NOAA Technical Memorandum ERL GLERL-51

A TWO-DIMENSIONAL LAKE WAVE PREDICTION SYSTEM

David J. Schwab
John R. Bennett
Edward W. Lynn

Great Lakes Environmental Research Laboratory
Ann Arbor, Michigan
May 1984

UNITED STATES DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
John V. Byrne, Administrator

Environmental Research Laboratories
Vernon E. Derr Director

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOM Environmental Research Laboratories. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

CONTENTS

FIGURES

TABLE

A TWO-DIMENSIONAL LAKE WAVE PREDICTION SYSTEM*

David J. Schwab, John R. Bennett, and Edward W. Lynn

This report describes a series of computer programs that can be used to forecast wave height, period, and direction for any part of the Great Lakes. The programs only require the user to specify the overlake wind forecast.

1. INTRODUCTION

Wave forecasts on the Great Lakes have been automatically produced by the National Weather Service since 1974 with a procedure developed by Pore (1974). This system uses statistical (Model Output Statistics or MOS) wind forecasts as input to Bretschneider's empirical equations (Bretschneider, 1970) for forecasting wave height. Twice daily, forecasts are made at 64 selected locations around the lakes; each extends to 36 h. Local wave forecasters have found these forecasts to be useful, but some have expressed interest in a system that would allow interaction between them and the wave forecast program. In this way, a range of wind forecasts could be used to generate a range of wave forecasts or local modifications could be made to the automated wind forecasts.

Recently, a two-dimensional numerical wave prediction model was developed for use on the Great Lakes and tested against 2 months of observed wave data in Lake Erie by Schwab et at. (1984). The model is a parametric type that numerically solves a local momentum balance equation on a numerical grid covering the lake. Input is overlake wind at each point. The model provides estimates of wave height, period, and direction at each grid point. Model results for hourly values of wave height and direction agreed very well with measurements taken at a wave research tower in the eastern part of Lake Erie. Model results for wave period agreed satisfactorily with measurements taken at the tower. Wave height measurements from the NOAA Data Buoy Center (NDBC) Nomad buoy in the western part of the lake were consistently higher than the model results, although the correlation was high. Wave period measurements from the buoy agreed satisfactorily with the model results.

This model was also evaluated for operational use in Canada by Clodman (1983). He tested an earlier version of the model in several idealized and real cases. His conclusion was that the model would be useful for operational forecasting if its application was limited to deep water.

The purpose of this report is to describe a series of computer programs that use the above-described numerical model to implement an interactive wave forecast system. The forecaster specifies wind speed and direction for a specified number of 12-h periods and the programs compute wave height, direction, and period at each grid point for the entire forecast period. The

---

*GLERL Contribution No. 406.

calculated wave height is equivalent to the traditional 'significant' wave height, the average height of the one-third highest waves. Wave period is the period corresponding to the spectral frequency with the highest energy in the wave spectrum. Wave direction is the direction of the wave momentum vector. The forecaster can then examine the results in a variety of formats, including two-dimensional maps of wave height, direction, or period at a given time or time series of wave height, direction, and period at a given location. This report includes a brief description of the numerical method, plots of the grids covering the lakes, an annotated sample run of the wave prediction system (appendix A), and the FORTRAN source code for the computer programs (appendix B). It is intended to serve both as a user's manual for the system and documentation of the programs for adaptation on other computer systems. Users intending to access the system on the GLERL computer should contact the authors for the proper telephone communications procedure.

2. THE NUMERICAL MODEL

The numerical wave forecast model is based on one originally proposed by Donelan (1977). Schwab *et* at. (1984) modified the model to conform to the GLERL two-dimensional lake circulation modeling system (Schwab et *al.*, 1981; Bennett *et* at., 1983; Schwab and Sellers, 1980). The version used here is a further simplification of the Donelan (1977) model to make it more useful for operational forecasting.

The basic model equations relate the time rate of change of wave momentum and the divergence of the wave momentum flux to input from the wind as follows:

$$\frac{\partial M_x}{\partial t} + \frac{\partial T_{xx}}{\partial x} + \frac{\partial T_{xy}}{\partial y} = \frac{\tau_x^w}{\rho_w}$$

$$\frac{\partial M_y}{\partial t} + \frac{\partial T_{yx}}{\partial x} + \frac{\partial T_{yy}}{\partial y} = \frac{\tau_y^w}{\rho_w}. \qquad (1)$$

The x and y momentum components are

$$M_x = g \int_0^\infty \int_0^{2\pi} \frac{F(f,\theta)}{c(f)} \cos\theta \, d\theta \, df$$

$$M_y = g \int_0^\infty \int_0^{2\pi} \frac{F(f,\theta)}{c(f)} \sin\theta \, d\theta \, df. \qquad (2)$$

Here $F(f,\theta)$ is the wave energy spectrum as a function of frequency, f, and direction, 8. C(f) is the phase speed. Schwab *et* at. (1984) show that, if we assume that deep water linear theory applies and that wave energy is distributed about the mean wave direction as cosine squared, the momentum fluxes can be expressed as

$$T_{xx} = g\left(\frac{\sigma^2}{4}\cos^2\theta_o + \frac{\sigma^2}{8}\right)$$

$$T_{xy} = T_{yx} = g\left(\frac{\sigma^2}{4}\cos\theta_o \sin\theta_o\right)$$

$$T_{yy} = g\left(\frac{\sigma^2}{4}\sin^2\theta_o + \frac{\sigma^2}{8}\right),$$

where σ^2 is the variance

$$\sigma^2 = \int_0^\infty E(f)df.$$

They further provide relationships between group velocity (or peak energy frequency), variance, and momentum that allow (1) and (2) to be solved for M_x, M_y, and σ once the right-hand side is specified. The right-hand side we use here is

$$\frac{\vec{\tau}^w}{\rho_w} = 0.028\ Df\ \left|\vec{u} - 0.83\ \vec{C}_p\right|(\vec{u} - 0.83\ \vec{C}_p),$$

where Df is a form drag coefficient defined as $Df = [0.4/\ln(50/\sigma)]^2$ with σ in meters. The factor 0.028 is the empirical fraction of the stress that is retained by the waves. The important features of the momentum input formula are that the drag coefficient increases for higher waves and that the momentum input depends on the square of the difference between wind speed and 0.83 times wave group velocity.

The main assumption in the present formulation of the model that would be of concern to a forecaster is that the waves are assumed to be governed by linear deep water theory. This means that shoaling, refraction, and breaking are ignored so that, even in the grid boxes nearest shore, the waves are assumed to be unaffected by finite water depth. In practice, this is not a severe limitation in most of the lakes, but may be of some importance in Lake St. Clair, certain parts of Lake Erie, Saginaw Bay, and Green Bay.

The only difference between the formulation detailed in Schwab et *al.* (1984) and the one used in the computer program WAVE in appendix B is that the "fossil" wave field (the wave field remaining from the previous storm) has been removed in the present version. We found that including the "fossil" field had only a small effect on the results. The computation time is doubled when the "fossil" field is included and we felt that the small potential improvement in accuracy (see section 6) did not justify the increased computational effort.

3. DESCRIPTION OF TEE PROGRAMS

There are three computer programs in the wave prediction system. The first program obtains wind information from the forecaster. The second program is the numerical wave model. The third program allows the forecaster to examine the results of the wave model in a variety of formats. The programs are written in FORTRAN 77. A listing of the programs is included in appendix B.

The first program, called INTFCA, asks for the name and agency of the user, the start date and time of the forecast, and wind direction and speed forecasts for a 12-h period. The date is used only for putting a time tag on wind prompting messages and output displays. The program checks for valid month (1-12), day, year (1900-99), and hour (0-23). The user must choose one of the following time intervals between wind forecasts: 1, 2, 3, 4, or 6 h. The program prompts for wind direction and speed at the chosen interval for a period of 12 h from the starting time. Wind direction is in degrees (o-360; 0 = wind from the north) and wind speed is in knots (o-100). If the user wants to use linear interpolation for wind direction and speed, instead of typing the wind direction and speed, the "/" (slash) character should be entered. After 12 h of wind data have bee" entered, the user is given the option of extending the run by another 12 h. A maximum of 10 days of wind data with a l-h interval or 60 days with a 6-h interval can be accomodated. After all wind data are typed in, a table of wind speed and direction is produced and the user is given the opportunity to change any erroneous values. The program then writes meteorological data records to a disk file for use by the numerical wave model.

The second program, WAVE, is the numerical wave model. It reads the meteorological data produced by INTFCA and calculates wave height, direction, and period for each hour of the run, storing the results on FORTRAN unit numbers 11, 12, and 13, respectively. Note that, although the program has a provision for setting an initial condition for the wave field, we chose not to use this feature for the operational wave prediction system, mainly because there is no simple way to incorporate a nonzero initial condition. Therefore the model starts with a calm sea surface. It is advisable to begin the prediction at least 12 to 24 h previous to the critical forecast time to give the waves a chance to grow past the point at which the unrealistic initial condition has any effect.

The final program, DSPLAY, allows the user to examine the results of the calculation either in the form of a time series at a specified location in the lake or as a map of wave height, period, or direction at a specified time. For time series data, the program checks whether the specified location is within the computational grid. If it is not, an error message is printed and the program asks for another location. If the user elects to have maps printed, a list of times for which maps are required may be entered. The user can leave the terminal while the maps print out. Finally, the program asks for and records any comments the user may wish to include.

4. COMPUTATIONAL GRIDS

The computational grids used by the model for Lakes Superior, Michigan, Huron, St. Clair, Erie, and Ontario are shown in figures 1-6. The upper lakes have 15-km grids, the lower lakes 10-km grids, and Lake St. Clair a 5-km grid. Isle Royale and Michipicoten Island are included in the Lake Superior grid. Beaver Island in Lake Michigan, Manitoulin Island in Lake Huron, and Pelee Island in Lake Erie are also included. The figures show the relation between the artificial computational grid boundary and the actual shoreline of the lake. The forecaster should be aware of the difference between the two shorelines because some areas of the real lake are not part of the computational grid and the wave prediction system will not produce results for points in these areas. The grid sizes for the various lakes were chosen as an acceptable compromise between realistic resolution of the shoreline and computational speed. The computing time involved in the numerical model increases very rapidly with decreasing grid size.

5. WIND INPUT

The wave prediction system asks for values of wind direction and speed at the chosen interval of 1, 2, 3, 4, or 6 h. If some values are not available, the programs will interpolate between values. Although the program has the

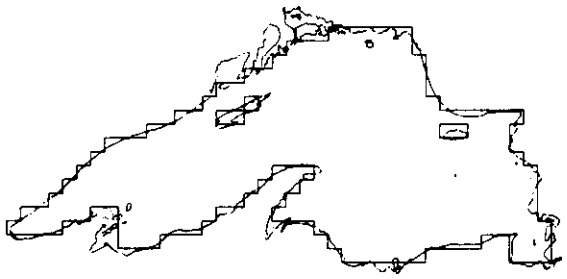

FIGURE 1.--*The 15-km* grid *for Lake Superior.*

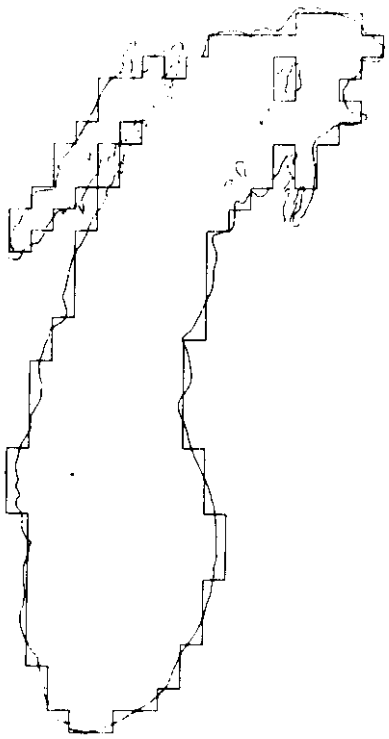

FIGURE 2.--*The 15-km grid for Lake Michigan.*

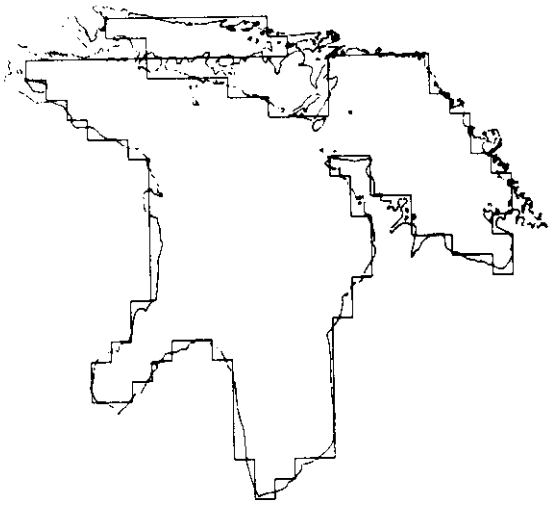

FIGURE 3.--*The 15-km grid* for *Lake Huron.*

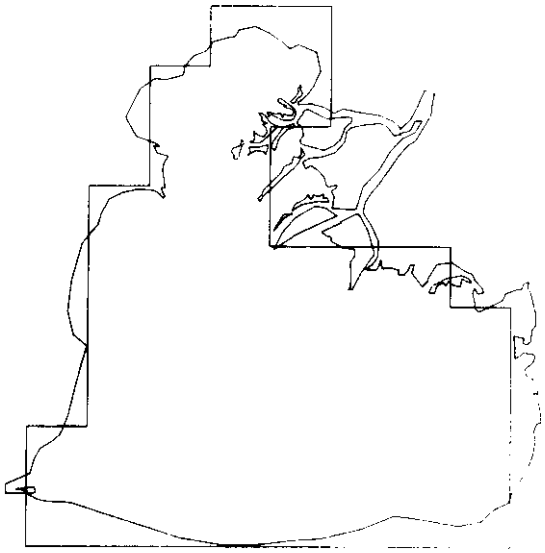

FIGURE 4.--*The S-km grid for Lake St. Clair.*

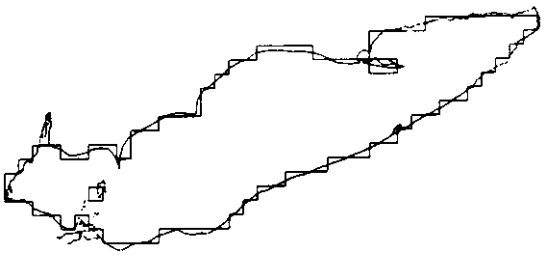

FIGURE 5.--*The 10-km grid for Lake Erie.*

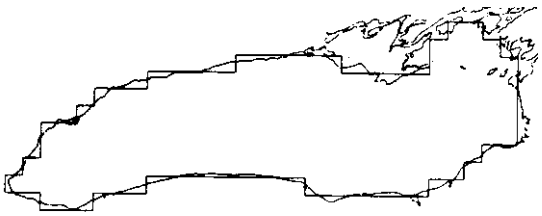

FIGURE 6.--*The 10-km grid* for *Lake Ontario.*

capability to incorporate spatially variable wind fields, the operational version uses a uniform wind over the entire lake. It is suggested that, if forecasts are to be made for two widely separated points on the same lake where wind conditions are expected to be quite different, two runs of the system be made, one for each set of winds.

The program also has the capability to adjust wind measurements for height and stability, but again these features are not used in the operational model. We have assumed that the winds provided by the forecaster are representative overwater winds at 5 m (16 ft) above the water surface under neutral conditions (air temperature equal to water temperature). If the available wind measurements are from a different height above the water, or the winds are overland winds, or the overlake air temperature is quite different from

the water temperature, the wind speed input to the wave prediction system should be adjusted accordingly. That is, measurements from less than 5 m should be increased, measurements from greater than 5 m should be decreased, and overland winds should generally be increased. (See Schwab and Morton, 1984.)

6. MODEL TESTS

Donelan (1977) compares results of an early version of the wave prediction model, run with wind observed at meteorological buoys in Lake Ontario, to wave height observed by three Waverider buoys in the lake. The hindcast only covers a 2-day period in August 1972, but the comparison was very encouraging. Clodman (1983) tested the model for several idealized cases and examined the differences between results from the two-dimensional, time-dependent numerical model and several empirical wave forecasting methods. His conclusion was that for the case of a steady wind "probably our model gives results a little too low for low wind speeds and high for high wind speeds, but one cannot say for certain since no absolute standard of accuracy exists."

Schwab et *al.* (1984) provide a much more comprehensive test of the model. They compare results from the model (driven by wind observed at a research tower 6-km offshore in the eastern part of Lake Erie and at the NDBC Nomad buoy in the western part of the lake) to hourly observations of wave height and period at the tower and the buoy and wave direction at the tower for the entire months of September and October 1981. The model results agreed very well with wave height measurements at the tower (correlation coefficient 0.93, standard error 20 cm). Wave period agreement at the tower was satisfactory, although the model generally gave a lower estimate of wave period than was observed. Both the model results and the observations showed a surprising systematic deviation of the wave direction at the tower from the wind direction for certain wind directions. This deviation could not be attributed to refraction effects. At the NDBC buoy, wave period results were consistent with those at the tower, although the modeled periods were generally lower than the observed periods. Wave height measurements at the buoy were consistently higher than the modeled wave heights, although the correlation was high (correlation coefficient 0.88). The reason for the consistent discrepancy at the buoy was not clear; it is currently being investigated.

Two additional runs of the wave model have been made with the 1981 Lake Erie data set. Both runs use the version of the numerical model given in appendix B that does not contain the "fossil" wave field. One run uses the same S-km grid as Schwab et *al.* (1984), and the second uses the 10-km grid described in section 4 (fig. 5). The model results for wave height at the tower and buoy are compared with the results of Schwab et *al.* (1984) in figure 7 and table 1. It can be seen that removing the "fossil" wave field from the computations had a very insignificant effect on the results and that the 10-km grid gives nearly as accurate a result as the 5-km grid. Based on these results. we feel that in a real forecast situation uncertainties in the forecast wind will remain the main source of error in the wave forecasts, not grid resolution or model inadequacies.

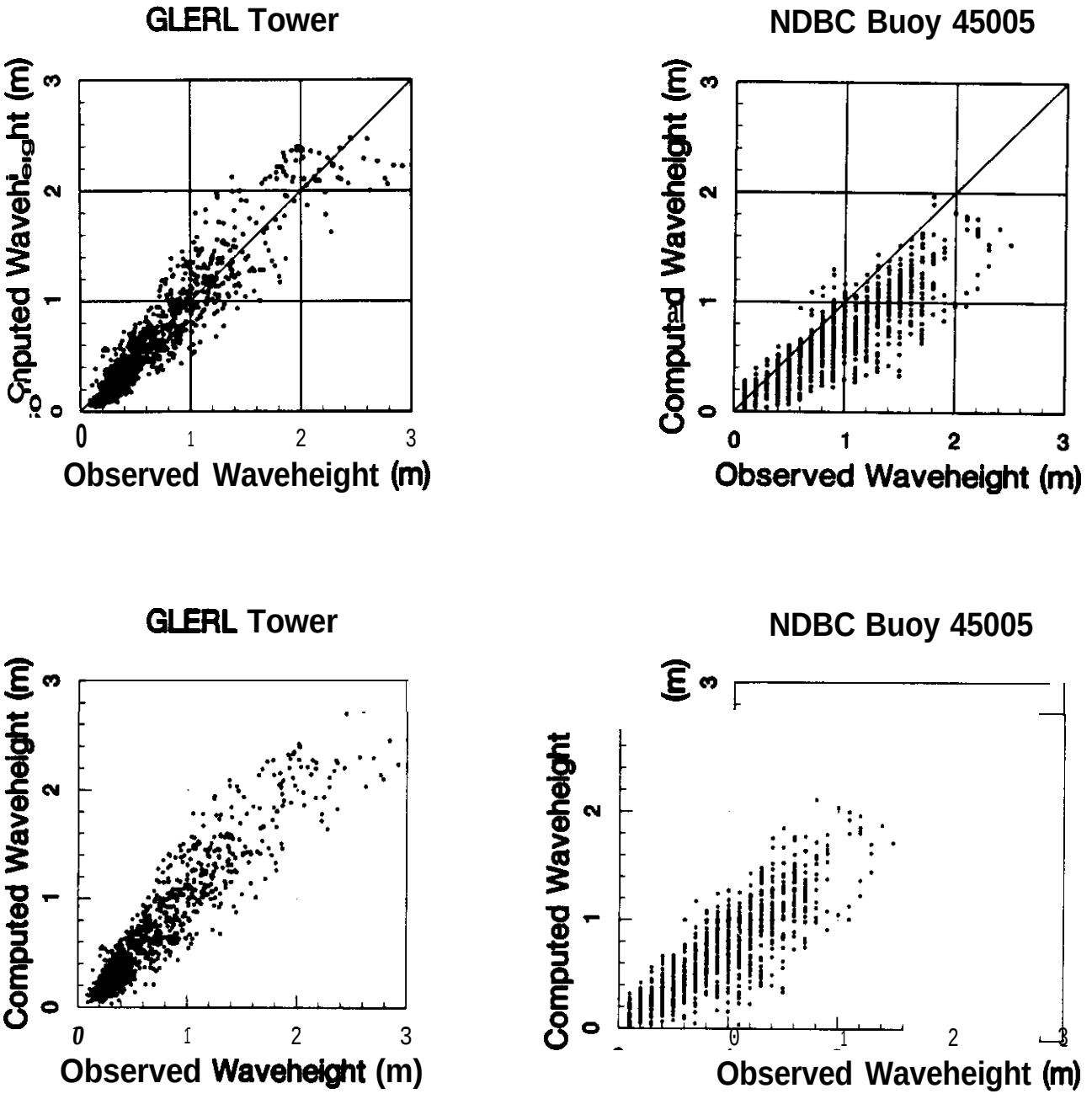

FIGURE 7.--*Comparison of computed and observed wave height at the GLERL tower and at NDBC 45005 in Lake Erie* for *September and October 1981. Tke upper* figures are from *Sckwb et* al. *(1984) and correspond to a S-km grid with "fossil" wve field* included in *the computations. The lower* figures are for a S-km grid *with* no *"fossil" wave field.*

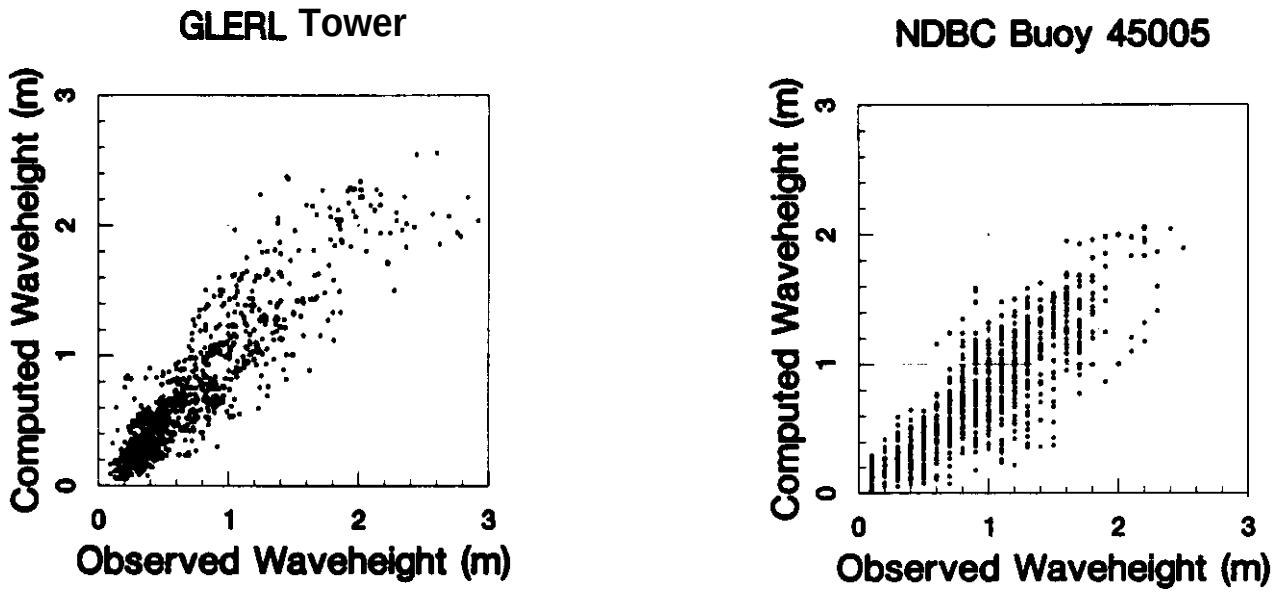

FIGURE 7.(cont.)--*Comparison* of ***computed and observed wave height at the GLERL tower and at NDBC 45005 in Lake Erie for September and October 1981. The figures here are*** for a *10-km grid with no "fossil" wave* field.

TABLE 1.--***Statistics from comparisons of observed and*** model ***wave heights in Lake Erie*** for ***September and October 1981***

	Wave height at tower			Wave height at NDBC 45005		
	(1)	(2)	(3)	(1)	(2)	(3)
Correlation coefficient	0.93	0.93	0.90	0.88	0.87	0.87
Root-mean-square error	0.20 m	0.21 m	0.24 m	0.33 m	0.30 m	0.28 m
Slope	0.99	0.99	0.99	0.68	0.76	0.79
Intercept	-0.00 m	-0.01 m	0.04 m	0.03 m	0.01 m	0.01 m
Standard error	0.20 m	0.21 m	0.24 m	0.17 m	0.20 m	0.20 m

(1) Results from Schwab *et al.* (1984) with a 5-km grid and "fossil" included.
(2) Results with a 5-km grid, no "fossil."
(3) Results with a 10-km grid, no "fossil."

7. REFERENCES

Bennett, J. R., A. H. Clites, and D. J. Schwab (1983): *A two-dimensional lake circulation modeling system: Programs to compute particle trajectories and the motion of dissolved substances.* NOAA Technical Memorandum ERL GLERL-46. National Technical Information Service, Springfield, Va. 22161. 51 pp.

Bretschneider, C. L. (1970): Forecasting relations for wave generation. *Look Lab, Hawaii, Vol. 1, No. 3,* pp. 31-44.

Clodman, S. (1983): Testing and future *applications* of *the Donelan numerical lake wve model.* Atmospheric Environment Service. Meteorological Services Branch, International Report MSRB-83-4. Atmospheric Environment Service, Downsview, Ont. 20 pp.

Donelan, M. A. (1977): A simple numerical model for wave and wind stress prediction. Unpublished manuscript. National Water Research Institute, Burlington, Ont. 28 pp.

Pore, N. A. (1974): Great Lakes wave forecasts. National Weather Service, Technical Procedures Bulletin no. 127. National Weather Service, Silver Spring, Md. 9 pp.

Schwab, D. J., and D. L. Sellers (1980): *Computerized bathymetry and shore*lines of *the Great Lakes.* NOAA Data Report ERL GLERL-16. National Technical Information Service, Springfield, Va. 22161. 13 pp.

Schwab, D. J., J. R. Bennett, and A. T. Jessup (1981): *A two-dimensional take circulation modeling system.* NOAA Technical Memorandum ERL GLERL-38. National Technical Information Service, Springfield, Va. 22161. 79 pp.

Schwab, D. J., J. R. Bennett, P. C. Liu, and M. A. Donelan (1984): Application of a simple numerical wave prediction model to Lake Erie. *J. Geophys. Res.* (In *press.)*

Schwab, D. J., and J. A. Morton (1984): Estimation of overlake wind speed from overland wind speed: A comparison of three methods. *J. Great Lakes Rea.* 10(1):68-72.

Appendix A.--ANNOTATED SAMPLE RUN OF THE WAVE PREDICTION SYSTEM

This appendix contains a sample run of the wave prediction system on the GLERL computer. User responses in the sample are underlined, but in an actual run they are not. All characters in user responses are ***upper case*** and all responses end with a carriage return. Numbers on the right side of the page refer to the footnotes below.

GREAT LRKES ENVIRONMENTRL RESERRCH LABORATORY VAX11/780 (1)

Username: WAVES (2)

GREAT L W E S ENVIRONMENTAL RESEARCH LABORATORY
WAVE PREDICTION SYSTEM, REVISION 2.0 2/84

THIS PROVISIONAL VERSION OF THE GLERL WAVE PREDICTION
SYSTEM IS AVAILABLE FOR TESTING AND EVALUATION PURPOSES.
WE WWLD APPRECIATE ANY COMMENTS OR SUGGESTIONS, GOOD
OR BAD. THANK YOU!

DAVE SCHWAB FTS 378-2120 COMMERCIAL (313) 668-2120

ENTER LAKE NAME: ERIE (3)
PLEASE ENTER YOUR NAME FIND AGENCY (E.G. MARVIN MILLER/NWS)
MARVIN MILLER / NWS CLEVELAND (4)
DO YOU WANT DETAILED INSTRUCTIONS ABOUT HOW TO
USE THIS PROGRAM ? (INPUT YES... OR... NO)
NO (5)
INPUT STARTING DATE FIND TIME AS MONTH, DAY, YEAR, HOUR (0-23).
THIS WILL BE HOUR 0 (ZERO) (E.G. 7, 28, 1983, 14)
4. 15, 1984, 6
INPUT TIME IN HOURS BETWEEN WIND DATA POINTS (1, 2, 3, 4 OR 6)
6 (6)
INPUT WINDS (DIRECTION, SPEED) IN DEGREES FIND KNOTS (E.G.. 270, 10)
TYPE '/' FOR LINEAR INTERPOLATION
INPUT WIND FIT 0 HOURS 4/15/1984 6:00
220.10
INPUT WIND FIT 6 HOURS 4/15/1984 12:00
245.29
INPUT WIND AT 12 HOURS 4/15/1984 18:00
290.35
IS A FORECAST LONGER THEN 12 HOURS DESIRED?
(INPUT YES.. . OR.. . NO)
YES (7)
FORECAST WILL BE EXTENDED TO 24 HOURS
INPUT WIND FIT 18 HOURS 4/16/1984 0:00
300.25
INPUT WIND FIT 24 HOURS 4/16/1984 6:00
/ (8)
IS A FORECAST LONGER THEN 24 HOURS DESIRED?
(INPUT YES.. . OR.. . NO)
YES (9)

FORECAST WILL BE EXTENDED TO 36 HOURS
INPUT WIND FIT 30 HOURS 4/16/198412:00
45,15
INPUT WIND FIT 36 HOURS 4/16/198418:00
90,10
IS A FORECAST LONGER THEN 36 HOURS DESIRED?
(INPUT YES.. . OR.. . NO)
NO (10)

HOUR	DATE	TIME	WIND DIR.	WIND SPEED
0	4/15/1984	6:00	220	10.0
6	4/15/1984	12:00	245	25.0
12	4/15/1984	18:00	290	35.0
18	4/16/1984	0:00	300	2s. 0
24	4/16/1984	6:00	352	20.0
30	4/16/1984	12:00	4s	1s. 0
36	4/16/1984	18:00	90	10.0

(11)

IS WIND DATA O.K.? (INPUT YES.. . OR.. . NO)
YES
CALCULATING WAVES. . . PLEASE STAND B Y
ENTER H IF YOU WANT TO SEE WAVE HEIGHT MAP
ENTER D IF YOU WANT TO SEE WAVE DIRECTION MAP
ENTER P IF YOU WANT TO SEE WAVE PERIOD MAP
ENTER T IF YOU WANT A TIME SERIES OF WAVE HEIGHT,
DIRECTION AND PERIOD FIT A GIVEN LOCATION
ENTER Q IF YOU WANT TO QUIT
T (12)
INPUT LATITUDE OF POINT AT WHICH YOU WANT TO SEE TIME SERIES
IN DEG, MIN, SEC (E.G. 44,06,00)
OR IN DECIMAL DEGREES (E.G. 44.1,0,0)
41.6.0.0 (13)
INPUT LONGITUDE OF POINT AT WHICH YOU WANT TO SEE TIME SERIES
IN DEG, MIN. SEC (E. G . 84.18.30)
OR IN DECIMAL DEGREES (E.G. 84: 35; 0,0)
81,45,0 (14)
LATITUDE 41 DEG 36.0 MIN
LONGITUDE 81 DEG 45.0 MIN

HOUR	DATE	TIME	WAVE HEIGHT (FT)	WAVE DIR. (DEG)	WAVE PERIOD (SEC)
6	4/15/1984	12:00	2.7	252.	3.6
12	4/15/1984	18:00	8.5	268.	6.5
18	4/16/1984	0:00	9.3	290.	6.7
24	4/16/1984	6:00	4.9	289.	5.2
30	4/16/1984	12:00	2.0	329.	4.0
36	4/16/1984	18:00	2.1	17.	3.6

ENTER H IF YOU WANT TO SEE WAVE HEIGHT MAP
ENTER D IF YOU WANT TO SEE WAVE DIRECTION MAP
ENTER P IF YOU WANT TO SEE WAVE PERIOD MAP
ENTER T IF YOU WANT A TIME SERIES OF WAVE HEIGHT,
DIRECTION AND PERIOD AT A GIVEN LOCATION
ENTER Q IF YOU WANT TO QUIT
H (15)

ENTER THE TIME(S) IN HOURS (6-36) AFTER THE START TIME
4/15/1984 6:00 FIT WHICH YOU WANT TO SEE WAVE HEIGHTS
YOU MAY MAKE UP TO 25 ENTRIES, ONE ENTRY PER LINE
END THE LIST WITH -1
18
24
-1

(16)

WAVE HEIGHTS IN TENTHS OF FEET FIT 18 HOURS 4/16/1984 0:00

WAVE HEIGHTS IN TENTHS OF FEET FIT 24 HOURS 4/16/1984 6:00

```
ENTER H IF YOU WANT TO SEE WAVE HEIGHT MAP
ENTER D IF YOU WFINT TO SEE WAVE DIRECTION MAP
ENTER P IF YOU WFINT TO SEE WAVE PERIOD MAP
ENTER T IF YOU WANT A TIME SERIES OF WAVE HEIGHT,
 DIRECTION AND PERIOD FIT A GIVEN LOCATION
ENTER Q IF YOU WFINT TO QUIT
D                                                                    (17)
ENTER THE TIME(S) IN HOURS (6-36) AFTER THE START TIME
 4/15/1984 6:00 AT WHICH YOU WFINT TO SEE WAVE DIRECTIONS
YOU MAY MAKE UP TO 25 ENTRIES, ONE ENTRY PER LINE
END THE LIST WITH -1
18
-1                                                                   (18)
WAVE DIRECTIONS IN TENS OF DEGREES FIT| a HOURS 4/16/1984 0:00
**********************************************************************************
**********************************************************************************
**********************************************************************************
************************************************************ 25 25 25 25*********
****************************************************** 26 26 25 25 25 25 25 26 26
**************************************************** 26 26 26 26 26 26 26 26 26 26
********************************************** 27 26 26 26 26 26 26 26 26 26 26 26
******************************************** 25 25 26 26 26 26 26 26 26 26 26 26
************************************ 24 25 26 26 26 26 26 26 26 26 26 26 26 26 27
******************************** 27 28 27 26 26 26 26 26 26 26 26 26 26 26 27 27 27
********* 29 28****** 29 23*** 25 25 27 20 20 20 20 20 27 27 27 27 27 27 27 27 27
****** 24 26 27 29 30 20 27 26 24 23 25 28 28 28 28 28 28 28 28 20 20 28******
*** 29 28 27 26 26 27 28 29 30 31 32 29 27 26 26 26 26 26 27 27***************
*** 29 29 29 29 29 29*** 30 30 30 29 28 27 27 26 26 26 26**********************
********* 31 31 30 30 30 30 30 29 29 29 29 29 29 29  29************************
*************** 31*** 31 31 31 30 30 30 29 29 29 29*****************************
************************ 32 31 31 30*****************************************
**********************************************************************************
*******************************************
*************** 23 24 24 24 25 25 2 6 26***
*** 29 29 28 28 27  26  25  25 25 25 26******
*** 30 29 29 29 20 27 26 26 26 26*********
 26****** 24 24 24 25 25 26 27***********
 26 26 26 26 25 25 25 25 26**************
 26 26 26 26 26 26 26 26*****************
 26 26 27 27 27 27************************
 27 27 27 27******************************
 27 27 27*********************************
 27***************************************
*******************************************
*******************************************
*******************************************
*******************************************
*******************************************
*******************************************
*******************************************
```

```
ENTER H IF YOU WANT TO SEE WAVE HEIGHT MAP
ENTER D IF YW WANT TO SEE WAVE DIRECTION MAP
ENTER P IF YOU WANT TO SEE WAVE PERIOD MAP
ENTER T IF YOU WANT A TIK SERIES OF WAVE HEIGHT,
 DIRECTION AND PERIOD AT A GIVEN LOCATION
ENTER Cl IF YOU WANT TO QUIT
Q                                                              (19)
PLEASE TYPE AS MANY LINES AS YOU WISH IF YW HAVE ANY
COMMENTS OR SUGGESTIONS CONCERNING THIS WAVE MODEL
END WITH A CARRIAGE RETURN
I HAVE NO COMMENTS AT THIS TIME                                (20)

GOODBYE !

  WAVES      logged out at 27-FEB-1984 14:24:05.23
```

Footnotes:

(1) When carrier detect is established, press the return key to get initial system message.

(2) After initial system message, type "WAVES" in response to "Usernumber:" prompt.

(3) In response to "ENTER LAKE NAME:" prompt, enter "SUPERIOR", "MICHIGAN", "HURON", "ST. CLAIR", "ERIE", or "ONTARIO".

(4) Name and agency are used to keep track of who's using the model.

(5) Detailed instructions contain a condensed version of sections of this report.

(6) The forecaster chooses a 6 hour time increment.

(7) The forecast is extended by 12 hours.

(8) The program will interpolate wind direction and speed at this hour from previous and succeeding values.

(9) The forecast is extended by 12 more hours.

(10) The forecast stops at 36 hours.

(11) Note interpolated values of wind direction and speed for hour 24 (halfway between 18 and 30 hour values).

(12) Time series is selected.

(13) Latitude is specified in decimal degrees followed by zeroes for minutes and seconds.

(14) Longitude is specified in degrees and minutes.

(15) Wave height map is selected.

(16) Map will be printed for hours 18 and 24.

(17) Wave direction map is selected.

(18) Map will be printed for hour 18.

(19) User elects to terminate session.

(20) Comments here would be appreciated.

Appendix B.--SOURCE CODE FOR FORTRAN COMPUTER PROGRAMS INTFCA, WAVE, AND DISPLAY

```
      PROGRAM INTFCA
C************************ PROGRAM INTFCR ******************************
C
C PURPOSE: THE PURPOSE OF PROGRAM INTFCA IS TO PROVIDE FIN INTERFACE
C   BETWEEN A TIME-SHARING USER AND THE WAVE PREDIC-
C   TION SYSTEM. THE USER IS REQUIRED TO SUPPLY THE STRRTING DATE AND
C   TIME FOR PREDICTION AND METEOROLOGICAL DATA FOR THE NEXT 12 HOURS
C   AT HOUR INTERVALS SPECIFIED BY THE USER. RNY NUMBER
C   OF ADDITIONAL 12 HOUR PERIODS MAY BE SPECIFIED (UP TO 10 DAYS).
C   PROGRAM INTFCA PRODUCES A CONTROL FILE RND A METEOROLOGICAL
C   DATA FILE FOR THE WAVE PREDICTION MODEL.
C
C INPUT:
C   LOGICAL UNIT 5:
C    TIME-SHARING USER INPUT
C   LOGICAL UNIT 7:
C    BATHYMETRIC DATA FILE
C
C OUTPUT:
C   LOGICAL UNIT 6:
C    PROMPTING FOR TIME-SHARING USER
C   LOGICAL UNITa :
C    METEOROLOGICRL DATA FILE FOR PROGRAM WAVE
C   LOGICAL UNIT 10:
C    CONTROL FILE FOR PROGRAM WAVE
C   LOGICAL UNIT 15:
C    GENERAL INFORMATION FILE
C
C SUBROUTINES:
C   HELPER - THIS SUBROUTINE GIVES DETAILED INSTRUCTIONS ABOUT HOW
C     TO USE THE WAVE PREDICTION SYSTEM.
C   RDATE - THIS SUBROUTINE CALCULATES THE RELATIVE DATE AND TIME
C     FROM A GIVEN DATE FIND TIME AND RN OFFSET IN HOURS.
C
C HISTORY: WRITTEN BY E.L. LYNN, GLERL, OCTOBER, 1903
C
      CHARACTER*80 FINS
      CHARACTER*50 LAKE
      CHARACTER*10 CDATE, CTIME
      LO6 I CAL DONE
      DIMENSION NUMDAY(12)
      DIMENSION WD(241),WS(241)
C
C LOGICAL UNIT NUMBERS FOR INPUT, PROMPTING, METEOROLOGICAL DATA
C   FILES, CONTROL FILE, AND GENERAL INFORMATION FILE
C
      DATA LUNI, LUNO, LUNB, LUNMW, LUNW, LUNX /5,6,7,8,10,15/
C
C STRRTING HOUR FOR WIND, INITIAL ENDING HOUR, AND INCREMENT
C
      DATA IHPREV, IHPOST, IHINC /0,12,1/
```

```
C
C    NUMBER OF DAYS IN EACH MONTH. USED TO CHECK DATE INPUT
C
      DATA NUMDAY/31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/
C
C   REWIND DATA FILES BEFORE WRITTING TO THEM
C
      REWIND LUNMW
      REWIND LUNW
      REWIND LUNX
C
C    LIMIT ON FORECASTING TIME INCREMENTS. NUMBER OF DAYS OF FORECAST
C    CAN BE CALCULATED BY ((241-1)*(HOURS BETWEEN DATA POINTS))/24
C
      DATA LMTIME /241/
C
C   SAVE REAL DATE TIME FIND LAKE NAME ON INFORMATION FILE
C   HERE DATE FIND TIME ARE VAX FORTRAN LIBARAY FUNCTIONS
C
      CALL DATE (CDATE)
      CALL TIME(CTIME)
      WRITE(LUNX, *) CDATE, CTIME
C
C    GET FORECASTER NAME RND AGENCY, WRITE TO STAT FILE
C
      ASSIGN 30 TO LABEL
   30 WRITE(LUNO,*)
     1'PLEASE ENTER YOUR NAME AND AGENCY (E.G. MARVIN MILLER/NWS)'
      RERD(LUNI,'(A)', END=5000,ERR=5000) ANS
      WRITE(LUNX,*) ANS
C
C   READ THE FIRST LINE OF BATHYMETRIC DATA FILE TO GET LAKE NAME
C
      OPEN (7,READONLY, STATUS=' OLD' )
      READ (LUNB, 1600) LAKE
      WRITE(LUNX,*) LAKE
C
C    WILL HELP FILE BE REOUIIRED ?
C
      ASSIGN 40 TO LABEL
   40 WRITE(LUNO,*)
     1' DO YOU WANT DETAILED INSTRUCTIONS ABOUT HOW TO'
      WRITE(LUNO,*)
     1' USE THIS PROGRAM ?  (INPUT YES...OR...NO)'
      READ(LUNI,'(A)', END=5000,ERR=5000) ANS
      IF (FINS . NE. ' YES' . FIND. FINS . NE. 'NO' ) GOT0 5000
      IF(ANS . EQ. 'YES' ) CALL HELPER
```

```
C
C GET MODEL START DATE AND TIME
C
  100 WRITE (LUNO,*)
     1' INPUT STARTING DATE RND TIME AS MONTH, DAY, YEAR, HOUR (0-23).'
     2,'THIS WILL BE HOUR 0 (ZERO) (E.G. 7,28,1983,14)'
      ASSIGN 100 TO LABEL
      READ (LUNI,*,END=5000,ERR=5000) MON, IDAY, IY, IH
      DONE = .TRUE.
C
C  TEST FOR VALID YEAR, MONTH, DAY AND HOUR
C
      IF (MOD (IY,4) .EQ. 0) NUMDAY(2) =  2 9
      IF (MON .LT. 1 . OR. MON .GT. 12) THEN
       WRITE (LUNO,*) MON,' IS NOT A VALID MONTH'
       MON = 1
       DONE = .FALSE.
      END IF
      IF (IDAY . LT. 1 .OR. IDAY . GT. NUMDAY (MON)) THEN
       WRITE (LUNO,*) IDAY,' IS NOT A VALID DAY'
       DONE = .FALSE.
      END IF
      IF (IY .LT. 1900 .OR. IY .GT. 1999) THEN
       WRITE (LUNO,*) IY,' IS NOT A VALID YEAR'
       DONE = .FALSE.
      END IF
      IF (IH .LT. 0 .OR. IH .GT. 23) THEN
       WRITE (LUNO,*) IH,' IS NOT A VALID HWR'
       DONE = .FALSE.
      END IF
      IF (. NOT. DONE) GO TO 100
C
C GET TIME IN HOURS BETWEEN WIND DATA POINTS
C
      ASSIGN 125 TO LABEL
  125 WRITE (LUNO,*)
     1' INPUT TIME IN HOURS BETWEEN WIND DATA POINTS (1,2,3,4 OR 6)'
      READ (LUNI,*,END=5000,ERR=5000) IHINC
      IF (IHINC . NE. 1 . AND. IHINC .NE. 2 . AND. IHINC .NE. 3 . AND.
     1   IHINC .NE. 4 . AND. IHINC . NE. 6) THEN
       WRITE (LUNO,*)
     1 IHINC,' IS NOT ONE OF THESE (1,2,3,4 OR 6) PLEASE RE-ENTER'
       GOT0 i25
      END IF
C
C SET UP METEOROLOGICAL DATA FILES
C
```

```
      WRITE(LUNO,*)' INPUT WINDS (DIRECTION, SPEED) IN DEGREES AND',
     1 ' KNOTS (E.G. 270,10)'
      WRITE(LUNO,*)'TYPE ''/'' FOR LINEAR INTERPOLATION'
      ICNT = 0
      ISTART=IHPREV
      ISTOP=IHPOST
      DONE=.FALSE.
150   DO 300 ITIME=ISTART,ISTOP,IHINC
      ICNT = ICNT + 1
      CALL RDATE(IY,MON,IDAY,IH,ITIME,IYM,MONM,IDAYM,IHM)
200   WRITE(LUNO,1300) ITIME,MONM,IDAYM,IYM,IHM
      ASSIGN 200 TO LABEL
      WD(ICNT) = -1.
      READ(LUNI,*,ERR=5000) WD(ICNT), WS(ICNT)
C
C IF / WAS TYPED, WD(ICNT) IS NOT CHANGED. INTERPOLATION REQUIRED
C
      IF(WD(ICNT).EQ.-1.) THEN
       IF(ICNT.EQ.1) THEN
        ASSIGN 330 TO LABEL
        GOTO 5000
  330   WRITE(LUNO,*)'WINDS MUST BE SPECIFIED FOR THE FIRST HOUR'
        GOTO 200
       END IF
      ELSE
c
C TEST FOR VALID SPEED AND DIRECTION
C
       IF(WD(ICNT).LT.0. .OR.WD(ICNT).GT.360.) THEN
        WRITE(LUNO,*)'BAD WIND DIRECTION',WD(ICNT)
        GOTO 200
       END IF
       IF(WS(ICNT).LT.0. .OR.WS(ICNT).GT.100.) THEN
        WRITE(LUNO,*)'BAD WIND SPEED',WS(ICNT)
        GOTO 200
       END IF
      END IF
  300 CONTINUE
C
C CHECK FOR LENGTH OF MODEL RUN - IF LONGER THAN IHPOST,
C EXTEND IT BY IHPOST HOURS
C
400   IF(DONE.EQV..TRUE.) GOTO 405
      WRITE(LUNO,1400) ISTOP
      WRITE(LUNO,*)'(INPUT YES...OR...NO)'
      ASSIGN 400 TO LABEL
      READ(LUNI,'(A)',END=5000,ERR=5000) FINS
      IF (FINS.NE.'YES'.AND.ANS.NE.'NO') GO TO 5000
      IF(ANS.EQ.'YES') THEN
       ISTART=ISTOP+IHINC
       ISTOP=ISTOP+IHPOST
```

```
C
C    TEST IF MAXIMUM HOUR FORECAST HRS BEEN EXCEEDED
C    IF SO CALCULATE A NEW MAXIMUM 'ISTOP' AND SET 'DONE' TO .TRUE.
C    SD LONGER FORECAST OPTION WILL NOT BE ASKED AGAIN
C
        IF(((ISTOP-IHPREV)/IHINC)+1.GT.LMTIME) THEN
         ISTOP = ((LMTIME-1) * IHINC) + IHPREV
         WRITE(LUNO,*)
     1  'INCREASING FORECAST BY', IHPOST,
     2 'HOURS WOULD EXCEED THE MAXIMUM.'
         DONE = .TRUE.
        END IF
       WRITE(LUNO,1500) ISTOP
        GO TO 150
       ENDIF
C
C READ THROUGH THE WIND DATA AND INTERPOLATE IF NECESSARY
C
  405 IFIRST = 1
      NUMWND = (ISTOP-IHPREV)/IHINC + 1
      DO 410 ICNT = 1, NUMWND
      IF(WD(ICNT).GE.0.) THEN
       GAPS = FLOAT(ICNT-IFIRST)
       IF(GAPS.GE.2.) THEN
C
C DOES THE SHORTEST DISTANCE BETWEEN THE TWO DIRECTIONS CROSS 0 ?
C
        XW = 1./GAPS
        DMAX = AMAX1(WD(IFIRST),WD(ICNT))
        DMIN = AMIN1(WD(IFIRST),WD(ICNT))
        IF((DMIN-DMAX.GT.-360. .AND. DMIN-DMAX .LT. -180.) .OR.
     1   (DMAX-DMIN.GT. 180. .AND. DMAX-DMIN .LT. 360.)) THEN
         DO 420 I=IFIRST+1, ICNT-I
         IF(WD(IFIRST).GT.WD(ICNT)) THEN
          WD(I) = (XW*WD(ICNT) + (1.-XW)*(WD(IFIRST)-360.))
         ELSE
          WD(I) = (XW*(WD(ICNT)-360.) + (1.-XW)*WD(IFIRST))
         END IF
         WS(I) = XW*WS(ICNT) + (1.-XW)*WS(IFIRST)
         IF(WD(I).LT.0.) WD(I) = WD(I) + 360.
         XW = XW +(1./GAPS)
  420    CONTINUE
C
C ARE THE TWO DIRECTIONS 180 OUT OF PHASE ?
C
```

```
          ELSE IF(ABS(WD(IFIRST)-WD(ICNT)).EQ.180.)THEN
          DO 425 I = IFIRST+1,ICNT-1
          IF(XW.LT..5)THEN
           WD(I) = WD(IFIRST)
          ELSE
           WD(I) = WD(ICNT)
          END IF
          WS(I) = XW*WS(ICNT) + (1.-XW)*WS(IFIRST)
          XW = XW +(1./GAPS)
  425     CONTINUE
C
C DOES THE SHORTEST DISTANCE BETWEEN THE TWO DIRECTIONS NOT CROSS 0 ?
C
         ELSE
          DO 430 I = IFIRST+1,ICNT-1
          WD(I) = XW*WD(ICNT) + (1.-XW)*WD(IFIRST)
          WS(I) = XW*WS(ICNT) + (1.-XW)*WS(IFIRST)
          XW = XW +(1./GAPS)
  430     CONTINUE
         END IF
        END IF
        IFIRST = ICNT
       END IF
  410 CONTINUE

C
C IF WIND WAS NOT SPECIFIED FOR THE LAST HOUR FILL IN WIND
C LOCATIONS FROM THE LAST WIND THAT WAS SPECIFIED
C
       IF(WD(NUMWND).LT.0.)THEN
        DO 440 ICNT =IFIRST+1,NUMWND
        WD(ICNT) = WD(ICNT-1)
        WS(ICNT) = WS(ICNT-1)
  440 CONTINUE
       END IF
C
C   WRITE OUT THE WIND DATA IN TABULAR FORM
C
  450 WRITE(LUNO, * )'  HOUR       DATE         TIME    WIND DIR.    WIND SPEED
     1'
       ICNT = 0
       ISTART = IHPREV
       DO 500 ITIME = ISTART,ISTOP,IHINC
       ICNT = ICNT + 1
       CALL RDATE(IY,MON,IDAY,IH,ITIME,IYM,MONM,IDAYM,IHM)
       IWD = INT(WD(ICNT))
       WRITE(LUNO,1200) ITIME,MONM,IDAYM,IYM,IHM,IWD,WS(ICNT)
  500 CONTINUE
C
C   OPTION TO CHANGE WIND DATA SPEED FIND DIRECTION
C
```

```
  550 WRITE(LUNO,*)'IS WIND DATA O.K.?  (INPUT YES...OR...NO)'
      ASSIGN 550 TO LABEL
      READ(LUNI,'(A)',END=5000,ERR=5000) ANS
      IF(ANS .NE.'YES' .AND. ANS .NE. 'NO') GOTO 5000
      IF(ANS .EQ. 'NO') THEN
C
C   CORRECTION IN WIND IS REQUIRED
C
       WRITE(LUNO,*)
     1 'TO CHANGE WIND DATA(DIRECTION,SPEED) TYPE NEW VALUES'
       WRITE(LUNO,*)
     1 'IN DEGREES AND KNOTS (E.G. 270,10) OR ''/''TO LEAVE IT ALONE'
       ICNT = 0
      WRITE(LUNO,*)'  HOUR       DATE        TIME    WIND DIR.    WIND SPEED
     1'
       DO 600 ITIME=ISTART, ISTOP,IHINC
        ICNT = ICNT + 1
        CALL RDATE(IY, MON, IDAY, IH, ITIME, IYM, MONM, IDAYM,IHM)
  650   IWD = INT(WD(ICNT))
        WRITE(LUNO,1200) ITIME, MONM, IDAYM, IYM, IHM, IWD,WS(ICNT)
        READ(LUNI,*,END=675,ERR=675) WD(ICNT), WS(ICNT)
C
C   TEST FOR VALID WIND SPEED AND DIRECTION
C
        IF(WD(ICNT).LT. 0 .OR. WD(ICNT) .GT. 360) THEN
         WRITE(LUNO,*)'BAD WIND DIRECTION',WD(ICNT)
         GO TO 650
        END IF

        IF(WS(ICNT) .LT. 0. . OR. WS(ICNT).GT. 100.) THEN
         WRITE(LUNO,*)'BAD WIND SPEED',WS(ICNT)
         GOTO 650
        END IF
675     REWIND LUNI
600     CONTINUE
        GO TO 450
      END IF
C
C  SET UP METEOROLOGICAL DATA RECORD
C
      T = 0.
      ICNT = 0
      DO 700 ITIME = ISTART,ISTOP,IHINC
       ICNT = ICNT + 1
C
C CONVERT SPEED TO METERS PER SECOND
C
       WS(ICNT)=WS(ICNT)*0.5144
C
C  SET LATITUDE AND LONGITUDE
C
       XLAT=43.
       XLON=85.
```

```
C
C SET HEIGHT FIT 5 METERS
C
      Z=5.
C
C AIR AND WATER TEMPERATURE SET TO 4 DEGREES
C
      TA=4.
      TW=4.
C
C  WRITE METEOROLOGICAL DATA RECORD
C
      WRITE(LUNMW, 1000) T, XLAT, XLON, Z, TA, TW, WS(ICNT), WD(ICNT)
C
C INCREMENT TIME
C
      T=T+IHINC
  700 CONTINUE
C
C  WRITE END-OF-DATA INDICATOR
C
      WRITE(LUNMW, 1000) - T
      REWIND LUNMW
C
C WRITE CONTROL RECORD FOR WAVE
C   DT = TIME STEP = WIND INCREMENT
C   TT = DURATION OF RUN
C   DADD = WATER LEVEL INCREMENT = 0
C


      DT=IHINC
      TT=ISTOP-IHPREV
      DADD=0.
      WRITE(LUNW, 1000) DT, TT, DADD
      REWIND LUNW
C
C  SAVE START DATE AND TIME, IHPREV, ISTOP, IHINC ON THIS FILE TOO
C
      WRITE(LUNX, 1100) IY, MON, IDAY, IH
      WRITE(LUNX, 1100) IHPREV, ISTOP, IHINC
      REWIND LUNX
C
C   PRINT MESSAGE
C
      WRITE(LUNO, *) 'CALCULATING WAVES...PLEASE STAND BY'
      GOTO 9999
1000  FORMAT(8F10.3)
1100 FORMAT(8I10)
1200  FORMAT('', I5, 4X, I2, '/', I2, '/', 14, 2X, 13, ':00', 4X, I5, 7X, F6.1)
1300  FORMAT(' INPUT WIND FIT', 13, ' HOURS ', 12, '/', 12, '/', 14, ' ', 12,
     1':00')
```

```
1400 FORMAT('IS A FORECAST LONGER THEN', I3, 'HOURS DESIRED?')
1500 FORMAT('FORECAST WILL BE EXTENDED TO', 13, 'HOURS')
1600 FORMAT (A50)
C
C CONE HERE FOR ERROR CONDITION ON INPUT
C
5000  WRITE(LUNO,*)' IMPROPER INPUT.. . PLEASE RE-ENTER'
      REWIND LUNI
      GO TO LABEL,(30, 40, 100, 125, 200, 400, 550)
9999 CONTINUE
      END



      SUBROUTINE HELPER
      WRITE(6,*)'HELP NOT AVAILABLE FIT THIS TINE, SORRY'
      RETURN
      END

      SUBROUTINE RDATE(IY, MON, IDAY, IH, IHINC, IYM, MONM, IDAYM, IHM)
C
C PURPOSE:      THIS SUBROUTINE ADJUSTS TIME BY INCREMENTING OR
C               DECREMENTING THE TINE ARGUMENTS(YEAR, MON., DAY, HOUR)
C               BY THE SPECIFIED NUMBER OF HOURS SPECIFIED BY 'IHINC'.
C
C ARGUMENTS:
C ON INPUT:     IY - STARTING YEAR (NO LIMIT RESTRICTION)
C               NON - STARTING MONTH (1-12)
C               IDAY - STARTING DAY (1 - # OF DRY IN PARTICULAR MONTH)
C               IH - STARTING HOUR (0 - 23)
C               IHINC - NUMBER OF HOURS TO INCREMENT OR DECREMENT TIME.
C                       POSITIVE FOR TIME ADVANCE
C                       NEGITIVE FOR TINE RETRACT
C                       NO LIMIT ON THE NUMBER OF HOURS TO CHANGE TINE.
C
C ON OUTPUT:    IYM - THE ADJUSTED YEAR (NO LIMIT RESTRICTION)
C               MONM - THE ADJUSTED MONTH (1-12)
C               IDAYM - THE ADJUSTED DAY (1 - # OF DAYS IN PARTICULAR MONTH)
C               IHN - THE ADJUSTED HOUR (0 - 23)
C
C HISTORY:      WRITTEN BY E. W. LYNN S-1983, GLERL, ANN ARBOR MI.
C
      DIMENSION NUMDAY(12)
      DATA NUMDAY/31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/
C
C   INITIALIZE THE OUTPUT PARAMETERS TO THE INPUT PARAMETERS
C   AND ADD THE HOUR INCREMENT 'IHINC' TO THE HOUR TIME
C
      IYM = IY
      MONN = NON
      IDAYM = IDAY
      IHM = IH + IHINC
```

```
C
C    ADJUST THE NUMBER OF DAYS IN FEB. DEPENDING IF LEAP YEAR.
C
   10    IF(MOD(IYM,4).EQ.0)THEN
           NUMDAY(2)= 27
         ELSE
           NUMDAY(2)=29.
         END IF
C
C    GO TO THE REQUIRED ROUTINE DEPENDING ON WEATHER TIME NEEDS
C    INCREMENTING OR DECREMENTING.
C
       IF(IHINC.GE.0)THEN
C
C    ADJUST ADVANCED TIME (IYM, MONM, IDAYM,IHM).
C
         IF(IHM.GT.23)THEN
           IHM = IHM-24
           IDAYM = IDAYM + 1
           IF(IDAYM.GT.NUMDAY(MONM))THEN
             IDAYM = IDAYM - NUMDAY(MONM)
             MONM=MONM+1
             IF(MONM.GT.12)THEN
               NONN=MONM-12
               IYM= IYM + 1
             END IF
           END IF
         END IF
         IF(IHM.GT.23)GOT0 10
       ELSE
C
C    ADJUST BACKWARD TINE(IYM, MONM, IDAYM, IHM)
C
         IF(IHM.LT.0)THEN
           IDAYM=IDAYM-1
           IHM=IHM+24
           IF(IDAYM.LT.1)THEN
           I= NONM - 1
           IF(I.EQ.0) I=  12
             IDRYM = IDAYM+NUMDAY(I)
             MONM = MONM-1
             IF(MONM.LT.1)THEN
               IYM = IYM - 1
               MONM = MONM+12
             END IF
           END IF
         END IF
         IF(IHM.LT.0)GOT0 10
       END IF
       RETURN
       END
```

```
      PROGRAM WAVE
C*****************************PROGRAM WAVE*****************************
C
C PURPOSE :
C
C                  TO PREDICT SURFACE GRAVITY WAVES FROM THE WIND.
C
C  INPUT :
C   LOGICAL UNIT 5 :
C     TIME-SHORING USER INPUT
C
C   LOGICAL UNIT 7 :
C     BATHYMETRIC DATA FILE :
C                THE FORMAT OF THE BATHYMETRIC DATA FILE IS DESCRIBED
C                IN SCHWAB AND SELLERS (1980): 'COMPUTERIZED BATHY-
C                METRY FiND SHORELINES OF THE GREAT LAKES', NOAA DATA
C                REPORT ERL-GLERL-16, AS DESCRIBED IN SCHWAB, BENNETT,
C                FIND JESSUP (1981) : 'A TWO-DIMENSIONAL LAKE CIRCULATION
C                MODELING SYSTEM', NOAA TECHNICAL MEMORANDUM ERL-GLERL-3
C                FIVE ADDITIONAL FIELDS ARE PRESENT ON BATHYMETRIC
C                DATA HEADER RECORD NUMBER 2.  THESE ARE:
C                                                  FORMAT CARD COLUMNS
C                MINIMUM DEPTH                       I5        45-49
C                BASE GRID ROTATION FROM E-W         F6.2      50-55
C                I DISPLACEMENT                      I5        56-60
C                J DISPLACEMENT                      I5        61-65
C                ROTATION RNGLE FROM BASE            F6.2      66-71
C                 (ANGLES MEASURED IN DEGREES COUNTERCLOCKWISE)
C
C                THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
C                BETWEEN LATITUDE, LONGITUDE PAIRS FIND GRID DISTANCES.
C                ONLY THE BATHYMETRIC PART OF THE FILE IS USED, SHORE-
C                LINE INFORMATION NEED NOT BE INCLUDED.
C
C   LOGICAL UNIT 8 :
C     METEOROLOGICAL DATA FILE :
C                                                  FORMAT CARD COLUMNS
C                TLAST - TINE FROM BEGINNING       G10.4      1 - 10
C                        OF RUN (H)
C                PLAT - LATITUDE IN DEGREES NORTH  610.4      11 - 20
C                PLON - LONGITUDE IN DEGREES WEST  610.4      21 - 30
C                Z - HEIGHT OF INSTRUMENT (M)      010.4      31 - 40
C                TA - TEMPERATURE O F AIR (C)      G10.4      41 - 50
C                T W - TEMPERATURE O F WATER (C)   G10.4      51 - 60
C                WS - WIND SPEED (M/S)             G10.4      61 - 70
C                WD - WIND DIRECTION (DEG)         G1O. 4     71 - a0
C
C          ALL DATA FOR THE SAME TIME FIRE GROUPED TOGETHER, WITH A
C          MAXIMUM OF 25 STATIONS IN A GROUP.
C
C          • NOTE: END-OF-FILE IS INDICATED BY A RECORD WITH A
C                  NEGATIVE TINE.
```

```
C
C   LOGICAL UNIT 10 :
C     RECORD 1 - CONTROL PARAMETER RECORD :
C                                                          FORMAT    CARD
C                DT - TINE STEP IN HOURS                   610.2
C                TT - DURATION OF RUN IN HOURS             G10.2       1
C                DADD - MEAN WATER LEVEL RELATIVE          G1O. 2      2
C                       TO LOW WATER DATUM IN METERS
C
C  OUTPUT :
C    LOGICAL UNIT 6 :
C      CONTROL PARAMETERS, BATHYMETRY, AND A LIST OF THE
C      NETEOROLOGICQL DATA RECORDS.
C
C  COMMON BLOCKS :
C                /CPARM/ D T , TT, DADD,   - CONTROL PARAMETERS
C                /GPARM/ RPARM(23), IPARM(54) - REAL FIND INTEGER
C                   PARAMETERS DESCRIBING THE BATHYMETRIC GRID.
C                   SEE SUBROUTINE RGRID FOR DETAILS.
C                /TPARM/ NDTT - NUMBER OF SMALL TIME STEPS TAKEN
C                   DURING LAST LARGE TINE STEP
C
C  SUBROUTINES :
C
C       RGRID - READS THE BATHYMETRIC DATA FILE
C       PGPARM - PRINTS GRID PARAMETERS
C       PRNT - FORMATS AND PRINTS OUTPUT ON GRID
C       FUNCTION WINDX - READS METEOROLOGICAL DATA AND CALCULATES
C             X FIND Y COMPONENTS OF EQUIVALENT NEUTRAL STABILITY WIND
C       FUNCTION UZL - USED BY WIND
C       FUNCTION UZ - USED BY WIND
C       FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN GIVEN
C                        LATITUDE FIND LONGITUDE
C       FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN GIVEN
C                        LATITUDE RND LONGITUDE
C
C               THE USER MUST SUPPLY TW0 SUBROUTINES TO SUPPLY
C               INITIAL CONDITIONS AND GENERATE OUTPUT.  THEY ARE:
C
C       WINIT(D, CG, XMOM, YMOM, SIG, IDIM) - SUPPLIES INITIAL CONDITIONS
C
C       WOUTP(T, D, CG, XMOM, YMOM, SIG, CD, IDIM) - GENERATES USER-REQUIRED OUTPUT
C            WOUTP IS CALLED BY WAVE EVERY TIME STEP (DT, SPECIFIED
C            BY THE USER).  T IS THE TIME (IN SECONDS) FROM THE BEGINNI
C            OF THE RUN.  D IS THE DEPTH ARRAY.
C            IDIM IS THE FIRST DIMENSION OF D.
C
C  HISTORY :
C     WRITTEN BY J.R. BENNETT, 1982, GLERL, ANN ARBOR, MI., BY
C     MODIFYING A PROGRAM WRITTEN BY M. A. DONELAN, NATIONAL WATER
C     RESEARCH INSTITUTE, BURLINGTON, ONTARIO.
```

```
C
C     FOR THE INTERRCTIVE VERSION OF THE WAVE MODEL ALL PRINT
C     STATEMENTS AND SUBROUTINE CRLLS ARE COMMENTED OUT.
C     SUBROUTINES PCPARM, PRNT AND WINIT HAVE BEEN REMOVED
C     UNITS 11, 12 AND 13 ARE USED IN WOUTP TO OUTPUT WAVE
C     HEIGHT, DIRECTION AND PERIOD RESPECTIVELY.
C     D. J. SCHWAB NOVEMBER, 1983
C
C
C     THERE ARE SIX MAJOR ARRAYS USED TO DESCRIBE THE WAVE FIELD.
C     THEY ARE DIMENSIONED (IM, JM). THE SIX ARRAYS ARE:
C
C          XMON - THE X-COMPONENT OF MOMENTUM
C          YNON - THE Y-COMPONENT OF MOMENTUM
C          C    - THE MAGNITUDE OF THE MEAN PHASE SPEED
C          S    - THE WAVE HEIGHT STANDARD DEVIRTION
C          cs   - THE COSINE OF THE MEAN WAVE DIRECTION
C          SN   - THE SINE OF THE MEAN WAVE DIRECTION
C
C         THE THREE OTHER ARRAYS ARE:
C
C          D    - THE DEPTH; THIS IS USED ONLY TO DETERMINE WHETHER A GRID
C                 POINT IS IN THE LAKE OR NOT. THE MAGNITUDE OF THE DEPTH
C                 IS NOT USED IN THIS DEEP WATER VERSION OF THE MODEL.
C          CD   - THE DRAG COEFFICIENT OF AIR OVER WRTER. THE PREDICTION
C                 OF CD IS A BY-PRODUCT OF DONELAN'S THEORY; REMOVING ITS
C                 CALCULATION DOES NOT AFFECT THE WAVE PREDICTION METHOD.
C          uv   - THE X AND Y COMPONENTS OF THE WIND
C
      RERL D(41,35),C(41,35),XMOM(41,35),YMOM(41,35),S(41,35)
     1,CS(41,35),SN(41,35),CD(41,35)
      COMMON/CPARM/DT, TT, DADD
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/TPARM/NDTT
C
C   DIMENSIONS OF ARRAYS
C
      DATA IDIM, JDIM/41,35/
C
C   PHYSICAL CONSTRNTS
C
      DATA G, VK/9.81, 0.4/
      DATA RHOAIR, RHOH20, GAMMA/1.2233, 1000., 0.028/
C
C RNEMONETER HEIGHT
C
      DATA Z /10./
C
C   COMPUTE CONSTRNTS
C
      FAC2=0.5*GAMMA*RHOAIR/(RHOH20*G)
      GTPI=G/(2.*ATAN2(0.,-1.))
```

```
C
C  OPEN STATEMENT REWIRED BY VAX FORTRAN
C
      OPEN (7,READONLY, STATUS-' OLD')
C
C READ CONTROL PARAMETERS
C
      READ(10,510) DT,TT,DADD
C     WRITE(6,520)DT,TT,DADD
      NSTEPS=TT/DT+0.5
      DT=DT*3600.
C
C READ BATHYMETRIC GRID INFORMATION
C
      CALL RGRID(7,D, IDIM, JDIM)
      IM=IPARM(1)
      JM=IPARM(2)
      DS=RPARM(3)
      DMIN=RPARM(5)+DADD
      IMM1=IM-1
      JMM1=JM-1
C
C IF WATER LEVEL INCREMENT RESULTS IN A NEGATIVE DEPTH, STOP
C
      IF(DMIN. LT.@.) SO TO 500
C
C PRINT BATHYMETRIC DATA FILE HERDER INFORMATION
C
C     C & L PSPARM(6)
C
C PRINT DEPTH FIELD
C
C     WRITE(6,540)
C     CALL PRNT (6,D,IDIM,IM,JM,0.)
C
C SET INITIAL CONDITIONS
C
      D O 10 I=1, IM
      D O 10 J=1, JM
      CD(I,J)=0.
      XMOM(I,J)=0.0
      YMOM(I,J)=0.0
      C(I,J)=0.0
      CS(I,J)=1.0
      SN(I,J)=0.0
      S(I,J)=0.0
      IF(D(I,J).LT.RPARM(5)) GO T O 10
      D(I,J)=D(I,J)+DADD
   10 CONTINUE
C     CALL WINIT(D,C,XMOM,YMOM, S, IDIM)
```

```
C
C          THIS PROGRAM USES TWO TIME STEPS. THE LARGE TINE STEP (DT)
C    IS SPECIFIED BY THE USER AND CONTROLS THE FREQUENCY AT WHICH DATA
C    CAN BE PRINTED OR STORED FOR LATER ANALYSIS BY WOUTP, THE USER SUP-
C    PLIED OUTPUT ROUTINE. THE SMALL TIME STEP (DTT) IS CALCULATED BY
C    THE PROGRAM FROM THE FORMULA FOR NUMERICAL STABILITY OF THE UPWIND
C    FINITE DIFFERENCE SCHEME. THE NUMBER OF SMALL TIME STEPS PER LARGE
C    TIME STEP IS NDTT; IT IS STORED IN COMMON BLOCK /TPARM/.
C          INSIDE EACH SMALL TIME STEP THERE ARE TWO LOOPS OVER THE SPATIAL
C    COORDINATES I AND J. I N THE FIRST LOOP THE WAVE ADVECTION TERMS ARE
C    CALCULATED. IN THE SECOND LOOP TN WIND INPUT OF MOMENTUM IS ADDED
C    AND THE DIAGNOSTIC VARIABLES ARE CALCULATED.
C
C
C  BEGIN MAIN TIME LOOP
C
      T=0.0
      DO 30 NS=1,NSTEPS
         T=T+DT/2.
C
C  CALCULATE SMALL TIME STEP
C
        XMAX=IM*DS
        YMAX=JM*DS
        WMAX=0.
       WMAX=AMAX1(WMAX,ABS(WINDX(T,0.,0.,1,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,0.,0.,2,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,0.,YMAX,1,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,0.,YMAX,2,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,XMAX,0.,1,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,XMAX,0.,2,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,XMAX,YMAX,1,Z)))
       WMAX=AMAX1(WMAX,ABS(WINDX(T,XMAX,YMAX,2,Z)))
       DO 40 I=2,IMM1
       DO 40 J=2,JMM1
       IF(D(I,J).LT.DMIN) GO TO 40
       DO 50 N=1,2
       WMAX=AMAX1(WMAX,ABS(C(I,J)))
   50 CONTINUE
   40 CONTINUE
        NDTT=IFIX((1.414*WMAX*DT)/DS)+1
        DTT=DT/NDTT
      DTTDS=DTT*0.25/DS
      DTTGAM=FAC2*DTT
```

```
C
C BEGIN LOOP OVER SMALL TIME STEPS
C
      DO 60 NDT=1,NDTT
C
C   FIRST LOOP OVER I,J:CALCULATE TM WAVE ADVECTION TERMS
C
      Do 78 I-2,IMM1
      DO 70 J=2,JMM1
      IF(D(I,J).LT.DMIN)GO TO 70
C
C   X-COMPONENT
C
       IF(CS(I,J).GE.0.0)XFLXD=((S(I,J)*CS(I,J))**2)
     1-((S(I-1,J)*CS(I-1,J))**2)
       IF(CS(I,J).LT.0.0)XFLXD=((S(I+1,J)*CS(I+1,J))
     1**2)-((S(I,J)*CS(I,J))**2)
       IF(SN(I,J).LT.0.0)YFLXD=(CS(I,J+1)*SN(I,J+1)*
     1S(I,J+1)**2)-(CS(I,J)*SN(I,J)*S(I,J)**2)
       IF(SN(I,J).GE.0.0)YFLXD=(CS(I,J)*SN(I,J)*S
     1(I,J)**2)-(CS(I,J-1)*SN(I,J-1)*S(I,J-1)**2)
       SIP=S(I+1,J)
       SIM=S(I-1,J)
       IF(D(I+1,J).LT.DMIN) SIP=2.*S(I,J)-S(I-1,J)
       IF(D(I-1,J).LT.DMIN) SIM=2.*S(I,J)-S(I+1,J)
       XMOM(I,J)=XMOM(I,J)-DTTDS*(XFLXD+YFLXD+0.25*(SIP**2
     1-SIM**2))
C
C   Y-COMPONENT
C
       IF(CS(I,J).LT.0.0)XFLXD=(CS(I+1,J)*SN(I+1,J)*
     1S(I+1,J)**2)-(CS(I,J)*SN(I,J)*S(I,J)**2)
       IF(CS(I,J).GE.0.0)XFLXD=(CS(I,J)*SN(I,J)*
     1S(I,J)**2)-(CS(I-1,J)*SN(I-1,J)*S(I-1,J)**2)
       IF(SN(I,J).GE.0.0)YFLXD=((SN(I,J)*S(I,J))**2)
     1-((SN(I,J-1)*S(I,J-1))**2)
       IF(SN(I,J).LT.0.0)YFLXD=((SN(I,J+1)*S(I,J+1))**2)
     1-((SN(I,J)*S(I,J))**2)
       SJP=S(I,J+1)
       SJM=S(I,J-1)
       IF(D(I,J+1).LT.DMIN) SJP=2.*S(I,J)-S(I,J-1)
       IF(D(I,J-1).LT.DMIN) SJM=2.*S(I,J)-S(I,J+1)
       YMOM(I,J)=YMOM(I,J)-DTTDS*(XFLXD+YFLXD+0.25*(SJP**2
     1-SJM**2))
  a0   CONTINUE
C
   70 CONTINUE
```

```
C
C     SECOND LOOP OVER I, J: CALCULATE THE WIND INPUT OF MOMENTUM
C     AND COMPUTE THE DIAGNOSTIC VARIABLES (CS, SN, S, C, CD)
C
      DO 90 I=2,IMM1
      DO 98 J=2,JMM1
      IF(D(I,J).LT.DMIN) GO TO 90
C
C     CALCULATE THE VERTICAL MOMENTUM FLUX IN TM 2 DIRECTIONS,
C     THE WIND AND THE WAVE FIELD
C
      X=(I-0.5)*DS
      Y=(J-0.5)*DS
      UWIND=WINDX(T,X, Y,1,Z)
      VWIND=WINDX(T,X, Y,2,Z)
      WSPDSQ=UWIND*UWIND+VWIND*VWIND
      WNDSPD=SQRT(WSPDSQ)+1.E-20
      CSWIND=UWIND/WNDSPD
      SNWIND=VWIND/WNDSPD
      DTFAC=DTTGAM*WSPDSQ
      CTHE1=CSWIND*CS(I,J)+SNWIND*SN(I,J)
      SG=AMAX1(0.005,S(I,J)*ABS(CTHE1))
      DRAG1=(VK/ALOG(50./SG))**2
      SG=AMAX1(0.005,S(I,J)*ABS(CTHE1))
      DRAG2=(VK/ALOG(50./SG))**2
      B=C(I,J)*.83/WNDSPD
      A=1.-B*CTHE1
      WI1=DRAG1*A*ABS(A)
      A=CTHE1-B
      WI2=DRAG2*A*ABS(A)
      VFLXA=WI1*CSWIND+WI2*CS(I,J)
      VFLYA=WI1*SNWIND+WI2*SN(I,J)
      XMOM(I,J)=XMOM(I,J)+DTFAC*VFLXA
      YMOM(I,J)=YMOM(I,J)+DTFAC*VFLYA
C
C     COMPUTE THE NEW COSINES AND SINES; AND COMPUTE THE PHASE
C     SPEED(C) AND WAVE HEIGHT STANDARD DEVIATION(S).
C
      CM=SQRT(XMOM(I,J)**2+YMOM(I,J)**2)+1.E-10
      CS(I,J)=XMOM(I,J)/CM
      SN(I,J)=YMOM(I,J)/CM
      U=UWIND*CS(I,J)+VWIND*SN(I,J)
      IF(U.LE.0.0) GO TO 150
      FM=0.01708735*(((U*U)/(CM*CM*CM))**0.14285714)
      IF((U*FM).GT.0.760545) GO TO 160
  150 FM=(CM*14343.09)**(-.3333333333)
  160 S(I,J)=SQRT(CM*GTPI/FM)
      C(I,J)=GTPI/FM
```

```
C
C     THE FOLLOWING STATEMENTS ARE USED ONLY FOR CALCULATING C D
C     AND CAN BE REMOVED IF CD IS NOT REQUIRED.
C
C       IF(NDT.NE.NDTT) GO TO 96
C       SX=0.5*(VFLXA+VFLXF)+0.0007*CSWIND
C       SY=0.5*(VFLYA+VFLYF)+0.0007*SNWIND
C       CD(I,J)=SQRT(SX*SX+SY*SY)
C
    96 CONTINUE
C
C     END LMP OVER SMALL TIME STEPS
C
    60 CONTINUE
C
C   UPDATE TIME
C
          T=NS*DT
C
C CALL OUTPUT ROUTINE
C
       CALL WOUTP(T,D,C,XMOM,YMOM,S,CD,IDIM)
C
C END OF MAIN TIME LOOP
C
    36 CONTINUE
       GO TO 505
   566 WRITE(6,530) DADD
   505 CONTINUE
   516 FORMAT (3G10.2)
   520 FORMAT('1',//,'DT = ',F6.2,'     TT = ',F9.2,'     DADD = ',F6.2)
   530 FORMAT(1X,'THE WATER LEVEL INCREMENT',F8.2,'RESULTS IN A',
      1        'NEGATIVE MINIMUM DEPTH - PROGRAM TERMINATED')
   540 FORMAT(1X,'DEPTH RELATIVE TO MEAN WATER LEVEL')
       END
```

```
      SUBROUTINE RGRID(LUN, D, IDIM, JDIM)
c PURPOSE:
C                  TO READ a STANDARD BATHYMETRIC GRID DATA FILE
C                  AND RETURN GRID PARAMETERS AND DEPTHS.
C ARGUMENTS:
C ON INPUT:
C                  LUN - LOGICAL UNIT NUMBER OF BATHYMETRIC DATA FILE
C                  IDIM - FIRST DIMENSION OF ARRAY D IN
C                         DIMENSION STATEMENT OF CALLING PROGRAM
C                  JDIM - SECOND DIMENSION OF ARRAY D IN
C                         DIMENSION STATEMENT OF CALLING PROGRAM
C ON OUTPUT:
C                   D - DEPTH ARRAY. ZERO FOR LAND, AVERAGE DEPTH
C                       OF GRID BOX IN METERS FOR WATER.
C                   RPARM - ARRAY CONTAINING REAL-VALUED BATHYMETRIC
C                           GRID PARAMETERS AS FOLLOWS:
C                              1. BASE LATITUDE
C                              2. BASE LONGITUDE
C                              3. GRID SIZE (M)
C                              4. MAXIMUM DEPTH (M)
C                              5. MINIMUM DEPTH (M)
C                              6. BASE ROTATION (COUNTERCLOCKWISE FROM E-U)
C                              7. ROTATION FROM BASE (COUNTERCLOCKWISE)
C                              8-11. GEOGRAPHIC-TO-MAP COORDINATE CONVERSION
C                                    COEFFICIENTS FOR X
C                              12-15. GEOGRAPHIC-TO-MAP COORDINATE CONVERSION
C                                     COEFFICIENTS FOR Y
C                              16-19. MAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                                     COEFFICIENTS FOR LONGITUDE
C                              26-23. MAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                                     COEFFICIENTS FOR LATITUDE
C                  IPARM - ARRAY CONTAINING INTEGER-VALUED BATHYMETRIC
C                          GRID PARAMETERS a 6 FOLLOWS:
C                             1. NUMBER OF GRID BOXES IN X DIRECTION
C                             2. NUMBER OF GRID BOXES IN Y DIRECION
C                             3. I DISPLACEMENT - THE NUMBER OF NEW GRID
C                                SQUARES IN THE X-DIRECTION FROM THE NEW
C                                GRID ORIGIN TO THE OLD 6RID ORIGIN
C                             4. J DISPLACEMENT - THE NUMBER OF NEW GRID
C                                SQUARES IN THE Y-DIRECTION FROM THE NEW
C                                6RID ORIGIN TO THE OLD GRID ORIGIN
C                             S-54. LAKE NAME (50A1)
C                   NOTE: IF 6RID 16 TOO LARGE FOR DIMENSIONS OF D,
C                         THE IPARM ARRAY I s SET T o ZERO
```

```
C
C COMMON BLOCK:
C                       /GPARM/RPARM(23),IPARM(54)
C
      DIMENSION D(IDIM,JDIM)
      COMMON /GPARM/ RPARM(23), IPARM(54)
      R E R D(LUN,30)(IPARM(I),I=5,54),IPARM(1),IPARM(2),
     1(RPARM(I),I=1,6), IPARM(3), IPARM(4), RPARM(7)
      R E R D(LUN,60)(RPARM(I),I=8,23)
      IM = IPARM(1)
      JM = IPARM(2)
      I F(IPARM(1).GT.IDIM.OR.IPARM(2).GT.JDIM) G O T O 1 0
      READ (LUN,40) ((D(I,J),I=1,IM),J=1,JM)
      RETURN
   10 DO 26 I = 1, 54
   20 IPARM(I) = 0
      WRITE(6,50)
   30 FORMRT(50(A1)/2I5, 2F12.7, 3FS.0, F6.2, 2I5, F6.2)
   40 FORMAT(19F4.0, 4 X )
   50 FORMAT('BATHYMETRIC G R I D TOO LARGE - INCREASE DIMENSIONS OF',
     1       'NDEPTH AND D E P T H I N MAIN PROGRAM')
   60 FORMAT(4E15.6, 2 0 X )
   70 RETURN
      END



      FUNCTION WINDX(T, XS, YS, K, ZWIND)
C
C PURPOSE :
C               TO RETURN EQUIVALENT NEUTRAL STABILITY WIND COMPONENT
C               F O R LOCATION(XS,YS) FIT TIME T.  XS AND YS ARE IN METERS
C               RELATIVE TO THE GRID ORIGIN. THE PARAMETER K
C               INDICATES WHICH COMPONENT OF WIND IS RETURNED.
C               K = 1 FOR THE X COMPONENT AND K = 2 FOR THE Y COMPONENT.
C               BOTH COMPONENT6 ARE LINEAR FUNCTIONS OF X FIND Y.
C               ZWIND IS THE HEIGHT ABOVE THE WATER IN METERS RT
C               WHICH EQUIVALENT NEUTRAL STABILITY WIND IS DETERMINED.
C
```

```
C
C INPUT:
C LOGICAL UNIT 8:
C      METEOROLOGICAL DATA FILE
C      CARD COLUMN        FORMAT      PARAMETER
C          1-10             GIO. 4    TLAST - TIME (HRS) FROM BEGINNING
C                                           OF RUN
C          11-20            G10.4     PLAT - LATITUDE IN DEGREES NORTH
C          21-30            G1O. 4    PLON - LONGITUDE IN DEGREES WEST
C          31-40            610.4     Z - HEIGHT OF INSTRUMENTS
C          41-50            G10.4     TA - TEMPERATURE OF AIR
C          51-60            010.4     TW - TEMPERATURE O F WATER
C          61-70            010.4     ws - WIND SPEED (M/S)
C          71-80            G10.4     W D - WIND DIRECTION (DEG)
C
C          ALL DATA FOR THE SAME TIM E ARE GROUPED TOGETHER, WITH a
C          MAXIMUM OF 25 STATIONS IN a GROUP.
C
C     • NOTE: END-OF-FILE IS INDICATED BY a RECORD WITH a
C             NEGATIVE TIME
C
C OUTPUT :
C LOGICAL UNIT 6:
C      PRESENTATION O F METEOROLOGICAL DATA STATION B Y STATION
C
C COMMON BLOCKS:
C               /GPARM/RPARM(23), IPARM(54)
C
C SUBPROGRAMS :
C      FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN
C                       GIVEN LaTITUDE AND LONGITUDE
C      FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN
C                       GIVEN LATITUDE AND LONGITUDE
C      SUBROUTINE UZL - RETURNS DRAG COEFFICIENT AND WIND PROFILE
C                       PARAMETERS
C      FUNCTION UZ - RETURNS WIND SPEED AT HEIGHT ZWIND BASED ON
C                       WIND PROFILE PARAMETERS
```

```
C ALGORITHM:    THIS FUNCTION READS METEOROLOGICAL DATA FROM a FILE,
C               CALCULATES THE NON-NEUTRAL WIND PROFILE, AND CONVERTS
C               WIND SPEED TO EQUIVALENT NEUTRAL WIND SPEED AT
c               HEIGHT ZWIND.  ALL DATA FOR THE SAME TIME
C               ARE GROUPED TOGETHER, WITH a MAXIMUM OF 25 STATIONS IN
C               a GROUP.  THE TWO COMPONENTS OF THE WIND ARE ASSUMED
C               TO BE LINEAR FUNCTIONS OF X AND Y AND THE BEST-FIT LINEA
C               SURFACE FOR THE GIVEN DATA POINTS IS CALCULATED. EACH
c               TIME WIND IS CALLED, THE TIME PARAMETER IS CHECKED
C               AGAINST THE TIME OF THE LAST SET OF METEOROLOGICAL DATA
C               READ.  IF IT IS GREATER, ANOTHER GROUP IS READ.  IF IT
C               IS LESS, THE APPROPRIATE WIND COMPONENT IS CALCULATED
C               USING LINEAR INTERPOLATION BETWEEN THE LAST TWO LINEAR
C               SURFACES COMPUTED FOR THAT COMPONENT.  IF THE END-OF-FIL
C               IS ENCOUNTERED, WIND IS CALCULATED WITH THE LAST LINEAR
C               SURFACE COEFFICIENTS.
C
C               MANY OF THE VARIABLES USED IN THIS FUNCTION ARE ASSUM-
c               ED TO HAVE RETAINED THEIR VALUES FROM THE PREVIOUS
c               CALL.  IF YOUR FORTRAN SYSTEM DOES NOT DO THIS AUTO-
C               MATICALLY, PROVISION MUST BE MADE TO RETAIN VALUES
C               FOR THE VARIABLES: ISTA, AOLD1, BOLD1, COLD1, AOLD2,
C               BOLD2, COLD2, ANEW1, BNEW1, CNEW1, ANEW2, BNEW2, CNEW2,
C               TOLD, TNEW, TLAST, PLAT, PLON, WS, WD.
C
C  ARGUMENTS:
C               T - TIME IN SECONDS FROM BEGINNING OF RUN AT WHICH
C                   STRESS IS TO BE CALCULATED
C               XS - X DISTANCE IN METERS FROM GRID ORIGIN AT WHICH
C                    WIND IS TO BE CALCULATED
C               YS - Y DISTANCE IN METER6 FROM GRID ORIGIN AT WHICH
C                    WIND IS TO BE CALCULATED
C               K - SPECIFIES WHETHER X COMPONENT OR Y COMPONENT
C                   OF STRESS IS RETURNED.  K = 1 FOR THE X COMPONENT
C                   AND K = 2 FOR THE Y COMPONENT
C               ZWIND - HEIGHT ABOVE WATER IN METERS AT WHICH WIND IGRE
```

```
C
C HISTORY :
C            WRITTEN BY J. R. BENNETT, 1982, GLERL, ANN ARBOR, MI BY
C            MODIFYING FUNCTION TAU.  SEE SCHWAB, BENNETT, AND JESSUP (1
C
C           *MODIFIED AUGUST, 1983 TO FIX PROBLEM WITH CO-LINEAR
C            METEOROLOGICAL DATA POINTS
C
      DIMENSION X(25), U(2), Y(25), AWIN(25, 2)
      LOGICAL XEQ, YEQ
      COMMON /GPARM/ RPARM(23), IPARM(54)
      DATA RHOAW, ISTA /1.25E-3, 0/
      DATA LUN /8/
      DATA AOLD1, BOLD1, COLD1, AOLD2, BOLD2, COLD2, TOLD /7*0./
C
c  STATEMENT FUNCTION TO CALCULATE DETERMINANT OF 3 BY 3 MATRIX
C
      DET(A11, A21, A31, A12, A22, A32, A13, A23, A33) =
     1 a  l  l * A22 * A33 - all *  a23  •   a32   +
     2  al2  *  a23  *   a31  -  al2  •   a21   *  a33  +
     3  al3  *  a21  *  a32  -  al3  *  a22  *   a31
C
C  IF THIS IS THE FIRST TIME THROUGH, READ a RECORD
C
      IF (ISTA . EQ. 0) GO TO 30
C
C  CHECK IF NECESSARY TO READ ANOTHER RECORD
C
      IF (T .LT. TNEW .OR. TNEW .LT. 0.) GO TO 90
   1 0 AOLD1 = ANEW1
      BOLD1 = BNEW1
      COLD1  = CNEW1
      AOLD2 = ANEW2
      BOLD2 = BNEW2
      COLD2 = CNEW2
      TOLD = TNEW
   2  0 TNEW = TLAST
      IF (TNEW .LT. 0) GO TO 90
C
C  FIND AWIN, THE WIND FOR THE CURRENT STATION AT HEIGHT ZWIND ABOVE WAT
C
      X (ISTA) = XDIST(PLAT, PLON)
      Y (ISTA) = YDIST(PLAT, PLON)
      TD = Ta - TW
      CALL UZL(WS, Z, TD, Z, CD, CH, Z0, FL)
      W S Z  = UZ(ZWIND, WS, C D , Z0, FL)
      CDD = CD *1.E3
```

```
C
C INITIAL GUESS AT EQUIVALENT NEUTRAL WIND SPEED
C
      USN = WSZ
C
C ITERATE TO FIND EQUIVALENT NEUTRAL UIND SPEED
C
      DO 25 IT6 = 1, 10
        CALL UZL(WSN, ZWIND, 0., ZWIND, CDN, CH, Z0, FL)
        WSOLD = WSN
        USN = SQRT(CD/CDN) * WS
        IF(ABS(WSN - WSOLD).LT. 0.01) GO TO 26
25    CONTINUE
26    WSZ = WSN
      AWIN(ISTA,1) = WSZ * COS((270. - WD - RPARM(6) - RPARM(7))*
     1               ATAN(1.)/45.)
      AWIN(ISTA,2) = WSZ * SIN((270. - WD - RPARM(6) - RPARM(7))*
     1               ATAN(1.)/45.)
      TTLAST = TLAST / 3660.
C     IF(ISTA.LE.1) WRITE (6,166)
      XKM = X(ISTA)/1066.
      YKN = Y(ISTA)/1000.
C     WRITE(6,150) TTLAST, PLAT, PLON, Z, TA, TW, WS, WD, CDD,
C    1 XKM, YKN
30    READ (LUN,130) TLAST, PLAT, PLON, Z, TA, TN, WS, WD
      TLAST=TLAST * 3600.
      IF(T.LT.TLAST.AND.ISTA.EQ.0) GO TO 166
      ISTA = ISTA + 1
C
C IF FIRST TIME THROUGH, FIND RUIN
C
      IF(ISTA.EQ.1) GO TO 26
C
C CHECK IF LAST RECORD at TIME TLAST Ha6 BEEN READ
C
      IF(TLAST.EQ. TNEW) GO TO 20
C
C NOW FIND THE BEST-FIT LINEAR SURFACE
```

```
C
      SX  = 0.
      SY = 0.
      SXY = 0 .
      6 X 2 = 0 .
      SY2 = 6 .
      SXWIN1 = 0 .
      SYWIN1 = 0 .
      SAWIN1 = 0 .
      SXWIN2 = 0 .
      SYWIN2 = 0 .
      SAWIN2 = 0 .
      N  = ISTA - 1
      AN = FLOAT(N)
      XEQ = .TRUE.
      YEQ = .TRUE.
      DO 46 IN = 1, N
        S X = S X + X(IN)
        S Y = S Y + Y(IN)
        SXY = S X Y + X(IN) * Y(IN)
        6X2 = SX2 + X(IN) ● * 2
        S Y 2 = SY2 + Y(IN) ** 2
        SXWIN1 = SXWIN1 + X(IN) * AWIN(IN, 1)

        SYWIN1 = SYWIN1 + Y(IN) * AWIN(IN, 1)
        SXWIN2 = SXWIN2 + X(IN) ● AWIN(IN, 2)
        S Y Y I N 2 = SYWIN2 + Y(IN) * AWIN(IN, 2)
        SAWIN1 = SAWIN1 + AWIN(IN, 1)
        SAWIN2 = SAWIN2 + AWIN(IN, 2)
        I F (X(IN) .NE. X(1)) X E Q = .FALSE.
        I  F (Y(IN) .NE. Y(1)) YEQ = .FALSE.
   40 CONTINUE
C
C  CALCULATE COEFFICIENT6 ANEW, BNEW, CNEW, WHERE
C                          WINDY = ANEW ● X + BNEW ● Y + CNEW
C                          FOR EACH COMPONENT.
C
      ANEW1 = 0 .
      ANEW2 = 6 .
      BNEW1 = 0 .
      BNEW2 = 0 .
      CNEW1 = SAWIN1 / AN
      CNEW2 = SAWIN2 / F I N
C
C  CHECK FOR ON2 DATA POINT ONLY
C
      I F (XEQ .AND. YEQ) G O  T O  8 0
C
C  CHECK IF DATA POINTS ARE CO-LINEAR
```

```
C
      IF(N.EQ.2)GO TO 60
      DO 50 I = 3, N
        A = SQRT((X(1) - X(I))**2 + (Y(1) - Y(I))**2)
        B = SQRT((X(1) - X(I-1))**2 + (Y(1) - Y(I-1))**2)
        C = SQRT((X(I) - X(I-1))**2 + (Y(I) - Y(I-1))**2)
        S =  (A + B + C)  / 2.
        D = SQRT(S * (S-A) * (S-B) * (S-C))/AMAX1(A, 6, C,1.)
C
C IF NOT CO-LINEAR, GO TO 75
C
        IF(D.GT.100.)60 TO 75
   50 CONTINUE
C
C CALCULATE COEFFICIENTS FOR CO-LINEAR DATA POINT6
C
   60 WRITE(6,175)
      DO 70 I=2,N
        IF(X(I).EQ.X(1).AND.Y(I).EQ.Y(1))GO TO 76
        J-I
   70 CONTINUE
      ALPHA = ATAN2(Y(J) - Y(1), X(J) - X(1))
      CN = COS(ALPHA)
      SN = SIN(ALPHA)
      SS = CN * SX + SN * SY
      SS2 = [illegible] * 6X2 - 2. * [illegible] * [illegible] * SXY + [illegible] * [illegible] * SY2
      SSWIN1 = CN * SXWIN1 + SN * SYWIN1
      SSWIN2 = CN * SXWIN2 + SN * SYWIN2
      D = AN * SS2 - SS * SS
      ANEW1 = CN * (AN * SSWIN1 - 6 6 * SAWIN1) / D
      ANEW2 = CN it (AN * SSWIN2 - 6 6 * SAWIN2) / D
      BNEW1 = SN * (AN * SSWIN1 - SG * SAWIN1) / D
      BNEW2 = SN * (AN * SSWIN2 - GG * SAWIN2) / D
      CNEW1 = (SS2 * SAWIN1 - SSWIN1 * SS) / D
      CNEW2 = (SS2 * SAWIN2 - SSWIN2 * SS) / D
      GO TO 86
C
C CALCULATE COEFFICIENT6 FOR BEST-FIT LINEAR SURFACE
C
   75 D = DET(SX2, SXY, SX, SXY, SY2, SY, SX, SY, AN)
      ANEW1 = DET(SXWIN1, SYWIN1, SAWIN1, SXY, SY2, SY, SX, SY, AN) / D
      ANEW2 = DET(SXWIN2, SYWIN2, SAWIN2, SXY, SY2, SY, 6X, SY, AN) / D
      BNEW1 = DET(SX2, SXY, SX, SXWIN1, SYWIN1, SAWIN1, SX, SY, AN) / D
      BNEW2 = DET(SX2, SXY, SX, SXWIN2, SYWIN2, SAWIN2, SX, SY, AN) / D
      CNEW1 = DET(SX2, SXY, SX, SXY, SY2, SY, SXWIN1, SYWIN1, SAWIN1)/D
      CNEW2 = DET(SX2, SXY, SX, SXY, SY2, SY, SXWIN2, SYWIN2, SAWIN2)/D
   80 CONTINUE
      ISTA = 1
      IF(T.GT.TNEW)GO TO 10
C     WRITE(6,170)
```

```
C
C  CALCULATE WIND COMPONENTS AT LOCATION (XS, YS) AT NEW AND OLD TIMES
C
   90  IF (K.EQ.2) 60 TO 166
       WINNEW = ANEW1 • X6 + BNEW1 • YS + CNEW1
       WINOLD = AOLD1 • XS + BOLD1 • YS + COLD1
       GO TO 110
  100 WINNEW = ANEW2 * XS + BNEW2 • YS + CNEW2
       WINOLD = AOLD2 * XS + BOLD2 • YS + COLD2
C
C INTERPOLATE IN TIME
C
  116 WINDX = WINOLD
       IF (TNEW .NE. TOLD) WINDX = WINDX + ((T-TOLD)/(TNEW-TOLD)) •
      1     (WINNEW-WINOLD)
C
C MODIFICATION TO GET A, B FIND C FOR X FIND Y COMPONENTS OF WIND WITH K-3 TO 6
C
       IF(K.E0.3) WINDX=AOLD1+(T-TOLD)*(ANEW1-AOLD1)/(TNEW-TOLD)
       IF(K.EQ.4) WINDX=BOLD1+(T-TOLD)*(BNEW1-BOLD1)/(TNEW-TOLD)
       IF(K.EQ.5) WINDX=COLD1+(T-TOLD)*(CNEW1-COLD1)/(TNEW-TOLD)
       IF (K. EO. 6) WINDX=AOLD2+(T-TOLD)*(ANEW2-AOLD2)/(TNEW-TOLD)
       IF(K.EQ.7) WINDX=BOLD2+(T-TOLD)*(BNEW2-BOLD2)/(TNEW-TOLD)
       IF(K. EO. 8) WINDX=COLD2+ (T-TOLD)* (CNEW2-COLD2)/(TNEW-TOLD)
C
C  D J S 10/17/83
C
       GO TO 180
  126 WRIT2 (6,146) TLAST, T
       STOP.
136 FORMAT (8810.4)
140 FORMAT ('TIME OF FIRST METEOROLOGICAL DATA', 810.4,
   1        ' SECONDS IS ', ' GREATER THAN T = ', 616.4,
   2        '-PROGRAM TERMINATED')
150 FORMAT ('',F5.1,' • ', F16.7, ' • ', F16.7, '*', F4.1, ' • ',
   1        FS.2, ' • ',F5.2,' • ', F6.2, ' • ', F4.6, ' • ', FS.2,
   2        '*',F5.0,' • ',F5.0)
160  FORMAT (' ', 95('*')/
   1        'TIME • LATITUDE • LONGITUDE • Z * T-AIR * ',
   2        'T-H2O * W-SPD • W-DIR • CDE3 • X • Y'/' ', 95('*'))
176 FORMAT ('', 95('*'))
175 FORMAT ('THESE DATA POINTS ARE CO-LINEAR OR NEARLY CO-LINEAR')
186 RETURN
     END
```

```
      SUBROUTINE UZL(UM, ZM, TD, ZTM, CD, CH, Z0, FL)
C
C PURPOSE:
C                TO CALCULATE THE BULK AERODYNAMIC COEFFICIENTS FOR
C                MOMENTUM AND HEAT OVER a LAKE SURFACE AS FUNCTIONS
C                OF WIND SPEED AND AIR-SEA TEMPERATURE DIFFERENCE.
C
C ALGORITHM:
C                THERE IS AN OUTER ITERATION IN WHICH THE ROUGHNESS
C                LENGTH IS VARIED ACCORDING TO CWRNOCK'S FORMULA AND
C                AN INNER ITERATION IN WHICH THE STABILITY LENGTH
C                (MONIN-OBUKHOV LENGTH) IS VARIED ACCORDING TO THE
C                BUSINGER-DYER FORMULATION.  THE CONSTANT IN CHARMOCK'S
C                FORMULA 16 CHOSEN SO THAT UNDER NEUTRAL CONDITION6 THE
C                10 M DRAG COEFFICIENT IS 0.0016.
C
C ARGUMENTS:
C ON INPUT:
C                UM - WIND SPEED (M/S)
C                ZM - ANEMOMETER HEIGHT (M)
C                TD - AIR-SEA TEMPERATURE DIFFERENCE (DEG K)
C                ZTM - THERMOMETER HEIGHT
C                      (INITIALLY ASSUMED EQUAL TO ANEMOMETER HEIGHT)
C ON OUTPUT:
C                CD - BULK AERODYNAMIC COEFFICIENT FOR MOMENTUM
C                CH - BULK AERODYNAMIC COEFFICIENT FOR HEAT.
C                Z0 - RWGHNESS LENGTH (M)
C                FL - STABILITY LENGTH (M)
C
C HISTORY:
C                WRITTEN BY J. R. BENNETT AND J. D. BOYD, 1979, GLERL,
C                ANN, ARBOR, MI;  BASED ON a CONSTANT ROUGHNESS VERSION
C                WRITTEN BY PAUL LONG AND WILL SHAFFER OF THE TECHNIQUES
C                DEVELOPMENT LABORATORY, NOAA, SILVER SPRINGS, MD,
C
C

                 DATA C1, C2, C3 /.684E-4, 4.28E-3, -4.43E-4/
                 DATA B1, B2, B3 /1.7989E-3, 4.865E-4, 3.9028E-5/
                 EPS = .01
                 IF (UM .LT. .001) UM = .001
                 FK = .35
                 TBAR = 278.
                 ALPHA = 4.7
                 BETA = .74
                 GAMM = 15.
                 GAMT = 9.
                 UST1 = 0.04 * UM
                 H = ZM
                 DTHETA = TD
                 IF (ABS(DTHETA) .LT. 1.E-7) DTHETA = SIGN(1.E-7, DTHETA)
```

```
C
C INITIAL GUESS FOR Z0
C
      Z0 = .00459 * UST1 * UST1
      S = UM * UM * TBAR / (9.8*DTHETA)
      IF(ABS(S).GT.1.E6) S = SIGN(1.E6,S)
      X = ALOG(H/Z0)
C
C       INITIAL GUESS FOR L
C
      FL = S / X
      DO 60 ITER = 1, 20
        X = ALOG(H/Z0)
        IF(ABS(FL).GT. 3.E6) FL = SIGN(3.E6,FL)
        IF (FL.GT.0.) GO TO 20
C
C UNSTABLE SECTION (L LT 0 OR DT LT 0)
C
        FLI = 1. / FL
C
C ASSUME 5 ITERATIONS SUFFICIENT
C
        DO 10 I = 1, 5
          X1=GAMT * FLI
          ARG1 = SQRT(1. - X1*H)
          ARG2 = SQRT(1. - X1*Z0)
          A = BETA * ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*
     1    (ARG2 -0.9999999)))
          X1 = GAMM * FLI
          ARG1 = (1. - X1*H) ** (.25)
          ARG2 = (1. - X1*Z0) ** (.25)
          B = ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*
     1    (ARG2 - 0.9999999))) +2. * (ATAN(ARG1) - ATAN(ARG2))
          FL = S * A / (B*B)
          IF(ABS(FL).GT. 3.E6) FL = SIGN(3.E6,FL)
          FLI = 1. / FL
   10   CONTINUE
        GO TO 50
C
C STABLE SECTION
C
C TRY MILDLY STABLE-
C
```

```
   20    CONTINUE
         AA = X * X
         X1 = H - Z0
         BB = 9.4 * X1 * X - .74 * S * X
         CC = 4.7 * X1
         MM = MM * MM - MM * .
         ROOT = BB * BB - 4. * AA * CC
         IF (ROOT.LT.0.) GO TO 30
         FL = (-BB + SQRT(ROOT)) / (2.*AA)
         IF (FL .LE. H) GO TO 36
         B = X + 4.7 * X1 / FL
         A = BETA * X + 4.7 * X1 / FL
         GO TO 50
C
C STRONGLY STABLE-
C
   30    CONTINUE
         IF (FL.LE. Z0) FL = Z0 + 1.E-5
         DO 40 I = 1,5
           ARG1 = FL / ZO
           X1 = ALOG(ARG1)
           X2 = ALOG(H/FL)
           ARG1 = 1. - 1. / ARG1
           A = .74 * X1 + 4.7 * ARG1 + 5.44 * X2
           B = X1 + 4.7 * ARG1 + 5.7 * X2
           FL = A * 6 / (B*B)
           IF (FL.LE. Z0) FL = Z0 + 1.E-5
           IF (FL .GT. H) FL = H
   40    CONTINUE
C
C  CALCULATE USTAR AND Z0NEW
C
   50    CONTINUE
         TSTAR = FK * DTHETA / A
         USTAR = FK * UM / 6
         Z0NEW = .00459*USTAR * USTAR
         IF (ITER .GT. 5 .AND. ABS( (USTAR - UST1)/UST1) .LT. EPS)
     1     GO TO 80
         UST1 = USTAR
         Z0 = Z0NEW
   60 CONTINUE
C
C IF COME HERE, TOO MANY ITERATIONS (UGH - UGH)
C
      WRITE(6,70)
   70 FORMAT('0TOO MANY ITERRTIONS ON Z0 IN SUBROUTINE UZL - CHECK',
     1       'METEOROLOGICAL DATA - PROGRAM TERMINATED')
      STOP
   80 CONTINUE
      Z0 = Z0NEW
      CD = (USTAR/UM) ** 2
      CH = FK * FK /(A*B)
   90 RETURN
      END
```

```
      FUNCTION UZ (Z,UM, CD, Z0, FL)
C
C PURPOSE:
C
C         TO DETERMINE WIND SPEED AT HEIGHT Z GIVEN WIND SPEED (UM)
C         FIND DRAG COEFFICIENT (CD) FROM ANOTHER HEIGHT AND THE
C         ROUGHNESS LENGTH (Z0) AND STABILITY LENGTH (FL) OF THE
C         PROFILE. USUALLY FUNCTION UZ IS USED AFTER SUBROUTINE UZL
C         IN ORDER TO FIND WIND SPEED AT A DIFFERENT HEIGHT THAN
C         THE OBSERVATION HEIGHT.
C
C  ALGORITHM:
C
C         THE WIND PROFILE IS ASSUMED TO CONFORM TO THE BUSINGER-
C         DYER FORMULRTION USED IN SUBROUTINE UZL.
C
C  ARGUMENTS:
C
C         Z - HEIGHT AT WHICH WIND SPEED 16 REQUIRED (METERS)
C         UM- WIND SPEED AT OBSERVATION HEIGHT (METERS PER SECOND)
C         CD - DRAG COEFFICIENT CORRESPONDING TO OBSERVATION HEIGHT
C         Z0 - ROUGHNESS LENGTH (METERS)
C         FL -STABILITY LENGTH (METERS)
C
C  HISTORY:
C
C         WRITTEN BY D. J SCHWAB, 1963, GLERL, ANN ARBOR, MI.
C         SEE SUBROUTINE UZL IN SCHWAB, BENNETT, AND JESSUP (1981)
C
      IF(FL.GT. 0.) GO TO 10
C
C  UNSTABLE PROFILE
C
      x1-15. /FL
      ARG1=(1.-X1*Z)**0.25
      ARG2=(1.-X1*Z0)**0.25
      B=ALOG((ARG1-1.)*(ARG2+1.)/((ARG1+1.)*(ARG2-0.9999999)))+
     1    2.*(ATAN(ARG1)-ATAN(ARG2))
      GO TO 30
C
C  STABLE SECTION
C
   10 IF(FL.LE.Z) GO TO 20
C
C  MILDLY STABLE PROFILE
C
      B=ALOG(Z/Z0)+4.7*(Z-Z0)/FL
      GO TO 30
C
C  STRONGLY STRBLE PROFILE
C
   26 CONTINUE
      ARG1=FL/Z0
      X1=ALOG(ARG1)
```

```
      X2=ALOG(Z/FL)
      ARG1=1.-1./ARG1
      B=X1+4.7*ARG1+5.7*X2
C
C  CALCULATE USTAR AND U Z
C
   30 CONTINUE
      USTAR=UM*SQRT(CD)
      UZ=USTAR*B/0.35
      RETURN
      END



      FUNCTION XDIST(RLAT,RLON)
C
C  PURPOSE :     TO RETURN X DISTANCE IN METER6 FROM THE GRID ORIGIN
C               DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN
C                LATITUDE AND LoN GITUDE
C  ARGUMENTS:
C                RLAT - LATITUDE OF POINT
C                RLON - LONGITUDE (WEST) OF POINT
C                RPARM, IPARM - ARRAYS CONTaININ BATHYMETRIC GRID
C                       PARAMETERS a 6 DESCRIBED I N SUBROUTINE RGRID
C
C  COMMON BLOCK:  /GPARM/ RPARM(23), IPARM(54)
C
      COMMON /GPARM/ RPARM(23), IPARM(54)
      P I = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * P I / 1 6 0 .
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - R L O N
C
C  F I N D XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C BATHYMETRIC G R I D
C
      X  P = RPARM(8) * D L O N  + RPARM(9) * DLAT + RPARM(10) * D L O N * DLAT +
     1      RPARM(11) * D L O N ** 2
      Y  P = RPARM(12) * D L D N  + RPARM(13) * DLAT + RPARM(14) * DLAT *
     1      DLON + RPARM(15) * D L O N ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      XDIST = (XP*COS(ALPHA) + YP*SIN(ALPHA)) • 1000.
C
C N O N TRANSLATE
C
      X D I S T = X D I S T  + IPARM(3) * RPARM(3)
      RETURN
      END
```

```
      FUNCTION YDIST(RLAT,RLON)
C
C PURPOSE :      TO RETURN Y DISTANCE IN METER6 FROM THE GRID ORIGIN
C                DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN
C                LATITUDE AND LONGITUDE
C  ARGUMENTS:
C                RLAT - LATITUDE OF POINT
C                RLON - LONGITUDE (WEST) OF POINT
C                RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                       PARAMETERS as DESCRIBED IN SUBROUTINE RGRID
C
C COMMON BLOCK:  /GPARM/ RPARM(23),IPARM(54)
C
      COMMON /GPARM/ RPARM (23), IPARM (54)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C FIND XPRIME,YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8) * DLON + RPARM(9) * DLAT + RPARM(10) * DLON * DLAT +
     1     RPARM(11) * DLON ** 2
      YP = RPARM(12) * DLON + RPARM(13) * DLAT + RPARM(14) * DLAT *
     1     DLON + RPARM(15) * DLON ** 2
C
C TRANSFORM TO 'UNPRIMED' SYSTEM
C
C FIRST ROTATE
C
      YDIST = (YP*COS(ALPHA) -XP*SIN(ALPHA)) * 1000.
C
C NOW TRANSLATE
C
      YDIST = YDIST + IPARM(4) * RPARM(3)
      RETURN
      END
```

```
      SUBROUTINE WOUTP(T, D, CG, XMOM, YMOM, SIG, CD, IDIM)
      DIMENSION D(IDIM,1),CG(IDIM,1),XMOM(IDIM,1),YMOM(IDIM,1),
     1 SIG(IDIM,1),CD(IDIM,1)
      DIMENSION DUM(41,35),CDUM(6)
      COMMON/CPARM/DT, TT, DADD
      COMMON/GPARM/RPARM(23),IPARM(54)
      COMMON/TPARM/NDTT
      DATA NR/0/
      PI=4.*ATAN(1.)
      z-10.
      IM=IPARM(1)
      JM=IPARM(2)
      DS=RPARM(3)
      DMIN=RPARM(5)+DADD
      IF(NR. EO. 0) THEN
       OPEN(11,ACCESS='DIRECT',STATUS='NEW',RECL=IM*JM)
       OPEN (12, ACCESS=' DIRECT', STATUS=' NEW', RECL=IM*JM)
       OPEN (13, ACCESS-' DIRECT', STATUS=' NEW', RECL=IM*JM)
      ENDIF
      NR=NR+1
      D O 1 0 I=1,IM
      D O 1 0 J=1,JM
      DUM(I,J)=-999.0
      IF(D(I,J).LT.DMIN) GO TO 10
      DUM(I,J)=4.*SIG(I,J)
   10 CONTINUE
C
C  SAVE SIGNIFICANT WAVEHEIGHT
C
      WRITE(11,REC=NR)((DUM(I,J),I=1,IM),J=1,JM)
      DO 20 I=1,IM
      D O 20 J=1,JM
      IF(D(I,J).LT.DMIN) GO TO 20
      DUM(I,J)=2.*PI*CG(I,J)/9.8
20    CONTINUE
C
C SAVE WAVE PERIODS
C
      WRITE(13,REC=NR)((DUM(I,J),I=1,IM),J=1,JM)
      DO 30 I-1,IW
      DO 30 J=1,JM
      IF(D(I,J).LT.DMIN) GO TO 30
      U=XMOM(I,J)
      V=YMOM(I,J)
```

```
C
C CONVERT DIR. TO METEOROLOGICAL CONVENTION AND SUBTRACT GRID ROTATION
C
      DUM(I,J) = 0.
      IF(U.EQ.0. .AND. V.EQ.0.) GOTO 30
      DUM(I,J)=ATAN2(-U,-V)*180./PI - RPARM(6) - RPARM(7)
      IF(DUM(I,J).LT.0.) DUM(I,J) = DUM(I,J) + 360
      IF(DUM(I,J).GT.360.) DUM(I,J) = DUM(I,J) - 360.
30    CONTINUE
C
C   SAVE WAVE DIRECTIONS
C
      WRITE(12,REC=NR)((DUM(I,J),I=1,IM),J=1,JM)
      RETURN
      END
      PROGRAM DSPLAY
C********************* PROGRAM DSPLAY *******************************
C
C PURPOSE: THE PURPOSE OF PROGRAM DSPLAY 16 TO DISPLAY THE RESULTS
C   OF THE WAVE PREDICTION SYSTEM ON a TIME-SHARING TERMINAL.
C   WAVE HEIGHT, DIRECTION AND PERIOD COMPUTED BY PROGRAM WAVE
C   CAN a UBE DISPLAYED as TWO-DIMENSIONAL FIELDS AT a GIVEN TIME.
C   THE TIME SERIES OF WAVE HElG HT, DIRECTION AND PERIOD AT a GIVEN
C   LOCATION CAN ALSO BE CALCULATED AND PRINTED.
C
C INPUT:
C   LOGICAL UNIT 5:
C    TIME-SHARING USER INPUT
C   LOGICAL UNIT 7:
C    BATHYMETRIC DATA FILE
C   LOGICAL UNIT 10:
C    CONTROL DATA FILE FOR PROGRAM WAVE
C   LOGICAL UNIT 11:
C    WAVE HEIGHT COMPUTED BY PROGRAM WAVE
C   LOGICAL UNIT 12:
C    WAVE DIRECTION COMPUTED BY PROGRAM WAVE
C   LOGICAL UNIT 13:
C    WAVE PERIOD COMPUTED BY PROGRAM WAVE
C   LOGICAL UNIT 15:
C    GENERAL INFORMATION FILE
C
C OUTPUT:
C   LOGICAL UNIT 6:
C    PROMPTING FOR TIME-SHARING USER AND DISPLAYS OF DEPTH,
C    WAVE HEIGHT, DIRECTION AND PERIOD
C   LOGICAL UNIT 15:
C    GENERAL INFORMATION FILE
C
```

```
C  SUBROUTINES:
C   RGRID - READS THE BATHYMETRIC DATA FILE
C   PRNTMP - FORMATS AND PRINTS MAPS OF WAVE HEIGHT, PERIOD
C    AND DIRECTION
C   FUNCTION RLAT - RETURNS LATITUDE GIVEN X AND Y DISTANCE
C    FROM ORIGIN
C   FUNCTION RLON - RETURNS LONGITUDE GIVEN X AND Y DISTANCE
C    FROM ORIGIN
C   FUNCTION XDIST - RETURN6 X DISTANCE FROM GRID ORIGIN
C    GIVEN LATITUDE AND LONGITUDE
C   FUNCTION YDIST - RETURN6 Y DISTANCE FROM GRID ORIGIN
C    GIVEN LATITUDE AND LONGITUDE
C   RDATE - THIS SUBROUTINE CALCULATES THE RELATIVE DATE AND TIME
C    FROM a GIVEN DATE AND TINE AND AN OFFSET IN HOURS
C
C  COMMON BLOCKS:
C   /CPARM/ DT, TT, DADD - CONTROL PARAMETERS FROM RLID
C   /GPARM/ RPARM(23), IPARM(54) - REAL AND INTEGER PARAMETERS
C    DESCRIBING THE BATHYMETRIC GRID. SEE SUBROUTINE RGRID.
C
C  HISTORY: WRITTEN BY E.W. LYNN, GLERL, NOVEMBER 1963
C
C
      CHARACTER*3 ANS
      DINENGION NSTIME(25)
      CHARACTER*80 LINE
      DIMENSION D(41,35),DUM1(41,35),DUM2(41,35),DUM3(41,35)
      COMMON /CPARM/ DT, TT, DADD
      COMMON /GPARM/ RPARM(23), IPARM(54)
C
C  DIMENSION6 OF D AND S
C
      DATA IDIM, JDIM/41,35/
C
C  LOGICAL UNIT NUMBERS
C
      DATA LUNI,LUNO,LUNB, LUNW, LUNH, LUND, LUNP, LUNX
     1     / 5,   6,   7,   16,   11,    12,    13,   15 /
C
C  ERROR SET ROUTINE FOR VAX FORTRAN TO DISABLE OUTPUT CONVERSION ERROR
C
      CALL ERRSET(63,.TRUE.,.FALSE.,.FALSE.,.FALSE.)
C
C  OPEN STATEMENT REQUIRED BY VAX FORTRAN
C
      OPEN (7,READONLY,STATUS='OLD')
```

```
C
C READ BATHYMETRIC GRID
C
      REWIND LUNB
      CALL RGRID(LUNB, D, IDIM, JDIM)
      IM=IPARM(1)
      JM=IPARM(2)
      DS=RPARM(3)
C
C READ CONTROL FILES
C
      REWIND LUNW
      READ(LUNW,*) DT, TT, DADD
      REWIND LUNX
      READ(LUNX,*)
      READ(LUNX,*)
      READ(LUNX, • )
      READ(LUNX,*) IY, IMON, IDAY, IH
      READ(LUNX,*) IHPREV, IHPOST, IHINC
      DMIN=RPARM(5)+DADD
C
C OPEN DATA FILES CONTAINING WAVE INFORMATION
C
      OPEN(11,ACCESS='DIRECT',STATUS='OLD',RECL=IM*JM)
      OPEN(12,ACCESS='DIRECT',STATUS='OLD',RECL=IM*JM)
      OPEN (13, ACCESS='DIRECT', STATUS='OLD', RECL=IM*JM)
C
c ASK FOR OPTION (ALL OPTIONS RETURN HERE EXCEPT 'Q')
C
  166 WRITE(LUNO,*)
     1'ENTER H IF YOU WANT TO SEE WAVE HEIGHT MAP'
      WRITE(LUNO,*)
     1'ENTER D IF YW WANT TO SEE WAVE DIRECTION MAP'
      WRITE(LUNO,*)
     1'ENTER P IF YOU WANT TO SEE WAVE PERIOD Map'
      WRITE(LUNO,*)'ENTER T IF YOU WANT a TIME SERIES OF WAVE HEIGHT,'
      WRITE(LUNO,*)'DIRECTION AND PERIOD AT a GIVEN LOCATION'
      WRITE(LUNO,*)'ENTER 0 IF YW WANT TO QUIT'
      ASSIGN 100 TO LABEL
      READ(LUNI,'(A)', END=5000, ERR=5000) ANS
```

```
C
C GEE WHICH COMMAND WAS GIVEN AND ASSIGN LOGICAL UNIT NUMBER
C
      IF(ANS.EQ.'H') THEN
       LUN = 11
       MP = 1
       GOT0 511
      ELSE IF (FINS. EQ. 'D') THEN
       LUN = 12
       MP = -1
       60T0 511
      ELSE IF(ANS. EO. 'P') THEN
       LUN = 13
       MP = 0
       GOT0 511
      ELSE IF(ANS.EQ.'T') THEN
       GOT0 801
      ELSE IF (ANS. EO. '0') THEN
       GOT0 900
      ELSE
       GOT0 100
      END IF
C
C   GET TIMES AFTER START TIME AT WHICH PLOT6 WILL BE GENERATED
C
  511 WRITE(LUNO,2000) IHINC, IHPOST
      IF(ANS.EQ.'H') THEN
       WRITE(LUNO,2100) IMON, IDAY, IY,I   H
      ELSE IF (ANS. EO. 'D') THEN
       WRITE(LUNO,2200) IMON, IDAY,I Y , I N
      ELSE
       WRITE(LUNO,2300) IMON, IDAY,I  Y  ,IH
      END I F
      WRITE(LUNO,*)
     1'YOU MAY MAKE UP TO 25 ENTRIES, ONE ENTRY PER LINE'
      WRITE(LUNO,*)'END THE LIST WITH -1'
C
      D  O 504 NUMPLT = 1, 25
  501 READ(LUNI,*,END=502,ERR=502) NSTIME(NUMPLT)
      IF(NSTIME(NUMPLT).EQ.-1) THEN
       60T0 503
      ELSE IF(NSTIME(NUMPLT). LT. IHINC. OR. NSTIME(NUMPLT). GT. IHPOST) THEN
       WRITE(LUNO,2400) IHINC, IHPOST
       GOTO 501
      ELBE
       GOT0 504
      END IF
  502 WRITE (LUNO,*)
     1' IMPROPER INPUT...PLEASE RE-ENTER, END LIST WITH -1'
      REWIND 5
      GOT0 501
  504 CONTINUE
  503 NUMPLT = NMPLT - 1
```

```
C
C PLOT THE WAVES FIT TIME INCREMENTS SPECIFIED BY 'NSTIME'
C
      DO 800 IJK = 1, NUMPLT
      XT = NSTIME(IJK)
C
C   PRINT RELATIVE DATE, REAL TIME RND UNITS
C
      CALL RDATE(IY, IMON, IDAY, IH, IFIX(XT), IYM, IMONM, IDAYM, IHM)
      IF(ANS .EQ. 'H') THEN
       WRITE(LUNO, 1700) IFIX (XT), IMONM, IDRYM, IYM, IHM
      ELSE IF(ANS . EQ. 'D') THEN
       WRITE(LUNO, 1800) IFIX(XT), I M O N M , IDAYM, IYM, IHM
      ELSE
       WRITE(LUNO, 1900) IFIX (XT), IMONM, IDAYM, IYM, IHM
      END IF
C
C GET WAVE SITUATION FROM PREVIOUS TIMESTEP
C
620   NR=IFIX(XT/DT)
      READ(LUN, REC=NR) ((DUM1(I, J), I=1, IM), J=1, JM)
C
C GET WAVE SITUATION FROM NEXT TIMESTEP
C
      NR=NR+1
      NUMREC=IFIX(TT/DT)
      IF(NR. GT. NUMREC) NR=NUMREC
      READ(LUN, REC=NR) ((DUM2(I, J), I=1, IM), J=1, JM)
C
C  INTERPOLATION FACTOR
C
      x w = (NR*DT - XT) / DT
C
C INTERPOLATE BETWEEN TIME STEPS
C
      DO 630 I = 1, IM
      DO 630 J = 1, JM
      IF(DUM1(I, J) . EQ. -999. ) T H E N
       DUM3(I, J) = -999.
       GOT0 630
      END IF
      IF(ANS .EQ. 'D') THEN
C
C  CALCULATE THE X, Y COMPONIENTS OF THE DIRECTIONS
C
       XA = COSD(DUM1(I, J))
       YA = SIND(DUM1(I, J))
       XB = COSD(DUM2(I, J))
       Y B = SIND(DUM2(I, J))
```

```
C
C INTERPOLATE BETWEEN X,Y COMPONIENTS OF DIRECTIONS
C
       X N = XW*XA + (1.-XW)*XB
       Y N = XW*YA + (1.-XW)*YB
C
C IF THE DIR. ARE 180 DEGREES OUT OF PHASE(XN=YN=0) THEN USE THE FIRST
C
       IF(XN .EQ. 0 . . FIND. YN . EQ. 0.) THEN
        DUM3(I,J) = DUM1(I,J)
       ELSE
C
C  CALCULATE THE DIRECTIONS IN DEGREES FROM X,Y COMPONIENTS
C
        DUM3(I, J) = ATAN2(YN, XN) * (180. /3.1415927)
       END IF
       IF(DUM3(I,J). L T . 0.) DUM3(I,J) = DUM3(I,J) + 3 6 0 .
      ELSE
       DUM3(I,J) = XW*DUM1(I,J) + (1.-XW)*DUM2(I,J)
      END IF
C
C  IF WAVE HEIGHTS THEN CONVERT TO FEET
C
      IF(ANS .EQ.'H')DUM3(I,J)=DUM3(I,J)  •  3.28
  630 CONTINUE
C
C PRINT THIS NEW INTERPOLATED ARRAY
C
      CALL PRNTMP(LUNO,DUM3,IDIM, IM, JM,-999.,MP)
  800 CONTINUE
C
C GET NEXT OPTION
C
      GOTO 100
C
C HERE FOR TIME SERIES OF WAVE HEIGHT, DIRECTION AND PERIOD
C ***********************************************************
C
  801 CONTINUE
150   WRITE(LUNO,*)
     1' INPUT LATITUDE OF POINT AT WHICH YOU WANT TO SEE TIME SERIES'
      WRITE(LUNO,*)' I  N      DEG, MIN,SEC      (E. G. 44,06,00)'
      WRITE(LUNO, • ) '  OR IN DECIMAL DEGREES (E.G. 44.1,0,0)'
      ASSIGN 150 TO LABEL
      READ(LUNI,*, END-5000, ERR-5000)  XLD, XLM, XLS
      XLAT=XLD+XLM/60.+XLS/3600.
```

```
C
C    CALCULATE THE DEGREES AND DECIMAL MINUTES
C
       XLATD = IFIX(XLAT)
       XLRTM = (XLAT-XLATD) * 60.
C
C    CHECK FOR NON-VALID LATITUDE ENTRY
C
       IF(XLM .LT. 0. .OR. XLM .GT. 60.) THEN
        WRITE(LUNO, *) XLM,
      1' IS NOT A VALID LATITUDE MINUTE RESPONSE'
        GO TO 150
       END IF
       IF(XLS .LT. 0. .OR. XLS .GT. 60.) THEN
        WRITE(LUNO, *) XLS,
      1' IS NOT A VALID LATITUDE SECOND RESPONSE'
        GO TO 150
       END IF
C
200    WRITE (LUNO,*)
      1' INPUT LONGITUDE OF POINT FIT WHICH YOU WANT TO SEE TIME SERIES'
       WRITE(LUNO,*)' I N DEG, MIN, SEC (E. G. 84, 18, 30)'
       WRITE(LUNO,*)' O R I N DECIMAL D E G R E E S (E.G. 84.35, 0, 0)'
       FISSION 200 TO LABEL
       READ(LUNI,*,END=5000,ERR=5000) XLD, XLM, XLS
       XLON=XLD+XLM/60.+XLS/3600.
C
C    CALCULATE THE DEGREES FIND DECIMAL MINUTES
C
       XLOND = IFIX(XLON)
       XLONM = (XLON - XLOND) * 6 0 .
C
C    CHECK FOR NON-VALID LONGITUDE ENTRY
C
       IF(XLM .LT. 0. .OR. XLM .GT. 60.) THEN
        WRITE(LUNO, *) XLM,
      1' IS NOT A VALID LONGITUDE MINUTE RESPONSE'
        GO TO 200
       END IF
       IF(XLS .LT. 0. .OR. XLS .GT. 60.) THEN
        WRITE(LUNO, *) XLS,
      1' IS NOT A VALID LONGITUDE SECOND RESPONSE'
        GO TO 200
       END IF
```

```
C
C CHECK IF TIME SERIES LOCATION IS WITHIN BATHYMETRIC GRID
C
      X=XDIST(XLAT,XLON)
      Y=YDIST(XLAT,XLON)
      I=IFIX(X/DS)+1
      J=IFIX(Y/DS)+1
      IF(I.LT.1.OR.I.GT.IM.OR.J.LT.1.OR.J.GT.JM) THEN
       WRITE(LUNO,*)'TIME SERIES LOCRTION:'
       WRITE(LUNO,1200)IFIX(XLATD),XLATM
       WRITE(LUNO,1300)IFIX(XLOND),XLONM
       WRITE(LUNO,*)'IS OUTSIDE OF GRID AREA'
       GO TO 150
      ENDIF
C
C CHECK IF TIME SERIES LOCATION IS ON LAND
C
      IF(D(I,J)+DADD.LT.DMIN) THEN
       WRITE(LUNO,*)'TIME SERIES LOCATION:'
       WRITE(LUNO,1200)IFIX(XLATD),XLATM
       WRITE(LUNO,1300)IFIX(XLOND),XLONM
       WRITE(LUNO,*)'IS NOT IN A LAKE GRID SQUARE'
       GO TO 150
      ENDIF
C
C   PRINT VALID LAT. LON. MESSAGE TO OUTPUT
C
      WRITE(LUNO,1200) IFIX(XLATD), XLATM
      WRITE(LUNO,1300)IFIX(XLOND),XLONM
C
C LOOP TO READ WAVE DATA RECORDS AND TO PRINT OUT TIME SERIES LOCATIONS
C
      WRITE(LUNO,1600)
      IREC = 0
      DO 700 ITIME = DT, TT, DT
      IREC = IREC + 1
      CALL RDATE(IY,IMON,IDAY,IH,ITIME,IYM,IMONM,IDAYM,IHM)
      READ(LUNH,REC=IREC) ((DUM1(II,JJ),II=1,IM),JJ=1,JM)
      READ(LUND,REC=IREC) ((DUM2(II,JJ),II=1,IM),JJ=1,JM)
      READ(LUNP,REC=IREC) ((DUM3(II,JJ),II=1,IM),JJ=1,JM)
      DUM1(I,J) = DUM1(I,J) + 3.26
      WRITE(LUNO,1500) ITIME, IMONM, IDAYM, IYM, IHM,
     1  DUM1(I,J), DUM2(I,J), DUM3(I,J)
  700 CONTINUE
C
C  GET NEXT OPTION
C
      GO TO 100
```

```
C
C    FISK FOR COMMENTS OR SUGGESTIONS THEN QUIT
C
  900 WRITE (LUNO,*)
     1'PLEASE TYPE AS MANY LINES AS YOU WISH IF YW HAVE ANY',
     2'COMMENTS OR SUGGESTIONS CONCERNING THIS WAVE MODEL'
      WRITE(LUNO,*)'END WITH A CARRIAGE RETURN'
  910 READ(LUNI,1400,END=920,ERR=920) LINE(1:80)
      IF(LINE .EQ.'') GOT0 920
      WRITE(LUNX, 1450) LINE(1:80)
      GOT0 910
  920 WRITE(LUNX,*)'****************'
      WRITE(LUNO,*)'GOODBYE!'
      GOT0 99999
1280 FORMAT(' LRTITUDE ',I3,' DEG ',F4.1,' MIN')
1300 FORMAT(' LONGITUDE ',I3,'DEG ',F4.1,' MIN')
1400 FORMAT(A80)
1450 FORMAT('',A80)
1500  FORMAT(' ',15,3X, 12,'/', 12,'/',I4,2X, 13,':00',
     19X, F4.1,13X,F4.0,14X, F4.1)
1600 FORMAT('HOUR        DATE       TIME    WAVE HEIGHT(FT)',
     1'    WAVE DIR.(DEG)    WAVE PERIOD(SEC)')
1790 FORMAT('WAVE HEIGHTS IN TENTHS OF FEET AT',I3,' HOURS',
     1I2,'/',I2,'/',I4,' ',I2,':00')
1800 FORMAT('WAVE DIRECTIONS IN TENS OF DEGREES AT',13,' HOURS',
     1I2,'/',I2,'/',I4,' ',I2,':00')
1900 FORMAT('WAVE PERIODS IN SECONDS AT',13,'HOURS',
     1I2,'/',I2,'/',I4,' ',I2,':00')
2000 FORMAT(' ENTER THE TINE(S) IN HOURS (',I2,'-',I2,')',
     1' AFTER THE START TINE')
2100  FORMAT('',I2,'/',I2,'/',14,'',12,':00',
     1' AT WHICH YW WANT TO SEE WAVE HEIGHTS')
2200  FORMAT('',12,'/',12,'/',14,'',12,':00',
     1' FIT WHICH YOU WANT TO SEE WAVE DIRECTIONS')
2300  FORMAT('',12,'/',I2,'/',14,'',12,':00',
     1' FIT WHICH YW WANT TO SEE WAVE PERIODS')
2400 FORMAT(' TIME SPECIFIED IS NOT IN THE RANGE(',
     112,'-',12,') RE-ENTER')
C
C COME HERE FOR ERROR CONDITION ON INPUT
C
5000 REWIND LUNI
      WRITE(LUNO, • )' IMPROPER INPUT...PLEASE RE-ENTER'
      GO TO LABEL,(100)
99999 CONTINUE
      END
```

```
      SUBRWTINE RGRID(LUN, D, IDIM, JDIM)
C PURPOSE:
C                   TO READ A STANDARD BRTHYMETRIC GRID DATA FILE
C                   AND RETURN GRID PARAMETERS AND DEPTHS.
C ARGUMENTS:
C ON INPUT:
C                   LUN - LOGICAL UNIT NUMBER OF BATHYMETRIC DATA FILE
C                   IDIM - FIRST DIMENSION OF ARRAY D IN
C                          DIMENSION STATEMENT OF CALLING PROGRAM
C                   JDIM - SECOND DIMENSION OF ARRAY D IN
C                          DIMENSION STATEMENT O F CALLING PROGRAM
C ON OUTPUT:
C                    D - DEPTH ARRAY. ZERO FOR LAND, AVERAGE DEPTH
C                        OF GRID BOX IN METERS FOR WATER.
C                    RPARM - ARRAY CONTAINING REAL-VALUED BATHYMETRIC
C                            GRID PARAMETERS AS FOLLOWS:
C                       1.  BASE LATITUDE
C                       2.  BASE LONGITUDE
C                       3.  GRID SIZE (M)
C                       4.  MAXIMUM DEPTH (M)
C                       5.  MINIMUM DEPTH (M)
C                       6.  BASE ROTATION (COUNTERCLOCKWISE FROM E-U)
C                       7.  ROTATION FRON BASE (COUNTERCLOCKWISE)
C                       S-11. GEOGRAPHIC-TO-MAP COORDINATE CONVERSION
C                             COEFFICIENTS FOR X
C                       12-15.  GEOGRAPHIC-TO-MAP COORDINATE CONVERSION
C                               COEFFICIENTS FOR Y
C                       15-19.  MAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                               COEFFICIENTS FOR LONGITUDE
C                       20-23.  MAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                               COEFFICIENTS FOR LATITUDE
C                   IPARM - ARRAY CONTAINING INTEGER-VALUED BRTHYMETRIC
C                           GRID PARAMETERS AS FOLLOWG:
C                       1.  NUMBER OF GRID BOXES IN X DIRECTION
C                       2.  NUMBER OF GRID BOXES IN Y DIRECION
C                       3.  I DISPLACEMENT - THE NUMBER OF NEW GRID
C                           SQUARES IN THE X-DIRECTION FROM TM NEW
C                           SRID ORIGIN TO THE OLD GRID ORIGIN
C                       4.  J DISPLACEMENT - THE NUMBER OF NEW GRID
C                           SQUARES IN THE Y-DIRECTION FROM THE NEU
C                           GRID ORIGIN TO THE OLD GRID ORIGIN
C                       5-54.  LAKE NAME (50A1)
C                    NOTE: IF GRID IS TOO LARGE FOR DIKNSIONS OF D,
C                          THE IPARM ARRAY IS SET TO ZERO
C
C COMMON BLOCK:
C                   /GPARM/RPARM(23), IPARM(54)
C
```

```
      DIMENSION D(IDIM,JDIM)
      COMMON /GPARM/ RPARM(23),IPARM(54)
      READ (LUN,30) (IPARM(I),I=5,54), IPARM(1), IPARM(2),
     1(RPARM(I),I=1,6), IPARM(3), IPARM(4), RPARM(7)
      RERD(LUN,60)(RPARM(I),I=8,23)
      IN=IPARM(1)
      JM=IPARM(2)
      IF(IPARM(1).GT.IDIM.OR.IPARM(2).GT.JDIM)GO TO 10
      RERD(LUN,40)((D(I,J),I=1,IM),J=1,JM)
      RETURN
   10 DO 20 I=1, 54
   20 IPARM(I)= 0
      WRITE(6,50)
   30 FORMAT(50(A1)/2I5,2F12.7, 3FS.0, F6.2, 215, F6.2)
   40 FORMAT(19F4.0,4X)
   50 FORMAT('BATHYMETRIC GRID TOO LARGE - INCREASE DIMENSIONS OF',
     1       'NOEPTH AND DEPTH IN MAIN PROGRAM')
   60 FORMAT(4E15.6, 20X)
   70 RETURN
      END



      FUNCTION RLAT(X,Y)
C
C PURPOSE:  TO RETURN LATITUDE OF a POINT ON THE GRID
C                   DESCRIBED BY THE COMNON BLOCK /GPARM/, GIVEN THE
C                   X AND Y DISPLACEMENTS FROM THE GRID ORIGIN
C ARGUMENTS:
C                   x - X DISTANCE FROM THE GRID ORIGIN (M)
C                   Y - Y DISTANCE FROM THE GRID ORIQIN (M)
C                   RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                          PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C COMMON BLOCK: /GPARM/RPARM(23),IPARM(54)
C
      COMMON/GPARM/RPARM(23),IPARM(54)
      PI=ATAN2(0.,-1.)
      ALPHA=RPARM(7) * PI/180.
C
C TRANSFORMTN POINTS TO THE 'PRIMED' COORDINATE SYSTEM,
C IE., THAT OF THE STANDARD BATHYMETRIC GRID
C
C FIRST TRANSLATE
C
      XX = x - IPARM(3) * RPARM(3)
      YY = Y - IPARM(4) * RPARM(3)
```

```
C
C  NOW ROTATE
C
      XP=(XX*COS(ALPHA)-YY*SIN(ALPHA))/1000.
      YP=(YY*COS(ALPHA)-XX*SIN(ALPHA))/1000.
      DLAT = RPARM(20)  *  XP + RPARM(21) * YP + RPARM(22) * XP * YP +
     1       RPARM(23)  *  XP ** 2
      RLAT = RPARM(1) + DLAT
      RETURN
      END
      FUNCTION RLON(X,Y)
C
C  PURPOSE:  TO RETURN LONGITUDE OF A POINT ON THE GRID
C                    DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN THE
C                    X AND Y DISPLACEMENTS FROM THE GRID ORIGIN
C  ARGUMENTS:
C                    X - X DISTANCE FROM THE GRID ORIGIN (M)
C                    Y - Y DISTANCE FROM THE GRID ORIGIN (M)
C                    RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                           PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C  COMMON BLOCK: /GPARM/RPARM(23),IPARM(54)
C
C
      COMMON/GPARM/RPARM(23),IPARM(54)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
C
C  TRANSFORM THE POINTS TO THE 'PRIMED' COORDINATE SYSTEM,
C  IE., THAT OF THE STANDARD BATHYMETRIC GRID
C
C  FIRST TRANSLATE
C
      XX = X - IPARM(3) * RPARM(3)
      YY = Y - IPARM(4) * RPARM(3)
C
C   ROTATE
C
      XP=(XX*COS(ALPHA)-YY*SIN(ALPHA))/1000.
      YP=(YY*COS(ALPHA)+XX*SIN(ALPHA))/1000.
      DLON = RPARM(16)  *  XP + RPARM(17) * YP + RPARM(18) * XP * YP +
     1       RPARM(19)  *  XP ** 2
      RLON = RPARM(2) - DLON
      RETURN
      END
```

```
      FUNCTION XDIST(RLAT,RLON)
C
C  PURPOSE :      TO RETURN X DISTANCE IN METERS FROM TN GRID ORIGIN
C                 DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN
C                 LaTITUDE AND LONGITUDE
C  ARGUMENTS:
C                 RLAT - LaTITUDE OF POINT
C                 RLON - LONGITUDE WEST) OF POINT
C                 RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC G R I D
C                        PARAMETERS a s DESCRIBED I N SUBROUTINE RGRID
C
C  COMMON BLOCK:  /GPARM/ RPARM(23),IPARM(54)
C
      COMMON /GPARM/ RPARM(23), IPARM(54)
      P I = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * P  I / 180.
      DLAT = RLAT - RPARM(1)
      D L O N = RPARM(2) - R L O N
C
C FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      X P = RPARM(8)  •  D L O N + RPARM(9)  •  M A T + RPARM(10) * D L O N * DLAT +
     1      RPARM(11)  •  DLON ** 2
      Y P = RPARM(12) * D L O N + RPARM(13)  •  DLAT + RPARM(14)  •  DLAT  •
     1      D L O N + RPARM(15) * D L O N •  * 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C F I R S T ROTATE
C
      X D I S T = (XP*COS(ALPHA) + YP*SIN(ALPHA)) * 1000.
C
C  N W TRANSLATE
C
      X D I S T = X D I S T  + IPARM(3) * RPARM(3)
      RETURN
      END
      FUNCTION YDIST(RLAT,RLON)
C
C  PURPOSE :      TO RETURN Y DISTANCE IN METERS FROM THE GRID ORIGIN
C                 DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN
C                 LaTITUDE AND LONGITUDE
C  ARGUMENTS:
C                 RLAT - LATITUDE OF POINT
C                 RLON - LONGITUDE (WEST) OF POINT
C                 RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC G R I D
C                        PARAMETERS a s DESCRIBED I N sU B R O UTI N E RGRID
C
C  COMMON B L O C K : /GPARM/ RPARM(23),IPARM(54)
C
```

```
      COMMON /GPARM/ RPARM(23), IPARM(54)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI/180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8) * DLON + RPARM(9) * DLAT + RPARM(10) * DLON *DLAT +
     1     RPARM(11)* DLON * t 2
      YP = RPARM(12) * DLON + RPARM(13) * DLAT + RPARM(14)* DLAT *
     1     DLON + RPARM(15)* DLON * * 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      YDIST = (YP*COS(ALPHA) -XP*SIN(ALPHA)) *
C
C  N W TRANSLATE
C
      YDIST = YDIST + IPARM(4) * RPARM(3)
      RETURN
      END
      SUBRWTINE RDATE(IY,MON,IDAY, IH, IHINC, IYM, MONM,IDAYM,IHM)
C
C PURPOSE:      THIS SUBROUTINE ADJUSTS TINE BY INCREMENTING OR
C               DECRENENTINS THE TINE ARGUMENTS(YEAR, MON., DAY, HOUR)
C               BY THE SPECIFIED NUMBER OF HOURS SPECIFIED BY 'IHINC'.
C
C ARGUMENTS:
c  ON INPUT:    IY - STaRTING YEAR(NO LIMIT RESTRICTION)
C               MON - STaRTING MONTH(1-12)
C               IDAY - STaRTING DAY(1 - # OF DAY IN PARTICULAR MONTH)
C               IH - STARTING HOUR(0-23)
C               IHINC - NUMBER OF HOURS TO INCRERENT OR DECREMENT TIME.
C                       POSITIVE FOR TIME ADVANCE
C                       NEGITIVE FOR TIM RETRACT
C                       NO LIMIT ON THE NUMBER OF HOURS TO CWNGE TINE.
C
C ON OUTPUT:    IYM - THE ADJUSTED YEAR(NO LIMIT RESTRICTION)
C               MOW-TH E ADJUSTED MONTH (1-12)
C               IDAYM - THE ADJUSTED DAY (1 - # O F DAYS I N PARTICULAR MONTH)
C               IHM - THE ADJUSTED HWR (0-23)
C
C HISTORY:      WRITTENBY E. W. LYNN S-1983, GLERL, ANN ARBOR MI.
C
```

```
      DIMENSION NUMDAY(12)
      DATA NUMDAY/31,28,31,30,31,30,31,31,30,31,30,31/
C
C   INITIALIZE THE OUTPUT PARAMETERS TO THE INPUT PARAMETERS
C   AND ADD THE HOUR INCREMENT 'IHINC' TO THE HWR TIME
C
      IYM = IY
      MONM = MON
      IDAYM = IDAY
      IHM = IH + IHINC
C
C   ADJUST THE NUMBER OF DAYS IN FEB. DEPENDING IF LEAP YEAR.
C
   10   IF(MOD(IYM,4).EQ.0) THEN
          NUMDAY(2) = 27
        ELSE
          NUMDAY(2) = 28
        END IF
C
C   GO TO THE REQUIRED ROUTINE DEPENDING ON WEATHER TIME NEEDS
C   INCREMENTING OR DECREMENTING.
C
      IF(IHINC.GE.0) THEN
C
C   ADJUST ADVANCED TIME(IYM,MONM,IDAYM,IHM).
C
        IF(IHM.GT.23) THEN
          IHM = IHM - 24
          IDAYM = IDAYM + 1
          IF(IDAYM.GT.NUMDAY(MONM)) THEN
            IDAYM = IDAYM - NUMDAY(MONM)
            MONM = MONM + 1
            IF(MONM.GT.12) THEN
              MONM = MONM - 12
              IYM = IYM + 1
            END IF
          END IF
        END IF
        IF(IHM.GT.23) GOTO 10
      ELSE
C
C   ADJUST BACKWARD TIK(IYM,MONM,IDAYM,IHM)
C
        IF(IHM.LT.0) THEN
          IDAYM = IDAYM - 1
          IHM = IHM + 24
          IF(IDAYM.LT.1) THEN
          I = MONM - 1
          IF(I.EQ.0) I = 12
            IDAYM = IDAYM + NUMDAY(I)
```

```
            MONM = MONM - 1
            IF(MONM . LT. 1) THEN
              IYM = IYM - 1
              MONM = MONM + 12
            END IF
          END IF
        END IF
        IF(IHM .LT. 0) GOT0 10
      END IF
      RETURN
      END
      SUBROUTINE PRNTMP(LUN, a, IDIM, IMAX, JMAX, SPVAL, MP)
C
C PURPOSE:
C                     TO NORMALIZE AND PRINT THE TWO DIMENSIONAL ARRAY a
C
C ALGORITHM:
C                     DATA ARE NORMALIZED BY 10**MP AND THEN
C                     FORMATTED as 3 DIGIT INTEGERS. POINTS AT WHICH
C                     A(I,J) IS EQUAL TO SPVAL ARE MASKED BY ASTERISKS.
C                     DATA ARE OUTPUT IN BLOCKS WITH J DECREASING DOWN
C                     AND I INCREASING ACROSS THE PAGE. THE NUMBER OF
C                     VALUES PRINTED ACROSS a PAGE IS AN INTERNAL PARA-
C                     METER IN THE SUBRWTINE.
C ARGUMENTS:
C                     LUN - LOGICAL UNIT NUMBER ON WHICH TO PRINT
C                     A - TWO-DIMENSIONAL ARRAY TO BE NORMALIZED AND
C                        PRINTED. UNCHANGED BY PRNT.
C                     IDIM - FIRST DIMENSION OF a AND IN DIMENSION
C                        STATEMENT OF CALLING PROGRAM
C                     IMAX - MAXIMUM I VALUE ACTUALLY USED IN a AND D
C                     JMAX - MAXIMUM J VALUE ACTUALLY USED IN a AND D
C                     SPVAL - SPECIAL MASKING VALUE: ONLY THE A(I,J)
C                             WHICH ARE NOT EQUAL TO SPVAL ARE PRINTED
C                     MP - POWER OF TEN BY WHICH WE MUST MULTIPLY ELEMENTS
C                          OF a so THAT IT FALLS BETWEEN 100 AND 1000.
C
C
      DIMENSION INTEG(30), A(IDIM,43)
C
C NCOL IS THE NUMBER OF VALUES TO PRINT ACROSS a PAGE
C
      NCOL = 26
C
C PRINT THE GRID
C
      I1 = 1
      II = (IMAX - 1) / NCOL + 1
      IRMDR = IMAX - NCOL * (II-1)
      DO 50 L = 1, II
```

```
C
C WHEN L=II ONLY PRINT IRMDR VALUES
C
      IF(L.EQ.II) NCOL = IRMDR
C
C PRINT THE POWER
C
      I2 = 11 + NCOL - 1
      DO 40 JJ = 1, JMAX
        J = JMAX - JJ + 1
        DO 30 I = I1,I2
          I3 = 1 + I - I1
          INTEG(I3) = -9999
          IF(A(I,J).NE. SPVAL) INTEG(I3) = INT(A(I,J)*10.**MP +
     1    SIGN(0.5,A(I,J)*10.**MP))
   30   CONTINUE
        WRITE(LUN,70)(INTEG(I), I=1,NCOL)
   40 CONTINUE
      I1 = I2 + 1
      WRITE(LUN,*) ' '
   50 CONTINUE
   70 FORMAT('',2613)
      RETURN
      END
```