION, REMOVE THE LEADING 'C'
C      FROM THE FOLLOWING STATEMENT:
C      DOUBLE PRECISION A,EIVEC,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,
C      &          COSX2,SINCS,      CONST
C
C      ALSO CHANGE: SQRT TO DSQRT, ABS TO DABS, AND CONST IN
C      FIRST STATEMENT BELOW TO 1.0D-12
C      *****
C      DIMENSION A(MVMAX),EIVEC(NMAX2)
C
C      ***** GENERATE IDENTITY MATRIX ***
C      CONST=1.0E-6
C      IF (IC.NE.1) THEN
C          IQ=-N
C          DO 20 J=1,N
C              IQ=IQ+N
C          DO 20 I=1,N
C              IJ=IQ+I
C              EIVEC(IJ)=0.0
C              IF (I.EQ.J) THEN
C                  EIVEC(IJ)=1.0
C              END IF
C      20  CONTINUE
C      END IF
C      ANORM=0.0
C      DO 35 I=1,N
C      DO 35 J=I,N
C          IF (I.NE.J) THEN
C              IA=I+(J*J-J)/2
C              ANORM=ANORM+A(IA)*A(IA)
C          END IF
C      35 CONTINUE
C      IF (ANORM.GT.0.0) THEN
C          ANORM=1.414*SQRT(ANORM)
C          ANRMX=ANORM*CONST/N

```

HPDPRE.FOR

```

      IND=0
      THR=ANORM
45    THR=THR/FLOAT(N)
50    L=1
55    M=L+1
60    MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
      IF (ABS(A(LM)).GE.THR) THEN
        IND=1
        LL=L+LQ
        MM=M+MQ
        X=0.5*(A(LL)-A(MM))
        Y=-A(LM)/ SQRT(A(LM)*A(LM)+X*X)
        IF (X.LT.0.0) THEN
          Y=-Y
        END IF
        SINX=Y/ SQRT(2.0*(1.0+( SQRT(1.0-Y*Y))))
        SINX2=SINX*SINX
        COSX= SQRT(1.0-SINX2)
        COSX2=COSX*COSX
        SINCS =SINX*COSX
        ILQ=N*(L-1)
        IMQ=N*(M-1)
      DO 125 I=1,N
        IQ=(I*I-I)/2
        IF (I.NE.L) THEN
          IF (I.NE.M) THEN
            IF (I.LT.M) THEN
              IM=I+MQ
            ELSE IF (I.GT.M) THEN
              IM=M+IQ
            END IF
            IF (I.LT.L) THEN
              IL=I+LQ
            ELSE
              IL=L+IQ
            END IF
            X=A(IL)*COSX-A(IM)*SINX
            A(IM)=A(IL)*SINX+A(IM)*COSX
            A(IL)=X
          END IF
        END IF
        IF (IC.NE.1) THEN
          ILR=ILQ+I
          IMR=IMQ+I
          X=EIVC(ILR)*COSX-EIVC(IMR)*SINX
          EIVC(IMR)=EIVC(ILR)*SINX+EIVC(IMR)*COSX
          EIVC(ILR)=X
        END IF
125   CONTINUE

```

HPDPRE.FOR

```

      X=2.0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
    END IF
    IF (M.NE.N) THEN
      M=M+1
      GO TO 60
    ELSE
      IF (L.NE.N-1) THEN
        L=L+1
        GO TO 55
      ELSE
        IF (IND.EQ.1) THEN
          IND=0
          GO TO 50
        ELSE
          IF (THR.GT.ANRMX) GO TO 45
        END IF
      END IF
    END IF
    END IF
    END IF
    IQ=-N
    DO 185 I=1,N
      IQ=IQ+N
      LL=I+(I*I-I)/2
      JQ=N*(I-2)
    DO 185 J=1,N
      JQ=JQ+N
      MM=J+(J*J-J)/2
      IF (A(LL).LT.A(MM)) THEN
        X=A(LL)
        A(LL)=A(MM)
        A(MM)=X
        IF (IC.NE.1) THEN
          DO 180 K=1,N
            ILR=IQ+K
            IMR=JQ+K
            X=EIVEC(ILR)
            EIVEC(ILR)=EIVEC(IMR)
            EIVEC(IMR)=X
          180      CONTINUE
        END IF
      END IF
    185 CONTINUE
    RETURN
  END
C
  SUBROUTINE ABC(NTM,NTMAX,TSO,IBND,ZTS,UTS,AA,BB,CC,IBOUND,
```

HPDPRE.FOR

```

      &Z,U,A,B,C)
C *****
C   DEVELOP IBOUND, U, Z, A, B, AND C JETS USING SUBR. TAIL
C   AS NEEDED.
C
C   NTM = LENGTH OF TIME SERIES
C   NTMAX = MAXIMUM ALLOWED LENGTH OF TIME SERIES (FOR
C           DIMENSIONING PURPOSES)
C
C   LET K=1 FOR WAVE HEIGHT, K=2 FOR WAVE PERIOD,
C       K=3 FOR SINE DIRECTION, K=4 FOR COS DIRECTION:
C   [TSO(K,J);J=1,NTM] = RANKED TIME SERIES DATA FOR TYPE K
C   IBND(K,1) = 0, IF POPULATION K IS UNBOUNDED TO LEFT
C   IBND(K,1) = 1, IF POPULATION K IS BOUNDED TO LEFT
C   IBND(K,2) = 0, IF POPULATION K IS UNBOUNDED TO RIGHT
C   IBND(K,2) = 1, IF POPULATION K IS BOUNDED TO RIGHT
C
C   [ZTS(K,I,J);I=1,3;J=1,2] = Z-ARRAY FOR TYPE K DATA AS
C       DEFINED IN SUBR. TAIL.
C   [UTS(K,I,J);I=1,3;J=1,2] = U-ARRAY FOR TYPE K DATA AS
C       DEFINED IN SUBR. TAIL.
C   [AA(K,I);I=1,2],[BB(K,I);I=1,2],[CC(K,I);I=1,2] = A(I),
C       B(I),C(I) VALUES DEFINED IN SUBR. TAIL, FOR
C       TYPE K DATA.
C   [ITAIL(J);J=1,2] = INTERNAL FLAG USED IN CALL TO TAIL
C
C   DMAX = LARGEST DIRECTION IN CENTERED UNRAVELED VALUES
C   DMIN = SMALLEST DIRECTION IN CENTERED UNRAVELED VALUES
C       (SEE SUBROUTINE UNRAVL FOR MORE DETAILS.)
C
C   WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING
C *****
C   DIMENSION TSO(4,NTMAX),ZTS(4,3,2),UTS(4,3,2),IBND(4,2)
C   DIMENSION AA(4,2),BB(4,2),CC(4,2),ITAIL(2)
C   DIMENSION IBOUND(2),Z(3,2),U(3,2),A(2),B(2),C(2)
C
C   ***** ASSIGN APPROPRIATE TAIL CONSTANTS FOR H, P, AND D ***
C   ***** FOR WAVE HEIGHT: ***
C       IBND(1,1)=1
C       IBND(1,2)=0
C       UTS(1,3,1)=0.0
C       ZTS(1,3,1)=0.0
C
C   ***** FOR WAVE PERIOD: ***
C       IBND(2,1)=1
C       IBND(2,2)=0
C       UTS(2,3,1)=0.0
C       ZTS(2,3,1)=0.0
C
C   ***** FOR SINE OF WAVE DIRECTION: ***

```

HPDPRE.FOR

```

IBND(3,1)=1
IBND(3,2)=1
UTS(3,3,1)=0.0
UTS(3,3,2)=1.0
ZTS(3,3,1)=-1.0
ZTS(3,3,2)=1.0

```

C ***** FOR COSINE OF WAVE DIRECTION: ***

```

IBND(4,1)=1
IBND(4,2)=1
UTS(4,3,1)=0.0
UTS(4,3,2)=1.0
ZTS(4,3,1)=-1.0
ZTS(4,3,2)=1.0

```

DO 5 K=1,4

```

ITAIL(1)=1
ITAIL(2)=1

```

C ***** DEVELOP REST OF TAIL CONSTANTS FOR LOWER TAIL ***

IF (IBND(K,1).EQ.1.AND.TSO(K,1).EQ.ZTS(K,3,1)) THEN

C ***** SMALLEST VALUE EQUALS LOWER BOUND ***

```

AA(K,1)=0.0
BB(K,1)=0.0
CC(K,1)=1.0
ZTS(K,1,1)=ZTS(K,3,1)
ZTS(K,2,1)=ZTS(K,3,1)
UTS(K,1,1)=UTS(K,3,1)
UTS(K,2,1)=UTS(K,3,1)
ITAIL(1)=0
GO TO 850

```

ELSE IF (IBND(K,1).EQ.1.AND.TSO(K,1).NE.ZTS(K,3,1)) THEN

C ***** BOUNDED BELOW BUT SMALLEST DATA VALUE ABOVE THE ***

C ***** LOWER BOUND. THE THIRD VALUE WAS PREVIOUSLY ***

C ***** SET TO LOWER BOUND, SO SET SECOND VALUE ***

C ***** EQUAL TO SMALLEST DATA VALUE ***

```

ZTS(K,2,1)=TSO(K,1)
UTS(K,2,1)=1.0/(NTM+1.0)
ZLAST=TSO(K,1)
IQUIT=0

```

DO 1 J=2,NTM

C ***** LOOK FOR NEXT VALUE DIFFERENT FROM PREVIOUS ***

JJ=J

IF (TSO(K,J).NE.ZLAST) THEN

C ***** VALUE DIFFERENT FROM PREVIOUS FOUND. SET ***

C ***** FIRST VALUE EQUAL TO THE FOUND VALUE. ***

```

ZTS(K,1,1)=TSO(K,J)
UTS(K,1,1)=J/(NTM+1.0)
IQUIT=1
GO TO 3

```

END IF

1 CONTINUE

HPDPRE.FOR

```

3      CONTINUE
      IF (JJ.EQ.NTM.AND.IQUIT.EQ.0) THEN
C          ***** NO VALUE DIFFERENT FROM PREVIOUS FOUND,      ***
C          ***** SO USE BIGGEST VALUE FOR FIRST ZTS.          ***
          ZTS(K,1,1)=TSO(K,NTM)+0.1
          UTS(K,1,1)=NTM/(NTM+1.0)
      END IF
      ELSE IF (IBND(K,1).EQ.0) THEN
C          ***** UNBOUNDED TO LEFT. SET THIRD ZTS TO SMALLEST ***
C          ***** DATA VALUE                                    ***
          ZTS(K,3,1)=TSO(K,1)
          UTS(K,3,1)=1.0/(NTM+1.0)
          ZLAST=TSO(K,1)
          IQUIT=0
          DO 6 J=2,NTM
C          ***** SEARCH FOR NEXT VALUE DIFFERENT FROM LAST. ***
              JJ=J
              IF (TSO(K,J).NE.ZLAST) THEN
C                  ***** DIFFERENT VALUE FOUND. SET SECOND ZTS ***
C                  ***** EQUAL TO THIS FOUND VALUE.              ***
                  ZTS(K,2,1)=TSO(K,J)
                  UTS(K,2,1)=J/(NTM+1.0)
                  ZLAST=TSO(K,J)
                  IQUIT=1
                  GO TO 7
              END IF
6          CONTINUE
7          CONTINUE
          IF (JJ.EQ.NTM.AND.IQUIT.EQ.0) THEN
C          ***** NO DIFFERENT VALUE FOUND. SET SECOND ZTS      ***
C          ***** EQUAL TO LARGEST DATA VALUE, AND FIT          ***
C          ***** EXPONENTIAL LOWER TAIL.                        ***
          ZTS(K,2,1)=TSO(K,NTM)+0.1
          UTS(K,2,1)=NTM/(NTM+1.0)
          DEN=ALOG(UTS(K,3,1))-ALOG(UTS(K,2,1))
          AAA=ZTS(K,3,1)-ZTS(K,2,1)
          AAA=AAA/DEN
          BBB=ZTS(K,2,1)*ALOG(UTS(K,3,1))
          BBB=BBB-ZTS(K,3,1)*ALOG(UTS(K,2,1))
          ZTS(K,1,1)=BBB/DEN
          AA(K,1)=0.0
          BB(K,1)=SQRT(AAA/2.0)
          CC(K,1)=2.0
          UTS(K,1,1)=0.0
          ITAIL(1)=0
          GO TO 850
          END IF
          IQUIT=0
          DO 10 J=JJ+1,NTM
C          ***** SEARCH FOR STILL ANOTHER VALUE DIFFERENT FROM ***
C          ***** THE PREVIOUS.                                    ***

```

HPDPRE.FOR

```

      JJ=J
      IF (TSO(K,J).NE.ZLAST) THEN
C        ***** DIFFERENT VALUE FOUND.  SET FIRST ZTS EQUAL ***
C        ***** TO THIS FOUND VALUE. ***
          ZTS(K,1,1)=TSO(K,J)
          UTS(K,1,1)=J/(NTM+1.0)
          IQUIT=1
          GO TO 11
      END IF
10     CONTINUE
11     CONTINUE
      IF (JJ.EQ.NTM.AND.IQUIT.EQ.0) THEN
C        ***** DIFFERENT VALUE NOT FOUND.  SET FIRST ZTS ***
C        ***** EQUAL TO THE LARGEST DATA VALUE. ***
          ZTS(K,1,1)=TSO(K,NTM)+0.1
          UTS(K,1,1)=NTM/(NTM+1.0)
      END IF
      END IF
850    CCNTINUE

C      ***** DEVELOP REST OF TAIL CONSTANTS FOR UPPER TAIL ***
C      IF (IBND(K,2).EQ.1.AND.TSO(K,NTM).EQ.ZTS(K,3,2)) THEN
C        ***** LARGEST VALUE EQUALS UPPER BOUND ***
          AA(K,2)=0.0
          BB(K,2)=0.0
          CC(K,2)=1.0
          ZTS(K,1,2)=ZTS(K,3,2)
          ZTS(K,2,2)=ZTS(K,3,2)
          UTS(K,1,2)=UTS(K,3,2)
          UTS(K,2,2)=UTS(K,3,2)
          ITAIL(2)=0
          GO TO 851
      ELSE IF (IBND(K,2).EQ.1.AND.TSO(K,NTM).NE.ZTS(K,3,2)) THEN
C        ***** BOUNDED ABOVE BUT LARGEST DATA VALUE BELOW THE ***
C        ***** UPPER BOUND.  THE THIRD VALUE WAS PREVIOUSLY ***
C        ***** SET TO UPPER BOUND, SO SET SECOND VALUE ***
C        ***** EQUAL TO LARGEST DATA VALUE. ***
          ZTS(K,2,2)=TSO(K,NTM)
          UTS(K,2,2)=NTM/(NTM+1.0)
          ZLAST=TSO(K,NTM)
          IQUIT=0
          DO 2 J=2,NTM
C            ***** LOOK FOR NEXT VALUE DIFFERENT FROM PREVIOUS ***
              JJ=J
              IF (TSO(K,NTM-J+1).NE.ZLAST) THEN
C                ***** VALUE DIFFERENT FROM PREVIOUS FOUND. SET ***
C                ***** FIRST VALUE EQUAL TO THE FOUND VALUE. ***
                  ZTS(K,1,2)=TSO(K,NTM-J+1)
                  UTS(K,1,2)=(NTM-J+1.0)/(NTM+1.0)
                  IQUIT=1

```

HPDPRE.FOR

```

                GO TO 36
            END IF
2       CONTINUE
36      CONTINUE
        IF (JJ.EQ.NTM.AND.IQUIT.EQ.0) THEN
C          ***** NO VALUE DIFFERENT FROM PREVIOUS FOUND,      ***
C          ***** SO USE SMALLEST VALUE FOR FIRST ZTS.        ***
            ZTS(K,1,2)=TSO(K,1)-0.1
            UTS(K,1,2)=1.0/(NTM+1.0)
        END IF
        ELSE IF (IBND(K,2).EQ.0) THEN
C          ***** UNBOUNDED TO RIGHT. SET THIRD ZTS TO LARGEST ***
C          ***** DATA VALUE.                                  ***
            ZTS(K,3,2)=TSO(K,NTM)
            UTS(K,3,2)=NTM/(NTM+1.0)
            ZLAST=TSO(K,NTM)
            IQUIT=0
            DO 8 J=2,NTM
C          ***** SEARCH FOR NEXT VALUE DIFFERENT FROM LAST. ***
                JJ=J
                IF (TSO(K,NTM-J+1).NE.ZLAST) THEN
C          ***** DIFFERENT VALUE FOUND. SET SECOND ZTS      ***
C          ***** EQUAL TO THIS FOUND VALUE.                  ***
                    ZTS(K,2,2)=TSO(K,NTM-J+1)
                    UTS(K,2,2)=(NTM-J+1.0)/(NTM+1.0)
                    ZLAST=TSO(K,NTM-J+1)
                    IQUIT=1
                    GO TO 9
                END IF
            CONTINUE
8       CONTINUE
9       CONTINUE
        IF (JJ.EQ.NTM.AND.IQUIT.EQ.0) THEN
C          ***** NO DIFFERENT VALUE FOUND. SET SECOND ZTS    ***
C          ***** EQUAL TO SMALLEST DATA VALUE, AND FIT      ***
C          ***** EXPONENTIAL UPPER TAIL.                     ***
            ZTS(K,2,2)=TSO(K,1)-0.1
            UTS(K,2,2)=1.0/(NTM+1.0)
            DEN=ALOG(1.0-UTS(K,3,2))-ALOG(1.0-UTS(K,2,2))
            AAA=ZTS(K,2,2)-ZTS(K,3,2)
            AAA=AAA/DEN
            BBB=ZTS(K,2,2)*ALOG(1.0-UTS(K,3,2))
            BBB=BBB-ZTS(K,3,2)*ALOG(1.0-UTS(K,2,2))
            ZTS(K,1,2)=BBB/DEN
            AA(K,2)=0.0
            BB(K,2)=SQRT(AAA/2.0)
            CC(K,2)=2.0
            UTS(K,1,2)=0.0
            ITAIL(2)=0
            GO TO 851
        END IF
        IQUIT=0

```

HPDPRE.FOR

```

      DO 12 J=JJ+1,NTM
C      ***** SEARCH FOR STILL ANOTHER VALUE DIFFERENT FROM ***
C      ***** THE PREVIOUS ZTS VALUE. ***
      JJ=J
      IF (TSO(K,NTM-J+1).NE.ZLAST) THEN
C      ***** DIFFERENT VALUE FOUND. SET FIRST ZTS EQUAL***
C      ***** TO THIS FOUND VALUE. ***
      ZTS(K,1,2)=TSO(K,NTM-J+1)
      UTS(K,1,2)=(NTM-J+1.0)/(NTM+1.0)
      IQUIT=1
      GO TO 14
      END IF
12     CONTINUE
14     CONTINUE
      IF (JJ.EQ.NTM.AND.IQUIT.EQ.0) THEN
C      ***** DIFFERENT VALUE NOT FOUND. SET FIRST ZTS ***
C      ***** EQUAL TO THE SMALLEST DATA VALUE. ***
      ZTS(K,1,2)=TSO(K,1)-0.1
      UTS(K,1,2)=1.0/(NTM+1.0)
      END IF
      END IF
851    CONTINUE

      IF (ITAIL(1).EQ.1.OR.ITAIL(2).EQ.1) THEN
C      ***** SET UP USE OF SUBR. TAIL ***
      IBOUND(1)=IBND(K,1)
      IBOUND(2)=IBND(K,2)
      DO 15 J=1,2
        DO 16 I=1,3
          Z(I,J)=ZTS(K,I,J)
          U(I,J)=UTS(K,I,J)
16        CONTINUE
15      CONTINUE
      CALL TAIL(IBOUND,Z,U,A,B,C,ITAIL)
      DO 17 J=1,2
        IF (ITAIL(J).EQ.1) THEN
          AA(K,J)=A(J)
          BB(K,J)=B(J)
          CC(K,J)=C(J)
17        CONTINUE
      END IF
      END IF
5     CONTINUE
      RETURN
      END

C      SUBROUTINE TAIL(IBOUND,Z,U,A,B,C,ITAIL)
C      *****
C      FITS CURVE TO RIGHT AND LEFT TAILS OF A DISTRIBUTION
C
C      IF (ITAIL(1).EQ.0) THEN LEFT TAIL IS NOT COMPUTED

```

HPDPRE.FOR

```

C      IF (ITAIL(1).EQ.1) THEN LEFT TAIL IS COMPUTED
C      IF (ITAIL(2).EQ.0) THEN RIGHT TAIL IS NOT COMPUTED
C      IF (ITAIL(2).EQ.1) THEN RIGHT TAIL IS COMPUTED
C
C      IBOUND(1) = 0, IF THE DISTRIBUTION FUNCTION DOMAIN IS
C      UNBOUNDED TO THE LEFT.
C      = 1, IF THE DISTRIBUTION FUNCTION DOMAIN IS
C      BOUNDED TO THE LEFT.
C      IBOUND(2) = 0, IF THE DISTRIBUTION FUNCTION DOMAIN IS
C      UNBOUNDED TO THE RIGHT.
C      = 1, IF THE DISTRIBUTION FUNCTION DOMAIN IS
C      BOUNDED TO THE RIGHT.
C      Z(1,1) = LINK VALUE FOR LEFT TAIL.
C      Z(1,2) = LINK VALUE FOR RIGHT TAIL.
C      Z(2,1) = INTERMEDIATE VALUE FOR THE LEFT TAIL.
C      Z(2,2) = INTERMEDIATE VALUE FOR THE RIGHT TAIL
C      Z(3,1) = LEFTMOST VALUE IN FITTING VALUE FOR LEFT TAIL.
C      (EQUALS LOWER LIMIT IF IBOUND(1) = 1).
C      Z(3,2) = RIGHTMOST VALUE FOR RIGHT TAIL.
C      (EQUALS UPPER LIMIT IF IBOUND(2) = 1.)
C
C      U(I,J) = DISTRIBUTION FUNCTION VALUE ASSOCIATED WITH
C      Z(I,J). (NOTE: IF IBOUND(1) = 1, THEN
C      U(3,1) = 0. ALSO, IF IBOUND(2) = 1, THEN
C      U(3,2) = 1.0.)
C
C      THE EQUATION FITTED IS
C
C       $Z - Z_1 = (A + B * T) * C$ , FOR RIGHT TAIL
C       $Z - Z_1 = -(A + B * T) * C$ , FOR LEFT TAIL
C
C      WHERE
C
C       $Z_1$  = LINK VALUE OF Z
C       $T = \sqrt{-2 * \text{ALOG}(U)}$ , UNBOUNDED AT LEFT
C       $T = U - U(1,1)$ , IF BOUNDED ON THE LEFT.
C       $T = \sqrt{-2 * \text{ALOG}(1 - U)}$ , UNBOUNDED TO THE RIGHT
C       $T = U(1,2) - U$ , IF BOUNDED TO THE RIGHT
C
C      THE REQUIRED COEFFICIENTS ARE REPORTED OUT AS
C
C      A(1),B(1),C(1) = LEFT TAIL VALUES
C      A(2),B(2),C(2) = RIGHT TAIL VALUES
C
C      WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING
C      *****
C      DIMENSION IBOUND(2),Z(3,2),U(3,2),A(2),B(2),C(2)
C      DIMENSION ITAIL(2)
C
C      DO 1 J=1,2
C      IF (ITAIL(J).EQ.1) THEN

```

HPDPRE.FOR

```

Z1=Z(1,J)
Z2=Z(2,J)
Z3=Z(3,J)
IF (IBOUND(J).EQ.0) THEN
  IF (J.EQ.1) THEN
    T1=SQRT(-2.0*ALOG(U(1,1)))
    T2=SQRT(-2.0*ALOG(U(2,1)))
    T3=SQRT(-2.0*ALOG(U(3,1)))
  ELSE IF (J.EQ.2) THEN
    T1=SQRT(-2.0*ALOG(1.0-U(1,2)))
    T2=SQRT(-2.0*ALOG(1.0-U(2,2)))
    T3=SQRT(-2.0*ALOG(1.0-U(3,2)))
  END IF
ELSE IF (IBOUND(J).EQ.1) THEN
  T1=0.0
  IF (J.EQ.1) THEN
    T2=U(1,1)-U(2,1)
    T3=U(1,1)
  ELSE IF (J.EQ.2) THEN
    T2=U(2,2)-U(1,2)
    T3=1.0-U(1,2)
  END IF
END IF
PRINT*, 'Z1,Z2,Z3= ',Z1,Z2,Z3
PRINT*, 'T1,T2,T3= ',T1,T2,T3
W=(Z2-Z1)/(Z3-Z1)
WW=(T2-T1)/(T3-T1)
PRINT*, 'C: W,WW= ',W,WW
C(J)=ALOG(W)/ALOG(WW)
PRINT*, 'C= ',C(J)
W=ABS(Z3-Z1)
WW=ABS(T3-T1)
PRINT*, 'BB: W,WW= ',W,WW
BB=ALOG(W)/C(J)-ALOG(WW)
B(J)=EXP(BB)
A(J)=-B(J)*T1
END IF
1 CONTINUE
RETURN
END

```

F: PROGRAM LISTING FOR HPDSIM.FOR

HPDSIM.FOR

PROGRAM THPDSM

```

C *****
C   TEST OF SUBR. HPDSIM
C *****

C   ***** PROGRAM PARAMETER VALUES ***
PARAMETER (NSMAX=600,MNO2P1=301,NMAX=300)
PARAMETER (IYB=1956,IMB=9,IYE=1958,IME=6)
PARAMETER (NYR=20)

C   ***** PROGRAM DIMENSION STATEMENTS ***
DIMENSION TS(4,NSMAX),TSO(4,NMAX),SCALE(2)
DIMENSION EM(8,8,MNO2P1),NEV(MNO2P1)
DIMENSION XR(NSMAX),XI(NSMAX),AA(4,2),BB(4,2),CC(4,2)
DIMENSION IBND(4,2),ZTS(4,3,2),UTS(4,3,2)
DIMENSION IBOUND(2),Z(3,2),U(3,2),A(2),B(2),C(2)
DIMENSION VSIM(NSMAX),RSIM(NSMAX),R(NMAX)

COMPLEX TSN(4,NSMAX)
CHARACTER TITLE*50,ALPH*34,ALINE*50

DIMENSION WHEEL(5,NSMAX,3),IDAT(3),MODA(12),AVG(4,3),VR(4,3)
DIMENSION MONTHS(12),IYEARS(NYR),CUTOFF(12),IJM(12),IJY(NYR)
DIMENSION IDH(250)

C   ***** PARAMETER AND DIMENSION STATEMENTS REQUIRED FOR FFT ***
C   FFT PARAMETER VALUES:
C       MAXN=LARGEST VALUE TO BE USED FOR N (LENGTH OF SEQUENCE)
C       MAXN2=LARGEST POWER OF 2 LESS THAN OR EQUAL TO MAXN
C       MAXN3=LARGEST POWER OF 3 LESS THAN OR EQUAL TO MAXN
C       MAXN5=LARGEST POWER OF 5 LESS THAN OR EQUAL TO MAXN
C       MAXNO2=MAXN2/2
C       MAXNO3=MAXN3/3
C       MAXNO5=MAXN5/5
C
C   FFT COMMON BLOCK *****
PARAMETER (MAXN=1200,MAXN2=1024,MAXN3=729,MAXN5=625)
PARAMETER (MAXNO2=512,MAXNO3=243,MAXNO5=125)
C   *****
DIMENSION W(MAXN),WR(MAXN2),WI(MAXN2),WRD(MAXN3),WID(MAXN3)
DIMENSION WRT(MAXN5),WIT(MAXN5)
DIMENSION IRB(MAXNO2),CO(MAXNO2),SI(MAXNO2)
DIMENSION IRBD(MAXNO3),COD(MAXNO3),SID(MAXNO3)
DIMENSION IRBT(MAXNO5),COT(MAXNO5),SIT(MAXNO5)
DIMENSION CO1(MAXNO5),SI1(MAXNO5),CO2(MAXNO5),SI2(MAXNO5)
DIMENSION CO3(MAXNO5),SI3(MAXNO5)
COMMON /FFT/W,WR,WI,WRD,WID,WRT,WIT,IRB,CO,SI,IRBD,COD,SID
COMMON /FFT/IRBT,COT,SIT,CO1,SI1,CO2,SI2,CO3,SI3
C   *****
C   PARAMETER (MAXN=5000,MAXN2=4096,MAXN3=2187,MAXN5=3125)
C   PARAMETER (MAXNO2=2048,MAXNO3=729,MAXNO5=625)

```

HPDSIM.FOR

```

C  NOTE: THE ABOVE COMMON BLOCK PARAMETERS WOULD ALLOW THE USE OF
C      ANY N.LE.5000 OF THE FORM N=(2**K)*(3**L)*(5**M) FOR
C      ***** END FFT PARAMETER AND DIMENSION STATEMENTS ***

C      ***** OPEN FILES ***
      OPEN (3,FILE='THDPRE.OUT',STATUS='OLD')
      OPEN (2,FILE='HPDSIM.TAB',STATUS='UNKNOWN')
      OPEN (4,FILE='HPDSIM.DOC',STATUS='UNKNOWN')
      OPEN (5,FILE='HPDSIM.OUT',STATUS='UNKNOWN')
      OPEN (7,FILE='HPDSIM.RAM',STATUS='UNKNOWN')
      OPEN (8,FILE='HPDSIM.MAT',STATUS='UNKNOWN')
      OPEN (9,FILE='HPDSIM.VRS',STATUS='UNKNOWN')

C      ***** INITIALIZE PROGRAM CONSTANTS AND FLAGS ***
      ISEED0=67676767
      ISEQ=0
      JDEL=40
      CONV=3.14159265/180.0
      ICOUNT=0

C      ***** ESTABLISH NUMBER OF DAYS PER EACH MONTH ***
      DO 431 J=1,12
        IF (J.EQ.4.OR.J.EQ.6.OR.J.EQ.9.OR.J.EQ.11) THEN
          MODA(J)=30
        ELSE IF (J.EQ.2) THEN
          MODA(J)=28
C      ***** NOTE: LEAP YEAR ADJUSTMENT MADE IN YEAR-LOOP ***
        ELSE
          MODA(J)=31
        END IF
      431 CONTINUE

C      ***** LOOP OVER YEARS TO BE SIMULATED ***
      DO 430 JY=IYB,IYE

C      ***** SET MODA(J) FOR FEB. TO 29 FOR LEAP YEARS ***
      ILEAP=JY-(JY/4)*4
      IF (ILEAP.EQ.0) MODA(2)=29

C      ***** READ OUTPUT FROM HPDPRE ***
      REWIND (3)

C      ***** READ OVERALL DEFINING PARAMETERS ***
      READ (3,'(A50)') TITLE
      READ (3,420) NNYR,NNY,NMO
420  FORMAT (14X,3I5)
      READ (3,421) (IYEARS(J),J=1,NNYR)
421  FORMAT (9X,20I3)
      READ (3,422) (MONTHS(J),J=1,12)
422  FORMAT (9X,12I3)
      READ (3,423) (CUTOFF(J),J=1,12)
423  FORMAT (9X,5F12.6)

```

HPDSIM.FOR

```

      READ (3,425) (IJY(J),J=1,NYR)
425   FORMAT (6X,20I3)
      READ (3,426) (IJM(J),J=1,12)
426   FORMAT (6X,20I3)

```

```

      IF (JY.EQ.IYB) THEN

```

```

C      ***** WRITE OVERALL DEFINING PARAMETERS ***
      WRITE (2,'(A50)') TITLE
      WRITE (2,520) NYR,NNY,NMO
520   FORMAT (1X,'NYR,NNY,NMO= ',3I5)
      WRITE (2,521) (IYEARS(J),J=1,NYR)
521   FORMAT (1X,'IYEARS= ',20I3)
      WRITE (2,522) (MONTHS(J),J=1,12)
522   FORMAT (1X,'MONTHS= ',12I3)
      WRITE (2,523) (CUTOFF(J),J=1,12)
523   FORMAT (1X,'CUTOFF= ',5F12.6)
      WRITE (2,525) (IJY(J),J=1,NYR)
525   FORMAT (1X,'IJY= ',20I3)
      WRITE (2,526) (IJM(J),J=1,12)
526   FORMAT (1X,'IJM= ',20I3)
      WRITE (5,'(A50)') TITLE
      WRITE (5,520) NYR,NNY,NMO
      WRITE (5,521) (IYEARS(J),J=1,NYR)
      WRITE (5,522) (MONTHS(J),J=1,12)
      WRITE (5,523) (CUTOFF(J),J=1,12)
      WRITE (5,525) (IJY(J),J=1,NYR)
      WRITE (5,526) (IJM(J),J=1,12)
      END IF

```

```

C      ***** READ THE MONTHLY DATA ***
      DO 427 JM=1,12
      IDATE=JY*100+JM
      IF (MONTHS(JM).NE.0) THEN
601   WRITE (*,601) JY
      FORMAT (/1X,'YEAR BEING SIMULATED= ',I4)
      WRITE (*,600) JM
600   FORMAT (1X,'      MONTH BEING READ IN = ',I2)
      READ (3,'(A34)') ALPH
      READ (3,'(A34)') ALPH
      READ (3,424) JJM
424   FORMAT (26X,I3)
      READ (3,310) (SCALE(K),K=1,2)
310   FORMAT (16X,2F15.5)
      READ (3,110) N,M,NTM,NTOP,NSIM
110   FORMAT (19X,5I8)
      READ (3,111) DT
111   FORMAT (9X,6I5.5)
      READ (3,112) BW,FCO,FRACT
112   FORMAT (15X,3G15.5)
      READ (3,'(A34)') ALPH
      READ (3,114) (TSO(1,J),J=1,N)

```

HPDSIM.FOR

```

114      FORMAT (1X,5G15.5)
        READ (3,'(A34)') ALPH
        READ (3,114) (TS0(2,J),J=1,N)
        READ (3,'(A34)') ALPH
        READ (3,114) (TS0(3,J),J=1,N)
        READ (3,'(A34)') ALPH
        READ (3,114) (TS0(4,J),J=1,N)
        DO 3 J=1,NTOP
            READ (3,105) JTEMP,FTEMP,NEV(J)
105      FORMAT (12X,I5,F12.6,I5)
            DO 4 I=1,8
                READ (3,106) (EM(I,II,J),II=1,NEV(J))
106      FORMAT (1X,8G12.6)
            4      CONTINUE
            3      CONTINUE
            DO 70 K=1,4
                READ (3,300) (IBND(K,J),J=1,2)
300      FORMAT (19X,2I5)
                READ (3,301) (AA(K,J),J=1,2)
301      FORMAT (17X,2G15.5)
                READ (3,302) (BB(K,J),J=1,2)
302      FORMAT (17X,2G15.5)
                READ (3,303) (CC(K,J),J=1,2)
303      FORMAT (17X,2G15.5)
                DO 72 I=1,3
                    READ (3,304) (ZTS(K,I,J),J=1,2)
304      FORMAT (27X,2G15.5)
                    READ (3,305) (UTS(K,I,J),J=1,2)
305      FORMAT (27X,2G15.5)
                72      CONTINUE
                70      CONTINUE
            END IF

C      ***** SIMULATE MONTH IF WITHIN SPECIFIED INTERVAL ***
            IF ((JY.EQ.IYB.AND.JM.GE.IMB.AND.MONTHS(JM).NE.0).OR.
&      (JY.GT.IYB.AND.JY.LT.IYE.AND.MONTHS(JM).NE.0).OR.
&      (JY.EQ.IYE.AND.JM.LE.IME.AND.MONTHS(JM).NE.0)) THEN

                WRITE (*,642)
1642      FORMAT (1X,' (SIMULATION REQUESTED FOR THIS MONTH)')
                ISEQ=ISEQ+1
                IPOS=ISEQ-(ISEQ/3)*3+1
                IF (IPOS.GT.1) THEN
                    IPM=IPOS-1
                ELSE IF (IPOS.EQ.1) THEN
                    IPM=3
                END IF

C      ***** CREATE NORM SCORE TIME SER. FOR H,P, AND D ***

                NTOPP=60

```

HPDSIM.FOR

```

&      CALL SIMNS(NSIM,TSN,ISEEDO,EM,NEV,NTOPP,XR,XI,NSMAX,
620      MNO2P1,DT,TITLE)
      WRITE (*,620)
      FORMAT (1X,'NORMAL SCORES CREATED')

      DO 75 K=1,4
        DO 76 L=1,NSIM
          VSIM(L)=REAL(TSN(K,L))
76      CONTINUE
        DO 77 J=1,2
          IBOUND(J)=IBND(K,J)
          A(J)=AA(K,J)
          B(J)=BB(K,J)
          C(J)=CC(K,J)
          DO 78 I=1,3
            Z(I,J)=ZTS(K,I,J)
            U(I,J)=UTS(K,I,J)
78      CONTINUE
77      CONTINUE

        DO 79 L=1,N
          R(L)=TSO(K,L)
79      CONTINUE

      IF (JY.EQ.1957.AND.JM.EQ.1.AND.K.EQ.3) THEN
        IDOC=1
      ELSE
        IDOC=0
      END IF

      CALL INTERP(N,NMAX,NSIM,NSMAX,VSIM,RSIM,IBOUND,
&      Z,U,A,B,C,R,K,IDOC)

      WRITE (9,980) JY,JM,K
980      FORMAT (//1X,'JY,JM,K= ',3I8)

      DO 80 L=1,NSIM
        IF (K.LT.3) THEN
          TS(K,L)=RSIM(L)*SCALE(K)
        ELSE
          TS(K,L)=RSIM(L)
        END IF
        IF (JY.EQ.1957.AND.JM.EQ.1.AND.K.GE.3) THEN
          WRITE (4,973) L,RSIM(L),TS(3,L),TS(4,L)
973      FORMAT (15,3F12.3)
        END IF

      WRITE (9,981) L,VSIM(L),RSIM(L),TS(K,L)
981      FORMAT (1X,'L,VSIM,RSIM,TS= ',15,3F12.3)

```

HPDSIM.FOR

```

80          CONTINUE

75          CONTINUE
            WRITE (*,621)
621         FORMAT (1X,'BACKWARD INTERPOLATION FINISHED')
            IDAT(IPOS)=IDATE

C          ***** COMPUTE MONTHLY AVERAGES AND STD. DEV. ***
            DO 433 JTS=1,4
              SUM=0.0
              DO 434 JDH=1,NSIM
                SUM=SUM+TS(JTS,JDH)
434          CONTINUE
              AVG(JTS,IPOS)=SUM/NSIM
              SUM=0.0
              DO 445 JDH=1,NSIM
                TSDEV=TS(JTS,JDH)-AVG(JTS,IPOS)
                SUM=SUM+TSDEV*TSDEV
445          CONTINUE
              VR(JTS,IPOS)=SUM/(NSIM-1)

C          ***** STORE MONTHLY TIME SERIES IN WHEEL ARRAY ***
            DO 432 JDH=1,NSIM
              WHEEL(JTS,JDH,IPOS)=TS(JTS,JDH)
432          CONTINUE

            IF (JY.EQ.1957.AND.JM.EQ.2.AND.JTS.GE.3) THEN
              WRITE (4,970)
970          FORMAT (//1X,'J,JTS,WHEEL(JTS,J,1),WHEEL(JTS,J,2):')
              DO 971 J=1,NSIM
                WRITE (4,972) J,JTS,WHEEL(JTS,J,IPM),WHEEL(JTS,J,IPOS)
972          FORMAT (1X,2I5,2F12.3)
971          CONTINUE
              WRITE (4,'(1X)')
            END IF

433          CONTINUE
            WRITE (*,624)
624         FORMAT (1X,'SIMULATION STORED IN WHEEL ARRAY')

C          ***** RAMP BETWEEN MONTHS IF APPROPRIATE ***
            JYP=IDAT(IPM)/100
            JMP=IDAT(IPM)-JYP*100
            DIFMO=(JY-JYP)*12+JM-JMP

C          ***** TEST TO SEE IF PREV. SIM. IS PREV. MO. ***

```

HPDSIM.FOR

```

C          ***** RAMP, IF ANSWER IS YES          ***
          IF (DIFMO.EQ.1) THEN
625        WRITE (*,625)
          FORMAT (1X,'RAMPING')

          LMO=IDAT(IPM)-(IDAT(IPM)/100)*100
          JDHL=120+8*MODA(LMO)-JDEL
          JDHR=120-JDEL
          JDHTOP=2*JDEL
          IF (JY.EQ.1957.AND.JM.EQ.2) THEN
700        WRITE (7,700) JDHL,JDHR,JDEL,JDHTOP
          FORMAT (1X,'JDHL,JDHR,JDEL,JDHTOP= ',4I8)
          END IF
          DO 435 JDH=1,JDHTOP
            JLFT=JDHL+JDH
            JRGT=JDHR+JDH
            X=(JDH-JDEL-0.5)/(JDEL-0.5)
            PLFT=0.25*X*X*X-0.75*X+0.5
            PRGT=1.0-PLFT

            DO 436 JTS=1,4
              WLFT=WHEEL(JTS,JLFT,IPM)
              WRGT=WHEEL(JTS,JRGT,IPOS)
              AVL=AVG(JTS,IPM)
              AVR=AVG(JTS,IPOS)
              VARL=VR(JTS,IPM)
              VARR=VR(JTS,IPOS)
              AVO=PLFT*AVL+PRGT*AVR
              SDO=SQRT(PLFT*VARL+PRGT*VARR)
              IF (JTS.LE.2) THEN
                VALUE=SQRT(PLFT)*WLFT+SQRT(PRGT)*WRGT
                RATIO=SQRT(PLFT)*AVL+SQRT(PRGT)*AVR
                RATIO=AVO/RATIO
                XX=SDO/(RATIO*AVO)
                XXX=SDO/AVO
                VVO=ALOG(1.0+XX*XX)
                VVW=ALOG(1.0+XXX*XXX)
                BETA=SQRT(VVW/VVO)
                ALPHA=(ALOG(AVO)-0.5*VVO)-
                & (ALOG(RATIO*AVO)-0.5*VVW)/BETA

                VALUE=EXP(BETA*(ALOG(VALUE)-ALPHA))
              ELSE IF (JTS.GE.3) THEN
                WLDEV=(WLFT-AVL)/SQRT(VARL)
                WRDEV=(WRGT-AVR)/SQRT(VARR)
                VALUE=SQRT(PLFT)*WLDEV+SQRT(PRGT)*WRDEV
                VALUE=SDO*VALUE+AVO
              END IF
            END DO
          END DO

```

HPDSIM.FOR

```

      IF (JY.EQ.1957.AND.JM.EQ.2) THEN
        WRITE (7,701) JDH,JLFT,JRGT,X,PLFT,PRGT
701      FORMAT (1X,'JDH,JLFT,JRGT,X,PLFT,PRGT= ',3I8,3F9.2)
        WRITE (7,702) JTS,WLFT,WRGT,VALUE
702      FORMAT (1X,'JTS,WLFT,WRGT,VALUE= ',I8,3F9.2)
        WRITE (7,714) AVL,AVR,VARL,VARR
714      FORMAT (1X,'AVL,AVR,VRL,VRR= ',4F12.5)
        WRITE (7,715) AVO,SDO,RATIO
715      FORMAT (1X,'AVO,SDO,RATIO= ',3F12.5)
        WRITE (7,716) XX,XXX,VVO,VVW
716      FORMAT (1X,'XX,XXX,VVO,VVW= ',4F12.5)
        WRITE (7,717) BETA,ALPHA,VALUE
717      FORMAT (1X,'BETA,ALPHA,PREV VALUE= ',3F12.5)
        WRITE (7,718) VALUE
718      FORMAT (1X,'NEW VALUE= ',F12.5)
      END IF

      IF (X.LT.0.0) THEN
        WHEEL(JTS,JLFT,IPM)=VALUE
      ELSE IF (X.GE.0.0) THEN
        WHEEL(JTS,JRGT,IPOS)=VALUE
      END IF
436      CONTINUE
435      CONTINUE
      END IF

C      ***** WRITE SIMULATED TIME SERIES ***
      IF (ISEQ.NE.1) THEN

        LYR=IDAT(IPM)/100
        LMO=IDAT(IPM)-LYR*100
        WRITE (*,626) LYR,LMO
626      FORMAT (1X,'WRITING SIM. TIME SERIES, YR:',I5,
        &      ', MO:',I3)
        MOEND=8*MODA(LMO)

        IF (LYR.EQ.1957) THEN
          IF (LMO.LT.10) THEN
            WRITE (8,881) LMO
881      FORMAT ('hpd0',I1,'=[')
          ELSE IF (LMO.GE.10) THEN
            WRITE (8,882) LMO
882      FORMAT ('hpd',I2,'=[')
          END IF
        END IF

        DO 437 JDH=1,MOEND
          JDHM1=JDH-1
          IDH(JDH)=((JDHM1/8)+1)*100+(JDHM1-(JDHM1/8)*8)*3
          WHE1=WHEEL(1,120+JDH,IPM)

```

HPDSIM. FOR

```

WHE2=WHEEL(2,120+JDH,IPM)
WHE3=WHEEL(3,120+JDH,IPM)
WHE4=WHEEL(4,120+JDH,IPM)
DIREC=ATAN2(WHE3,WHE4)
DIREC=DIREC/CONV
IF (DIREC.LT.0.0) THEN
    DIREC=DIREC+360.0
END IF
WHEEL(5,120+JDH,IPM)=DIREC
438 WRITE (5,438) IDAT(IPM),IDH(JDH),WHE1,WHE2,DIREC
    FORMAT (1X,I6,I4,3X,4F12.5)

IF (LYR.EQ.1957) THEN
828 WRITE (8,828) (WHEEL(JTS,120+JDH,IPM),JTS=1,5)
    FORMAT (5F12.5)
END IF

437 CONTINUE
IF (LYR.EQ.1957) THEN
    WRITE (8,883)
883 FORMAT ('J')
END IF

IF ((LYR.EQ.1956.AND.LMO.EQ.12).OR.
    & (LYR.EQ.1957.AND.LMO.LT.12)) THEN
    MONOW=LMO+1
    IF (MONOW.GT.12) MONOW=MONOW-12
    IF (MONOW.EQ.1) THEN
826 WRITE (8,826) LYR,LMO
        & FORMAT ('% SIM. TIME SERIES, YR:',IS,
            ' MO:',I3)
    END IF

    IF (MONOW.LT.10) THEN
        WRITE (8,831) MONOW
831 FORMAT ('ts00',I1,'=[')
    ELSE IF (MONOW.GE.10) THEN
        WRITE (8,832) MONOW
832 FORMAT ('ts0',I2,'=[')
    END IF

    DO 829 JS=1,N
        WRITE (8,830) (TS0(L,JS),L=1,4)
830 FORMAT (4F12.3)
829 CONTINUE
        WRITE (8,833)
833 FORMAT ('J')

END IF

NSO=MOEND/10

```

HPDSIM.FOR

```

      NREM=MOEND-NSQ*10
      WRITE (2,199) TITLE
199      FORMAT (//1X,'SIMULATED (H,T,D) TIME SERIES,',
      &      ' TITLE: ',A50)
      DO 81 I=1,NSQ
          LSTART=(I-1)*10+1
          LEND=LSTART+9
          WRITE (2,'(1X)')
          WRITE (2,200) (IDH(L),L=LSTART,LEND)
200      FORMAT (1X,'(YR/MO) DA/HR: ',10I6)
          WRITE (2,201) IDAT(IPM),(WHEEL(1,120+L,IPM),
      &      L=LSTART,LEND)
201      FORMAT (1X,I6,' HEIGHT: ',10F6.2)
          WRITE (2,202) (WHEEL(2,120+L,IPM),L=LSTART,LEND)
202      FORMAT (1X,' PERIOD: ',10F6.1)
          WRITE (2,203) (WHEEL(5,120+L,IPM),L=LSTART,LEND)
203      FORMAT (1X,' DIRECTION: ',10F6.0)
          WRITE (2,260) (WHEEL(3,120+L,IPM),L=LSTART,LEND)
260      FORMAT (1X,' SIN DIRECTION: ',10F6.2)
          WRITE (2,261) (WHEEL(4,120+L,IPM),L=LSTART,LEND)
261      FORMAT (1X,' COS DIRECTION: ',10F6.2)
      81      CONTINUE
      IF (NREM.NE.0) THEN
          LSTART=MOEND-NREM+1
          LEND=MOEND
          WRITE (2,'(1X)')
          WRITE (2,200) (IDH(L),L=LSTART,LEND)
          WRITE (2,201) IDAT(IPM),(WHEEL(1,120+L,IPM),
      &      L=LSTART,LEND)
          WRITE (2,202) (WHEEL(2,120+L,IPM),L=LSTART,LEND)
          WRITE (2,203) (WHEEL(5,120+L,IPM),L=LSTART,LEND)
          WRITE (2,260) (WHEEL(3,120+L,IPM),L=LSTART,LEND)
          WRITE (2,261) (WHEEL(4,120+L,IPM),L=LSTART,LEND)
      END IF
      END IF
      ELSE
          WRITE (*,629)
629      FORMAT (1X,' (NO SIMULATION REQUESTED)')
      END IF
427      CONTINUE
430      CONTINUE

C      ***** WRITE LAST MONTH SIMULATED ***
      IPM=IPOS
      LYR=IDAT(IPM)/100
      LMO=IDAT(IPM)-LYR*100
      WRITE (*,626) LYR,'MO
      MOEND=8*MODA(LMO)
      DO 737 JDH=1,MOEND
          JDHM1=JDH-1
          IDH(JDH)=((JDHM1/8)+1)*100+(JDHM1-(JDHM1/8)*8)*3

```

HPDSIM.FOR

```

      WHE1=WHEEL(1,120+JDH,IPM)
      WHE2=WHEEL(2,120+JDH,IPM)
      WHE3=WHEEL(3,120+JDH,IPM)
      WHE4=WHEEL(4,120+JDH,IPM)
      DIREC=ATAN2(WHE3,WHE4)
      DIREC=DIREC/CONV
      IF (DIREC.LT.0.0) THEN
        DIREC=DIREC+360.0
      END IF
      WHEEL(5,120+JDH,IPM)=DIREC
      WRITE (5,438) IDAT(IPM),IDH(JDH),WHE1,WHE2,DIREC
737 CONTINUE
      NSQ=MOEND/10
      NREM=MOEND-NSQ*10
      WRITE (2,199) TITLE
      DO 781 I=1,NSQ
        LSTART=(I-1)*10+1
        LEND=LSTART+9
        WRITE (2,'(1X)')
        WRITE (2,200) (IDH(L),L=LSTART,LEND)
        WRITE (2,201) IDAT(IPM),(WHEEL(1,120+L,IPM),L=LSTART,LEND)
        WRITE (2,202) (WHEEL(2,120+L,IPM),L=LSTART,LEND)
        WRITE (2,203) (WHEEL(5,120+L,IPM),L=LSTART,LEND)
        WRITE (2,260) (WHEEL(3,120+L,IPM),L=LSTART,LEND)
        WRITE (2,261) (WHEEL(4,120+L,IPM),L=LSTART,LEND)
781 CONTINUE
      IF (NREM.NE.0) THEN
        LSTART=MOEND-NREM+1
        LEND=MOEND
        WRITE (2,'(1X)')
        WRITE (2,200) (IDH(L),L=LSTART,LEND)
        WRITE (2,201) IDAT(IPM),(WHEEL(1,120+L,IPM),L=LSTART,LEND)
        WRITE (2,202) (WHEEL(2,120+L,IPM),L=LSTART,LEND)
        WRITE (2,203) (WHEEL(5,120+L,IPM),L=LSTART,LEND)
        WRITE (2,260) (WHEEL(3,120+L,IPM),L=LSTART,LEND)
        WRITE (2,261) (WHEEL(4,120+L,IPM),L=LSTART,LEND)
      END IF
      STOP
      END

C
      SUBROUTINE SIMNS(NSIM,TSN,ISEEDO,EM,NEV,NTOP,XR,XI,NSMAX,
&MNO2P1,DT,TITLE)
C *****
C      SIMULATE NORMAL SCORE SEQUENCES FOR H,T,AND D
C
C      NSIM = LENGTH OF SIMULATED SERIES
C      [HN(J);J=1,NSIM] = WAVE HEIGHT NORMAL SCORES
C      [TN(J);J=1,NSIM] = WAVE PERIOD NORMAL SCORES
C      [D(J);J=1,NSIM] = WAVE DIRECTION NORMAL SCORES
C
C      ISEEDO = INITIAL OR STARTING SEED NUMBER, SHOULD BE

```

HPDSIM.FOR

```

C          A POSITIVE INTEGER LESS THAN 2,147,395,600
C
C      NTOP = INTEGER READ FROM OUTPUT FILE HPD.OUT
C      [EM(I,J,L); I=1,6; J=1,6; L=1,NTOP] = SIMULATION MULTIPLIERS
C      READ FROM OUTPUT IN HPD.OUT
C
C      WRITTEN BY LEON BORGMAN
C *****
C      DIMENSION Z(8),U(8),EM(8,8,MNO2P1),NRAN2(12),NRNORM(12)
C      DIMENSION XR(NSMAX),XI(NSMAX),NEV(MNO2P1),TSNN(5)
C      CHARACTER TITLE*50
C      COMPLEX TSN(4,NSMAX)
C
C ***** SET CONSTANTS ***
C      IFLAG=0
C      NO2P1=NSIM/2+1
C      DF=1.0/(NSIM*DT)
C      IDOC=0
C
C ***** LOOP OVER SERIES CREATING FOURIER COEFFICIENTS ***
C      IF (IDOC.NE.0) THEN
C          DO 41 L=1,12
C              NRAN2(J)=0
C              NRNORM(J)=0
41      CONTINUE
C      END IF
C      DO 1 L=1,NSIM
C          IF (L.EQ.1) THEN
C              DO 20 K=1,4
C                  TSN(K,L)=0.0
20      CONTINUE
C          ELSE IF (L.GT.1.AND.L.LE.NTOP) THEN
C              DO 2 J=1,NEV(L)
C                  CALL RAN2(ISEED0,ISEED1,IFLAG,UU)
C                  IFLAG=1
C                  CALL RNORM(UU,Z(J))
C
C                  IF (IDOC.NE.0) THEN
C                      JCAT=1.0+UU*10.0
C                      IF (JCAT.LT.1) JCAT=11
C                      IF (JCAT.GT.10) JCAT=12
C                      NRAN2(JCAT)=NRAN2(JCAT)+1
C                      JCAT=6.0+Z(J)/0.5
C                      IF (JCAT.LT.1) JCAT=11
C                      IF (JCAT.GT.10) JCAT=12
C                      NRNORM(JCAT)=NRNORM(JCAT)+1
C                  END IF
C
C              2      CONTINUE
C              DO 3 I=1,8
C                  SUM=0.0

```

HPDSIM.FOR

```

      DO 4 J=1,NEV(L)
        SUM=SUM+EM(I,J,L)*Z(J)
4      CONTINUE
      U(I)=SUM
3      CONTINUE
      TSN(1,L)=CMPLX(U(1),U(2))
      TSN(2,L)=CMPLX(U(3),U(4))
      TSN(3,L)=CMPLX(U(5),U(6))
      TSN(4,L)=CMPLX(U(7),U(8))
      ELSE IF (L.GT.NTOP.AND.L.LE.NO2P1) THEN
        DO 21 K=1,4
          TSN(K,L)=0.0
21      CONTINUE
        ELSE IF (L.GT.NO2P1) THEN
          DO 22 K=1,4
            TSN(K,L)=CONJG(TSN(K,NSIM-L+2))
22      CONTINUE
        END IF
      1 CONTINUE

      IF (IDOC.NE.0) THEN
        WRITE (4,105)
105     FORMAT (/1X,'HISTOGRAM FOR UNIFORM R.N.:')
        WRITE (4,106) (NRAN2(JCAT),JCAT=1,12)
106     FORMAT (1X,10I7)
        WRITE (4,107)
107     FORMAT (/1X,'HISTOGRAM FOR NORMAL R.N.:')
        WRITE (4,108) (NRNORM(JCAT),JCAT=1,12)
108     FORMAT (1X,10I7)
      END IF

C      ***** INVERSE FFT THE HN,PN,DN NORMAL SCORE TIME SERIES ***
      DO 23 K=1,4
        DO 54 L=1,NSIM
          XR(L)=REAL(TSN(K,L))*DF
          XI(L)=AIMAG(TSN(K,L))*DF
54      CONTINUE
        CALL FT235(NSIM,1.0,0,XR,XI)
        DO 5 L=1,NSIM
          TSN(K,L)=XR(L)
5      CONTINUE
      23 CONTINUE

      IF (IDOC.NE.0) THEN
        WRITE (4,102) TITLE
102     FORMAT (1X,'TITLE: ',A50)
        DO 40 K=1,4
          WRITE (4,100) K
100     FORMAT (/1X,'NORMAL SCORES FOR TIME SERIES: ',I5)
          NSO5=NSIM/5
          NSREM=NSIM-NSO5*5

```

HPDSIM.FOR

```

DO 112 L=1,NS05
  LST=(L-1)*5+1
  LEN=LST+4
DO 114 LSN=LST,LEN
  TSNN(LSN-LST+1)=REAL(TSN(K,LSN))
114 CONTINUE
  WRITE (4,101) LST,(TSNN(LSN),LSN=1,5)
101 FORMAT (1X,I5,5G12.4)
112 CONTINUE
  IF (NSREM.NE.0) THEN
    LST=5*NS05+1
    LEN=LST+NSREM-1
    WRITE (4,101) LST,(REAL(TSN(K,L)),L=LST,LEN)
  END IF
DO 110 JCAT=1,12
  NRNORM(JCAT)=0
110 CONTINUE
DO 109 L=1,NSIM
  JCAT=6.0+TSN(K,L)/0.5
  IF (JCAT.LT.1) JCAT=11
  IF (JCAT.GT.10) JCAT=12
  NRNORM(JCAT)=NRNORM(JCAT)+1
109 CONTINUE
  WRITE (4,111) K
111 FORMAT (/1X,'HISTOGRAM FOR NORMAL SCORES OF ',I2)
  WRITE (4,108) (NRNORM(JCAT),JCAT=1,12)
40 CONTINUE
END IF

RETURN
END

```

```

C
C SUBROUTINE RAN2(ISEED0,ISEED1,IFLAG,U)
C *****
C GENERATE UNIFORM PSEUDO-RANDOM NUMBERS ON (0,1) IN A
C MACHINE-INDEPENDENT WAY FOR ANY COMPUTER WHICH CAN
C OPERATE WITH INTEGERS FROM 0 TO 2**31.
C
C STARTING SEED NUMBER SHOULD BE GREATER THAN ZERO AND
C LESS THAN 2147395600. ON FIRST CALL TO SUBR.,
C IFLAG SHOULD BE SET TO ZERO, AND ISEED0 SHOULD
C CONTAIN THE STARTING SEED NUMBER. ON ALL LATER
C CALLS TO THE SUBR., IFLAG SHOULD BE SET TO ZERO,
C AND NO SEED NUMBER IS NEEDED SINCE THE SUBROUTINE
C USES THE LAST RESULT AS A SEED FOR THE NEXT.
C
C U = UNIFORM PSEUDO-RANDOM NUMBER ON RETURN
C
C THE PROGRAM ACHIEVES THE MACHINE INDEPENDENCE BY

```

HPDSIM.FOR

```

C      OPERATING WITH NUMBERS TO THE BASE NB, WHERE
C      NB IS SELECTED SO THAT 2*NB*NB+NB IS LESS THAN
C      THE LARGEST POSITIVE INTEGER THAT THE COMPUTER
C      CAN HANDLE.
C
C      THE RANDOM NUMBER GENERATION USES MODULUS ARITHMETIC.
C
C      NEXT = I * LAST + K (MODULUS NB*NB)
C      U = NEXT/(NB*NB), IN DECIMAL DIVISION
C
C      IN THE CODE BELOW,
C
C      I = 7777771 = 167 * NB + 38991
C      K = 1111113 = 23 * NB + 45293
C
C      THESE WERE ARBITRARILY SELECTED TO BE PRIME TO MOST
C      SMALL INTEGERS.
C
C      THE COMPUTATIONS PROCEED BY EXPRESSING I, K, AND LAST
C      IN BASE NB AS:
C
C      I = IO + I1*NB
C      K = KO + K1*NB
C      LAST = ISEEDO + ISEED1*NB
C
C      THEN THE ARITHMETIC IS DONE FROM
C
C      NEXT = (IO+I1*NB)*(ISEEDO+ISEED1*NB)+(KO+K1*NB)
C            = (IO*ISEEDO+KO)+(IO*ISEED1+I1*ISEEDO+K1)*NB
C            + (I1*ISEED1)*NB*NB
C
C      THE LAST TERM NEED NOT BE COMPUTED, SINCE IT IS
C      ZERO (TO MODULUS NB*NB). THE OTHER TWO TERMS ARE
C      THEMSELVES EXPANDED IN BASE NB AS:
C
C      LO = IO*ISEEDO+KO = L00 + L01*NB
C      L1 = IO*ISEED1+I1*ISEEDO+K1 = L10 + L11*NB
C
C      THEN NEXT MAY BE WRITTEN AS:
C
C      NEXT = LO + L1*NB
C            = (L00+L01*NB)+(L10+L11*NB)*NB
C            = (L00)+(L01+L10)*NB+(L11)*NB*NB
C
C      AS BEFORE, THE LAST TERM IS ZERO (TO MODULUS NB*NB) AND
C      NEED NOT BE COMPUTED. THE TERM L01+L10 IS REDUCED,
C      TO A BASE NB NUMBER AS:
C
C      LLL = L01 + L10 = LLLO + LLL1*NB
C
C      THEN

```

HPDSIM.FOR

```

C
C      NEXT = L00 + (LLLO+LLL1*NB)*NB
C          = L00 + LLLO*NB + LLL1*NB*NB
C
C      AGAIN, THE LAST TERM IS ZERO (MODULUS NB*NB), SO THE
C      FINAL REDUCTION FOR NEXT IS:
C
C      NEXT = L00 + LLLO*NB
C      U = NEXT/(NB*NB), DECIMAL DIVISION
C
C      ONLY THE LAST DIVISION IS DONE IN DECIMAL OPERATIONS.
C      THE OTHERS ARE ALL INTEGER COMPUTATIONS. THE LARGEST
C      INTEGER ENCOUNTERED IN THE COMPUTATIONS WOULD BE
C      BOUNDED ABOVE BY
C
C       $10*ISEED1 + I1*ISEED0 + K1 < NB*NB + NB*NB + NB = 2*NB*NB + NB$ 
C
C      HENCE, IF NB IS SELECTED SO THAT  $2*NB**2 + NB$  IS SMALLER
C      THAN THE MAXIMUM ALLOWED POSITIVE INTEGER FOR THE
C      COMPUTER, THE CALCULATIONS WILL NOT OVERFLOW.
C
C      IN THE CODE BELOW:
C
C       $2*NB**2 + NB - 2**31 = 0$ 
C
C      GIVES NB = 46340 AND ALLOWS A MODULUS BASE OF
C       $NB*NB = 2147395600$  WITHOUT OVERFLOW. FOR OTHER COMPUTERS,
C      OTHER NUMBERS MAY BE SELECTED. THE ONES GIVEN HERE WORK
C      FOR THE COMMON LIMITATION THAT INTEGERS LIE BETWEEN
C       $(-2**31 + 1)$  AND  $(+2**31)$ .
C
C      WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING
C      *****
C
C      I0=38991
C      I1=167
C      K0=45293
C      K1=23
C      NB=46340
C      XNB2=2147395600.
C      IDOC=0
C      IF (IFLAG.EQ.0) THEN
C          ISEED1=ISEED0/NB
C          ISEED0=ISEED0-ISEED1*NB
C      END IF
C      IF (IDOC.NE.0) THEN
C          WRITE (4,100) ISEED0, ISEED1
100  FORMAT (/1X, ' ISEED0, ISEED1= ', 2I15)
C      END IF
C      L0=I0*ISEED0+K0
C      L1=I0*ISEED1+I1*ISEED0+K1

```

HPDSIM. FOR

```

      L01=L0/NB
      L00=L0-L01*NB
      L11=L1/NB
      L10=L1-L11*NB
      LLL=L01+L10
      ISEED1=LLL-(LLL/NB)*NB
      ISEED0=L00
      L=ISEED0+ISEED1*NB
      U=L/2147395600.0

      IF (IDOC.NE.0) THEN
        WRITE (4,101) NB,XNB2
101      FORMAT (1X,'NB,XNB2= ',I12,F15.0)
        WRITE (4,102) IO,I1,K0,K1
102      FORMAT (1X,'IO,I1,K0,K1= ',4I14)
        WRITE (4,103) L0,L1
103      FORMAT (1X,'L0,L1= ',2I14)
        WRITE (4,104) L00,L01,L10,L11
104      FORMAT (1X,'L00,L01,L10,L11= ',4I14)
        WRITE (4,105) ISEED0,ISEED1
105      FORMAT (1X,'ISEED0,ISEED1= ',2I14)
        WRITE (4,106) L,U
106      FORMAT (1X,'L,U= ',I14,F14.6)
      END IF
      RETURN
      END

C
      SUBROUTINE RNORM(U,Z)
C *****
C      SUBROUTINE OUTPUTS Z, GIVEN F(Z)=U FOR THE STANDARD
C      NORMAL
C *****
      P=U
      IF (U.GT.0.5) P=1.0-U
      T=SQRT(ALOG(1.0/(P*P)))
      FN=(T*0.010328+0.802853)*T+2.515517
      FD=((T*0.001308+0.189269)*T+1.432788)*T+1.0
      Z=T-FN/FD
      IF (U.LT.0.5) Z=-Z
      RETURN
      END

C
      SUBROUTINE FT235(N,SGN,LIST,XR,XI)
      DIMENSION XR(1),XI(1)
C *****
C      (XR(J),J=1,N)=REAL PART OF SEQUENCE TO BE FOURIER TRANSFORMED.
C      (XI(J),J=1,N)=IMAG. PART OF SEQUENCE TO BE FOURIER TRANSFORMED.
C      THE DISCRETE FOURIER TRANSFORM IS OF THE FORM:
C      A(M)=SUM FOR J=1 TO N OF
C              (XR+I*XI)*EXP(I*2*PI*SGN*(M-1)*(J-1)/N),
C      FOR 1.LE.M.LE.N

```

HPDSIM.FOR

```

C      THUS, SGN DEFINES THE SIGN OF THE EXPONENT, AND IS EITHER 1.0
C      OR -1.0.
C      A TABLE OF TRIGONOMETRIC AND BIT REVERSAL FUNCTIONS ARE
C      USED IN THE FFT. IF LIST=0, THE TABLE IS COMPUTED. IF
C      LIST=1, THE TABLE IS NOT COMPUTED AND IS PRESUMED TO BE
C      PRESENT FROM EARLIER CALLS TO THE SUBROUTINE.
C      CONSEQUENTLY, LIST=0 ON FIRST CALL TO SUBR.
C      LIST=1 FOR SUBSEQUENT CALLS AT SAME N VALUE.
C
C      PARAMETER VALUES:
C      MAXN=LARGEST VALUE TO BE USED FOR N (LENGTH OF SEQUENCE)
C      MAXN2=LARGEST POWER OF 2 LESS THAN OR EQUAL TO MAXN
C      MAXN3=LARGEST POWER OF 3 LESS THAN OR EQUAL TO MAXN
C      MAXN5=LARGEST POWER OF 5 LESS THAN OR EQUAL TO MAXN
C      MAXNO2=MAXN2/2
C      MAXNO3=MAXN3/3
C      MAXNO5=MAXN5/5
C
C      COMMON BLOCK *****
C      PARAMETER (MAXN=1200,MAXN2=1024,MAXN3=729,MAXN5=625)
C      PARAMETER (MAXNO2=512,MAXNO3=243,MAXNO5=125)
C      *****
C      DIMENSION W(MAXN),WR(MAXN2),WI(MAXN2),WRD(MAXN3),WID(MAXN3)
C      DIMENSION WRT(MAXN5),WIT(MAXN5)
C      DIMENSION IRB(MAXNO2),CO(MAXNO2),SI(MAXNO2)
C      DIMENSION IRBD(MAXNO3),COD(MAXNO3),SID(MAXNO3)
C      DIMENSION IRBT(MAXNO5),COT(MAXNO5),SIT(MAXNO5)
C      DIMENSION CO1(MAXNO5),SI1(MAXNO5),CO2(MAXNO5),SI2(MAXNO5)
C      DIMENSION CO3(MAXNO5),SI3(MAXNO5)
C      COMMON /FFT/W,WR,WI,WRD,WID,WRT,WIT,IRB,CO,SI,IRBD,COD,SID
C      COMMON /FFT/IRBT,COT,SIT,CO1,SI1,CO2,SI2,CO3,SI3
C      *****
C      PARAMETER (MAXN=5000,MAXN2=4096,MAXN3=2187,MAXN5=3125)
C      PARAMETER (MAXNO2=2048,MAXNO3=729,MAXNO5=625)
C      NOTE: THE ABOVE COMMON BLOCK PARAMETERS WOULD ALLOW THE USE OF
C      ANY N.LE.5000 OF THE FORM N=(2**K)*(3**L)*(5**M) FOR
C      INTEGER K, L, AND M WITHOUT CHANGING DIMENSIONS.
C      THE CHOICES OF N WHICH MATCH THESE REQUIREMENTS ARE:
C
C      2      3      4      5      6      8      9      10
C      12     15     16     18     20     24     25     27
C      30     32     36     40     45     48     50     54
C      60     64     72     75     80     81     90     96
C      100    108    120    125    128    135    144    150
C      160    162    180    192    200    216    225    240
C      243    250    256    270    288    300    320    324
C      360    375    384    400    405    432    450    480
C      486    500    512    540    576    600    625    640
C      648    675    720    729    750    768    800    810
C      864    900    960    972    1000   1024   1080   1125
C      1152   1200   1215   1250   1280   1296   1350   1440
C      1458   1500   1536   1600   1620   1728   1800   1875

```

HPDSIM. FOR

```

C      1920   1944   2000   2025   2048   2160   2187   2250
C      2304   2400   2430   2500   2560   2592   2700   2880
C      2916   3000   3072   3125   3200   3240   3375   3456
C      3600   3645   3750   3840   3888   4000   4050   4096
C      4320   4374   4500   4608   4800   4860   5000
C      HOWEVER, FOR A FIXED N, LESS COMPUTER STORAGE CAN BE
C      REQUIRED IF THE FOLLOWING "EXACT" DIMENSIONING IS
C      INTRODUCED:
C          N2=2**K
C          N3=3**L
C          N5=5**M
C          N02=N2/2
C          N03=N3/3
C          N05=N5/5
C      DIMENSION W(N),WR(N2),WI(N2),WRD(N3),WID(N3),WRT(N5),
C          WIT(N5),IRB(N02),CO(N02),SI(N02),IRBD(N03),COD(N03),
C          SID(N03),IRBT(N05),COT(N05),SIT(N05),CO1(N05),
C          SI1(N05),CO2(N05),SI2(N05),CO3(N05),SI3(N05)
C      THE MAXIMAL DIMENSIONING WAS TAKEN FROM THE LARGEST
C      POWER OF 2, OF 3, AND OF 5 WHICH IS LESS THAN 5000.
C
C      PROGRAMS WRITTEN BY A. YFANTIS, UNIV. NEVADA, LAS VEGAS,
C      AND LEON BORGMAN, UNIV. WYOMING, LARAMIE, WYO.
C
C *****
C      NN=N
C      K=0
C      1 ND=NN-(NN/2)*2
C          IF (ND.NE.0) GO TO 2
C          K=K+1
C          NN=NN/2
C          IF (NN.EQ.1) GO TO 2
C          GO TO 1
C      2 L=0
C      3 ND=NN-(NN/3)*3
C          IF (ND.NE.0) GO TO 4
C          L=L+1
C          NN=NN/3
C          IF (NN.EQ.1) GO TO 4
C          GO TO 3
C      4 M=0
C      5 ND=NN-(NN/5)*5
C          IF (ND.NE.0) GO TO 6
C          M=M+1
C          NN=NN/5
C          IF (NN.EQ.1) GO TO 6
C          GO TO 5
C      6 CONTINUE
C          IF (NN.NE.1) THEN
C              WRITE (2,100)
C      100  FORMAT ('WARNING: THE ASSUMPTIONS OF THE SUBR. ARE VIOLATED'//

```

HPDSIM.FOR

```

&'      N IS NOT EQUAL TO (2**K)*(3**L)*(5**M)')
      RETURN
      ELSE IF (K.NE.0.AND.L.EQ.0.AND.M.EQ.0) THEN
        NO2=N/2
        CALL FFT2(K,N,NO2,SGN,LIST,XR,XI,IRB,CO,SI)
      ELSE IF (K.EQ.0.AND.L.NE.0.AND.M.EQ.0) THEN
        NO3=N/3
        CALL FFT3(L,N,NO3,SGN,LIST,XR,XI,IRBD,COD,SID)
      ELSE IF (K.EQ.0.AND.L.EQ.0.AND.M.NE.0) THEN
        NO5=N/5
        CALL FFT5(M,N,NO5,SGN,LIST,XR,XI,IRBT,COT,SIT,CO1,SI1,
& CO2,SI2,CO3,SI3)
      ELSE IF (K.NE.0.AND.L.NE.0.AND.M.EQ.0) THEN
        N2=2**K
        N3=3**L
        NO2=N2/2
        NO3=N3/3
        CALL FFT23(K,L,N,N2,N3,NO2,NO3,SGN,LIST,XR,XI,IRB,IRBD,
& CO,SI,COD,SID,WR,WI,WRD,WID,W)
      ELSE IF (K.NE.0.AND.L.EQ.0.AND.M.NE.0) THEN
        N2=2**K
        N5=5**M
        NO2=N2/2
        NO5=N5/5
        CALL FFT25(K,M,N,N2,N5,NO2,NO5,SGN,LIST,XR,XI,IRB,IRBT,
& CO,SI,COT,SIT,WR,WI,WRT,WIT,W,CO1,SI1,CO2,SI2,CO3,SI3)
      ELSE IF (K.EQ.0.AND.L.NE.0.AND.M.NE.0) THEN
        N3=3**L
        N5=5**M
        NO3=N3/3
        NO5=N5/5
        CALL FFT35(L,M,N,N3,N5,NO3,NO5,SGN,LIST,XR,XI,IRBD,IRBT,
& COD,SID,COT,SIT,WRD,WID,WRT,WIT,W,CO1,SI1,CO2,SI2,CO3,SI3)
      ELSE
        N2=2**K
        N3=3**L
        N5=5**M
        NO2=N2/2
        NO3=N3/3
        NO5=N5/5
        CALL FFT235(K,L,M,N,N2,N3,N5,NO2,NO3,NO5,SGN,LIST,XR,XI,
& IRB,IRBD,IRBT,CO,SI,COD,SID,COT,SIT,WR,WI,WRD,WID,WRT,WIT,
& CO1,SI1,CO2,SI2,CO3,SI3,W)
      END IF
      RETURN
      END

C      SUBROUTINE FFT2(K,N,NO2,SGN,LIST,XR,XI,IRB,CO,SI)
C      *****
C      N=LENGTH OF DATA SERIES BEING TRANSFORMED.
C      K=LOG2(N).

```

HPDSIM.FOR

```

C      NO2=N/2
C      SGN= +1 OR -1 ACCORDING AS THE EXPONENT IN THE FFT TRANSFORMATION
C      IS POSITIVE OR NEGATIVE.
C      IF LIST=0, THE REVERSED-BIT LIST AND THE ASSOCIATED SINES AND
C      COSINES ARE COMPUTED. IF LIST=1, THE LIST IS NOT COMPUTED.
C      INSTEAD, THE LIST COMPUTED ON THE PREVIOUS CALL IS USED.
C      XR=REAL PART OF THE INPUT DATA SERIES. INITIALLY AND EQUALS REAL
C      PART OF OUTPUT ON RETURN FROM THE SUBR.
C      XI=IMAG. PART OF DATA SERIES ON INPUT AND IS IMAG. PART OF OUTPUT
C      ON RETURN FROM THE SUBR.
C      IRB=NO2-DIMENSIONAL VECTOR OF REVERSED BITS ON OUTPUT.
C      CO=NO2-DIMENSIONAL VECTOR OF COSINES ON OUTPUT.
C      SI=NO2-DIMENSIONAL VECTOR OF SINES ON OUTPUT.
C      *****
C      DIMENSION IRB(1),CO(1),SI(1),XR(1),XI(1)
C      IF (LIST.EQ.1) GO TO 3
C      THE REVERSED-BIT LIST AND TRIG. FUNCTIONS ARE GENERATED BY THE
C      PROCEDURE GIVEN IN TABLE 7-VI.
C      IRB(1)=0
C      DO 1 J=1,K
C      ID=2**(J-1)
C      DO 1 I=1,ID
C      IRB(I)=IRB(I)*2
C      IF (J.LT.K) IRB(I+ID)=IRB(I)+1
1  CONTINUE
C      FN=N
C      W=6.2831853/FN
C      DO 2 I=1,ID
C      FIR=IRB(I)/2
C      A=FIR*W
C      CO(I)=COS(A)
C      SI(I)=SIN(A)
2  CONTINUE
3  CONTINUE
C      FOR EACH COLUMN OF THE FFT (NC=COLUMN NUMBER), CALCULATE
C      NB=NUMBER OF BLOCKS IN THE COLUMN, LB=LENGTH OF BLOCKS, LB2=HALF
C      OF BLOCK LENGTH.
C      DO 4 NC=1,K
C      NB=2**(NC-1)
C      LB=N/NB
C      LB2=LB/2
C      THE BLOCKS OF THE COLUMN ARE LOOPED OVER. IS=SEQUENCE NUMBER
C      AT THE START OF THE BLOCK, IFF=SEQUENCE NUMBER HALF WAY THROUGH
C      BLOCK.
C      DO 4 IB=1,NB
C      C=CO(IB)
C      S=SGN*SI(IB)
C      IS=(IB-1)*LB+1
C      IFF=(IB-1)*LB+LB2
C      THE VALUES IN THE NEXT COLUMN ARE COMPUTED USING THE TRIG.
C      FUNCTIONS FROM THE PRE-GENERATED LIST.

```

HPDSIM.FOR

```

      DO 4 I=IS,IFF
      I2=I+LB2
      QR=XR(I2)*C-XI(I2)*S
      QI=XR(I2)*S+XI(I2)*C
      XR(I2)=XR(I)-QR
      XI(I2)=XI(I)-QI
      XR(I)=XR(I)+QR
      XI(I)=XI(I)+QI
4 CONTINUE
C   THE FFT COEFFICIENTS ARE UNSCRANBLED INTO NORMAL ORDER.
      DO 6 I=1,NO2
      DO 6 L=1,2
      IR=IRB(I)+L
      II=I+(L-1)*NO2
      IF (IR.LE.II) GO TO 6
      ZR=XR(IR)
      ZI=XI(IR)
      XR(IR)=XR(II)
      XI(IR)=XI(II)
      XR(II)=ZR
      XI(II)=ZI
6 CONTINUE
      RETURN
      END

C   SUBROUTINE FFT3(K,N,NO3,SGN,LIST,XR,XI,IRB,CO,SI)
      DIMENSION IRB(1),CO(1),SI(1),XR(1),XI(1)
C *****
C   THE INPUT TO THE SUBROUTINE CONSISTS OF THE VARIABLES
C   N, K, NO3, SGN, LIST, XR, AND XI. N IS THE NUMBER OF
C   MEASUREMENTS WHICH IS OF THE FORM N=3**K, NO3 IS EQUAL
C   TO N/3. SGN IS -1 IN THE CASE OF THE FOURIER TRANSFORM,
C   AND +1 IN THE CASE OF THE INVERSE FOURIER TRANSFORM.
C   XR, XI ARE THE REAL AND IMAGINARY PARTS RESPECTIVELY,
C   OF THE INPUT COMPLEX SEQUENCE OF NUMBERS. AFTER THE
C   FFT3 IS PERFORMED, XR, XI, ARE THE REAL AND IMAGINARY
C   PARTS OF THE TRANSFORMED DATA. IF LIST=0, THE REVERSE-
C   DIGIT LIST AND THE ASSOCIATED SINES AND COSINES ARE
C   COMPUTED. IF LIST=1 THE LIST IS NOT COMPUTED. INSTEAD
C   THE LIST COMPUTED OF THE PREVIOUS CALL IS USED. IRB,
C   CO, AND SI, ARE WORK VECTORS.
C *****
      T=6.2831853/3.0
      T1=2.*T
      FN=N
      IF(LIST .EQ. 1) GO TO 3
      IRB(1)=0
      DO 1 J=1,K
      ID=3**(J-1)
      DO 1 I=1,ID
      IRB(I)=IRB(I)*3

```

HPDSIM.FOR

```

      IF(J.EQ. K) GO TO 1
      IRB(I+ID)=IRB(I)+1
      IRB(I+2*ID)=IRB(I)+2
1     CONTINUE
      DO 2 I=1, ID
      FIR=IRB(I)/FN
      A=FIR*T
      CO(I)=COS(A)
      SI(I)=SIN(A)
2     CONTINUE
C   FOR EACH COLUMN OF THE FFT (NC=COLUMN NUMBER), NB=NUMBER
C   OF BLOCKS IN THE COLUMN, LB=LENGTH OF BLOCKS, LB3= A THIRD
C   OF BLOCK LENGTH.
3     C1=COS(T)
      S1=SGN*SIN(T)
      C2=COS(T1)
      S2=SGN*SIN(T1)
      DO 4 NC=1, K
      NB=3*(NC-1)
      LB=N/NB
      LB3=LB/3
C   THE BLOCKS OF THE COLUMN ARE LOOPED OVER. IS=SEQUENCE NUMBER
C   AT THE START OF THE BLOCK, IFF=SEQUENCE NUMBER A THIRD WAY
C   THROUGH BLOCK
      DO 4 IB=1, NB
      C=CO(IB)
      S=SGN*SI(IB)
      IS=(IB-1)*LB+1
      IFF=(IB-1)*LB+LB3
C   THE VALUES IN THE NEXT COLUMN ARE COMPUTED USING THE TRIG.
C   FUNCTIONS FROM THE PRE-GENERATED LIST.
      DO 4 I=IS, IFF
      I2=I+LB3
      I3=I+2*LB3
      QR1=XR(I2)*C-XI(I2)*S
      QI1=XR(I2)*S+XI(I2)*C
      QR2=XR(I3)*(C*C-S*S)-2.*XI(I3)*S*C
      QI2=(C*C-S*S)*XI(I3)+2.*S*C*XR(I3)
      XR(I2)=XR(I)+C1*QR1-S1*QI1+C2*QR2-S2*QI2
      XI(I2)=XI(I)+S1*QR1+C1*QI1+S2*QR2+C2*QI2
      XR(I3)=XR(I)+C2*QR1-S2*QI1+C1*QR2-S1*QI2
      XI(I3)=XI(I)+S2*QR1+C2*QI1+S1*QR2+C1*QI2
      XR(I)=XR(I)+QR1+QR2
      XI(I)=XI(I)+QI1+QI2
4     CONTINUE
C   THE FFT COEFFICIENTS ARE INSCRAMBLED INTO NORMAL ORDER
      DO 6 I=1, NO3
      DO 6 L=1, 3
      IR=IRB(I)+L
      II=I+(L-1)*NO3
      IF (IR.LE.II) GO TO 6

```

HPDSIM.FOR

```

      ZR=XR(IR)
      ZI=XI(IR)
      XR(IR)=XR(II)
      XI(IR)=XI(II)
      XR(II)=ZR
      XI(II)=ZI
6     CONTINUE
      RETURN
      END

C
      SUBROUTINE FFT5(K,N,NOS,SGN,LIST,XR,XI,IRB,CO,SI,CO1,SI1,
+      CO2,SI2,CO3,SI3)
      DIMENSION IRB(1),CO(1),SI(1),XR(1),XI(1),CO1(1),
+      SI1(1),SI2(1),CO2(1),CO3(1),SI3(1)
C *****
C     THE INPUT TO THE SUBROUTINE CONSISTS OF THE VARIABLES
C     N, K, NOS, SGN, LIST, XR, XI. N IS THE NUMBER OF MEAS-
C     UREMENTS WHICH IS OF THE FORM N=5**K, NOS IS EQUAL TO
C     N/5. SGN IS -1 IN THE CASE OF THE FOURIER TRANSFORM,
C     AND +1 IN THE CASE OF THE INVERSE FOURIER TRANSFORM.
C     XR AND XI ARE THE REAL AND IMAGINARY PARTS RESPECTIVELY
C     OF THE INPUT COMPLEX SEQUENCE OF NUMBERS. AFTER THE FFT5
C     IS PERFORMED XR, XI ARE THE REAL AND IMAGINARY PARTS OF
C     THE TRANSFORMED DATA. IF LIST=0, THE REVERSE-DIGIT LIST
C     AND THE ASSOCIATED SINES AND COSINES ARE COMPUTED. IF
C     LIST=1, THE LIST IS NOT COMPUTED. INSTEAD THE LIST COM-
C     PUTED ON THE PREVIOUS CALL IS USED. IRB, CO, SI, CO1,
C     SI1, CO2, SI2, CO3, AND SI3, ARE WORK VECTORS.
C *****
      T=6.2831853/5.
      T1=2.*T
      T2=3.*T
      T3=4.*T
      FN=N
      IF(LIST .EQ. 1) GO TO 3
      IRB(1)=0
      DO 1 J=1,K
      ID=5**(J-1)
      DO 1 I=1,ID
      IRB(I)=IRB(I)*5
      IF(J .EQ. K) GO TO 1
      IRB(I+ID)=IRB(I)+1
      IRB(I+2*ID)=IRB(I)+2
      IRB(I+3*ID)=IRB(I)+3
      IRB(I+4*ID)=IRB(I)+4
1     CONTINUE
      DO 2 I=1,ID
      FIR=IRB(I)/FN
      A=FIR*T
      CO(I)=COS(A)
      SI(I)=SIN(A)

```

HPDSIM. FOR

```

      A1=2.*A
      C01(I)=COS(A1)
      S11(I)=SIN(A1)
      A2=3.*A
      C02(I)=COS(A2)
      S12(I)=SIN(A2)
      A3=4.*A
      C03(I)=COS(A3)
      S13(I)=SIN(A3)
2    CONTINUE
C    FOR EACH COLUMN OF THE FFT (NC=COLUMN NUMBER), CALCULATE
C    NB=NUMBER OF BLOCKS IN THE COLUMN, LB=LENGTH OF BLOCKS,
C    LB5= A FIFTH OF BLOCK LENGTH
3    S1=SGN*SIN(T)
      C1=COS(T)
      S2=SGN*SIN(T1)
      C2=COS(T1)
      S3=SGN*SIN(T2)
      C3=COS(T2)
      S4=SGN*SIN(T3)
      C4=COS(T3)
      DO 4 NC=1,K
        NB=5**(NC-1)
        LB=N/NB
        LB5=LB/5
C      THE BLOCKS OF THE COLUMN ARE LOOPED OVER, IS=SEQUENCE NUMBER
C      AT THE START OF THE BLOCK, IFF=SEQUENCE NUMBER A FIFTH WAY
C      THROUGH BLOCK.
        DO 4 IB=1,NB
          IS=(IB-1)*LB+1
          IFF=(IB-1)*LB+LB5
C      THE VALUES OF THE COLUMN ARE COMPUTED USING THE TRIG. FUNC-
C      TIONS FROM THE PRE-GENERATED LIST.
          DO 4 I=IS, IFF
            SAA=SGN*SI(IB)
            SA1=SGN*S11(IB)
            SA2=SGN*S12(IB)
            SA3=SGN*S13(IB)
            I2=I+LB5
C      THE FFT COEFFICIENTS ARE UNSCRAMBLED INTO NORMAL ORDER
            I3=I+2*LB5
            I4=I+3*LB5
            I5=I+4*LB5
            QR1=XR(I2)*C01(IB)-XI(I2)*SAA
            QI1=XR(I2)*SAA+XI(I2)*C01(IB)
            QR2=XR(I3)*C01(IB)-XI(I3)*SA1
            QI2=XR(I3)*SA1+XI(I3)*C01(IB)
            QR3=XR(I4)*C02(IB)-XI(I4)*SA2
            QI3=XR(I4)*SA2+XI(I4)*C02(IB)
            QR4=XR(I5)*C03(IB)-XI(I5)*SA3
            QI4=XR(I5)*SA3+XI(I5)*C03(IB)

```

HPDSIM.FOR

```

      XR(I2)=XR(I)+QR1*C1-QI1*S1+QR2*C2-QI2*S2+QR3*C3-QI3*S3+QR4*C4-QI4*
      +S4
      XI(I2)=XI(I)+QR1*S1+QI1*C1+QR2*S2+QI2*C2+QR3*S3+QI3*C3+QR4*S4+QI4*
      +C4
      XR(I3)=XR(I)+QR1*C2-QI1*S2+QR2*C4-QI2*S4+QR3*C1-QI3*S1+QR4*C3-QI4*
      +S3
      XI(I3)=XI(I)+QR1*S2+QI1*C2+QR2*S4+QI2*C4+QR3*S1+QI3*C1+QR4*S3+QI4*
      +C3
      XR(I4)=XR(I)+QR1*C3-QI1*S3+QR2*C1-QI2*S1+QR3*C4-QI3*S4+QR4*C2-QI4*
      +S2
      XI(I4)=XI(I)+QR1*S3+QI1*C3+QR2*S1+QI2*C1+QR3*S4+QI3*C4+QR4*S2+QI4*
      +C2
      XR(I5)=XR(I)+QR1*C4-QI1*S4+QR2*C3-QI2*S3+QR3*C2-QI3*S2+QR4*C1-QI4*
      +S1
      XI(I5)=XI(I)+QR1*S4+QI1*C4+QR2*S3+QI2*C3+QR3*S2+QI3*C2+QR4*S1+QI4*
      +C1
      XR(I)=XR(I)+QR1+QR2+QR3+QR4
      XI(I)=XI(I)+QI1+QI2+QI3+QI4
4     CONTINUE
C     THE FFT COEFFICIENTS ARE UNSCRAMBLED INTO NORMAL ORDER
      DO 6 I=1,NOS
      DO 6 L=1,5
      IR=IRB(I)+L
      II=I+(L-1)*NOS
      IF (IR.LE.II) GO TO 6
      ZR=XR(IR)
      ZI=XI(IR)
      XR(IR)=XR(II)
      XI(IR)=XI(II)
      XR(II)=ZR
      XI(II)=ZI
6     CONTINUE
      RETURN
      END
C
      SUBROUTINE FFT23(K1,K2,N,N1,N2,N02,N03,SGN,LIST,XR,XI,IRB,IRBD,CO,
      +SI,COD,SID,WR,WI,WRD,WID,W)
      DIMENSION XR(1),XI(1),IRB(1),IRBD(1),CO(1),SI(1),COD(1),
      +SID(1),WR(1),WI(1),WRD(1),WID(1),W(1)
C     *****
C     THE SUBROUTINE FFT23 JOINS THE FFT2 AND THE FFT3. WHEN
C     USED, BOTH, FFT2 AND FFT3, SUBROUTINES MUST BE PRESENT.
C     N IS THE NUMBER OF MEASUREMENTS WHICH IS OF THE FORM
C     N=(2**K1)*(3**K2). N1=2**K1 AND N2=3**K2, N02=N1/2,
C     N03=N2/3. SGN=-1 IF THE FOURIER TRANSFORM IS TO BE PER-
C     FORMED. IF LIST=0, THE REVERSED-DIGIT LIST AND THE AS-
C     SOCIATED SINES AND COSINES ARE COMPUTED. IF LIST=1, THE
C     LIST IS NOT COMPUTED. INSTEAD THE LIST COMPUTED ON THE
C     PREVIOUS CALL IS USED. XR IS THE REAL PART AND XI IS THE
C     IMAGINARY PART OF THE COMPLEX SEQUENCE TO BE TRANSFORMED.
C     AFTER THE FFT23 IS PERFORMED XR AND XI ARE THE REAL AND

```

HPDSIM.FOR

C IMAGINARY PART OF THE TRANSFORMED DATA. IRB, CO, AND SI
 ARE WORK VECTORS.

C *****

TPI=6.283185307

XN=N

DO 30 I=1,N1

FI=I-1

TPNF=TPI*FI/XN

DO 10 I=1,N2

L=J-1

WRD(J)=XR(N1*L+I)

WID(J)=XI(N1*L+I)

10 CONTINUE

CALL FFT3(K2,N2,N03,SGN,LIST,WRD,WID,IRBD,COD,SID)

DO 20 J=1,N2

L=J-1

XJ=J-1

THET=SGN*TPNF*XJ

C=COS(THET)

S=SIN(THET)

XR(N1*L+I)=WRD(J)*C-WID(J)*S

XI(N1*L+I)=WRD(J)*S+WID(J)*C

20 CONTINUE

30 CONTINUE

DO 60 I=1,N2

M=I-1

DO 40 J=1,N1

WR(J)=XR(N1*M+J)

WI(J)=XI(N1*M+J)

40 CONTINUE

CALL FFT2(K1,N1,N02,SGN,LIST,WR,WI,IRB,CO,SI)

DO 50 J=1,N1

XR(N1*M+J)=WR(J)

XI(N1*M+J)=WI(J)

50 CONTINUE

60 CONTINUE

DO 70 I=1,N

W(I)=XR(I)

70 CONTINUE

DO 80 I=1,N2

DO 80 J=1,N1

XR((J-1)*N2+I)=W(N1*(I-1)+J)

80 CONTINUE

DO 90 I=1,N

W(I)=XI(I)

90 CONTINUE

DO 95 I=1,N2

DO 95 J=1,N1

XI((J-1)*N2+I)=W(N1*(I-1)+J)

95 CONTINUE

RETURN

HPDSIM.FOR

END

C

```
SUBROUTINE FFT25(K1,K2,N,N1,N2,N02,N05,SGN,LIST,XR,XI,IRB,IRBD,
+CO,SI,COD,SID,WR,WI,WRD,WID,W,CO1,SI1,CO2,SI2,CO3,SI3)
  DIMENSION XR(1),XI(1),IRB(1),IRBD(1)
  DIMENSION CO(1),SI(1),COD(1),SID(1),WR(1),WI(1)
  DIMENSION WRD(1),WID(1),W(1),CO1(1),SI1(1)
  DIMENSION CO2(1),CO3(1),SI2(1),SI3(1)
```

C *****

```
C THE SUBROUTINE FFT25 JOINS THE FFT2 AND THE FFT5. HENCE
C WHEN USED THE SUBROUTINES FFT2 AND FFT5 MUST BE PRESENT.
C N IS THE NUMBER OF MEASUREMENTS, WHICH IS OF THE FORM
C  $N=(2**K1)*(5**K2)$ .  $N1=2**K1$ ,  $N2=5**K2$ ,  $N02=N1/2$ ,  $N05=N2/5$ ,
C  $SGN=-1$ , IF THE FOURIER TRANSFORM IS TO BE PERFORMED. IF
C  $LIST=0$  THE REVERSED-DIGIT LIST AND THE ASSOCIATED SINES AND
C COSINES ARE COMPUTED. IF  $LIST=1$ , THE LIST IS NOT COMPUTED.
C INSTEAD THE LIST COMPUTED ON THE PREVIOUS CALL IS USED.
C XR, XI ARE THE REAL AND THE IMAGINARY PARTS RESPECTIVELY
C OF THE INPUT COMPLEX SEQUENCE OF NUMBERS. AFTER THE FFT25
C IS PERFORMED XR AND XI ARE THE REAL AND IMAGINARY PARTS
C OF THE TRANSFORMED DATA. IRB, IRBD, CO, COD, SID, WR, WI,
C WRD, WID, W, CO1, SI1, CO2, SI2, CO3, AND SI3 ARE WORK
C VECTORS.
```

C *****

```
  TPI=6.283185307
  XN=N
  DO 30 I=1,N1
    FI=I-1
    TPNF=TPI*FI/XN
    DO 10 J=1,N2
      L=J-1
      WRD(J)=XR(N1*L+I)
      WID(J)=XI(N1*L+I)
10  CONTINUE
    CALL FFT5(K2,N2,N05,SGN,LIST,WRD,WID,IRBD,COD,SID,CO1,SI1,
+CO2,SI2,CO3,SI3)
    DO 20 J=1,N2
      L=J-1
      XJ=J-1
      THET=SGN*TPNF*XJ
      C=COS(THET)
      S=SIN(THET)
      XR(N1*L+I)=WRD(J)*C-WID(J)*S
      XI(N1*L+I)=WRD(J)*S+WID(J)*C
20  CONTINUE
30  CONTINUE
    DO 60 I=1,N2
      M=I-1
      DO 40 J=1,N1
        WR(J)=XR(N1*M+J)
        WI(J)=XI(N1*M+J)
```

PAGES _____
ARE
MISSING
IN
ORIGINAL
DOCUMENT

HPDSIM.FOR

```

+SI3(1),W(1)
C *****
C THIS SUBROUTINE JOINS THE FFT2, THE FFT3, AND THE FFT5. WHEN
C IT IS USED, THE SUBROUTINES FFT2, FFT3, AND FFT5, MUST BE PRESENT.
C N IS THE NUMBER OF MEASUREMENTS. N IS OF THE FORM N=(2**K1)*
C (3**K2)*(5**K3). N1=2**K1, N2=3**K2, N3=5**K3. NO2=N1/2,
C NO3=N2/3, NO5=N3/5. SGN=-1, IF THE FOURIER TRANSFORM IS TO BE
C PERFORMED, AND SGN=+1 IF THE INVERSE FOURIER TRANSFORM IS TO BE
C PERFORMED. IF LIST=0, THE REVERSED-DIGIT LIST AND THE ASSOCIATED
C SINES AND COSINES ARE COMPUTED. IF LIST=1, THE LIST IS NOT
C COMPUTED. INSTEAD, THE LIST COMPUTED ON THE PREVIOUS CALL IS
C USED. XR IS THE REAL PART AND XI IS THE IMAGINARY PART OF THE
C COMPLEX SEQUENCE TO BE TRANSFORMED. AFTER THE FFT235 IS PERFORMED
C XR AND XI ARE THE REAL AND IMAGINARY PARTS OF THE TRANSFORMED DATA.
C IRB, IRBD, IRBT, CO, SI, COD, SID, COT, SIT, WR, WI, WRD, WID, WRT,
C WIT, CO1, CO2, CO3, SI1, SI2, SI3, AND W ARE WORK VECTORS.
C *****
TPI=6.283185307
XN=N
XN2=N2
XN3=N3
DO 65 I=1,N1
XL=I-1
TP1=(XL*TPI)/XN
TP3=TP1*XN3
DO 30 J=1,N2
XJ=J-1
TP2=TPI*XJ/(XN2*XN3)+TP1
DO 10 K=1,N3
WRT(K)=XR(N1*N2*(K-1)+N1*(J-1)+I)
WIT(K)=XI(N1*N2*(K-1)+N1*(J-1)+I)
10 CONTINUE
CALL FFT5(K3,N3,NO5,SGN,LIST,WRT,WIT,IRBT,COT,SIT,CO1,SI1,CO2,SI2,
+CO3,SI3)
DO 20 K=1,N3
XK=K-1
THET=SGN*TP2*XK
C=COS(THET)
S=SIN(THET)
XR(N1*N2*(K-1)+N1*(J-1)+I)=WRT(K)*C-WIT(K)*S
XI(N1*N2*(K-1)+N1*(J-1)+I)=WIT(K)*C+WRT(K)*S
20 CONTINUE
30 CONTINUE
DO 60 K=1,N3
KM=K-1
DO 40 J=1,N2
WRD(J)=XR(N1*N2*KM+N1*(J-1)+I)
WID(J)=XI(N1*N2*KM+N1*(J-1)+I)
40 CONTINUE
CALL FFT3(K2,N2,NO3,SGN,LIST,WRD,WID,IRBD,COD,SID)
DO 50 J=1,N2

```

HPDSIM.FOR

```

      XJ=J-1
      THET=SGN*TP3*XJ
      C=COS(THET)
      S=SIN(THET)
      XR(N1*N2*KM+N1*(J-1)+I)=WRD(J)*C-WID(J)*S
      XI(N1*N2*KM+N1*(J-1)+I)=WRD(J)*S+WID(J)*C
50    CONTINUE
60    CONTINUE
65    CONTINUE
      DO 90 K=1,N3
      KM=K-1
      DO 80 J=1,N2
      KJ=J-1
      DO 70 I=1,N1
      WR(I)=XR(N1*N2*KM+N1*KJ+I)
      WI(I)=XI(N1*N2*KM+N1*KJ+I)
70    CONTINUE
      CALL FFT2(K1,N1,N02,SGN,LIST,WR,WI,IRB,CO,SI)
      DO 80 I=1,N1
      XR(N1*N2*KM+N1*KJ+I)=WR(I)
      XI(N1*N2*KM+N1*KJ+I)=WI(I)
80    CONTINUE
90    CONTINUE
C     THE FFT COEFFICIENTS ARE UNSCRAMBELED INTO NORMAL ORDER
      DO 100 I=1,N
      W(I)=XR(I)
100   CONTINUE
      DO 110 I=1,N1
      DO 110 J=1,N2
      MJ=J-1
      DO 110 K=1,N3
      MK=K-1
      XR((I-1)*N2*N3+N3*MJ+K)=W(N1*N2*MK+N1*MJ+I)
110   CONTINUE
      DO 120 I=1,N
      W(I)=XI(I)
120   CONTINUE
      DO 130 I=1,N1
      DO 130 J=1,N2
      MJ=J-1
      DO 130 K=1,N3
      MK=K-1
      XI((I-1)*N2*N3+N3*MJ+K)=W(N1*N2*MK+N1*MJ+I)
130   CONTINUE
      RETURN
      END

C
      SUBROUTINE INTERP(N,NMAX,NS, .SMAX,VSIM,RSIM,IBOUND,Z,U,
&A,B,C,R,K,IDOC)
C     *****
C     INTERPOLATE FROM VSIM TO RSIM

```

HPDSIM.FOR

```

C
C   [VSIM(J),J=1,N] = SIMULATED NORMAL SCORE SEQUENCE
C   [RSIM(J),J=1,N] = SIMULATED RESIDUALS BACK INTERPOLATED
C                       FROM VSIM.
C   IBOUND,Z,U,A,B,C: SEE THE DESCRIPTIVE HEADING FOR SUBR. TAIL
C                       FOR THE DEFINITIONS OF THESE ARRAYS.
C   R(J) = RANKED DATA RESIDUALS FOR BACK INTERPOLATION
C
C   WRITTEN BY LEON BORGMAN, UNIVERSITY OF WYOMING
C *****
C   DIMENSION IBOUND(2),Z(3,2),U(3,2),A(2),B(2),C(2)
C   DIMENSION VSIM(NSMAX),RSIM(NSMAX),R(NMAX)

C ***** BACK INTERPOLATE FROM VSIM TO RSIM ***
C   IF (IDOC.NE.0) THEN
C       WRITE (4,100)
100   FORMAT (//1X,'DOCUMENTATION OF BACK INTERP. FOR JK<20',/
C   & ' AND ALL TAIL INTERPOLATIONS')
C   END IF
C   DO 1 J=1,NSIM
C       CALL DFNORM(VSIM(J),UU)
C       IF (UU.GT.U(1,1).AND.UU.LT.U(1,2)) THEN
C           ***** INTERPOLATE WITHIN THE EMPIRICAL DIST. ***
C           JL=UU*(N+1.0)
C           JR=JL+1
C           UJL=JL/(N+1.0)
C           UJR=JR/(N+1.0)
C           RSIM(J)=((UJR-UU)*R(JL)+(UU-UJL)*R(JR))/(UJR-UJL)
C           IF (IDOC.NE.0.AND.J.LT.20) THEN
101             WRITE (4,101) K,J,VSIM(J),UU
102             FORMAT (1X,'K,J,VSIM,UU= ',2I5,2F15.5)
103             WRITE (4,102)
104             FORMAT (1X,'      INTERP. IN EMPIR. DISTN.')

```

HPDSIM.FOR

```

        WRITE (4,104)
104      FORMAT (1X,'      INTERP. IN LEFT TAIL.')
```

```

        WRITE (4,110) Z(1,1),Z(2,1),Z(3,1)
110      FORMAT (1X,'      Z1,Z2,Z3: ',3F15.5)
        WRITE (4,111) U(1,1),U(2,1),U(3,1)
111      FORMAT (1X,'      U1,U2,U3: ',3F15.5)
        WRITE (4,105) TT,A(1),B(1),ZZ1
105      FORMAT (1X,'      TT,A,B,ZZ1= ',4F15.5)
        WRITE (4,106) U(1,1),Z(1,1),C(1),RSIM(J)
106      FORMAT (1X,'      U1,Z11,C,RSIM= ',4F15.5)

        END IF
        ELSE IF (UU.GE.U(1,2)) THEN
C      ***** USE FITTED CURVE FOR RIGHT TAIL ***
        IF (IBOUND(2).EQ.0) THEN
C      ***** UNBOUNDED TO RIGHT CASE. ***
            TT=-2.0*ALOG(1.0-UU)
            TT=SQRT(TT)
        ELSE IF (IBOUND(2).EQ.1) THEN
C      ***** BOUNDED TO RIGHT BY Z(1,2). ***
            TT=UU-U(1,2)
        END IF
        ZZ2=ABS(A(2)+B(2)*TT)
        RSIM(J)=Z(1,2)+ZZ2*C(2)
        IF (IDOC.NE.0) THEN
            WRITE (4,101) K,J,VSIM(J),UU
            WRITE (4,107)
107      FORMAT (1X,'      INTERP. IN RIGHT TAIL.')
```

```

        WRITE (4,112) Z(1,2),Z(2,2),Z(3,2)
112      FORMAT (1X,'      Z1,Z2,Z3: ',3F15.5)
        WRITE (4,114) U(1,2),U(2,2),U(3,2)
114      FORMAT (1X,'      U1,U2,U3: ',3F15.5)
        WRITE (4,108) TT,A(2),B(2),ZZ2
108      FORMAT (1X,'      TT,A,B,ZZ2= ',4F15.5)
        WRITE (4,109) U(1,2),Z(1,2),C(2),RSIM(J)
109      FORMAT (1X,'      U12,Z12,C,RSIM= ',4F15.5)

        END IF
        END IF
1 CONTINUE

        RETURN
        END

C      SUBROUTINE DFNORM (Z,P)
C      *****
C      SUBROUTINE COMPUTES PROBABILITY ,P, THAT A STANDARD NORMAL
C      RANDOM VARIABLE IS LESS THAN OR EQUAL TO Z
C      THE METHOD USED IS FROM: HANDBOOK OF MATH. FUNCTIONS,
C      EDITED BY ABRAMOWITZ AND STEGUN, PAGE 932, EQ.26.6.17,
C      IN THE CHAPTER BY ZELEN AND SEVERO
C      *****
C      ZZ=Z
```

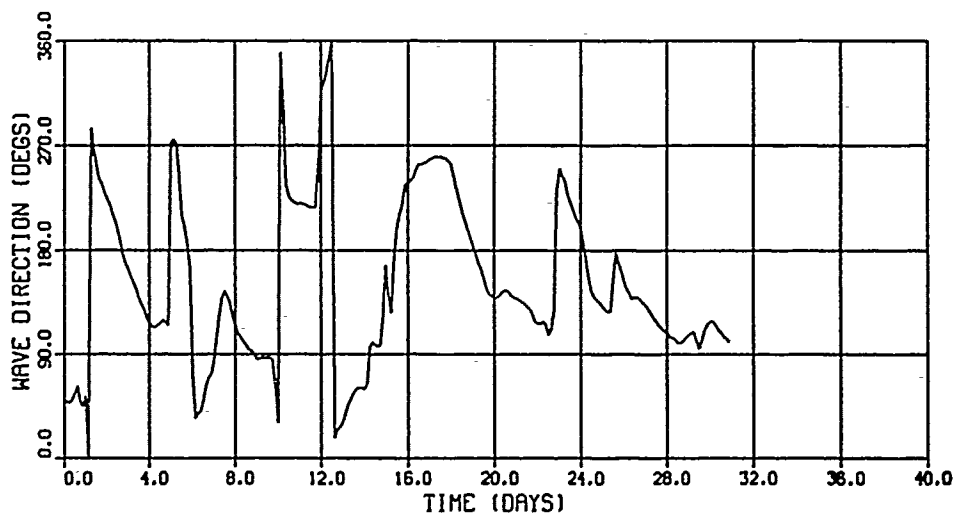
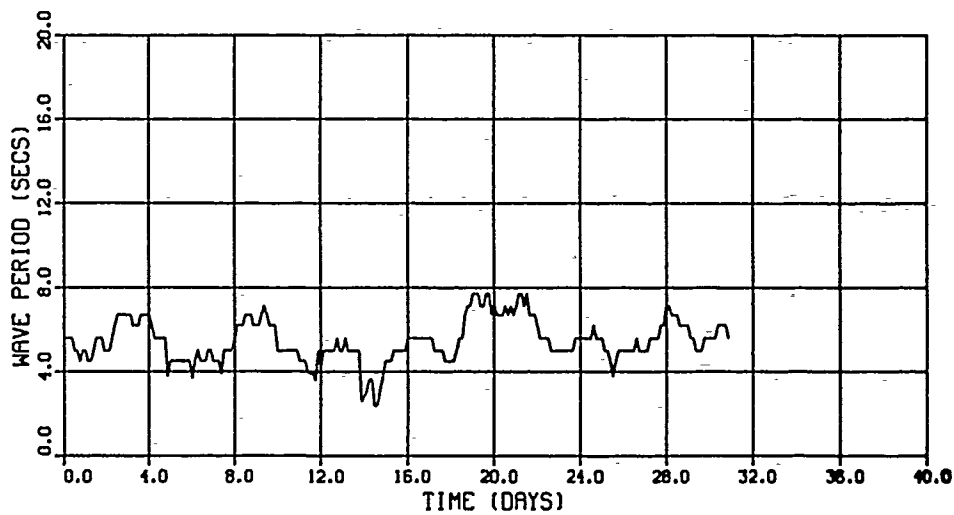
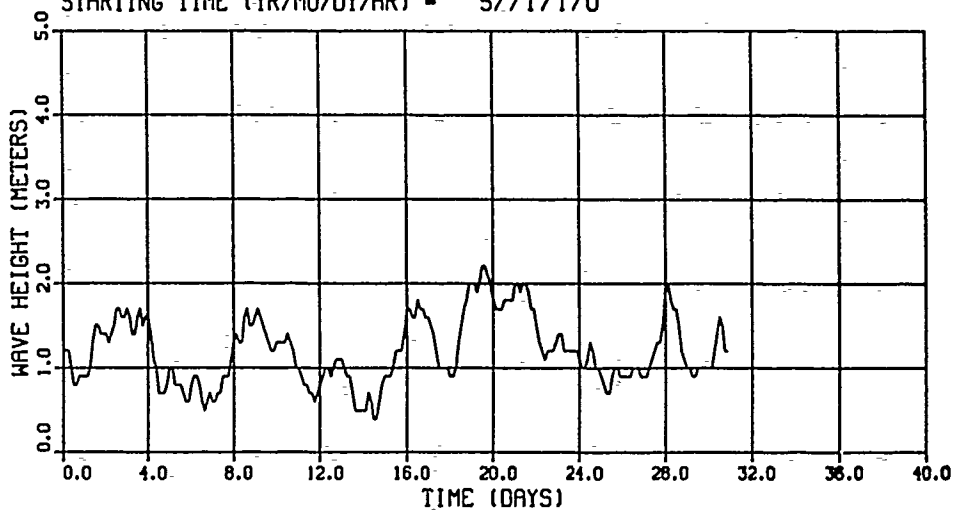
HPDSIM.FOR

```
IF (Z.LT.0.0) ZZ=-ZZ
T=1.0/(1.0+0.2316419*ZZ)
E=-(ZZ*ZZ)/2.0
IF (E.LT.-20.0) P=1.0
IF (E.LT.-20.0) GO TO 1
ZX=EXP(E)/2.50662828
P=1.0-ZX*(((1.330274429*T-1.821255978)*T
++1.781477937)*T-0.356563782)*T+.319381530)*T
1 IF (Z.LT.0.0) P=1.0-P
RETURN
END
```

APPENDIX G: A SELECTED YEAR OF WIS DATA

31 DAY WIS TIME SERIES

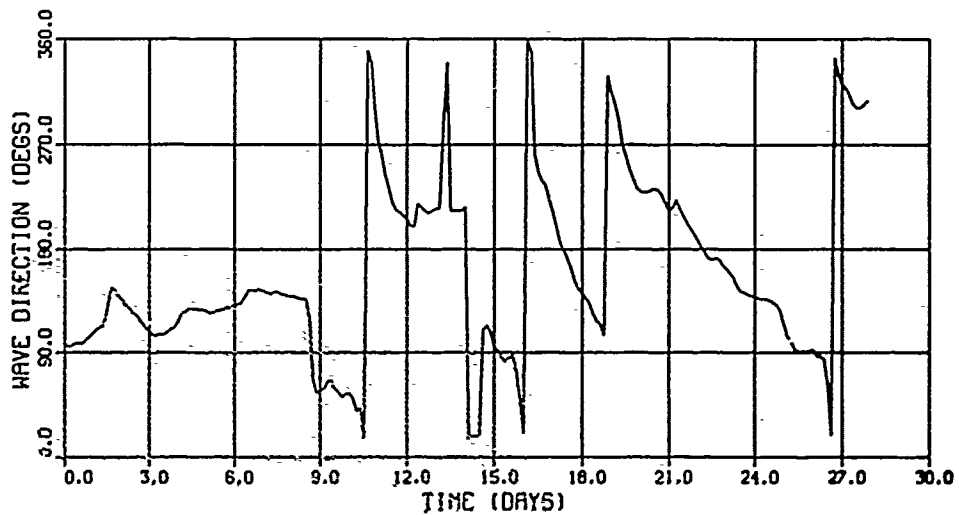
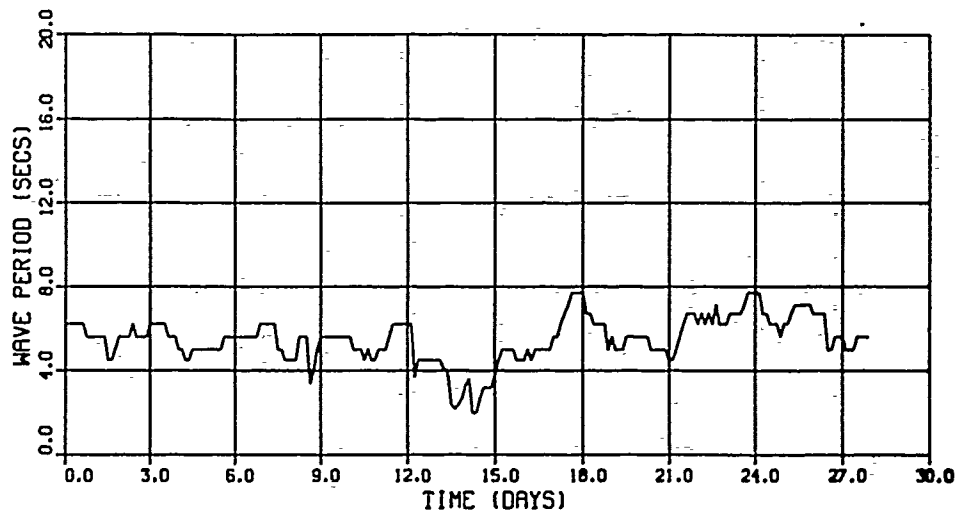
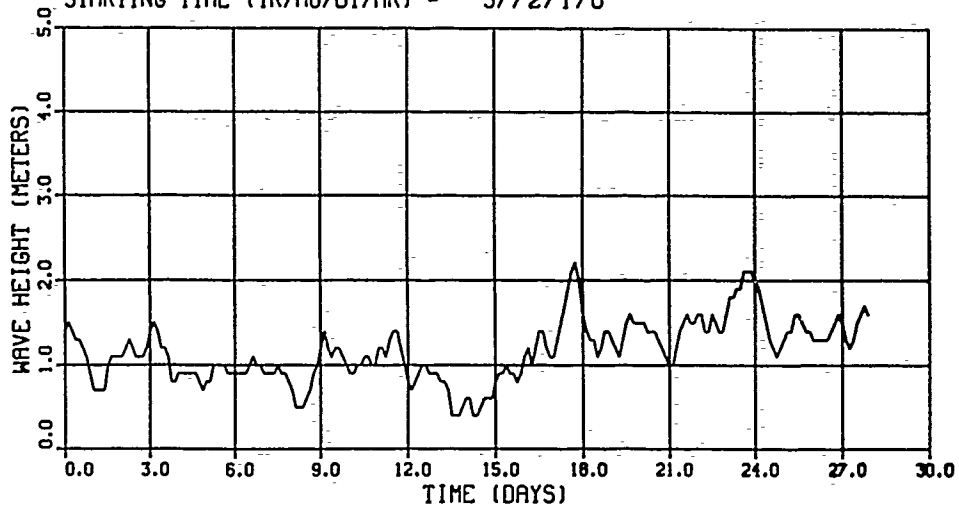
STARTING TIME (YR/MO/DY/HR) - 57/1/1/0



WIS Station #47 Wave Data for January 1957

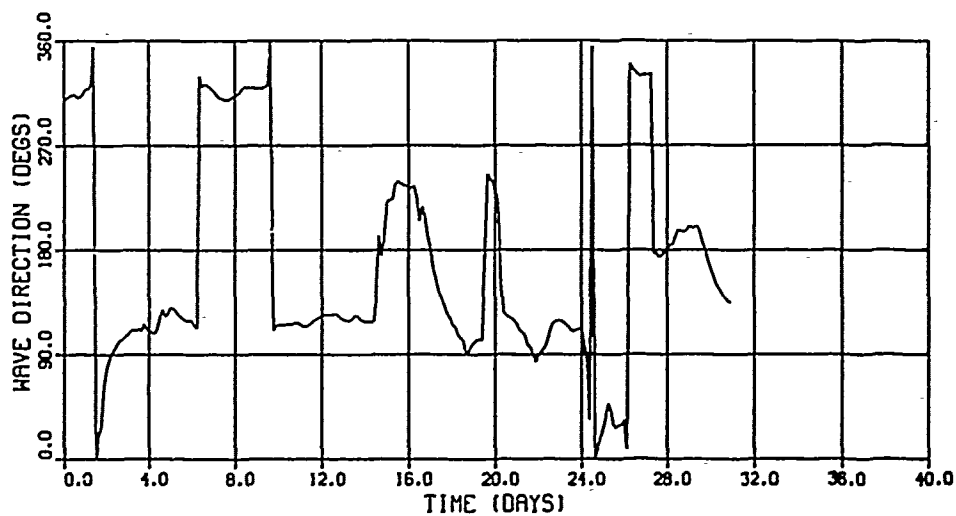
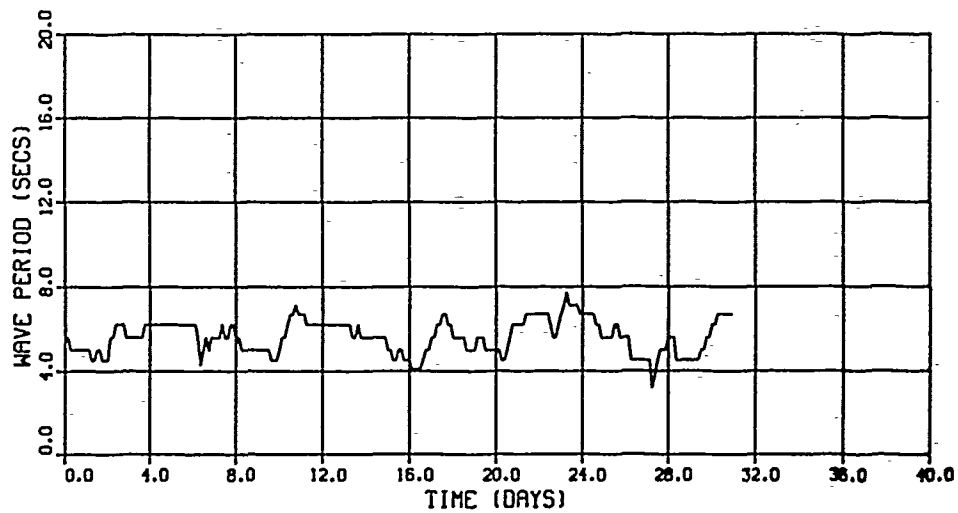
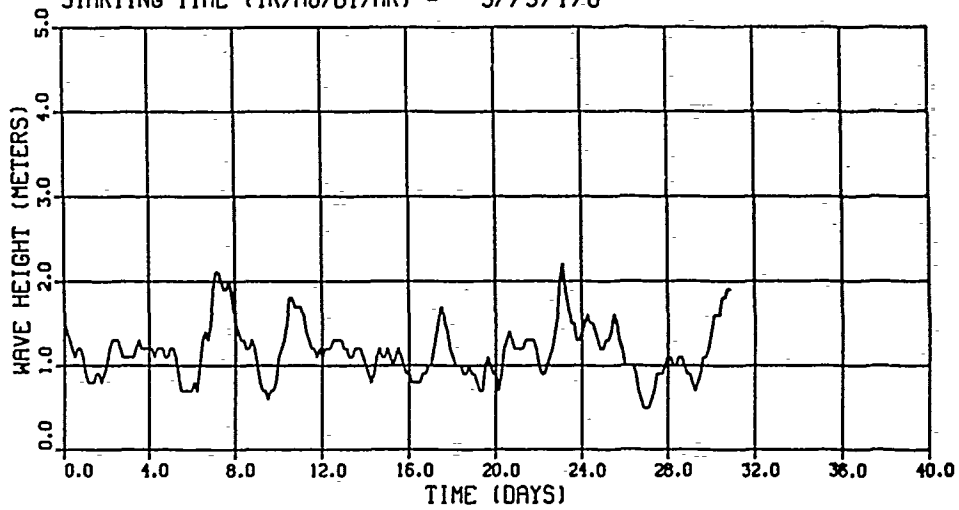
28 DAY WIS TIME SERIES

STARTING TIME (YR/MO/DY/HR) - 57/2/1/0



WIS Station #47 Wave Data for February 1957

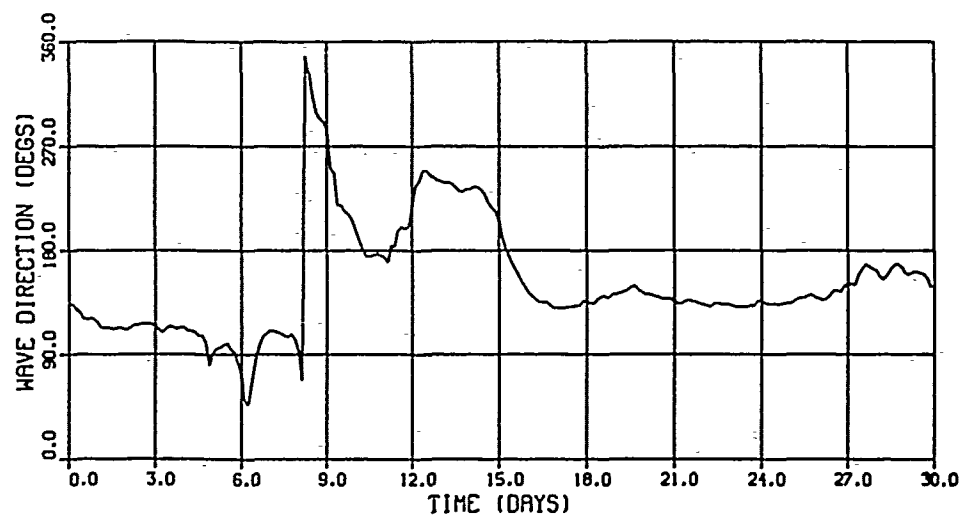
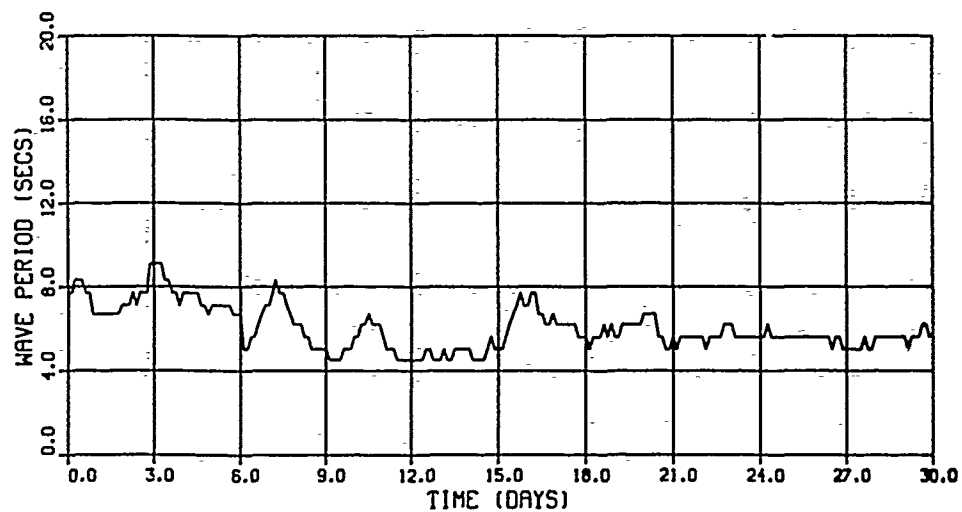
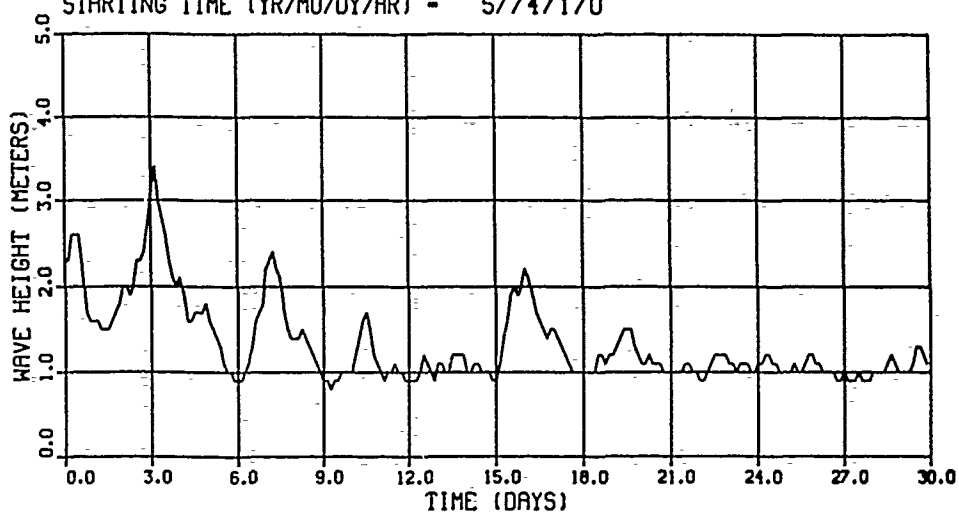
31 DAY WIS TIME SERIES
STARTING TIME (YR/MO/DY/HR) - 57/3/1/0



WIS Station #47 Wave Data for March 1957

30 DAY WIS-TIME SERIES

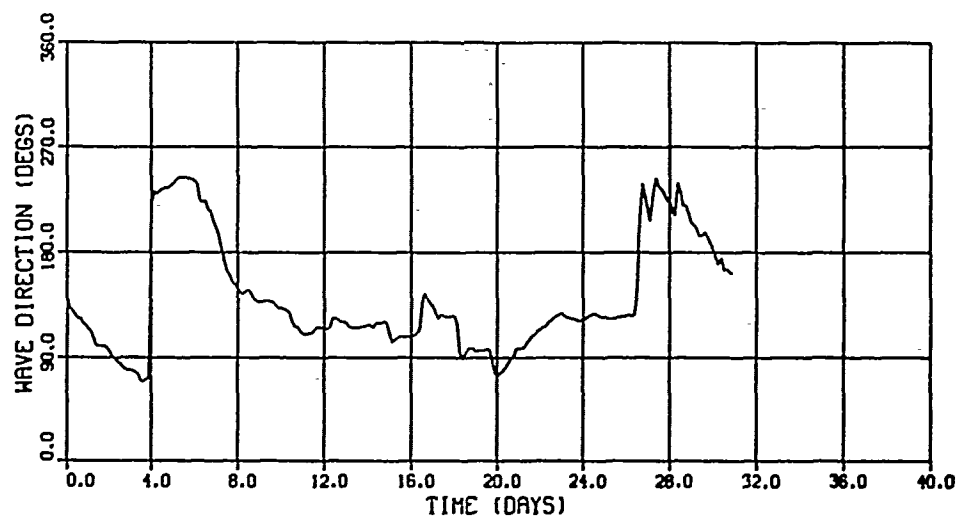
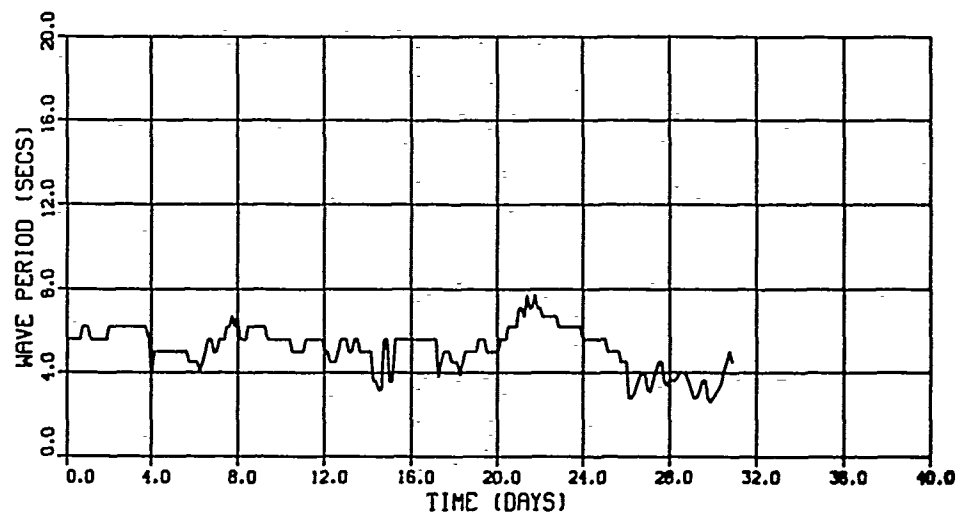
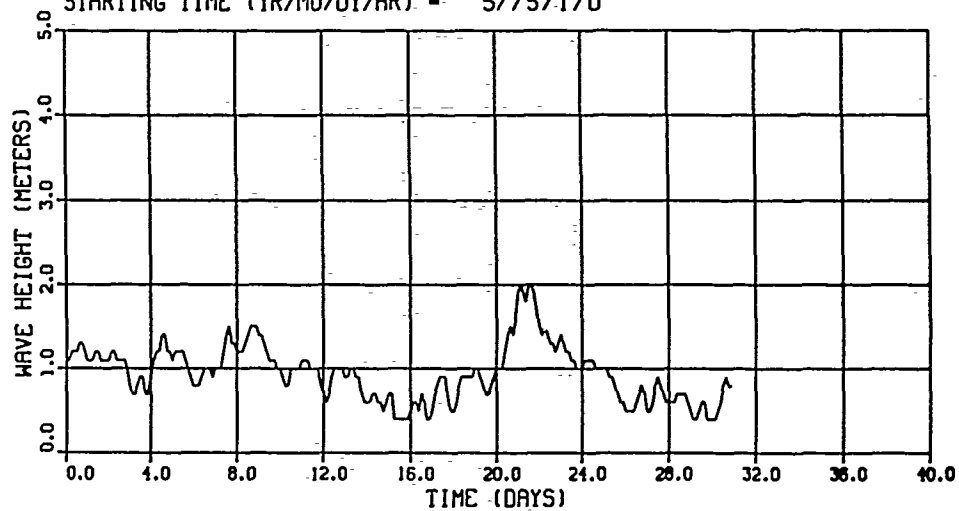
STARTING TIME (YR/MO/DY/HR) - 57/4/1/0



WIS Station #47 Wave Data for April 1957

31 DAY WIS TIME SERIES

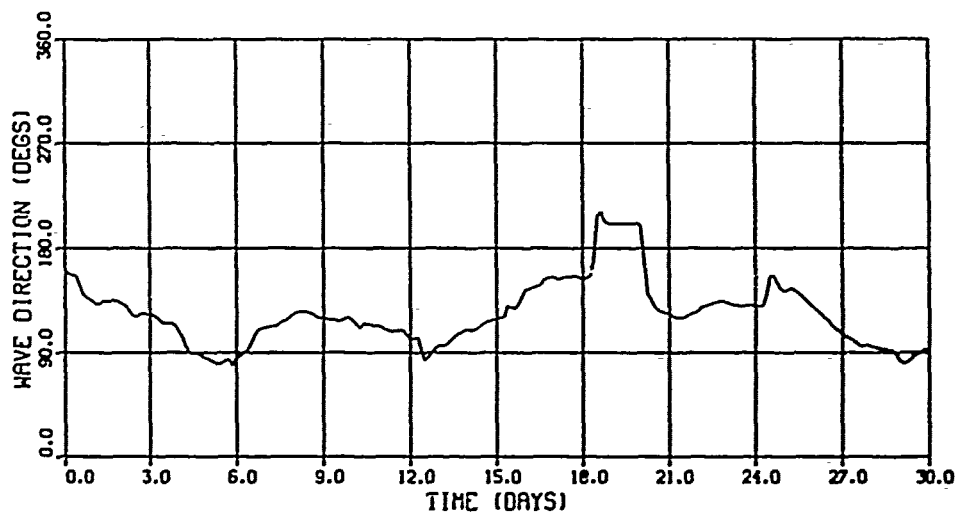
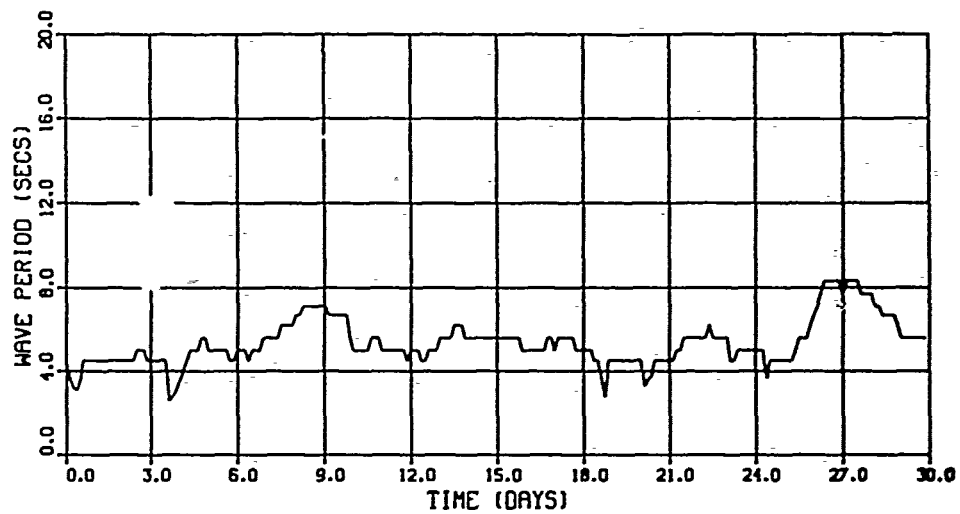
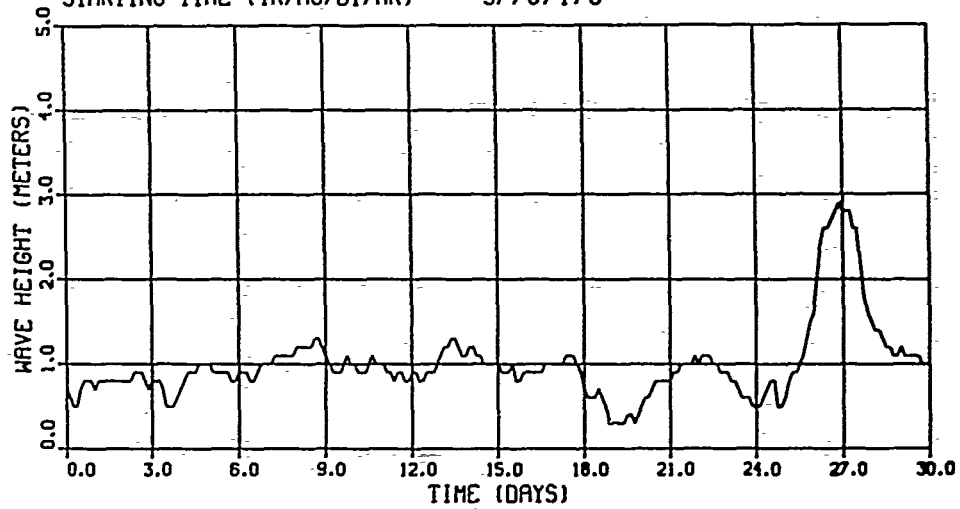
STARTING TIME (YR/MO/DY/HR) - 57/5/1/0



WIS Station #47 Wave Data for May 1957

30 DAY WIS TIME SERIES

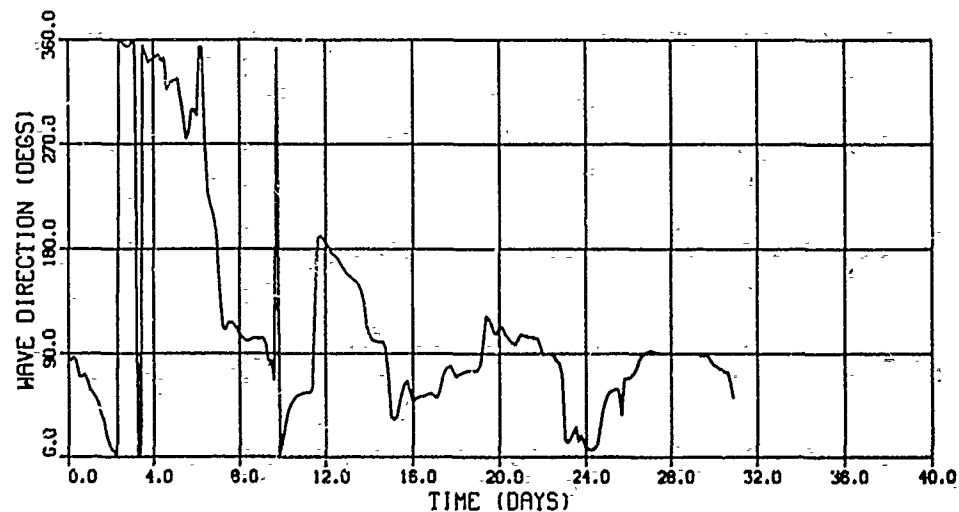
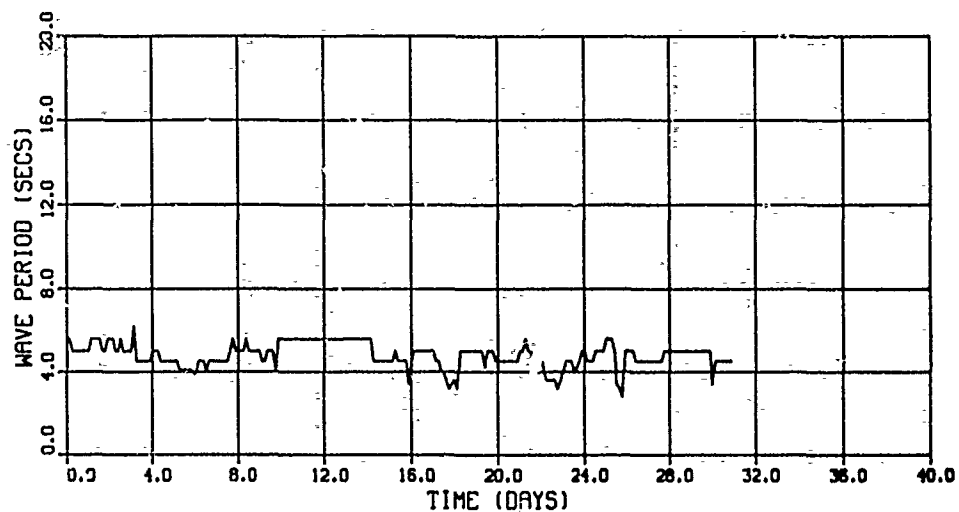
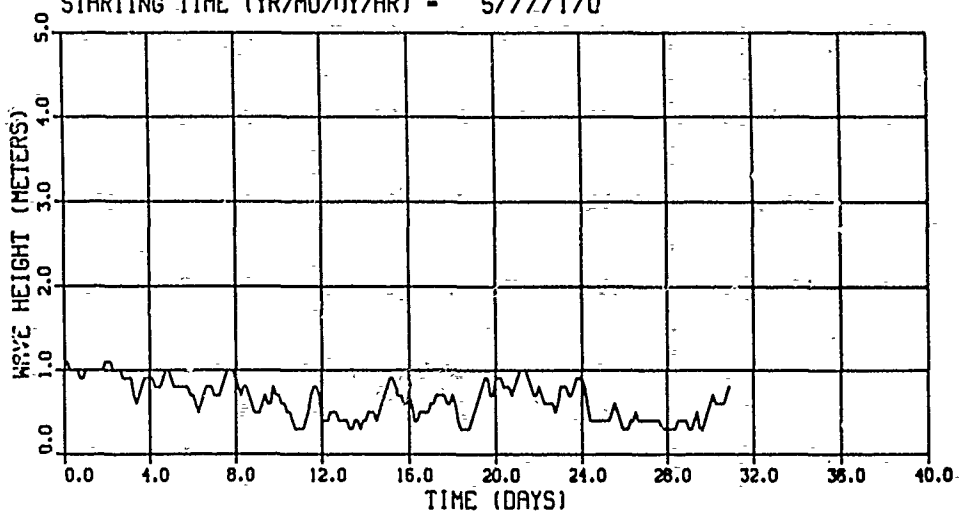
STARTING TIME (YR/MO/DY/HR) - 57/6/1/0



WIS Station #47 Wave Data for June 1957

31 DAY WIS TIME SERIES

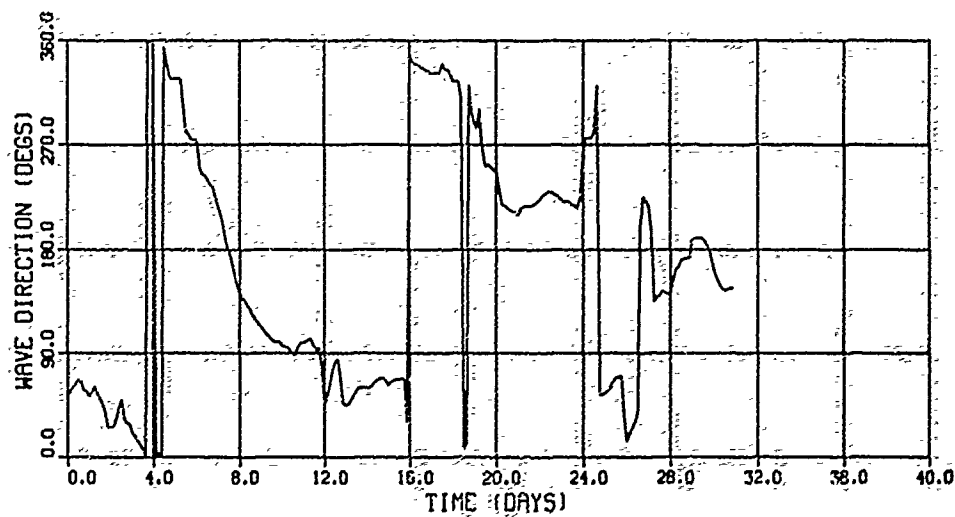
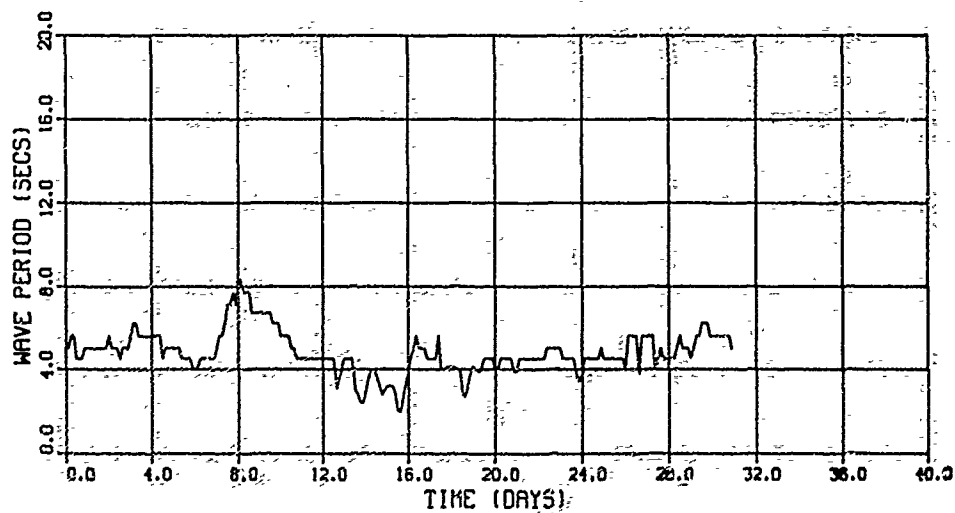
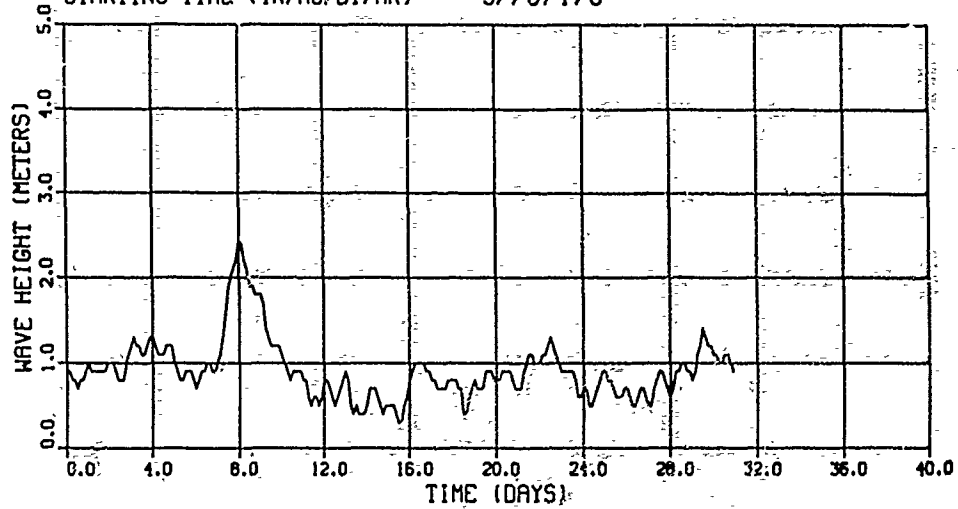
STARTING TIME (YR/MO/DY/HR) - 57/7/1/0



WIS Station #47 Wave Data for July 1957

31 DAY WIS TIME SERIES

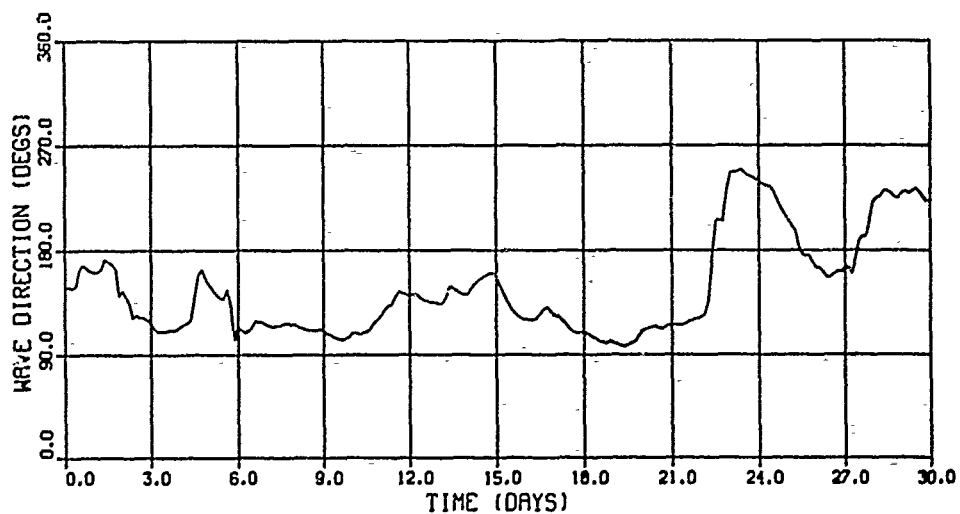
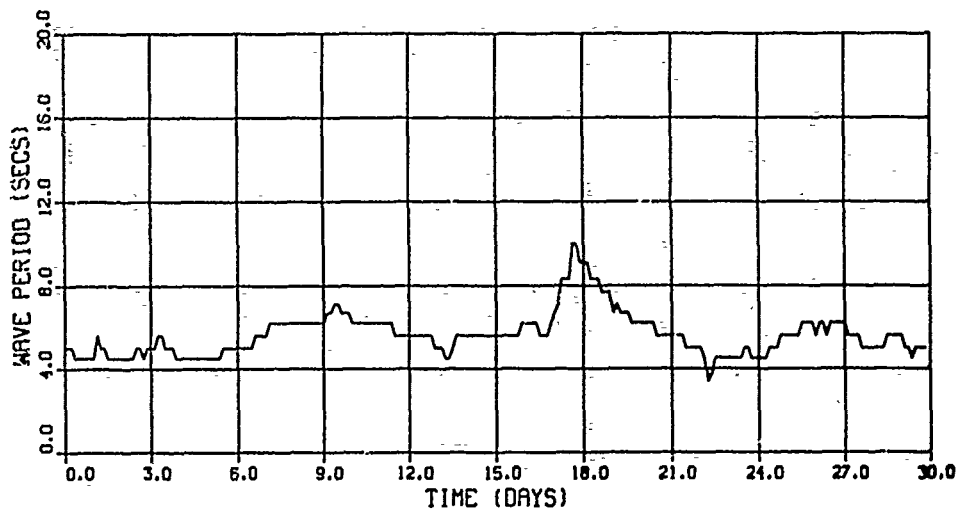
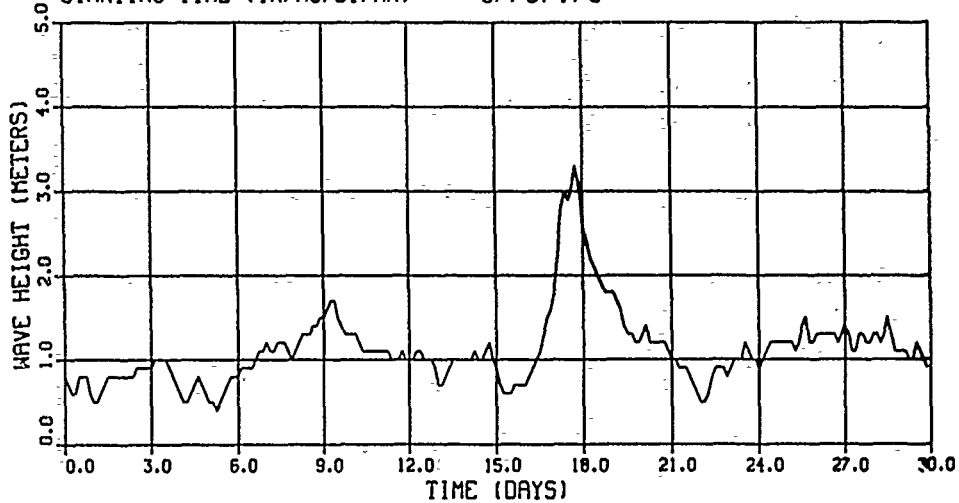
STARTING TIME (YR/MO/DY/HR) = 57/8/1/0



WIS Station #47 Wave Data for August 1957

30 DAY WIS TIME SERIES

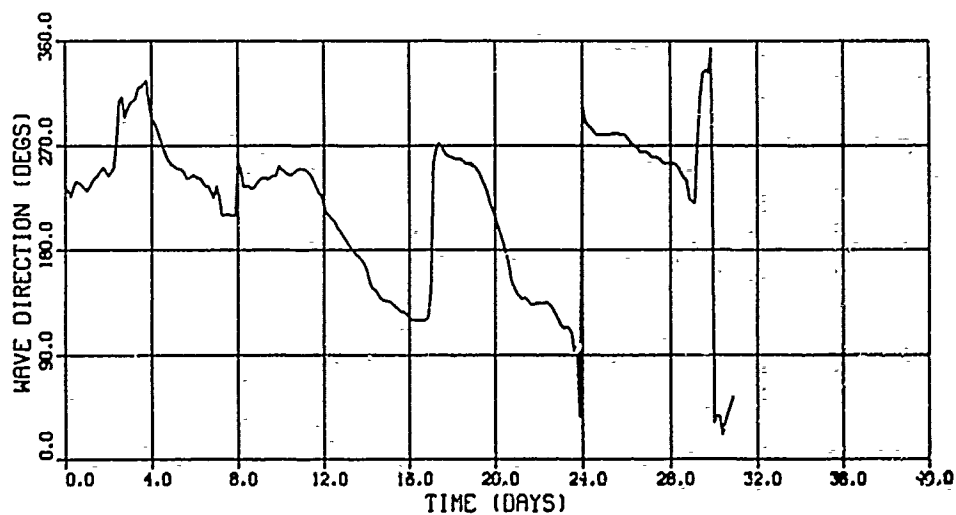
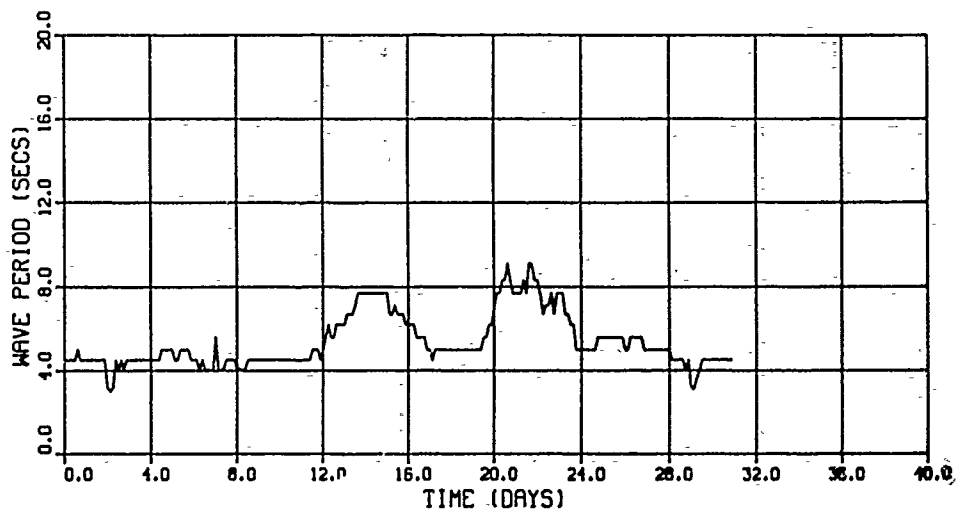
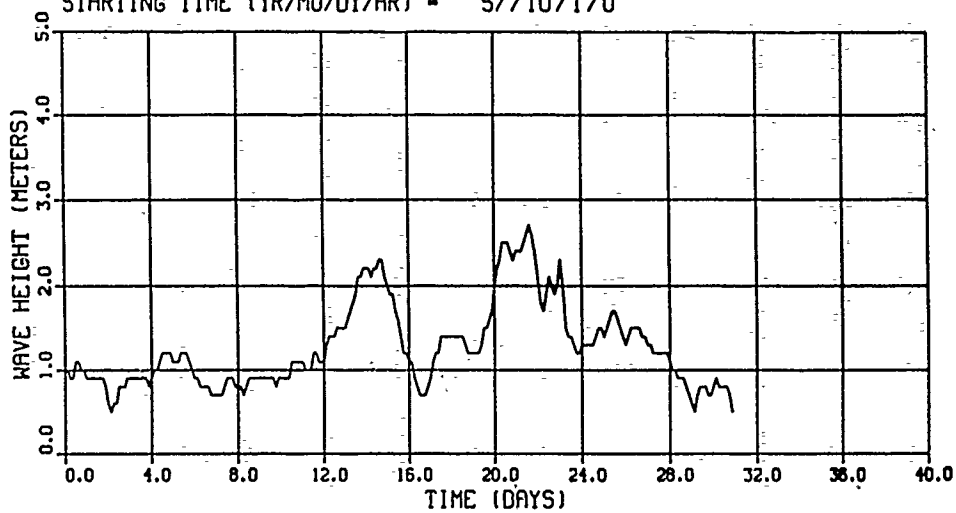
STARTING TIME (YR/MO/DY/HR) = 57/9/1/0



WIS Station 47 Wave Data for September 1957

31 DAY WIS TIME SERIES

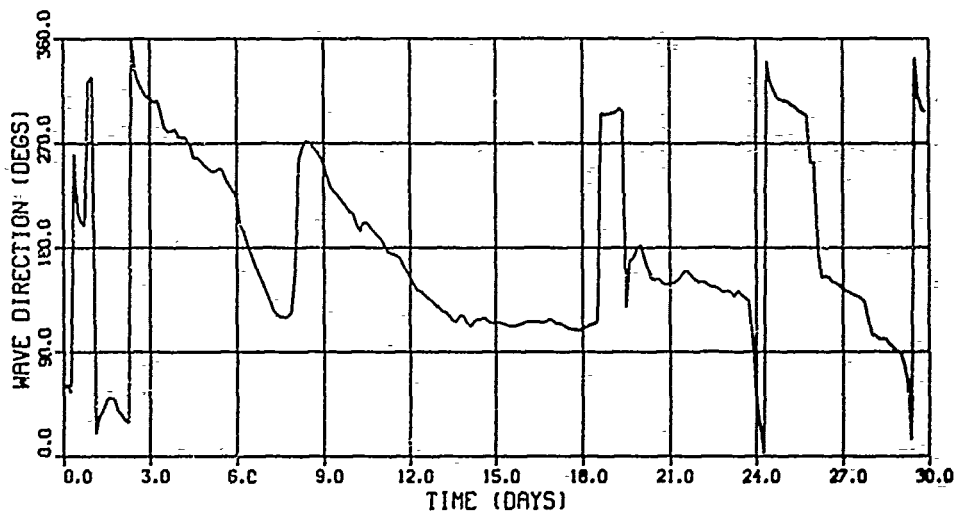
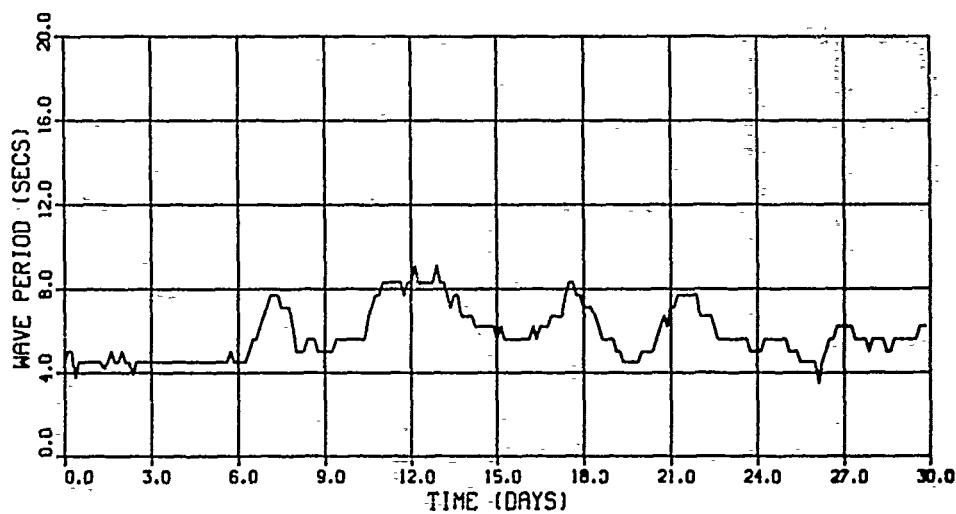
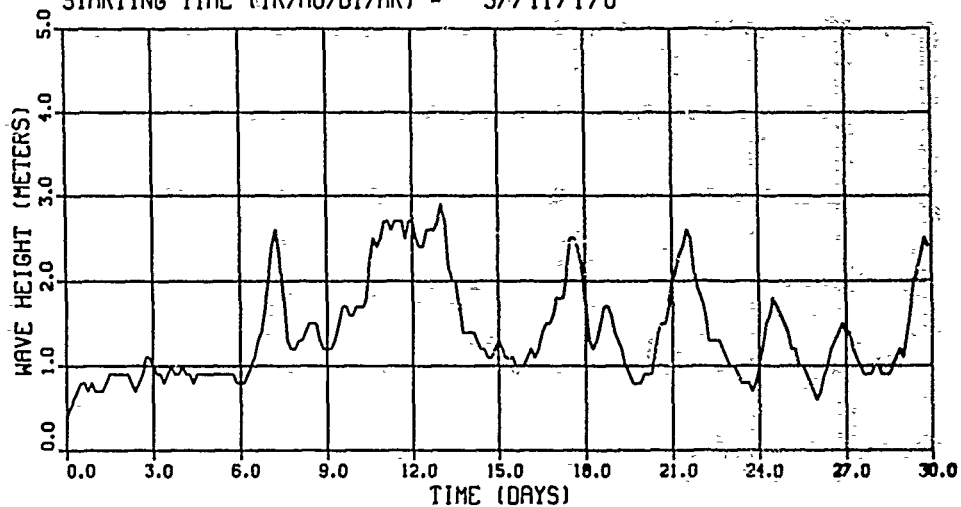
STARTING TIME (YR/MO/DY/HR) - 57/10/1/0



WIS Station 47 Wave Data for October 1957

30 DAY WIS TIME SERIES

STARTING TIME (YR/MO/DY/HR) = 57/11/1/0

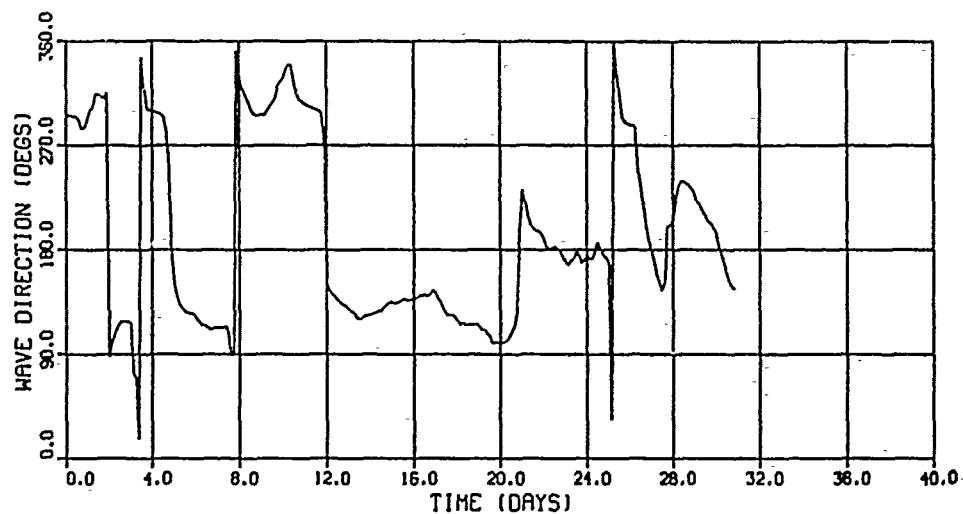
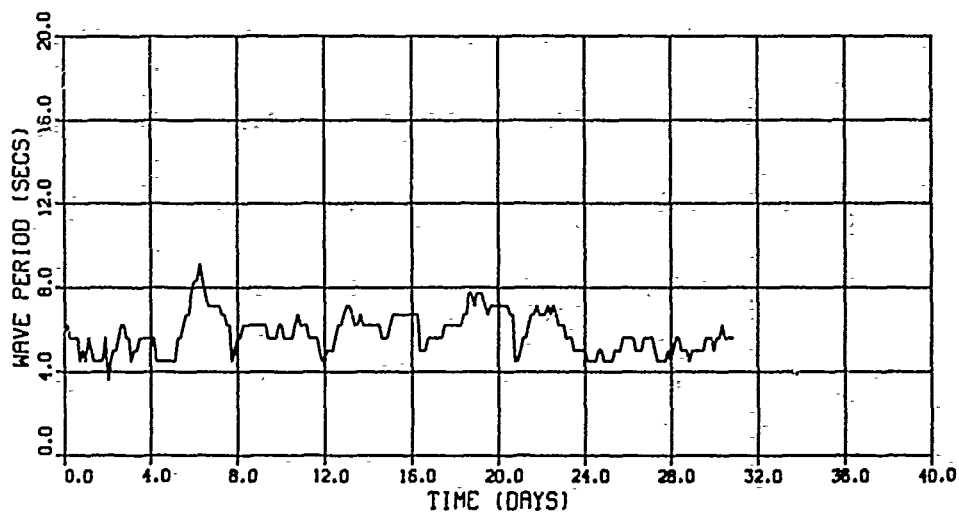
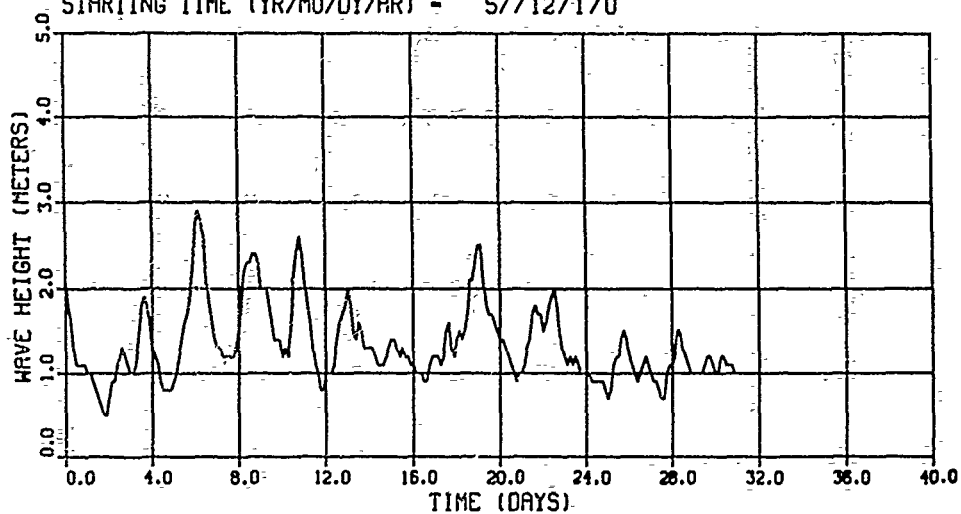


WIS Station 47 Wave Data for November 1957

G13

31 DAY WIS TIME SERIES

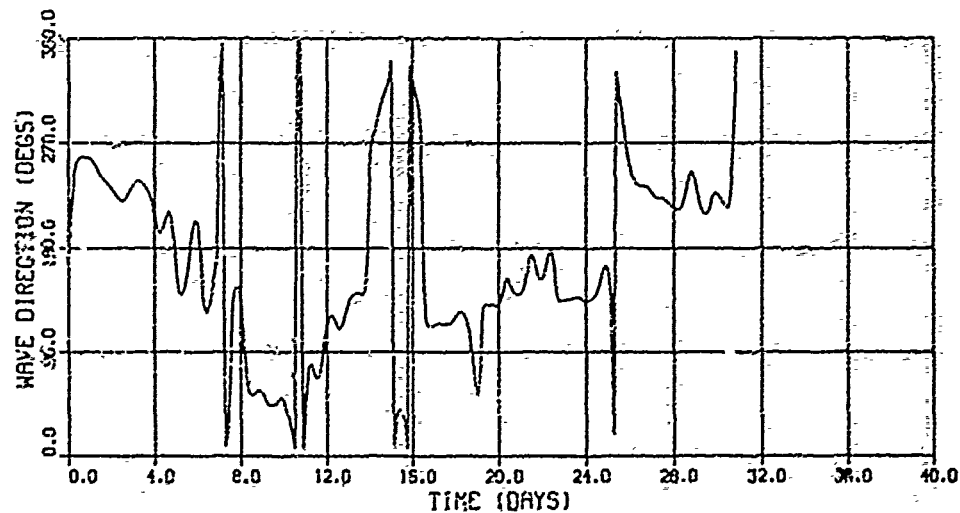
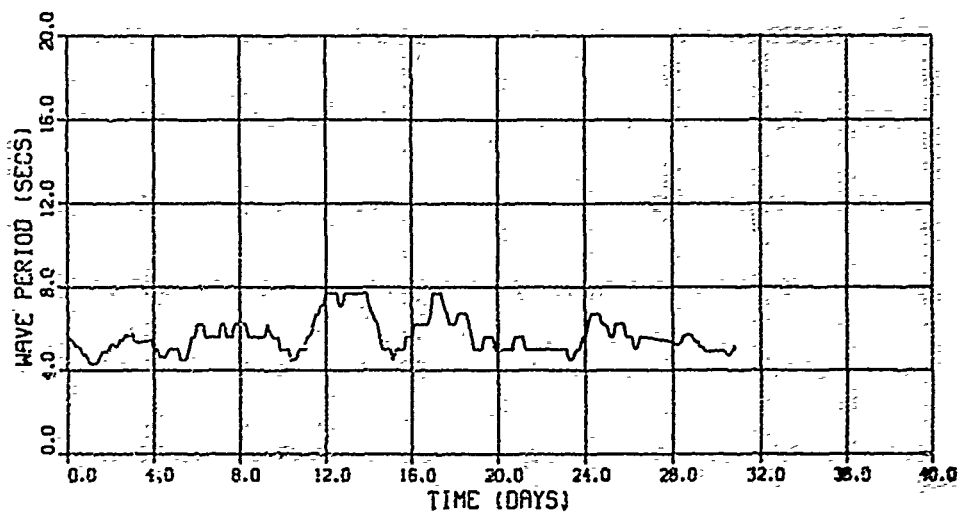
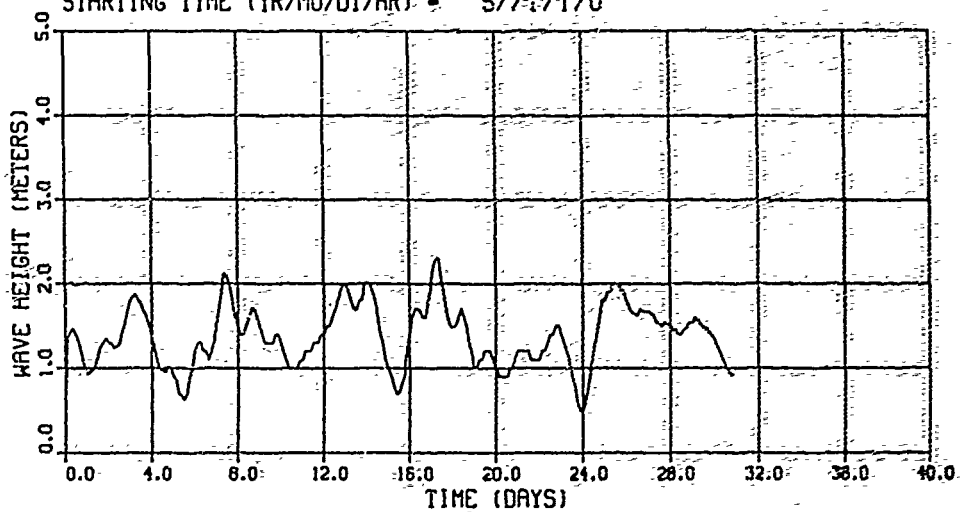
STARTING TIME (YR/MO/DY/HR) = 57/12/1/0



WIS Station #47 Wave Data for December 1957

APPENDIX H: A SIMULATED YEAR OF DATA IMITATING THE STATISTICS
OF THE YEAR IN APPENDIX B

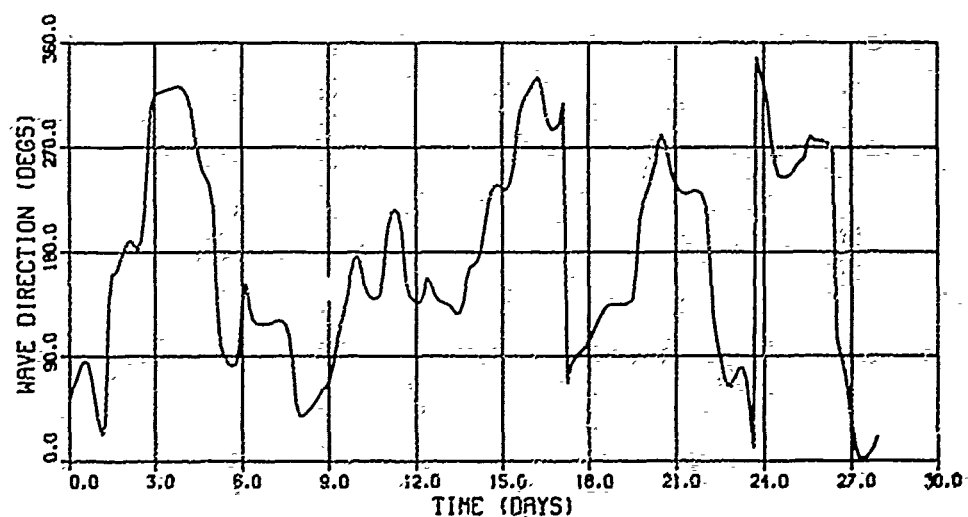
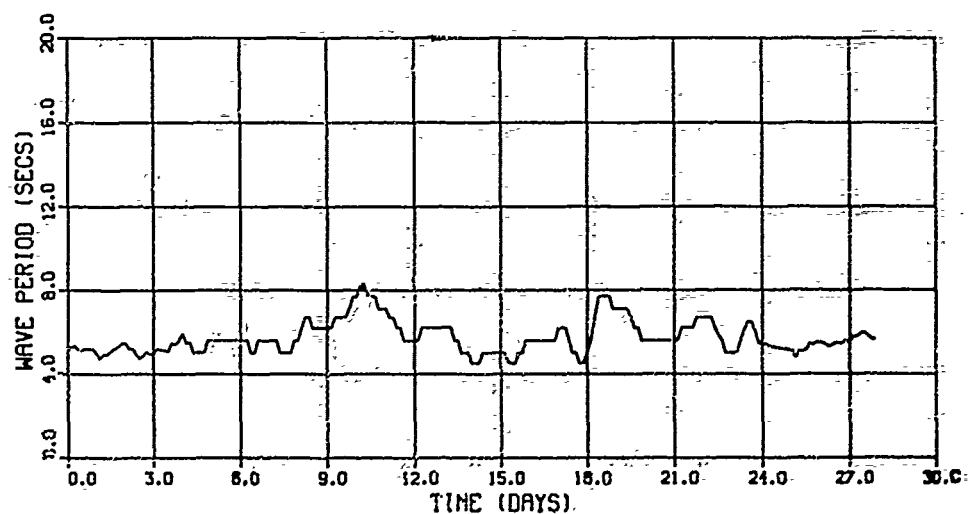
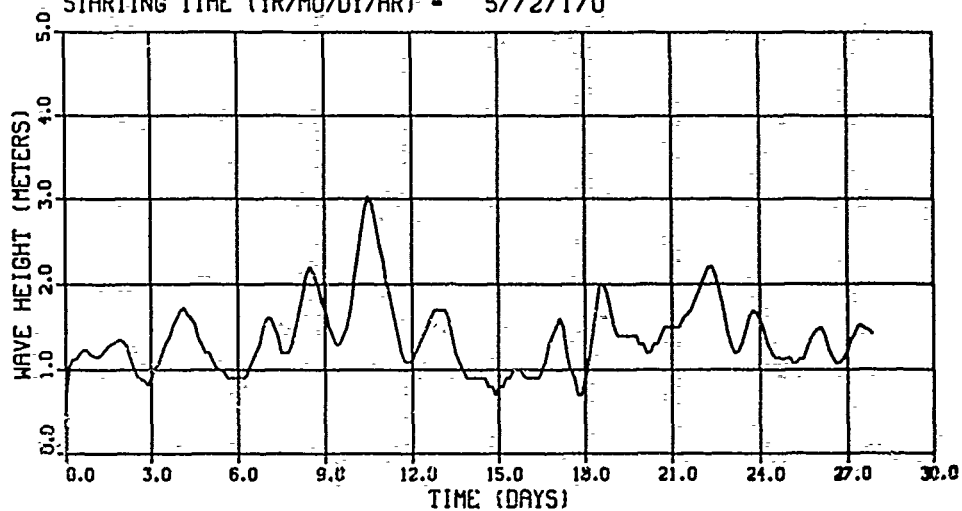
31 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) = 57/1/1/0



Simulated Wave Data for January 1957

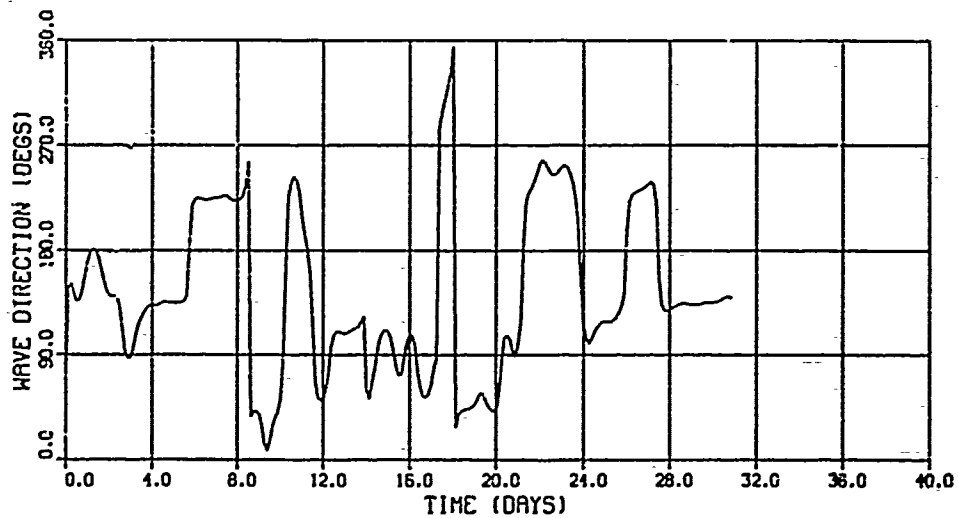
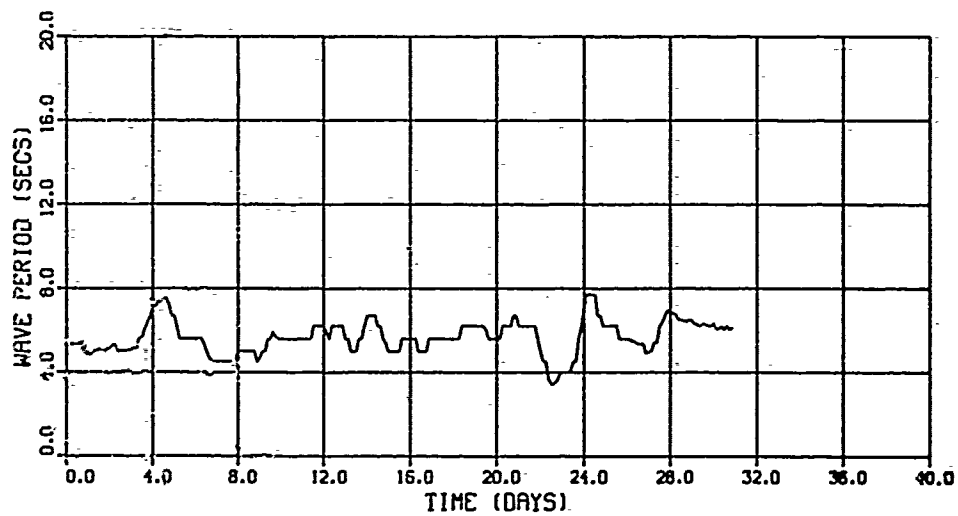
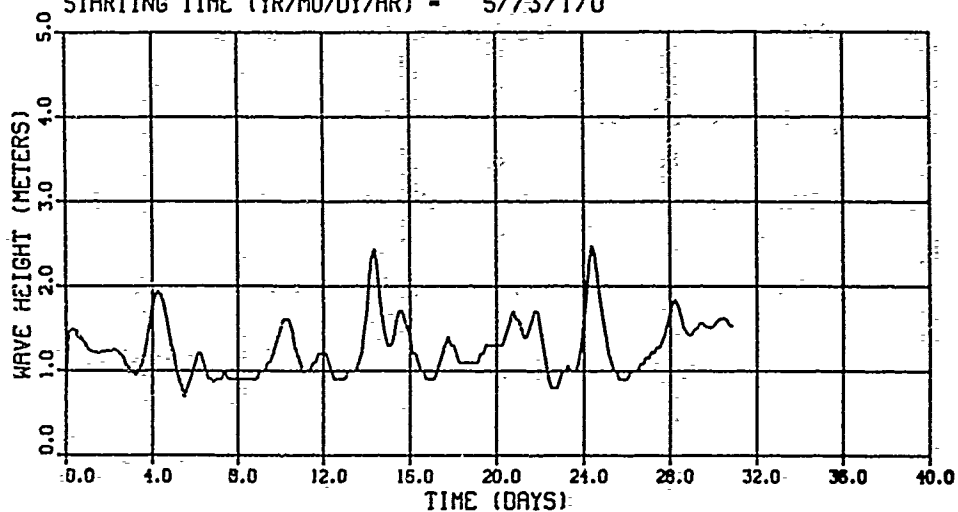
28 DAY SIMULATED HPD TIME SERIES

STARTING TIME (YR/MO/DY/HR) - 57/2/1/0



Simulated Wave Data for February 1957

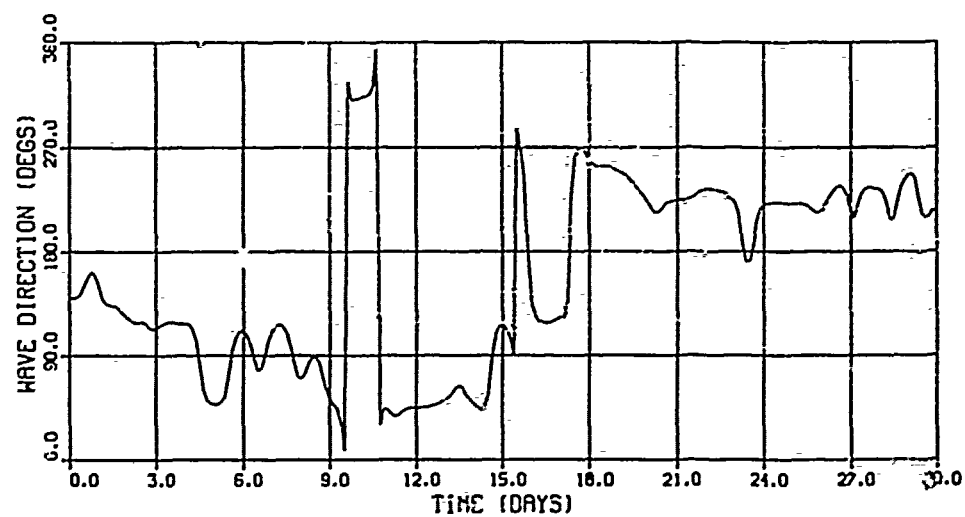
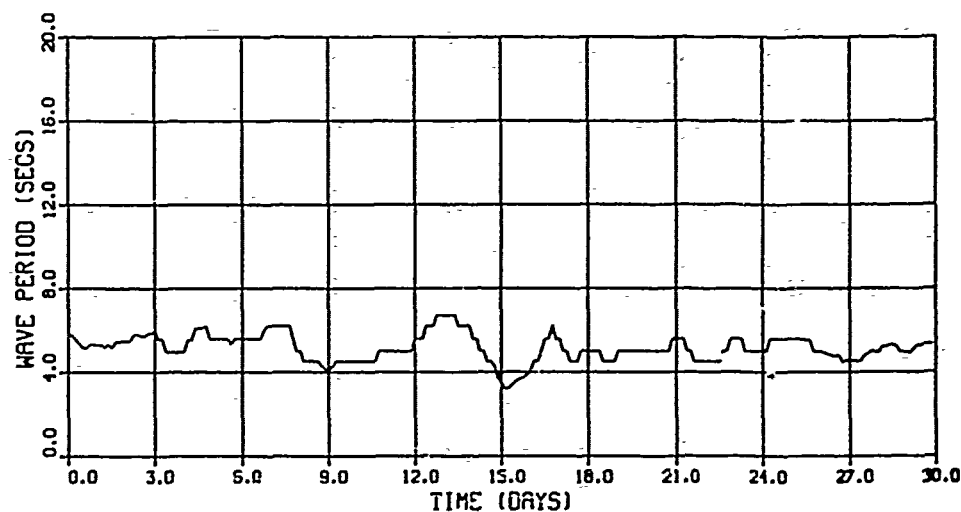
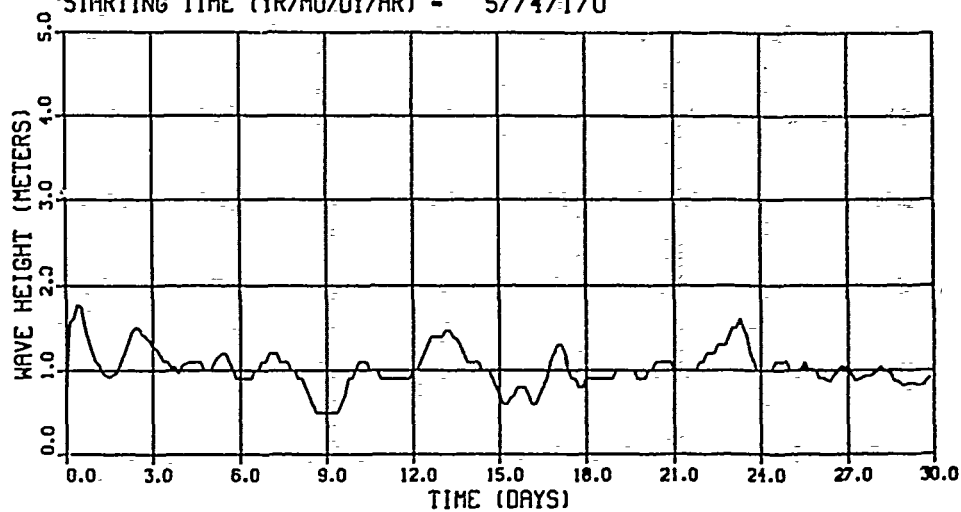
31 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) - 57/3/1/0



Simulated Wave Data for March 1957

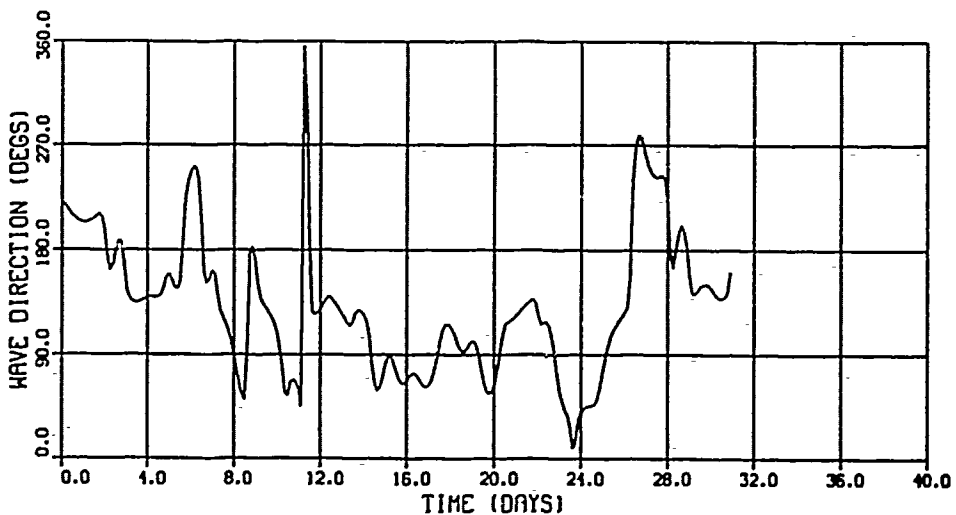
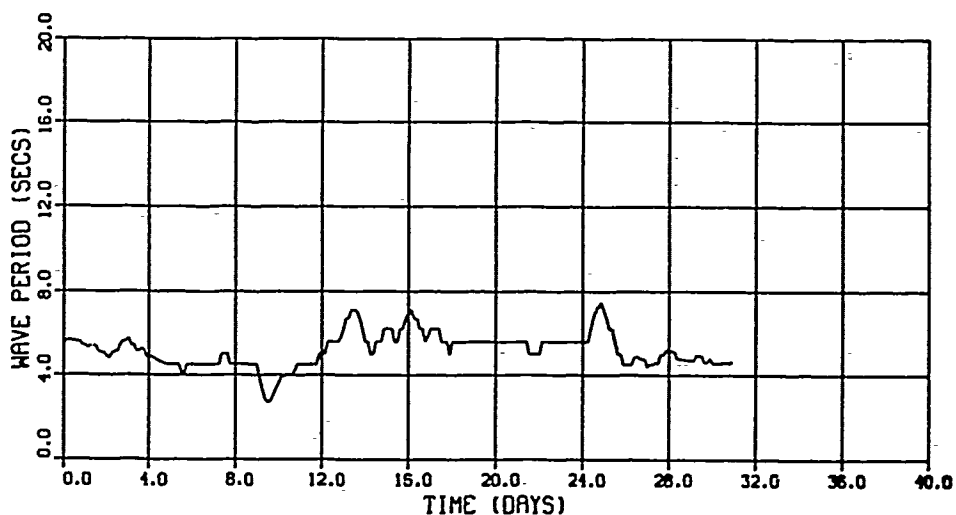
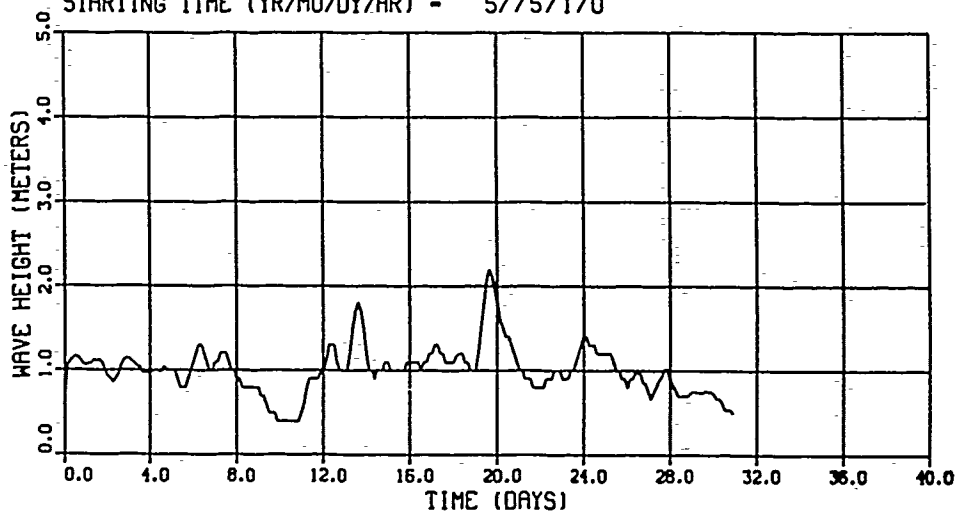
30 DAY SIMULATED HPD TIME SERIES

STARTING TIME (YR/MO/DY/HR) - 57/4/1/0



Simulated Wave Data for April 1957

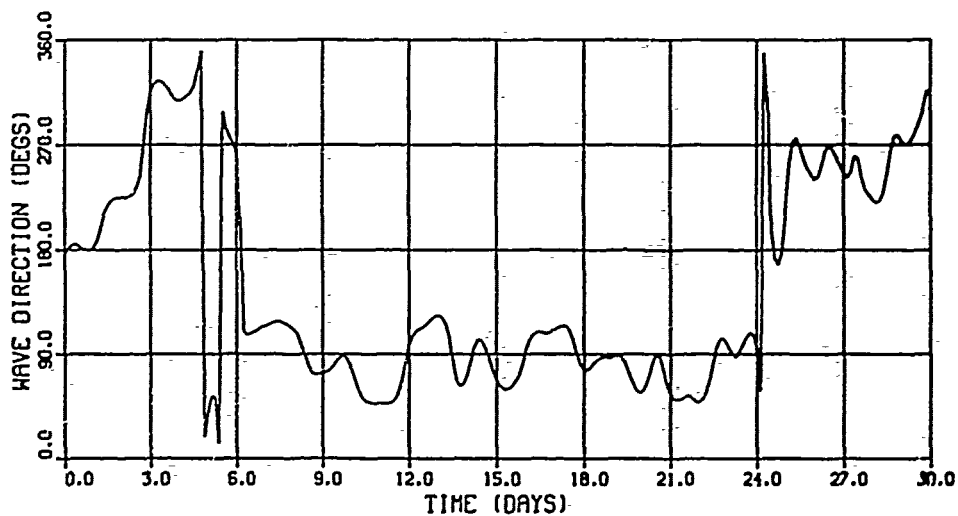
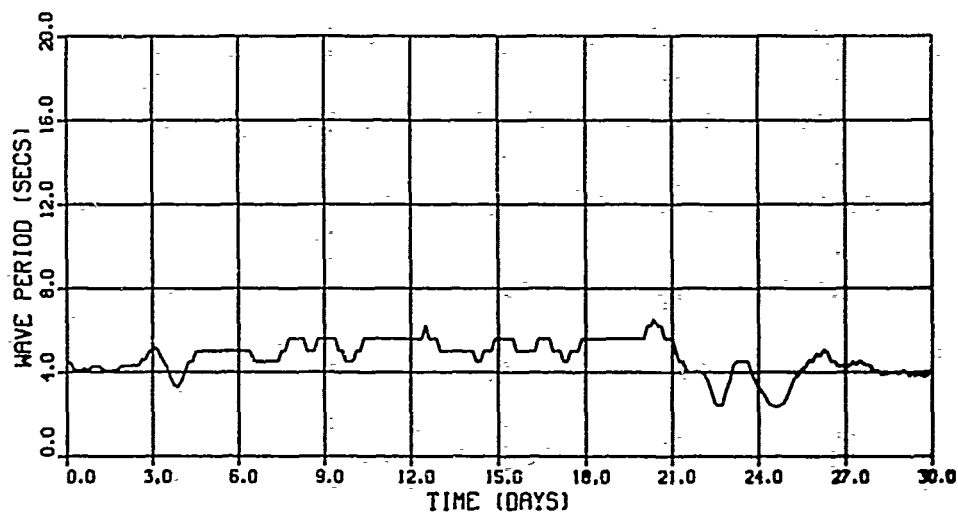
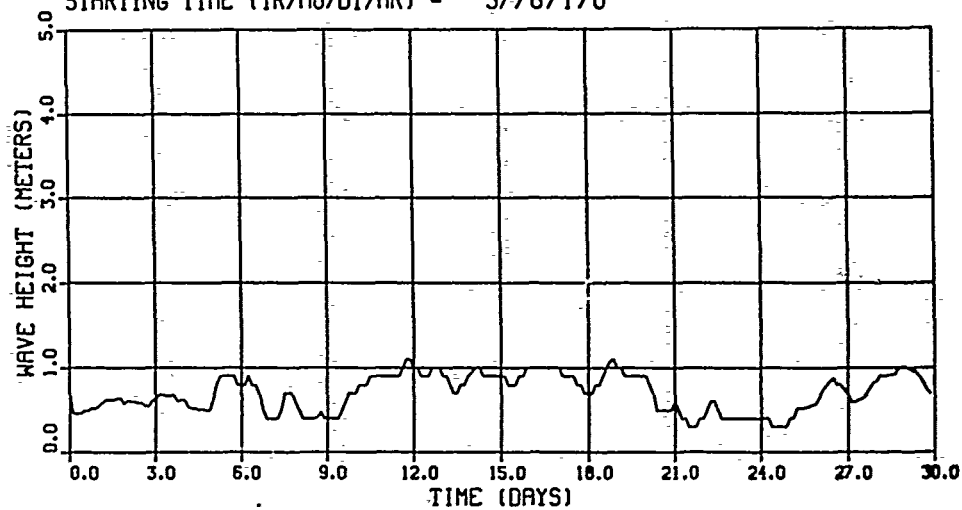
31 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) - 57/5/1/0



Simulated Wave Data for May 1957

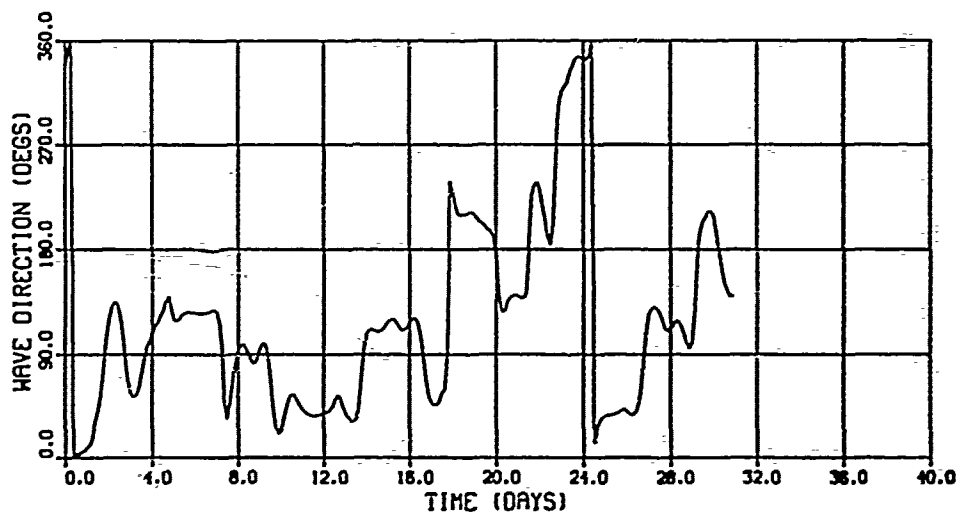
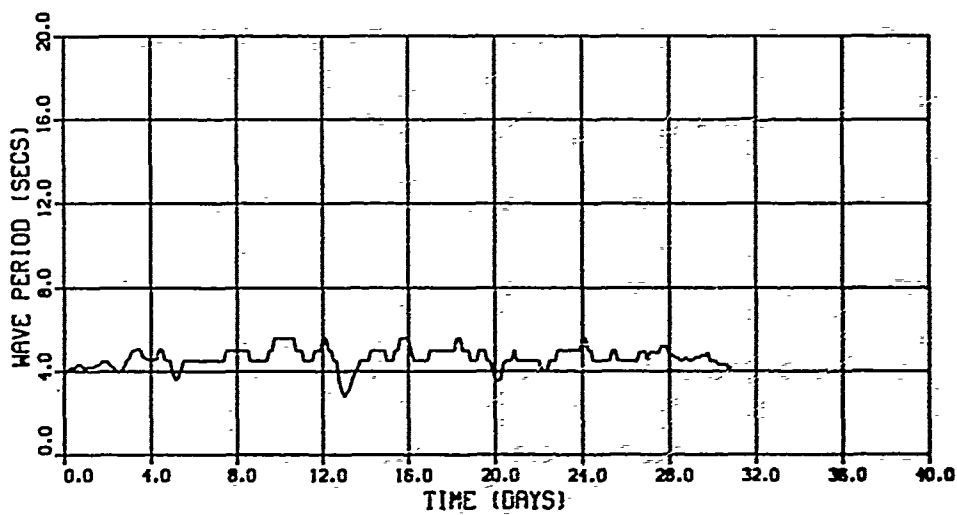
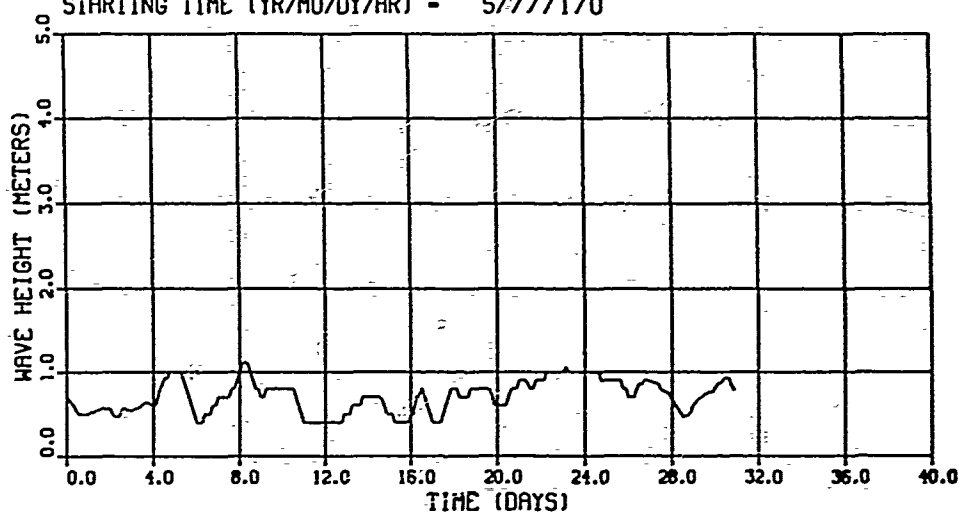
30 DAY SIMULATED HPD TIME SERIES

STARTING TIME (YR/MO/DY/HR) = 57/6/1/0



Simulated Wave Data for June 1957

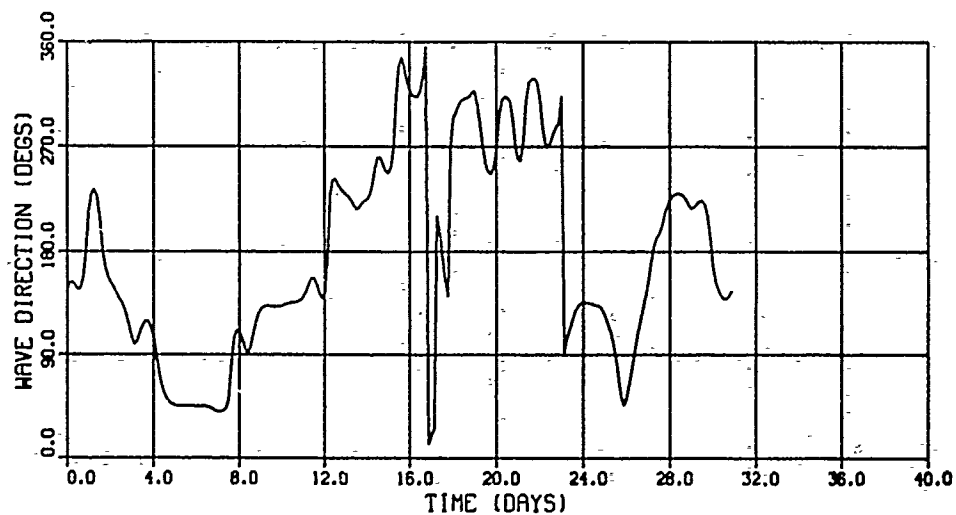
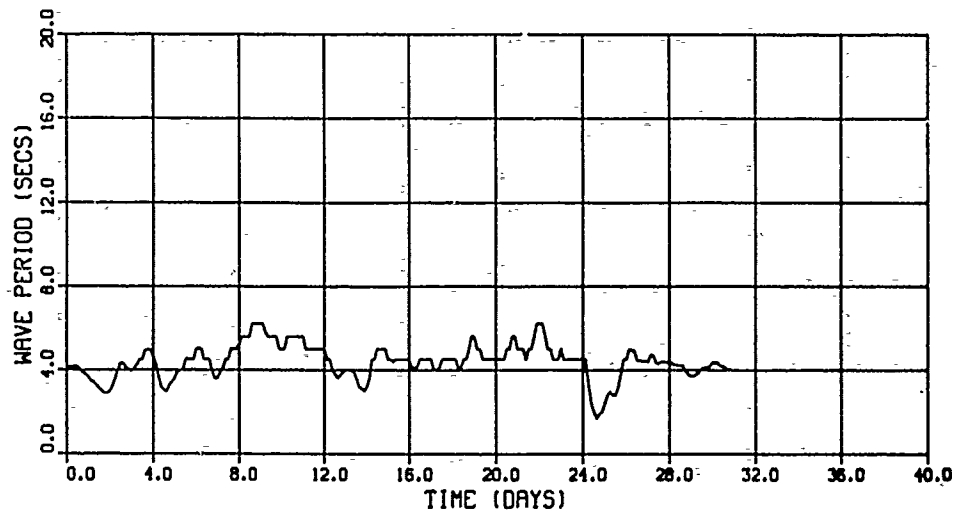
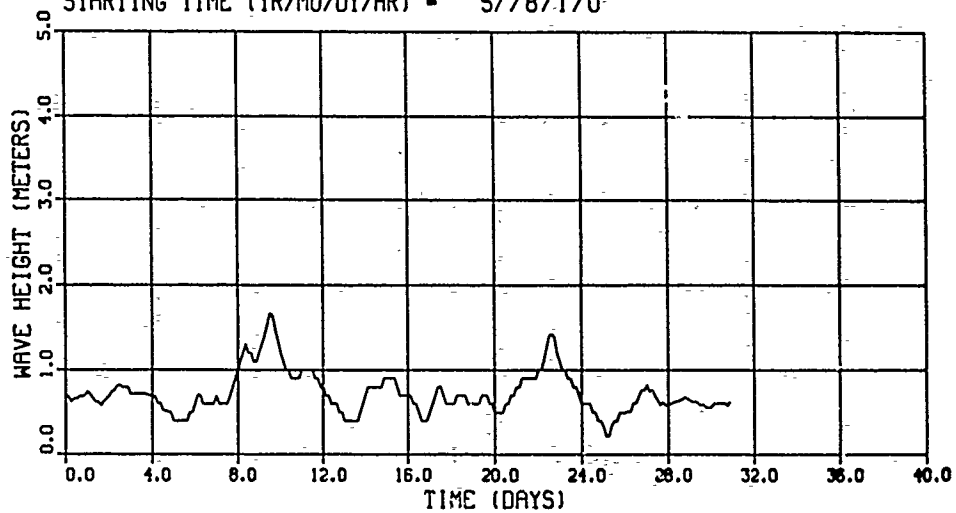
31 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) - 57/7/1/0



Simulated Wave Data for July 1957

31 DAY SIMULATED HPD TIME SERIES

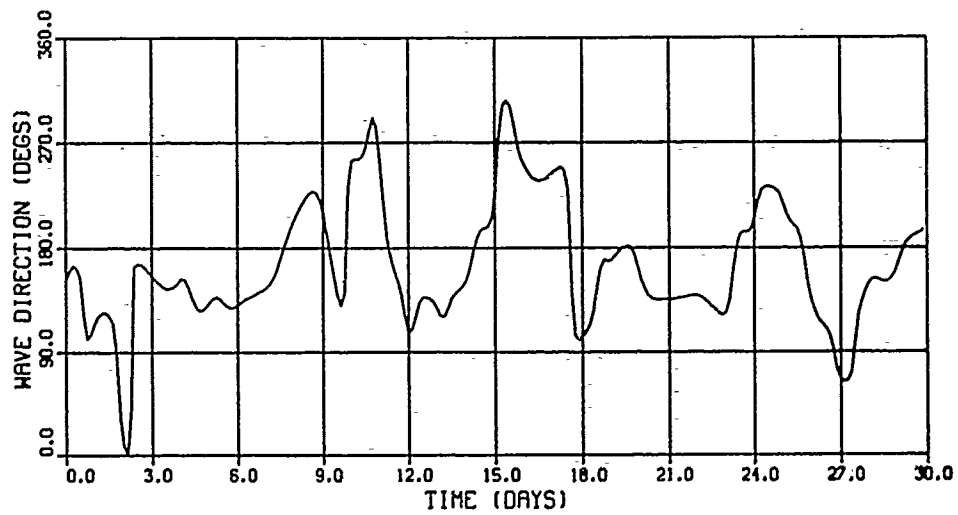
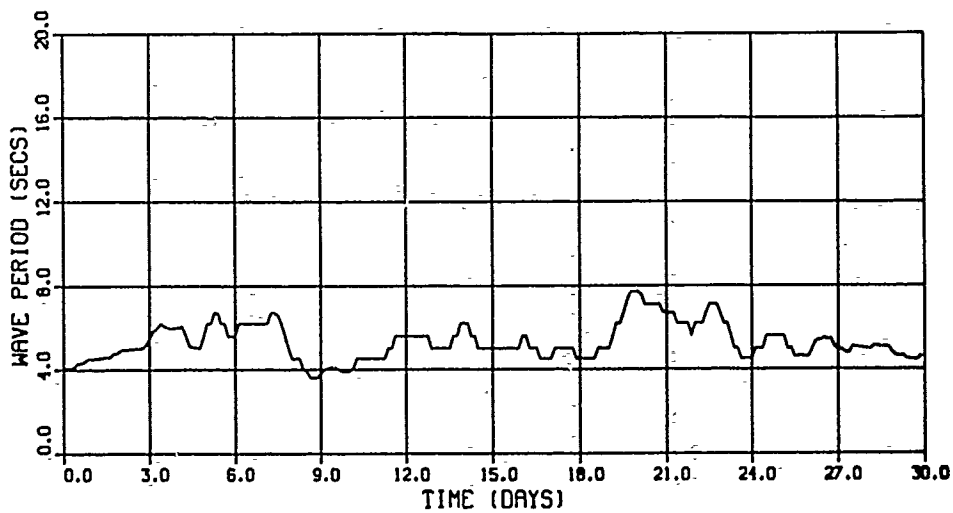
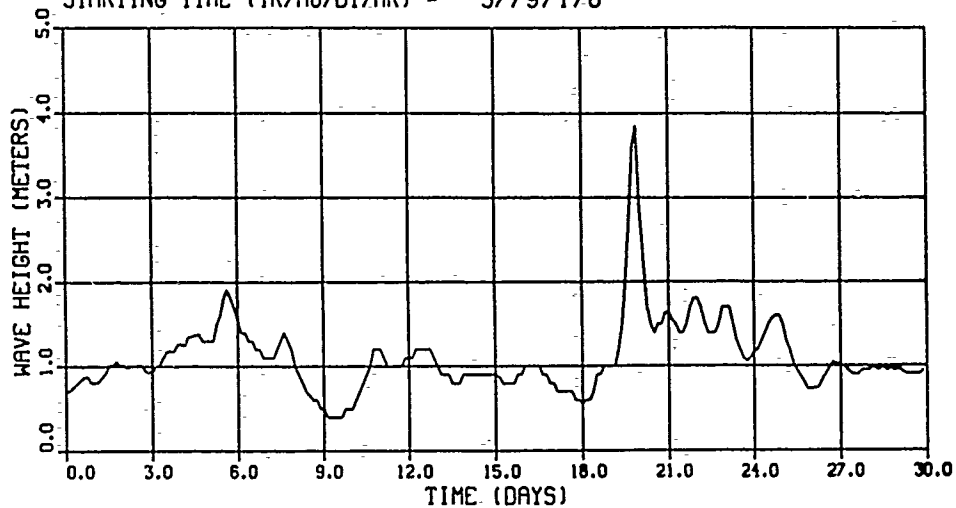
STARTING TIME (YR/MO/DY/HR) = 57/8/1/0



Simulated Wave Data for August 1957

30 DAY SIMULATED HPD TIME SERIES

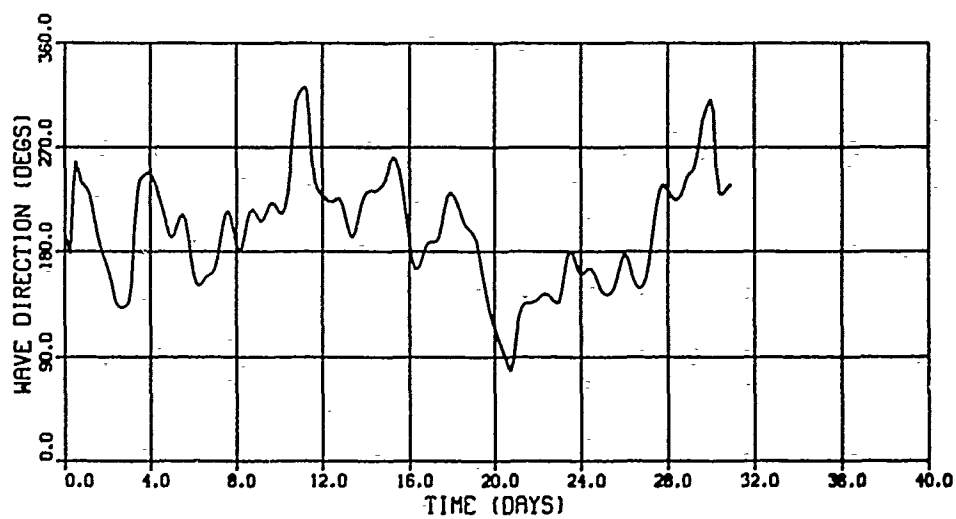
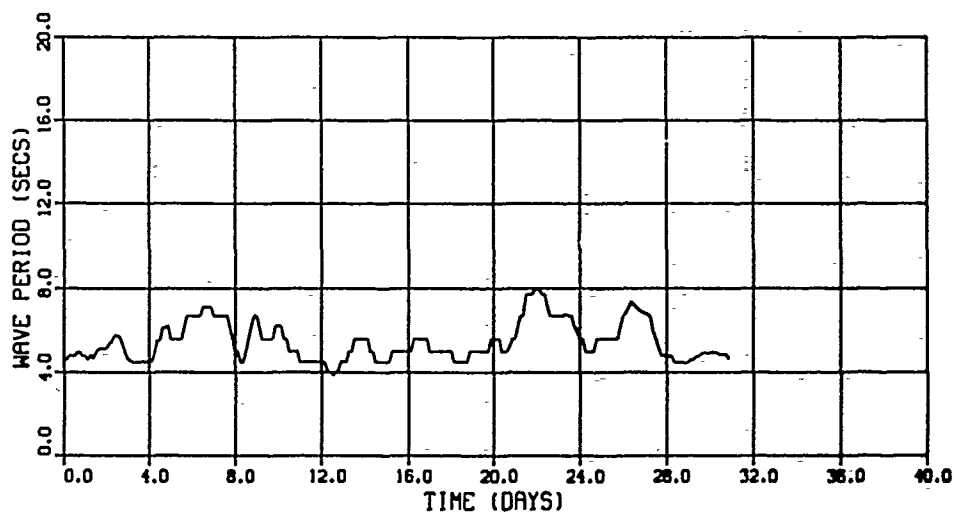
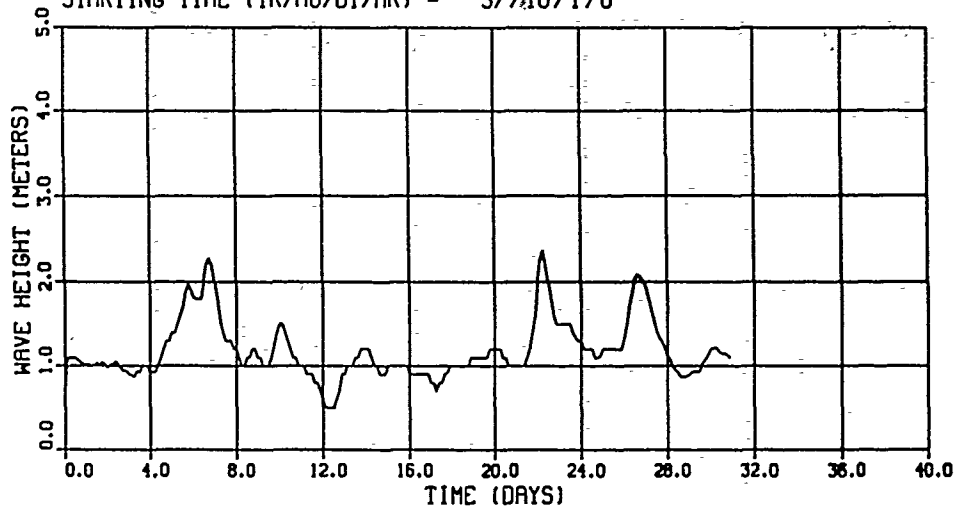
STARTING TIME (YR/MO/DY/HR) - 57/9/1/G



Simulated Wave Data for September 1957

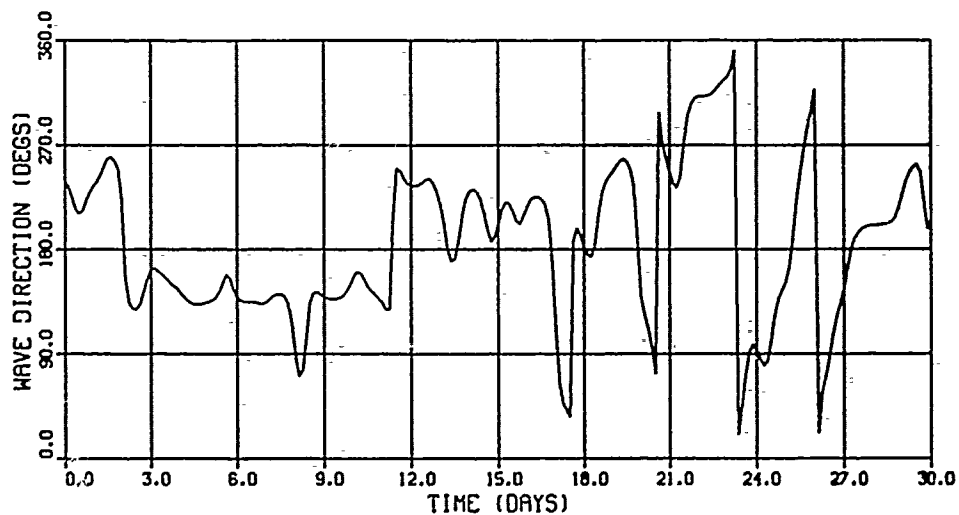
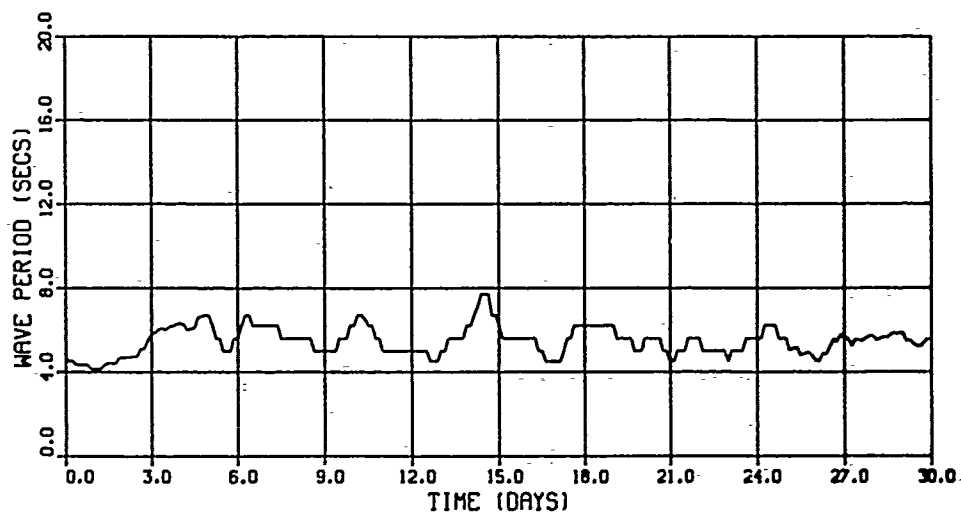
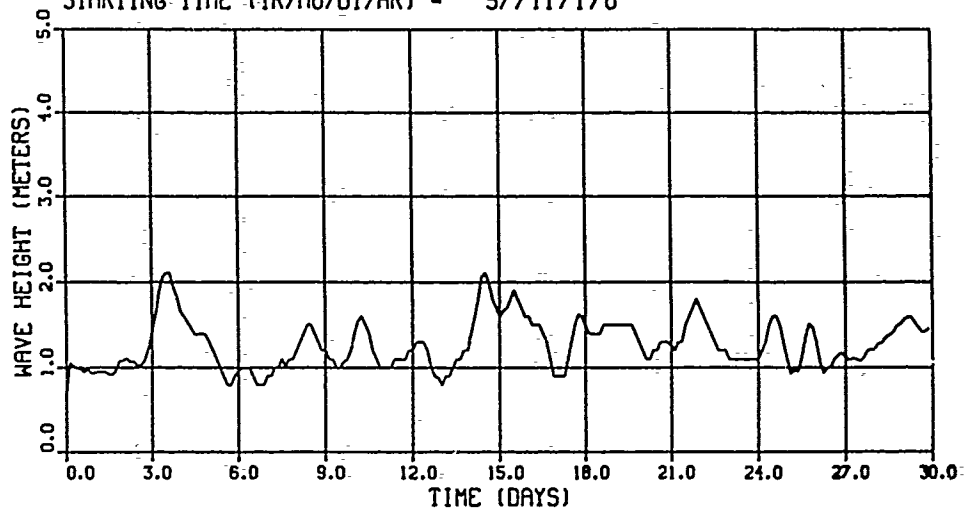
31 DAY SIMULATED HPD TIME SERIES

STARTING TIME (YR/MO/DY/HR) - 57/10/1/0



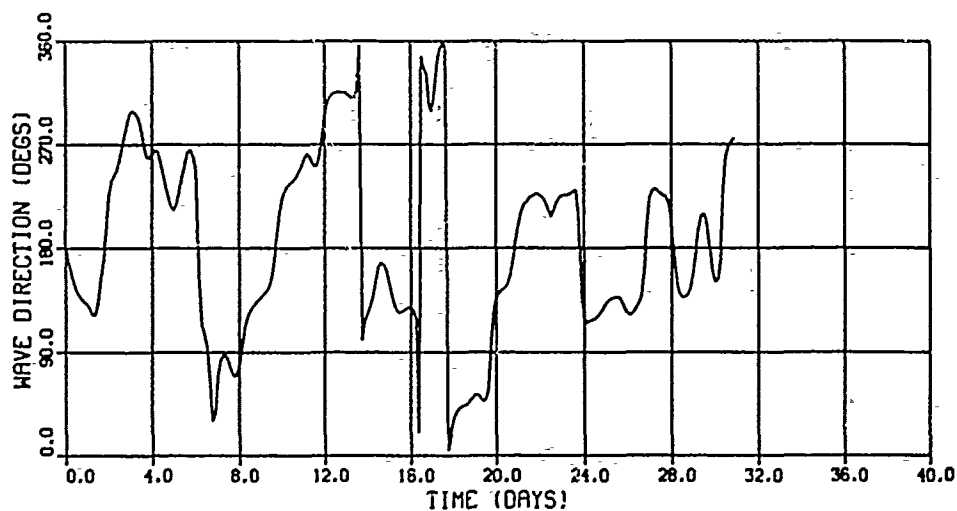
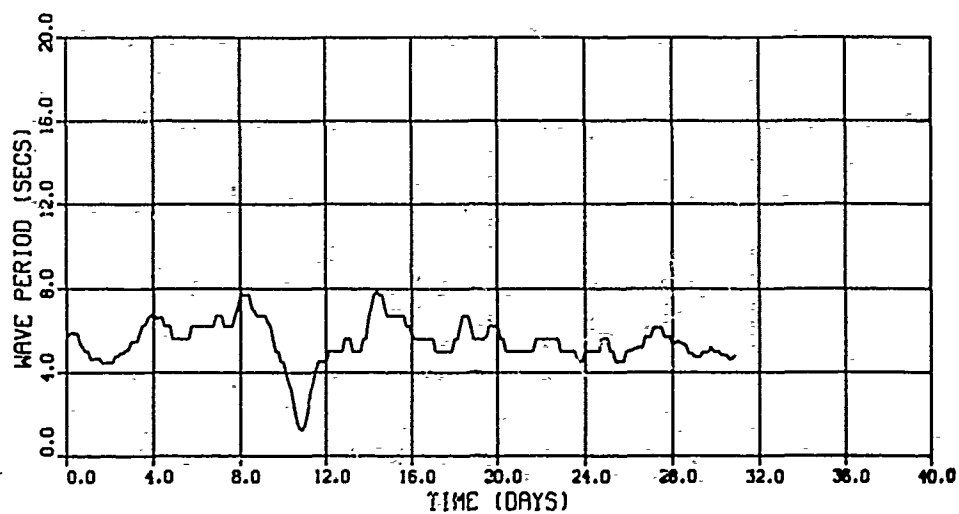
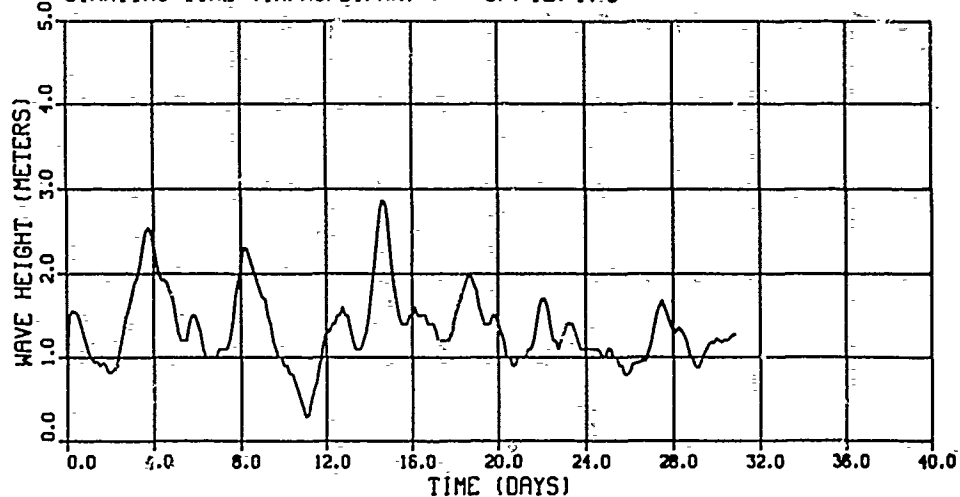
Simulated Wave Data for October 1957

30 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) = 57/11/1/0



Simulated Wave Data for November 1957

31 DAY SIMULATED HPD TIME SERIES
STARTING TIME (YR/MO/DY/HR) = 57/12/1/0

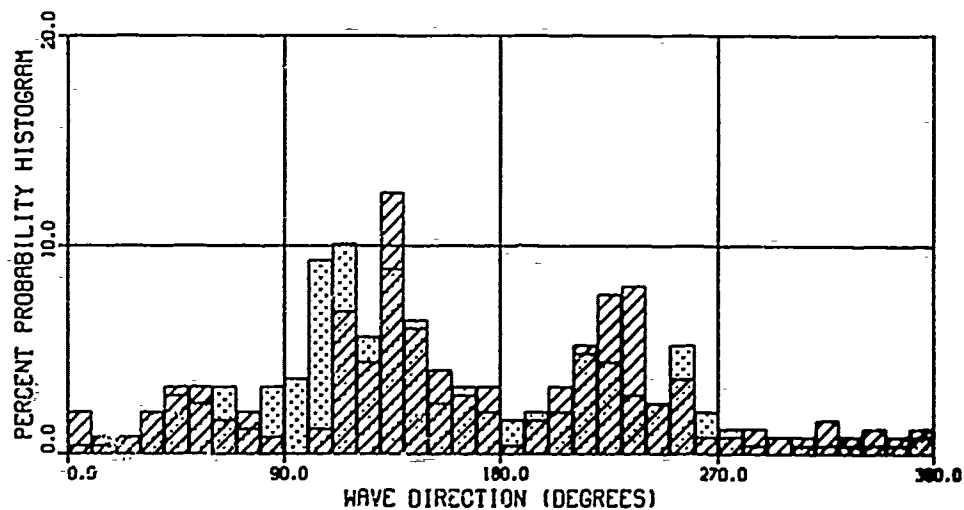
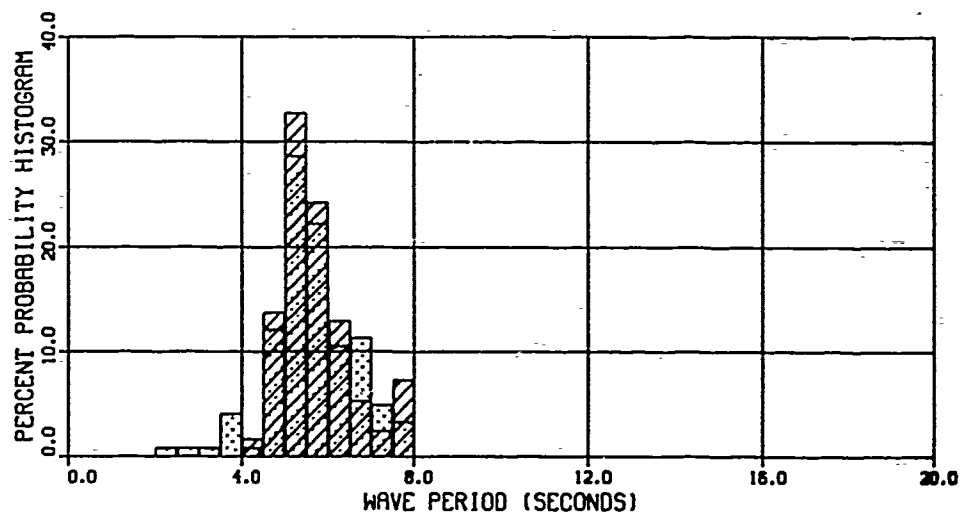
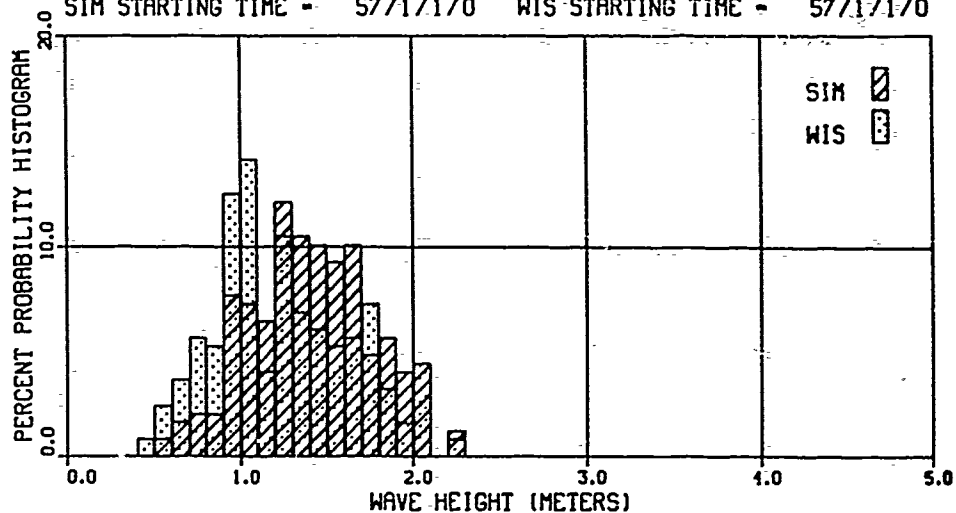


Simulated Wave Data for December 1957

APPENDIX I: COMPARISONS OF THE PROBABILITY DENSITIES FOR THE ACTUAL
WIS DATA IN APPENDIX G AND THE SIMULATIONS IN APPENDIX H

31 DAY SIMULATED VS 31 DAY WIS HPD TIME SERIES

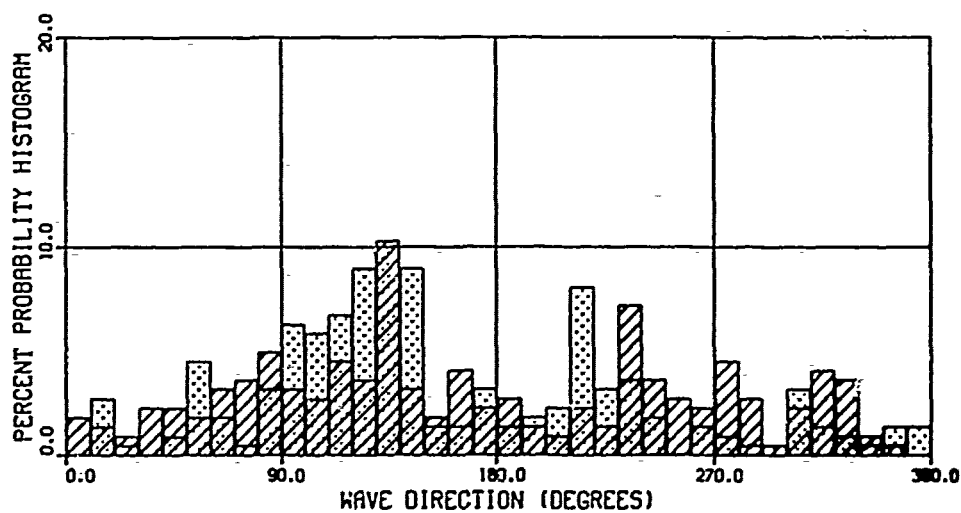
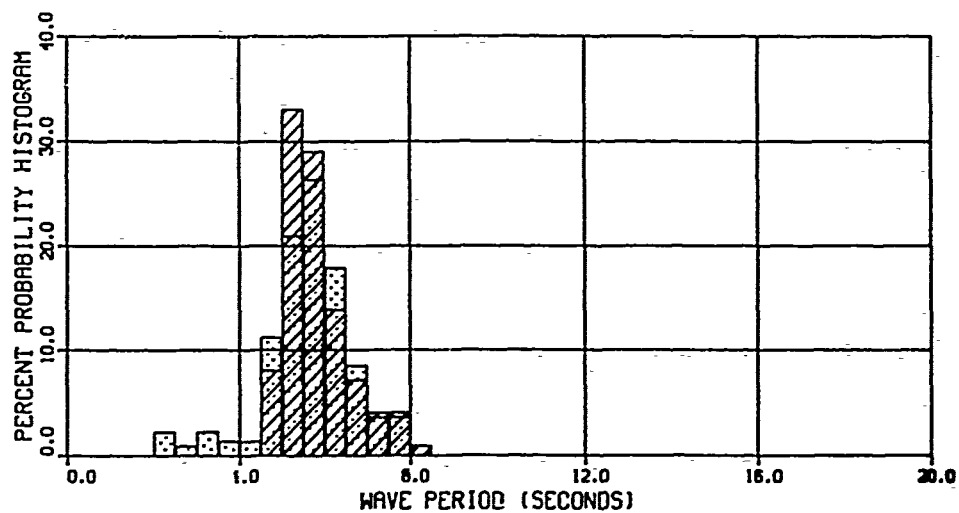
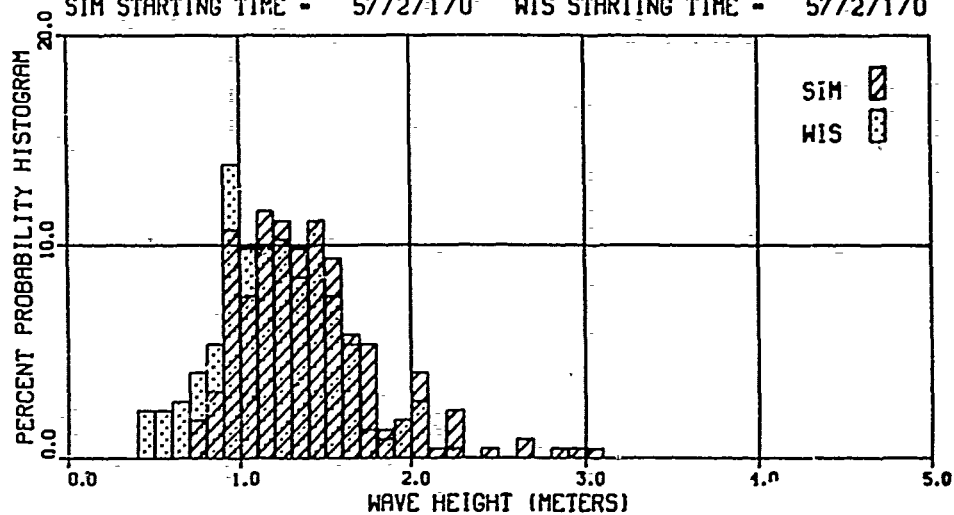
SIM STARTING TIME - 57/1/1/0 WIS STARTING TIME - 57/1/1/0



Probability Histogram Comparisons for January 1957

28 DAY SIMULATED VS 28 DAY HIS HPD TIME SERIES

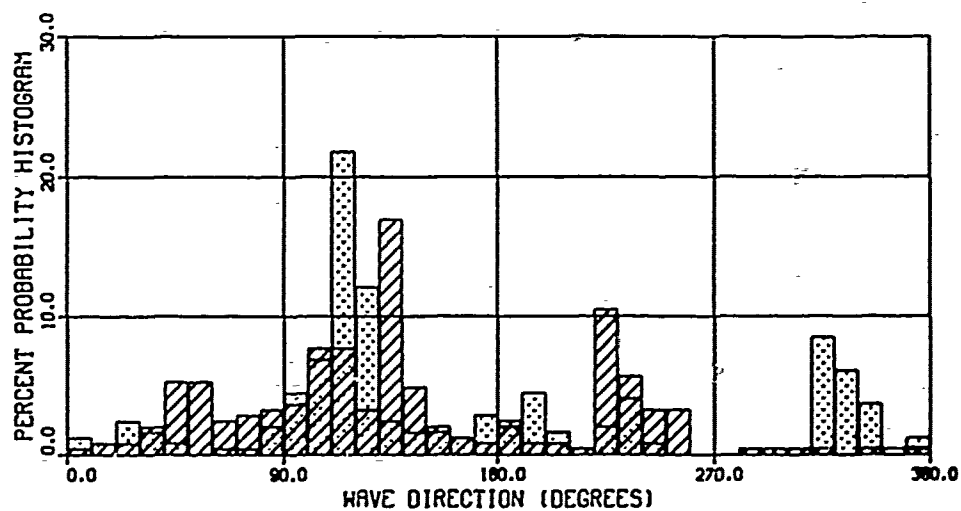
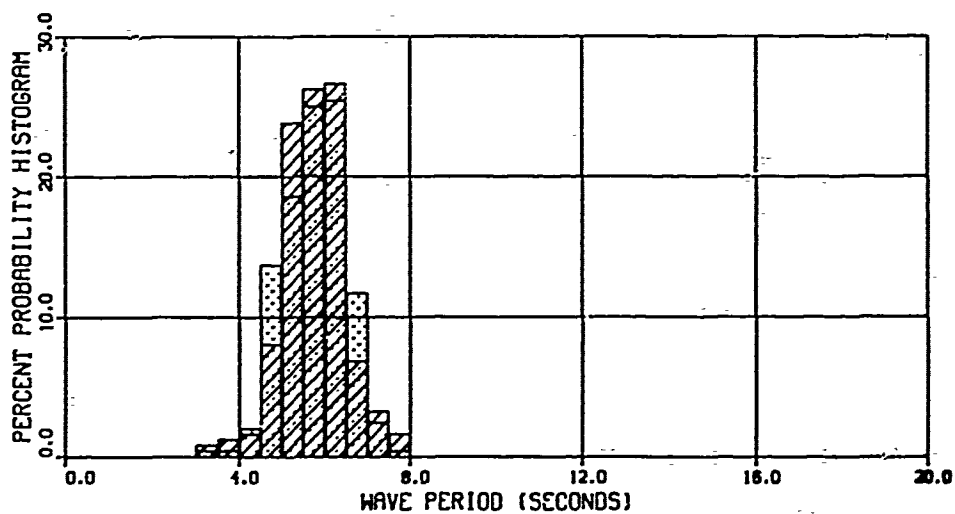
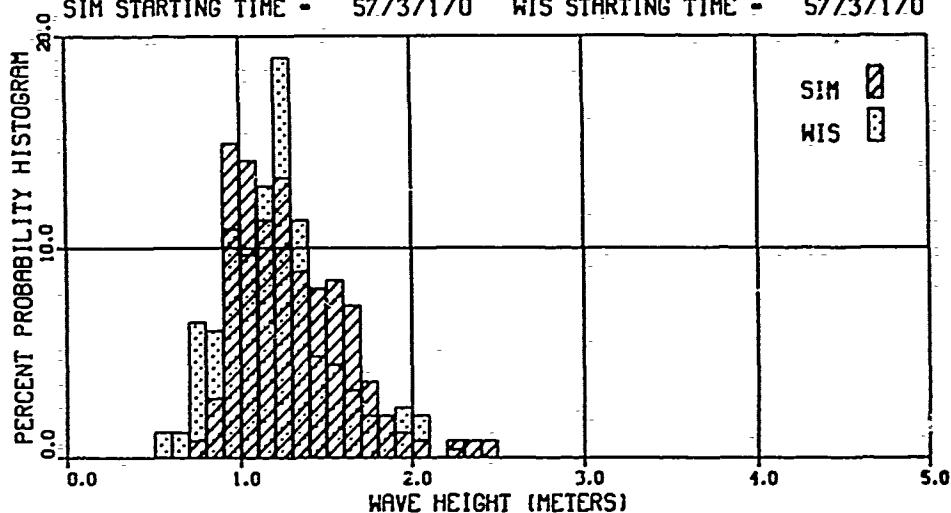
SIM STARTING TIME - 57/2/1/0 HIS STARTING TIME - 57/2/1/0



Probability Histogram Comparisons for February 1957

31 DAY SIMULATED VS 31 DAY WIS HPD TIME SERIES

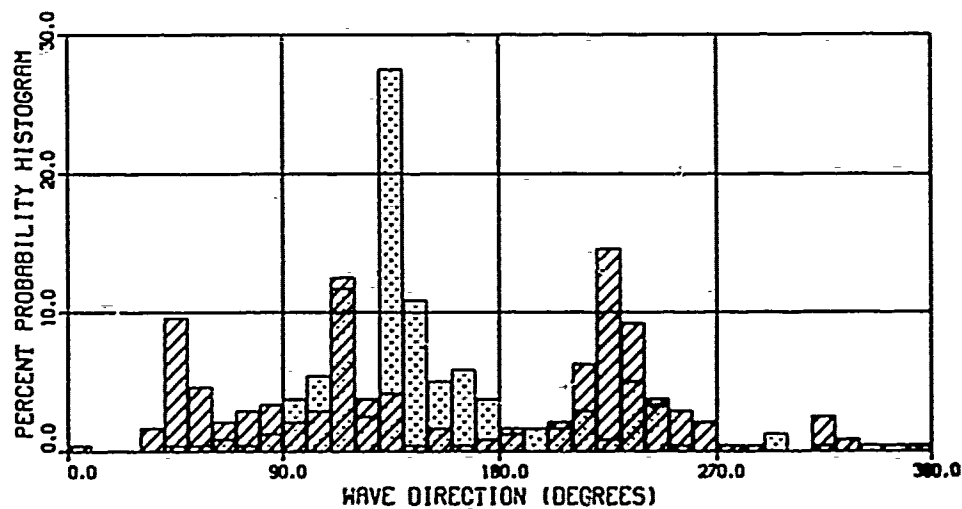
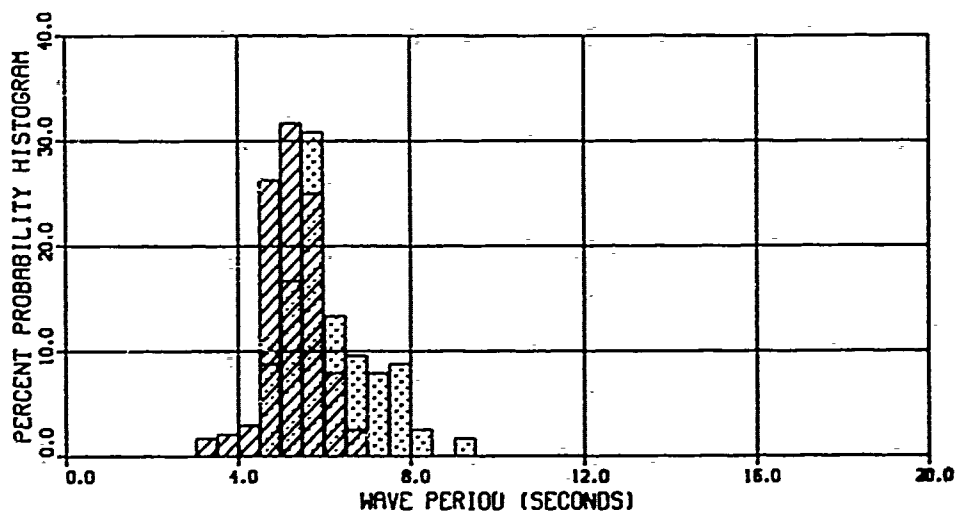
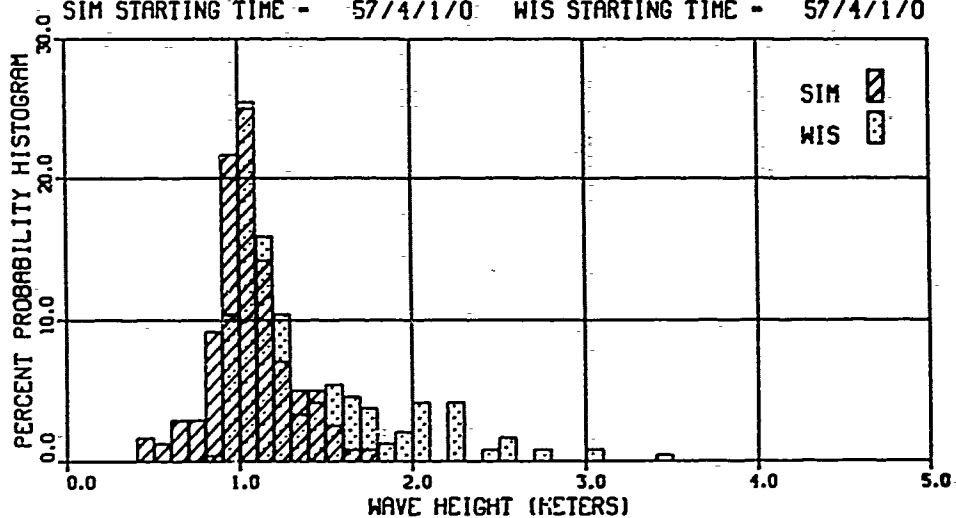
SIM STARTING TIME - 57/3/1/0 WIS STARTING TIME - 57/3/1/0



Probability Histogram Comparisons for March 1957

30 DAY SIMULATED VS 30 DAY HIS HPD TIME SERIES

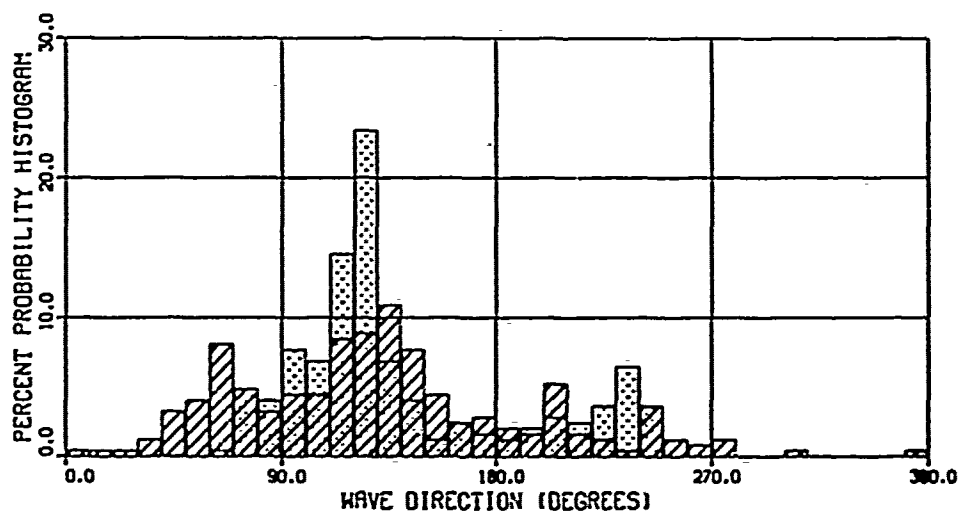
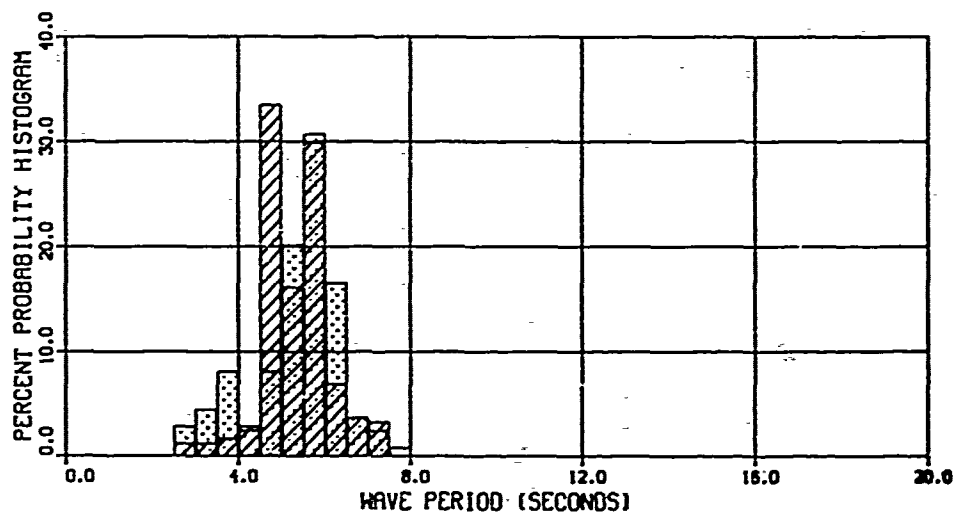
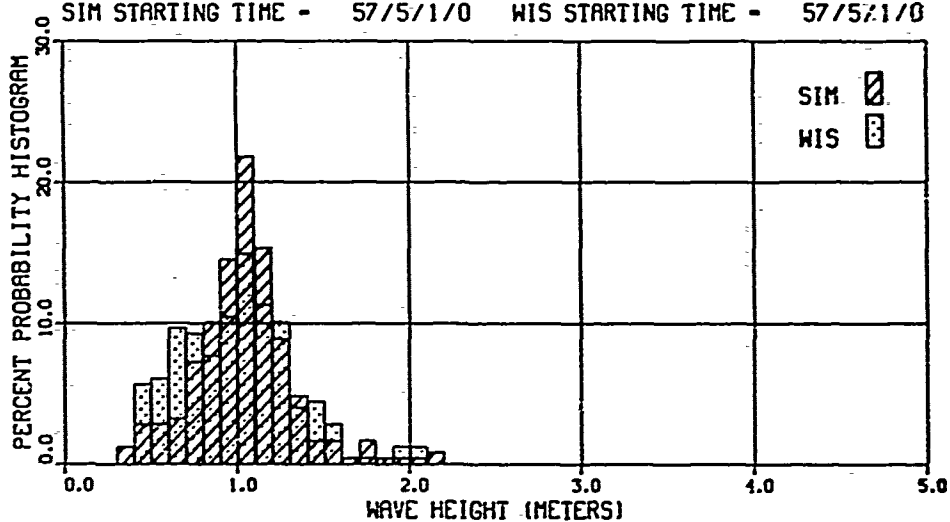
SIM STARTING TIME - 57/4/1/0 HIS STARTING TIME - 57/4/1/0



Probability Histogram Comparisons for April 1957

31 DAY SIMULATED VS 31 DAY HIS HPD TIME SERIES

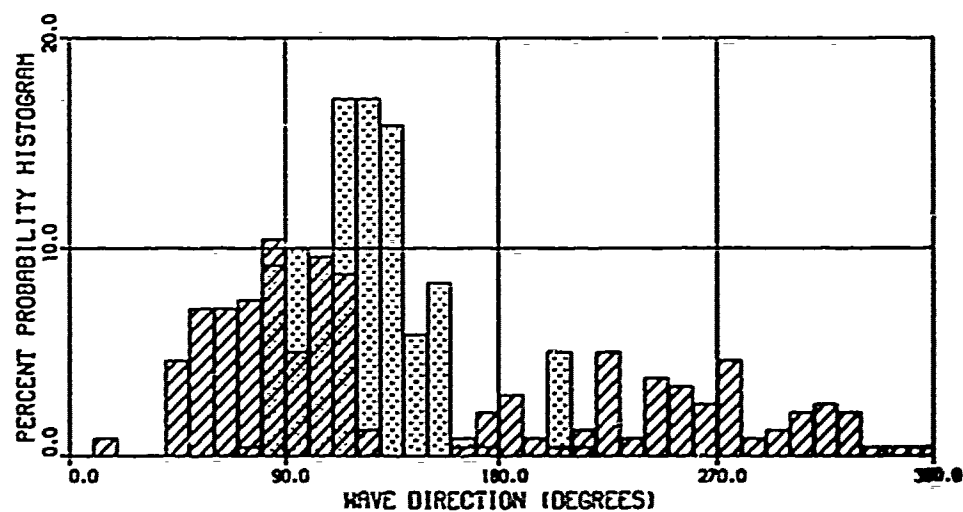
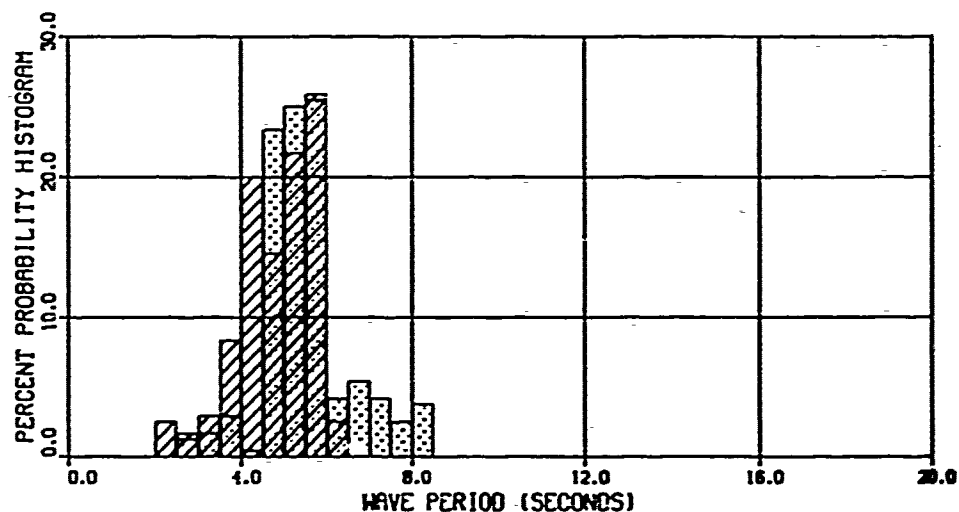
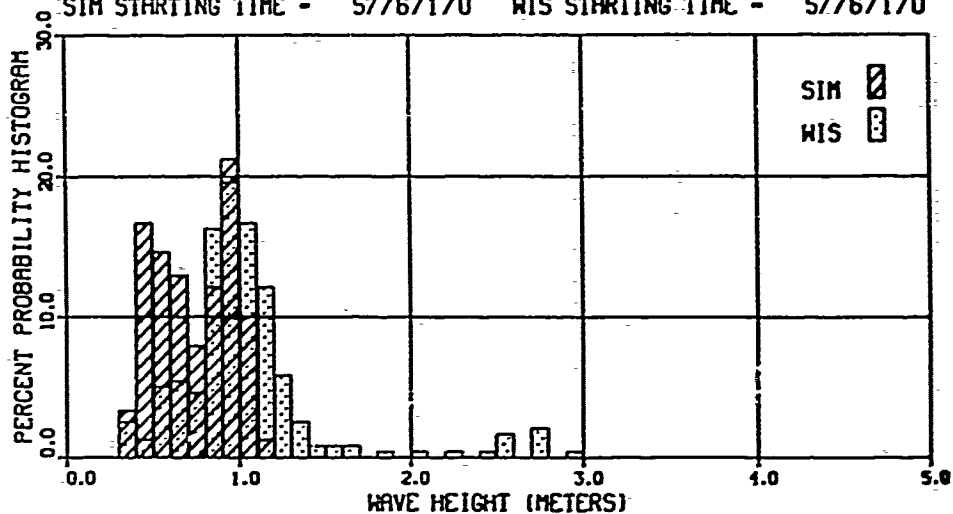
SIM STARTING TIME - 57/5/1/0 HIS STARTING TIME - 57/5/1/0



Probability Histogram Comparisons for May 1957

30 DAY SIMULATED VS 30 DAY HIS HPD TIME SERIES

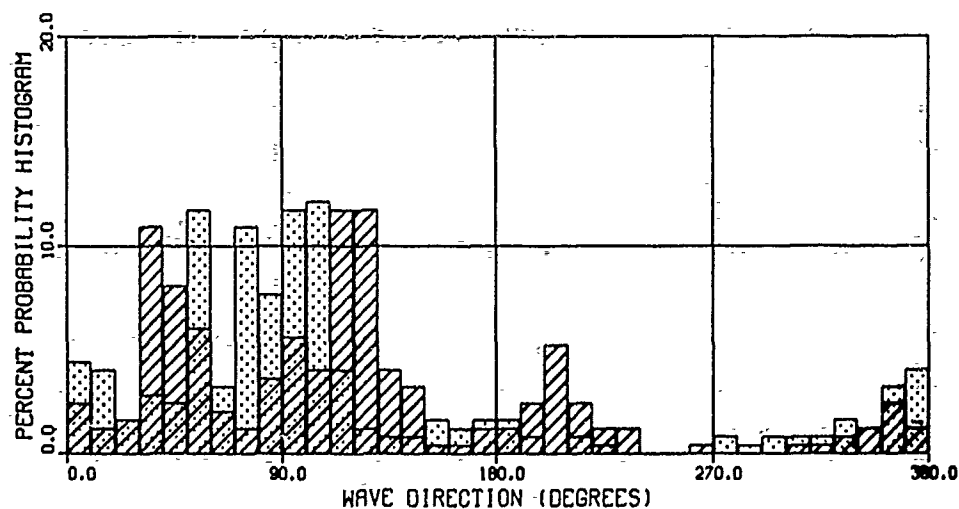
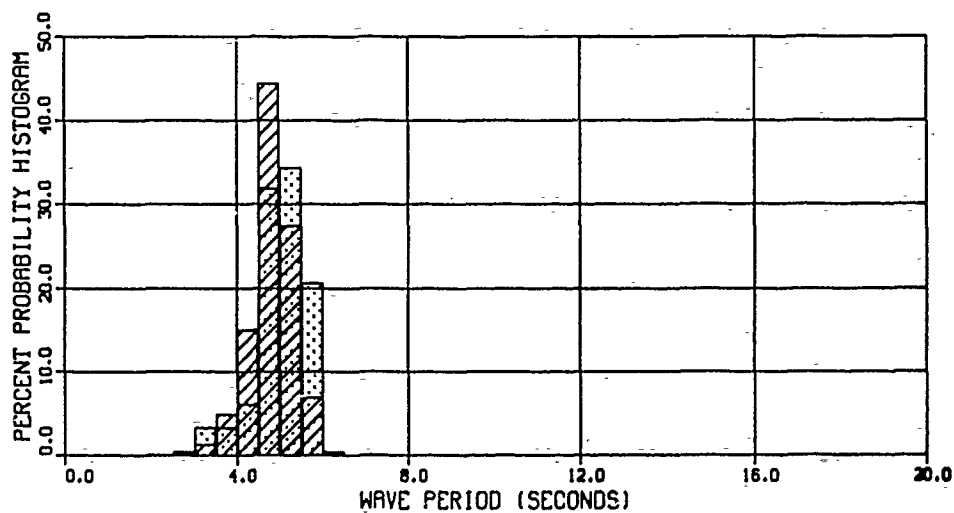
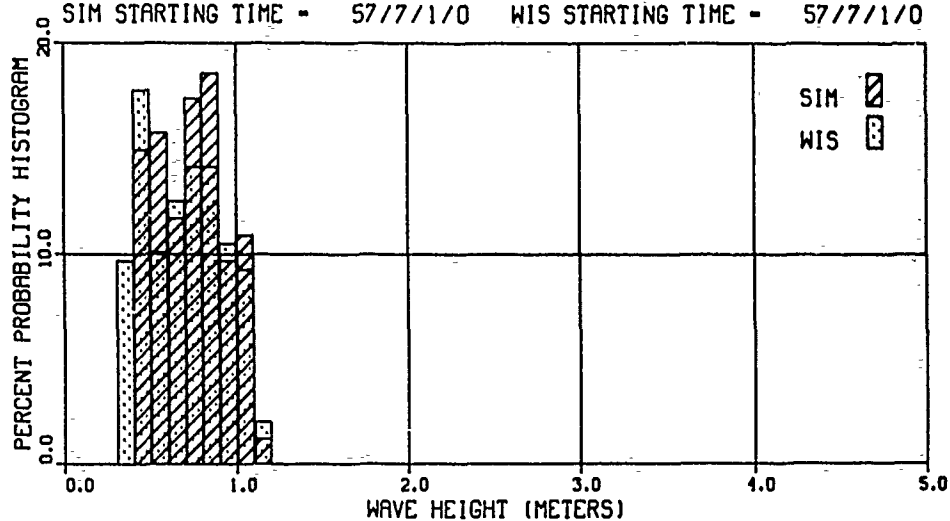
SIM STARTING TIME - 57/6/1/0 HIS STARTING TIME - 57/6/1/0



Probability Histogram Comparisons for June 1957

31 DAY SIMULATED VS 31 DAY WIS HPD TIME SERIES

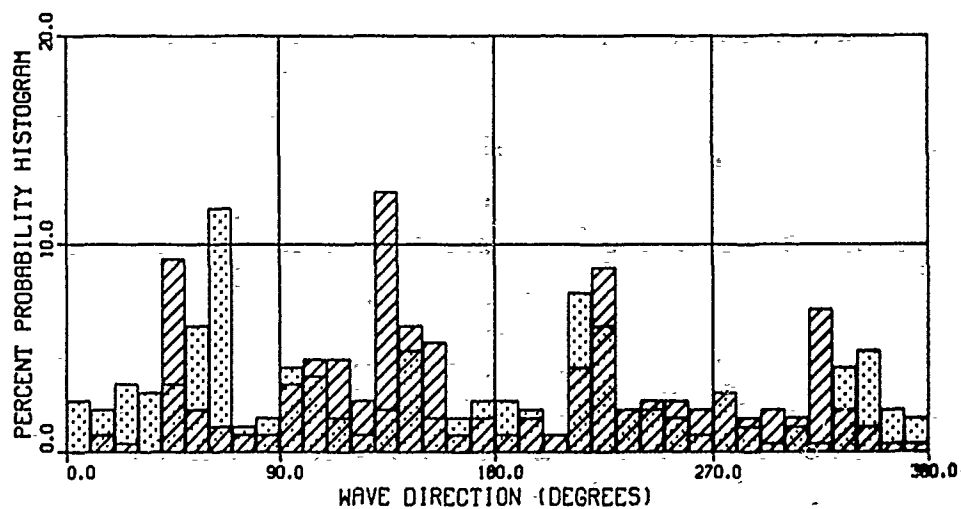
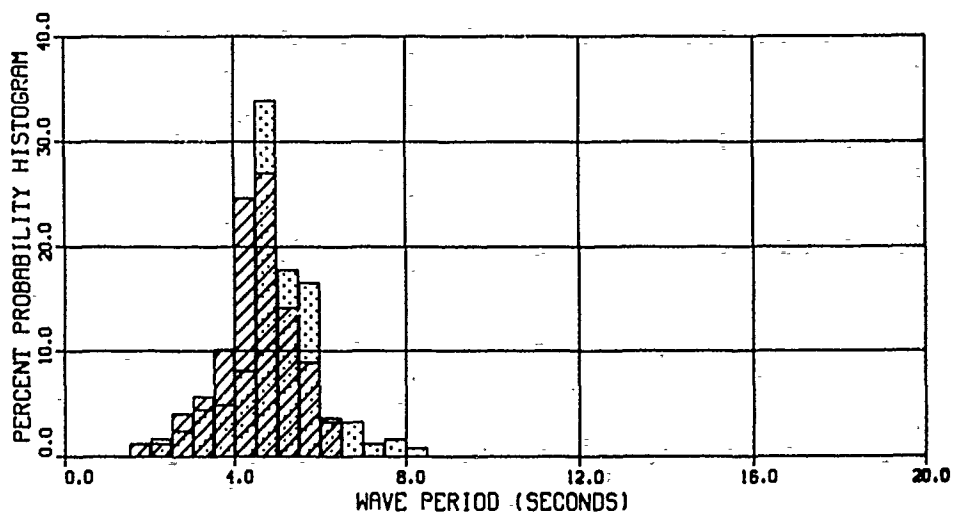
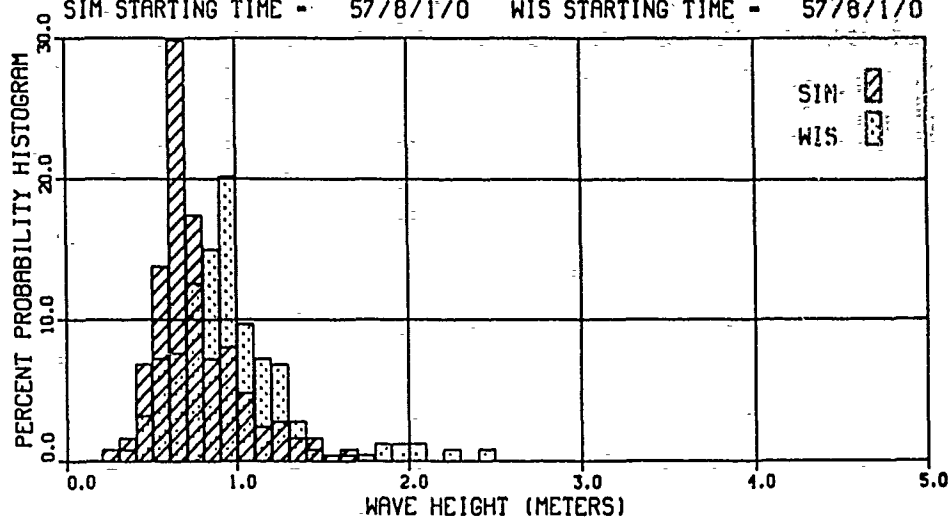
SIM STARTING TIME - 57/7/1/0 WIS STARTING TIME - 57/7/1/0



Probability Histogram Comparisons for July 1957

31 DAY SIMULATED VS 31 DAY WIS HPD TIME SERIES

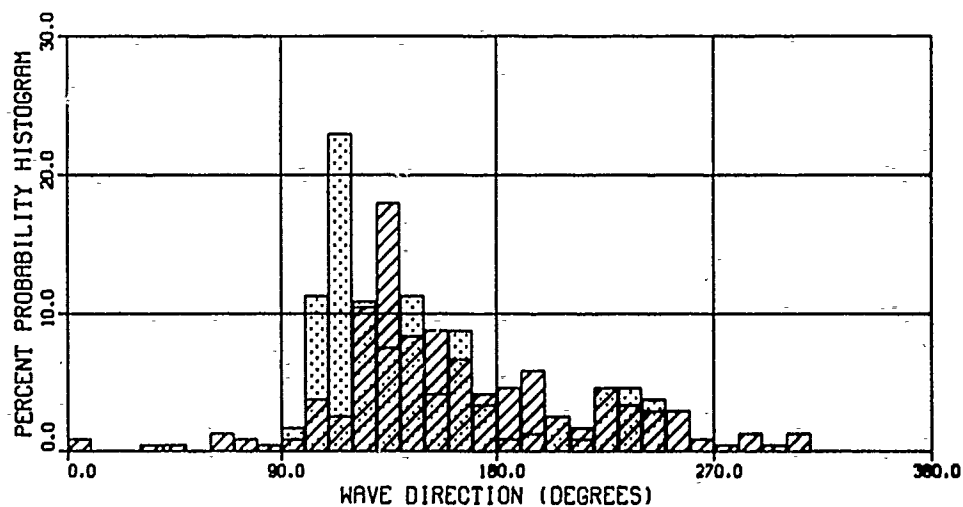
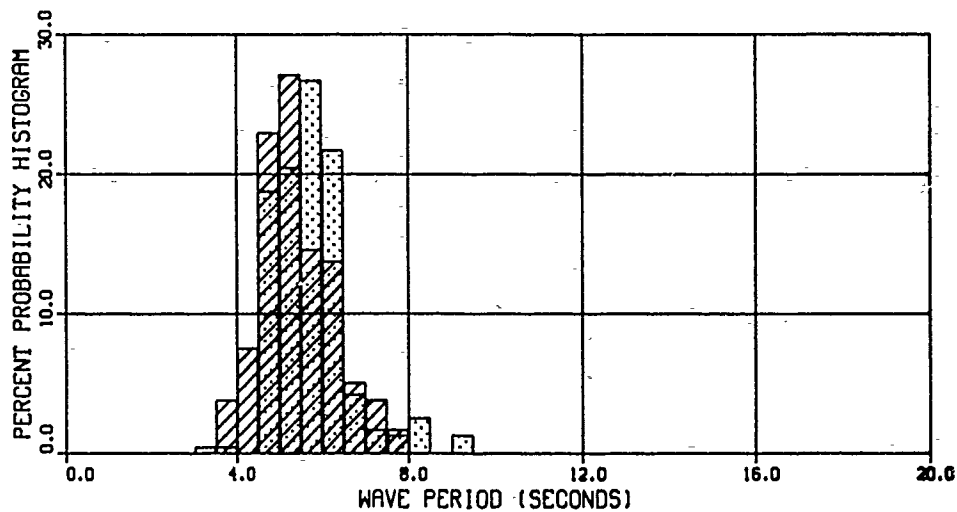
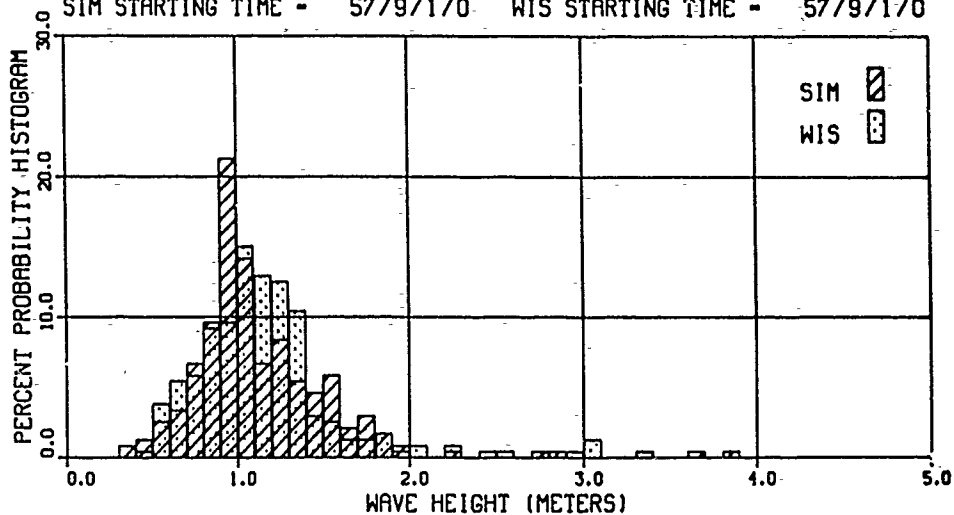
SIM STARTING TIME - 57/8/1/0 WIS STARTING TIME - 57/8/1/0



Probability Histogram Comparisons for August 1957

30 DAY SIMULATED VS 30 DAY WIS HPD TIME SERIES

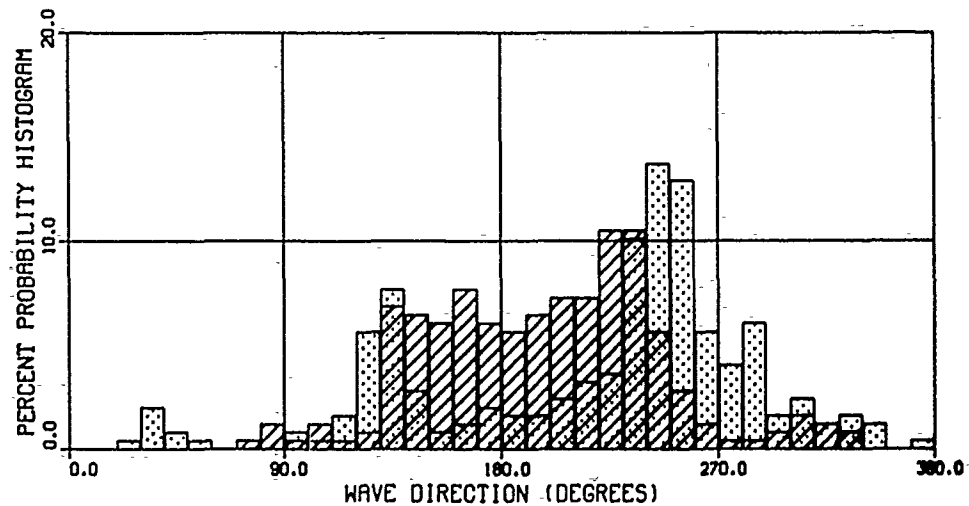
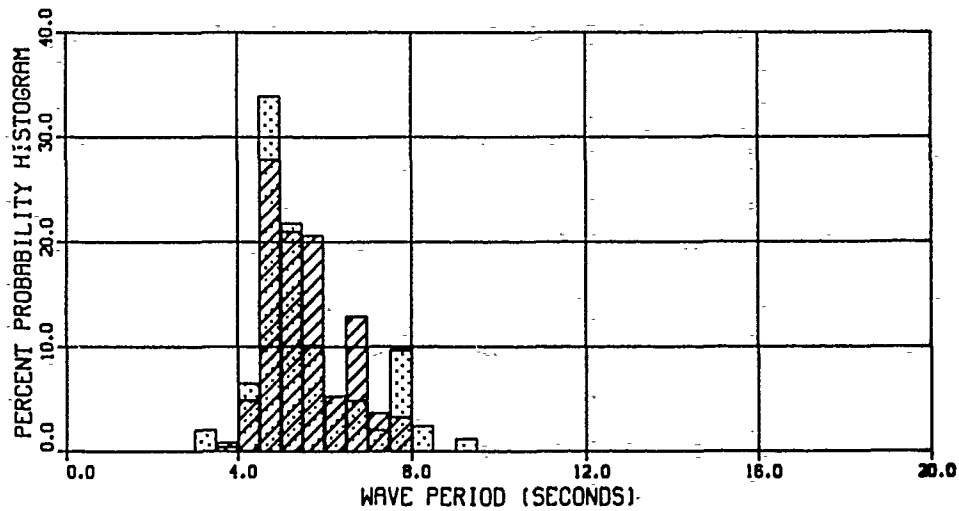
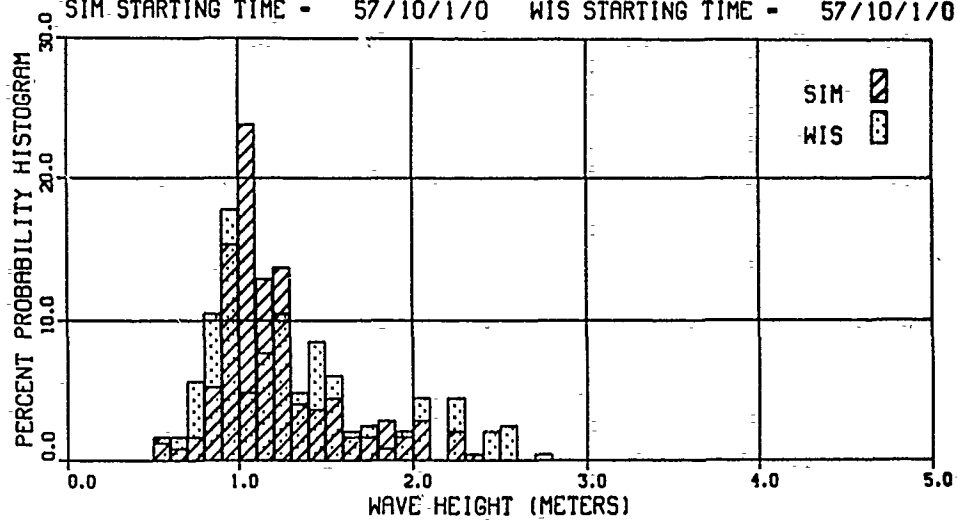
SIM STARTING TIME - 57/9/1/0 WIS STARTING TIME - 57/9/1/0



Probability Histogram Comparisons for September 1957

31 DAY SIMULATED VS 31 DAY WIS HPD TIME SERIES

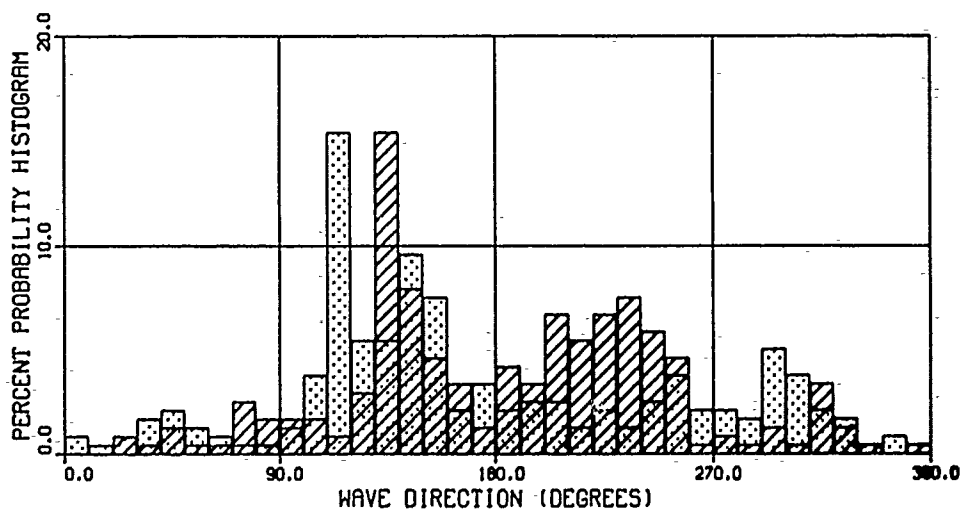
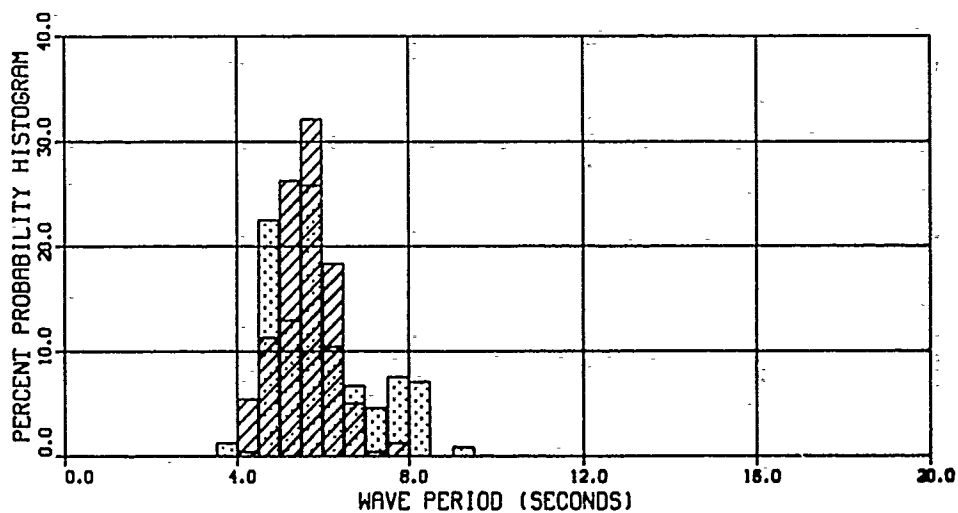
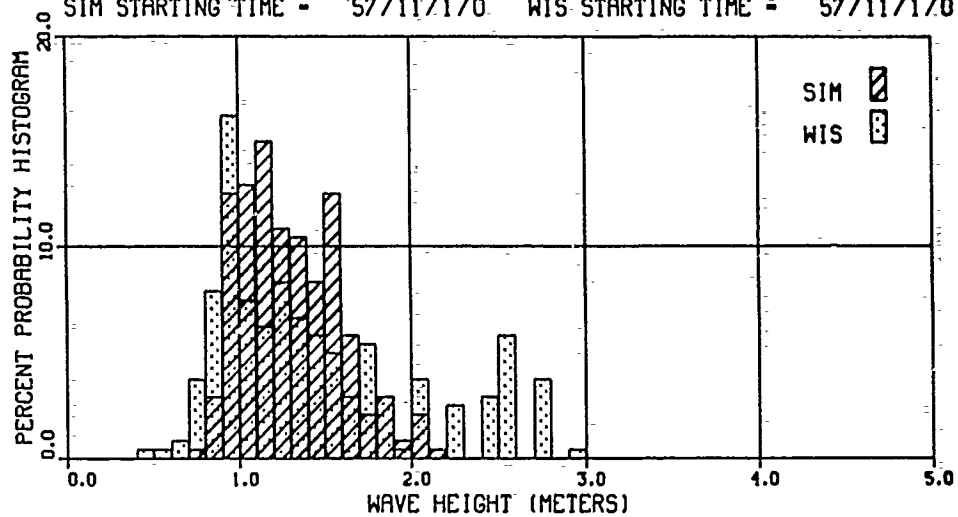
SIM STARTING TIME - 57/10/1/0 WIS STARTING TIME - 57/10/1/0



Probability Histogram Comparisons for October 1957

30 DAY SIMULATED VS 30 DAY WIS HPD TIME SERIES

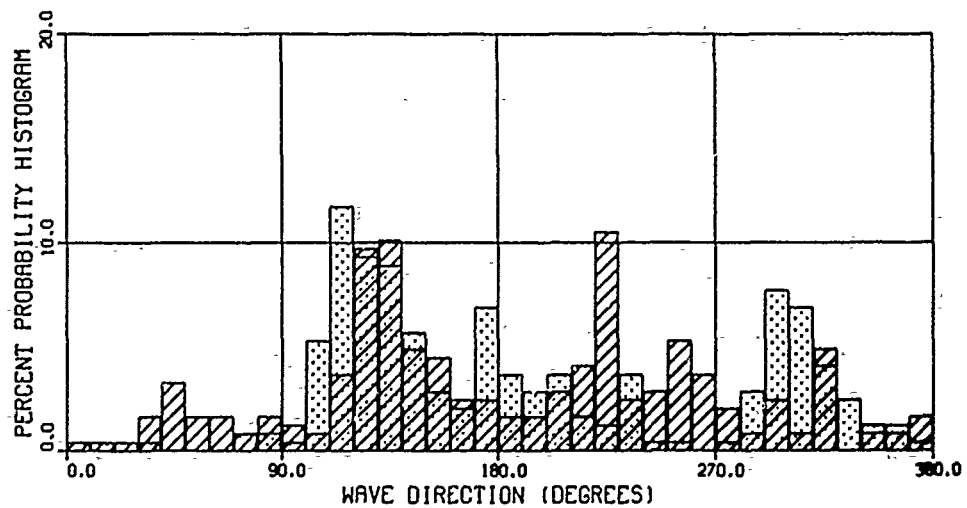
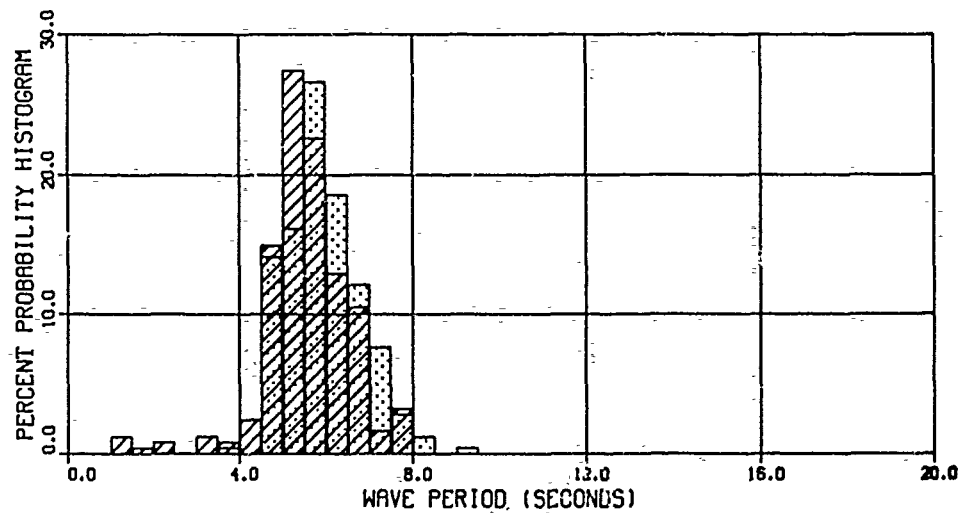
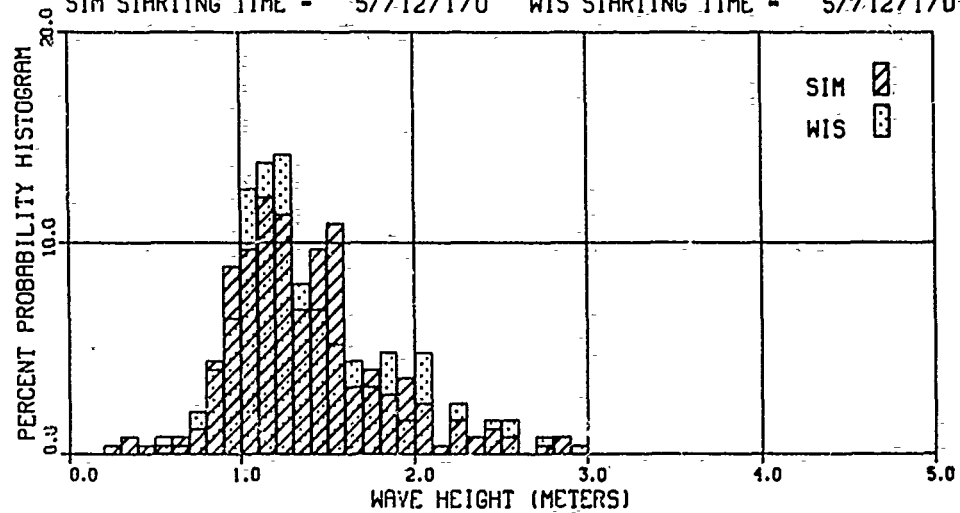
SIM STARTING TIME - 57/11/1/0 WIS STARTING TIME - 57/11/1/0



Probability Histogram Comparisons for November 1957

31 DAY SIMULATED VS 31 DAY WIS HPD TIME SERIES

SIM STARTING TIME - 57/12/1/0 WIS STARTING TIME - 57/12/1/0



Probability Histogram Comparisons for December 1957