REFRACT: A Refraction Program for Water Waves

Version 2.0

Robert A. Dalrymple

RESEARCH REPORT NO. CACR-91-09

September, 1991

Center for Applied Coastal Research
University of Delaware
Newark, DE 19716

REFRACT: A Refraction Program for Water Waves

Version 2.0

Robert A. Dalrymple

Center for Applied Coastal Research
University of Delaware Newark, DE 19716

1 Introduction and Summary

REFRACT is a program to compute the wave heights and direction over an offshore area, including the effects of wave shoaling, refraction, wave-current interaction and nonlinearity. The theoretical basis of the program is described in a companion paper by R.A. Dalrymple, "A Model for the Refraction of Water Waves," *Journal of Waterway, Port, Coastal and Ocean Eng.*, ASCE, 114, 4, 1988 . The two principles are the irrotationality of the wave number (which for planar bathymetry reduces to Snell's Law) and the conservation of wave action, which, in the absence of currents, becomes the conservation of wave energy. In this version 2.0, the finite difference method has been changed to a two-step Lax-Wendroff procedure to remove some directional bias inherent in the previous method.

REFRACT operates over a given bathymetry, provided in the form of a rectangular grid of size M x N in the x and y directions respectively, where the x direction is the onshore direction. Wave data must be specified either at the offshore grid row or in deep water. The data consists of the wave period, the wave amplitude (half the wave height), and the wave angle. These wave data can be provided in terms of deep water data (and the program assumes Snell's law shoaling to the offshore grid row) or as data valid for the offshore grid row. All of the input data are supplied to REFRACT via two files, INDAT.DAT and REFDAT.DAT. Note: All data is in the metric system.

The output of the program is displayed on the screen as the data are generated, plus they are stored in two disk files. The file, HEIGHT.DAT, contains the wave height data for the bathymetric grid and the file, ANGLE.DAT, contains the corresponding wave angle data in radians.

This manual provides an explaination of data required in the input files and describes the format of the output files. Two example problems are presented for to describe the workings of the model.

Contents

2 Theoretical Background

Waves propagating across variable bathymetry change direction due to differences in wave speed along the wave crest. If a coordinate system is chosen with x onshore and y alongshore, then a wave train can be described by

$$\eta(x,y,t) = a(x,y)\cos(k\cos\theta x + k\sin\theta y - \sigma t) \tag{1}$$

where k is the wave number ($= 2\pi/L$, where L is the wavelength), σ is the angular frequency of the wave, ($\sigma = 2\pi/T$, where T is the given wave period), and θ is the angle made by the wave propagation direction and the x axis. The wave number satisfies the following dispersion relationship, from linear wave theory (see Dean and Dalrymple, 1984),

$$\sigma^2 = gk\tanh kh \tag{2}$$

which relates the wave frequency, wave number and water depth, h.

The wave number, k, is irrotational, which leads to a general form of Snell's Law for refraction.

$$\frac{\partial k\sin\theta}{\partial x} - \frac{\partial k\cos\theta}{\partial y} = 0 \tag{3}$$

This equation relates changes in wave direction to changes in wave speed, through the wave number change. This equation can be rewritten as

$$\frac{\partial A}{\partial x} - \frac{\partial B}{\partial y}, \tag{4}$$

where B is defined as

$$B = k\cos\theta = \sqrt{k^2 - A^2}.$$

If the x axis is divided into segments, $x = i\Delta x$, $i = 0,1,2,...,m$, and if the y axis is subdivided as $y = j\Delta y$, $j = 0,1,2,...,n$, then finite difference methods can be used to solve the above equation for $A(x,y) \rightarrow A_{i,j}$, given that θ is known on the offshore grid row (i=0). The solution is marched in from this offshore row in the increasing i direction. Dalrymple (1988) solved the equation using a second order accurate scheme which was asymmetric, leading to minor errors in symmetric problems. This asymmetry led to the adoption of a new algorithm, based on the two-step Lax-Wendroff method.

The procedure first involves determining the determination of a mid-grid value of $A_{i+1/2,j+1/2}$ from the governing equation,

$$\frac{A_{i+1/2,j+1/2} - (A_{i,j+1} + A_{i,j})/2}{\Delta x/2} - \left(\frac{B_{i,j+1} - B_{i,j}}{\Delta y}\right) = 0 \tag{5}$$

Solving for $A_{i+1/2,j+1/2}$,

$$A_{i+1/2,j+1/2} = \frac{A_{i,j+1} + A_{i,j}}{2} + \frac{\Delta x}{2\Delta y}(B_{i,j+1} - B_{i,j}) \tag{6}$$

From $A_{i+1/2,j+1/2}$, $B_{i+1/2,j+1/2}$ is determined.

Then the second step is made to find $A_{i+1,j}$,

$$\frac{A_{i+1,j} - A_{i,j}}{\Delta x} - \frac{B_{i+1/2,j+1/2} - B_{i+1/2,j-1/2}}{\Delta y} = 0 \tag{7}$$

Finally, we obtain

$$A_{i+1,j} = A_{i,j} + \frac{\Delta x}{\Delta y}\left(B_{i+1/2,j+1/2} - B_{i+1/2,j-1/2}\right) \tag{8}$$

The two-step method is also second order accurate.

3 Operation

3.1 Input Files

The program requires two data files to run: INDAT.DAT, which contains wave data, and REFDAT.DAT, which contains the bathymetric (and current) data. These files must be constructed by the user prior to running REFRACT. The format of these files is as follows:

INDAT.DAT (input file):

IUN1,IUN2,IUN3,IUN4
M,N,IDEEP
IU,NONLIN,ICUR,IBC
DX,DY
T,TIDE
AMP,THETA0

These parameters are defined as:

IUN1,IUN2,IUN3,IUN4

Four logical unit numbers for reading/writing files. IUN1 corresponds to the file, REFDAT.DAT, IUN2 corresponds to the output file, HEIGHT.DAT, and IUN3 is the screen (0 for PC, 6 for Sun), IUN4 is the number for the output file, ANGLE.DAT. Example values are (1,2,0,4). The value 3 must not be used on the PC. The format for these data is 4I4.

M,N,IDEEP

M is the number of grid rows in the x (onshore) direction, N the number of columns (alongshore), IDEEP is the switch that tells the program that deep water wave data are being used (IDEEP equal to 0) or that the wave data are for the most offshore row (IDEEP equal to 1). Format is 3I4.

IU,NONLIN,ICUR,IBC

IU is the units switch (1 for metric, 2 for English system), NONLIN is the nonlinearity switch (0, linear model, 1 for nonlinear), ICUR (set to 1 if mean currents are provided), IBC is a boundary condition switch (0 for transmitting, 1 for reflecting). For almost all cases, it is best to begin modelling with IBC=1, so that the ,maximum effect of the boundaries can be ascertained. Then for final runs IBC can be set to 0. Format is 4I4.

DX,DY

DX, DY are the grid spacings in the x and y directions. Format is 2F8.4.

T,TIDE

Wave period and the tidal stage (which is added to the depths provided in REFDAT.DAT). Specification of the tidal stage permits the calculation of wave refraction for various tidal stages without the necessity of changing the REFDAT.DAT file. Format is 2F8.4.

AMP,THETA0

Wave amplitude and direction (degrees). The direction is measured from the x axis towards the y axis. Angles are restricted to the range -90° to 90° . Format is 2F8.4.

REFDAT.DAT (input file):

{D(i,j)}
{U(i,j)}
{V(i,j)}

where these arrays are defined as follows.

{D(i,j)}

Formatted depth grid, with the data supplied first in the longshore direction (j) and then the onshore direction (i), beginning with the offshore grid row. The program reads the data row by row as in the following section of code.

```
      DO 2 I=1,M
          READ(IUN1,10) (D(I,J),J=1,N)
    2 CONTINUE
 10   FORMAT(16F9.2)
```

If currents are used in the calculation (ICUR = 1), then the current field must be input as well. Otherwise no additional data is necessary.

{U(i,j)}

Onshore current grid (only if ICUR = 1). Formatted the same as the depths.

{V(i,j)}

Longshore current grid (only if ICUR = 1). Formatted the same as the depths.

3.2 Output Files

The program generates wave heights and corresponding wave angles at each of the depth locations from offshore to inshore, printing the results at each row on to the screen. In addition, wave height and angle data are written to output files, HEIGHT.DAT and ANGLE.DAT.

HEIGHT.DAT (output file)

These data consist of the wave height at all rows, from offshore to inshore: (H(i,j),j=1,N) in 16F9.2 format. Note that the file must be present (if only empty) prior to starting the program. An empty file can be created in the following way in DOS:

```
C> COPY CON:  HEIGHT.DAT
   (blank line)
   control F6
```

ANGLE.DAT (output file)

Another output file, which contains the calculated wave angles by rows. The data file must exist prior to running the program. The wave angles are in degrees. From offshore to inshore, the wave angles are (TH(i,j), j=1,N) in 16F9.2 format.

4 Example Problem 1.

The simple case of a wave propagating over a plane parallel contour line ispresented. This case is for a shallow water situation, with the depthsshoaling from 0.45 m to 0.074 m. over a distance of 7.75 m. The input waveheight is 0.08 m. and the wave period is 1.4 s. The incident wave angle is 15° .

The INDAT.DAT file for this example:

```
 1   2   0   4              IUN values
28   8   1                  M,N, IDEEP
 1   0   0   0              IU,NONLIN,ICUR,IBC
 .25     .25                DX,DY
1.4      0.0                PERIOD, TIDE
 .04    15.0                AMPLITUDE, DIRECTION (Degrees)
```

The input REFDAT.DAT file:

.45	.45	.45	.45	.45	.45	.45	.45
.43	.43	.43	.43	.43	.43	.43	.43
.42	.42	.42	.42	.42	.42	.42	.42
.41	.41	.41	.41	.41	.41	.41	.41
.40	.40	.40	.40	.40	.40	.40	.40
.40	.40	.40	.40	.40	.40	.40	.40
.38	.38	.38	.38	.38	.38	.38	.38
.37	.37	.37	.37	.37	.37	.37	.37
.36	.36	.36	.36	.36	.36	.36	.36
.35	.35	.35	.35	.35	.35	.35	.35
.33	.33	.33	.33	.33	.33	.33	.33
.32	.32	.32	.32	.32	.32	.32	.32
.31	.31	.31	.31	.31	.31	.31	.31
.30	.30	.30	.30	.30	.30	.30	.30
.28	.28	.28	.28	.28	.28	.28	.28
.27	.27	.27	.27	.27	.27	.27	.27
.26	.26	.26	.26	.26	.26	.26	.26
.25	.25	.25	.25	.25	.25	.25	.25
.23	.23	.23	.23	.23	.23	.23	.23
.22	.22	.22	.22	.22	.22	.22	.22
.21	.21	.21	.21	.21	.21	.21	.21
.20	.20	.20	.20	.20	.20	.20	.20
.18	.18	.18	.18	.18	.18	.18	.18
.17	.17	.17	.17	.17	.17	.17	.17
.16	.16	.16	.16	.16	.16	.16	.16
.15	.15	.15	.15	.15	.15	.15	.15
.13	.13	.13	.13	.13	.13	.13	.13
.12	.12	.12	.12	.12	.12	.12	.12

After execution of the program, the HEIGHT.DAT file, which contains the wave heights, is:

```
          28          8      0.2500000      0.2500000
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.08     0.08     0.08     0.08     0.08     0.08     0.08     0.08
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
 0.09     0.09     0.09     0.09     0.09     0.09     0.09     0.09
```

The file, ANGLE.DAT, contains (in degrees):

```
          28               8           0.2500000           0.2500000
   15.00     15.00     15.00     15.00     15.00     15.00     15.00     15.00
   14.77     14.77     14.77     14.77     14.77     14.77     14.77     14.77
   14.66     14.66     14.66     14.66     14.66     14.66     14.66     14.66
   14.54     14.54     14.54     14.54     14.54     14.54     14.54     14.54
   14.41     14.41     14.41     14.41     14.41     14.41     14.41     14.41
   14.41     14.41     14.41     14.41     14.41     14.41     14.41     14.41
   14.15     14.15     14.15     14.15     14.15     14.15     14.15     14.15
   14.02     14.02     14.02     14.02     14.02     14.02     14.02     14.02
   13.88     13.88     13.88     13.88     13.88     13.88     13.88     13.88
   13.73     13.73     13.73     13.73     13.73     13.73     13.73     13.73
   13.43     13.43     13.43     13.43     13.43     13.43     13.43     13.43
   13.28     13.28     13.28     13.28     13.28     13.28     13.28     13.28
   13.11     13.11     13.11     13.11     13.11     13.11     13.11     13.11
   12.95     12.95     12.95     12.95     12.95     12.95     12.95     12.95
   12.60     12.60     12.60     12.60     12.60     12.60     12.60     12.60
   12.42     12.42     12.42     12.42     12.42     12.42     12.42     12.42
   12.23     12.23     12.23     12.23     12.23     12.23     12.23     12.23
   12.03     12.03     12.03     12.03     12.03     12.03     12.03     12.03
   11.62     11.62     11.62     11.62     11.62     11.62     11.62     11.62
   11.41     11.41     11.41     11.41     11.41     11.41     11.41     11.41
   11.18     11.18     11.18     11.18     11.18     11.18     11.18     11.18
   10.95     10.95     10.95     10.95     10.95     10.95     10.95     10.95
   10.46     10.46     10.46     10.46     10.46     10.46     10.46     10.46
   10.20     10.20     10.20     10.20     10.20     10.20     10.20     10.20
    9.93      9.93      9.93      9.93      9.93      9.93      9.93      9.93
    9.64      9.64      9.64      9.64      9.64      9.64      9.64      9.64
    9.04      9.04      9.04      9.04      9.04      9.04      9.04      9.04
    8.71      8.71      8.71      8.71      8.71      8.71      8.71      8.71
```

5 Example Problem 2.

As an illustration of the new numerical algorithm, waves normally incident on a triangularly shaped offshore sand bar were studied. As shown below the resulting wave heights and directions are symmetric about the submerged point.

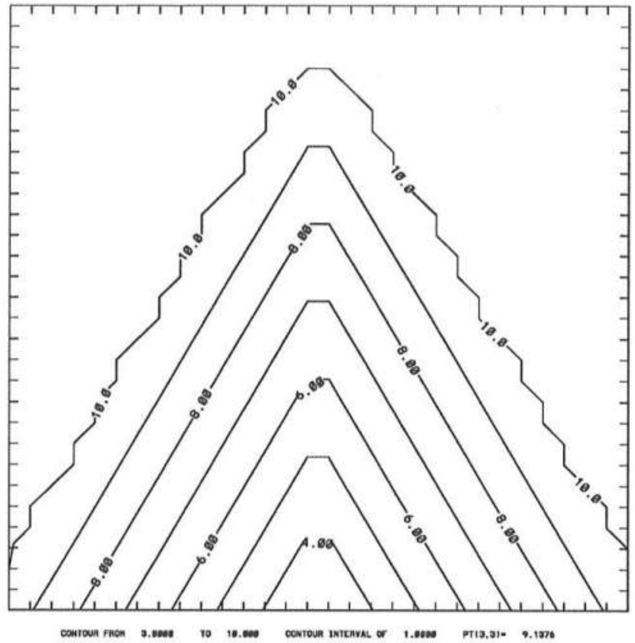

Figure 1: Bottom Bathymetry for Triangular Shoal

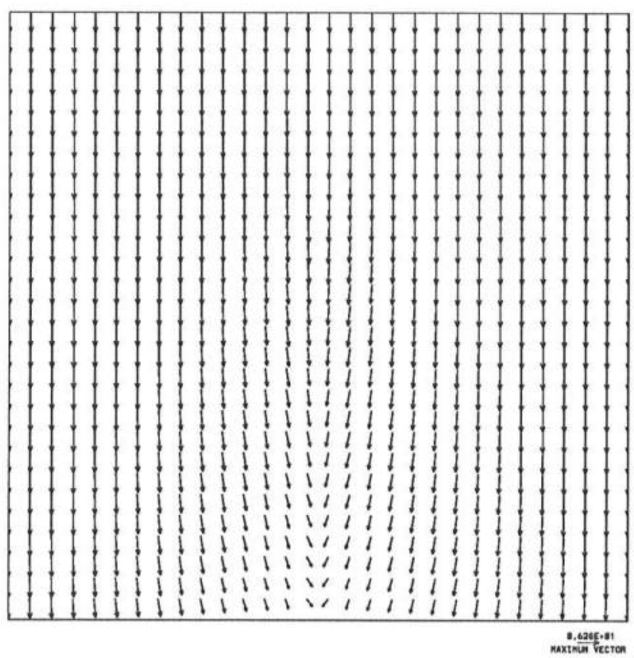

Figure 2: Wave Height and Direction Vectors for Triangular Shoal

6 Program Listing

```
C*********************************************************************
C*
C*          PROGRAM TO COMPUTE WAVE REFRACTION USING
C*             IRROTATIONALITY OF THE WAVE NUMBER
C*
C*             INCLUDES WAVE-CURRENT INTERACTION
C*
C*                   ROBERT A. DALRYMPLE
C*
C*                      January, 1991
C*
C*     BASED ON A TWO STEP LAX-WENDROFF  FINITE DIFFERENCES
C*
C*                     FULLY  EXPLICIT
C*
C*********************************************************************
      PARAMETER(IX=31,IY=31)
      COMMON /AMP/K(2,IY),B(2,IY),A(2,IY),D(IX,IY),AS(IY)
      COMMON U(IX,IY),V(IX,IY),WAX(2,IY),WAY(2,IY)
      COMMON UCGX(2,IY),VCGY(2,IY)
      COMMON /FREQ/OMEGA,SIGMA(IY),CG(IY),NONLIN
      COMMON /WAVE/HO,T,TH(IY),H(IY)
      DIMENSION AS(IY),BS(IY),WXS(IY),WYS(IY)
      REAL K
      OPEN(3,FILE='INDAT.DAT')
C*--------------------------------------------------------------------
C*                 DATA FILES                                        *
C*                                                                   *
C*                                                                   *
C*    REFDAT.DAT --  MUST CONTAIN:                                   *
C*                                                                   *
C*        ((D(I,J),J=1,N),I=1,M)      BATHYMETRIC GRID       (16F8.4) *
C*                                                                   *
C*        IF CURRENTS ARE PRESENT (ICUR = 1) THEN REFDAT.DAT ALSO HAS *
C*                                                                   *
C*        ((U(I,J),J=1,N),I=1,M)      ONSHORE CURRENT FIELD       "   *
C*        ((V(I,J),J=1,N),I=1,M)      ALONGSHORE CURRENT FIELD    "   *
C*                                                                   *
C*                                                                   *
C*    INDAT.DAT --  MUST CONTAIN:                                    *
C*                                                                   *
C*        IUN1,IUN2,IUN3,IUN4    LOGICAL DEVICE NUMBERS       (15I4) *
C*                                                                   *
```

```
C*           IUN1  :  INPUT  DISK FILE, REFDAT.DAT                          *
C*           IUN2  :  OUTPUT DISK FILE, HEIGHT.DAT -- WAVE HEIGHTS          *
C*           IUN3  :  SCREEN OUTPUT -- USE ZERO FOR PC                      *
C*           IUN4  :  OUTPUT DISK FILE, ANGLE.DAT  -- WAVE ANGLE            *
C*             (NOTE:  do not use 3 as a number!)                           *
C*                                                                          *
C*       M,N,IDEEP              GRID DIMENSIONS, OFFSHORE DATA SWITCH       *
C*                                                                          *
C*           IDEEP = 0 MEANS DEEP WATER WAVE DATA                           *
C*           IDEEP = 1 MEANS DATA FOR OFFSHORE GRID ROW                     *
C*                                                                          *
C*       IU,NONLIN,ICUR,IBC     UNITS, MODEL, CURRENTS, BOUNDARY            *
C*                                      TYPE SWITCHES                       *
C*           UNITS  = 1  METRIC         2  ENGLISH SYSTEM                   *
C*           NONLIN = 0  LINEAR MODEL   1  NONLINEAR                        *
C*           ICUR   = 0  NO CURRENTS    1  CURRENT CASE                     *
C*           IBC    = 0  TRANSMITTING   1  REFLECTING                       *
C*                                                                          *
C*       DX,DY                  GRID SIZE                         (16F8.4)  *
C*       T,TIDE                 WAVE PERIOD AND TIDAL OFFSET                *
C*       AMP,THETAO             WAVE AMPLITUDE AND DIRECTION                *
C*                                 (THETAO BETWEEN -90 TO 90 DEGREES)       *
C*                                                                          *
C*     HEIGHT.DAT  --  MUST EXIST ON DEFAULT DRIVE BEFORE RUNNING           *
C*                                                                          *
C*        ON COMPLETION, IT CONTAINS:                                       *
C*                                                                          *
C*           M,N,DX,DY                                                      *
C*           (H(J),J=1,N)  IN 16F8.4 FORMAT                                 *
C*                                                                          *
C*     ANGLE.DAT  --  MUST EXIST ON DEFAULT DRIVE BEFORE RUNNING            *
C*                                                                          *
C*        ON COMPLETION, IT CONTAINS:                                       *
C*                                                                          *
C*           M,N,DX,DY                                                      *
C*           TH(J),J=1,N   IN DEGREES AND IN 16F8.4 FORMAT                  *
C*                                                                          *
C*-------------------------------------------------------------------------
C*    DATA
      PI=3.1415927
      G=9.81
      ERROR=0.000001
      CONVD=180./PI
C*-------------------------------------------------------------------------
C*    READ INDAT.DAT DATA                                                   *
C*-------------------------------------------------------------------------
```

```
      READ(3,33) IUN1,IUN2,IUN3,IUN4
   33 FORMAT(15I4)
      READ(3,33) M,N,IDEEP
      READ(3,33) IU,NONLIN,ICUR,IBC
      IF(IU.EQ.2) G=32.17
      READ(3,34) DX,DY
   34 FORMAT(16F8.4)
      READ(3,34) T,TIDE
      OMEGA=2.*PI/T
      READ(3,34) AMP, THETAO
      IF(ABS(THETAO).GT.90.) THEN
        WRITE(IUN3,121)
  121   FORMAT('  WAVE ANGLE TOO LARGE--STOPPING')
        STOP
      ENDIF
      CLOSE (3)
      HO=2.*AMP
C*---------------------------------------------------------------------
C*    READ DEPTH AND CURRENT DATA  FROM REFDAT.DAT                       *
C*---------------------------------------------------------------------
      OPEN(IUN1,FILE='REFDAT.DAT')
      OPEN(IUN2,FILE='HEIGHT.DAT')
      OPEN(IUN4,FILE='ANGLE.DAT')
C*---------------------------------------------------------------------
C*        IF THETAO NEGATIVE, MIRROR IMAGE THE DEPTH AND CURRENT GRID  *
C*        AND CHANGE THE SIGN FOR THE MIRROR IMAGE DEPTHS              *
C*---------------------------------------------------------------------
      IF (THETAO.LT.O.) THEN
              THETAO=-THETAO
              JF=N
              JL=1
              JS=-1
      ELSE
              JF=1
              JL=N
              JS=1
      ENDIF
      DO 2 I=1,M
          READ(IUN1,10) (D(I,J),J=JF,JL,JS)
    2 CONTINUE
      IF (TIDE.NE.0.0) THEN
       DO 53 I=1,M
         DO 53 J=1,N
           D(I,J)=D(I,J)+TIDE
   53  CONTINUE
      ENDIF
```

```
      IF(ICUR.EQ.1) THEN
       DO 3 I=1,M
           READ(IUN1,10) (U(I,J),J=JF,JL,JS)
    3  CONTINUE
       DO 4 I=1,M
           READ(IUN1,10) (V(I,J),J=JF,JL,JS)
    4  CONTINUE
      ELSE
       DO 13 I=1,M
         DO 13 J=1,N
           U(I,J)=0.0
           V(I,J)=0.0
   13  CONTINUE
      ENDIF
 10   FORMAT(16F8.4)
      CLOSE(IUN1)
      R=DX/(2.*DY)
      DEN=1.+R*R
      WRITE(IUN2,333) M,N,DX,DY
      WRITE(IUN4,333) M,N,DX,DY
 333  FORMAT(2I14,2F18.7)
      WRITE(IUN3,100)
 100  FORMAT(/////'                              REFRACT'//'   A REFRACTION MODEL FOR
     1 THE CALCULATION OF WAVE HEIGHTS'/'         AND DIRECTIONS OVER A GI
     2VEN BATHYMETRY'//)
      WRITE(IUN3,99)
  99  FORMAT('   Developed by Robert A. Dalrymple'/'                   18 H
     1arvest Lane'/'                  Newark, DE 19711')
      WRITE(IUN3,101) HO,T,THETAO
 101  FORMAT(///'  INCIDENT WAVE PARAMETERS:'//'  Height =',F6.2
     1,', Period =',F6.2,', Wave Angle=',F6.2)
      IF(NONLIN.EQ.1)THEN
        WRITE(IUN3,103)
 103    FORMAT(/'  Nonlinear Model')
      ELSE
        WRITE(IUN3,105)
 105    FORMAT(/'  Linear Model'//)
      ENDIF
      IF (IBC.EQ.1) THEN
        WRITE(IUN3,104)
 104    FORMAT(/'  Reflecting Downwave Boundary Condition')
      ELSE
        WRITE(IUN3,106)
 106    FORMAT(/'  Transmitting Boundary Condition')
      ENDIF
C*-----------------------------------------------------------------
```

```
C*                                                                    *
C*      ESTABLISH DEEP WATER DATA                                     *
C*                                                                    *
C*--------------------------------------------------------------------
      CALL WVNUM(D(1,1),0.,0.,1,1,0.,0.,K(1,1),1,DB,G)
      CGO=CG(1)
      CO=OMEGA/K(1,1)
      THETAO=THETAO/CONVD
C*--------------------------------------------------------------------
C*    IF INPUT DATA ARE DEEP WATER VALUES (IDEEP.EQ.0) THEN CONVERT   *
C*               BY  SNELLS LAW                                       *
C*--------------------------------------------------------------------
      IF (IDEEP.EQ.0)THEN
C*                        CONVERT TO VALUES AT OFFSHORE GRID BOUNDARY
         CGOO=G*T/(4.*PI)

         COSO=COS(THETAO)
         SINO=SIN(THETAO)
         THETAO=ASIN(SINO*CO/(2.*CGOO))
         HO=HO*SQRT(CGOO/CG(1))*SQRT(COSO/COS(THETAO))
      ENDIF
      COSO=COS(THETAO)
      SINO=SIN(THETAO)
 14   FORMAT(16F8.4)
C*--------------------------------------------------------------------
C*                                                                    *
C*    SET UP BOUNDARY CONDITIONS ON A AND B                           *
C*                                                                    *
C*           OFFSHORE ROW                                             *
C*--------------------------------------------------------------------
      DO 15 J=1,N
        TH(J)=THETAO
        H(J)=HO
        CALL WVNUM(D(1,J),THETAO,HO,1,J,U(1,J),V(1,J),K(1,J),2,DB,G)
        A(1,J)=K(1,J)*SIN(TH(J))
        B(1,J)=K(1,J)*COS(TH(J))
 15   CONTINUE
      WRITE(IUN3,*) '     '
      WRITE(IUN3,*) ' INITIAL THETA VALUES'
      WRITE(IUN3,*) '     '
      WRITE(IUN3,14) (TH(JJ)*CONVD*JS,JJ=JF,JL,JS)
C*--------------------------------------------------------------------
C*    WRITE ANGLES TO FILE, ANGLE.DAT IN RADIANS                      *
C*--------------------------------------------------------------------
      WRITE(IUN4,14) (TH(JJ)*CONVD*JS,JJ=JF,JL,JS)
      WRITE(IUN3,*) '     '
```

```
      WRITE(IUN3,*) ' INITIAL HEIGHT VALUES'
      WRITE(IUN3,*) '    '
      WRITE(IUN3,14) (H(JJ),JJ=JF,JL,JS)
      WRITE(IUN2,14) (H(JJ),JJ=JF,JL,JS)
      AINT=A(1,1)
C*----------------------------------------------------------------------
C*                                                                    *
C*    SET UP OFFSHORE BOUNDARY CONDITION FOR WAVE ACTION              *
C*                                                                    *
C*----------------------------------------------------------------------
      DO 200 J=1,N
        EF=HO*HO
        UCGX(1,J)=U(1,J)+CG(1)*COS(THETAO)
        VCGY(1,J)=V(1,J)+CG(1)*SIN(THETAO)
        WAX(1,J)=EF*UCGX(1,J)/SIGMA(J)
 200    WAY(1,J)=EF*VCGY(1,J)/SIGMA(J)
      WAXO=WAX(1,1)
      B(2,1)=B(1,1)
      WRITE(IUN3,*) '    '
C*----------------------------------------------------------------------
C*                                                                    *
C*    START THE SOLUTION TECHNIQUE                                    *
C*                                                                    *
C*----------------------------------------------------------------------
      NM1=N-1
      ITER1=5
      ITMAX=40
      IF (NONLIN.EQ.0) ITER1=2
      DO 500 I=1,M-1
        I1=I+1
        XX=FLOAT(I)*DX
        WRITE(IUN3,999) I1, XX
 999    FORMAT(/'  ROW',I4,', X =',F6.2/)
C*----------------------------------------------------------------------
C*                                                                    *
C*    FIX UPWAVE BOUNDARY--SNELL'S LAW                                *
C*                                                                    *
C*----------------------------------------------------------------------
        A(2,1)=AINT
        WAX(2,1)=WAXO
        DO 22 IT=1,ITER1
          H(1)=SQRT(ABS(WAX(2,1)*SIGMA(1)/(CG(1)*COS(TH(1))+U(I1,1))))
          IF(D(I1,1).LT.0.) THEN
            WRITE(IUN3,*) ' DRY LAND: I=',I1
            STOP
          ELSE
```

```
            CALL WVNUM(D(I1,1),TH(1),H(1),I1,1,U(I1,1),
     -                V(I1,1),K(2,1),1,AINT,G)
          END IF
          BFIX=SQRT(K(2,1)**2.-AINT**2.)
          TH(1)=ATAN2(AINT,BFIX)
          UCGX(2,1)=U(I1,1)+CG(1)*COS(TH(1))
          VCGY(2,1)=V(I1,1)+CG(1)*SIN(TH(1))
          EB=(.8*D(I1,1))**2
          WAXB=EB*UCGX(2,1)/SIGMA(1)
          IF (WAXB.LT.ABS(WAXO)) THEN
C                                    BREAKING OCCURS
C              WRITE(IUN3,*)' BREAKING AT COLUMN 1'
            WAX(2,1)=WAXB
            WAY(2,1)=EB*VCGY(2,1)/SIGMA(1)
          ELSE
            WAX(2,1)=WAXO
            WAY(2,1)=WAXO*VCGY(2,1)/UCGX(2,1)
          END IF
          B(2,1)=BFIX
   22   CONTINUE
   23   B(2,1)=BFIX
C       COMPARISON TO SNELL'S LAW
C         THT=ASIN((SIGMA(1)/K(2,1))/(CO/SINO-V(I1,1)))
C         HH=HO*SQRT(CGO/CG(1))*SQRT(COSO/COS(THT))
C         HH=HH*SQRT(1.-SINO*V(I1,1)/CO)
C         WRITE(IUN3,*) ' SNELL''S LAW: H,THETA,D',HH,THT*CONVD,D(I1,1)
        THT=TH(1)
C*---------------------------------------------------------------------
C*                                                                   *
C*    ITERATE FOR J TH SOLUTION (ITERATIONS ARE NEEDED FOR THE LINEAR *
C*      MODEL WITH CURRENTS AND THE NONLINEAR MODEL)                  *
C*                                                                   *
C*---------------------------------------------------------------------
      ITER2=ITER1
      IF(ICUR.EQ.0.AND.NONLIN.EQ.0) ITER2=1
      DO 70 ITERA=1,ITER2
         IBFLAG=0
         DO 60 J=1,N-1
C*---------------------------------------------------------------------
C*       FOR EACH COLUMN, OBTAIN A NEW WAVE NUMBER AT I+1,J+1         *
C*---------------------------------------------------------------------
          L=J+1
         CALL WVNUM(D(I1,L),THT,H(L),I1,L,U(I1,L),V(I1,L),K(2,L),2,DB,G)
C*---------------------------------------------------------------------
C*                                                                   *
C*       FIRST LAX HALF STEP:  COMPUTE A(i+1/2,j+1/2)                 *
```

```
C*                                                                     *
C*---------------------------------------------------------------------
          AS(J)=(A(1,J+1)+A(1,J))/2. + R*(B(1,J+1)-B(1,J))
          XKB=(K(1,J)+K(1,J+1)+K(2,J)+K(2,J+1))/4.
  BS(J)=SQRT(XKB*XKB-AS(J)**2)
   60    CONTINUE
C*---------------------------------------------------------------------
C*       SET BOUNDARY CONDITION                                        *
C*---------------------------------------------------------------------
         XKB=(K(1,N)+K(2,N))/2.
         IF(IBC.EQ.0) THEN
    AS(N)=A(1,N)
          BS(N)=SQRT(XKB**2-AS(N)**2)
         ELSE
          AS(N)=0.0
          BS(N)=XKB
         ENDIF
C*---------------------------------------------------------------------
C*                                                                     *
C*       SECOND LAX  STEP                                              *
C*                                                                     *
C*---------------------------------------------------------------------
         DO 61 J=2,N
          A(2,J)=A(1,J)+2.*R*(BS(J)-BS(J-1))
  ARG=(K(2,J)**2-A(2,J)**2)
           IF(ARG.LT.0.0) THEN
              IF(JF.EQ.N) THEN
                   JJJ=N-J+1
              ELSE
                   JJJ=J
              ENDIF
              B(2,J)=B(1,J)
              WRITE(*,*) ' FIXUP IN B AT',JJJ
           ELSE
              B(2,J)=SQRT(ARG)
              THT=ASIN(A(2,J)/K(2,J))
              TH(J)=THT
           ENDIF
   61    CONTINUE
C*---------------------------------------------------------------------
C*                                                                     *
C*         SOLVE FOR WAVE ACTION, WAX(I+1,J) USING FINITE DIFFERENCES  *
C*                                                                     *
C*               FIRST LAX HALF STEP                                   *
C*                                                                     *
C*---------------------------------------------------------------------
```

```
      DO 62 J=1,N-1
        WXS(J)=(WAX(1,J+1)+WAX(1,J))/2. - R*(WAY(1,J+1)-WAY(1,J))
        L=J+1
        UCGX(2,L)=(U(I1,L)+CG(L)*COS(TH(L)))
        VCGY(2,L)=(V(I1,L)+CG(L)*SIN(TH(L)))
        VUR=(VCGY(1,J)+VCGY(2,J)+VCGY(1,L)+VCGY(2,L))
     1        /(UCGX(1,J)+UCGX(2,J)+UCGX(1,L)+UCGX(2,L))
        WYS(J)=WXS(J)*VUR
C     BE SURE TO CHANGE DEFINITION FOR VCGY FOR HALF STEP
   62 CONTINUE
      UCGX(2,N)=(U(I1,N)+CG(N)*COS(TH(N)))
      VCGY(2,N)=(V(I1,N)+CG(N)*SIN(TH(N)))
VUR=(VCGY(1,N)+VCGY(2,N))/(UCGX(1,N)+UCGX(2,N))
C*-----------------------------------------------------------------------
C*    SET BOUNDARY CONDITION                                           *
C*-----------------------------------------------------------------------
IF(IBC.EQ.0) THEN
  WXS(N)=WAX(1,N)
  WYS(N)=WXS(N)*VUR
ELSE
        WXS(N)=WAX(1,N)+R*2.*WAY(1,N)
        WYS(N)=WXS(N)*VUR
ENDIF
C*-----------------------------------------------------------------------
C*                                                                     *
C*          SECOND LAX  STEP                                           *
C*                                                                     *
C*-----------------------------------------------------------------------
      DO 63 J=2,N
       WAX(2,J)=WAX(1,J)-2.*R*(WYS(J)-WYS(J-1))
       WAYFIX=WAX(2,J)*VCGY(2,J)/UCGX(2,J)
C*-----------------------------------------------------------------------
C*       CHECK FOR BREAKING                                            *
C*-----------------------------------------------------------------------
           EB=(.8*D(I1,J))**2
           WAXB=EB*UCGX(2,J)/SIGMA(J)
           IF (WAXB.LT.ABS(WAX(2,J))) THEN
C                      BREAKING OCCURS
             WAX(2,J)=WAXB
                    IF(IBC.EQ.0)THEN
            WAYFIX=EB*VCGY(2,J)/SIGMA(J)
           ENDIF
           IBFLAG=1
           IF(JF.EQ.N) THEN
                 JJJ=N-J+1
           ELSE
```

```
                  JJJ=J
              ENDIF
c              WRITE(*,*) ' BREAKING, COLUMN',JJJ,
c     1                       D(I1,J),H(J)
            END IF
C*-----------------------------------------------------------------------
C*          END BREAKING                                                  *
C*-----------------------------------------------------------------------
         H(J)=SQRT(ABS(WAX(2,J)*SIGMA(J)/UCGX(2,J)))
         WAY(2,J)=WAYFIX
   63    CONTINUE
   70    CONTINUE
C*-----------------------------------------------------------------------
C*                                                                        *
C*       WRITE OUT RESULTS OF THE CALCULATIONS                            *
C*                                                                        *
C*-----------------------------------------------------------------------
         WRITE(IUN3,*)     '   '
         WRITE(IUN3,*) ' THETA VALUES:'
         WRITE(IUN3,*)     '   '
         WRITE(IUN3,14) (TH(JJ)*CONVD*JS,JJ=JF,JL,JS)
         WRITE(IUN4,14) (TH(JJ)*CONVD*JS,JJ=JF,JL,JS)
         WRITE(IUN3,*) '  '
         WRITE(IUN3,*)' WAVE HEIGHTS:'
         WRITE(IUN3,*)     '   '
         WRITE(IUN3,14) (H(JJ),JJ=JF,JL,JS)
         WRITE(IUN2,14) (H(JJ),JJ=JF,JL,JS)
         WRITE(IUN3,*) '  '
C*-----------------------------------------------------------------------
C*                                                                        *
C*    ROLL DOWN THE VALUES OF A,B,K                                       *
C*                                                                        *
C*-----------------------------------------------------------------------
         DO 490 J=1,N
           A(1,J)=A(2,J)
           B(1,J)=B(2,J)
           K(1,J)=K(2,J)
           UCGX(1,J)=UCGX(2,J)
           VCGY(1,J)=VCGY(2,J)
           WAX(1,J)=WAX(2,J)
           WAY(1,J)=WAY(2,J)
  490    CONTINUE
  500 CONTINUE
      CLOSE(IUN2)
      CLOSE(IUN3)
      CLOSE(IUN4)
```

```
      STOP
      END
C**********************************************************************
C*                                                                    *
      SUBROUTINE WVNUM(D,TH,H,I,J,U,V,RK,ITYPE,RKS,G)
C*                                                                    *
C*    SUBROUTINE WVNUM COMPUTES WAVE NUMBER INCLUDING WAVE-            *
C*    CURRENT INTERACTION, USING SECANT METHOD.                       *
C*                                                                    *
C*    OMEGA = SIGMA + U dot K                                         *
C*                                                                    *
C*    NOTE:  ITYPE DENOTES TYPE OF SOLUTION                           *
C*           ITYPE=2, NORMAL SOLUTION                                 *
C*           ITYPE=1, SNELLS LAW                                      *
C**********************************************************************
      PARAMETER(IY=31)
      COMMON /FREQ/OMEGA,SIGMA(IY),CG(IY),NONLIN
      EPSK=0.000001
      AMP=H/2.
      IF(D.LT..01) THEN
        D=.01
        WRITE(IUN3,*) ' DRY LAND AT', J
      ENDIF
      RK=OMEGA*OMEGA/(G*SQRT(TANH(OMEGA*OMEGA*D/G)))
      IF(RK*D.GT.7.)  THEN
          CG(J)=OMEGA/(2.*RK)
          GO TO 120
      ENDIF
      COSI=COS(TH)
      SINI=SIN(TH)
      DO 100 ITER=1,50
        SIGMA(J)=OMEGA-RK*U*COSI-RK*V*SINI
        IF (ITYPE.EQ.1) SIGMA(J)=OMEGA-RK*U*COSI-V*RKS
        SIG=SIGMA(J)
        A2=SIG**2
        ARG=RK*D
        F1=EXP(ARG)
        F2=1.0/F1
        SH=(F1-F2)/2.0
        SE=2.0/(F1+F2)
        SECH2=SE*SE
        TT=SH*SE
        EP=RK*AMP
C*---------------------------------------------------------------------
C*      ADD NONLINEAR TERMS                                           *
C*---------------------------------------------------------------------
```

```
      IF (NONLIN.EQ.1) THEN
        SH4=SH**4.
        DD=(COSH(4.*ARG)+8.-2.*TT*TT)/(8.*SH4)
        DDK=D*((SINH(4.*ARG)-TT*SECH2)/SH4-DD/TT)/2.
        F1=TT**5
        F1K=5.*F1/(SH/SE)
        F2=(ARG/SH)**4
        F2K=4.*F2*(1.-ARG/TT)/RK
        TT=TANH(ARG+F2*EP)
        FACT=(1.+F1*EP*EP*DD)
c        UR=RK*H/(2.*ARG**3)
      ELSE
        DD=0.
        FACT=1.
        DDK=0.
        F1=0.
        F2=0.
        F1K=0.
        F2K=0.
      ENDIF
C*--------------------------------------------------------------------
C*        USE NEWTON-RAPHSON ITERATION                               *
C*--------------------------------------------------------------------
        FK=G*RK*TT*FACT-A2
C        IF(ITER.EQ.1) THEN
C                            NEWTON-RAPHSON
         FFK=G*(SECH2*(ARG+EP*(F2K*K+F2))+TT)*FACT+
     1          G*RK*TT*EP*(F1*(2.*AMP*DD+EP*DDP)+F1K*EP*DDK)
         IF (ITYPE.EQ.1)  THEN
              FFK=FFK+2.0*(U*COSI)*SIG
         ELSE
              FFK=FFK+2.0*(U*COSI+V*SINI)*SIG
         ENDIF
         RKNEW=RK-FK/FFK
C     ELSE
C             SECANT METHOD
C          RKNEW=RK-FK*(RK-RKO)/(FK-FKO)
C      ENDIF
      IF(ABS(RKNEW-RK).LE.(ABS(EPSK*RKNEW)))THEN
        RK=RKNEW
        ARG=RK*D
        TT=TANH(ARG)
        SECH2=1./(COSH(ARG)**2)
        SIGMA(J)=OMEGA-U*RK*COSI-RK*V*SINI
        CG(J)=(1.+RK*D*SECH2/TT)*SIGMA(J)/(RK*2.)
        GO TO 110
```

```
        ENDIF
   99 RKO=RK
      FKO=FK
      RK=RKNEW
  100 CONTINUE
      WRITE(IUN3,101)RK,D,H
  101 FORMAT(' INTERATION FOR K FAILED TO CONVERGE:
     -K,D,H,METH',3F10.5,I3)
C       IF (NONLIN.EQ.1) THEN
C         IF (UR.GT.1.)WRITE(IUN3,*) ' URSELL NO.', UR,I,J
C       ELSE
C       ENDIF
      RETURN
  110 CONTINUE
C       IF (NONLIN.EQ.1) THEN
C         IF (UR.GT.1.)WRITE(IUN3,*) ' URSELL NO.',UR,I,J
C       ELSE
C       ENDIF
      IF(RK.GT.0.0)GO TO 120
      WRITE(IUN3,130)D,COSI,SINI,U,V,I,J
  130 FORMAT(' RK IS NEGATIVE: D,COSI,SINI,U,V,I,J',
     *5F10.5,2I5)
  120 RETURN
      END
```