NASA-TM-74076

NASA Technical Memorandum 74076

Minimal-Resource Computer Program for Automatic Generation of Ocean Wave Ray or Crest Diagrams in Shoaling Waters

Lamont R. Poole, Stuart R. LeCroy,
and W. Douglas Morris

DECEMBER 1977

NASA

NASA Technical Memorandum 74076

Minimal-Resource Computer Program for Automatic Generation of Ocean Wave Ray or Crest Diagrams in Shoaling Waters

Lamont R. Poole
Langley Research Center, Hampton, Virginia
Stuart R. LeCroy
Vought Corporation, Hampton, Virginia
W. Douglas Morris
Langley Research Center, Hampton, Virginia

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1977

Page intentionally left blank

Page intentionally left blank

CONTENTS

SUMMARY

An improved computer program for studying the linear ocean wave refraction process has been developed. The program features random-access modular bathymetry data storage to minimize computer resource requirements. The user can select from three bottom topography approximation techniques which provide varying degrees of bathymetry data smoothing. This allows the user to assess the sensitivity of computed results to the degree of smoothing chosen. Wave-ray patterns can be generated and plotted automatically with either a specified uniform initial (deepwater) ray spacing or a constrained final (nearshore) ray spacing. As an option, wave-crest patterns can be constructed and plotted by using a cubic polynomial to approximate individual crest segments between adjacent rays.

INTRODUCTION

The prime objective of the NASA Oceanology Program is development and demonstration of the feasibility of satellite systems such as SEASAT-A for the acquisition of global data on ocean conditions. The broad-scale data provided by such systems can be used to initialize analytical models which would then provide short-term, higher resolution forecasts of ocean conditions for ship routing and offshore activities. One such analytical tool is the wave refraction model, which can predict changes in the speed and direction of waves as they travel from deep water across the continental shelf toward the coastline. Such changing patterns affect the distribution of wave energy (wave height) over the area through which the waves are traveling and along the shoreline.

Early models, such as that developed in reference 1, used graphical techniques for constructing refraction diagrams. A wave refraction diagram is a map showing either a group of wave crests at a given time or the successive positions of a particular wave crest as it moves shoreward. The first models required the construction of each wave crest as it advanced toward shore, but later models required only the construction of wave rays which are locally perpendicular to each crest. The wave-ray technique eliminated the time-consuming process of constructing each crest, at the expense of reducing the visual familiarity of the end product. This graphical ray technique was later incorporated into a computer program by Griswold in 1963 (ref. 2), but the program's usefulness was limited because a spatial grid of wave speeds was required in calculating wave-ray paths. The refraction model developed by Wilson (ref. 3) used a grid of water depths which eliminated many of the problems associated with the wave-speed grid; however, the model could not provide wave height. The model developed by Dobson (ref. 4) succeeded in coupling wave-ray construction using a grid of water depths with a finite-difference solution to the wave intensity equation developed by Munk and Arthur (ref. 5) to obtain wave heights. The Dobson model formed the basis for the model used at the Langley Research Center (LaRC) in a recent broad-scale analytical study of wave refraction in the Virginian Sea region of the mid-Atlantic continental shelf (ref. 6).

Several modifications have been incorporated into the LaRC refraction model since the Virginian Sea study. Bathymetry data are stored and used in modular form by using random-access techniques to pass data modules as required from a peripheral mass-storage file to the central memory of the computer (ref. 7). This technique allows very large geographical regions to be studied with a minimum of computer resources but does not restrict the applicability of the model to smaller areas. Ocean bottom topography can be approximated by using either the quadratic least squares, the cubic least squares, or the constrained bicubic interpolation approaches (ref. 8). With these three approximation options, the user can select different degrees of input data smoothing and assess the sensitivity of computed results to the degree of smoothing selected. In the normal mode, the spatial position of the initial point on each wave ray in a particular set is computed automatically to achieve a uniform deepwater ray density. In the previous version of the model, the spatial coordinates of each initial point were input on individual computer cards. By selecting the uniform deepwater ray density, the user can readily identify areas of relatively high or low energy on the resultant refraction diagram as areas of convergent or divergent ray groups. As an alternative, the user can select a control option which provides for automatic computation of deepwater ray spacing to satisfy specified constraints on nearshore ray density. Selection of this option could provide increased directional information in the nearshore zone while sacrificing the easy identification of high or low energy regions. As a third option, wave crests can be constructed by using a cubic polynomial to approximate the crest segment between corresponding points on adjacent rays. Selection of this crest output option also results in selection of the aforementioned controlled nearshore ray density option.

This paper presents a description of the modified LaRC wave refraction model as developed for use on the Control Data Cyber 173 or 175 computer under Network Operating System (NOS) 1.2. A central memory field length of 66000_8 is required, along with the peripheral mass-storage file used for input of bathymetry data. A description, flow diagram, and source code listing are provided for the main program and each subroutine. Program input and output parameters are listed and described. Sample graphic outputs are provided. A list of system subroutines used, a detailed description of the wave-crest approximation technique and the accompanying subroutine for controlling nearshore ray density, and a description of a utility program which can be used for modularizing a regional bathymetry array are included in the appendixes.

SYMBOLS

a_1,a_2,a_3 coefficients of cubic polynomial used to approximate wave-crest segment

b ray separation distance, meters

c wave speed, meters/second

d water depth, meters

g acceleration due to gravity, meters/second^2

H	wave height, meters
K_r	refraction coefficient
K_s	shoaling coefficient
L	wavelength, meters
p	coefficient in equation governing ray separation, defined by equation (10), seconds^{-1}
q	coefficient in equation governing ray separation, defined by equation (11), seconds^{-2}
s	spatial coordinate along wave ray
t	time, seconds
x	alongshore coordinate, nautical miles
x_o, x_m	computational boundaries along x-axis
y	offshore coordinate, nautical miles
y_o, y_m	computational boundaries along y-axis
α	ray angle with respect to x-axis, degrees
β	ray separation factor
κ	ray curvature, radians/meter
σ	radian wave frequency, radians/second
θ	angle between adjoining first-order wave-crest segments, degrees

Subscripts:

i	ray number index
j	index of computational point along ray
o	initial

A prime denotes differentiation with respect to water depth d.

PROBLEM DESCRIPTION

The problem considered in the present paper is that of computing the refraction characteristics of ocean waves propagating from deep water across the continental shelf toward the coastline. The results of linear wave theory, which assumes a progressive sinusoidal wave of constant frequency and small steepness, have been adopted; and steady-state conditions of constant tide height and no winds are assumed. The governing equations have been given in references 4 and 5 and are discussed briefly in this section. Also discussed is the method of solution, which is based primarily on the method presented in reference 4. Two additional features enhance the solution in terms of bathymetry data storage requirements and alternate numerical techniques for approximating ocean bottom topography.

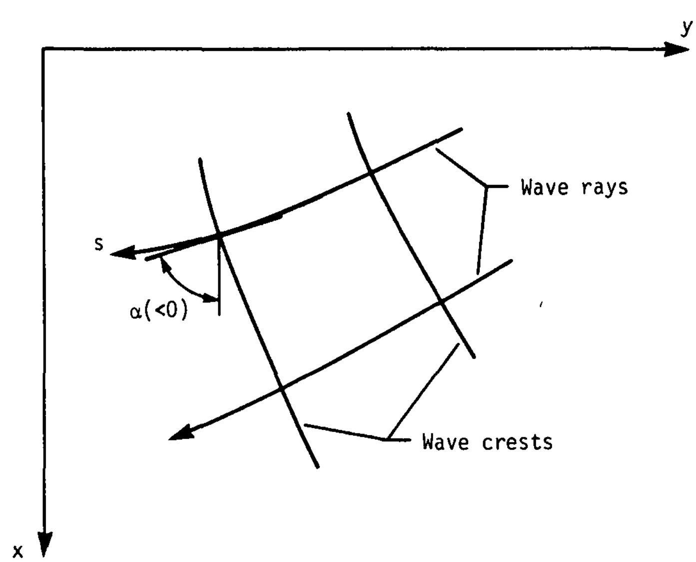
The path of any wave ray (orthogonal to a wave crest) can be represented by the equations

$$\frac{dx}{dt} = c \cos \alpha \tag{1}$$

$$\frac{dy}{dt} = c \sin \alpha \tag{2}$$

$$\kappa = \frac{d\alpha}{ds} = \frac{1}{c} c' \left(\frac{\partial d}{\partial x} \sin \alpha - \frac{\partial d}{\partial y} \cos \alpha \right) \tag{3}$$

where d is the local water depth. Ray orientation relative to regional coordinates is depicted in the following sketch:



The local wave phase speed c can be expressed according to linear wave theory by

$$c - c_o \tanh \frac{\sigma d}{c} = 0 \tag{4}$$

where c_o is the initial (deepwater) phase speed and σ is the fixed radian wave frequency. On the basis of properties of the hyperbolic tangent function, the quantity $\tanh \sigma d/c$ in equation (4) can be replaced by 1 if the value of $\sigma d/c$ is sufficiently large (deepwater range), or by the quantity $\sigma d/c$ if the value of $\sigma d/c$ is sufficiently small (shallow-water range). For the purposes of the present paper, deep water is assumed if the local water depth exceeds 0.25 times the initial wavelength (ref. 6). Shallow water is assumed if the local depth is less than 0.005 times the wavelength. All other depths lie within the intermediate range in which the full transcendental form of equation (4) must be used. By differentiating equation (4) with respect to d, it can be shown that

$$c' = \frac{\sigma c\left[1 - \left(\frac{c}{c_o}\right)^2\right]}{\frac{c^2}{c_o} + \sigma d\left[1 - \left(\frac{c}{c_o}\right)^2\right]} \tag{5}$$

By assuming that energy is neither transmitted across wave rays nor dissipated by bottom friction or percolation, the wave height at any point along a wave ray can be computed by

$$H = H_o K_s K_r \tag{6}$$

where H_o is the initial wave height. The local shoaling coefficient K_s can be expressed as (ref. 4)

$$K_s = \left[\frac{c_o}{c\left(1 + \frac{2\sigma d/c}{\sinh 2\sigma d/c}\right)}\right]^{1/2} \tag{7}$$

The local refraction coefficient K_r is given by

$$K_r = \left(\frac{b_o}{b}\right)^{1/2} = (\beta)^{-1/2} \tag{8}$$

The ray separation factor β is governed by the differential equation

$$\frac{d^2\beta}{dt^2} + p\,\frac{d\beta}{dt} + q\beta = 0 \tag{9}$$

in which

$$p = -2c'\left(\frac{\partial d}{\partial x} \cos \alpha + \frac{\partial d}{\partial y} \sin \alpha\right) \tag{10}$$

and it can be shown that

$$q = cc'\left\{\sin^2 \alpha\left[\frac{\partial^2 d}{\partial x^2} + U\left(\frac{\partial d}{\partial x}\right)^2\right] - 2 \sin \alpha \cos \alpha\left(\frac{\partial^2 d}{\partial x\, \partial y} + U \frac{\partial d}{\partial x} \frac{\partial d}{\partial y}\right)\right.$$

$$\left. + \cos^2 \alpha\left[\frac{\partial^2 d}{\partial y^2} + U\left(\frac{\partial d}{\partial y}\right)^2\right]\right\} \tag{11}$$

where

$$U = \frac{-2\sigma c^2/c_o}{\left\{\frac{c^2}{c_o} + \sigma d\left[1 - \left(\frac{c}{c_o}\right)^2\right]\right\}^2} \tag{12}$$

The variables of primary influence in the solution of the governing equations are the local water depth and its spatial variation. For a large geographical area such as the continental shelf, a practical method for the input of bathymetry data to the refraction model is through a spatial grid format in which known depth values are supplied at the nodes of the grid. Such a format is used in the present model, but instead of storing the bathymetry data array for the entire region of interest in central memory, as with the model described in reference 4, overlapping subregional modules of bathymetry data are read as required into memory from a peripheral mass-storage file by using nonsequential random-access techniques. This method of data storage is described in detail in reference 7 and was shown to decrease computer storage requirements by 75 percent for a refraction model of the mid-Atlantic continental shelf.

In conjunction with the grid-type format for bathymetry data, a technique is required for determining depth values at intermediate points between nodes and for approximating spatial derivatives of the depth as required in equations (3), (10), and (11). The present model offers the selection of three such bottom topography approximation techniques, which are discussed in reference 8. These include the quadratic least squares technique, which was used in the model described in reference 4, the cubic least squares technique, and the constrained bicubic interpolation technique. Both least squares techniques offer some smoothing of the input bathymetry data but do not guarantee continuity in the computed depth or its derivatives. The constrained bicubic interpolation technique guarantees continuity in the depth values and first-order and mixed second-order partial derivatives; it has the possible disadvantage of using actual input data without smoothing.

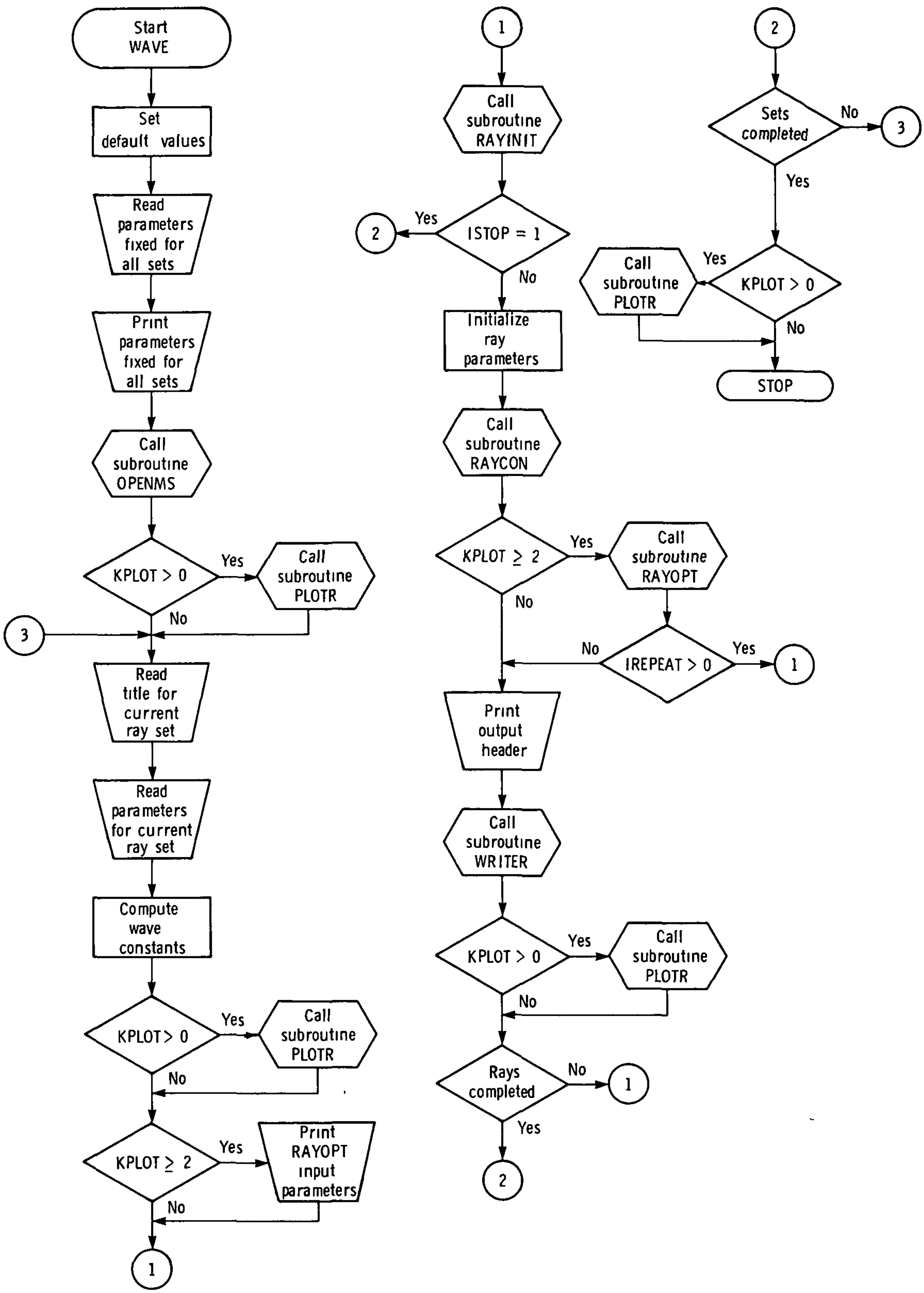
Apart from the techniques for bathymetry data storage and bottom topography approximation, the basic solution technique used in the present model is described in detail in reference 4. The path of each wave ray is constructed in equal time steps by an iterative solution of equations (1) to (3). At each step the wave height is calculated by applying Fox's finite-difference solution (see ref. 4 for details) to equation (9) to determine the next ray separation factor and then using this result in equation (6).

PROGRAM DESCRIPTION

The LaRC wave refraction computer program is coded in FORTRAN Extended (FTN) Version 4 for use on the Control Data Cyber 173 or 175 computer. It consists of a main program WAVE and subroutines RAYINIT, RAYCON, REFRAC, CURVE, DEPTH, HEIGHT, WRITER, PLOTR, RAYOPT, and MATRIX, whose names are descriptive of the functions they perform. The functions performed by routines WAVE, RAYCON, REFRAC, CURVE, and HEIGHT are similar to those of their counterpart routines in Dobson's model (ref. 4). Subroutine DEPTH is an expanded version of its counterpart in Dobson's model, with additional logic for alternate topography approximation techniques and for determination of the required bathymetry data module. Subroutine WRITER stores computed parameters in arrays and prints these arrays after a complete ray path has been constructed. Subroutine RAYINIT computes the spatial coordinates of the initial point on each ray in a particular set, based on the initial ray direction and a specified distance between adjacent rays and wave crests. Subroutine RAYOPT provides the user, if desired, with control over the final nearshore spacing between adjacent rays. This control option is also exercised if the user desires a plot of wave crests, which are constructed by segments in subroutine PLOTR. This subroutine also contains the logic for plotting both wave crests and wave rays. Finally subroutine MATRIX performs matrix multiplications which are required in determining coefficients in the bottom topography approximating polynomials. In addition, the program uses a number of system subroutines to satisfy various data handling and plotting requirements; these subroutines are listed in appendix A.

Main Program WAVE

The main program WAVE is used for input and output of parameters pertinent to the ray set being computed, including setting default values and checking for input errors. The random-access disk file named TAPE1, which is used for input of bathymetry data modules, is initialized by using the system subroutine OPENMS. Initialization of parameters for each ray in the set is also performed in WAVE. If one of the plotting options is chosen, subroutine PLOTR is called to perform plot maintenance such as to open and close the plot file and to control frame advances and positioning of frame origins. The flow diagram and listing of program WAVE follow.


Start WAVE
Set default values
Read parameters fixed for all sets
Print parameters fixed for all sets
Call subroutine OPENMS
KPLOT > 0
Yes
No
Call subroutine PLOTR
3
Read title for current ray set
Read parameters for current ray set
Compute wave constants
KPLOT > 0
Yes
No
Call subroutine PLOTR
KPLOT ≥ 2
Yes
No
Print RAYOPT input parameters
1
1
Call subroutine RAYINIT
ISTOP = 1
Yes
2
No
Initialize ray parameters
Call subroutine RAYCON
KPLOT ≥ 2
Yes
No
Call subroutine RAYOPT
IREPEAT > 0
Yes
1
No
Print output header
Call subroutine WRITER
KPLOT > 0
Yes
No
Call subroutine PLOTR
Rays completed
No
1
Yes
2
2
Sets completed
No
3
Yes
KPLOT > 0
Yes
Call subroutine PLOTR
No
STOP

```
      PROGRAM WAVE (INPUT,OUTPUT,TAPE5=INPUT,TAPE1)
*
************************************************************************
*                                                                      *
*       A PROGRAM TO CONSTRUCT REFRACTION DIAGRAMS AND COMPUTE WAVE    *
*       HEIGHTS FOR WAVES MOVING INTO SHOALING WATER.                  *
*                                                                      *
***** THIS PROGRAM REQUIRES A GRID OF BATHYMETRY DATA TO BE INPUT TO   *
*      MEMORY. CREATE A RANDOM ACCESS DISK FILE (TAPE1) TO STORE THE   *
*      DEPTH VALUES IN MODULE FORM. EACH MODULE IS TO CONTAIN 42       *
*      SUB-ROWS AND 32 SUB-COLUMNS. SUB-COLUMNS 30,31, AND 32 OF A     *
*      MODULE OVERLAY SUB-COLUMNS 1,2,AND 3 OF THE ADJACENT MODULE IN  *
*      THE NEXT COLUMN. SUB-ROWS 40,41,AND 42 OF A MODULE OVERLAY      *
*      SUB-ROWS 1,2, AND 3 OF THE ADJACENT MODULE IN THE NEXT ROW.     *
*      DEPTH AT POINTS IN THE ROWS AND COLUMNS WHICH LIE OUTSIDE THE   *
*      REGIONAL MI BY MJ BATHYMETRY ARRAY ARE SET EQUAL TO THE DEPTHS  *
*      ALONG THE BOUNDARIES OF THE ARRAY. THE MODULES ARE INDEXED BY   *
*      ROWS BEGINNING IN THE COLUMN ADJACENT TO THE X-AXIS.            *
*                                                                      *
***** NAMELIST INPUT PARAMETERS *****                                  *
*                                                                      *
* **   BASIC DATA = NPUT1  **                                          *
*                                                                      *
*      DCON        = MULTIPLIER TO CONVERT DEPTH UNITS TO FEET FOR     *
*                    INTERNAL COMPUTATIONS                             *
*                      (DEFAULT = 1.)                                  *
*      DELTAS      = MINIMUM STEP LENGTH ALONG RAY IN SHALLOW WATER IN *
*                    GRID UNITS                                        *
*                      (DEFAULT = 0.01)                                *
*      GRID        = NUMBER OF FEET PER GRID DIVISION                  *
*                      (INPUT REQUIRED)                                *
*      GRINC       = STEP LENGTH ALONG RAY IN DEEP WATER IN GRID UNITS *
*                      (DEFAULT = 1.)                                  *
*      IOUTUNT     = OUTPUT UNITS - NAUTICAL MILES FOR SPATIAL         *
*                    COORDINATES X,Y; FOR ALL OTHER PARAMETERS,        *
*                      1 = U.S. CUSTOMARY UNITS                        *
*                      2 = SI UNITS  (DEFAULT)                         *
*      JPRTFRQ(1)  = PRINT FREQUENCY FOR DEPTH/WL > 0.5                *
*                      (DEFAULT = 20)                                  *
*      JPRTFRQ(2)  = PRINT FREQUENCY FOR 0.25 < DEPTH/WL < 0.5         *
*                      (DEFAULT = 10)                                  *
*      JPRTFRQ(3)  = PRINT FREQUENCY FOR DEPTH/WL < 0.25               *
*                      (DEFAULT = 5)                                   *
*      KPLOT       = PLOT OPTION                                       *
*                      0 = NO PLOT  (DEFAULT)                          *
*                      1 = PLOT WAVE RAYS WITH UNIFORM INITIAL SPACING *
*                      2 = PLOT WAVE RAYS WITH CONTROLLED FINAL SPACING*
*                      3 = PLOT WAVE CRESTS                            *
```

```
*     LIMNPT       = MAXIMUM NUMBER OF COMPUTATIONAL POINTS ALONG RAY     *
*                       (DEFAULT = 400)                                   *
*     MI           = TOTAL NUMBER OF ROWS IN REGIONAL BATHYMETRY ARRAY    *
*                       (INPUT REQUIRED)                                  *
*     MJ           = TOTAL NUMBER OF COLUMNS IN REGIONAL BATHYMETRY       *
*                    ARRAY                                                *
*                       (INPUT REQUIRED)                                  *
*     NOMOD        = NUMBER OF 42 BY 32 MODULES NEEDED TO STORE MI BY     *
*                    MJ ARRAY. MAXIMUM = 100                              *
*                       (INPUT REQUIRED)                                  *
*     NOROW        = NUMBER OF MODULES ALONG X-AXIS IN NOMOD              *
*                       (INPUT REQUIRED)                                  *
*     NOSETS       = NUMBER OF SETS OF RAYS TO BE COMPUTED                *
*                       (DEFAULT = 1)                                     *
*     PLTLNGX      = LENGTH OF X-AXIS FOR GRAPHICS DISPLAY  (INCHES)      *
*                       (DEFAULT = 10.0)                                  *
*     PLTLNGY      = LENGTH OF Y-AXIS FOR GRAPHICS DISPLAY  (INCHES)      *
*                       (DEFAULT = 10.0)                                  *
*     PLTSCAL      = X-AND Y-AXES SCALE FACTOR. CHANGE IN PHYSICAL X OR   *
*                    Y VALUE PER INCH OF PLOT AXES  (NAUTICAL MILES PER   *
*                    INCH)                                                *
*                       (INPUT REQUIRED IF PLOTTING)                      *
*     TIDE         = HEIGHT OF TIDE  (FEET)                               *
*                       (DEFAULT = 0.)                                    *
*                                                                         *
* **  SET DATA = NPUT2  **                                                *
*                                                                         *
*   * THE FOLLOWING SET OF DATA IS REQUIRED FOR ALL CASES.                *
*                                                                         *
*     A            = INITIAL DIRECTION OF RAY  (DEGREES)                  *
*                       (INPUT REQUIRED)                                  *
*     HO           = DEEP WATER WAVE HEIGHT  (FEET)                       *
*                       (INPUT REQUIRED)                                  *
*     NCREST       = SPACING OF CREST TIC-MARKS OR CREST CURVES IN        *
*                    INTEGER MULTIPLE OF THE PROGRAM COMPUTATIONAL TIME   *
*                    STEP                                                 *
*                       (DEFAULT = 5)                                     *
*     NSURFCE      = BOTTOM TOPOGRAPHICAL APPROXIMATION TECHNIQUE         *
*                       1 = QUADRATIC LEAST SQUARES  (DEFAULT)            *
*                       2 = CUBIC LEAST SQUARES                           *
*                       3 = CONSTRAINED BICUBIC INTERPOLATION             *
*     RAYSPC       = INITIAL SPACING BETWEEN RAYS IN GRID UNITS           *
*                       (DEFAULT = 5.)                                    *
*     T            = WAVE PERIOD  (SECONDS)                               *
*                       (INPUT REQUIRED)                                  *
*                                                                         *
*   * THE FOLLOWING PARAMETERS ARE REQUIRED ONLY IF KPLOT IS GREATER      *
*     THAN OR EQUAL TO 2.                                                 *
*                                                                         *
```

```
*     DISTMAX     = MAXIMUM PERMISSIBLE FINAL SEPARATION DISTANCE       *
*                   BETWEEN ADJACENT RAYS IN GRID UNITS                 *
*                     (DEFAULT = 999.)                                  *
*     DISTMIN     = MINIMUM PERMISSIBLE FINAL SEPARATION DISTANCE       *
*                   BETWEEN ADJACENT RAYS IN GRID UNITS                 *
*                     (DEFAULT = 25.)                                   *
*     KCREST      = POINT BETWEEN ADJACENT RAYS AT WHICH TO BEGIN       *
*                   PLOTTING THE CURVED CREST SEGMENT WHEN ANGLE        *
*                   BETWEEN ADJOINING FIRST-ORDER CREST SEGMENTS        *
*                   IS LESS THAN RAYANG. MAXIMUM = 7                    *
*                     (DEFAULT = 1)                                     *
*     RAYANG      = MINIMUM ACCEPTABLE ANGLE BETWEEN ADJOINING          *
*                   FIRST-ORDER CREST SEGMENTS BELOW WHICH CREST        *
*                   PLOTTING WILL BE MODIFIED  (DEGREES)                *
*                     (DEFAULT = 120.)                                  *
*     RAYMAX      = MAXIMUM PERMISSIBLE INITIAL RAY SPACING IN GRID     *
*                   UNITS                                               *
*                     (DEFAULT = 25.)                                   *
*     RAYMIN      = MINIMUM PERMISSIBLE INITIAL RAY SPACING IN GRID     *
*                   UNITS                                               *
*                     (DEFAULT = 0.01)                                  *
*                                                                       *
******** END OF NAMELIST INPUT ********                                 *
*                                                                       *
* **  INPUT TITLE INFORMATION **                                        *
*                                                                       *
*     TITL        = IDENTIFYING TITLE FOR EACH SET                      *
*                     (INPUT CARD FORMAT = 8A10)                        *
*                                                                       *
***** DATA INPUT ORDER *****                                            *
*                                                                       *
*     NPUT1/SET 1=TITL,NPUT2/SET 2=TITL,NPUT2/ETC                       *
*                                                                       *
*************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      NAMELIST/NPUT1/DCON,DELTAS,GRID,GRINC,IOUTUNT,JPRTFRQ,KPLOT,
     +LIMNPT,MI,MJ,NOMOD,NOROW,NOSETS,PLTLNGX,PLTLNGY,PLTSCAL,TIDE
     +/NPUT2/A,DISTMAX,DISTMIN,HO,KCREST,NCREST,NSURFCE,RAYANG,RAYMAX,
     +RAYMIN,RAYSPC,T
*
****  SET INPUT DEFAULT VALUES ****
```

```
*
      DCON = 1.0
      DELTAS = 0.01
      GRID = 0.0
      GRINC = 1.0
      IOUTUNT = 2
      IREPEAT = 0
      JPRTFRQ(1) = 20
      JPRTFRQ(2) = 10
      JPRTFRQ(3) = 5
      KPLOT = 0
      LIMNPT = 400
      MI = 0
      MJ = 0
      NOMOD = 0
      NOROW = 0
      NOSETS = 1
      PLTLNGX = 10.
      PLTLNGY = 10.
      PLTSCAL = 999.
      TIDE = 0.
      A = 999.
      DISTMAX = 999.
      DISTMIN = 25.
      HO = 999.
      KCREST = 1
      NCREST = 5
      NSURFCE = 1
      RAYANG = 120.
      RAYMAX = 25.
      RAYMIN = 0.01
      RAYSPC = 5.
      T = 999.
*
** READ BASIC DATA **
*
      READ (5,NPUT1)
*
** CHECK FOR INPUT ERROR **
*
      IF (EOF(5)) 10,20
   10 PRINT 190
      GO TO 180
   20 IF (GRID .GT. 0.0) GO TO 30
      PRINT 200
      GO TO 180
   30 IF (MI .GT. 0 .AND. MJ .GT. 0 .AND. NOMOD .GT. 0 .AND. NOROW .GT.
     +0) GO TO 40
      PRINT 210
```

```
   40  IF (KPLOT .EQ. 0 .OR. PLTSCAL .LT. 999.0) GO TO 50
       PRINT 220
       GO TO 180
*
** PRINT BASIC INPUT PARAMETERS **
*
   50  PRINT 230, DCON,DELTAS,GRID,GRINC,IOUTUNT,JPRTFRQ,KPLOT,
      +LIMNPT,MI,MJ,NOMOD,NOROW,NOSETS,PLTLNGX,PLTLNGY,PLTSCAL,TIDE
*
** OPEN MASS STORAGE DISK FILE (TAPE1) CONTAINING MODULAR BATHYMETRY DATA **
*
       NOMOD = NOMOD+1
       CALL OPENMS (1,IBLK,NOMOD,0)
*
** OPEN PLOT FILE IF PLOT OPTION USED **
*
       IF (KPLOT .GT. 0) CALL PLOTR (1)
*
** GENERATE CURRENT RAY SET **
*
       DO 170 NOSET=1,NOSETS
*
** READ TITLE INFORMATION **
*
       READ 250, TITL
*
** CHECK FOR TITLE CARD **
*
       IF (EOF(5)) 60,70
   60  PRINT 260
       GO TO 180
   70  TITC1 = 2H $
       ENCODE (10,270,TITC2) TITL(1)
       IF (TITC1 .NE. TITC2) GO TO 80
       PRINT 260
       GO TO 180
   80  CONTINUE
*
** READ INPUT PARAMETERS FOR CURRENT SET **
*
       READ (5,NPUT2)
*
** CHECK FOR INPUT ERROR **
*
       IF (EOF(5)) 90,100
   90  PRINT 280
       GO TO 180
  100  IF (A .LT.9999.0 .AND. HO .LT. 999.0 .AND. T .LT. 999.0) GO TO 110
       PRINT 290
```

```
      GO TO 180
*
** INITIALIZE PARAMETERS FIXED FOR CURRENT SET **
*
  110 SIG = 6.28318531/T
      CO = 5.1204062*T
      WLO = CO*T
      DRC = 0.25*WLO
      DTGR = GRINC/CO
      AST = A
      DXY = 999.99
      ISTOP = 0
      NORAY = 1
      UNIT = DTGR*GRID
*
** PLOT TITLE AND AXES INFORMATION IF PLOT OPTION USED **
*
      IF (KPLOT .GT. 0) CALL PLOTR (2)
*
** PRINT SPECIAL INPUT PARAMETERS IF PLOTTING WAVE CREST OR RAYS WITH **
** CONTROLLED FINAL SPACING                                           **
*
      IF (KPLOT .GE. 2) PRINT 240, DISTMAX,DISTMIN,KCREST,RAYANG,
     +RAYMAX,RAYMIN
*
** FIND RAY STARTING POINT **
*
  120 CALL RAYINIT (X,Y,AST)
      IF (ISTOP .EQ. 1) GO TO 160
*
** INITIALIZE RAY PARAMETERS **
*
      A = AST
      CXY = CO
      NPT = 1
      NTICOFF = NCREST*(NTIC-1)
      NWRITE = 1
      WL = WLO
      B1=B2=RK=SK=1.0
      PDX=PDY=0.
      CALL DEPTH (X,Y)
*
** CALCULATE RAY PATH **
*
      CALL RAYCON (X,Y,A)
*
** SAVE INDEX OF FIRST AND LAST RAY POINT WITH RESPECT TO **
** REFERENCE CORNER                                       **
*
```

```
      NFRST = NTICOFF+1
      NLAST = NTICOFF+NPT
*
** COMPUTE INITIAL RAY SPACING TO MEET FINAL CONSTRAINTS IF KPLOT **
** EQUALS 2 OR 3                                                   **
*
      IF (KPLOT .GE. 2) CALL RAYOPT (IREPEAT)
      IF (IREPEAT .GT. 0) GO TO 120
*
** PRINT CONSTANT RAY PARAMETERS **
*
      PRINT 300, TITL,T,UNIT,RAYSPC,NCREST,NSURFCE,NORAY,NOSET
*
** PRINT OUTPUT HEADER INFORMATION **
*
      PRINT 310
*
** PRINT OUTPUT UNITS **
*
      IF (IOUTUNT-2) 130,140
  130 PRINT 320
      GO TO 150
  140 PRINT 330
  150 CONTINUE
*
** PRINT COMPUTED PARAMETERS FOR CURRENT RAY **
*
      CALL WRITER (X,Y,A,H)
*
** PLOT WAVE RAYS IF PLOT OPTION USED **
*
      IF (KPLOT .GT. 0) CALL PLOTR (3)
      NORAY = NORAY+1
      GO TO 120
  160 PRINT 340, NOSET
  170 CONTINUE
      PRINT 350, NOSETS
  180 CONTINUE
*
** END PLOTTING IF PLOT OPTION USED **
*
      IF (KPLOT .GT. 0) CALL PLOTR (4)
      STOP
*
  190 FORMAT (1HO,*INPUT ERROR = NO DATA INPUT UNDER NPUT1*)
  200 FORMAT (1HO,*INPUT ERROR = NO DATA FOR GRID*)
  210 FORMAT (1HO,*INPUT ERROR = NO DATA FOR MI OR MJ OR NOMOD OR NOROW*
     +)
  220 FORMAT (1HO,*INPUT ERROR = NO DATA FOR PLTSCAL*)
```

```
230 FORMAT (1H1,12X,*BASIC PROGRAM PARAMETERS*//
   +2X,*DEPTH UNIT MULT...............DCON        =*,F8.2/
   +2X,*MIN STEP SIZE.................DELTAS      =*,F6.2/
   +2X,*GRID SIZE.....................GRID        =*,F8.2/
   +2X,*DEEP WATER STEP SIZE..........GRINC       =*,F6.2/
   +2X,*OUTPUT UNIT OPTION............IOUTUNT     =*,I2/
   +2X,*PRINT FREQ 1 (D/L>0.5)........JPRTFRQ(1)  =*,I3/
   +2X,*PRINT FREQ 2 (0.25<D/L<0.5)...JPRTFRQ(2)  =*,I3/
   +2X,*PRINT FREQ 3 (D/L<0.25).......JPRTFRQ(3)  =*,I3/
   +2X,*PLOT OPTION...................KPLOT       =*,I2/
   +2X,*MAX NO OF RAY POINTS..........LIMNPT      =*,I4/
   +2X,*GRID LIMITS   ABSCISSA........MI          =*,I4/
   +2X,*              ORDINATE........MJ          =*,I4/
   +2X,*NO OF DEPTH MODULES...........NOMOD       =*,I3/
   +2X,*NO OF ROW MODULES IN NOMOD....NUROW       =*,I3/
   +2X,*NO OF SETS OF RAYS............NOSETS      =*,I3/
   +2X,*PLOT X AXES LENGTH............PLTLNGX     =*,F6.2/
   +2X,*PLOT Y AXES LENGTH............PLTLNGY     =*,F6.2/
   +2X,*PLOT SCALE FACTOR.............PLTSCAL     =*,F6.2/
   +2X,*TIDE HEIGHT...................TIDE        =*,F6.2)
240 FORMAT (1H0,1X,*MAX FINAL RAY SPACING.........DISTMAX     =*,F6.1/
   +2X,*MIN FINAL RAY SPACING.........DISTMIN     =*,F6.1/
   +2X,*INITIAL PT FOR CREST CURVE....KCREST      =*,I4/
   +2X,*MIN ANGLE BETWEEN SEGMENTS....RAYANG      =*,F6.1/
   +2X,*MAX INITIAL RAY SPACING.......RAYMAX      =*,F6.1/
   +2X,*MIN INITIAL RAY SPACING.......RAYMIN      =*,F7.2)
250 FORMAT (8A10)
260 FORMAT (1H0,*INPUT ERROR = NO TITLE CARD*)
270 FORMAT (A2)
280 FORMAT (1H0,*INPUT ERROR = NO DATA INPUT UNDER NPUT2*)
290 FORMAT (1H0,*INPUT ERROR = NO DATA FOR A OR HO OR T*)
300 FORMAT (1H1,8A10/1X,*PERIOD =*,F6.2,* SEC...*,*TIME STEP =*,F7.2,
   +* SEC...*,*RAY SPACING =*,F5.1,*...CREST SPACING =*,I3,
   +*...BOTTOM APPROX CODE =*I2/1X,*RAY NO =*,I3,*...SET NO =*,I3)
310 FORMAT (1H0,*  POINT        X          Y          ANGLE      DEPTH    LENGT
   +H   SPEED   REFRACTION  SHOALING  HEIGHT*)
320 FORMAT (11X,*(N.MI.)   (N.MI.)    (DEG)      (FT)       (FT)     (FT/SEC
   +)     COEF       COEF       (FT)*/)
330 FORMAT (11X,*(N.MI.)   (N.MI.)    (DEG)      (M)        (M)      (M/SEC)
   +      COEF       COEF       (M)*/)
340 FORMAT (1H0,*SET NO*,I3,* ENDED. RAYS LIE OUTSIDE BOUNDARY*)
350 FORMAT (1H0,*ALL SETS COMPLETED. NUMBER OF SETS =*,I3)
    END
```

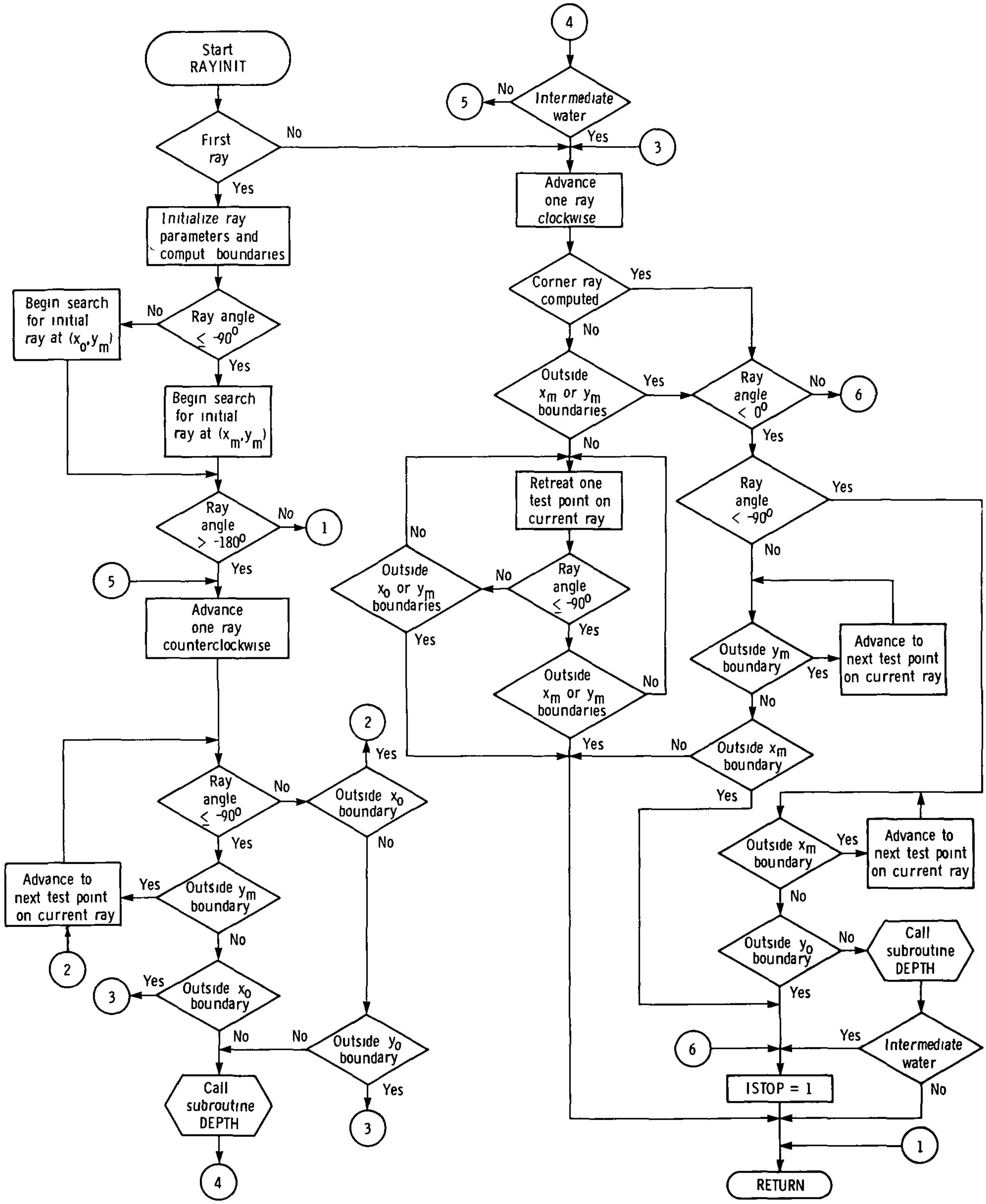
Subroutine RAYINIT

Subroutine RAYINIT computes the coordinates of the initial point on each ray in a particular set. It is assumed that the coastline and isobaths for the geographical region to be considered are aligned in a crude sense with the x-axis. Rays are to be initiated, depending on their initial direction, at a sequence of deepwater points along the seaward computational boundaries of the region (x_o, y_m, x_m), which are set arbitrarily a distance of 4 grid units inward from the limits of the bathymetry grid. (See fig. 1.)

The initial ray in the set is found by forming a counterclockwise sequence of trial rays beginning at a reference corner of the region. For initial ray directions $\alpha_o > -90^o$, the reference corner lies at (x_o, y_m), whereas for $\alpha_o \leq -90^o$, it lies at (x_m, y_m). For $\alpha_o = -180^o$, the initial ray is begun at the reference corner (x_m, y_m), and no sequence of trial rays is formed. For all other initial directions, the counterclockwise sequence of trial rays (illustrated in fig. 1 for $\alpha_o > -90^o$) is formed, with each ray separated by a distance of RAYSPC grid units orthogonal to the initial ray direction. Steps are made inward along each trial ray in increments of DTIC = NCREST*GRINC until the test point along the ray lies within the appropriate seaward computational boundary. If that point lies in deep water, the counterclockwise sequence is continued. If the first test point lying within the computational boundary is in intermediate water, the counterclockwise sequence is terminated and the previous ray in the sequence is chosen as the initial ray in the set. The sequence is also terminated if the entire trial ray lies outside the computational boundaries or if the first intraboundary point along the trial ray lies on land. In these instances, the previous ray in the sequence is also chosen as the initial ray in the set.

The coordinates of the initial point on each ray in the set are found by reversing the procedure and forming a clockwise sequence of rays beginning with the initial ray (illustrated in fig. 2 for $\alpha_o > -90^o$). The initial point along each ray is selected as that trial point advanced along the ray by an increment of DTIC and lying both within and closest to the appropriate seaward boundary. The clockwise sequence (and the ray set) is terminated when a trial ray is formed along which no appropriate initial point can be found.

The flow diagram and listing of subroutine RAYINIT follow.



Start RAYINIT
First ray
Initialize ray parameters and comput boundaries
Ray angle ≤ -90°
Begin search for initial ray at (x_o,y_m)
Begin search for initial ray at (x_m,y_m)
Ray angle > -180°
Advance one ray counterclockwise
Ray angle ≤ -90°
Outside x_o boundary
Outside y_m boundary
Advance to next test point on current ray
Outside x_o boundary
Outside y_o boundary
Call subroutine DEPTH
Intermediate water
Advance one ray clockwise
Corner ray computed
Outside x_m or y_m boundaries
Retreat one test point on current ray
Ray angle ≤ -90°
Outside x_o or y_m boundaries
Outside x_m or y_m boundaries
Ray angle < 0°
Ray angle < -90°
Outside y_m boundary
Advance to next test point on current ray
Outside x_m boundary
Outside x_m boundary
Advance to next test point on current ray
Outside y_o boundary
Call subroutine DEPTH
Intermediate water
ISTOP = 1
RETURN
Yes
No

```
      SUBROUTINE RAYINIT (X,Y,AST)
*
************************************************************************
*                                                                      *
*     THIS SUBROUTINE COMPUTES THE COORDINATES OF THE INITIAL POINT    *
*     FOR EACH RAY IN A CLOCKWISE SEQUENCE ALONG THE SEAWARD           *
*     BOUNDARIES OF THE REGION. EACH RAY IS SEPARATED BY RAYSPC GRID   *
*     UNITS FROM THE PREVIOUS ONE. COMPUTATIONS BEGIN AT A POINT       *
*     ALONG THE RAY WHICH LIES IN DEEP WATER AND IS ADVANCED BY AN     *
*     INTEGER MULTIPLE OF THE DEEP-WATER STEP LENGTH BETWEEN CREST     *
*     TIC MARKS(DTIC) FROM THE CREST WHICH WOULD PASS THROUGH THE      *
*     REFERENCE CORNER.                                                *
*                                                                      *
************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      IF (NORAY .EQ. 1) GO TO 10
      DXRAY=-RAYSPC*SARAD
      DYRAY=RAYSPC*CARAD
      GO TO 110
*
*                          INITIALIZATION SECTION
*
* COMPUTE FIXED DEEP-WATER SEPARATION DISTANCES FOR RAYS AND CREST
* TIC-MARKS. INITIALIZE COMPUTATIONAL BOUNDARIES.
*
   10 XO=4.
      YO=4.
      XM=MI-4
      YM=MJ-4
      ARAD=AST/57.2957795
      SARAD=SIN(ARAD)
      CARAD=COS(ARAD)
      DXRAY=-RAYSPC*SARAD
      DYRAY=RAYSPC*CARAD
      DTIC=NCREST*GRINC
      DXTIC=DTIC*CARAD
      DYTIC=DTIC*SARAD
*
* CHOOSE REFERENCE CORNER - (XO,YM) IF RAY ANGLE > -90 DEG.
*                           (XM,YM) IF RAY ANGLE < -90 DEG.
```

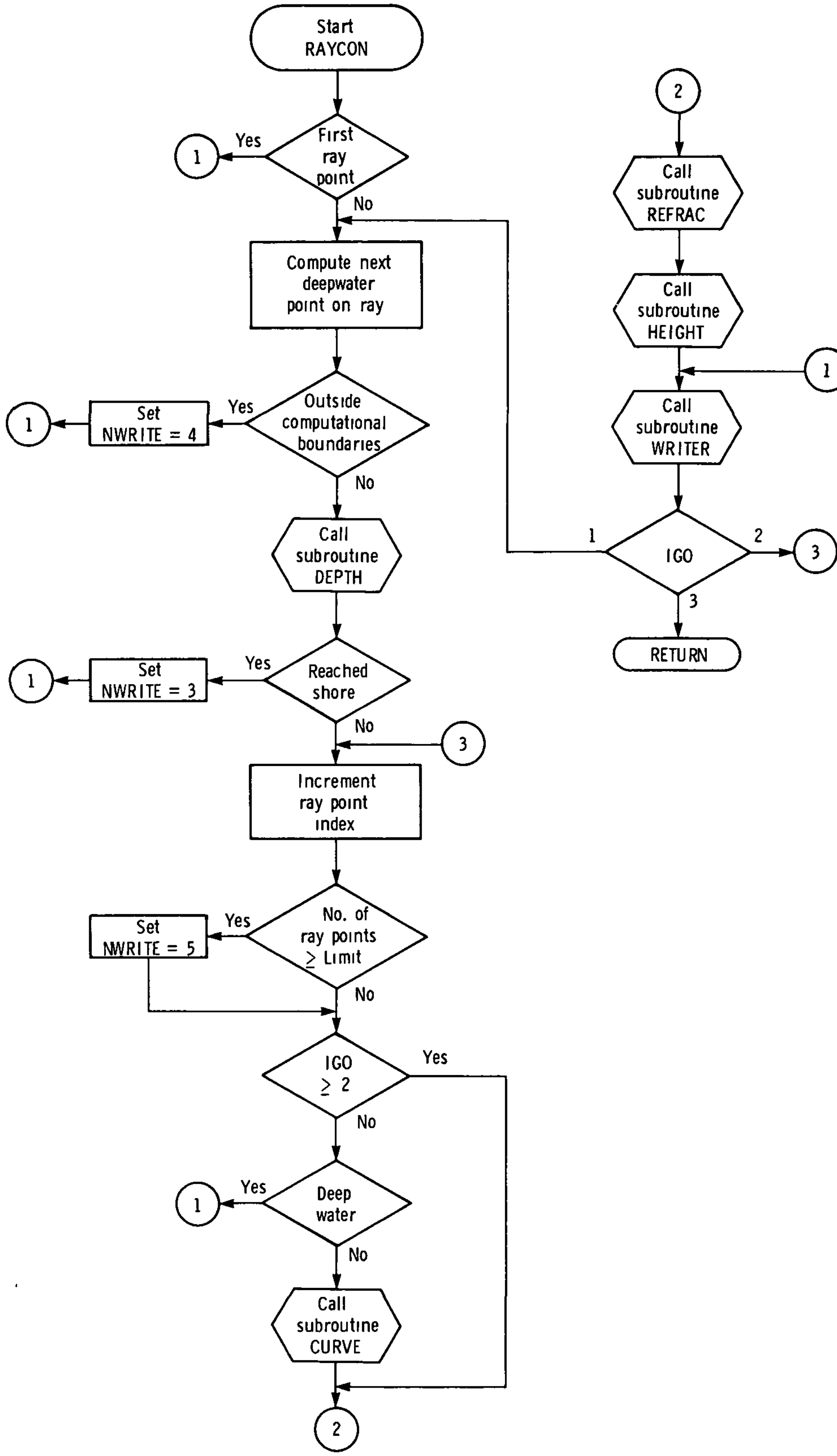
```
*
      IF (AST.LT.-89.99) GO TO 20
      XC=XO
      YC=YM
      GO TO 30
   20 XC=XM
      YC=YM
*
*              LOCATE FIRST RAY IN SET
*
* IF RAY ANGLE IS -180 DEG, FIRST RAY BEGINS AT REFERENCE CORNER.
*
   30 IF (AST.GT.-179.99) GO TO 40
      X=XTEST=XC
      Y=YTEST=YC
      NTIC=NCORN=1
      RETURN
*
* ANGLE > -180 DEG. START SEARCH FOR INITIAL RAY AT REFERENCE CORNER.
*
   40 NTIC=1
      NCORN=0
      XTEST=XC
      YTEST=YC
*
* ADVANCE COUNTERCLOCKWISE ONE RAY AND MOVE TO CORRESPONDING TEST POINT.
*
   50 XTEST=XTEST-DXRAY
      YTEST=YTEST-DYRAY
      NCORN=NCORN+1
*
* BASED ON ANGLE, TEST TO SEE IF CURRENT TEST POINT IS INSIDE REGION.
*
   60 IF (AST.LT.-89.99) GO TO 70
      IF (XTEST.LT.XO) GO TO 90
      IF (YTEST.LT.YO) GO TO 100
      GO TO 80
   70 IF (YTEST.GT.YM+0.01) GO TO 90
      IF (XTEST.LT.XO) GO TO 100
*
* INTERIOR TEST POINT FOUND. CHECK LOCAL DEPTH.
*
   80 CALL DEPTH (XTEST,YTEST)
*
* IF DEEP WATER, ADVANCE COUNTERCLOCKWISE ONE RAY AND MOVE TO
* CORRESPONDING TEST POINT. IF INTERMEDIATE WATER, THE PREVIOUS
* RAY IS THE INITIAL RAY - GO TO STATEMENT 110 TO MOVE TO INITIAL RAY.
*
      IF (DXY.LE.DRC) GO TO 110
```

```
      GO TO 50
*
* CURRENT TEST POINT OUTSIDE REGION. ADVANCE ONE TEST POINT
* ALONG CURRENT RAY.
*
  90  XTEST=XTEST+DXTIC
      YTEST=YTEST+DYTIC
      NTIC=NTIC+1
      GO TO 60
*
* ALL TEST POINTS ON TRIAL RAY OUTSIDE REGION. SET PREVIOUS RAY
* TO THE INITIAL RAY.
*
  100 CONTINUE
*
* ADVANCE CLOCKWISE ONE RAY AND MOVE TO CORRESPONDING TEST POINT.
*
  110 XINIT=XTEST+DXRAY
      YINIT=YTEST+DYRAY
*
* IF THE RAY PASSING THROUGH THE REFERENCE CORNER HAS BEEN COMPUTED,
* CRITERIA FOR ENDING RAY SET MUST BE TESTED. IF NOT, MAKE SURE THE
* PRESENT TEST POINT IS THE FIRST INTERIOR TEST POINT.
*
      IF (NORAY.GT.NCORN) GO TO 160
      IF (XINIT.GT.XM+.01.OR.YINIT.GT.YM+.01) GO TO 160
  120 XTEMP=XINIT-DXTIC
      YTEMP=YINIT-DYTIC
      IF (AST.LT.-89.99) GO TO 130
      IF(XTEMP.LT.X0-0.01.OR.YTEMP.GT.YM+0.01) GO TO 150
      GO TO 140
  130 IF (XTEMP.GT.XM+.01.OR.YTEMP.GT.YM+.01) GO TO 150
*
* NEW INITIAL TEST POINT. RESET COORDINATES AND REDUCE VALUE OF TIC-MARK
* COUNTER BY 1. GO BACK TO STATEMENT 120 AND TRY AGAIN.
*
  140 XINIT=XTEMP
      YINIT=YTEMP
      NTIC=NTIC-1
      GO TO 120
*
* CORRECT INITIAL TEST POINT FOUND.
*
  150 X=XTEST=XINIT
      Y=YTEST=YINIT
      RETURN
*
* RAY PASSING THROUGH REFERENCE CORNER HAS ALREADY BEEN COMPUTED.
* TEST VARIOUS CRITERIA FOR ENDING RAY SET.
```

```
*
  160 IF (AST.LT.-.01) GO TO 170
*
* ZERO DEGREE CASE. FINAL RAY ALREADY COMPUTED.
*
      ISTOP=1
      RETURN
  170 IF (AST.LT.-90.01) GO TO 210
*
* RAY ANGLE > -90 DEG. CHECK LOCATION OF PRESENT TEST POINT.
*
  180 IF (YINIT.LE.YM) GO TO 190
*
* Y-COORDINATE OF CURRENT TEST POINT GREATER THAN YM. ADVANCE ONE
* TEST POINT AND RECHECK LOCATION.
*
      XINIT=XINIT+DXTIC
      YINIT=YINIT+DYTIC
      NTIC=NTIC+1
      GO TO 180
  190 IF (XINIT.LT.XM) GO TO 200
*
* X-COORDINATE OF CURRENT TEST POINT GREATER THAN XM. NO FURTHER RAYS
* POSSIBLE.
*
      ISTOP=1
      RETURN
  200 X=XTEST=XINIT
      Y=YTEST=YINIT
      RETURN
*
* RAY ANGLE < -90 DEG. CHECK LOCATION OF CURRENT TEST POINT.
*
  210 IF (XINIT.LE.XM+0.001) GO TO 220
*
* X-COORDINATE OF CURRENT TEST POINT GREATER THAN XM. ADVANCE ONE TEST POINT
* AND RECHECK LOCATION.
*
      XINIT=XINIT+DXTIC
      YINIT=YINIT+DYTIC
      NTIC=NTIC+1
      GO TO 210
  220 IF(YINIT.LE.YO) GO TO 230
*
* INTERIOR TEST POINT FOUND. CHECK LOCAL DEPTH.
*
      CALL DEPTH (XINIT,YINIT)
      IF (DXY.LE.DRC) GO TO 230
*
* NEW RAY CAN BE COMPUTED
*
      X=XTEST=XINIT
      Y=YTEST=YINIT
      RETURN
*
* INITIAL TEST POINT LIES EITHER OUTSIDE REGION OR IN INTERMEDIATE WATER
* NO FURTHER RAYS POSSIBLE.
*
  230 ISTOP=1
      RETURN
      END
```

Subroutine RAYCON

Subroutine RAYCON controls the computation of each point along the wave ray. Points which lie in deep water are computed directly in RAYCON, while the routine calls subroutine REFRAC to compute ray points lying in shoaling water. After the ray point is computed, subroutine HEIGHT is called to compute the wave height and subroutine WRITER is called to store computed parameters for printing and plotting. Computation of the ray path is terminated if the ray goes outside the computational boundaries, if the shore is reached, or if the maximum number of ray computational points is exceeded. The program flag NWRITE designates the type of termination condition which is met. The different phases of the ray path are indicated by the program parameter IGO. When the ray is in deep water, IGO = 1. When the ray reaches shoaling water, IGO is set to 2 in subroutine REFRAC. When a termination condition has been reached, IGO is set to 3 in subroutine WRITER. The flow diagram and listing of subroutine RAYCON follow.



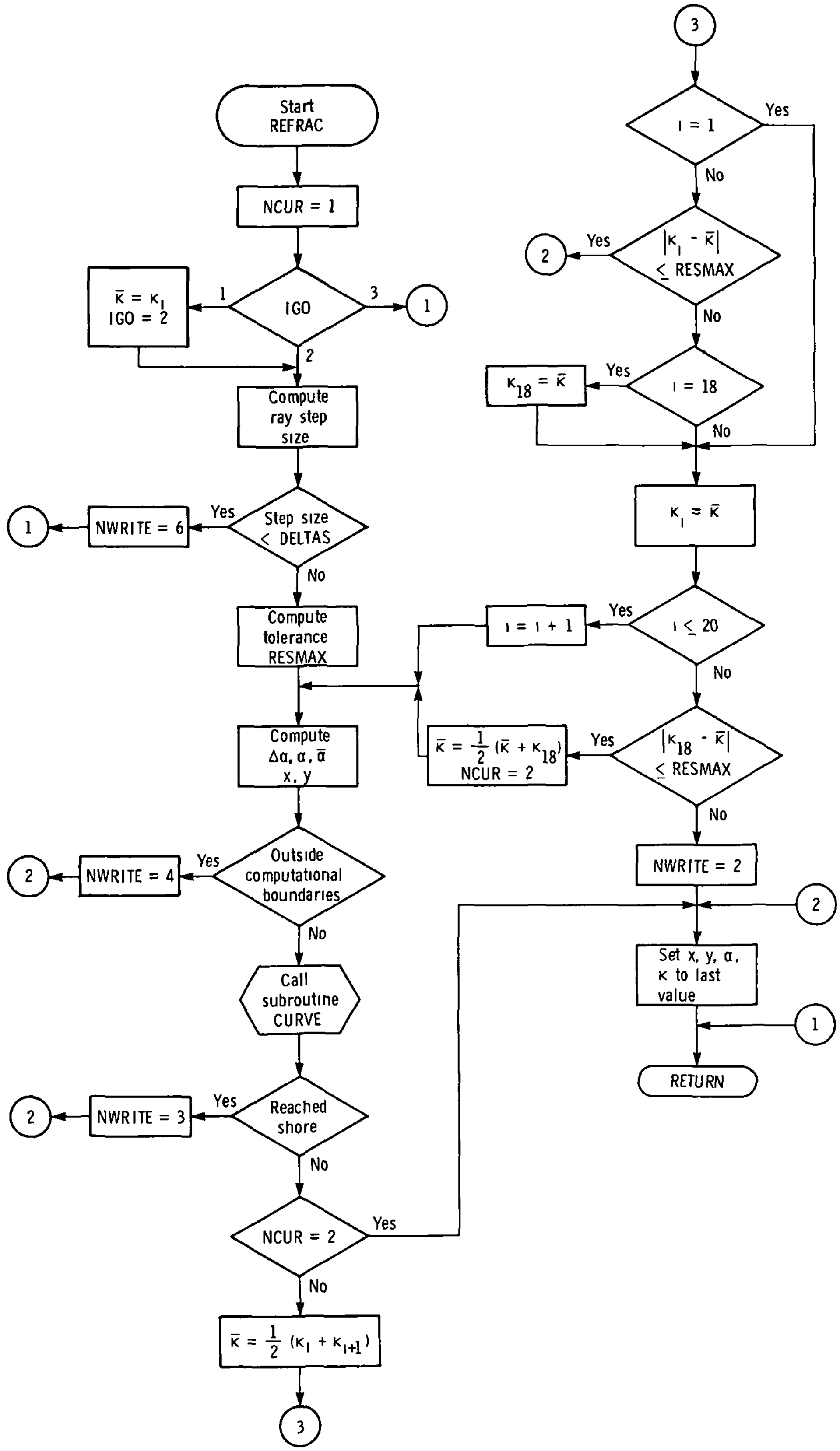
Start RAYCON
First ray point
Yes
No
1
Compute next deepwater point on ray
Outside computational boundaries
Yes
Set NWRITE = 4
No
Call subroutine DEPTH
Reached shore
Yes
Set NWRITE = 3
No
3
Increment ray point index
No. of ray points ≥ Limit
Yes
Set NWRITE = 5
No
IGO ≥ 2
Yes
No
Deep water
Yes
No
Call subroutine CURVE
2
Call subroutine REFRAC
Call subroutine HEIGHT
Call subroutine WRITER
IGO
RETURN

```
      SUBROUTINE RAYCON (X,Y,A)
*
******************************************************************************
*                                                                            *
*     THIS SUBROUTINE ASSUMES DEEP WATER AND CALCULATES THE NEXT             *
*     POINT ON THE RAY. IF THE WAVE HAS ENTERED SHOALING WATER, THE          *
*     PREVIOUSLY CALCULATED POSITION OF THE NEXT POINT IS DISCARDED.         *
*     SUBROUTINE CURVE IS CALLED TO CALCULATE THE INITIAL RAY                *
*     CURVATURE, WHICH IS USED IN SUBROUTINE REFRAC TO CALCULATE THE         *
*     NEXT POINT ON THE RAY.                                                 *
*                                                                            *
******************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
** SET RAY INITIAL CONDITIONS **
*
      ANG = A/57.2957795
      COSA = COS(ANG)
      SINA = SIN(ANG)
      H = HO
      IGO = 1
      IF (NPT .EQ. 1) GO TO 70
   10 PX = X
      PY = Y
*
** COMPUTE NEXT POINT ON RAY FOR DEEP WATER **
*
      DS = GRINC
      X = X+DS*COSA
      Y = Y+DS*SINA
*
** TEST TO SEE IF RAY POINT OUTSIDE COMPUTATIONAL BOUNDARIES **
*
      IF (X .GE. XM .OR. X .LE. XO) GO TO 20
      IF (Y .LT. YM .AND. Y .GT. YO) GO TO 30
*
* OUTSIDE COMPUTATIONAL BOUNDARIES. SET FLAG AND EXIT **
*
   20 NWRITE = 4
      GO TO 70
*
```

```
** COMPUTE DEPTH **
*
  30  CALL DEPTH (X,Y)
*
** TEST TO SEE IF SHORE HAS BEEN REACHED **
*
      IF (DXY .GT. 0.) GO TO 40
*
** SHORE REACHED. SET FLAG AND EXIT **
*
      NWRITE = 3
      GO TO 70
  40  NPT = NPT + 1
*
** TEST TO SEE IF MAX NO OF RAY POINTS REACHED **
*
      IF (NPT .LT. LIMNPT) GO TO 50
*
** MAX NO OF POINTS REACHED. SET FLAG **
*
      NWRITE = 5
*
** TEST TO SEE IF IN SHOALING WATER **
*
  50  IF (IGO .GE. 2) GO TO 60
      IF (DXY .GE. DRC) GO TO 70
*
** IN SHOALING WATER. START REFRACTION CALCULATIONS **
*
      X = PX
      Y = PY
*
** COMPUTE INITIAL RAY CURVATURE **
*
      CALL CURVE (X,Y,ANG,FK)
*
** COMPUTE NEXT POINT ON RAY FOR SHOALING WATER **
*
  60  CALL REFRAC (X,Y,ANG,FK)
*
** COMPUTE WAVE HEIGHT **
*
      CALL HEIGHT (ANG,H)
*
** SAVE PRINT AND PLOT DATA **
*
  70  A = 57.2957795*ANG
      CALL WRITER (X,Y,A,H)
*
** REPEAT FOR NEXT POINT ON RAY **
*
      IF (IGO-2) 10,40,80
  80  RETURN
      END
```

Subroutine REFRAC

In subroutine REFRAC, equations (1) to (3) are solved iteratively to find the next point on the ray. If the curvature is oscillating between two values, the curvature is taken as the average of these two values. If the iteration process is converging very slowly or not at all, the ray is stopped. Subroutine CURVE is called to compute the curvature. Several ray termination conditions are checked in this subroutine. The ray is terminated if it goes outside the computational boundaries or if it reaches the shore. In addition, the ray is terminated if the wave speed has been reduced so that the incremental distance between successive ray points is less than the minimum step length allowed DELTAS which is an input parameter. The ray path phase flag IGO is set to 2 in this routine. The flow diagram and listing of subroutine REFRAC follow.

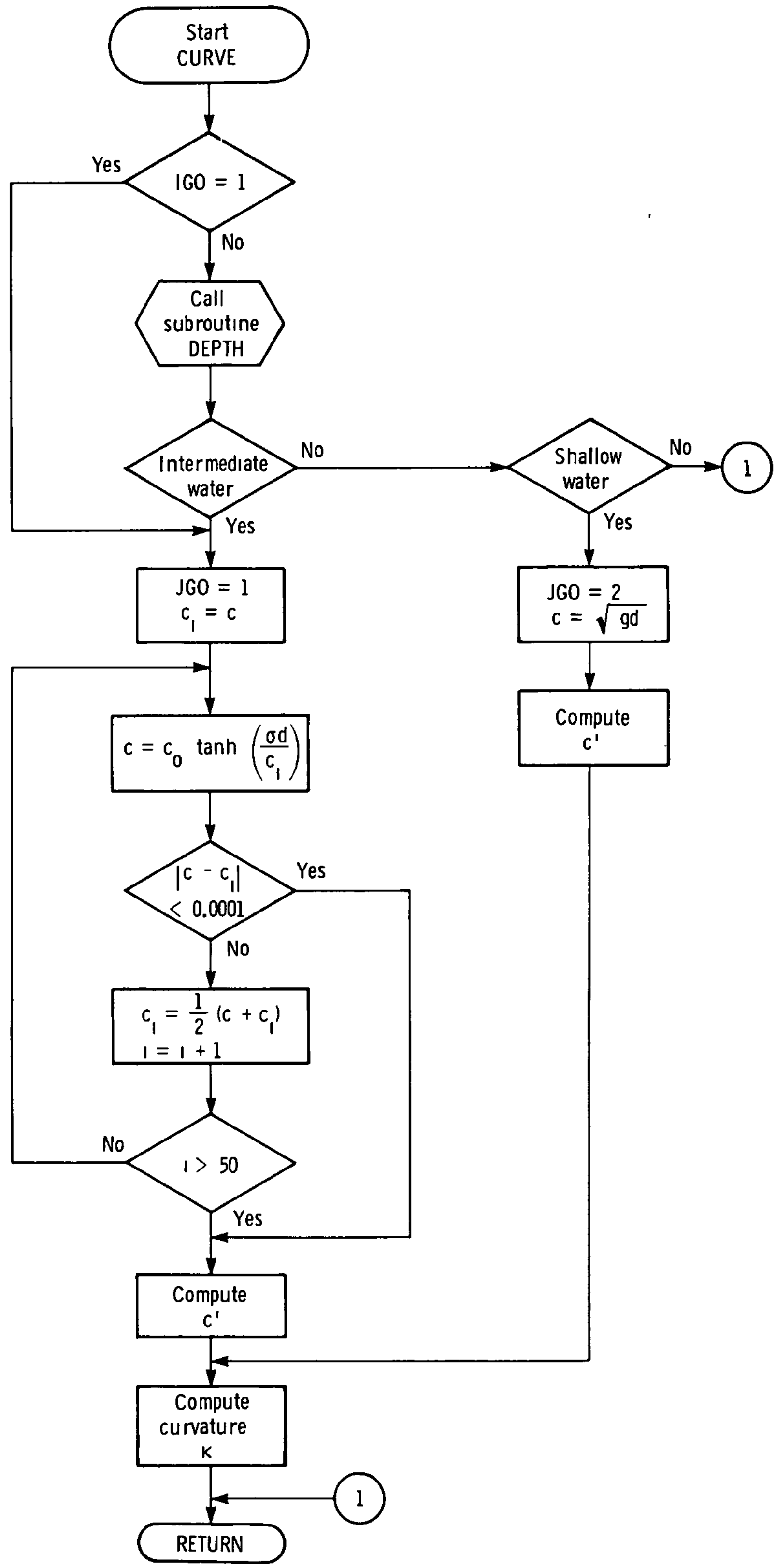

Start
REFRAC
NCUR = 1
IGO
$\bar{\kappa} = \kappa_1$
IGO = 2
Compute
ray step
size
Step size
< DELTAS
NWRITE = 6
Compute
tolerance
RESMAX
Compute
$\Delta\alpha, \alpha, \bar{\alpha}$
x, y
Outside
computational
boundaries
NWRITE = 4
Call
subroutine
CURVE
Reached
shore
NWRITE = 3
NCUR = 2
$\bar{\kappa} = \frac{1}{2}(\kappa_i + \kappa_{i+1})$
i = 1
$|\kappa_i - \bar{\kappa}|$
≤ RESMAX
i = 18
$\kappa_{18} = \bar{\kappa}$
$\kappa_i = \bar{\kappa}$
i ≤ 20
i = i + 1
$|\kappa_{18} - \bar{\kappa}|$
≤ RESMAX
$\bar{\kappa} = \frac{1}{2}(\bar{\kappa} + \kappa_{18})$
NCUR = 2
NWRITE = 2
Set x, y, α,
κ to last
value
RETURN
Yes
No

```
      SUBROUTINE REFRAC (X,Y,A,FK)
*
**************************************************************************
*                                                                        *
*     THIS SUBROUTINE SOLVES THE REFRACTION EQUATIONS ITERATIVELY TO     *
*     FIND THE NEXT POINT ON THE RAY. IF THE ITERATIVE PROCESS IS        *
*     OSCILLATING BETWEEN TWO VALUES, THEY ARE AVERAGED. IF THE          *
*     PROCEDURE IS CONVERGING VERY SLOWLY, OR NOT AT ALL,THE RAY IS      *
*     STOPPED.                                                           *
*                                                                        *
**************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      NCUR = 1
      IF (IGO-2) 10,20,110
*
** SET INITIAL CONDITIONS **
*
   10 FKM = FK
      IGO = 2
   20 DS = CXY*DTGR
*
** TEST TO SEE IF STEP SIZE TOO SMALL **
*
      IF (DS .GE. DELTAS) GO TO 30
*
** STEP SIZE TOO SMALL. SET FLAG AND EXIT **
*
      NWRITE = 6
      GO TO 110
*
** COMPUTE TOLERANCE ON CONVERGENCE FOR CURVATURE **
*
   30 RESMAX = 0.00005/DS
*
** ITERATE FOR CURVATURE SOLUTION AND NEXT POINT ON RAY **
*
   40 DO 80 I=1,20
      DELA = FKM*DS
      AA = A+DELA
      AM = A+0.5*DELA
```

```
      XX = X+DS*COS(AM)
      YY = Y+DS*SIN(AM)
*
** TEST TO SEE IF RAY POINT OUTSIDE COMPUTATIONAL BOUNDARIES **
*
      IF (XX .GE. XM .OR. XX .LE. XO) GO TO 50
      IF (YY .LT. YM .AND. YY .GT. YO) GO TO 60
*
** OUTSIDE COMPUTATIONAL BOUNDARIES. SET FLAG AND EXIT **
*
   50 NWRITE = 4
      GO TO 100
*
** COMPUTE RAY CURVATURE **
*
   60 CALL CURVE (XX,YY,AA,FKK)
*
** TEST TO SEE IF SHORE HAS BEEN REACHED **
*
      IF (DXY .GT. 0.) GO TO 70
*
** SHORE REACHED. SET FLAG AND EXIT **
*
      NWRITE = 3
      GO TO 100
   70 IF (NCUR .EQ. 2) GO TO 100
      FKM = 0.5*(FK+FKK)
      IF (I .EQ. 1) GO TO 80
*
** TEST TO SEE IF SOLUTION HAS CONVERGED **
*
      IF (ABS(FKP-FKM) .LE. RESMAX) GO TO 100
      IF (I .EQ. 18) FK18 = FKM
   80 FKP = FKM
*
** TEST TO SEE IF SOLUTION HAS CONVERGED **
*
      IF (ABS(FKM-FK18) .LE. RESMAX) GO TO 90
*
** DID NOT CONVERGE. SET FLAG AND EXIT **
*
      NWRITE = 2
      GO TO 100
*
** CURVATURE AVERAGE **
*
   90 FKM = 0.5*(FKM+FK18)
      NCUR = 2
      GO TO 40
  100 X = XX
      Y = YY
      A = AA
      FK = FKK
  110 RETURN
      END
```

Subroutine CURVE

Subroutine CURVE initially calls subroutine DEPTH to determine whether the ray is in intermediate water or shallow water. It then sets the flag JGO to either 1 for intermediate water or 2 for shallow water. The local wave speed c and the rate of change of wave speed with depth c' are then computed according to the appropriate equations. By using these values, the ray curvature is computed. Several parameters required in subroutine HEIGHT for computation of the wave separation factor β are computed in this routine. The flow diagram and listing of subroutine CURVE follow.

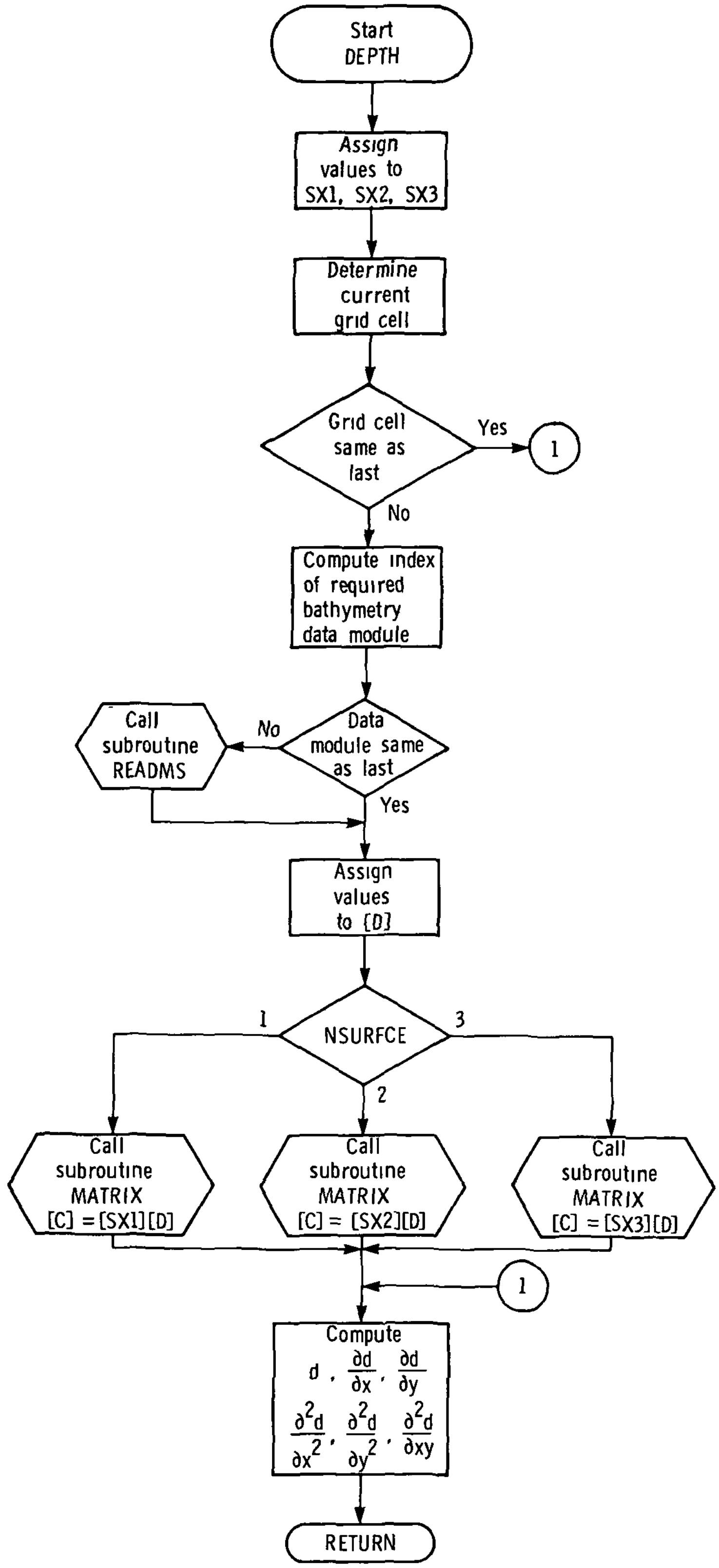

Start
CURVE
IGO = 1
Yes
No
Call
subroutine
DEPTH
Intermediate
water
No
Yes
Shallow
water
No
1
Yes
JGO = 1
$c_1 = c$
JGO = 2
$c = \sqrt{gd}$
Compute
c'
$c = c_0 \tanh\left(\frac{\sigma d}{c_1}\right)$
$|c - c_1|$
< 0.0001
Yes
No
$c_1 = \frac{1}{2}(c + c_1)$
$i = i + 1$
No
$i > 50$
Yes
Compute
c'
Compute
curvature
κ
1
RETURN

```
      SUBROUTINE CURVE (X,Y,A,FK)
*
***********************************************************************
*                                                                     *
*     THIS SUBROUTINE CHECKS THE LOCAL DEPTH, CALCULATES THE WAVE     *
*     SPEED FOR THE APPROPRIATE REGIME, COMPUTES THE RATE OF CHANGE   *
*     OF SPEED WITH RESPECT TO DEPTH, AND CALCULATES THE CURVATURE    *
*     AT THE SPECIFIED RAY POINT.                                     *
*                                                                     *
***********************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      IF (IGO .EQ. 1) GO TO 20
*
** COMPUTE DEPTH **
*
      CALL DEPTH (X,Y)
*
** TEST TO SEE IF IN INTERMEDIATE WATER **
*
      IF (200.0*DXY .GT. WL) GO TO 20
*
** IN SHALLOW WATER **
*
** TEST TO SEE IF SHORE HAS BEEN REACHED **
*
      IF (DXY . GT. 0.) GO TO 10
*
** SHORE REACHED. EXIT SUBROUTINE **
*
      GO TO 60
   10 JGO = 2
*
** COMPUTE WAVE SPEED FOR SHALLOW WATER **
*
      ARG = 32.1725*DXY
      CXY = SQRT(ARG)
*
** COMPUTE CPRIME FOR SHALLOW WATER **
*
      CPRIME = 16.08625/CXY
```

```
      GO TO 50
*
** IN INTERMEDIATE WATER **
*
   20 CI = CXY
      JGO = 1
*
** ITERATE TO SOLVE FOR INTERMEDIATE WATER WAVE SPEED **
*
      DO 30 I=1,50
      ARG = (DXY*SIG)/CI
      CXY = CO*TANH(ARG)
      RESID = CXY - CI
      IF (ABS(RESID) .LT. 0.0001) GO TO 40
   30 CI = 0.5*(CXY+CI)
*
** COMPUTE INTERMEDIATE WATER CPRIME **
*
   40 RCCO = CXY/CO
      SCMC = (1.-RCCO*RCCO)*SIG
      V = SCMC*DXY+RCCO*CXY
      CPRIME = CXY*SCMC/V
*
** COMPUTE RAY CURVATURE **
*
   50 FK = (SIN(A)*PDX-COS(A)*PDY)*CPRIME/CXY
   60 RETURN
      END
```

Subroutine DEPTH

Subroutine DEPTH first determines the bathymetry data module in which the current ray point resides. If the point lies outside the boundaries of the module currently residing in central memory, the required module is read into memory by using the system subroutine READMS. Subroutine DEPTH selects the subset of data points in the module that are to be used to compute the bottom topography approximation surface. It then, on option, computes coefficients to a least squares quadratic, a least squares cubic, or a constrained bicubic polynomial which is used to approximate the local topography. The depth at the current ray point is then computed by evaluating the approximating polynomial at that point. Partial derivatives required in subroutine HEIGHT for computing the wave separation factor β are also computed by evaluating analytical partial derivatives of the approximating polynomial. The input parameters TIDE and DCON are used to simulate a constant tide effect and to accommodate various units for the bathymetry data values. The flow diagram and listing of subroutine DEPTH follow.


Start
DEPTH
Assign values to SX1, SX2, SX3
Determine current grid cell
Grid cell same as last
Yes
1
No
Compute index of required bathymetry data module
Data module same as last
No
Call subroutine READMS
Yes
Assign values to {D}
NSURFCE
1
2
3
Call subroutine MATRIX [C] = [SX1][D]
Call subroutine MATRIX [C] = [SX2][D]
Call subroutine MATRIX [C] = [SX3][D]
1
Compute
d, ∂d/∂x, ∂d/∂y
∂²d/∂x², ∂²d/∂y², ∂²d/∂xy
RETURN

```
      SUBROUTINE DEPTH (X,Y)
*
************************************************************************
*                                                                      *
*     THIS SUBROUTINE CALLS INTO CORE THE BATHYMETRY DATA MODULE IN    *
*     WHICH THE CURRENT POINT RESIDES. IT THEN FITS A LEAST SQUARES    *
*     QUADRATIC OR CUBIC POLYNOMIAL TO THE DEPTH VALUES AT 12 GRID     *
*     POINTS SURROUNDING THE SPECIFIED POINT. ALTERNATIVELY, A         *
*     CONSTRAINED BICUBIC POLYNOMIAL CAN BE USED TO FIT THE DATA BY    *
*     USING 16 GRID POINTS. THE DEPTH AND THE PARTIAL DERIVATIVES      *
*     CAN THEN BE EVALUATED BY USING THE POLYNOMIAL APPROXIMATION.     *
*                                                                      *
************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      DIMENSION C(16),D(16),SX1(6,12),SX2(10,12),SX3(16,16)
*
*  INPUT CONSTANT MATRIX FOR QUADRATIC LEAST SQUARES, SX1, BY COLUMNS .
*
      DATA (SX1(I),I=1,72)/.30861244,2*.05322967,-.125,.05263158,
     +-.125,.23684211,.19677034,.10586124,-.125,-.05263158,-.125,
     +.21770335,2*.14413876,-.125,.05263158,-.125,.23684211,.10586124,
     +.19677034,-.125,-.05263158,-.125,-.08492823,.090311,.03349282,
     +.125,-.15789474,0.,-.05143541,-.06758373,-.03349282,.125,.15789474
     +,0.,-.05143541,-.03349282,-.06758373,0.,.15789474,.125,-.08492823,
     +.03349282,.090311,0.,-.15789474,.125,.00598086,-.18241627,
     +.12440191,.125,-.15789474,0.,.13038278,-.34031101,-.12440191,.125,
     +.15789474,0.,.13038278,-.12440191,-.34031101,0.,.15789474,.125,
     +.00598086,.12440191,-.18241627,0.,-.15789474,.125/
*
*  INPUT CONSTANT MATRIX FOR CUBIC LEAST SQUARES, SX2, BY COLUMNS
*
      DATA (SX2(I),I=1,120)/.3125,2*-.625,-.125,.05263158,-.125,4*.25,
     +.3125,.625,-.625,-.125,-.05263158,-.125,-.25,.25,-.25,.25,.3125,
     +2*.625,-.125,.05263158,-.125,4*-.25,.3125,-.625,.625,-.125,
     +-.05263158,-.125,.25,-.25,.25,-.25,-.03125,-.02083333,.0625,.125,
     +-.1578947,0.,.08333333,-.25,2*0.,-.03125,-.02083333,-.0625,.125,
     +.1578947,0.,.08333333,.25,2*0.,-.03125,-.0625,-.02083333,0.,
     +.1578947,.125,2*0.,.25,.08333333,-.03125,.0625,-.02083333,0.,
     +-.1578947,.125,2*0.,-.25,.08333333,-.03125,.02083333,-.0625,
     +.125,-.1578947,0.,-.08333333,.25,2*0.,-.03125,.02083333,.0625,
     +.125,.1578947,0.,-.08333333,-.25,2*0.,-.03125,.0625,.02083333,
     +0.,.1578947,.125,2*0.,-.25,-.08333333,-.03125,-.0625,.02083333,
     +0.,-.1578947,.125,2*0.,.25,-.08333333/
*
*  INPUT CONSTANT MATRIX FOR CONSTRAINED BICUBIC INTERPOLATION, SX3, BY COLUMNS
*
      DATA (SX3(I),I=1,256)/1.,2*0.,-3.,0.,-3.,2.,2*0.,2.,0.,9.,-6.,0.,
     +-6.,4.,0.,1.,0.,-2.,2*0.,1.,0.,-3.,0.,2.,6.,-4.,0.,-3.,2.,2*0.,1.,
     +2*0.,-2.,0.,-3.,0.,1.,0.,6.,-3.,2.,-4.,2.,4*0.,1.,2*0.,2*-2.,0.,
     +1.,4.,-2.,1.,-2.,1.,3*0.,3.,2*0.,-2.,4*0.,-9.,6.,0.,6.,-4.,3*0.,
     +-1.,2*0.,1.,4*0.,3.,-2.,0.,-3.,2.,7*0.,3.,3*0.,-6.,3.,-2.,4.,-2.,
     +7*0.,-1.,3*0.,2.,-1.,1.,-2.,1.,11*0.,9.,-6.,0.,-6.,4.,11*0.,-3.,
     +2.,0.,3.,-2.,11*0.,-3.,3.,0.,2.,-2.,11*0.,1.,-1.,0.,-1.,1.,5*0.,
     +3.,3*0.,-2.,0.,-9.,6.,0.,6.,-4.,8*0.,3.,0.,-2.,-6.,4.,0.,3.,-2.,
     +5*0.,-1.,3*0.,1.,0.,3.,-3.,0.,-2.,2.,8*0.,-1.,0.,1.,2.,-2.,0.,
     +-1.,1./
```

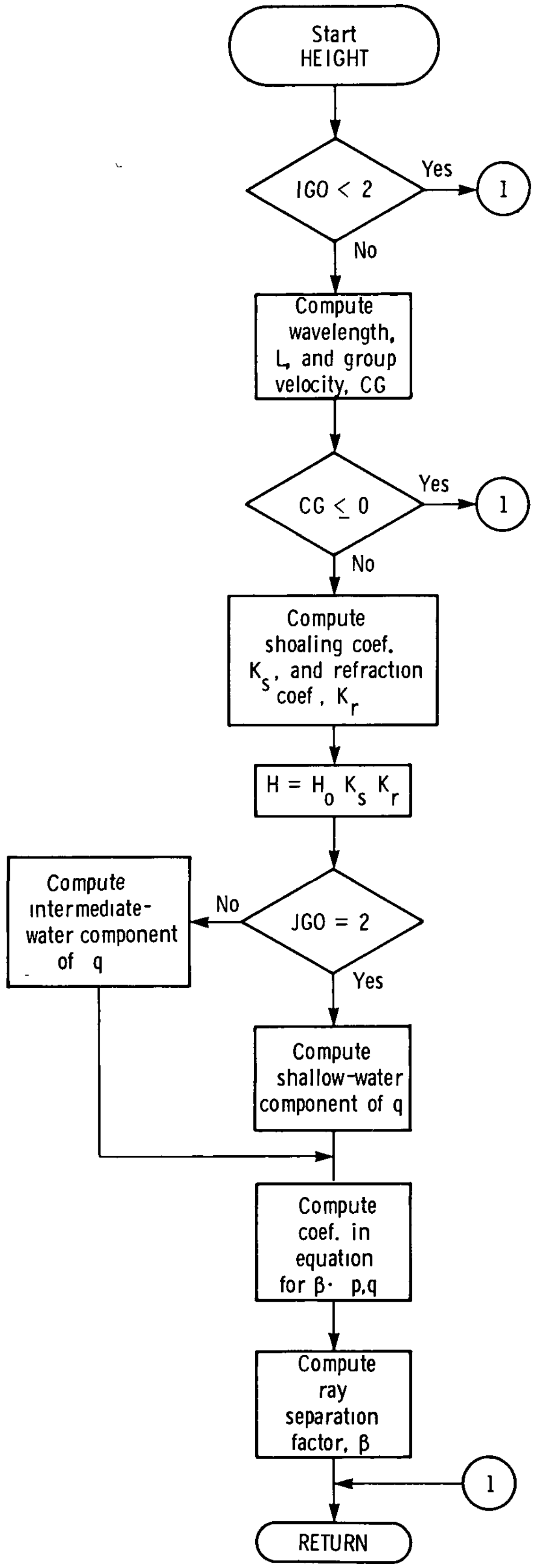
```
*
      IF (NSURFCE .EQ. 2) GO TO 10
*
*  PLACE LOCAL ORIGIN AT THE CENTRAL CELL CORNER CLOSEST TO X AND Y AXES FOR
*  CONSTRAINED BICUBIC INTERPOLATION AND QUADRATIC LEAST-SQUARES
*
      XP=AMOD (X,1.)
      YP=AMOD (Y,1.)
      GO TO 20
*
*
*  PLACE LOCAL ORIGIN AT THE CENTER OF THE CENTRAL CELL FOR CUBIC LEAST-SQUARES
*
   10 XP=AMOD (X,1.)-.5
      YP=AMOD (Y,1.)-.5
*
*  CALCULATE REGIONAL INDICES OF NODAL POINT P1 AND THE MODULE ROW AND
*  COLUMN NUMBERS USING GENERAL EQUATIONS.
*
   20 I1=X+1.
      IM=I1/39+1
      J1=Y+1.
      JM=J1/29+1
*
      IF (NPT.EQ.1) GO TO 30
*
* IS CURRENT CELL SAME AS THE PREVIOUS ONE.  IF SO, THERE IS NO NEED TO DO
* ANYTHING AND THE PROGRAM SKIPS TO CALCULATION OF THE LOCAL DEPTH, DXY.
*
      IF (I1.NE.I1P) GO TO 30
      IF (J1.EQ.J1P) GO TO 150
   30 I1P=I1
      J1P=J1
*
*  CALCULATE LOCAL INDICES OF NODAL POINT P1 USING GENERAL EQUATIONS.
*
      IL1=MOD (I1,39)
      JL1=MOD (J1,29)
      IF (IL1.GT.1) GO TO 40
*
*  IF IL1 IS LESS THAN 2, WE MUST USE THE ALTERNATE EQUATIONS FOR MODULE
*  ROW NUMBER AND LOCAL ROW INDEX.  THUS, SET IMADD=-1 AND IL1ADD=39.
*
      IMADD=-1
      IL1ADD=39
      GO TO 50
   40 IMADD=0
      IL1ADD=0
   50 IF (JL1.GT.1) GO TO 60
*
*  IF JL1 IS LESS THAN 2, WE MUST USE THE ALTERNATE EQUATIONS FOR MODULE
*  COLUMN NUMBER AND LOCAL COLUMN INDEX.  THUS, SET JMADD=-1 AND JL1ADD=29.
*
      JMADD=-1
      JL1ADD=29
      GO TO 70
   60 JMADD=0
      JL1ADD=0
   70 IM=IM+IMADD
      JM=JM+JMADD
      I=IL1+IL1ADD
      J=JL1+JL1ADD
      INDEX=(JM-1)*NOROW+IM
      IF (NPT.EQ.1) GO TO 80
```

```
*
* IF INDEX = INDEXP, THE CURRENT MODULE IS THE SAME AS THE PREVIOUS ONE
* AND THERE IS NO NEED TO READ THE RANDOM-ACCESS FILE.  SKIP TO FILLING
* THE D ARRAY.
*
      IF (INDEX.EQ.INDEXP) GO TO 90
 80   INDEXP=INDEX
*
*  READ MASS STORAGE DISK FILE (TAPE1) CONTAINING BATHYMETRY DATA MODULE
*
      CALL READMS (1,DEP,1344,INDEX)
*
 90   IF (NSURFCE.EQ.3) GO TO 100
*
*  FILL VECTOR FOR QUADRATIC OR CUBIC LEAST SQUARES APPROXIMATION
*
      D(1) = DEP(I,J)
      D(2) = DEP(I+1,J)
      D(3) = DEP(I+1,J+1)
      D(4) = DEP(I,J+1)
      D(5) = DEP(I+2,J)
      D(6) = DEP(I+2,J+1)
      D(7) = DEP(I+1,J+2)
      D(8) = DEP(I,J+2)
      D(9) = DEP(I-1,J+1)
      D(10) = DEP(I-1,J)
      D(11) = DEP(I,J-1)
      D(12) = DEP(I+1,J-1)
      GO TO 110
*
*  FILL VECTOR FOR CONSTRAINED BICUBIC INTERPOLATION
*
 100  D(1)=DEP(I,J)
      D(2)=0.5*(DEP(I+1,J)-DEP(I-1,J))
      D(3)=0.5*(DEP(I,J+1)-DEP(I,J-1))
      D(4)=0.25*(DEP(I+1,J+1)-DEP(I+1,J-1)-DEP(I-1,J+1)+DEP(I-1,J-1))
      D(5)=DEP(I+1,J)
      D(6)=0.5*(DEP(I+2,J)-DEP(I,J))
      D(7)=0.5*(DEP(I+1,J+1)-DEP(I+1,J-1))
      D(8)=0.25*(DEP(I+2,J+1)-DEP(I+2,J-1)-DEP(I,J+1)+DEP(I,J-1))
      D(9)=DEP(I+1,J+1)
      D(10)=0.5*(DEP(I+2,J+1)-DEP(I,J+1))
      D(11)=0.5*(DEP(I+1,J+2)-DEP(I+1,J))
      D(12)=0.25*(DEP(I+2,J+2)-DEP(I+2,J)-DEP(I,J+2)+DEP(I,J))
      D(13)=DEP(I,J+1)
      D(14)=0.5*(DEP(I+1,J+1)-DEP(I-1,J+1))
      D(15)=0.5*(DEP(I,J+2)-DEP(I,J))
      D(16)=0.25*(DEP(I+1,J+2)-DEP(I+1,J)-DEP(I-1,J+2)+DEP(I-1,J))
*
 110  IF (NSURFCE-2) 120,130,140
*
*  FIND COEFFICIENTS OF QUADRATIC LEAST SQUARES POLYNOMIAL BY
*  MATRIX MULTIPLICATION.   [C] = [SX1][D]
*
 120  CALL MATRIX (6,12,SX1,D,C)
      GO TO 150
*
*  FIND COEFFICIENTS OF CUBIC LEAST SQUARES POLYNOMIAL BY
*  MATRIX MULTIPLICATION.   [C] = [SX2][D]
*
```

```
  130 CALL MATRIX (10,12,SX2,D,C)
      GO TO 150
*
*  FIND COEFFICIENTS OF CONSTRAINED BICUBIC INTERPOLATING POLYNOMIAL BY
*  MATRIX MULTIPLICATION.    [C] = [SX3][D]
*
  140 CALL MATRIX (16,16,SX3,D,C)
*
*  EVALUATE DEPTH BY USING POLYNOMIAL APPROXIMATION
*
  150 DXY=C(1)+(C(3)+(C(6)+C(10)*YP)*YP)*YP+(C(2)+(C(5)+(C(9)+C(11)*YP)*
     +YP)*YP)*XP+(C(4)+(C(8)+(C(12)+C(13)*YP)*YP)*YP)*XP**2+(C(7)+
     +(C(14)+(C(15)+C(16)*YP)*YP)*YP)*XP**3
*
*  COMPUTE PARTIAL DERIVATIVES OF LOCAL DEPTH WITH RESPECT TO SPATIAL
*  COORDINATES FROM POLYNOMIAL APPROXIMATION
*
      PDX=C(2)+(C(5)+(C(9)+C(11)*YP)*YP)*YP+2.*(C(4)+(C(8)+(C(12)+C(13)*
     +YP)*YP)*YP)*XP+3.*XP**2*(C(7)+(C(14)+(C(15)+C(16)*YP)*YP)*YP)
      PDY=C(3)+(C(5)+(C(8)+C(14)*XP)*XP)*XP+2.*YP*(C(6)+(C(9)+(C(12)+
     +C(15)*XP)*XP)*XP)+3.*YP**2*(C(10)+(C(11)+(C(13)+C(16)*XP)*XP)*XP)
      PD2X=2.*(C(4)+(C(8)+(C(12)+C(13)*YP)*YP)*YP)+6.*XP*(C(7)+(C(14)+
     +(C(15)+C(16)*YP)*YP)*YP)
      PD2Y=2.*(C(6)+(C(9)+(C(12)+C(15)*XP)*XP)*XP)+6.*YP*(C(10)+(C(11)+
     +(C(13)+C(16)*XP)*XP)*XP)
      PD2XY=C(5)+2.*(C(9)*YP+C(8)*XP)+3.*(C(11)*YP**2+C(14)*XP**2)+
     +4.*C(12)*XP*YP+6.*XP*YP*(C(13)*YP+C(15)*XP)+9.*C(16)*XP**2*YP**2
*
*  CHANGE DEPTH UNITS BY DCON
*
      DXY=DXY*DCON+TIDE
      PDX=PDX*DCON
      PDY=PDY*DCON
      PD2X=PD2X*DCON
      PD2Y=PD2Y*DCON
      PD2XY=PD2XY*DCON
*
      RETURN
      END
```

Subroutine HEIGHT

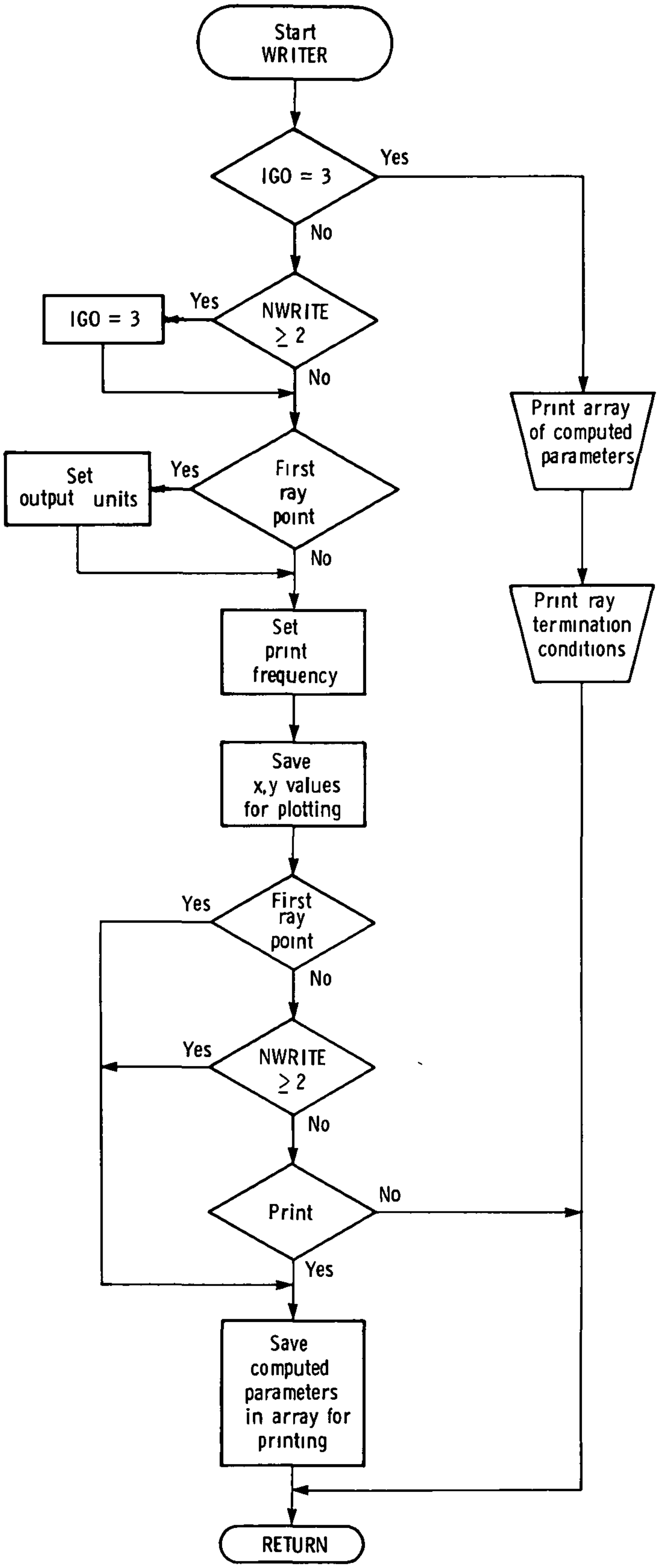
Subroutine HEIGHT calculates the local shoaling coefficient (eq. (7)) and then computes the refraction coefficient (eq. (8)) by using the value of β calculated at the previous ray point. These coefficients are then used to compute the local wave height. Fox's finite-difference solution for wave intensity is solved to give a new value of β. The flow diagram and listing of subroutine HEIGHT follow.


Start
HEIGHT
IGO < 2
Yes
1
No
Compute wavelength, L, and group velocity, CG
CG ≤ 0
Yes
1
No
Compute shoaling coef. K_s, and refraction coef, K_r
H = H_o K_s K_r
JGO = 2
No
Compute intermediate-water component of q
Yes
Compute shallow-water component of q
Compute coef. in equation for β· p,q
Compute ray separation factor, β
1
RETURN

```
      SUBROUTINE HEIGHT (A,H)
*
****************************************************************************
*                                                                          *
*     THIS SUBROUTINE CALCULATES THE SHOALING COEFFICIENT, SK, AND         *
*     COMPUTES THE REFRACTION COEFFICIENT, RK, USING THE VALUE OF          *
*     BETA AT THE PREVIOUS POINT ON THE RAY. THESE ARE THEN USED TO        *
*     CALCULATE THE LOCAL WAVE HEIGHT. THEN THE FINITE-DIFFERENCE          *
*     FORM OF THE EQUATION OF WAVE INTENSITY IS SOLVED TO GIVE A NEW       *
*     VALUE OF BETA.                                                       *
*                                                                          *
****************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      IF (IGO .LT. 2) GO TO 30
      WL = WLO*RCCO
      GN = 12.5663706*DXY/WL
      CG = (1.0+GN/SINH(GN))*CXY
      IF (CG .LT. 0.) GO TO 30
      SK = SQRT(CO/CG)
*
** COMPUTE REFRACTION COEF USING MUNK-ARTHUR EQN **
*
      RK = ABS(1./B2)
      RK = SQRT(RK)
*
** COMPUTE WAVE HEIGHT **
*
      H = HO*SK*RK
*
** TEST TO SEE IF IN SHALLOW OR INTERMEDIATE WATER **
*
      IF (JGO .EQ. 2) GO TO 10
*
** INTERMEDIATE WATER VALUE FOR U **
*
      U = -2.*SIG*RCCO*CXY/(V*V)
      GO TO 20
*
** SHALLOW WATER VALUE FOR U **
*
   10 U = -0.5/DXY
*
** COMPUTE RAY SEPARATION FACTOR **
*
   20 COSA = COS(A)
      SINA = SIN(A)
      P = -(COSA*PDX+SINA*PDY)*CPRIME*DTGR*2.
      Q = ((PD2X+U*PDX*PDX)*SINA*SINA-(PD2XY+U*PDX*PDY)*2.*SINA*COSA+
     +(PD2Y+U*PDY*PDY)*COSA*COSA)*CPRIME*CXY*DTGR*DTGR*2.
      B3 = ((P-2.)*B1+(4.-Q)*B2)/(P+2.)
      B1 = B2
      B2 = B3
   30 RETURN
      END
```

Subroutine WRITER

Subroutine WRITER is used for storing computed parameters in appropriate arrays and then for printing the same parameters after the complete ray path has been constructed. The output units for spatial coordinates (x,y) are in nautical miles. The output units for all other computed parameters can be in either U.S. Customary Units or the International System of Units (SI), as specified by the parameter IOUTUNT. The first and last points on a ray are always printed, and intermediate points are printed according to the print frequencies specified by the input array JPRTFRQ. Position data are saved in this routine to be used in the plotting subroutine PLOTR. The flag IGO is set to 3 when the ray has reached a termination condition, and a message specifying the particular termination condition is printed. The flow diagram and listing of subroutine WRITER follow.

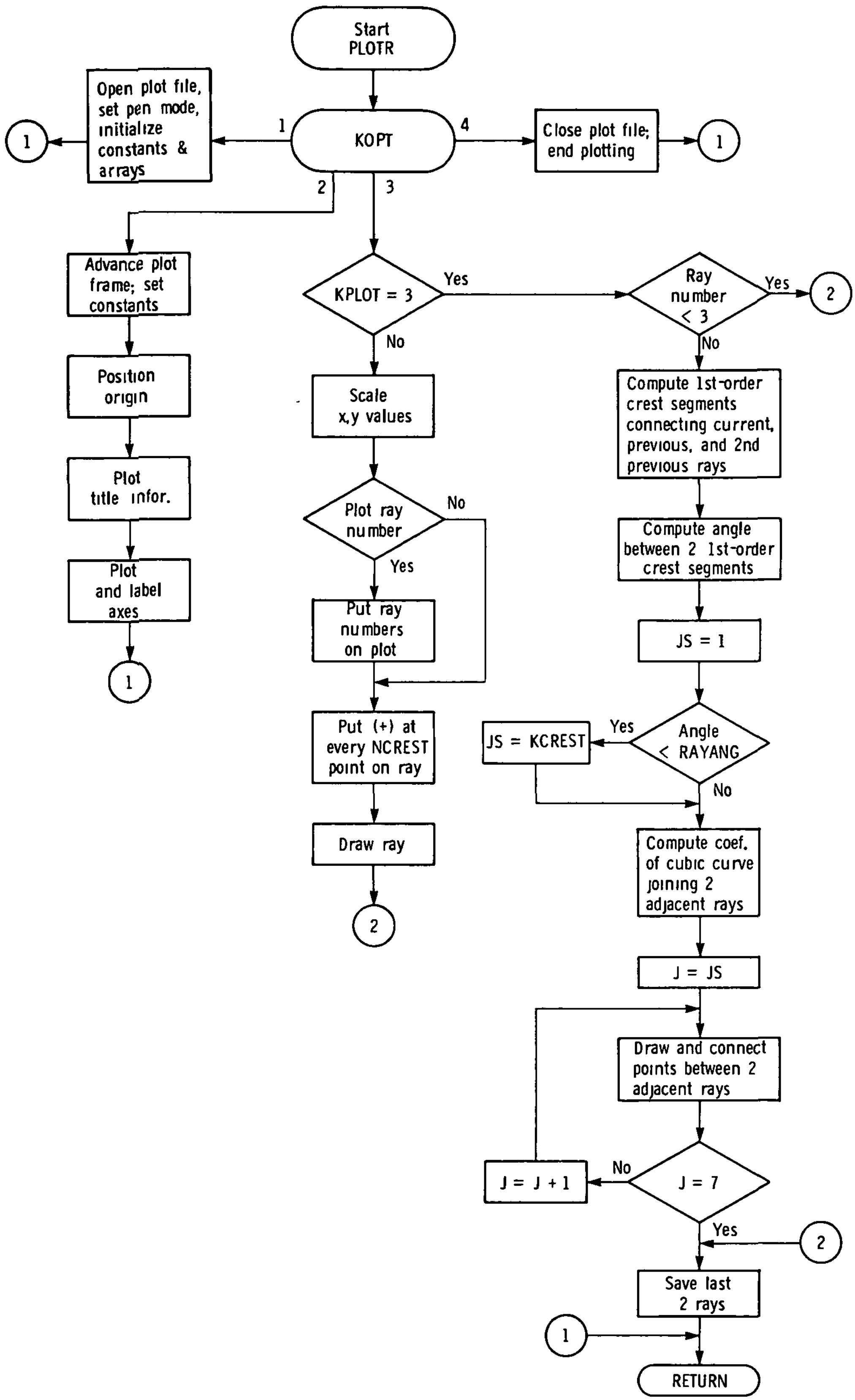

Start WRITER
IGO = 3
Yes
No
NWRITE ≥ 2
Yes
IGO = 3
No
First ray point
Yes
Set output units
No
Set print frequency
Save x,y values for plotting
First ray point
Yes
No
NWRITE ≥ 2
Yes
No
Print
No
Yes
Save computed parameters in array for printing
Print array of computed parameters
Print ray termination conditions
RETURN

```
      SUBROUTINE WRITER (X,Y,A,H)
*
*************************************************************************
*                                                                       *
*     THIS SUBROUTINE STORES AND PRINTS COMPUTED RAY PARAMETERS AND     *
*     THE RAY TERMINATION CONDITION. X AND Y COORDINATES ARE SAVED      *
*     IN THIS SUBROUTINE FOR EVENTUAL PLOTTING.                         *
*                                                                       *
*************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      DIMENSION NPT1(400),X1(400),Y1(400),A1(400),DXY1(400),WL1(400),
     +CXY1(400),RK1(400),SK1(400),H1(400)
*
      IF (IGO .EQ. 3) GO TO 30
      IF (NWRITE .GE. 2) IGO=3
*
** SET OUTPUT UNITS **
*
      IF (NPT .GT. 1) GO TO 10
      NC = 0
      AMULT = 1.
      XYMULT = 1./6076.1155
      IF (IOUTUNT .EQ. 2) AMULT = 0.3048
*
** SET PRINT FREQUENCY **
*
   10 IPRINT = JPRTFRQ(3)
      IF (DXY/WL .GT. 0.25 .AND. DXY/WL .LT. 0.5) IPRINT = JPRTFRQ(2)
      IF (DXY/WL .GE. 0.5) IPRINT = JPRTFRQ(1)
*
** SAVE X AND Y RAY COORDINATES FOR PLOTTING WHICH ARE INDEX **
** WITH RESPECT TO REFERENCE CORNER                          **
*
      XPLOT(NPT+NTICOFF) = X
      YPLOT(NPT+NTICOFF) = Y
*
** SAVE FIRST AND LAST POINT **
*
      IF (NPT .EQ. 1 .OR. NWRITE .GE. 2) GO TO 20
*
```

```
** PRINT OUTPUT CONTROL **
*
      IF (MOD (NPT,IPRINT) .NE. 0) GO TO 50
*
** SAVE COMPUTED RAY PARAMETERS **
*
   20 NC = NC +1
      NPT1(NC) = NPT
      X1(NC) = XYMULT*GRID*X
      Y1(NC) = XYMULT*GRID*Y
      A1(NC) = A
      DXY1(NC) = AMULT*DXY
      WL1(NC) = AMULT*WL
      CXY1(NC) = AMULT*CXY
      RK1(NC) = RK
      SK1(NC) = SK
      H1(NC) = AMULT*H
      GO TO 50
*
** PRINT STORED ARRAY OF COMPUTED RAY PARAMETERS **
*
   30 DO 40 I=1,NC
   40 PRINT 60 ,NPT1(I),X1(I),Y1(I),A1(I),DXY1(I),WL1(I),CXY1(I),
     +RK1(I),SK1(I),H1(I)
*
** PRINT RAY TERMINATION CONDITIONS **
*
      IF (NWRITE .EQ. 2) PRINT 70
      IF (NWRITE .EQ. 3) PRINT 80
      IF (NWRITE .EQ. 4) PRINT 90
      IF (NWRITE .EQ. 5) PRINT 100
      IF (NWRITE .EQ. 6) PRINT 110
   50 RETURN
*
   60 FORMAT (3X,I4,2F10.2,2X,2F9.2,2F8.2,F10.3,F11.3,F9.2)
   70 FORMAT (1HO,*  RAY STOPPED, NO CONVERGENCE FOR CURVATURE*)
   80 FORMAT (1HO,*  RAY STOPPED, REACHED SHORE*)
   90 FORMAT (1HO,*  RAY STOPPED, REACHED BOUNDARY*)
  100 FORMAT (1HO,*  RAY STOPPED, NUMBER OF POINTS EXCEEDS MAXIMUM*)
  110 FORMAT (1HO,*  RAY STOPPED, INCREMENTAL DISTANCE ALONG RAY LESS TH
     +AN MINIMUM*)
      END
```

Subroutine PLOTR

Subroutine PLOTR is used for segment-by-segment construction of wave crests and for plotting the ray diagrams. The logic for constructing the crest patterns is detailed in appendix B. The plotting mode is controlled by the input parameter KPLOT. Axes, labels, and notations are the same for both the ray and crest modes of plotting. The input parameters PLTLNGX and PLTLNGY control the size of the plotting area, and the input parameter PLTSCAL scales the data to fit this area. Actual plotting is performed by using system subroutines which are described in appendix A. The flow diagram and listing of subroutine PLOTR follow.


Start PLOTR
KOPT
1
2
3
4
Open plot file, set pen mode, initialize constants & arrays
Close plot file; end plotting
Advance plot frame; set constants
Position origin
Plot title infor.
Plot and label axes
KPLOT = 3
Yes
No
Scale x,y values
Plot ray number
Put ray numbers on plot
Put (+) at every NCREST point on ray
Draw ray
Ray number < 3
Compute 1st-order crest segments connecting current, previous, and 2nd previous rays
Compute angle between 2 1st-order crest segments
JS = 1
Angle < RAYANG
JS = KCREST
Compute coef. of cubic curve joining 2 adjacent rays
J = JS
Draw and connect points between 2 adjacent rays
J = 7
J = J + 1
Save last 2 rays
RETURN

```
      SUBROUTINE PLOTR (KOPT)
*
**************************************************************************
*                                                                        *
*     THIS SUBROUTINE PLOTS RAY AND CREST DATA.                          *
*                                                                        *
**************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      DIMENSION XPL(400),YPL(400)
*
      GO TO (10,30,100,230), KOPT
*
** OPEN PLOT FILE AND SET PLOT SPEED **
*
   10 CALL PSEUDO
      CALL LEROY
      EPS = 0.001
      DO 20 I=1,500
   20 XPLOT(I)=YPLOT(I)=XPLOT1(I)=YPLOT1(I)=XPLOT2(I)=YPLOT2(I)=0.0
      GO TO 240
*
** ADVANCE PLOT FRAME AND SET CONSTANTS **
*
   30 CALL NFRAME
      AM=GRID/6076.1155
      COSRAY=COS(RAYANG/57.2957795)
*
** SET ORIGIN **
*
      CALL CALPLT (3.,1.5,-3)
*
** PLOT TITLE INFORMATION **
*
      CALL NOTATE (-1.5,1.0,0.15,TITL,90.,60)
      IF (NSURFCE-2) 40,50,60
   40 CALL NOTATE (-1.2,3.0,0.125,23HQUADRATIC LEAST SQUARES,90.,23)
      GO TO 70
   50 CALL NOTATE (-1.2,3.0,0.125,19HCUBIC LEAST SQUARES,90.,19)
      GO TO 70
   60 CALL NOTATE (-1.2,3.0,0.125,33HCONSTRAINED BICUBIC INTERPOLATION,
```

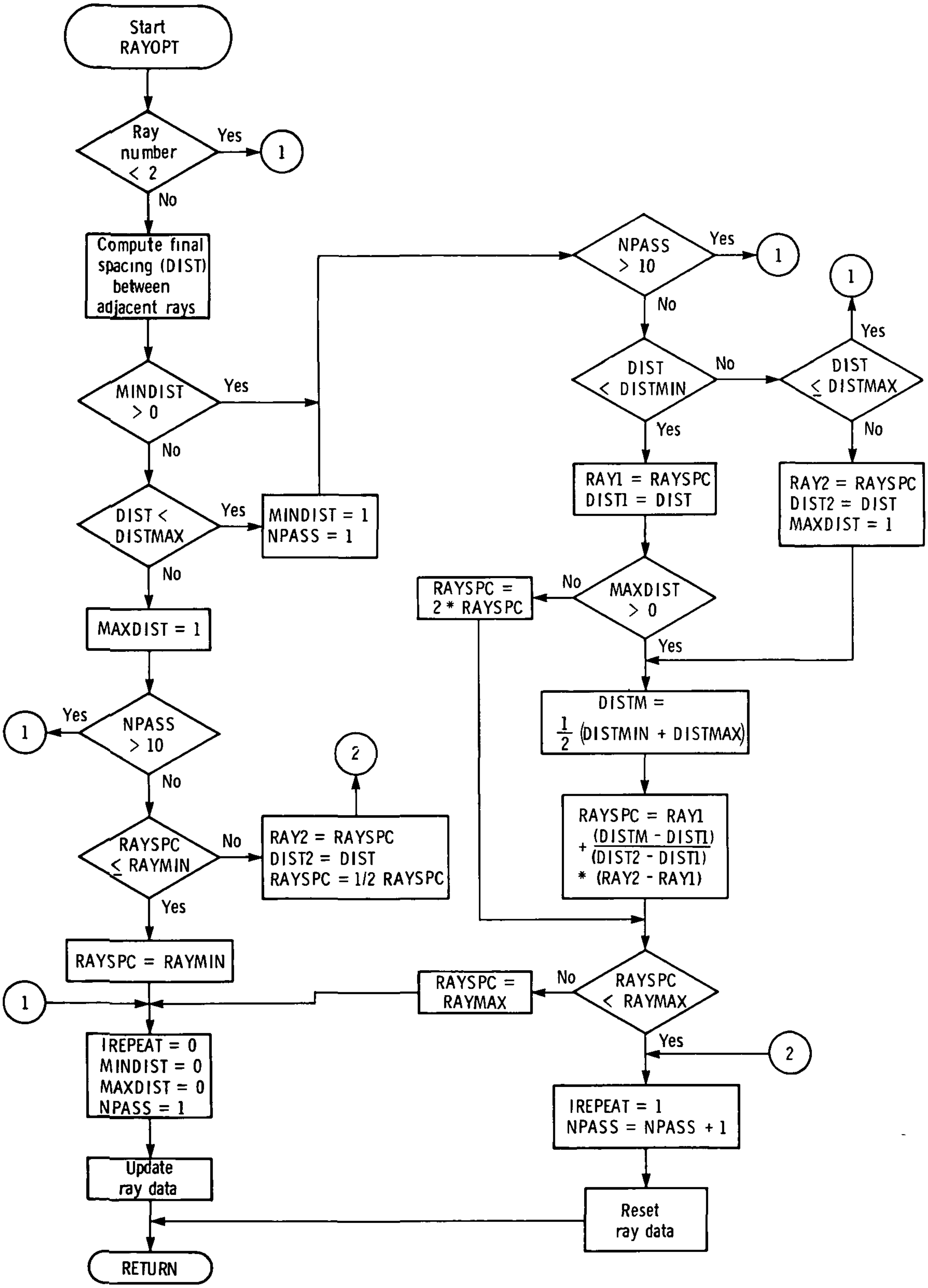
```
      +90.,33)
*
** PLOT AND LABEL AXES **
*
   70 Y1=-0.069
      TMAJ=10./PLTSCAL
      LMT=PLTLNGY*PLTSCAL+10.
      DO 80 I=10,LMT,10
      FPN=I-10
      CALL NUMBER (-0.15,Y1,0.12,FPN,90.,-1)
   80 Y1=Y1+TMAJ
      X1=0.06
      LMT=PLTLNGX*PLTSCAL+10.
      DO 90 I=10,LMT,10
      FPN=I-10
      CALL NUMBER (X1,-0.4,0.12,FPN,90.,-1)
   90 X1=X1+TMAJ
      YM=PLTLNGY/2.0-0.26
      CALL NOTATE (-0.6,YM,0.15,4HY,NM,90.,4)
      XM=PLTLNGX/2.0+0.08
      CALL NOTATE (XM,-1.20,0.15,4HX,NM,90.,4)
      CALL AXES (0.,0.,0.,PLTLNGX,0.,PLTSCAL,TMAJ,PLTSCAL,1H ,0.,1)
      CALL AXES (0.,0.,90.,PLTLNGY,0.,PLTSCAL,TMAJ,PLTSCAL,1H ,0.,-1)
      CALL CALPLT (0.,PLTLNGY,3)
      CALL CALPLT (PLTLNGX,PLTLNGY,2)
      CALL CALPLT (PLTLNGX,0.,2)
      GO TO 240
  100 IF (KPLOT .EQ. 3) GO TO 140
*
** PLOT RAY **
*
      NS=0
      DO 110 I=NFRST,NLAST
      NS=NS+1
      XPL(NS)=AM*XPLOT(I)/PLTSCAL
  110 YPL(NS)=AM*YPLOT(I)/PLTSCAL
      IF (MOD (NORAY,5) .NE. 0) GO TO 120
*
** PUT RAY NUMBER ON PLOT **
*
      XNO = XPL(1)+0.03
      YNO = YPL(1)+0.03
      CALL NUMBER (XNO,YNO,0.07,FLOAT(NORAY),90.,-1)
*
** PUT + SYMBOL AT EVERY NCREST POINT ON RAY **
*
  120 DO 130 I=1,NS,NCREST
  130 CALL PNTPLT (XPL(I),YPL(I),-22,1)
      CALL DRAW (XPL,YPL,NS)
```

```
      GO TO 190
  140 IF (NORAY .LT. 3) GO TO 190
*
** DETERMINE FIRST AND LAST RAY POINT WITH RESPECT TO REFERENCE **
** CORNER FOR PLOTTING CREST DATA                              **
*
      MS1 = NFRST
      IF (NFRST1 .GT. NFRST) MS1=NFRST1
      IF (NFRST2 .GT. NFRST .AND. NFRST2 .GT. NFRST1) MS1=NFRST2
      MS2 = NLAST
      IF (NLAST1 .LT. NLAST) MS2=NLAST1
      MS1=MS1+1
      MS2=MS2-1
*
** PLOT CREST DATA **
*
      DO 180 I=MS1,MS2,NCREST
*
** CALCULATE VECTOR COMPONENTS OF RAY SLOPES AND FIRST-ORDER **
** CREST SEGMENTS                                            **
*
      DXS1=XPLOT1(I+1)-XPLOT1(I-1)
      DYS1=YPLOT1(I+1)-YPLOT1(I-1)
      IF (ABS(DYS1) .LT. EPS) DYS1=SIGN(EPS,DYS1)
*
*
      SL1=-DXS1/DYS1
      DXPT=XPLOT(I+1)-XPLOT(I-1)
      DYPT=YPLOT(I+1)-YPLOT(I-1)
      IF (ABS(DYPT) .LT. EPS) DYPT=SIGN(EPS,DYPT)
      SL2=-DXPT/DYPT
      DXPS=XPLOT(I)-XPLOT1(I)
      IF (ABS(DXPS) .LT. EPS) DXPS=SIGN(EPS,DXPS)
      DYPS=YPLOT(I)-YPLOT1(I)
      D1=SQRT(DXPS**2+DYPS**2)
      IF (D1 .LT. EPS) D1=EPS
      IF (I .GT. NLAST2) GO TO 160
      DXS12=XPLOT2(I)-XPLOT1(I)
      DYS12=YPLOT2(I)-YPLOT1(I)
      D2=SQRT(DXS12**2+DYS12**2)
      IF (D2 .LT. EPS) D2=EPS
*
** CALCULATE ANGLE BETWEEN ADJOINING FIRST-ORDER CREST SEGMENTS **
*
      THETA=(DXPS*DXS12+DYPS*DYS12)/(D1*D2)
      JS=1
      IF (THETA .GT. COSRAY) JS=KCREST
*
** CALCULATE COEFFICIENTS OF CUBIC CURVE **
```

```
*
  160 A1=SL1
      A2=(3.0*DYPS-DXPS*(2.0*SL1+SL2))/DXPS**2
      A3=(-2.0*DYPS+DXPS*(SL1+SL2))/DXPS**3
      DX=DXPS/6.0
      NS=0
*
** CALCULATE AND DRAW POINTS ON CUBIC CURVE BETWEEN TWO RAYS **
*
      IF (JS .GE. 7) GO TO 180
      DO 170 J=JS,7
      NS=NS+1
      XPL(NS)=(XPLOT1(I)+(J-1)*DX)*AM/PLTSCAL
      D1=(J-1)*DX
      YPL(NS)=(YPLOT1(I)+D1*(A1+D1*(A2+D1*A3)))*AM/PLTSCAL
      IF (YPL(NS) .LT. 0.0) YPL(NS)=0.0
  170 CONTINUE
      CALL DRAW (XPL,YPL,NS)
  180 CONTINUE
  190 IF (NORAY .LT. 2) GO TO 210
*
** SAVE PREVIOUS TWO RAYS **
*
      DO 200 I=NFRST1,NLAST1
      XPLOT2(I) = XPLOT1(I)
  200 YPLOT2(I) = YPLOT1(I)
  210 DO 220 I=NFRST,NLAST
      XPLOT1(I) = XPLOT(I)
  220 YPLOT1(I) = YPLOT(I)
      NFRST2 = NFRST1
      NLAST2 = NLAST1
      NFRST1 = NFRST
      NLAST1 = NLAST
      GO TO 240
*
** END PLOTTING **
*
  230 CALL CALPLT(0.,0.,999)
  240 RETURN
      END
```

Subroutine RAYOPT

This subroutine determines the initial ray spacing required to satisfy specified final (nearshore) ray spacing conditions. Final ray spacing is defined as the distance between the shorewardmost synchronous points on adjacent rays. The routine would normally be used in conjunction with the wave-crest plotting option in order to facilitate the construction of curved crest segments between adjacent rays. A detailed discussion of this routine and pertinent input parameters is given in appendix C. The flow diagram and listing of subroutine RAYOPT follow.

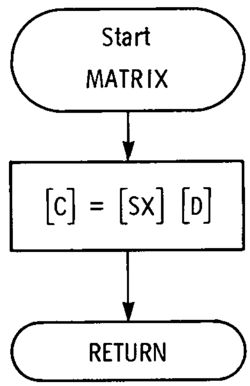

Start
RAYOPT
Ray number < 2
Yes
No
1
Compute final spacing (DIST) between adjacent rays
MINDIST > 0
Yes
No
DIST < DISTMAX
Yes
No
MINDIST = 1
NPASS = 1
MAXDIST = 1
NPASS > 10
Yes
No
RAYSPC ≤ RAYMIN
No
Yes
RAY2 = RAYSPC
DIST2 = DIST
RAYSPC = 1/2 RAYSPC
2
RAYSPC = RAYMIN
IREPEAT = 0
MINDIST = 0
MAXDIST = 0
NPASS = 1
Update ray data
RETURN
NPASS > 10
DIST < DISTMIN
DIST ≤ DISTMAX
RAY1 = RAYSPC
DIST1 = DIST
RAY2 = RAYSPC
DIST2 = DIST
MAXDIST = 1
MAXDIST > 0
RAYSPC = 2 * RAYSPC
DISTM = 1/2 (DISTMIN + DISTMAX)
RAYSPC = RAY1 + (DISTM - DIST1)/(DIST2 - DIST1) * (RAY2 - RAY1)
RAYSPC < RAYMAX
RAYSPC = RAYMAX
IREPEAT = 1
NPASS = NPASS + 1
Reset ray data

```
      SUBROUTINE RAYOPT (IREPEAT)
*
************************************************************************
*                                                                      *
*     THIS SUBROUTINE ITERATES FOR THE INITIAL RAY SPACING SO THAT     *
*     FINAL RAY SEPARATION CONDITIONS ARE SATISFIED .                  *
*                                                                      *
************************************************************************
*
      COMMON B1,B2,CO,CXY,CPRIME,DCON,DELTAS,DEP(42,32),DISTMAX,DISTMIN,
     +DRC,DTGR,DXY,GRID,GRINC,HO,IBLK(101),IGO,IOUTUNT,IREF,ISTOP,JGO,
     +JPRTFRQ(3),KCREST,KPLOT,LIMNPT,MI,MJ,NCREST,NFRST,NFRST1,NLAST,
     +NLAST1,NORAY,NOROW,NPT,NSURFCE,NTIC,NTICOFF,NWRITE,PDX,PDY,PD2X,
     +PD2Y,PD2XY,PLTLNGX,PLTLNGY,PLTSCAL,RAYANG,RAYMAX,RAYMIN,RAYSPC,
     +RCCO,RK,SK,SIG,TIDE,TITL(8),V,WL,WLO,XO,XM,XPLOT(500),XPLOT1(500),
     +XPLOT2(500),XTEST,YO,YM,YPLOT(500),YPLOT1(500),YPLOT2(500),YTEST
*
      IF (NORAY .LT. 2) GO TO 80
      NTEST = NLAST
      IF (NLAST .GT. NLAST1) NTEST=NLAST1
*
** COMPUTE FINAL SPACING BETWEEN ADJACENT RAYS **
*
      DIST = SQRT((XPLOT(NTEST)-XPLOT1(NTEST))**2+(YPLOT(NTEST)-
     +YPLOT1(NTEST))**2)
      IF (MINDIST .GT. 0) GO TO 30
      IF (DIST .LT. DISTMAX) GO TO 20
*
** IF DISTANCE BETWEEN RAYS IS TOO LARGE, REDUCE INITIAL RAY  **
** SEPARATION AND REPEAT. MAX NO OF PASSES EQUAL 10           **
*
      MAXDIST = 1
      IF (NPASS .GT. 10) GO TO 80
*
** IF INITIAL RAY SPACING LESS THAN MINIMUM, SET TO MINIMUM **
*
      IF (RAYSPC .LE. RAYMIN) GO TO 10
      RAY2 = RAYSPC
      DIST2 = DIST
      RAYSPC = 0.5*RAYSPC
      GO TO 70
   10 RAYSPC = RAYMIN
      GO TO 80
   20 MINDIST = 1
      NPASS=1
   30 IF (NPASS .GT. 10) GO TO 80
      IF (DIST .LT. DISTMIN) GO TO 40
      IF (DIST .LE. DISTMAX) GO TO 80
```

```
      RAY2 = RAYSPC
      DIST2 = DIST
      MAXDIST = 1
      GO TO 50
*
** IF FINAL DISTANCE BETWEEN RAYS IS TOO SMALL, FIND INITIAL RAY  **
** SEPARATION BY NEWTON-RAPHSON METHOD AND REPEAT. MAX NO         **
** OF PASSES EQUAL 10                                              **
*
   40 RAY1 = RAYSPC
      DIST1 = DIST
      IF (MAXDIST .GT. 0) GO TO 50
      RAYSPC = 2.0*RAYSPC
      GO TO 60
   50 DISTM = 0.5*(DISTMIN+DISTMAX)
      RAYSPC = RAY1+(DISTM-DIST1)*(RAY2-RAY1)/(DIST2-DIST1)
*
** IF INITIAL RAY SPACING GREATER THAN MAXIMUM, SET TO MAXIMUM **
*
   60 IF (RAYSPC .LT. RAYMAX) GO TO 70
      RAYSPC = RAYMAX
      GO TO 80
   70 XTEST = XTESTL
      YTEST = YTESTL
      NTIC = NTICL
      NPASS = NPASS+1
      IREPEAT = 1
      GO TO 90
*
** GOOD PASS, RESET AND UPDATE INITIAL CONDITIONS **
*
   80 MAXDIST = 0
      IREPEAT = 0
      MINDIST = 0
      XTESTL = XTEST
      YTESTL = YTEST
      NTICL = NTIC
      NPASS = 1
   90 RETURN
      END
```

Subroutine MATRIX

This is an auxiliary routine incorporated to perform the matrix multiplication required in the determination of the coefficients for the topography approximating polynomial used in subroutine DEPTH. The subroutine multiplies the invariant matrix SX which depends only on the topography approximating technique selected, by a column vector of depth values D to obtain the resultant column vector of coefficients C to be used with the particular polynomial. The flow diagram and listing of subroutine MATRIX follow.




```
      SUBROUTINE MATRIX (IR,JC,SX,D,C)
*
*************************************************************************
*                                                                       *
*     THIS SUBROUTINE MULTIPLIES A MATRIX TIMES A VECTOR.               *
*                                                                       *
* **  INPUT **                                                          *
*                                                                       *
*     IR        = NO OF ROWS IN MATRIX A                                *
*     JC        = NO OF COLUMNS IN MATRIX A                             *
*     SX(IR,JC) = INPUT MATRIX                                          *
*     D(JC)     = INPUT VECTOR                                          *
*     C(IR)     = OUTPUT VECTOR                                         *
*                                                                       *
*************************************************************************
*
      DIMENSION SX(IR,JC),D(16),C(16)
*
      DO 10 K=1,16
   10 C(K) = 0.
      DO 20 I=1,IR
      DO 20 J=1,JC
   20 C(I) = C(I)+SX(I,J)*D(J)
      RETURN
      END
```

PROGRAM USAGE

The computer program is coded in FORTRAN Extended (FTN) Version 4 for use on the Control Data Cyber 173 or 175 computers under NOS 1.2. A central memory field length of 66000_8 locations is required, along with a peripheral mass-storage file for input of bathymetry data as discussed subsequently. Central processing unit (CPU) time requirements for program execution vary with the size of the geographic region being considered, the spatial density of wave rays being computed within the region, and the graphic output mode selected.

Bathymetry Data Input

As a first step, a bathymetry data array corresponding to a square grid with a uniform spacing of GRID feet per division must be developed for the geographic region to be investigated. It is recommended that the data array be developed by using a special transverse Mercator map projection centered within the region of interest in order to minimize error in lengthy ray paths due to curvature of the Earth. Advantages of using such a projection are discussed in reference 6, and the projection itself is discussed in detail in reference 9. For input to the computer program, the regional bathymetry array must be divided into overlapped modules of dimensions 42 by 32. Modularization must be performed as a separate function and can be done using the utility program CREMOD, which is described in detail in appendix D. During program execution, the modularized bathymetry data file TAPE1 is initialized as a random-access file in main program WAVE by using the system subroutine OPENMS. Modules are then read from the file to central memory as required in subroutine DEPTH by using the system subroutine READMS.

Card Input

Other parameters required for program execution are input in card form by using standard Control Data NAMELIST. The program is designed to compute a number of sets (NOSETS) of refraction diagrams with a single computer run. Parameters which are fixed for all sets of diagrams to be computed are input under the NAMELIST group name NPUT1 and are listed with their default values in table I. If the user desires to use default values, only the parameters specifying the modular bathymetry grid configuration must be supplied. Parameters in effect for a single set are input under the NAMELIST group name NPUT2 and are listed with their default values in table II. If the user desires to use default values with this group, only the wave period and initial wave height and direction must be specified. In addition, a title card of up to 80 alphanumeric characters is required before each group of NPUT2 cards as identification of the conditions in effect for that set. Normal termination of program execution and closure of the plot file is effected after NOSETS of ray diagrams have been computed. However, in the event of user oversight, an end of file (EOF) is checked on card input on both NPUT1 and NPUT2 to insure proper termination.

Output Description

At the beginning of the printed output, the basic program input parameters are identified and the values assigned to them are printed. Omission by the user of any mandatory input parameter causes the program to print an appropriate error message and then terminate. If KPLOT is specified as 2 or 3, the additional input parameters required for those plotting options are identified and their values printed. At the start of printed values for each ray, a header is printed which gives the values of constants valid for the current ray set along with title information for the current ray. Computed ray parameters are printed at specified increments in the ray computational point index NPT; these increments are input in the array JPRTFRQ. Output units for spatial coordinates (x,y) are in nautical miles. Output units for other computed parameters can be either U.S. Customary Units or SI Units, as specified by the input parameter IOUTUNT. The computed parameters are printed in column format in the following order:

NPT	ray computational point index
X	alongshore coordinate x, nautical miles
Y	offshore coordinate y, nautical miles
A	ray direction α, degrees
DXY	water depth d, feet or meters
WL	wavelength L, feet or meters
C	wave speed c, feet/second or meters/second
RK	refraction coefficient K_r
SK	shoaling coefficient K_s
H	wave height H, feet or meters

At the end of printed parameters for each ray, the condition which led to ray termination is printed.

The plotted output, depending upon the value of KPLOT, gives either ray diagrams or crest diagrams. The length of each axis is determined by the input parameters PLTLNGX and PLTLNGY. Scaling of the data and the axis scales are controlled by the parameter PLTSCAL. Major tick marks are added at 10-nautical-mile intervals, and minor tick marks are added at 1-nautical-mile intervals along both axes. In plotting ray diagrams, a plus symbol (+) is drawn at increments of NCREST computational points along each ray, and every fifth ray is numbered. Initial ray spacing is specified by RAYSPC in the default mode or is computed to meet constraints on final (nearshore) ray spacing if that option is selected.

Sample Cases

In order to illustrate the input and output options of the program, three sample cases are shown. The cases represent ray and crest diagrams for the Baltimore Canyon region of the mid-Atlantic continental shelf, extending from near Wachapreague Inlet, Virginia, to near Manasquan Inlet, New Jersey. The bathymetry data array for this region was divided into a total of 72 modules of dimensions 42 by 32, with 8 rows and 9 columns of data modules. For each sample case, the basic program parameters, the input parameters for the ray set, and the computed parameters for a representative ray within the set are shown. Also the ray or crest diagram constructed for the set in question is presented. It should be noted that the land masses and isobaths have been added to the basic diagrams for display purposes.

Case 1.- For this case, a ray diagram was generated with equal initial spacing between the computed rays. KPLOT was specified as 1, and IOUTUNT was specified as 1, resulting in output parameters other than x and y being given in U.S. Customary Units. This case required 10 seconds for computer processing. Figure 3 gives the ray diagram for sample case 1. A listing of the input parameters for sample case 1 is given as follows:

NPUT1

DCON	1.0
DELTAS	0.01
GRID	3038.06
GRINC	1.0
IOUTUNT	1
JPRTFRQ(1)	20
JPRTFRQ(2)	10
JPRTFRQ(3)	5
KPLOT	1
LIMNPT	400
MI	300
MJ	254
NOMOD	72
NOROW	8
NOSETS	1
PLTLNGX	12.0
PLTLNGY	10.0
PLTSCAL	12.5
TIDE	0.0

NPUT2

A	-45.0
HO	6.0
NCREST	5
NSURFCE	1
RAYSPC	5.0
T	12.0

The output for sample case 1 follows:

```
                    BASIC PROGRAM PARAMETERS

DEPTH UNIT MULT...............DCON          =    1.00
MIN STEP SIZE.................DELTAS        =    .01
GRID SIZE.....................GRID          = 3038.06
DEEP WATER STEP SIZE..........GRINC         =  1.00
OUTPUT UNIT OPTION............IOUTUNT       = 1
PRINT FREQ 1 (D/L>0.5)........JPRTFRQ(1)    = 20
PRINT FREQ 2 (0.25<D/L<0.5)...JPRTFRQ(2)    = 10
PRINT FREQ 3 (D/L<0.25).......JPRTFRQ(3)    =  5
PLOT OPTION...................KPLOT         = 1
MAX NO OF RAY POINTS..........LIMNPT        = 400
GRID LIMITS     ABSCISSA......MI            = 300
                ORDINATE......MJ            = 254
NO OF DEPTH MODULES...........NOMOD         = 72
NO OF ROW MODULES IN NOMOD....NOROW         =  8
NO OF SETS OF RAYS............NOSETS        =  1
PLOT X AXES LENGTH............PLTLNGX       = 12.00
PLOT Y AXES LENGTH............PLTLNGY       = 10.00
PLOT SCALE FACTOR.............PLTSCAL       = 12.50
TIDE HEIGHT...................TIDE          =  0.00
```

SAMPLE CASE NO 1
PERIOD = 12.00 SEC...TIME STEP = 49.44 SEC...RAY SPACING = 5.0...CREST SPACING = 5...BOTTOM APPROX CODE = 1
RAY NO = 2...SET NO = 1

POINT	X (N.MI.)	Y (N.MI.)	ANGLE (DEG)	DEPTH (FT)	LENGTH (FT)	SPEED (FT/SEC)	REFRACTION COEF	SHOALING COEF	HEIGHT (FT)
1	5.54	125.00	-45.00	204.49	737.34	61.44	1.000	1.000	6.00
10	8.72	121.82	-45.00	201.70	737.34	61.44	1.000	1.000	6.00
20	12.12	118.44	-44.49	180.24	685.17	57.10	1.004	.931	5.61
30	15.42	115.12	-46.09	222.18	709.13	59.09	1.043	.949	5.94
40	18.77	111.65	-46.00	229.10	711.93	59.33	1.149	.952	6.56
50	22.11	108.19	-46.12	208.08	702.52	58.54	1.255	.943	7.10
60	25.39	104.76	-46.01	204.93	700.86	58.40	1.379	.942	7.79
70	28.72	101.33	-46.24	203.58	700.13	58.34	1.596	.941	9.01
80	31.72	97.81	-53.66	141.08	647.66	53.97	1.671	.916	9.18
85	32.94	96.03	-56.91	127.02	629.21	52.43	1.846	.914	10.12
90	34.12	94.24	-56.49	135.04	640.12	53.34	2.291	.915	12.58
95	35.26	92.36	-61.89	145.45	652.79	54.40	4.470	.917	24.61
100	36.15	90.38	-69.64	134.19	639.01	53.25	3.331	.915	18.28
105	36.72	88.32	-79.08	129.46	632.64	52.72	2.198	.914	12.05
110	37.01	86.20	-84.89	131.06	634.83	52.90	1.833	.914	10.06
115	37.12	84.08	-89.01	118.34	616.19	51.35	1.650	.913	9.04
120	37.07	81.98	-93.58	127.36	629.70	52.48	1.468	.914	8.05
125	36.86	79.87	-97.55	129.25	632.36	52.70	1.221	.914	6.69
130	36.53	77.74	-99.95	121.98	621.81	51.82	1.078	.913	5.90
135	36.16	75.67	-100.79	114.84	610.53	50.88	1.028	.913	5.63
140	35.73	73.63	-102.58	124.07	624.94	52.08	.959	.913	5.26
145	35.25	71.58	-103.86	111.18	604.38	50.36	.871	.913	4.77
150	34.76	69.61	-103.94	103.06	589.72	49.14	.809	.915	4.44
155	34.28	67.66	-104.04	99.63	583.08	48.59	.769	.916	4.22
160	33.81	65.73	-102.84	105.75	594.74	49.56	.738	.914	4.05
165	33.34	63.76	-104.19	103.18	589.96	49.16	.662	.915	3.63
170	32.84	61.87	-105.45	89.48	561.80	46.82	.587	.920	3.24
175	32.34	60.06	-105.00	82.16	544.70	45.39	.525	.925	2.91
180	31.88	58.30	-104.91	77.42	532.79	44.40	.477	.929	2.66
185	31.41	56.57	-104.74	74.71	525.64	43.80	.448	.932	2.50
190	30.98	54.82	-103.18	75.01	526.45	43.87	.433	.931	2.42
195	30.58	53.10	-103.50	64.51	496.38	41.37	.416	.945	2.36
200	30.18	51.52	-104.75	54.94	464.96	38.75	.397	.963	2.29
205	29.80	49.99	-103.43	55.96	466.52	39.04	.391	.961	2.25
210	29.43	48.44	-103.32	57.44	473.60	39.47	.391	.958	2.25
215	29.05	46.95	-106.16	37.03	392.32	32.69	.388	1.022	2.38
220	28.68	45.88	-118.30	8.32	194.01	16.17	.390	1.395	3.27
222	28.56	45.70	-129.72	-.23	120.26	10.02	.384	1.751	4.03

RAY STOPPED, REACHED SHORE

Case 2.- This case illustrates a ray diagram in which the final (nearshore) ray separation distance is controlled. KPLOT was set equal to 2 and the limiting conditions on initial and final ray separation distances are printed at the end of the program parameter list. The output units are in SI Units as specified by setting IOUTUNT equal to 2. This case required 22 seconds for computer processing. Figure 4 shows the ray diagram for this sample case. The listing of the input parameters for sample case 2 is given as follows:

NPUT1		NPUT2	
DCON	1.0	A	-45.0
DELTAS	0.01	HO	6.0
GRID	3038.06	NCREST	5
GRINC	1.0	NSURFCE	1
IOUTUNT	2	RAYSPC	1.0
JPRTFRQ(1)	20	T	12.0
JPRTFRQ(2)	10	DISTMAX	50.0
JPRTFRQ(3)	5	DISTMIN	30.0
KPLOT	2	KCREST	1
LIMNPT	400	RAYANG	120.0
MI	300	RAYMAX	15.0
MJ	254	RAYMIN	0.01
NOMOD	72		
NOROW	8		
NOSETS	1		
PLTLNGX	12.0		
PLTLNGY	10.0		
PLTSCAL	12.5		
TIDE	0.0		

The output for sample case 2 follows:

```
                 BASIC PROGRAM PARAMETERS

DEPTH UNIT MULT................DCON          =    1.00
MIN STEP SIZE..................DELTAS        =     .01
GRID SIZE......................GRID          = 3038.06
DEEP WATER STEP SIZE...........GRINC         =  1.00
OUTPUT UNIT OPTION.............IOUTUNT       = 2
PRINT FREQ 1 (D/L>0.5).........JPRTFRQ(1)    = 20
PRINT FREQ 2 (0.25<D/L<0.5)....JPRTFRQ(2)    = 10
PRINT FREQ 3 (D/L<0.25)........JPRTFRQ(3)    =  5
PLOT OPTION....................KPLOT         = 2
MAX NO OF RAY POINTS...........LIMNPT        = 400
GRID LIMITS    ABSCISSA........MI            = 300
               ORDINATE........MJ            = 254
NO OF DEPTH MODULES............NDMOD         = 72
NO OF ROW MODULES IN NDMOD.....NDROW         =  8
NO OF SETS OF RAYS.............NOSETS        =  1
PLOT X AXES LENGTH.............PLTLNGX       = 12.00
PLOT Y AXES LENGTH.............PLTLNGY       = 10.00
PLOT SCALE FACTOR..............PLTSCAL       = 12.50
TIDE HEIGHT....................TIDE          =  0.00

MAX FINAL RAY SPACING..........DISTMAX       =  50.0
MIN FINAL RAY SPACING..........DISTMIN       =  30.0
INITIAL PT FOR CREST CURVE.....KCREST        =   1
MIN ANGLE BETWEEN SEGMENTS.....RAYANG        = 120.0
MAX INITIAL RAY SPACING........RAYMAX        =  15.0
MIN INITIAL RAY SPACING........RAYMIN        =    .01
```

SAMPLE CASE NO 2
PERIOD = 12.00 SEC...TIME STEP = 49.44 SEC...RAY SPACING = 8.0...CREST SPACING = 5...BOTTOM APPROX CODE = 1
RAY NO = 4...SET NO = 1

POINT	X (N.MI.)	Y (N.MI.)	ANGLE (DEG)	DEPTH (M)	LENGTH (M)	SPEED (M/SEC)	REFRACTION COEF	SHOALING COEF	HEIGHT (M)
1	7.66	123.59	-45.00	61.37	224.74	18.73	1.000	1.000	1.83
10	10.81	120.43	-44.88	55.50	209.25	17.44	1.000	.931	1.70
20	14.12	117.16	-45.18	57.69	210.77	17.56	.998	.935	1.71
30	17.46	113.74	-46.14	69.35	216.81	18.07	.939	.952	1.63
40	20.82	110.26	-45.82	66.65	215.68	17.97	.885	.948	1.53
50	24.14	106.84	-45.59	62.07	213.41	17.78	.807	.941	1.39
60	27.49	103.46	-45.05	65.71	215.25	17.94	.717	.946	1.24
70	30.84	100.05	-46.69	59.98	212.21	17.68	.627	.938	1.08
75	32.36	98.30	-51.85	49.47	204.29	17.02	.577	.923	.97
80	33.64	96.46	-58.86	41.56	195.63	16.30	.540	.915	.90
85	34.61	94.50	-68.45	44.80	199.50	16.62	.575	.918	.97
90	35.30	92.38	-75.43	44.74	199.44	16.62	.725	.918	1.22
95	35.72	90.26	-81.54	37.64	190.22	15.85	1.151	.913	1.92
100	35.98	88.20	-84.31	36.29	188.16	15.68	1.427	.913	2.38
105	36.13	86.12	-87.41	36.35	188.25	15.69	.764	.913	1.28
110	36.18	84.06	-89.52	32.49	181.74	15.14	.603	.914	1.01
115	36.17	82.04	-90.89	34.19	184.73	15.39	.570	.913	.95
120	36.13	79.96	-91.60	37.70	190.31	15.86	.614	.913	1.03
125	36.02	77.84	-93.85	37.10	189.41	15.78	.697	.913	1.16
130	35.88	75.74	-94.65	33.98	184.37	15.36	.797	.913	1.33
135	35.67	73.67	-96.58	37.75	190.38	15.86	.909	.913	1.52
140	35.41	71.58	-98.08	34.15	184.66	15.39	1.025	.913	1.71
145	35.11	69.57	-97.98	31.64	180.17	15.01	1.177	.914	1.97
150	34.84	67.58	-97.98	30.75	178.48	14.87	1.381	.915	2.31
155	34.57	65.62	-97.56	30.89	178.75	14.90	1.775	.915	2.97
160	34.34	63.63	-95.93	32.51	181.79	15.15	3.442	.914	5.75
165	34.12	61.66	-97.42	27.90	172.61	14.38	2.712	.919	4.56
170	33.86	59.81	-98.08	24.80	165.43	13.79	1.727	.925	2.92
175	33.60	58.00	-98.20	23.53	162.20	13.52	1.364	.929	2.32
180	33.34	56.20	-98.51	24.11	163.69	13.64	1.145	.927	1.94
185	33.07	54.44	-97.97	22.74	160.14	13.34	1.002	.932	1.71
190	32.85	52.71	-96.89	21.39	156.40	13.03	.912	.937	1.56
195	32.64	51.03	-97.70	17.88	145.60	12.13	.831	.955	1.45
200	32.42	49.42	-97.51	17.83	145.45	12.12	.772	.956	1.35
205	32.20	47.84	-97.79	17.99	145.96	12.16	.715	.955	1.25
210	31.99	46.25	-98.07	14.30	132.57	11.05	.673	.985	1.21
215	31.75	44.94	-102.58	7.00	96.17	8.01	.634	1.117	1.30
220	31.51	44.16	-120.71	.96	36.66	3.06	.592	1.759	1.91
222	31.42	44.08	-175.24	.02	17.75	.42	.577	4.735	5.00

RAY STOPPED, NO CONVERGENCE FOR CURVATURE

Case 3.- The crest diagrams constructed from the rays computed in sample case 2 are illustrated by this sample case. KPLOT was set to 3 to specify this plotting option. The output units are U.S. Customary as specified by setting IOUTUNT to 1. The resulting crest diagram is shown in figure 5. This case required 22 seconds for computer processing. The listing of the input parameters for sample case 3 is given as follows:

NPUT 1

DCON	1.0
DELTAS	0.01
GRID	3038.06
GRINC	1.0
IOUTUNT	1
JPRTFRQ(1)	20
JPRTFRQ(2)	10
JPRTFRQ(3)	5
KPLOT	3
LIMNPT	400
MI	300
MJ	254
NOMOD	72
NOROW	8
NOSETS	1
PLTLNGX	12.0
PLTLNGY	10.0
PLTSCAL	12.5
TIDE	0.0

NPUT2

A	-45.0
HO	6.0
NCREST	5
NSURFCE	1
RAYSPC	1.0
T	12.0
DISTMAX	50.0
DISTMIN	30.0
KCREST	3
RAYANG	120.0
RAYMAX	15.0
RAYMIN	0.01

The output for sample case 3 follows:

```
                    BASIC PROGRAM PARAMETERS

DEPTH UNIT MULT...............DCON         =     1.00
MIN STEP SIZE.................DELTAS       =     .01
GRID SIZE.....................GRID         = 3038.06
DEEP WATER STEP SIZE..........GRINC        =   1.00
OUTPUT UNIT OPTION............IOUTUNT      = 1
PRINT FREQ 1 (D/L>0.5)........JPRTFRQ(1)   = 20
PRINT FREQ 2 (0.25<D/L<0.5)...JPRTFRQ(2)   = 10
PRINT FREQ 3 (D/L<0.25).......JPRTFRQ(3)   =  5
PLOT OPTION...................KPLOT        = 3
MAX NO OF RAY POINTS..........LIMNPT       = 400
GRID LIMITS   ABSCISSA........MI           = 300
              ORDINATE........MJ           = 254
NO OF DEPTH MODULES...........NOMOD        = 72
NO OF ROW MODULES IN NOMOD....NOROW        =  8
NO OF SETS OF RAYS............NOSETS       =  1
PLOT X AXES LENGTH............PLTLNGX      = 12.00
PLOT Y AXES LENGTH............PLTLNGY      = 10.00
PLOT SCALE FACTOR.............PLTSCAL      = 12.50
TIDE HEIGHT...................TIDE         =  0.00

MAX FINAL RAY SPACING.........DISTMAX      =  50.0
MIN FINAL RAY SPACING.........DISTMIN      =  30.0
INITIAL PT FOR CREST CURVE....KCREST       =   3
MIN ANGLE BETWEEN SEGMENTS....RAYANG       = 120.0
MAX INITIAL RAY SPACING.......RAYMAX       =  15.0
MIN INITIAL RAY SPACING.......RAYMIN       =    .01
```

SAMPLE CASE NO 3
PERIOD = 12.00 SEC...TIME STEP = 49.44 SEC...RAY SPACING = 15.0...CREST SPACING = 5...BOTTOM APPROX CODE = 1
RAY NO = 16...SET NO = 1

POINT	X (N.MI.)	Y (N.MI.)	ANGLE (DEG)	DEPTH (FT)	LENGTH (FT)	SPEED (FT/SEC)	REFRACTION COEF	SHOALING COEF	HEIGHT (FT)
1	44.18	124.75	-45.00	453.84	737.34	61.44	1.000	1.000	6.00
20	50.90	118.04	-45.00	419.81	737.34	61.44	1.000	1.000	6.00
40	57.97	110.96	-45.00	318.80	737.34	61.44	1.000	1.000	6.00
50	61.50	107.43	-45.00	289.89	737.34	61.44	1.000	1.000	6.00
60	65.04	103.89	-45.00	270.54	737.34	61.44	1.000	1.000	6.00
70	68.57	100.36	-45.00	260.54	737.34	61.44	1.000	1.000	6.00
80	72.11	96.82	-45.00	248.82	737.34	61.44	1.000	1.000	6.00
90	75.65	93.29	-45.00	255.28	737.34	61.44	1.000	1.000	6.00
100	79.18	89.75	-45.00	240.98	737.34	61.44	1.000	1.000	6.00
110	82.72	86.22	-45.00	251.21	737.34	61.44	1.000	1.000	6.00
120	86.25	82.68	-45.00	249.13	737.34	61.44	1.000	1.000	6.00
130	89.79	79.14	-45.00	245.69	737.34	61.44	1.000	1.000	6.00
140	93.32	75.61	-45.00	234.29	737.34	61.44	1.000	1.000	6.00
150	96.86	72.07	-45.00	241.40	737.34	61.44	1.000	1.000	6.00
160	100.39	68.54	-45.00	255.61	737.34	61.44	1.000	1.000	6.00
170	103.93	65.00	-45.00	251.24	737.34	61.44	1.000	1.000	6.00
180	107.47	61.47	-45.00	237.21	737.34	61.44	1.000	1.000	6.00
190	111.00	57.93	-45.00	242.68	737.34	61.44	1.000	1.000	6.00
200	114.54	54.40	-45.00	234.79	737.34	61.44	1.000	1.000	6.00
210	118.07	50.86	-45.00	232.28	737.34	61.44	1.000	1.000	6.00
220	121.61	47.33	-45.00	215.37	737.34	61.44	1.000	1.000	6.00
230	125.14	43.79	-45.00	213.50	737.34	61.44	1.000	1.000	6.00
240	128.68	40.25	-45.00	202.91	737.34	61.44	1.000	1.000	6.00
250	132.21	36.72	-45.00	200.61	737.34	61.44	1.000	1.000	6.00
260	135.65	33.27	-45.84	179.19	684.38	57.03	1.014	.930	5.66
270	138.89	29.92	-45.90	185.26	688.78	57.40	1.168	.933	6.54
280	142.11	26.54	-47.18	185.62	689.03	57.42	1.346	.933	7.54
290	145.23	23.10	-48.99	167.69	675.07	56.26	1.435	.925	7.97
295	146.71	21.38	-49.56	153.55	661.61	55.13	1.501	.920	8.28
300	148.11	19.69	-51.18	136.07	641.45	53.45	1.612	.915	8.85

RAY STOPPED, REACHED BOUNDARY

APPENDIX A

DESCRIPTION OF SYSTEM SUBROUTINES

The program described in this paper uses several system subroutines to satisfy various data handling and plotting requirements. This appendix lists the names of subroutines and their functions.

AXES	Draws a line, annotates the value of the variable at specified intervals with or without tick marks, and provides an axis identification label.
CALPLT	Creates a reference point for plotting and ends plotting.
CLOSMS	Closes the random-access mass-storage file.
DRAW	Draws a line between specified points.
ENCODE	Reformats data in memory.
LEROY	Sets plot speed.
NFRAME	Advances the plot frame and sets the plot origin.
NOTATE	Draws alphanumeric information for plot annotation and labeling.
NUMBER	Converts a floating point number to binary coded decimal (BCD) format and draws the resulting alphanumeric characters.
OPENMS	Initializes the random-access mass-storage file.
PNTPLT	Draws a given symbol centered on a given coordinate value.
PSEUDO	Initializes and writes a plot vector file.
READMS	Transfers data from mass storage to central memory.
WRITMS	Transfers data from central memory to mass storage.

APPENDIX B

CONSTRUCTION OF WAVE-CREST DIAGRAMS

The interaction of ocean waves with coastal bottom topography can be examined most readily by plotting wave-crest patterns rather than ray patterns. Wave-crest patterns can be constructed by connecting the same point in time on adjacent rays by a curve which is locally orthogonal to each of the rays it intersects. The simplest smooth curve which can be drawn perpendicular to two adjacent rays is represented by a third-order equation. A third-order curve passing through point j on ray $i-1$ can be written

$$y = y_{i-1,j} + a_1(x - x_{i-1,j}) + a_2(x - x_{i-1,j})^2 + a_3(x - x_{i-1,j})^3 \qquad \text{(B1)}$$

where a_1, a_2, and a_3 are coefficients to be determined. The coefficients can be found by constraining the curve to pass also through point j on ray i and constraining the slope of the cubic curve (dy/dx) to be equal to the normals at point j on rays $i-1$ and i (fig. 6(a)). The slopes required for the curve to be orthogonal to rays $i-1$ and i at point j can be approximated by

$$\left(\frac{dy}{dx}\right)_{i-1,j} = \frac{x_{i-1,j-1} - x_{i-1,j+1}}{y_{i-1,j+1} - y_{i-1,j-1}} \qquad \text{(B2)}$$

$$\left(\frac{dy}{dx}\right)_{i,j} = \frac{x_{i,j-1} - x_{i,j+1}}{y_{i,j+1} - y_{i,j-1}} \qquad \text{(B3)}$$

With these results, the three coefficients in equation (B1) are determined to be

$$a_1 = \left(\frac{dy}{dx}\right)_{i-1,j} \qquad \text{(B4)}$$

$$a_2 = \frac{3(y_{i,j} - y_{i-1,j}) - \left[2\left(\frac{dy}{dx}\right)_{i-1,j} + \left(\frac{dy}{dx}\right)_{i,j}\right](x_{i,j} - x_{i-1,j})}{(x_{i,j} - x_{i-1,j})^2} \qquad \text{(B5)}$$

$$a_3 = \frac{-2(y_{i,j} - y_{i-1,j}) + \left[\left(\frac{dy}{dx}\right)_{i-1,j} + \left(\frac{dy}{dx}\right)_{i,j}\right](x_{i,j} - x_{i-1,j})}{(x_{i,j} - x_{i-1,j})^3} \tag{B6}$$

The cubic equation is used to determine the points on the crest segment between two adjacent rays. To give the crest curve a smooth appearance, seven points are used to describe the crest segment.

When waves approach intermediate water and refraction begins to occur, adjacent rays can cross. Straightforward construction of crest segments in such situations is probably unrealistic in a physical sense and could often result in a zigzag crest pattern. To circumvent this problem, logic has been added to delete all or part of the crest segment between crossing adjacent rays $i-1$ and i. In order to signal the onset of this situation, the angle between adjoining first-order crest segments is computed. In figure 6(b), the angle θ between the first-order segment joining point j on ray $i-2$ to point j on ray $i-1$, and the first-order segment joining point j on ray $i-1$ to point j on ray i is given by

$$\theta = \cos^{-1}\left[\frac{(x_i - x_{i-1})(x_{i-2} - x_{i-1}) + (y_i - y_{i-1})(y_{i-2} - y_{i-1})}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}\,\sqrt{(x_{i-2} - x_{i-1})^2 + (y_{i-2} - y_{i-1})^2}}\right] \tag{B7}$$

Whenever this angle is less than a specified input parameter (RAYANG), the curved crest segment connecting point j on ray $i-1$ to point j on ray i is modified. Since a maximum of seven points are used to describe the crest segment, the modification is accomplished by starting the segment at an intermediate point between the rays rather than at point j on ray $i-1$. This results in crest patterns being disjointed when adjacent rays $i-1$ and i cross. The program input parameter specifying the point at which to begin construction of the modified curved crest segment is KCREST and can be any number between 1 and 7 inclusive. The disjointed effect is illustrated in figure 5 with RAYANG equal to 120° and KCREST equal to 3. In this case only two-thirds of the crest curve is drawn when the angle between adjoining first-order crest segments is less than 120°. The logic for this procedure is incorporated in subroutine PLOTR. Crest plotting is initiated when the input parameter KPLOT is equal to 3.

APPENDIX C

DISCUSSION OF SUBROUTINE RAYOPT

When wave rays approach shallow water and refraction occurs, cases of extreme convergence or divergence can occur, resulting in either large areas completely devoid of rays or small areas covered by many rays. To allow the user a degree of control over nearshore ray spacing, the subroutine RAYOPT was developed. In subroutine RAYOPT, desired nearshore ray spacing is achieved by perturbation of the initial ray spacing. Upper and lower bounds specified by input parameters are imposed on both the initial ray spacing and the final ray spacing. If desired results are not obtained in 10 iterations, then the 10th iteration is specified as the solution.

To start the RAYOPT procedure, an initially small value is input or assigned by default to the input parameter RAYSPC, which is the initial ray spacing. The initial ray in the set is found by using subroutine RAYINIT. A second ray is begun at a distance RAYSPC from the initial ray, its complete path is generated, and its final separation distance with respect to the preceding ray is computed. If this distance is greater than the input parameter DISTMAX, the current RAYSPC is halved and a new trial ray is generated. Conversely, if this distance is less than the input parameter DISTMIN, the current RAYSPC is doubled and a new trial ray is generated. This procedure continues until the final separation distance lies between the limits DISTMIN and DISTMAX. If at anytime in the iteration process RAYSPC falls below the input value RAYMIN or exceeds the input value RAYMAX, the iteration process terminates and RAYSPC is set to the respective limit. In some cases, the final ray separation distance oscillates between values of DISTMIN and DISTMAX. When this occurs, an acceptable initial ray spacing is computed by using a Newton-Raphson iteration scheme. For all cases, the final iterated value for RAYSPC is used as the first trial value in computing the next ray.

Graphical results obtained by using subroutine RAYOPT can be seen by comparing figures 3 and 4. Figure 3 presents a ray diagram computed with a uniform initial ray spacing (RAYSPC) of 5 grid units, with no control exercised over final spacing. There are large gaps between adjacent rays in the nearshore region in this diagram. In figure 4, final ray spacing is controlled with DISTMIN equal to 30 grid units and DISTMAX equal to 50 grid units. The initial ray spacing is constrained to lie between 0.01 grid units (RAYMIN) and 15 grid units (RAYMAX). As can be seen, the nearshore ray spacing is much more uniform than that in figure 3. In deeper areas in which no refraction occurs, ray spacing is increased by the subroutine until either RAYMAX or DISTMAX is exceeded. User control over the final ray spacing is particularly useful in cases in which crest curves are to be constructed between adjacent rays, as described in appendix B. Figure 7 shows the complete crest pattern constructed for the ray diagram shown in figure 3, which was generated by using a constant initial ray spacing of 5 grid units. In constructing the crest patterns of figure 7, RAYANG was specified as 120^{o} and KCREST was specified as 1. This crest pattern does not accurately depict nearshore wave fronts because of the sparsity of wave rays in certain nearshore areas. The crest pattern of figure 5 was constructed for the ray diagram of figure 4, in which final ray spacing was controlled. Here, a more realistic version of the wave fronts for the nearshore zone is seen.

An experimental period may be required on the part of the user to obtain an acceptable crest pattern for a particular application. Care should be taken when selecting values for the input parameters required in generating crest patterns. The following discussion on each of the required input parameters is included as a guide in specifying these parameters for any application:

DISTMAX — This input constant is the maximum permissible final ray separation distance. Too large a number causes wide nearshore ray separation, such as that seen in figure 3, and results in unsatisfactory crest patterns (fig. 7). Too small a number causes too many rays to be generated and results in patterns similar to that shown in figure 8, where DISTMAX is equal to 30 grid units. In order for the iteration process to converge in a few cycles, this number should differ from DISTMIN by a reasonable margin.

DISTMIN — This parameter is the minimum permissible final ray separation distance. This number should be large enough to insure sufficient nearshore separation between adjacent rays.

KCREST — This parameter controls modification of the curved crest segment to be drawn between adjacent rays which cross. Normally, a value equal to 2 or 3 produces an acceptable diagram. Failure to draw at least part of the curved segment (i.e., setting KCREST = 7) can result in large gaps in the pattern, as seen in figure 9 (compared with fig. 5).

RAYANG — This parameter is the limiting angle between adjoining first-order crest segments. If, in constructing a crest pattern, the angle between adjoining first-order crest segments is less than RAYANG, construction of the current curved segment begins at point KCREST rather than at the first point. A value for RAYANG around 120° should give acceptable patterns.

RAYMAX — This parameter is the maximum permissible initial ray spacing. RAYMAX should be set to a large value to allow an optimum search for an acceptable initial ray point. In addition, only a few rays are required to define the crest pattern in situations in which the rays remain in deep water over their entire path.

RAYMIN — This parameter is the minimum permissible initial ray spacing. In all instances, this number should be set to an arbitrarily small value in order that final ray constraints can be met.

APPENDIX D

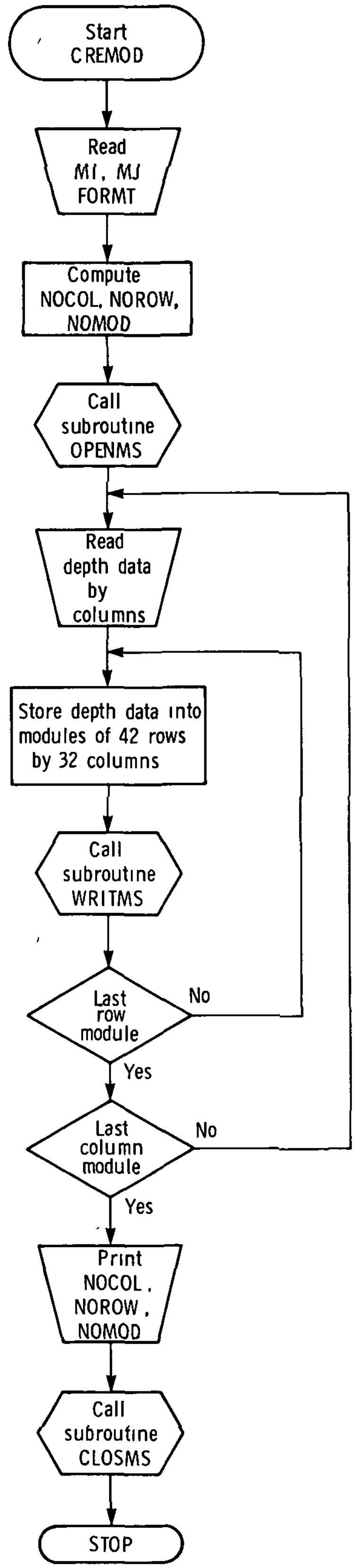
UTILITY PROGRAM CREMOD

In the wave refraction computer model described in this paper, bathymetry data are stored and retrieved in the form of overlapped modules of dimensions 42 by 32. Random-access techniques are used to pass data modules as required from a peripheral mass-storage file to the central memory of the computer. The potential user, however, generally has bathymetry data available in the form of an array of dimensions MI by MJ covering the entire region of interest. This appendix describes a utility program CREMOD which can be used to modularize the regional array supplied by the user. Upon completion of execution of this program, the modularized data reside on the peripheral file TAPE1, which then can be saved as a permanent file by the user for later input to the refraction model.

It is assumed that the regional array is supplied in card deck form and that the data are arranged in columnwise order with the first column corresponding to the x-axis of the region. The total number of rows of data MI and the total number of columns of data MJ are required input parameters. The user must also designate the format (FORMT) with which the data cards are read. To avoid possible extraneous results, it is strongly recommended that the format be chosen so that the input record length is an integral divisor of MI. The regional array is divided into a total of NOMOD overlapped modules which are indexed by columns, as shown in figure 10, and written to the disk file TAPE1 by using system subroutine WRITMS. If the regional configuration is such that portions of certain modules extend beyond the boundaries of the MI by MJ array (as illustrated in fig. 10), data values within those portions are set equal to the values in the corresponding column of row MI or in the corresponding row of column MJ, as the case requires. Values for NOROW, the number of module rows, NOCOL, the number of module columns, and NOMOD are printed when modularization is complete.

Program CREMOD uses the system subroutines OPENMS, WRITMS, and CLOSMS for initializing, writing records to, and closing the mass-storage file TAPE1. Reference 10 provides a detailed description of each of these routines. On the Control Data Cyber 175 computer under NOS 1.2, program CREMOD required a central memory field length of 64000_8 and, for a sample case in which MI and MJ were both 100, required 4 seconds of CPU time. The user should again be reminded to save the file TAPE1 as a permanent file after execution of program CREMOD. The flow diagram and listing of program CREMOD follow.

APPENDIX D



Start
CREMOD
Read
MI, MJ
FORMT
Compute
NOCOL, NOROW,
NOMOD
Call
subroutine
OPENMS
Read
depth data
by
columns
Store depth data into
modules of 42 rows
by 32 columns
Call
subroutine
WRITMS
Last
row
module
No
Yes
Last
column
module
No
Yes
Print
NOCOL,
NOROW,
NOMOD
Call
subroutine
CLOSMS
STOP

```
      PROGRAM CREMOD (INPUT,OUTPUT,TAPE5=INPUT,TAPE1)
      DIMENSION DEP(420,32),DMOD(42,32),IBLK(101)
*
      READ (5,180) MI,MJ,FORMT
*
*  DETERMINE NO. OF MODULE ROWS AND COLUMNS.
*
      NOCOL = MJ/29
      NCOUT = MOD (MJ,29)
      IF (NCOUT .LT. 4) GO TO 10
      NOCOL = NOCOL+1
      GO TO 20
   10 NCOUT = NCOUT+29
   20 NOROW = MI/39
      NROUT = MOD (MI,39)
      IF (NROUT .LT. 4) GO TO 30
      NOROW = NOROW+1
      GO TO 40
   30 NROUT = NROUT+39
   40 NOMOD = NOROW*NOCOL
*
*  OPEN MASS STORAGE FILE, TAPE1.
*
      CALL OPENMS (1,IBLK,NOMOD+1,0)
*
*  READ 3 COMPLETE COLUMNS OF DATA.
*
      READ (5,FORMT) ((DEP(I,J),I=1,MI),J=1,3)
      DO 170 NC=1,NOCOL
      IF (NC .LT. NOCOL) GO TO 70
*
*  FINAL MODULE COLUMN. ONLY (NCOUT-3) SUB-COLUMNS OF DATA LEFT.
*
      IF (NCOUT .EQ. 32) GO TO 70
      READ (5,FORMT) ((DEP(I,J),I=1,MI),J=4,NCOUT)
*
*  SET VALUES IN EXTERIOR COLUMNS = TO VALUES IN LAST COLUMN.
*
      J1 = NCOUT+1
      DO 60 J=J1,32
      DO 50 I=1,MI
      DEP(I,J) = DEP(I,NCOUT)
   50 CONTINUE
   60 CONTINUE
      GO TO 80
*
*  FULL MODULE COLUMN AVAILABLE. READ COLUMNS 4 THRU 32.
*
```

```
70  READ (5,FORMT) ((DEP(I,J),I=1,MI),J=4,32)
*
*  FILL DMOD ARRAY FOR NOROW MODULES IN THIS COLUMN. WRITE EACH TO
*  MASS STORAGE FILE.
*
80  DO 140 NR=1,NOROW
    IZERO = 39*(NR-1)
    NM = (NC-1)*NOROW+NR
    IF (NR .LT. NOROW) GO TO 90
*
*  FINAL MODULE ROW. NROUT-3 MORE SUB-ROWS OF DATA AVAILABLE.
*
    IMAX = NROUT
    GO TO 100
90  IMAX = 42
100 DO 130 J=1,32
    DO 110 I=1,IMAX
    DMOD(I,J) = DEP(IZERO+I,J)
110 CONTINUE
    IF (IMAX .EQ. 42) GO TO 130
    I1 = IMAX+1
    DO 120 I=I1,42
    DMOD(I,J) = DMOD(IMAX,J)
120 CONTINUE
130 CONTINUE
    CALL WRITMS (1,DMOD,1344,NM)
140 CONTINUE
    IF (NC .EQ. NOCOL) GO TO 170
*
*  SUB-COLUMNS 30-32 WILL BE SUB-COLUMNS 1-3 OF NEXT MODULE.
*
    DO 160 J=1,3
    DO 150 I=1,MI
    DEP(I,J) = DEP(I,J+29)
150 CONTINUE
160 CONTINUE
170 CONTINUE
    PRINT 190, NOROW,NOCOL,NOMOD
    CALL CLOSMS (1)
    RETURN
*
180 FORMAT (2I5,A10)
190 FORMAT (2X,*NOROW =*,I3/2X,*NOCOL =*,I3/2X,*NOMOD =*,I3)
    END
```

REFERENCES

1. Johnson, J. W.; O'Brien, M. P.; and Isaacs, J. D.: Graphical Construction of Wave Refraction Diagrams. H. O. Publ. No. 605, U.S. Navy, Jan. 1948.

2. Griswold, Gale M.: Numerical Calculation of Wave Refraction. J. Geophys. Res., vol. 68, no. 6, Mar. 15, 1963, pp. 1715-1723.

3. Wilson, W. Stanley: A Method for Calculating and Plotting Surface Wave Rays. Tech. Mem. No. 17, Coastal Eng. Res. Center, U.S. Army Corps Eng., Feb. 1966. (Available from DDC as AD 636 771.)

4. Dobson, R. S.: Some Applications of a Digital Computer to Hydraulic Engineering Problems. Tech. Rep. No. 80 (Contract Nonr 225(85)), Dep. Civil Eng., Stanford Univ., June 1967. (Available from DDC as AD 659 309.)

5. Munk, W. H.; and Arthur, R. S.: Wave Intensity Along a Refracted Ray. NBS Circ. 521, U.S. Dep. of Commer., 1951, pp. 95-108.

6. Goldsmith, Victor; Morris, W. Douglas; Byrne, Robert J.; and Whitlock, Charles H.: Wave Climate Model of the Mid-Atlantic Shelf and Shoreline (Virginia Sea) - Model Development, Shelf Geomorphology, and Preliminary Results. NASA SP-358, VIMS SRAMSOE No. 38, 1974.

7. Poole, Lamont R.: Random-Access Technique for Modular Bathymetry Data Storage in a Continental-Shelf Wave-Refraction Program. NASA TM X-3018, 1974.

8. Poole, Lamont R.: Comparison of Techniques for Approximating Ocean Bottom Topography in a Wave-Refraction Computer Model. NASA TN D-8050, 1975.

9. Universal Transverse Mercator Grid. Dep. Army Tech. Manual 5-241-8, U.S. Army Publications Directorate, Apr. 1973.

10. Cyber Record Manager Version 1, 7000 Record Manager Reference Manual. Publ. No. 60495700, Revision B, Control Data Corp., May 1977.

TABLE I.- DESCRIPTION OF ELEMENTS IN NAMELIST NPUT1

Variable name	Description	Default value
DCON	Multiplier to convert depth units to feet for internal computations	1.0
DELTAS	Minimum step length along ray in shallow water, grid units	0.01
GRID	Number of feet per grid division	None
GRINC	Step length along ray in deep water, grid units	1.0
IOUTUNT	Output units: nautical miles for spatial coordinates x and y; for all other parameters; 1 = U.S. Customary Units; 2 = SI Units	2
JPRTFRQ(1)	Print frequency of ray parameters when depth is greater than one-half the wavelength, time steps	20
JPRTFRQ(2)	Print frequency when depth is greater than one-fourth the wavelength but less than one-half the wavelength, time steps	10
JPRTFRQ(3)	Print frequency when depth is less than one-fourth the wavelength, time steps	5
LIMNPT	Maximum number of computation points along a ray	400
KPLOT	Plotting option: 0 = No plot 1 = Plot wave rays with uniform initial spacing 2 = Plot wave rays with controlled final spacing 3 = Plot wave crests	0
MI	Total number of rows in regional bathymetry data array	None
MJ	Total number of columns in regional bathymetry data array	None
NOMOD	Total number of 42 by 32 bathymetry data modules; maximum allowed is 100	None
NOROW	Number of modules in x-direction	None
NOSETS	Number of sets of rays to be computed	1
PLTLNGX	Length of x-axis for plot, inches	10.0
PLTLNGY	Length of y-axis for plot, inches	10.0
PLTSCAL	Scale factor for x- and y-axes; change in x or y value per inch of plot (input required for plotting only)	None
TIDE	Height of tide, feet	0.0

TABLE II.- DESCRIPTION OF ELEMENTS IN NAMELIST NPUT2

Variable name	Description	Default value
A	Initial ray direction, degrees	None
HO	Deepwater wave height, feet	None
NCREST	Spacing of crest tick marks or crest curves; integer multiple of program computational time step	5
NSURFCE	Bottom topography approximation technique: 1 = Quadratic least squares 2 = Cubic least squares 3 = Constrained bicubic interpolation	1
RAYSPC	Initial spacing between wave rays in deep water, grid units	5.0
T	Wave period, seconds	None
DISTMAX[a]	Maximum permissible final separation distance between adjacent rays, grid units	999.
DISTMIN[a]	Minimum permissible final separation distance between adjacent rays, grid units	25.
KCREST[a]	Point (between 1 and 7) between adjacent rays at which to begin plotting curved crest segment when angle between adjoining first-order segments is less than RAYANG	1
RAYANG[a]	Minimum acceptable angle between adjoining first-order crest segments, below which crest plotting is modified, degrees	120.
RAYMAX[a]	Maximum permissible initial ray spacing, grid units	25.
RAYMIN[a]	Minimum permissible initial ray spacing grid units	0.01

[a]These parameters are used only for ray sets in which nearshore ray spacing is controlled (KPLOT = 2 or 3).

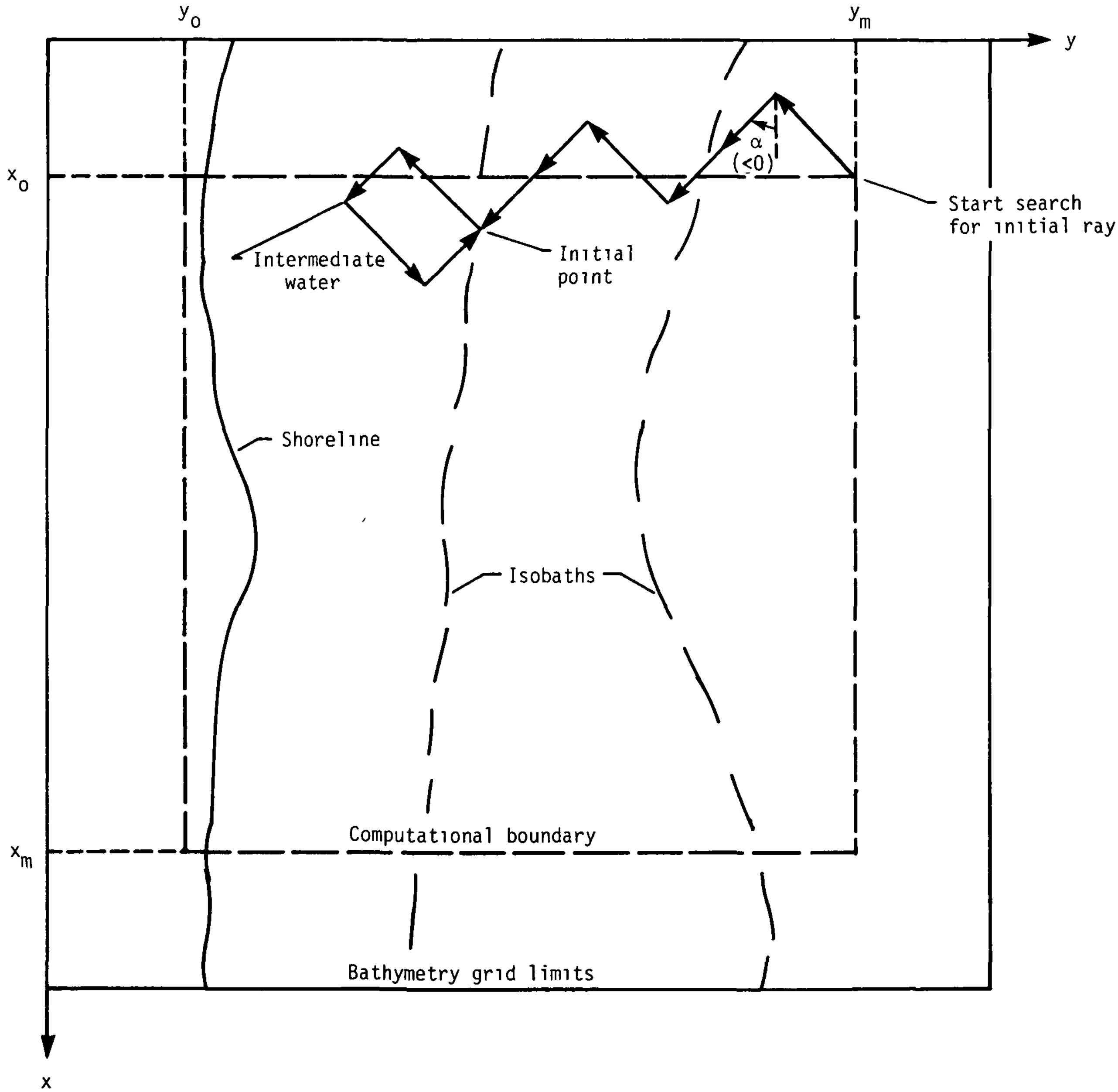


Figure 1.- Determination of starting point for first ray for ray direction greater than -90°.

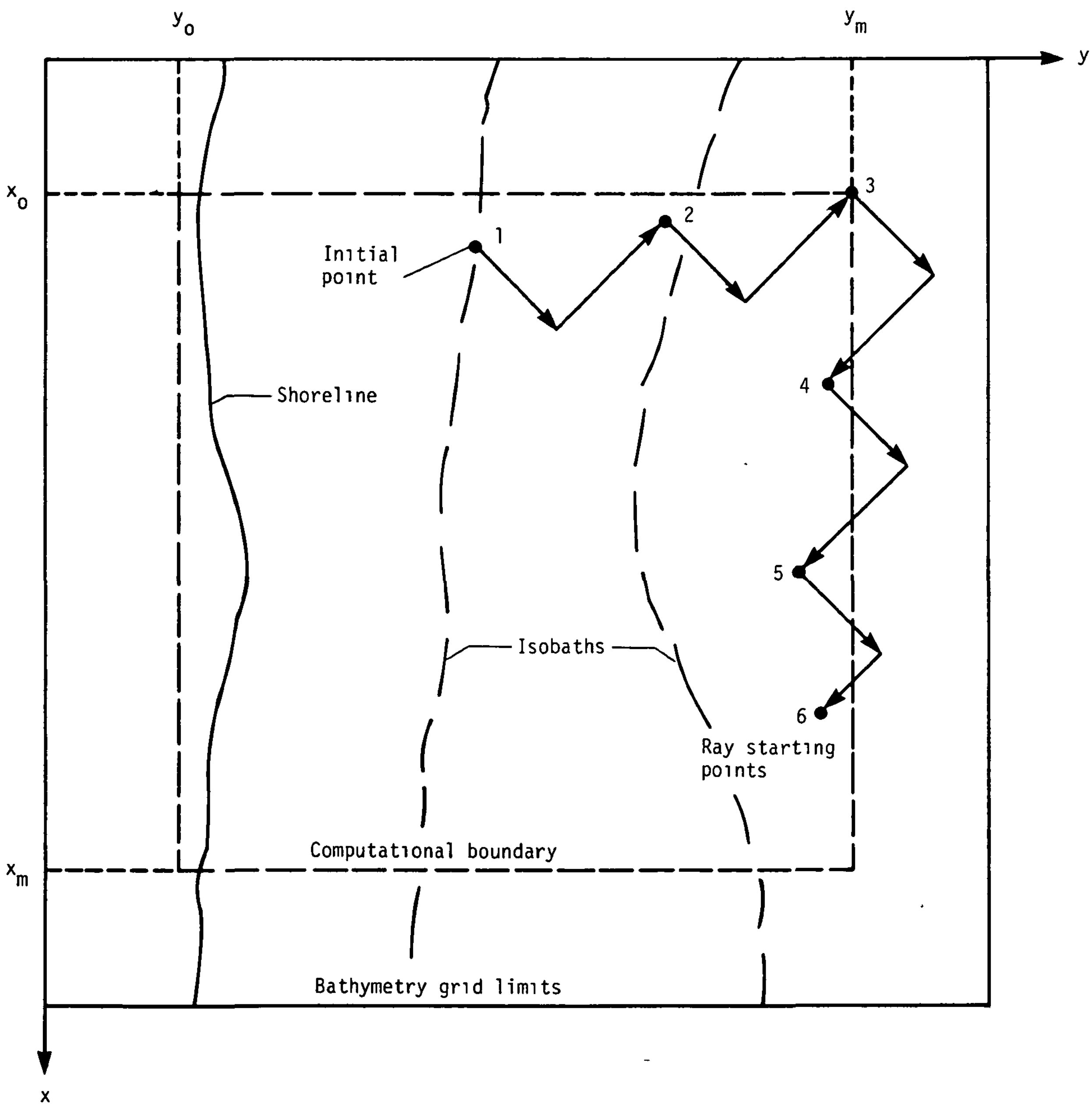


Figure 2.- Determination of ray starting points for ray direction greater than -90°.

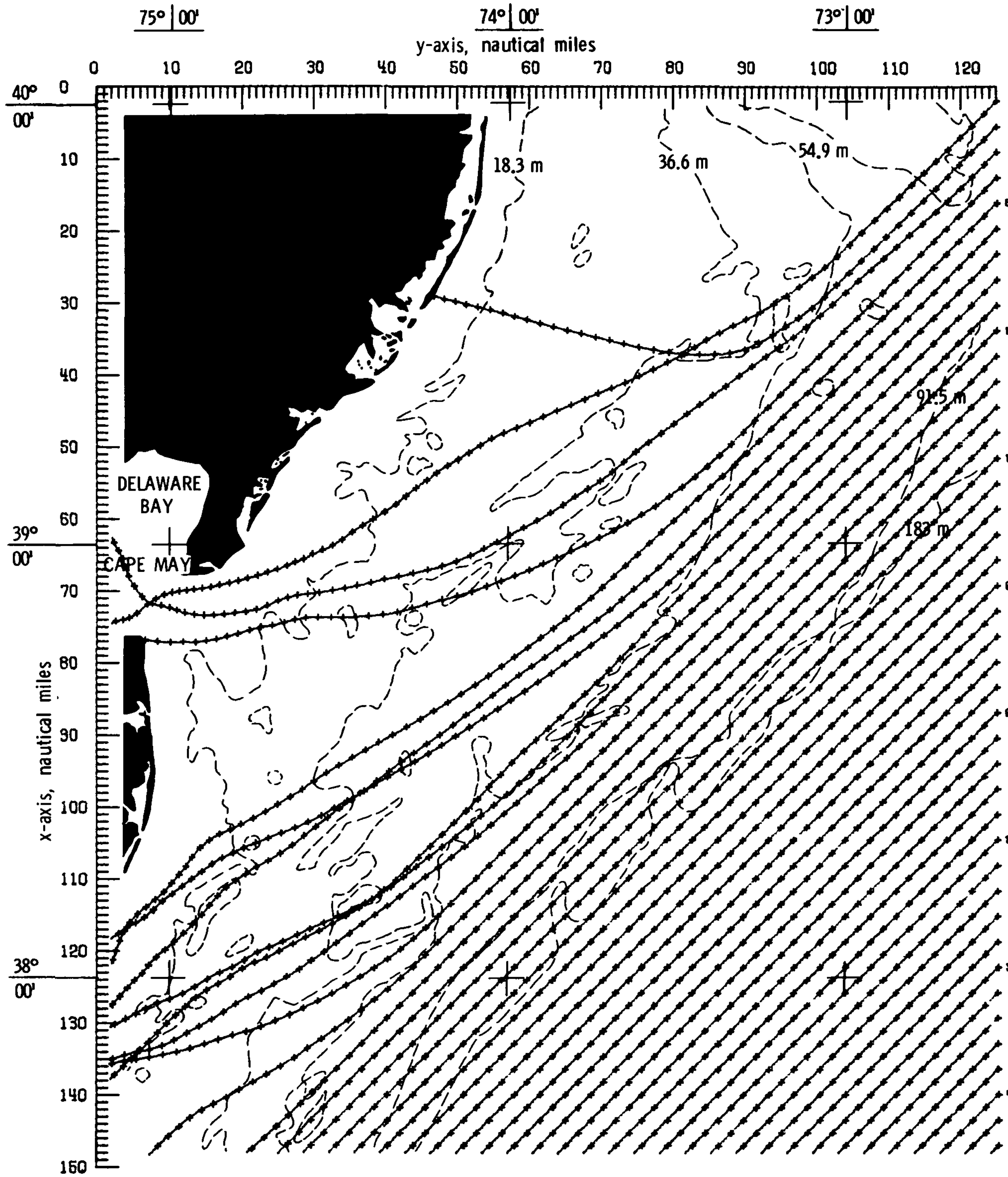


Figure 3.- Ray diagram with equally spaced initial points.

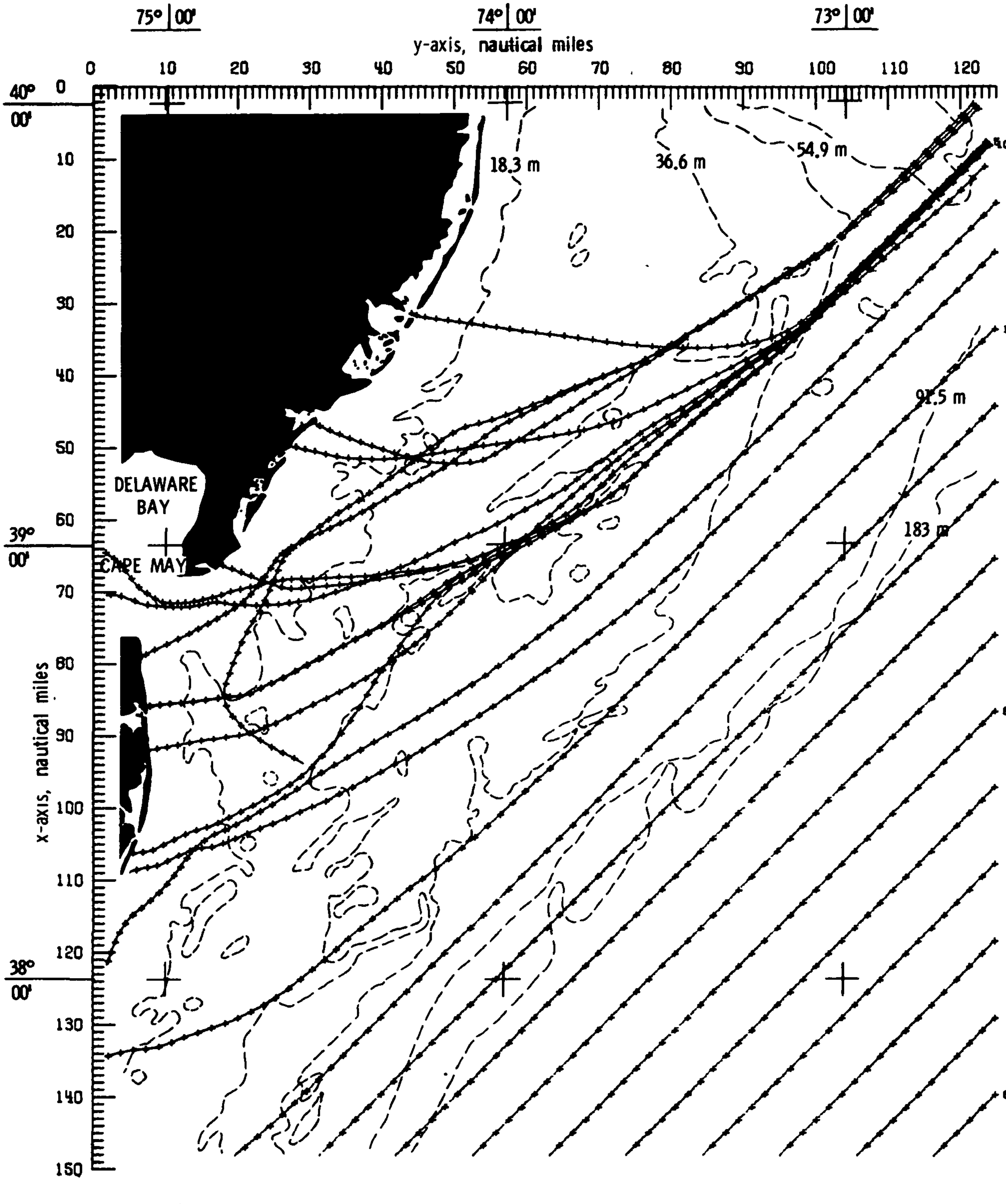


Figure 4.- Ray diagram with controlled final (nearshore) spacing.

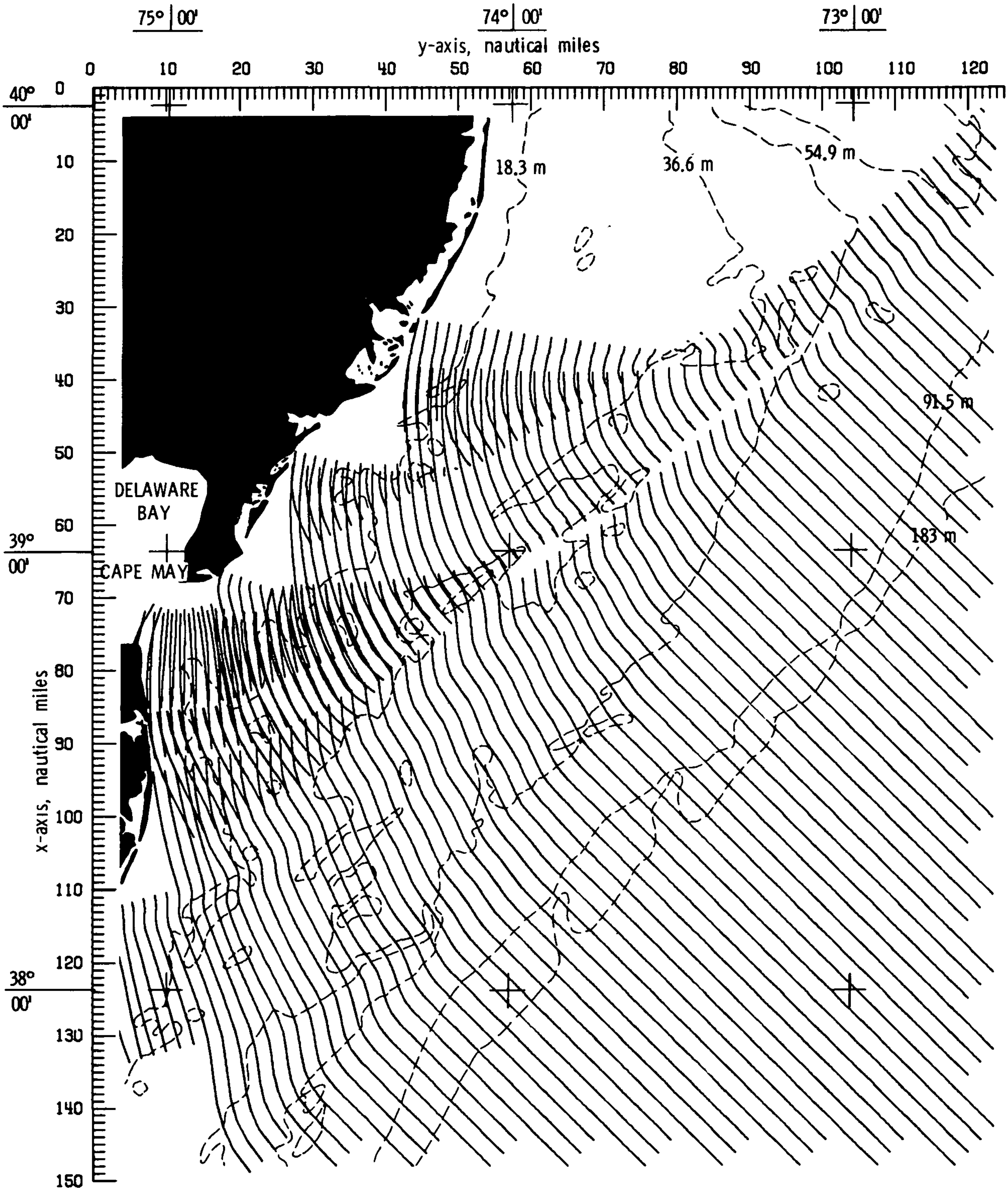
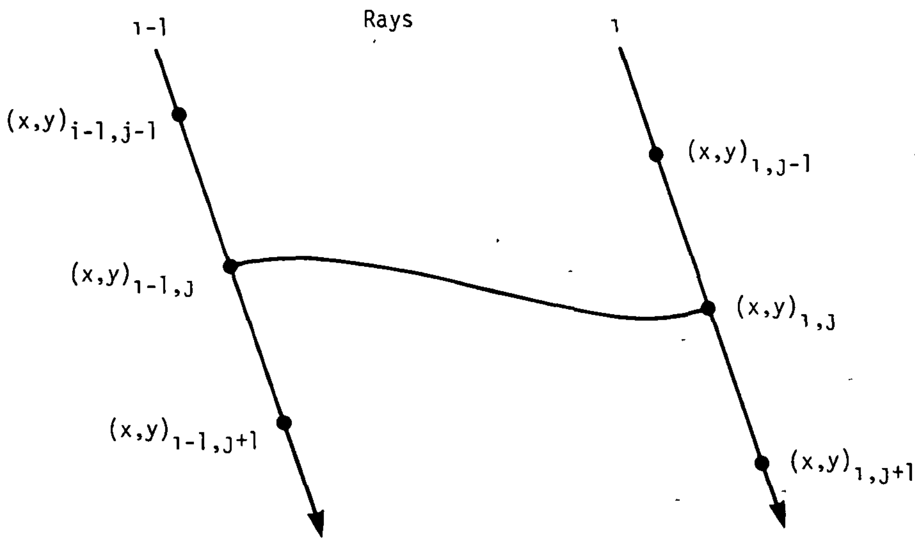
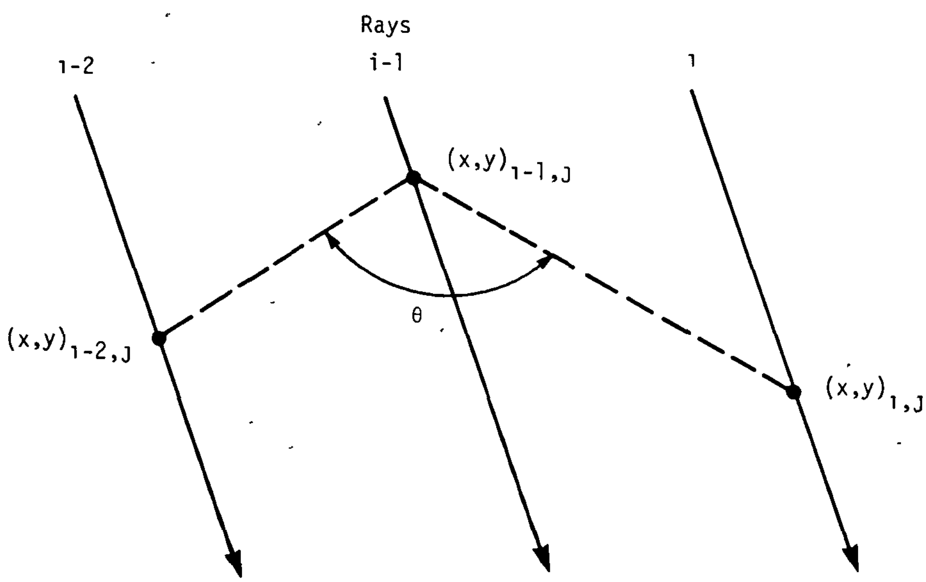


Figure 5.- Wave-crest diagram using modified segments in areas of crossing rays.



(a) Crest curve connecting two rays.



(b) Angle between adjoining first-order crest segments.

Figure 6.- Wave-crest geometry.

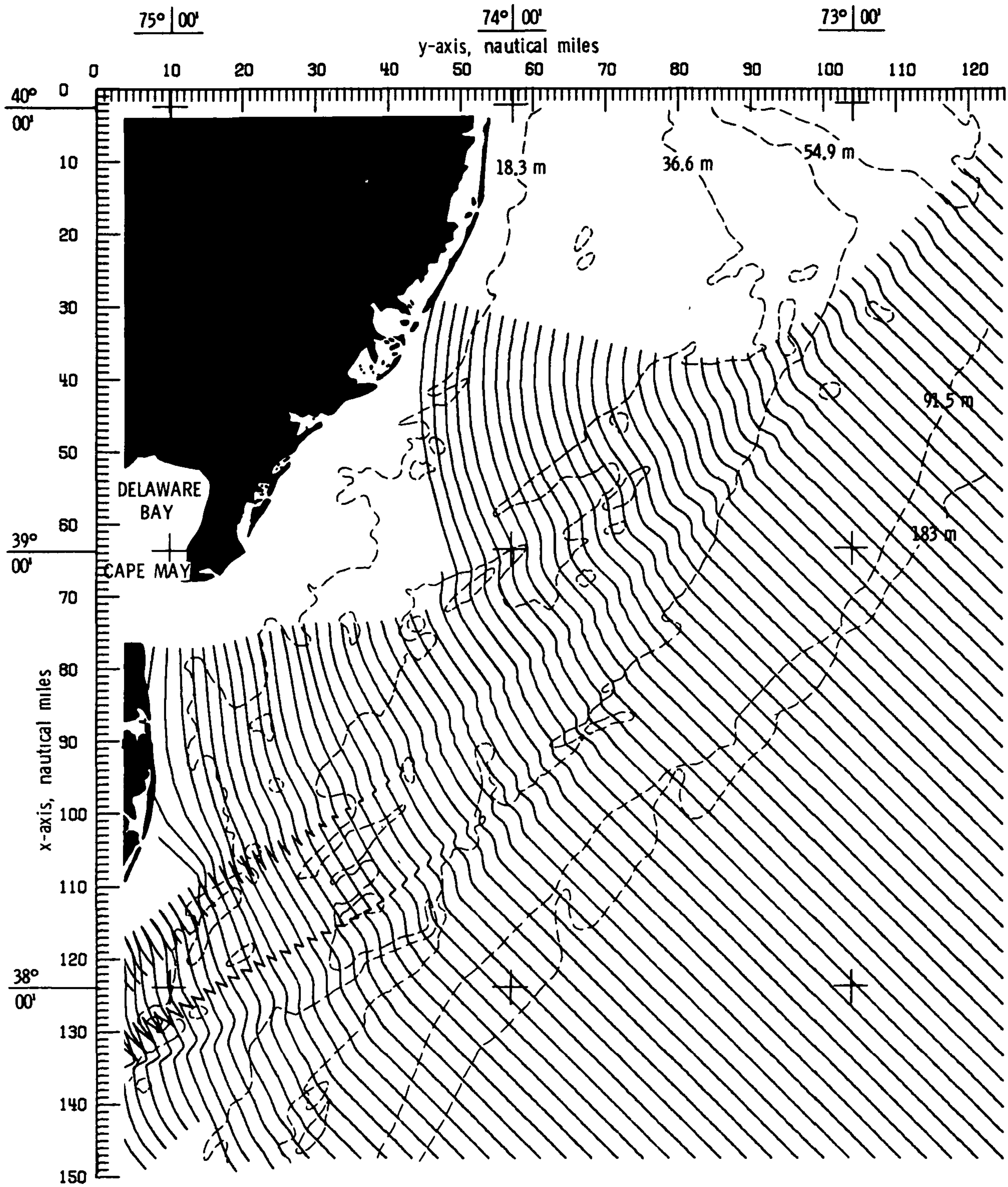


Figure 7.- Wave-crest diagram with equally spaced initial ray points.

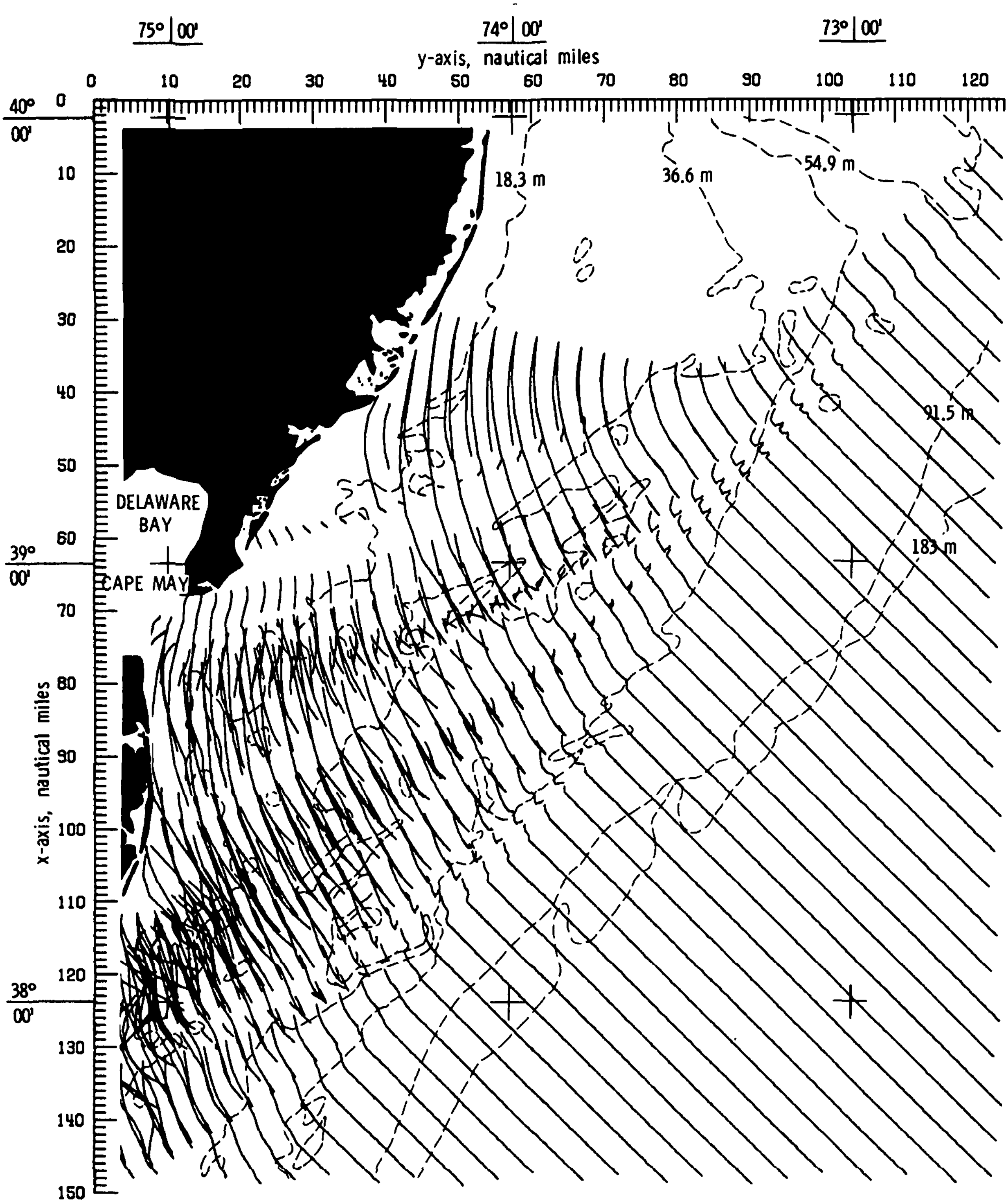


Figure 8.- Wave-crest diagram resulting from generation of too many rays due to small value of DISTMAX.

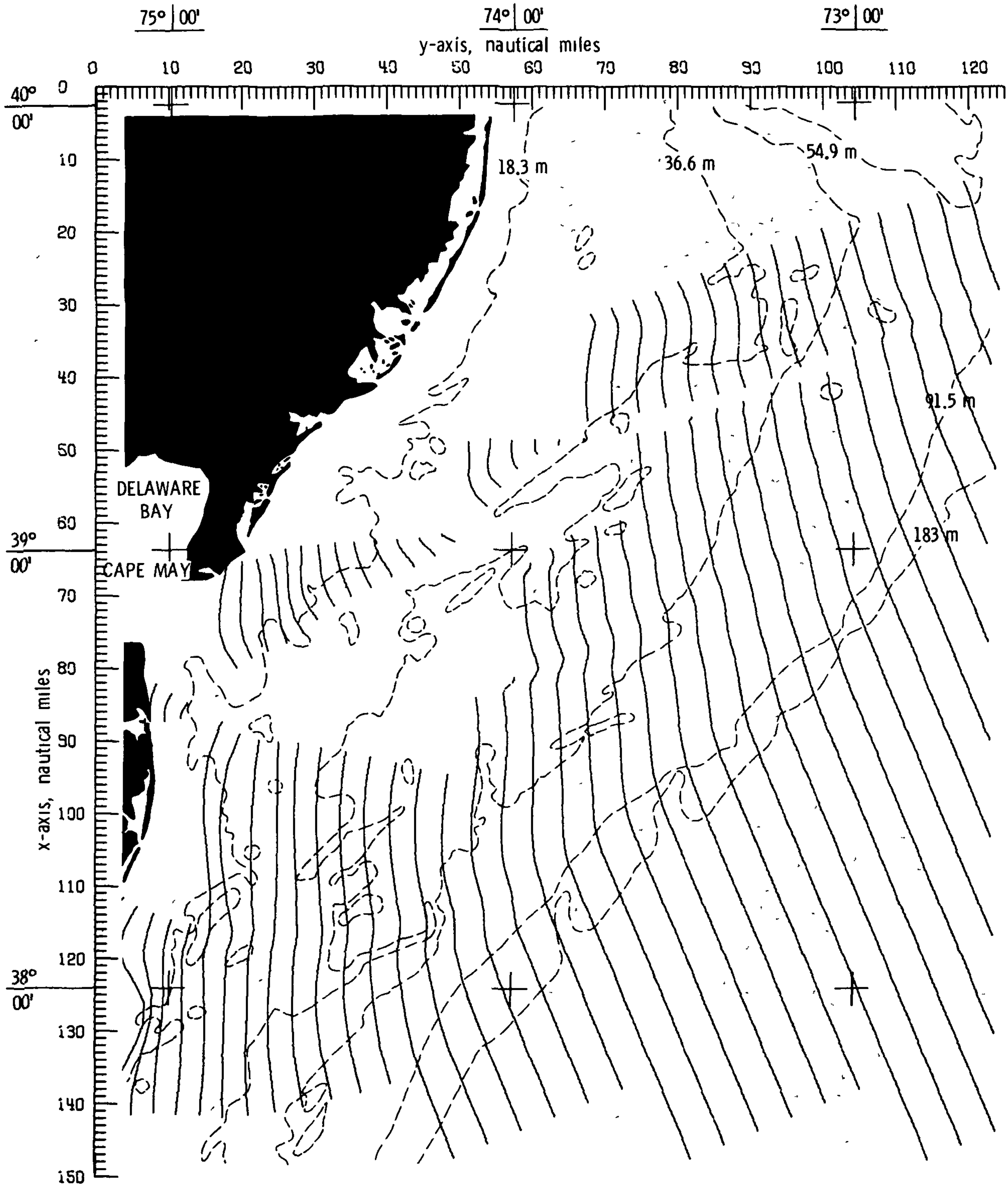


Figure 9.- Wave-crest diagram constructed with KCREST = 7.

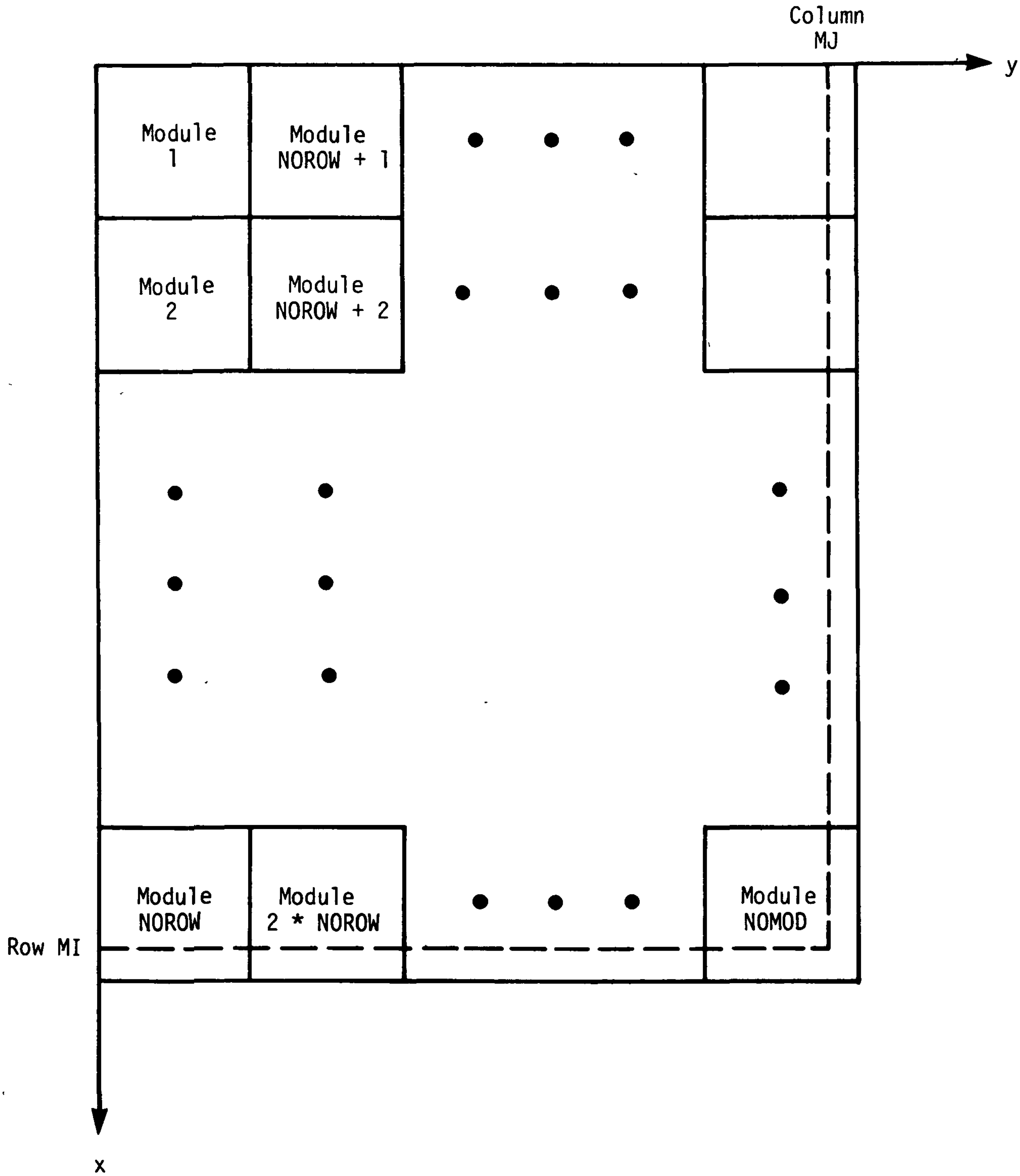


Figure 10.- Sequence of bathymetry data modules.

1. Report No NASA TM X-74076	2. Government Accession No	3. Recipient's Catalog No
4. Title and Subtitle MINIMAL-RESOURCE COMPUTER PROGRAM FOR AUTOMATIC GENERATION OF OCEAN WAVE RAY OR CREST DIAGRAMS IN SHOALING WATERS		5. Report Date December 1977
		6. Performing Organization Code
7. Author(s) Lamont R. Poole, Stuart R. LeCroy, and W. Douglas Morris		8. Performing Organization Report No L-11818
		10. Work Unit No 161-07-02-03
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		11. Contract or Grant No
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Lamont R. Poole and W. Douglas Morris: Langley Research Center.
Stuart R. LeCroy: Vought Corporation, Hampton, Virginia.

16. Abstract

This paper describes an improved computer program for studying linear ocean wave refraction. The program features random-access modular bathymetry data storage. Three bottom topography approximation techniques are available in the program which provide varying degrees of bathymetry data smoothing. Refraction diagrams are generated automatically and can be displayed graphically in three forms: ray patterns with specified uniform deepwater ray density, ray patterns with controlled nearshore ray density, or crest patterns constructed by using a cubic polynomial to approximate crest segments between adjacent rays.

17. Key Words (Suggested by Author(s)) Ocean wave refraction Wave refraction computer model Random-access bathymetry data storage Refraction diagram construction	18. Distribution Statement Unclassified - Unlimited Subject Category 48

19. Security Classif (of this report) Unclassified	20. Security Classif (of this page) Unclassified	21. No of Pages 93	22. Price* $6.00

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1977

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

8 2 1U,E, 113077 S
MCDONNELL DOUGLAS CORP
ATTN: PUBLICATIONS GROUP,
P O BOX 516
ST LOUIS MO 63166

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return