Naval Research Laboratory

Washington, DC 20375-5000 NRL Memorandum Report 5848 September 30, 1986

AD-A173 486

KELSEA
An Interactive Computer Code for Kelvin and Random Ambient Sea Waves

G.A. KERAMIDAS, H.T. WANG, AND J.A. BURKE

Fluid Dynamics Branch
Marine Technology Division

W. BAUMAN

Naval Sea Systems Command
Washington, DC 20362

AD-A173 486

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 5848	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory 6b OFFICE SYMBOL (If applicable) Code 5841	7a NAME OF MONITORING ORGANIZATION Office of Naval Research
6c ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000	7b ADDRESS (City, State, and ZIP Code) Arlington, VA 22207
8a NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code) Arlington, VA 22217	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61153N PROJECT NO TASK NO RR023-01-41 WORK UNIT ACCESSION NO. DN280-006

11 TITLE (Include Security Classification)
KELSEA
An Interactive Computer Code for Kelvin and Random Ambient Sea Waves

12 PERSONAL AUTHOR(S)
Keramidas, G.A., Wang, H.T., Burke, J.A., and Bauman,* W.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Interim	FROM ____ TO ____	1986 September 30	41

16 SUPPLEMENTARY NOTATION
Naval Sea Systems Command, Washington, DC 20362

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Hydrodynamics; Kelvin waves; Ambient sea waves; Computer code

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report covers the theoretical and numerical aspects of random ambient sea wave realizations, and the computer code for generating such waves. The mathematical model for random ambient sea waves is discussed in the first part of this report together with the superposition of these waves with Kelvin waves. The second part of the report covers the development and structure of the computer program. A detailed explanation is given for the input/output requirements of the computer program. Numerical results are given in the last part of this report and they cover a number of cases. The results are plotted as three dimensional surfaces for a realistic view of the generated waves.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL George A. Keramidas	22b TELEPHONE (Include Area Code) (202) 767-3389 22c OFFICE SYMBOL Code 5841

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

*U.S. Government Printing Office: 1985-507-047

CONTENTS

KELSEA
AN INTERACTIVE COMPUTER CODE
FOR
KELVIN AND RANDOM AMBIENT SEA WAVES

I. INTRODUCTION

In many ocean engineering applications it is of interest to generate a spatial realization of a given random sea state from an experimentally derived or analytical wave spectrum. Such two dimensional spatial realizations can be used to model the wave elevations around a large offshore structure or to model the interaction of these waves with ocean currents and/or ship waves.

There is a reasonably large number of studies in the literature on the general approach for generating spatial realizations of random ambient sea waves as a double sum of sinusoidal components from a given energy spectrum. The general approach is relatively simple and it is briefly discussed in the following section of this paper. Detailed descriptions of the general approach can be found in [1]-[3]. Although the basic approach has been well documented, its manner of application has varied widely. For example, the number and assigned location of the components required to obtain a random ambient sea state are not always clearly stated and a trial and error process is often used to obtain the wave elevation signal.

The present paper addresses the specific ways of implementing the generalized approach and theory for generating a random ambient sea, based on the approach given in [3]. It has been shown in this reference that relatively few components are required to obtain sea states which visually and statistically approximate those for a random sea.

The Kelvin waves generated by a ship traveling at constant forward speed are obtained by using the complex elevation spectra, which can be calculated by the computer program KELVIN [4]. The particular method used to calculate the Kelvin wave elevations over a rectangular grid is described in some detail and the linear superposition of the Kelvin and ocean waves is also discussed.

The mathematical model has been implemented in a computer program which is described in the last section. Examples of selected cases of random ambient sea states are given as representative examples in the same section. Furthermore, examples are also given for the superposition of Kelvin waves with ambient sea waves.

II. MATHEMATICAL MODEL

The development of random sea state realizations is based on expressions which involve a triple sum over the wave number k, frequency ω, and direction θ. If one neglects the nonlinear interactions between wave components, the following expression for the wave elevation $\eta(x, z)$ can be derived from the disperson relation between wave number and frequency

$$\eta(x, z) = \sum_{i=1}^{N}\sum_{j=1}^{M} A_{ij} \cos[k_i(x \cos\theta_j + z \sin\theta_j) + \phi_{ij}] \tag{1}$$

where x and z are respectively the coordinates parallel and normal to the predominant wave direction, A_{ij} is the amplitude of the (i, j)th wave component, $S(\omega_i, \theta_j)$ is the value of the energy spectrum, θ

Manuscript approved June 30, 1986.

is the wave direction measured from the X-axis, $k = \omega_i^2/g$ is the wave number for deep water and ϕ_{ij} is a random phase angle uniformly distributed between θ and 2π. For the case where $S(\omega, \theta)$ is the energy spectrum, A_{ij} is given by

$$A_{ij} = \sqrt{2S(\omega_i, \theta_j)\ \Delta\omega\Delta\theta} \tag{2}$$

where $\Delta\omega$ and $\Delta\theta$ are increments of ω and θ, respectively. This term should be treated correctly in order to avoid errors of a factor of two in the wave elevations. In the present study all spectra are given in terms of energy, which is consistent with Eqs. (1) and (2) and it is assumed to have the commonly used separable form

$$S(\omega, \theta) = S_\omega(\omega)\ G(\theta) \tag{3}$$

where $S_\omega(\omega)$ is the frequency spectrum and $G(\theta)$ is the spreading function which must satisfy the equation

$$\int_{-\pi}^{\pi} G(\theta)\, d\theta = 1 \tag{4}$$

Three widely used frequency spectra are presently considered: the Pierson-Moskowitz, the Jonswap and the Neumann. A fourth one, the Bretschneider, which is a subset of the Pierson-Moskowitz, is also included in the computer program as a special case. In their most general form the first two are functions of the wind velocity and they are comparable to the Neumann spectrum if only forms appropriate to fully developed seas are considered. To conform with this requirement an expression for the three spectra may be written in the following form

$$S_\omega = \frac{A}{\omega^m}\ e^{-(B/\omega^n)} \cdot 3.3^q \tag{5}$$

where 3.3 is a mean enhancement factor and the values of the other parameters for each of the spectra are given in Table 1, where ω_0 is the peak frequency given by

$$\omega_0 = .8771632\ (g/\mathrm{U}) \tag{6}$$

and σ is equal to 0.07 for $\omega \leqslant \omega_0$ and equal to 0.09 for $\omega > \omega_0$.

Table 1

	Pierson-Moskowitz	Jonswap	Neumann
A	0.0081*G**2	0.0081*G**2	33.1*π/4
B	0.74*G**4/U**4	0.74*G**4/U**4	2*G**2/U**2
m	5	5	6
n	4	4	2
q	0	$e^{-[-\omega-\omega_0]^2/(2\sigma^2\omega_0^2)}$	0

A number of similarities exist among the three frequency spectra as well as some distinct differences. The Pierson-Moskowitz and Jonswap spectra are similar but with a significant exception that the Jonswap is more sharply peaked in the region of the peak frequency ω. The Neumann spectrum differs from the other two in that the values for the parameters m and n are different. The Bretschneider spectrum has the same values as the Pierson-Moskowitz for the m, n and q, but it differs in the values for A and B.

Several types of expressions can be found in the literature for the spreading function $G(\theta)$. A relatively simple yet reasonably accurate expression for this function is the widely used cosine squared function given by

$$G(\theta) = \frac{2\nu}{\pi} \cos^2 \nu\theta, \quad |\theta| < \frac{\pi}{2\nu} \tag{7}$$

where ν is the coefficient which determines the range of θ over which the energy of the wind is spread. The simplest and most popular form for $G(\theta)$ is to let $\nu = 1$. In this form, the wind is assumed to generate waves in all directions which are within ± 90 degrees from the wind direction. Values of $\nu = 3/2$ and $\nu = 3/4$ correspond to spreadings of ± 60 degrees and ± 120 degrees respectively.

The definitions and specification of the frequency spectra and spreading function are required for the calculation of the wave amplitude coefficients A_{ij}, which are given by Eq. (2). The last parameters required by Eq. (1) are the values of the x and z coordinates. These are specified by defining a rectangular grid over the sea surface, which is described in the following section. Then, at each of the grid points the sea surface elevations are calculated by using Eqs. (1), (2), (5) and (7). These equations together with the parameters of Table 1 have been implemented in the computer program KELSEA for calculating the wave elevations for random ambient sea waves.

III. SUPERPOSITION OF AMBIENT AND KELVIN WAVES

The Kelvin wave elevations $\xi(x, z)$ are obtained by taking the inverse Fourier transform of the complex wave amplitude spectrum $A(u)$ calculated by the program KELVIN (4).

$$\xi(x, z) = \frac{k_0}{U} \frac{1}{4\pi} \mathrm{Re} \int_{-\infty}^{\infty} A(u) e^{-ik_0(sx+uz)} du \tag{8}$$

where $u = k_z/k_0$ is the dimensionless lateral wavenumber, $s = k_x/k_0$ is the dimensionless longitudinal wavenumber, $k_0 = g/U^{**}2$ is the fundamental wavenumber and Re represents the real part of the complex integral. The Kelvin waves are calculated for a coordinate system fixed to the ship and with its origin at the forward perpendicular. The x and z coordinates are in the longitudinal and lateral direction respectively and on the horizontal plane. The integral in Eq. (8) is calculated for evenly spaced values of the dimensionless wavenumber u. The values of x and z are chosen to lie in the rectangular grid, with equal spacing in each direction. The extent of the grid in the lateral z direction is equal to $2\lambda_0$, ($\pm\ \lambda_0$) on either side of the x axis, where λ_0 is the fundamental wavelength given by

$$\lambda_0 = \frac{2\pi}{k_0} = \frac{4\pi U^2}{g} \tag{9}$$

Noting that the cusp line of the Kelvin waves form an angle of 19.5 degrees with the x axis, the required extent of the grid in the x direction, X_M, is simply given by

$$X_M = \frac{\lambda_0}{\tan(19.5)} \tag{10}$$

The computer program KELSEA has the options of generating the ocean and Kelvin wave elevations separately or a superposition of the two. In performing the superposition, it is of interest to note that the ocean waves resulting from Eq. (1) are generated for a coordinate system which is fixed in space while the Kelvin waves are generated for a coordinate system which is fixed to the ship. In the general case where the wave elevations are functions of time, they would differ in the two coordinate systems since the wave frequencies encountered by a moving ship are different from those experienced by a stationary one. However, in the present case, time is not a variable since the Kelvin waves expressed by Eq. (8) are steady in the moving coordinate system and the ocean waves expressed by Eq. (1) represent an image of the ocean surface at a particular instant of time, say $t = 0$. Consequently, direct superposition of the waves is permissible, without the need to account for the difference in the velocities of the two coordinate systems.

Finally, the case of ambient waves with the wind direction at an angle θ_ω with the x-axis, is conveniently treated by simply substituting $\theta' = \theta + \theta_\omega$ in Eq. (1). Examples of ambient ocean waves for different wind directions are presented in the next section.

IV. COMPUTER PROGRAM

The mathematical model discussed in the previous sections has been incorporated into the computer program KELSEA. The major tasks performed by this program are discussed in this section as well as the structure of the KELSEA program. Furthermore, the interactive capabilities of the program are demonstrated by a typical example run and calculated results for certain input parameters are also given.

The KELSEA program is designed to perform three functions: a) It performs the random ambient sea state calculations according to specified input parameters, b) It computes the free surface elevations for Kelvin waves from a pre-calculated spectral file, and c) It superimposes the surface elevations for a Kelvin wave and the ambient sea waves. The superposition is a linear point to point superposition and should not be interpreted as an interaction between the sea waves and the Kelvin wave. The code is written in FORTRAN-77 and uses menus and inquiries to enhance its use and it is currently operating in a number of computer systems such as the HP-1000, HP-9000 and VAX 11/780 systems. The modular structure of the computer program is shown in Fig. 1.

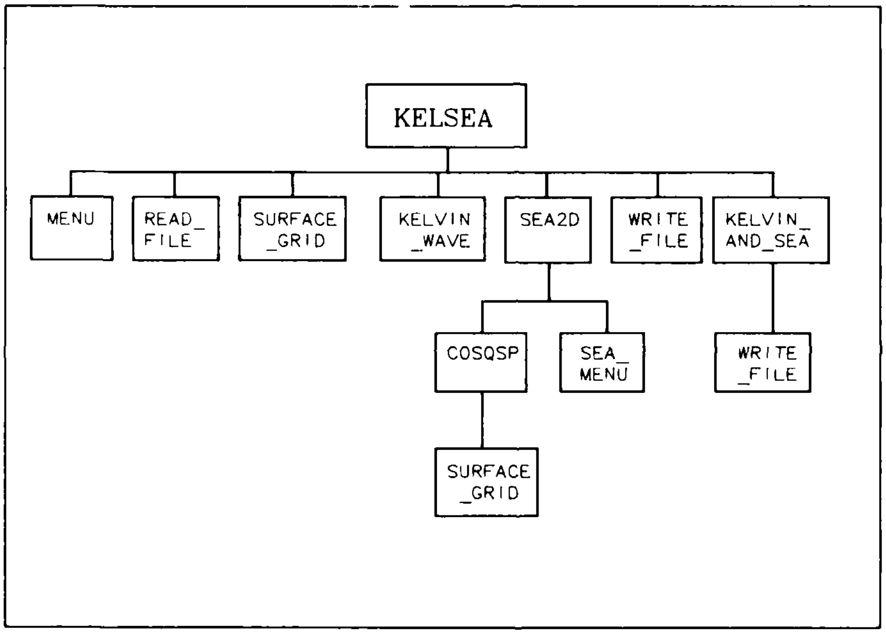
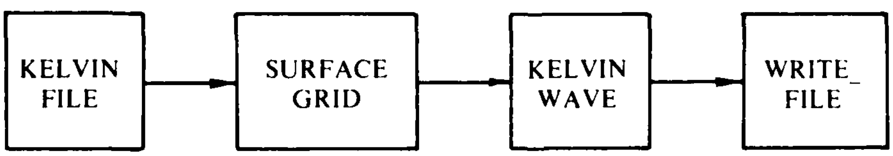


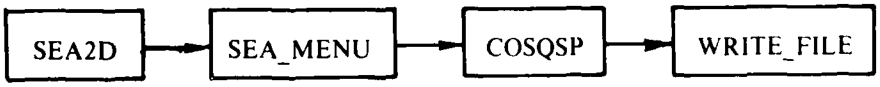
Fig. 1 — KELSEA computer program diagram

KELSEA is programmed to perform three independent functions. Each function is directed by a program menu. The first menu item is the Kelvin wave heights calculations. Illustrated below is the program flow diagram for calculating the free surface elevations of the Kelvin waves.

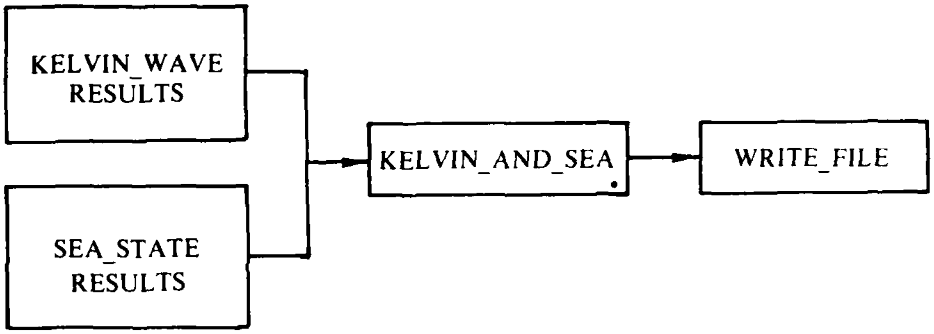


This task requires the input of a Kelvin spectral file, which is created from the KELVIN program. First, the free surface grid is generated in the SURFACE_GRID routine. Then the surface elevations for the Kelvin waves are calculated in the KELVIN_WAVE routine and the resulting output data are stored in a file for plotting.

The next task on the menu is the sea state calculations. Its input data is program-generated based on user specified parameters and spectrum type. It is important to note here that for every Kelvin spectral file there must be a sea state calculation corresponding to the same surface grid in order to correctly superimpose the calculations. The reason is that velocity will vary for every Kelvin spectral file and the grids will be dimensioned differently. Once the surface grid is created and the sea state elevations are computed in the COSQSP routine, the results are written into a file by the WRITE_FILE routine. The following diagram shows the program flow of the sea waves calculations.



The third task performed by the KELSEA program is the linear superposition of the Kelvin wave and sea state calculations.



This task requests for two input files: the Kelvin wave surface elevations data file and the sea wave elevations data file. These two files must be created before the linear superposition of the two can be performed. The previous tasks explained above generate the output files that can be used as the input here. The computations take place in the KELVIN_AND_SEA routine and the results are written into a file by the WRITE_FILE routine. Any of these output files can be plotted by the PLOT3D program [5] as three dimensional surfaces.

The three tasks for the KELSEA program are controlled by the program main menu.

The user is requested to make a selection from the program menu as illustrated below.

```
**************         MENU         **************
**                                              **
** Select Item by Number                        **
**                                              **
**    (1) Kelvin Wave Calculations              **
**    (2) Sea State Calculations                **
**    (3) Superposition of (1) & (2)            **
**    (4) End of Calculations                   **
**                                              **
** Specify Menu Item       :
```

IV.1 Kelvin Wave Calculations

To run the Kelvin wave calculations enter the corresponding number.

```
** Specify Menu Item      : 1
```

The program will inquire for the name of the Kelvin spectral data file to be read and stored for calculations.

```
************************************************************
**** Specify the Name of the Data File from which       ****
**** this Program will Read the Input Data              ****
**** File Name :KELVIN.DAT
```

The description of the file contents are printed on the screen as follows

```
Data Read from File          : KELVIN.DAT

Velocity (Ft./Sec.)      =      30.000
Characteristic Wave #    =         .04
Y Dist to P Symmetry     =        0.00
Displacement (Tons)      =    34461.18
Waterline Length (Ft.)   =      300.00
Wet Surf (Ft.**2)        =     1041.00
L Center Buoy. (Ft.)     =     -136.56
# of Pos. Wave Nos.      =      350
2 Hulls? (Y=1/N=0)       :        0
```

The next input required by the program is the number of points for the rectangular grid in the X and Z directions and the starting X-coordinate for grid. Currently the axes origin is at the forward perpendicular with the X-axis along the ship length taken as negative and with the starting position equal to a ship length. The user has the option of using the default values given or define new values for the number of points. In this case, enter the default values.

```
************************************************************
**** Specify the Number of Points for the Surface        ****
**** Grid in both X- and Z- directions.                  ****
**** Default values are NXPOINT=61, NZPOINT=41           ****
**** NX-POINT =61
**** NZ-POINT =41

************************************************************
**** Specify the starting value in the X direction      ****
**** for the rectangular grid in (ft). The default      ****
**** value is equal to the Ship Length. XMIN =
```

Once the surface grid has been defined, the program computes the free surface heights for the Kelvin wave. After the calculation process is completed, the next step is to write the results to a file. The program indicates the type of results which are to be saved as follows

```
>>>>>>>>>> KELVIN WAVE OUTPUT FILE <<<<<<<<<<
```

and it requests for a name of the output file to save the results.

```
**********************************************************
     **** Specify the Name of th. Data File For Storing      ****
     **** the Output Results for Plotting                    ****
     **** File Name :KELPLOT.DAT
```

The resulting information about the file just created is written on the screen, which concludes the task processing for the Kelvin wave calculations, and the program returns to the main menu.

```
Data Written to File  : KELPLOT.DAT

NX_POINT = 61
NZ_POINT = 41
```

IV.2 Sea State Calculations

The program menu appears on the screen for the user to make another selection. To calculate the sea state elevations enter the corresponding number (2) to the menu inquiry.

The following notice is displayed on the screen reminding the user that a corresponding Kelvin file must exist for the same grid dimensions and velocity.

```
**********************************************
**** NOTICE:                             ****
****    This task requires that a Kelvin  ****
****    spectral file has been read to    ****
****    determine the parameters needed   ****
****    for the surface grid calculations. ****
****    If a file is not available you will ****
****    be asked to specify the dimensions ****
****    of the physical space for the      ****
****    sea state calculations.            ****
****                                       ****
**********************************************
```

The next request by the program is the specification of the spectrum type. One of the options given below is selected by entering the corresponding number.

```
**   Specify the Spectrum Type
**   for the Sea State Calculations
**   Input     (1)  for PIERSON-MOSKOWITZ
**             (2)  for JONSWAP
**             (3)  for NEUMANN
**             (4)  for BRETSCHNEIDER
**   Specify the Approiate Spectrum : 1
```

The inquiries below are prompted one at a time in order to specify the appropriate value.

```
**   For  60 Deg., Spreading index = 1.50 **
**   For  90 Deg., Spreading index = 1.00 **
**   for 120 Deg., Spreading index = 0.75 **
**   Specify the Spreading Index (ge .5)  : 0.5

**   Specify the Seed Number
**   for Random Phase Angle (ISEED.GE.1)  : 1
```

** Specify Wind Speed [S.S. 3 = 16 Kts] : 16

** Specify the Wind Direction
** with respect to the X-axis in deg. : 45

** Specify the Number of Wave
** Frequencies (1 to 101) : 10

** Specify ZERO (0) for Equal ASW
** or ONE (1) for Equal Delta Freq. : 0

The next input required by the program is the number of points for the surface grid calculations and the starting value in the X direction.

```
**********************************************************
**** Specify the Number of Points for the Surface          ****
**** Grid in both X- and Z-directions.                    ****
**** Default Values are NX_POINT=61, NZ_POINT=41          ****
**** NX-POINT =61
**** NZ-POINT =41

**********************************************************

**** Specify the starting value in the X direction      ****
**** for the rectangular grid in (ft). The default      ****
**** value is equal to the Ship Length. XMIN =
```

If the user were to calculate the sea wave elevations only, then the inquiries below would be made. Since in the current example a KELVIN file was read, the program will use that information for the actual dimensions of the surface grid. Remember that the surface grid for the Kelvin and sea wave elevations must be of the same dimension and relating to the same ship velocity in order to make a rational superposition of the resulting calculations.

```
*****************************************************
**** Specify the Value for the Rectangular Grid      ****
**** in the X direction in ft. ( XMAX )              ****
**** XMAX =

*****************************************************
**** Specify the Value for the Rectangular Grid      ****
**** in the Z direction in ft. ( -ZMIN , +ZMAX )     ****
**** ZMAX =
```

The above two inquiries can be used if one wishes to generate an ambient sea state independent from a Kelvin wave.

The free surface grid and elevations are calculated for the sea state task. The output message below indicates to the user which results have been computed and to be written to a file.

```
>>>>>>>> AMBIENT SEA WAVE OUTPUT FILE <<<<<<<<
```

The user is to input the file name for the sea wave results.

```
*****************************************************
**** Specify the Name of the Data File For Storing   ****
**** The Output Results for Plotting                 ****
**** File Name :SEAPLOT.DAT
```

IV.3 Superposition of the Kelvin and Sea Wave Elevations

The program returns control to the main Menu for a selection to be made. Enter the corresponding number (3) for the next task.

There will be a message printed on the screen reminding the user that before the program continues, both data files for the Kelvin and sea wave elevations must exist, for the linear superposition to be processed. Answer the inquiry and continue.

```
*********************************************
**** NOTICE :                            ****
****   This task requires that the Kelvin ****
****   wave and the sea wave elevation    ****
****   data files have been created.      ****
****                                      ****
**** Enter (Y) if the files exist and     ****
**** continue with the calculations.      ****
**** Or, enter (N) if the files have not  ****
**** been created and return to the       ****
**** program menu to create the files.    ****
****                                      ****
**** ENTER Y/N :Y
```

The program will inquire for the names of these two data files and it will check for consistency of the two surface grids. If the surface grids corresponding to the two specified file are not the same an error message will be printed on the screen.

```
****************************************************
**** Specify the Name of the Input File for the ****
**** KELVIN WAVE Results                         ****
**** File Name :KELPLOT.DAT

****************************************************
**** Specify the Name of the Input File for the ****
**** SEA STATE Results                           ****
**** File Name :SEAPLOT.DAT
```

The output message below indicates to the user which output results are to be stored in a file for later use.

```
>>>>>>>> KELVIN/SEA WAVE OUTPUT FILE <<<<<<<<
******************************************************
**** Specify the Name of the Data File For Storing ****
**** The Output Results for Plotting               ****
**** File Name :KELSEA.DAT
```

Once again the program menu appears on the screen. All the tasks have been processed and the user can exit the program by entering the corresponding number (4) to the inquiry.

V. NUMERICAL RESULTS

The three output files created by the KELSEA program can be plotted as three dimensional surfaces. For the purpose of providing some examples of numerical results in this paper, a spectral input file will be used. This file has been generated by the program KELVIN [4], which calculates the Kelvin wave for a specific ship hull configuration. In this particular example the spectal file corresponds to a mono-hull ship configuration. The relevant parameters from the input file are given in Table 2.

The resulting output files for the Kelvin wave elevations are designated as KELWAV1.DAT and KELWAV2.DAT respectively. Similarly the output files for the sea wave elevations are designated as SEAWAV11.DAT, SEAWAV12.DAT, SEAWAV21.DAT and SEAWAV22.DAT. The input parameters used for generating these four files are given in Table 3.

Table 2

Input File : KELVIN1.DAT			Input File : KELVIN2.DAT		
Velocity (Ft./Sec.)	=	31.45	Velocity (Ft./Sec.)	=	31.45
Characteristic Wave #	=	.03	Characteristic Wave #	=	.03
Y Dist to P Symmetry	=	0.00	Y Dist to P Symmetry	=	80.00
Displacement (Tons)	=	2599.60	Displacement (Tons)	=	5199.20
Strut Length (Ft.)	=	300.00	Strut Length (Ft.)	=	300.00
S Wet Surf (Ft.**2)	=	15274.50	S Wet Surf (Ft.**2)	=	30549.00
L Center Buoy.(Ft.)	=	−157.825	L Center Buoy.(Ft.)	=	−157.825
# of Pos. Wave Nos.	=	300	# of Pos. Wave Nos.	=	300
2 Hulls? (Y=1/N=0)	:	0	2 Hulls? (Y=1/N=0)	:	1

Table 3

Input file	:	KELVIN1.DAT	Input file	:	KELVIN2.DAT
Output Files	:	KELWAV1.DAT	Output Files	:	KELWAV2.DAT
		SEAWAV11.DAT			SEAWAV21.DAT
		SEAWAV12.DAT			SEAWAV22.DAT

Sea State case : ONE			Sea State case : TWO		
Spectrum Type	=	1	Spectrum Type	=	1
Spreading Index	=	1.0	Spreading Index	=	1.0
Seed Number	=	1	Seed Number	=	1
Wind Velocity	=	16	Wind Velocity	=	16
Wind Direction	=	0	Wind Direction	=	90
Number of wave Freq.	=	5	Number of wave Freq.	=	5

The files SEAWAV11.DAT and SEAWAV21.DAT correspond to case one for the sea state. Similarly SEAWAV12.DAT and SEAWAV22.DAT correspond to case 2 for the sea state. The wind velocity of 16 knots corresponds to a Sea State of (3).

The numerical results from the six output files are presented in Figs. 2-7 as follows

KELWA1.DAT,	Fig. 2	KELWAV2.DAT,	Fig. 5
SEAWAV11.DAT,	Fig. 3	SEAWAV21.DAT,	Fig. 6
SEAWAV12.DAT,	Fig. 4	SEAWAV22.DAT,	Fig. 7

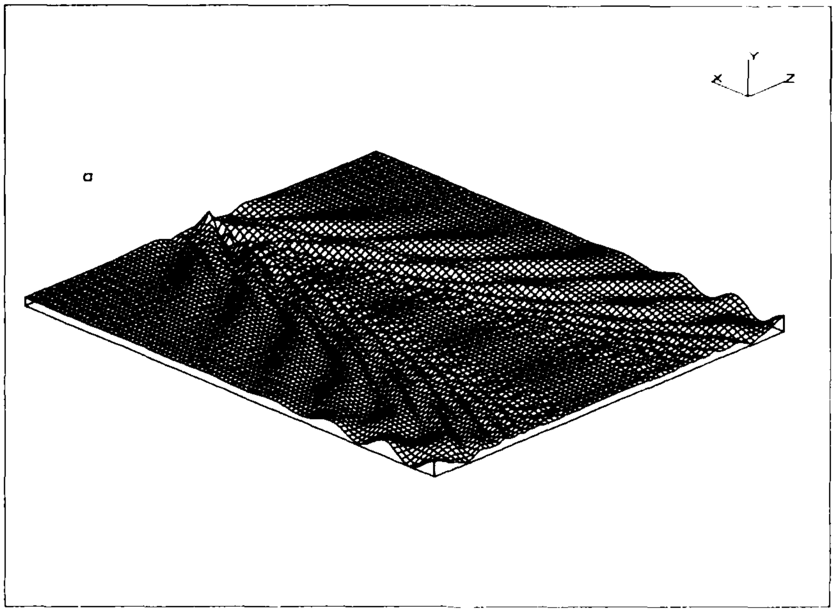


Fig. 2 — Graphical presentation of Kelvin wave elevations

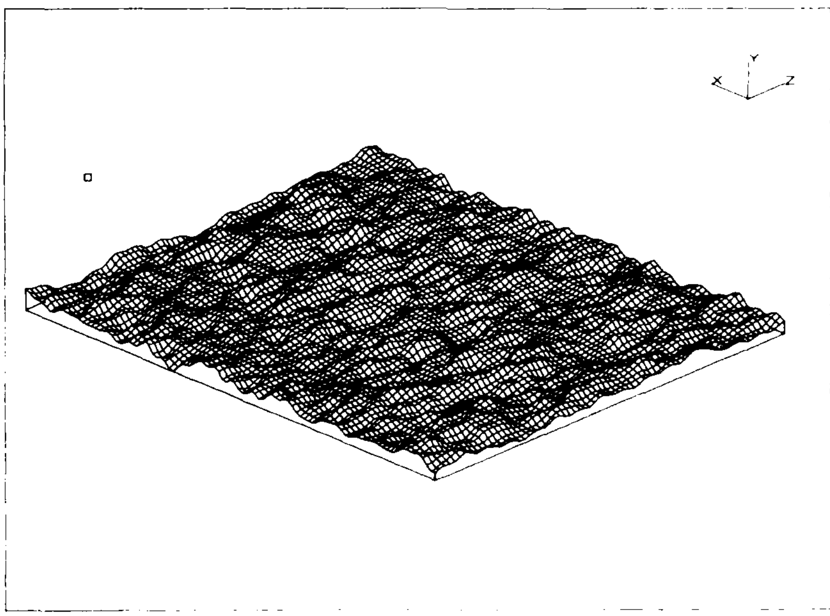


Fig. 3 — Graphical presentation of ambient sea wave elevation

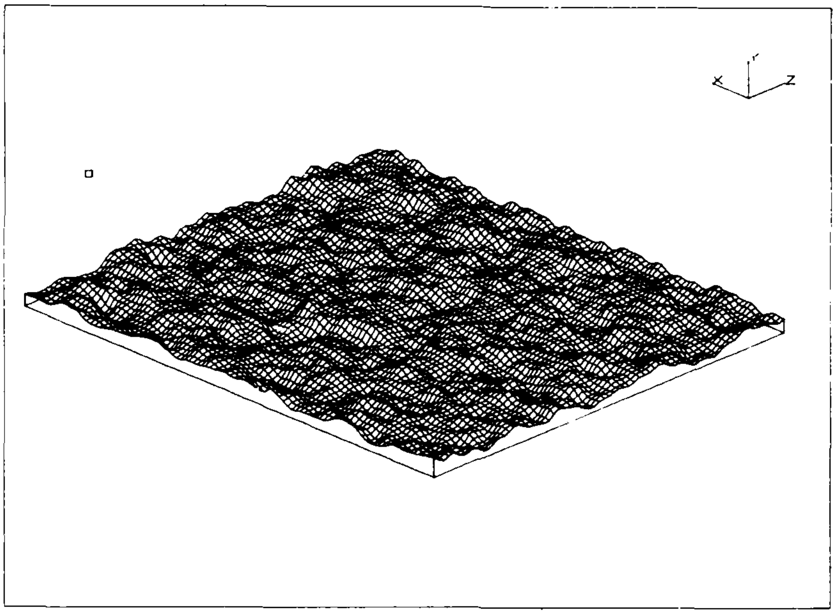


Fig. 4 — Graphical presentation of the superposition of the Kelvin waves and ambient sea waves

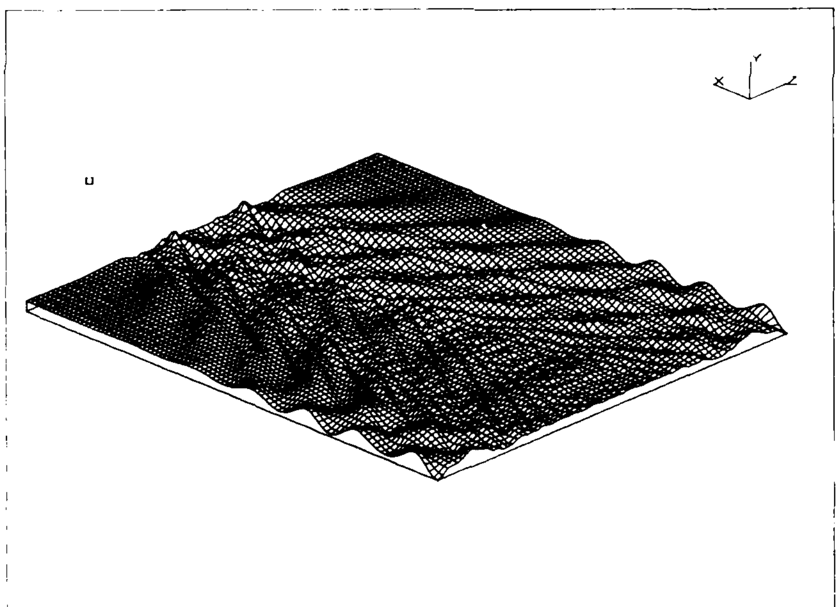


Figure 5

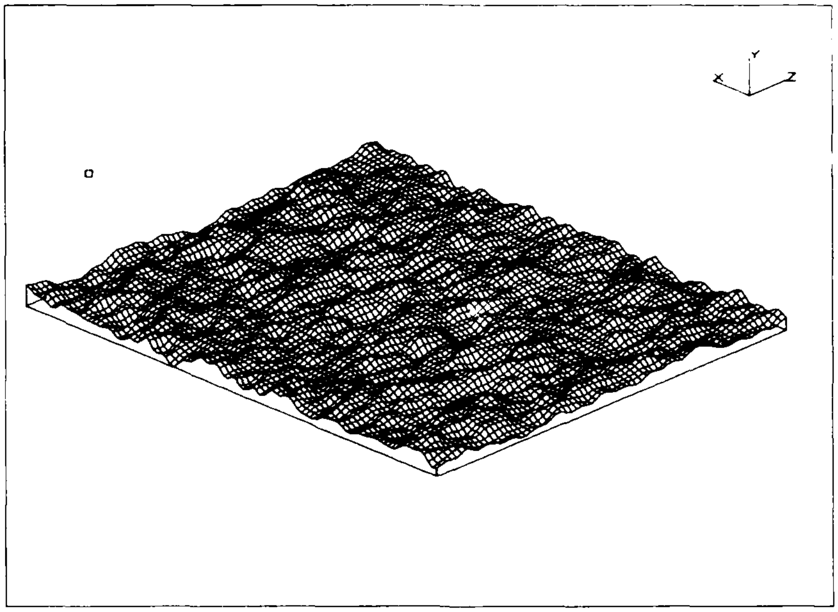


Figure 6

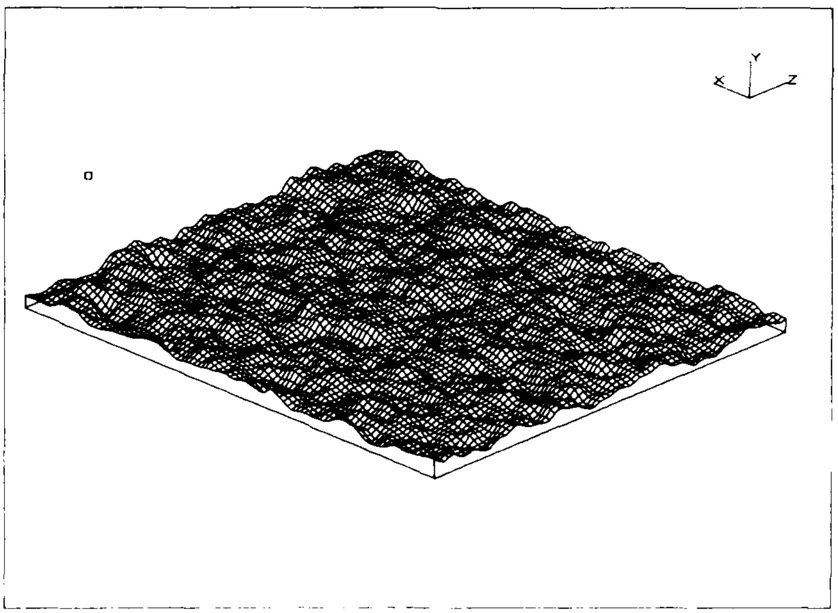


Figure 7

The superposition of each of the Kelvin wave elevation files with each of the four sea state files will result in the following output files:

KELWAV1.DAT + SEAWAV11.DAT → KELSEA111.DAT, Fig. 8

KELWAV1.DAT + SEAWAV12.DAT → KELSEA112.DAT, Fig. 9

KELWAV1.DAT + SEAWAV21.DAT → KELSEA121.DAT, Fig. 10

KELWAV1.DAT + SEAWAV22.DAT → KELSEA122.DAT, Fig. 11

KELWAV2.DAT + SEAWAV11.DAT → KELSEA211.DAT, Fig. 12

KELWAV2.DAT + SEAWAV12.DAT → KELSEA212.DAT, Fig. 13

KELWAV2.DAT + SEAWAV21.DAT → KELSEA221.DAT, Fig. 14

KELWAV2.DAT + SEAWAV22.DAT → KELSEA222.DAT, Fig. 15

The plotted results of these eight files are presented in Figs. 8 through 15. These results have been plotted with the PLOT3D computer program.

VI. CONCLUSIONS

The mathematical model for calculating ambient sea waves has been presented in this report. The modeling and calculations of the Kelvin wave elevations and their superposition onto the ambient sea waves also has been presented.

The computer program implementing these concepts has been described and the program structure has been discussed. A number of numerical examples have been given in order to illustrate the program capabilities.

The computer program has been developed in such a way that it is simple to operate and the user does not need to refer to manuals for exercising the program. Furthermore, the input parameters for the sea wave calculations add flexibility to the types of Sea States that one can generate.

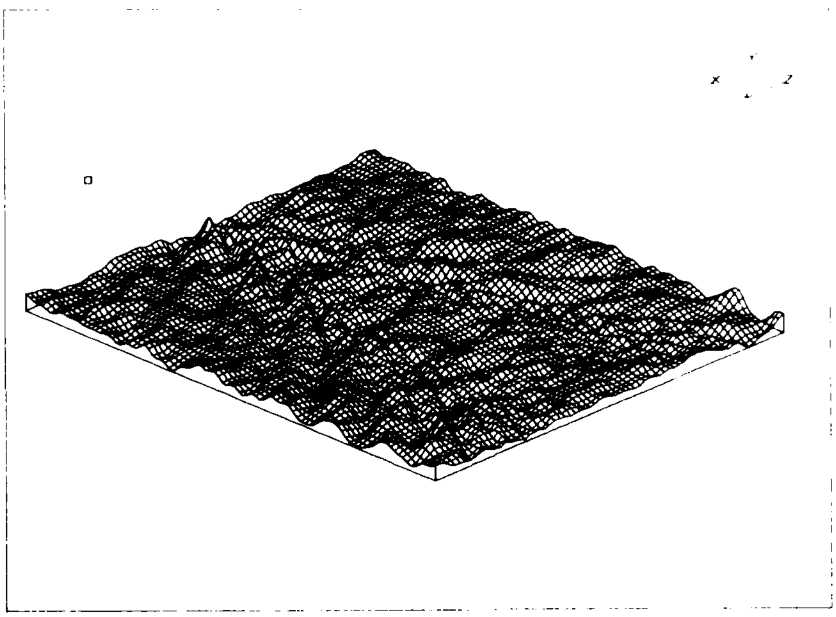


Figure 8

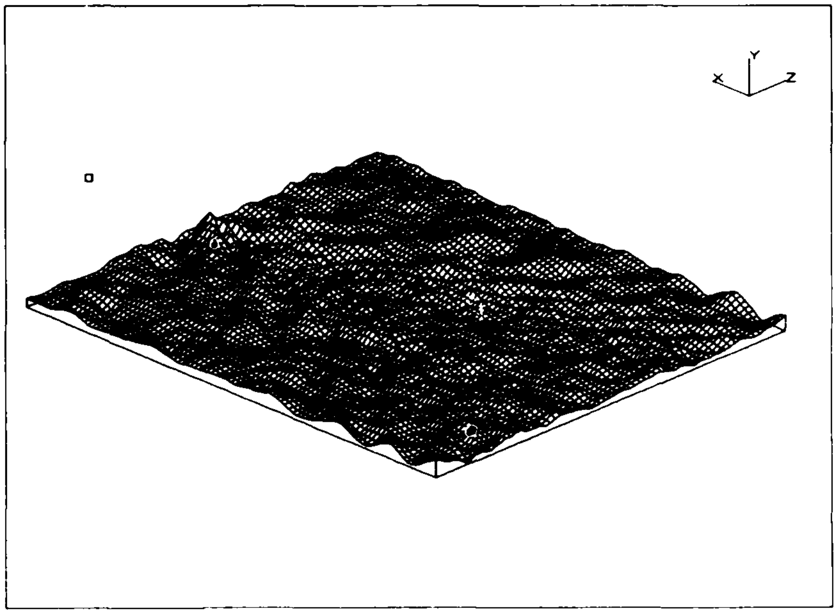


Figure 9

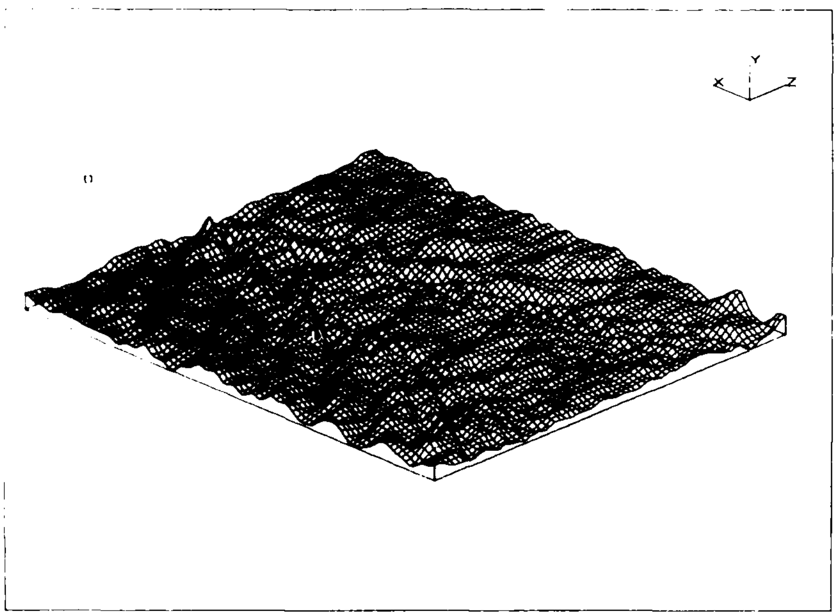


Figure 10

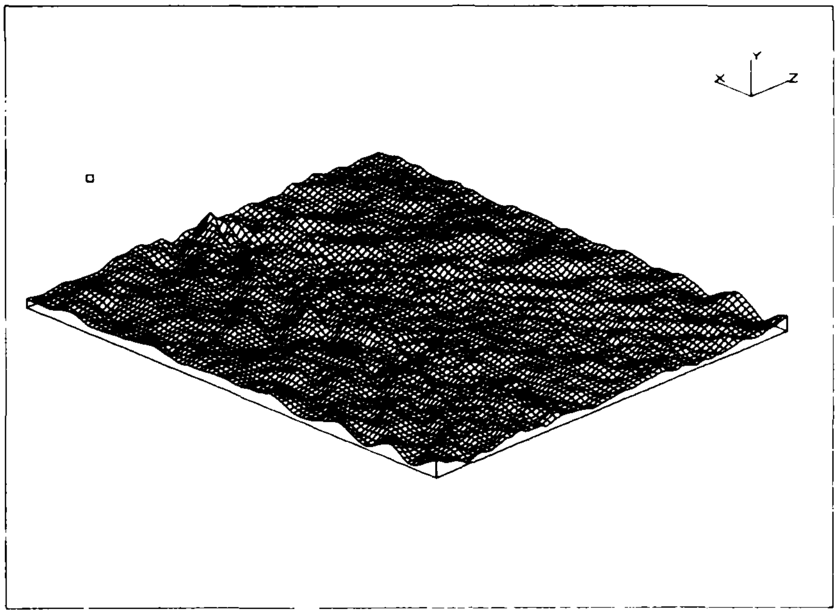


Figure 11

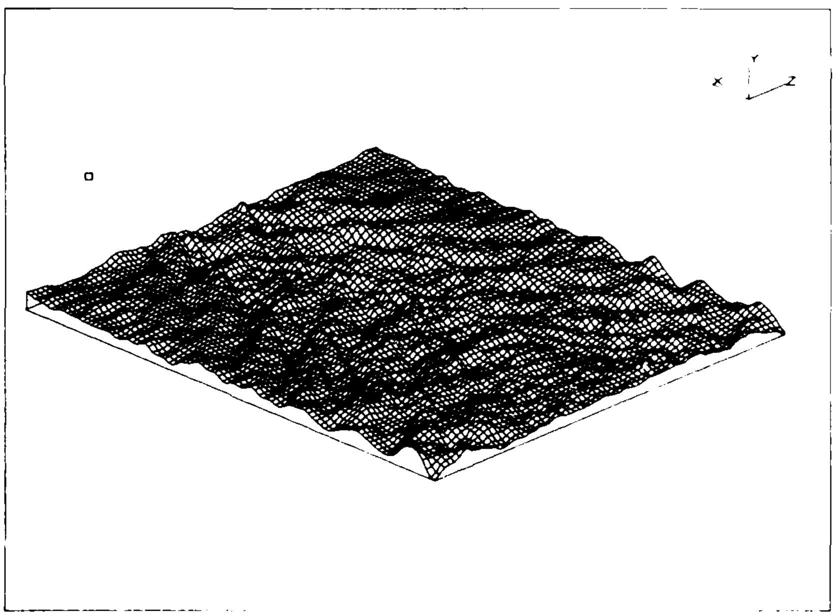


Figure 12

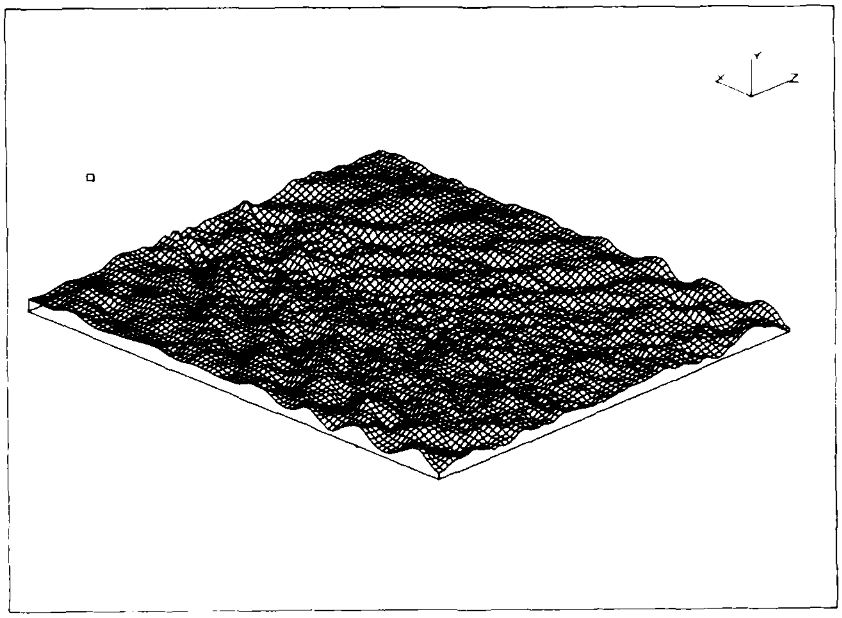


Figure 13

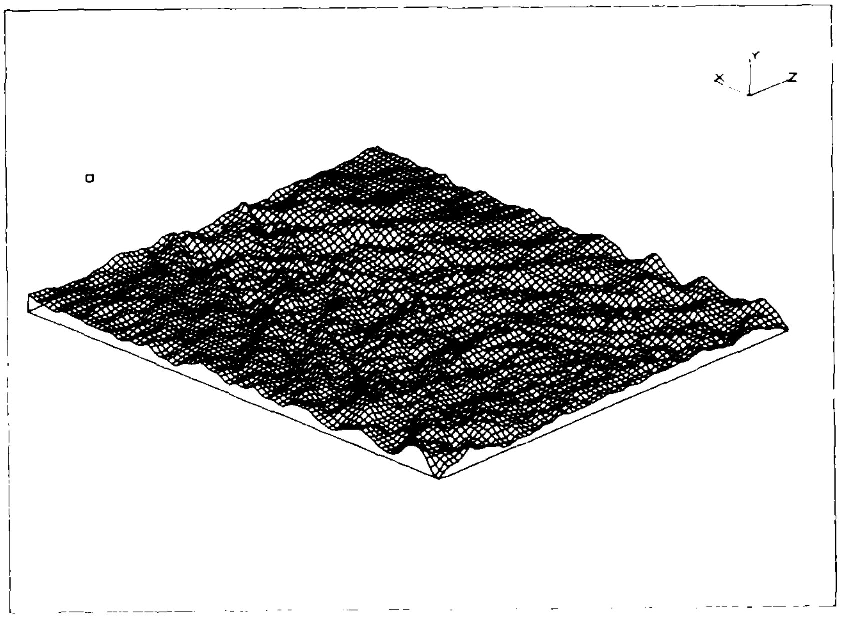


Figure 14

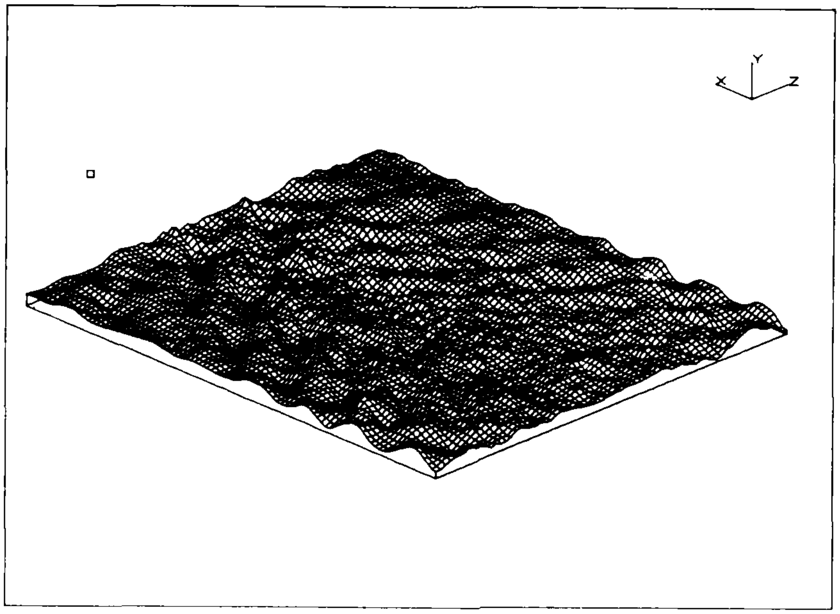


Figure 15

VII. REFERENCES

1. Kinsman, B., "Wind Waves", Prentice-Hall Inc., 1965, Englewood Cliffs, New Jersey, USA.

2. Sarpkaya, T. and Isaacson, M., "Mechanics of Wave Forces on Offshore structure," Van Nostrand Reinhold Co., 1981, New York, New York, USA.

3. Wang, H.T., "Temporal and Spatial Simulations of Random Ocean Waves," Fourth Int. Offshore Mechanics and Arctic Engineering Symposium, 1985, Dallas, Texas, Vol. 1, pp. 72-80.

4. Wang, H.T. and Keramidas, G.A., "Computer Model for the Calculation of the Kelvin spectrum with Viscous Corrections," CADMO86 Conference, 1986, Washington, D.C., USA.

5. G.A. Keramidas, E.W. Miner and W. Bauman, "PLOT3D: An Interactive Graphics Code for Three Dimensional Plots," NRL Memorandum Report 5410, Sept. 1984.

Appendix

KELSEA Program Listing

```
*=====================================================================*
*                                                                     *
     PROGRAM KELSEA
*                                                                     *
*=====================================================================*
*                                                                     *
* Revision List:                                                      *
*                                                                     *
* --Date-- --by-- -- D e s c r i p t i o n --                         *
*                                                                     *
*  Jun. 1985 -  First Revised Version for H-P Computers, Ver. 1.0     *
*  Sep. 1985 -  Final Version for H-P Computers, Ver. 1.2             *
*  Dec. 1985 -  VAX Version, Ver. 1.2V                                *
*                                                                     *
*=====================================================================*
**                                                                   **
*****  Free Surface Heights Calculation for a Kelvin Wave
*****  and Sea State Calculations.
**
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK2/ CGG(1000),CGM(1000)
      COMMON/BLOCK3/ U_KELVIN(1000),V_KELVIN(1000),W_KELVIN(1000)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      COMMON/BLOCK5/ VEE,AK,YSYM,DISPT,XST,ZST,AREAS,CB,N_POINTS,NSYM
      COMMON/BLOCK6/ WM(101),AZD(101),WZD(101),ENF(101),EN(161)
      COMMON/BLOCK7/ NSP,FNXP,ISEED,WVKT,THSTD,TZ,NF,IEDW
      DIMENSION LBUF(1000)
      CHARACTER*20 FILE1,FILE2
      COMPLEX CGG,CGM
**
      CALL LGBUF(LBUF,1000)
**
      GRID_SIZE = 0.0                   ! Default value.
      NX_POINT  = 61                    ! Default value.
      NZ_POINT  = 41                    ! Default value.
**
   10 CALL MENU(IDMENU)
**
      GO TO (100,200,300,500) IDMENU
  100 CONTINUE
**
*****  Read Spectral File for Height Calculations.
**
      CALL READ_FILE(FILE1)
**
      WRITE(1,'(/,5X,"Data Read from File    : ",A20)') FILE1
      WRITE(1,'(/,5X,"Velocity  (Ft./Sec.) =",F10.2,/
     +         5X,"Characteristic Wave #=",F10.2,/
     +         5X,"Y Dist to P Symmetry =",F10.2 )') VEE,AK,YSYM
      WRITE(1,'(/,5X,"Displacement (Tons)  =",F10.2,/
     +         5X,"Strut Length (Ft. )  =",F10.2,/
     +         5X,"S Wet Surf (Ft.**2)  =",F10.2,/
     +         5X,"L Center Buoy.(Ft.)  =",F10.2)') DISPT,XST,AREAS,CB
      WRITE(1,'(/,5X,"# of Pos. Wave Nos.  =",I10,/
     +         5X,"2 Hulls? (Y=1/N=0)   :",I10,/)') N_POINTS,NSYM
```

```
**
*****  Calculate Free Surface Elevation.
**
      CALL KELVIN_WAVE
**
*****  Store Data File for Plotting Routine.
**
      CALL WRITE_FILE(FILE2, 'KELVIN WAVE')
**
        NXPL = NX_POINT
        NZPL = NZ_POINT
      WRITE(1,'(/,5X,"Data Written to File  : ",A20)') FILE2
      WRITE(1,'(/,5X,"NX_POINT =",I4,/5X,"NZ_POINT =",I4)') NXPL,NZPL
      GO TO 10
**
*****  Sea State Calculations.
**
 200  CONTINUE
**
        IERROR = 0
      CALL SEA2D(IERROR)
      IF(IERROR.NE.0) GO TO 10
      CALL WRITE_FILE(FILE2,'AMBIENT SEA WAVE')
**
      GO TO 10
**
*****  Linear Superposition of the Sea State and Kelvin Wave.
**
  300 CONTINUE
      CALL KELVIN_AND_SEA
      GO TO 10
**
  500 CONTINUE
**
      STOP
      END
```

```
**
*********************************************************************
*                                                                   *
      SUBROUTINE MENU(IDMENU)
*                                                                   *
*********************************************************************
**                                                                 **
      CHARACTER*1 ANS
**
  10  WRITE(1,'( 5/)')
      WRITE(1,'(10X,"**************     MENU     **************")')
      WRITE(1,'(10X,"**                                      **")')
      WRITE(1,'(10X,"** Select Item by Number                **")')
      WRITE(1,'(10X,"**                                      **")')
      WRITE(1,'(10X,"**    (1) Kelvin Wave Calculations      **")')
      WRITE(1,'(10X,"**    (2) Sea State Calculations        **")')
      WRITE(1,'(10X,"**    (3) Superposition of (1) & (2)    **")')
      WRITE(1,'(10X,"**    (4) End of Calculations           **")')
      WRITE(1,'(10X,"**                                      **")')
      WRITE(1,'(10X,"** Specify Menu Item     : $")')
      READ(1,*) IDMENU
      IF(IDMENU.LT.1.OR.IDMENU.GT.4) GO TO 10
**
      IF (IDMENU.EQ.2) THEN
        WRITE(1,'(///10X,
     .           "*******************************************",
     .        /10X,"****  NOTICE :                          ****",
     .        /10X,"****    This task requires that a Kelvin ****",
     .        /10X,"****    spectral file has been read to   ****",
     .        /10X,"****    determine the parameters needed  ****",
     .        /10X,"****    for the surface grid calculations. ****",
     .        /10X,"****    If a file is not available you will ****",
     .        /10X,"****    be asked to specify the dimensions ****",
     .        /10X,"****    of the physical space for the sea ****",
     .        /10X,"****    state calculations.              ****",
     .        /10X,"****                                     ****",
     .        /10X,"*******************************************",
     .        /10X,"  ")')
      END IF
**
      IF (IDMENU.EQ.3) THEN
        WRITE(1,'(///10X,
     .           "*******************************************",
     .        /10X,"****  NOTICE :                          ****",
     .        /10X,"****     This task requires that the Kelvin ****",
     .        /10X,"****     wave and the sea state elevations ****",
     .        /10X,"****     data files have been created.   ****",
     .        /10X,"****                                     ****",
     .        /10X,"****  Enter (Y) if the files exist and   ****",
     .        /10X,"****  continue with the calculations.    ****",
     .        /10X,"****  Or, enter (N) if the files have not ****",
     .        /10X,"****  been created and return to the     ****",
     .        /10X,"****  program menu to create the files.  ****",
     .        /10X,"****                                     ****",
     .        /10X,"****  ENTER Y/N :$")')
        READ(1,'(A1)') ANS
        IF ((ANS.EQ.'n' .OR.(ANS.EQ.'N')) GOTO 10
      END IF
**
      RETURN
      END
```

```
**
*****************************************************************
*                                                               *
      SUBROUTINE SURFACE_GRID(INDEX)
*                                                               *
*****************************************************************
**                                                             **
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      COMMON/BLOCK5/ VEE,AK,YSYM,DISPT,XST,ZST,AREAS,CB,N_POINTS,NSYM
**
*****  Calculate Free surface Grid -
*****  Default values for NX & NZ points are 61 & 41.
*****  The starting value for the X-axis (XST) is typically
*****  equal to a ship length and the axes system has to correspond
*****  to the axes used to caclulate the Kelvin spectral functions.
*****  Currently the axes origin is at the forward perpendicular
*****  with the X-axis along the ship length taken as negative.
**
      PI = 3.141593
      WRITE(1,
     +     '("*****************************************************"
     +    ,/,"**** Specify the Number of Points for the Surface   ****"
     +    ,/,"**** Grid in both X- and Z- directions.             ****"
     +    ,/,"**** Default Values are NX_POINT=61, NZ_POINT=41    ****"
     +    ,/,"**** NX-POINT =_")')
      READ(1,'( I3)') NX_POINT
      WRITE(1,'("**** NZ-POINT =_")')
      READ(1,'( I3)') NZ_POINT
      WRITE(1,
     +     '("*****************************************************"
     +    ,/,"**** Specify the starting value in the X direction  ****"
     +    ,/,"**** for the rectangular grid in (ft). The default  ****"
     +    ,/,"**** value is equal to the Ship Length. Type / for  ****"
     +    ,/,"**** the default value.  XMIN =_")')
      READ(1, * ) XMIN
      IF(INDEX.EQ.2.AND.GRID_SIZE.EQ.0.0) THEN
      WRITE(1,
     +     '("*****************************************************"
     +    ,/,"**** Specify the ending value in the X direction    ****"
     +    ,/,"**** for the rectangular grid in (ft). ( XMAX )     ****"
     +    ,/,"**** XMAX =_")')
      READ(1, * ) XMAX
      WRITE(1,
     +     '("*****************************************************"
     +    ,/,"**** Specify the Value for the Rectangular Grid     ****"
     +    ,/,"**** in the Z direction in ft. ( -ZMAX , +ZMAX )  ****"
     +    ,/,"**** ZMAX =_")')
      READ(1, * ) ZMAX
**
        XST = XMIN
      ELMAX = ZMAX
**
      ELSE
      ELMAX = 4.0*PI*VEE*VEE/32.174
       XMAX = ELMAX/TAN(19.5*PI/180.0)
      END IF
```

```
**
          DX = XMAX/FLOAT(NX_POINT-1)
         ZST = ELMAX
          DZ = 2.0*ZST/FLOAT(NZ_POINT-1)
**
      GRID_SIZE = 2.0*ZST*(XMAX - XST)
**
      DO N = 1,NX_POINT
        X_POINT(N) = XST - FLOAT(N-1)*DX
      END DO
**
      DO  N = 1,NZ_POINT
        Z_POINT(N) = - ZST + FLOAT(N-1)*DZ
      END DO
**
      RETURN
      END
```

```
**
***********************************************************************
*                                                                     *
      SUBROUTINE KELVIN_WAVE
*                                                                     *
***********************************************************************
**                                                                   **
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK2/ CGG(1000),CGM(1000)
      COMMON/BLOCK3/ U_KELVIN(1000),V_KELVIN(1000),W_KELVIN(1000)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      COMMON/BLOCK5/ VEE,AK,YSYM,DISPT,XST,ZST,AREAS,CB,N_POINTS,NSYM
      COMPLEX CGG,CGM,DGZ,DGX,EYE
      COMPLEX CGGN,CGMN
      DIMENSION STT(101)
**
      EYE = (0.0,1.0)
      DO  I = 1,N_POINTS
        FI = 0.00
        IF(I.GT.1) FI = U_KELVIN(I) - U_KELVIN(I-1)
        IF(I.LT.N_POINTS) FI = FI + U_KELVIN(I+1) - U_KELVIN(I)
        CGG(I) = 0.5*CGG(I)*FI
        CGM(I) = 0.5*CGM(I)*FI
      END DO
**
*****  Calculate Free surface Grid.
**
      INDEX = 1
      CALL SURFACE_GRID(INDEX)
            DXN = DX
            DZN = DZ
           XSTN = XST
           ZSTN = -ZST + YSYM
**
*****  Calculate Free Surface Heights.
*****  Store in Matrix Y_POINT(I,J).
**
      DO  N = 1,N_POINTS
        UKELVIN = U_KELVIN(N)
        VKELVIN = V_KELVIN(N)
         CGM(N) =  CGM(N)*CEXP(-AK*EYE*(XSTN*VKELVIN - ZSTN*UKELVIN))
         CGG(N) =  CGG(N)*CEXP(-AK*EYE*(XSTN*VKELVIN + ZSTN*UKELVIN))
      END DO
**
      DO  I = 1,NX_POINT
**
        DO  J = 1,NZ_POINT
          STT(J) = 0.0
        END DO
**
```

```
        DO   N = 1,N_POINTS
          CGGN  =  CGG(N)
          CGMN  =  CGM(N)
            AS  =  REAL(CGGN + CGMN)
            BS  =  AIMAG(CGGN - CGMN)
**
         UKELVIN = U_KELVIN(N)
         VKELVIN = V_KELVIN(N)
**
             DGX  =  CEXP(-AK*EYE*DXN*VKELVIN)
          CGG(N)  =  CGG(N)*DGX
          CGM(N)  =  CGM(N)*DGX
**
             DGZ  =  CEXP(-AK*EYE*DZN*UKELVIN)
              SD  =  REAL(DGZ)
              TD  =  AIMAG(DGZ)
**
            DO   J = 1,NZ_POINT
              STT(J) = STT(J) + AS
              AST = AS*SD - BS*TD
              BS = BS*SD + AS*TD
              AS = AST
            END DO
**
        END DO
**
        DO   J = 1,NZ_POINT
          Y_POINT(I,J) = STT(J)*AK/VEE
        END DO
**
      END DO
**
      RETURN
      END
```

```
**
*******************************************************************
*                                                                 *
      SUBROUTINE READ_FILE(FILE_NAME)
*                                                                 *
*******************************************************************
**                                                               **
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK2/ CGG(1000),CGM(1000)
      COMMON/BLOCK3/ U_KELVIN(1000),V_KELVIN(1000),W_KELVIN(1000)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      COMMON/BLOCK5/ VEE,AK,YSYM,DISPT,XST,ZST,AREAS,CB,N_POINTS,NSYM
      COMPLEX CGG,CGM
      CHARACTER*20 FILE_NAME
      CHARACTER*50 TITLE
**
  100 WRITE(1,
     +     '("*****************************************************"
     +    ,/,"**** Specify the Name of the Data File from which    ****"
     +    ,/,"**** this Program will Read the Input Data           ****"
     +    ,/,"**** File Name :$")')
      READ(1,'(A20)') FILE_NAME
**
      OPEN(UNIT=52,FILE=FILE_NAME,STATUS='OLD',ERR=200)
**
      READ (52) VEE,AK,YSYM
      READ (52) DISPT,XST,AREAS,CB
      READ (52) N_POINTS,NSYM
      READ (52) (U_KELVIN(I),I=1,N_POINTS)
      READ (52) (V_KELVIN(I),I=1,N_POINTS)
      READ (52) (W_KELVIN(I),I=1,N_POINTS)
**
      DO  I = 1,N_POINTS
        READ (52) CGG(I),CGM(I),FW,FFW
      END DO
**
*****  The varibles FW and FFW are not used by this program but they are.
*****  These variables should be left in the above statements
*****  for file compatibility.
**
      CLOSE (UNIT=52)
      RETURN
**
  200 WRITE(1,'(4/,5X,
     .      "*****************************************************",
     ./,5X,"****  ERROR --> File error occured in opening the    ****",
     ./,5X,"****            specified file. File does not exist.****",
     . /,5X,"****           Try again.                          ****",
     ./,5X,"*****************************************************")')
      GO TO 100
**
      END
**
```

```
*********************************************************************
*                                                                   *
      SUBROUTINE WRITE_FILE(FILE_NAME,TITLE)
*                                                                   *
*********************************************************************
**                                                                 **
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      CHARACTER*20 FILE_NAME
      CHARACTER*(*) TITLE
**
  100 WRITE(1,'(">>>>> ",A20," OUTPUT FILE    <<<<<")') TITLE
      WRITE(1,
     +       '("****************************************************"
     +     ,/,"**** Specify the Name of the Data File For Storing  ****"
     +     ,/,"**** The Output Results for Plotting                ****"
     +     ,/,"**** File Name :$")')
      READ(1,'(A20)') FILE_NAME
**
      OPEN(UNIT=53,FILE=FILE_NAME,STATUS='NEW',ERR=200)
**
**
      WRITE (53) (X_POINT(I),I=1,NX_POINT)
      WRITE (53) (Z_POINT(J),J=1,NZ_POINT)
**
      DO  I = 1,NX_POINT
        WRITE(53) (Y_POINT(I,J),J=1,NZ_POINT)
      END DO
**
      CLOSE(UNIT=53)
      RETURN
**
  200 WRITE(1,'(4/,5X,
     .       "*****************************************************",
     ./,5X,"****  ERROR --> File error occured in opening the   ****",
     ./,5X,"****            specified file. File already exists.****",
     ./,5X,"****            Try again.                          ****",
     ./,5X,"*****************************************************")')
      GO TO 100
**
      END
```

```
***********************************************************************
*                                                                     *
      SUBROUTINE KELVIN_AND_SEA
*                                                                     *
***********************************************************************
**                                                                   **
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      DIMENSION Y_VALUE(101)
      CHARACTER*20 FILE_NAME
**
  100 WRITE(1,
     +      '("*********************************************************"
     +    ,/,"**** Specify the Name of the Input File for the      ****"
     +    ,/,"**** KELVIN WAVE Results                             ****"
     +    ,/,"**** File  Name :$")')
      READ(1,'(A20)') FILE_NAME
      IFLAG=1
**
      OPEN(UNIT=53,FILE=FILE_NAME,STATUS='OLD',ERR=1000)
      REWIND 53
**
      READ(53) (X_POINT(I),I=1,NX_POINT)
      READ(53) (Z_POINT(J),J=1,NZ_POINT)
**
      DO  I = 1,NX_POINT
        READ(53) (Y_POINT(I,J),J=1,NZ_POINT)
      END DO
**
      CLOSE(UNIT=53)
**
      XMIN = X_POINT(1)
      ZMIN = Z_POINT(1)
      XMAX = X_POINT(NX_POINT)
      ZMAX = Z_POINT(NZ_POINT)
**
  200 WRITE(1,
     +      '("*********************************************************"
     +    ,/,"**** Specify the Name of the Input File for the      ****"
     +    ,/,"**** SEA STATE Results                               ****"
     +    ,/,"**** File  Name :$")')
      READ(1,'(A20)') FILE_NAME
      IFLAG=2
**
      OPEN(UNIT=54,FILE=FILE_NAME,STATUS='OLD',ERR=1000)
**
      READ(54) (X_POINT(I),I=1,NX_POINT)
      READ(54) (Z_POINT(J),J=1,NZ_POINT)
**
      DO  I = 1,NX_POINT
        READ(54) (Y_VALUE(J),J=1,NZ_POINT)
        DO  JJ = 1,NZ_POINT
         Y_POINT(I,JJ) = Y_POINT(I,JJ) + Y_VALUE(JJ)
        END DO
      END DO
```

```
**
      CLOSE(UNIT=54)
**
      IF(XMIN.NE.X_POINT(1)) GO TO 1010
      IF(ZMIN.NE.Z_POINT(1)) GO TO 1020
      IF(XMAX.NE.X_POINT(NX_POINT)) GO TO 1030
      IF(ZMAX.NE.Z_POINT(NZ_POINT)) GO TO 1040
**
      CALL WRITE_FILE(FILE_NAME,'KELVIN/SEA WAVE')
      GOTO 9999
**
1000  CONTINUE
      WRITE(1,'(4/,5X,
     .     "*******************************************************",
     ./,5X,"****  ERROR -->  File error occured in opening the  ****",
     ./,5X,"****             specified file. Try again.         ****",
     ./,5X,"*******************************************************")')
**
      IF(IFLAG.EQ.1) GOTO 100
      IF(IFLAG.EQ.2) GOTO 200
**
1010  WRITE(1,'(4/,5X,
     .     "*******************************************************",
     ./,5X,"****  ERROR -->  XMIN Does not match for the grid   ****",
     ./,5X,"****           from the KELVIN and SEA STATE files. ****",
     ./,5X,"*******************************************************")')
      GO TO 9999
1020  WRITE(1,'(4/,5X,
     .     "*******************************************************",
     ./,5X,"****  ERROR -->  ZMIN Does not match for the grid   ****",
     ./,5X,"****           from the KELVIN and SEA STATE files. ****",
     ./,5X,"*******************************************************")')
      GO TO 9999
1030  WRITE(1,'(4/,5X,
     .     "*******************************************************",
     ./,5X,"****  ERROR -->  XMAX Does not match for the grid   ****",
     ./,5X,"****           from the KELVIN and SEA STATE files. ****",
     ./,5X,"*******************************************************")')
      GO TO 9999
1040  WRITE(1,'(4/,5X,
     .     "*******************************************************",
     ./,5X,"****  ERROR -->  ZMAX Does not match for the grid   ****",
     ./,5X,"****           from the KELVIN and SEA STATE files. ****",
     ./,5X,"*******************************************************")')
**
9999  CONTINUE
      RETURN
      END
```

```
**
*************************************************************************
*                                                                       *
      SUBROUTINE SEA2D(IERROR)
*                                                                       *
*************************************************************************
**                                                                     **
*****  2-D SEA STATE = 1-D SEA STATE * COSINE SQUARED SPREADING.
*****
*****  1-D SEA STATE CALC USING (1)PIERSON-MOSKOWITZ,
*****  (2)JONSWAP, (3)NEUMANN, (4) BRETSCHNEIDER.
**
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,DZ,GRID_SIZE
      COMMON/BLOCK6/ WM(101),AZD(101),WZD(101),ENF(101),EN(161)
      COMMON/BLOCK7/ NSP,FNXP,ISEED,WVKT,THSTD,TZ,NF,IEDW
      DIMENSION FR(202),FS(161)
**
      CALL SEA_MENU
**
      NI = 10
      TZ = 10.0
       WVFT = 1.6878*WVKT
         HS = 0.0182*WVKT**2
         GR = 32.174
         PI = 3.141593
      TWOPI = 2.*PI
        OMZ = 0.0
        GAM = 0.0
**
      GOTO (100,200,300,400), NSP
**
*****  PIERSON-MOSKOWITZ SPECTRUM          Equation 9, Reference (1)
**
  100   A = 0.0081*GR**2
        B = 0.74*GR**4/WVFT**4
       IA = 5
       IB = 4
      GO TO 500
**
*****  JONSWAP SPECTRUM                    Equation 10, Reference (1)
**
  200   A = 0.0081*GR**2
        B = 0.74*GR**4/WVFT**4
       IA = 5
       IB = 4
      OMZ = (GR/WVFT)*(0.592)**(0.25)      ! Peak frequency
      GAM = 3.3                            ! Mean enhancement factor
      GO TO 500
**
*****  NEUMANN SPECTRUM                    Equation 11, Reference (1)
**
  300   A = 33.1*PI/4.
        B = 2.*GR**2/WVFT**2
       IA = 6
       IB = 2
      GO TO 500
```

```
**
*****  BRETSCHNEIDER SPECTRUM.
**
  400    A = 483.5*HS**2*(1./TZ)**4
         B = 1944.5*(1./TZ)**4
        IA = 5
        IB = 4
**
*****  Compute energy integral by Simpson's rule.
**
  500 CONTINUE
**
           J = 1
        SUMP = 0.0
        NINT = NI
  510     N2 = NINT/2
          FN = NINT
         SUM = 0.0
        DENI = 0.0098*OMZ**2
        DENR = 0.0098*OMZ**2
        DENL = 0.0098*OMZ**2
       TSOB1 = 0.0162*OMZ**2 - 0.000001
        SMSW = 0.0
**
      DO  I = 1,N2
         FI = I
         WMI = 0.01*TWOPI + 0.99*(2.*FI - 1.)*TWOPI/FN
         WMR = WMI + 0.99*TWOPI/FN
         WML = WMI - 0.99*TWOPI/FN
         FNI = (A/WMI**IA)*EXP( -B/WMI**IB)
         FNR = (A/WMR**IA)*EXP( -B/WMR**IB)
         FNL = (A/WML**IA)*EXP( -B/WML**IB)
         IF(NSP.NE.2) GO TO 530
         IF(WMR.LT.OMZ) GO TO 520
         IF(DENL.GT.TSOB1) GO TO 520
         IF(WMI.GT.OMZ) DENI = 0.0162*OMZ**2
         IF(WMR.GT.OMZ) DENR = 0.0162*OMZ**2
         IF(WML.GT.OMZ) DENL = 0.0162*OMZ**2
  520    FNI = FNI*GAM**(EXP( -(WMI - OMZ)**2/DENI))
         FNR = FNR*GAM**(EXP( -(WMR - OMZ)**2/DENR))
         FNL = FNL*GAM**(EXP( -(WML - OMZ)**2/DENL))
  530    CONTINUE
         SUM = SUM + (FNR + 4.*FNI + FNL)*(0.33*TWOPI/FN)
         EN(I) = SUM
         FS(I) = WMR/TWOPI
         SMSW = SMSW + SQRT(1.98*FNI*TWOPI/FN)
      END DO
**
      ER = ABS((SUM - SUMP)/SUM)
      IF(ER - 0.001) 560,540,540
  540 NINT = 2*NINT
      IF(NINT.GE.321 ) GO TO 550
      SUMP = SUM
      J = J + 1
      GO TO 510
**
```

```
  550 WRITE(1,'(">>>>>>>>>>  ERROR  <<<<<<<<<<")')
      WRITE(1,'("< Intervals in integral for E exceeded 321 >")')
      WRITE(1,'("< In subroutine SE7.2D  >")')
      IERROR = 321
      RETURN
**
  560 JI = J
      FNF = NF
**
*****  Compute ASW(IW) and WM(IW) based on equal energy intervals.
**
      DEN = 0.92*EN(N2)/FNF
      NF1 = NF + 1
      ENP = 0.05*EN(N2)
       JE = 1
**
      DO  LE = 2,N2
  570   IF(EN(LE) - ENP) 590,580,580
  580   FR(JE) = FS(LE-1) + ((FS(LE)-FS(LE-1))/(EN(LE)-EN(LE-1)))*
     1  (ENP - EN(LE-1))
        ENP = ENP + DEN
        JE = JE + 1
        IF(JE.GT.NF1) GO TO 600
        GO TO 570
  590   CONTINUE
      END DO
**
*****  Compute E,E2,E4,EP based on FLL and FUL cutoffs.
**
  600 IF(IEDW.GE.1) GO TO 610
**
      DO  IW = 1,NF
        WM(IW) = PI*(FR(IW) + FR(IW + 1))
        AZD(IW) = SQRT(DEN)
      END DO
**
      GO TO 660
**
  610 DELW = TWOPI*(FR(NF1) - FR(1))/NF
      WL = TWOPI*FR(1)
      JE = 1
      FSP = FR(1)
      WP = WL
**
      DO  LE = 2,N2
  620   IF(FS(LE) - FSP) 640,640,630
  630   ENF(JE) = EN(LE-1) + ((EN(LE)-EN(LE-1))/(FS(LE)-FS(LE-1)))*
     1  (FSP - FS(LE-1))
        FSP = FSP + (FR(NF1) - FR(1))/NF
        JE = JE + 1
        IF(JE.GT.NF1) GO TO 650
        GO TO 620
  640   CONTINUE
      END DO
```

```
**
  650 DO  IW = 1,NF
        WN = WP + DELW
        WM(IW) = 0.5*(WP + WN)
        WP = WN
        AZD(IW) = SQRT(ENF(IW+1) - ENF(IW))
      END DO
**
  660 DO  IW = 1,NF
        AZD(IW) = 1.4142136*AZD(IW)
        WZD(IW) = WM(IW)
      END DO
**
      CALL COSQSP(NF,IERROR)
**
      RETURN
      END
```

```
**
**********************************************************************
***                                                                  *
      SUBROUTINE COSQSP(NF,IERROR)
***                                                                  *
**********************************************************************
**                                                                  **
*****  2-D SEA STATE CALC USING (2*N/PI)*COS(N*THETA)**2.
**
      COMMON/BLOCK1/ X_POINT(101),Z_POINT(101),Y_POINT(101,101)
      COMMON/BLOCK4/ NX_POINT,NZ_POINT,NY_POINT,DX,[illegible],GRID_SIZE
      COMMON/BLOCK6/ WM(101),AZD(101),WZD(101),ENF(101),EN(161)
      COMMON/BLOCK7/ NSP,FNXP,ISEED,WVKT,THSTD,T2,NF0,IEDW
      DIMENSION TR(101),TH(101)
**
         NI = 10
        EPS = 1.0E-8
         GR = 32.174
         PI = 3.141593
      PIRAD = PI/180.0
      TWOPI = 2.*PI
        NTH = 9
      IF(FNXP.GE.40) NTH = 1
      IF(NTH.EQ.1) FNXP = 1
        NAF = NF
**
*****  Compute the Energy Integral by Simpson's Rule.
**
       SUMP = 0.0
       NINT = NI
          J = 1
  100    N2 = NINT/2
         FN = NINT
      TH(1) = -.5*PI/FNXP
      EN(1) = 0.0
        SUM = 0.0
      DO  I = 1,N2
         FI = I
        TMI = (2.*FI - 1. - .5*FN)*PI/(FNXP*FN)
        TML = (2.*FI - 2. - .5*FN)*PI/(FNXP*FN)
        TMR = (2.*FI - .5*FN)*PI/(FNXP*FN)
        FNI = COS(FNXP*TMI)**2
        FNR = COS(FNXP*TMR)**2
        FNL = COS(FNXP*TML)**2
        SUM = SUM + 0.666667*(FNR + 4.*FNI + FNL)/FN
        EN(I+1) = SUM
        TH(I+1) = TMR
      END DO
**
      ER = ABS((SUM - SUMP)/SUM)
      IF(ER - 0.001) 130,110,110
  110 NINT = 2*NINT
      IF(NINT.GE.321) GO TO 120
      SUMP = SUM
      J = J + 1
      GO TO 100
**
```

```
  120 WRITE(1,'(">>>>>>>>>>>  ERROR  <<<<<<<<<<<")')
      WRITE(1,'("< Intervals in integral for E exceeded 321 >")')
      WRITE(1,'("< In subroutine COSQSP >")')
      IERROR = 321
      RETURN
**
  130 JI = J
**
*****  Compute AZD(IW) and TM(IW) based on equal energy intervals.
**
      N2 = N2 + 1
      DEN = (1.0 - EPS)*EN(N2)/NTH
      NTH1 = NTH + 1
      ENP = 0.0
      JE = 1
**
      DO  LE = 2,N2
  140   IF(EN(LE) - ENP) 160,150,150
  150   TR(JE) = TH(LE-1) + ((TH(LE) - TH(LE-1))/(EN(LE) - EN(LE-1)))*
     1  (ENP - EN(LE-1))
        ENP = ENP + DEN
        JE = JE + 1
        IF(JE.GT.NTH1)  GO TO 170
        GO TO 140
  160   CONTINUE
      END DO
**
  170 INDEX = 2
      CALL SURFACE_GRID(INDEX)
**
      DO  IX = 1,NX_POINT
        DO  IZ = 1,NZ_POINT
          JT = 0
          IE2 = 2*ISEED
          CALL SSEED(IE2)
          ELEV = 0.0
**
          DO  IF = 1,NAF
            AK = WZD(IF)**2/GR
            AZ = AZD(IF)
**
              DO  IW = 1,NTH
                JT = JT + 1
                EPT = URAN(JT)*TWOPI
                TEF = 0.5*(TR(IW)+TR(IW+1))+THSTD*PIRAD
                AKX = AK*COS(TEF)
                AKZ = AK*SIN(TEF)
                COSXY = AKX*X_POINT(IX) + AKZ*Z_POINT(IZ) + EPT
                ELEV = ELEV + AZ*SQRT(DEN)*COS(COSXY)
              END DO
            END DO
**
          Y_POINT(IX,IZ) = ELEV
        END DO
      END DO
**
      RETURN
      END
```

```
**
***********************************************************************
*                                                                     *
      SUBROUTINE SEA_MENU
*                                                                     *
***********************************************************************
**
      COMMON/BLOCK7/ NSP,FNXP,ISEED,WVKT,THSTD,TZ,NF,IEDW
**
      WRITE(1,'(/)')
      WRITE(1,'("** Specify the Spectrum Type           "
     +      ,/,"** for the Sea State Calculations      "
     +      ,/,"** Input (1) for PIERSON-MOSKOWITZ      "
     +      ,/,"**       (2) for JONSWAP                "
     +      ,/,"**       (3) for NEUMANN                "
     +      ,/,"**       (4) for BRETSCHNEIDER          "
     +      ,/,"** Specify the Appropriate Spectrum : $")')
      READ(1,*) NSP
**
      WRITE(1,'(/)')
      WRITE(1,'("** For  60 Deg., Spreading index = 1.50 **")')
      WRITE(1,'("** For  90 Deg., Spreading index = 1.00 **")')
      WRITE(1,'("** For 120 Deg., Spreading index = 0.75 **")')
      WRITE(1,'("** Specify the Spreading Index (ge .5)  : $")')
      READ(1,*) FNXP
**
      WRITE(1,'(/)')
      WRITE(1,'("** Specify the Seed  Number"
     +        ,/,"** for Random Phase Angle (ISEED.GE.1)  : $")')
      READ(1,*) ISEED
**
      WRITE(1,'(/)')
      WRITE(1,'("** Specify Wind Speed  [S.S. 3 = 16 Kts]: $")')
      READ(1,*) WVKT
**
      IF(WVKT .GT.  0.0 .AND. WVKT .LT. 16.5) TZ = 7.5
      IF(WVKT .GE. 16.5 .AND. WVKT .LT. 21.5) TZ = 8.8
      IF(WVKT .GE. 21.5 .AND. WVKT .LT. 27.5) TZ = 9.7
      IF(WVKT .GE. 27.5 .AND. WVKT .LT. 47.5) TZ =12.4
      IF(WVKT .GE. 47.5 .AND. WVKT .LT. 55.5) TZ =15.0
      IF(WVKT .GE. 55.5 .AND. WVKT .LE. 63.0) TZ =16.4
      IF(WVKT .GT. 63.0 )                     TZ =20.0
**
      WRITE(1,'(/)')
      WRITE(1,'("** Specify the Wind Direction "
     +        ,/,"** with respect to the X-axis in deg.   : $")')
      READ(1,*) THSTD
      IF(NSP.LT.4) GO TO  50
**
```

```
      WRITE(1,'(/)')
      WRITE(1,'("** Most Probable Modal Wave"
     +        ,/,"** Period is :",F5.1," Sec.")') TZ
      WRITE(1,'(/)')
      WRITE(1,'("** Specify the Desired Modal Wave "
     +        ,/,"** Period in Seconds     : $")')
      READ(1,*) TZ
**
   50 WRITE(1,'(/)')
      WRITE(1,'("** Specify the Number of Wave "
     +        ,/,"** Frequencies (1 to 101)   : $")')
      READ(1,*) NF
**
      WRITE(1,'(/)')
      WRITE(1,'("** Input ZERO (0) for Equal ASW       "
     +        ,/,"** or ONE (1) for Equal Delta Freq   : $")')
      READ(1,*) IEDW
**
      RETURN
      END
```