琉球大学学術リポジトリ

CATWAVES Prediction of Linear and Nonlinear Wave Motion in Unbounded Coastal Domains

メタデータ	言語: 出版者: 琉球大学工学部 公開日: 2007-10-28 キーワード (Ja): キーワード (En): Nonlinear wave evolution, Wave height distribution, Open problem, Model wave equation, Mild-slope equation, Finite element method, Mesh generation 作成者: Tsutsui, Shigeaki, 筒井, 茂明 メールアドレス: 所属:
URL	http://hdl.handle.net/20.500.12000/2221

CATWAVES
Prediction of Linear and Nonlinear Wave Motion in Unbounded Coastal Domains

Shigeaki TSUTSUI*

Abstract

The CATWAVES is a package of Fortran programs, based on the finite element method, for linear and nonlinear wave analyses in unbounded coastal domains. This package consists of three procedures: (I) the finite element mesh generation routines for preprocessing, (II) the finite element analysis routines of wave motion and (III) the graphics routines for post-processing. The model equation of wave motion implemented in CATWAVES is a variant of the mild-slope equation and combines the linear dispersion with weak nonlinearity. The boundary conditions applicable to the model are wave reflection along the shorelines and the boundary of coastal structures, discontinuous changes in water depth, such as occur in reef coasts, and the radiation condition at infinity. Initially, the CATWAVES imposes, on the user, preparation of two data files for the mesh generation routine, i.e., bathymetric and finite element mesh data. After preparation of these data, the system successively creates the input/output data files and leads to the final results together with high-quality graphics.

Keywords: Nonlinear wave evolution, Wave height distribution, Open problem, Model wave equation, Mild-slope equation, Finite element method, Mesh generation.

1. Introduction

The CATWAVES - Computer-Aided Testing of WAVES - is the system of linear and nonlinear wave analyses based on the finite element method and is a package of Fortran programs.

The linear analysis of wave motion in unbounded coastal domains is fundamental for environmental assessment of influence due to the changes in wave conditions in planning all sorts of coastal facilities. The nonlinear analysis of waves in shallow water, on the contrary, becomes significant especially in nonlinear wave evolution, for example, the harbor resonance with long waves.

In many analyses of wave motion based on the finite element method, the requirement of an enormous amount of input data is a serious shortcoming. Success of the method depends, to a large degree, on preprocessing, i.e., how easily the user can prepare the bathymetric data and create the finite element mesh over the model region. Visualization of the numerical results is also significant to understand the wave phenomena in the regions of the problem. With these in mind, therefore, this set of Fortran programs consists of three main procedures: (I) the mesh generation for preprocessing, (II) the linear and nonlinear finite element solution routines and (III) the post-processor to visualize the numerical results.

This document describes the programing structure of the system and demonstrates how to simulate linear and nonlinear wave motion in the near shore regions and to predict wave height distribution together with graphic results.

The CATWAVES was firstly developed (Tsutsui, 1989), during the author's stay at the Centre for Water Research, University of Western Australia, from September 1988 to June 1989. In the original codes, however, there might be some parts to be improved, especially, in the treatment of the radiation condition at infinity so as to develope the model for nonlinear wave evolution, in saving the computer storage, and in the method of numerical solution for the large and sparse system of algebraic equations. These have been revised in the past decade (e.g., Sulaiman et al., 1994; Tsutsui et al., 1993, 1996, 1998; Tsutsui, 1995, 1999, 2000). The chief features of the function of CATWAVES are as follows:

Interface

- All the routines work on any computer's operation system because there is no machine-dependent statement/command. As the data input interface, then, the user of CATWAVES can interactively input data necessary for computation, according to the instructions appearing on the screen.

Received 25 December 2000.

* Dept. of Civil Engineering and Architecture, Faculty of Engineering.

Part of the paper was presented at the Centre for Water Research, University of Western Australia, 1988, and in Proc. Coastal Eng., JSCE, Vol.46., 1999.

Mesh generation routine

- The mesh generation routines for bathymetry and the finite element analysis are available.
- In the finite element mesh generation, an algorithm of renumbering of node numbers, similar to a widely-used Cuthill & McKee (1969) ordering, is introduced to reduce the variable-band width of the coefficient matrix of the linear system.
- After interpolation of the nodal water depth with the aide of the bathymetric mesh data, the finite element mesh is set up.
- The change of reflection coefficients along the shorelines and the boundary of coastal structures is also available.
- The mesh generation routine, if necessary, creates the nonlinear finite element mesh, i.e., 6-nodes triangular elements, in specified regions.

Finite element solution routine

- The finite element method in terms of a new infinite element (Tsutsui, 1999, 2000) is implemented, the numerical results of which have the same accuracy with that by the coupling method (Chen & MEi, 1975; Mei, 1983).
- In order to save the computer storage, the algorithm of the finite element solution routine has been revised so that nonzero elements in the coefficient matrix are stored.

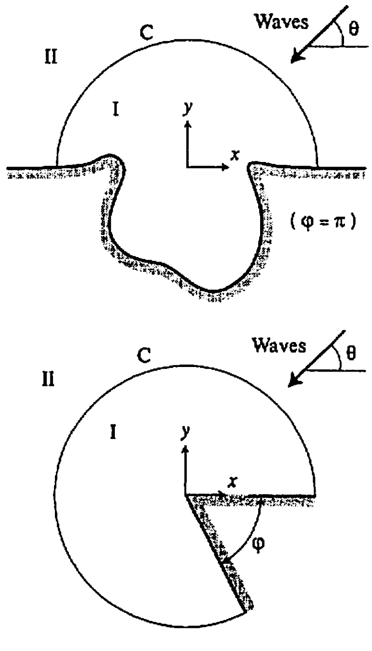
Graphics routine

- Three types of graphics are available, i.e., the plane view, the cross section and the bird's-eye view (perspective projection) both for the bathymetric and wave data.
- Two kinds of vector data for graphics are generated. The first vector data are for coding a graphics routine by the user. As an example, there is a viewer, CATFIG, written in the C-language and works with Mac OS[†]. The second vector data are written in the PostScript[††] language (Kohlet, 1997) to get graphics with high-quality.
- The procedure for cross section, however, generates only the data available for commercial-base software.

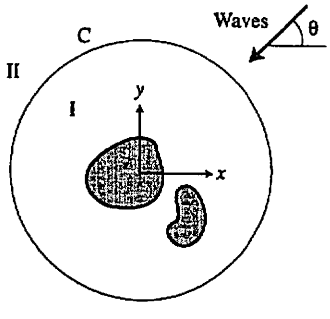
2. Background and Structure

2.1 Overview

The CATWAVES implements the finite element approximation for surface gravity wave motion governed by the mild-slope equation (Berkoff, 1972) for linear waves. Furthermore, the mathematical model implemented for nonlinear wave analysis (Tsutsui, et al., 1996, 1998) combines the full linear dispersion with weak nonlinearity, i.e., it is a variant of the mild-slope equation with weak nonlinear terms (See Appendix).



(1) Coastal harbor model (0 ≤ φ ≤ π).



(2) Islands model.

Figure 1. Two types of ocean topography handled by CATWAVES.

These model equations are capable of simulating wave refraction, diffraction and reflection for two types of ocean topography, as shown in Figure 1, i.e.,

- waves approaching a coastal harbor and
- waves approaching islands in the open ocean.

Additionally, the CATWAVES can handle wave motion in a region where the discontinuity of water depth occurs. Therefore, the boundary conditions handled in the model are:

- wave reflection along the shorelines and the boundary of coastal structures,
- water depth discontinuity, such as step-like changes in reef coasts, and
- the Sommerfeld radiation condition at infinity.

For linear and nonlinear wave analyses, the numerical method in terms of the infinite element for the exterior region (Tsutsui, 1999, 2000) is newly developed to handle the boundary condition at infinity. As the results, the finite element methodology, in this case, governs all the regions of the problem. The coupling method with the exterior series solution (Chen and Mei, 1975; Mei, 1983)

[†] Mac OS is the Macintosh operation system of Apple Computer, Inc.

[††] PostScript is a trademark of Adobe Systems, Inc.

is also applied only to the linear wave analysis.

Note that phenomena of wave breaking of progressive waves near the shorelines and of standing wave-type breaking in front of the coastal structures are not considered because the decision of breaker zones and/or breaking points is at this time very difficult. If possible, the treatment of wave breaking is straightforward (Tsutsui, 1993; Sulaiman, et al., 1994).

2.2 Background to the Numerical Model

The essential requirements for the finite element analysis in CATWAVES are:

- For the harbor model, there must be a straight coastline in the far field, along which the positive x-axis being set up, and the angle φ formed by the two straight coastlines in the far field is within the range from 0 to π, as shown in Figure 1 (1).
- Therefore, the CATWAVES can handle the unbounded domain where there exists a semi-infinite breakwater or a wedge-shaped coast, such as results of reclamation.
- All the bathymetric changes must be located near the coordinate origin, as shown in Figure 1 (1).
- For the islands model, similarly, islands must be located near the coordinate origin, as shown in Figure 1 (2).
- The wave field should be divided into two regions - (I) an interior region and (II) an exterior region - by the arc or the circle denoted by C in Figure 1.
- The coastal topography and the water depth must change only in the interior region (I) that is the wave field of the finite element model.
- On the contrary, the water depth in the exterior region (II) is assumed to be constant, where the infinite element is applied and/or the series solution of the coupling method is formulated.
- Two kinds of the method, therefore, are adopted to impose the boundary condition at infinity.
  (1) The interior and exterior regions are totally dealt with the finite and infinite elements.
  (2) The interior finite element solution and the exterior series solution are coupled on the temporary boundary C.

In appendix, the model equation of wave motion and the sequence of equations to be solved are described briefly. The user of CATWAVES is assumed to be familiar with the model equations, such as the mild-slope equation, and the finite element methodology (Zienkiewicz, et al., 1977, 1989). As such, no more explanation of either is included here.

2.3 Model Structure

The structure of CATWAVES is composed of three parts;

(I) mesh generation,
(II) finite element solution and
(III) graphics routines.

All the programs are coded in Fortran and there is no machine-depended statement/command. The flow of procedures and input/output (I/O) data files are shown in Figure 2.

- The user is required to create two kinds of initial data files: "depth0.&&&" for Routine 1 and "gmesh1.&&&" for Routine 4-1, where the characters "&&&" denote the part of the file name defined by the user.
- For this purpose, you should prepare data for line segments along the coastlines and depth contours by using a preprocessor.
- The programs coded in Fortran in each routine are defined as the routine name plus "m.f" for the main program and ".f" for the subprogram.
- Since the data files with ".&&&" except for the two initial data files will automatically be created in each routine, use them as the input data files in the succeeding process.
- Internal OPEN/CLOSE statements are used for the definition of I/O data files and hence the logical unit numbers (31, 32, ..., 91, 92, ...) used in the routines must be fixed.
- The I/O data file names shown in the parentheses, "depth0", "depth1", "depth2", "gmesh1", "gmesh2", "gmesh3", "waves2", "waves3", "ray", "cont", "cross" and "cset", are arbitrary but they might be fixed for convenience of introducing all the procedures with the aide of an example.
- All the routines work interactively so that you can input necessary data, such as I/O date file names and variables, according to the instructions appearing on the screen.
- Considering the problem concerned, however, you should specify the dimensional parameters that declare the array's dimensional size in the main program.
- Routines necessary for the nonlinear wave analysis are shown in the angle brackets in Figure 2.

(I) Mesh Generation Routine

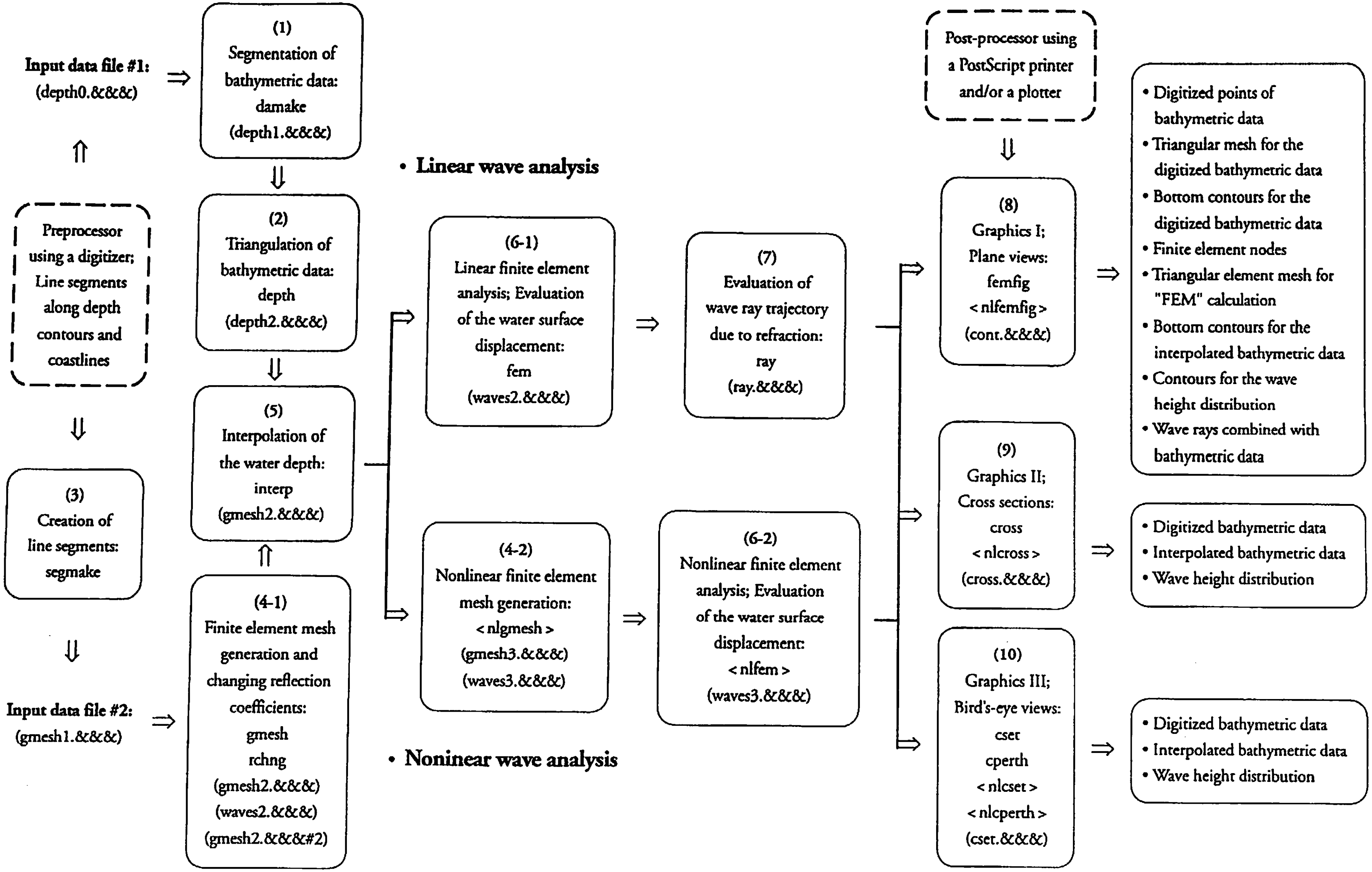
In order to easily prepare the bathymetric data and create the finite element mesh over the model region, the CATWAVES incorporates with an automatic triangular element mesh generation system (e.g., Lo, 1985).

For the linear wave analysis

- There are two kinds of triangular mesh for bathymetric and finite element regions.
- Bathymetric data are set up on the triangular element mesh by Routines 1 and 2, "damake" and "depth".
- Routine "segmake" in (3) creates line segments useful for the mesh generation data file "gmesh1.&&&".
- The triangular finite element mesh for the wave field is generated by Routine "gmesh" in (4-1).
- If necessary, the reflection coefficients along the shorelines and the boundary of coastal structures can be changed with the aide of Routine "rchng" in (4-1).
- The water depth at each triangular element node is linearly interpolated, by Routine "interp" in (5), from the digitized bathymetric data already saved in the file "depth2.&&&".

For the nonlinear wave analysis

- According to the user's request, Routine "gmesh" in (4-1) generates the new file "gmesh2.&&&#2" that has information



(I) Mesh generation and (II) Finite element solution routines. (III) Graphics routine. Treasury.

Figure 2. Structure of CATWAVES.

of the generated linear element mesh and is used to create the nonlinear finite element mesh.

- This is Routine "nlgmesh" in (4.2) that generates the 6-nodes triangular element mesh in the specified region for the nonlinear wave analysis.

(II) Finite Element Solution Routine

There are two main routines for the wave analysis:

- Routines "fem" and "nlfem" in (6) for linear and nonlinear wave analyse are the main components of CATWAVES and evaluate the water surface displacement at all the nodes of triangular mesh.
- Based on the digitized bathymetry, Routine "ray" in (7) evaluates the wave ray trajectory due to refraction, the results of which can be combined with bathymetric data in graphics routines.

(III) Graphics Routine

In order to visualize all the numerical results, vector data both for bathymetry and the wave height are generated with the post-processors, Routines 8, 9 and 10. The routines with "nl" in their names are for the numerical results of the nonlinear wave analysis.

- For plane views, Routines "femfig" and "nlfemfig" in (8) generate vector data describing the digitized points, finite element nodes, triangular element mesh, contours for bathymetry and wave height distribution, and wave rays.
- Data for the cross section of bathymetry and wave height distribution are created by Routines "cross" and "nlcross" in (9).
- Routines "cset", "cperth", "nlcset" and "nlcperth" in (10) are for getting bird's-eye views (perspective projection).

There are two kinds of vector data generated by the graphics routines 8 and 10.

- The first data are for convenience of providing for a plotting routine. A viewer for graphics, CATFIG, is presented. This is written in the C-language and works with Mac OS.
- The second data are written in the PostScript language, which should be handled by commercial-base graphics software and produce hi-quality graphics.

3. General Rules for Mesh Generation

3.1 Overview

Since the system of triangular element mesh generation plays a significant role in handling bathymetric data and finite element grids, the general rules behind mesh generation is described. With the aide of segmentation of coastlines and depth contours, the finite element mesh in CATWAVES consists of the typical five line segments, including three particular mesh patterns, and the central mesh.

3.2 Segmentation Processes

1. Divide, in general, the bathymetry and the wave field of the finite element model into several subregions by line segments, as shown in Figure 3. This process is normally essential because of the following reasons.

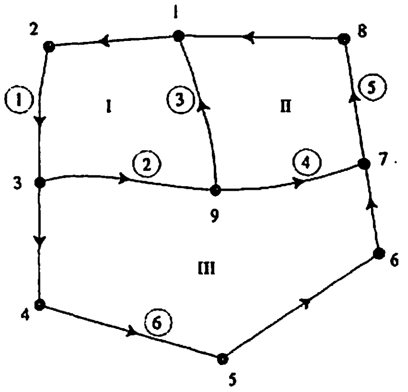


Figure 3. Division of the wave field into three subregions.

For bathymetric data

- In digitizing bathymetry, the data spacing depends on the density of depth contours and, therefore, you should digitize the shorelines and depth contours so as to get their smooth lines in graphics.

For the finite element mesh

- The computational accuracy of the water surface displacement is directly affected by the element mesh size that must be decided according to the local wave length. For example, the mesh size recommended is about 1/15-1/20 of the local wave length. Therefore, you should vary the element mesh size locally.
- Dividing the wave field such that there are nearly the same number of nodes in each subregion, you can save computing time.

2. The line segments are specified by the beginning and ending nodes, defined by the intersection of these line segments and by additional nodes.
3. Number all the nodes, line segments and subregions. For example, in Figure 3, the wave field is divided into three subregions (I, II and III) by six line segments, (1)-(6), with the beginning and ending nodes 1, 3, 7 and 9.

   Notes:

   - The important premise in the numbering of nodes is that the nodes along line segments for the exterior boundary should be numbered anti-clockwise. For example, the line segment (1) is defined by the nodes 1-2-3.
   - This means that, by following the boundary line segment, the particular subregion of interest, e.g., the region I in Figure 3, is always on the left-hand side.
   - An exception to this rule is numbering of nodes on the interior common line segments for the subregions, i.e., (2), (3) and (4) in Figure 3. Since unidirectional numbering both for the right and left subregions is impossible, the direction of numbering is arbitrary only in this case.

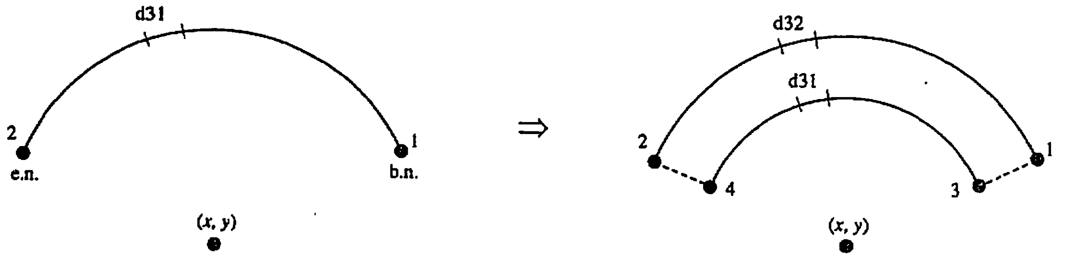
4. Define each subregion as a region surrounded by a set of these line segments. It is important that the first line segment be a line on which the nodes are numbered anti-clockwise according

Table 1. Five types of line segments and data structure.

Types	Index (is)	Data structure etc. (See Figure 4)
Straight line	1*	Beginning and ending nodes, (if necessary, intermediate nodes), spacing on a line.
Arbitrary curve	2*	Beginning, intermediate and ending nodes.
Circular arc	3*	Beginning and ending nodes, center coordinate, spacing on an arc. (A pair of arcs is made by two arcs between which the mesh is generated, as shown in Figure 5.)
Sharp corner	4*	Beginning, center and ending nodes, radius of an outer arc, spacing on an inner and outer arcs. (The refine mesh is progressively generated around a sharp corner, as shown in Figure 6.)
Fixed point	5*	Center node, radius of an outer circle, spacing on an inner and outer arcs. (The radiatory mesh is generated from a fixed point, as shown in Figure 7.)

(1) Straight line.

(2) Arbitrary curve.



(3) Circular arc.

(3') Pair of arcs.

(4) Sharp corner.

(5) Fixed point.

Figure 4. Five types of line segments used in the finite element mesh generation routine.

to the premise 3 above. For example, if the nodes are numbered in the direction of the arrows in Figure 3, the subregion II should be defined as (4)-(5)-(3) or (5)-(3)-(4).

5. If bathymetric discontinuity exists, such as step-like discontinuity of the water depth, you should divide the wave field into subregions by the line segments along discontinuity and number the subregions from deeper to shallower ones.

3.3 Five Types of Line Segments

Five types of line segments, as listed in Table 1 and shown in

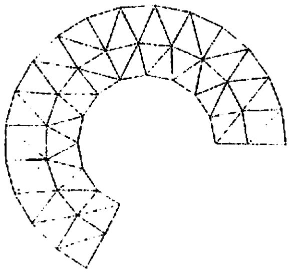


Figure 5. Pair of arcs between which the mesh is generated.

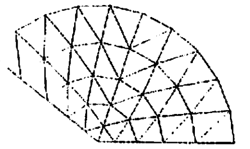


Figure 6. Mesh generation around a sharp coner.

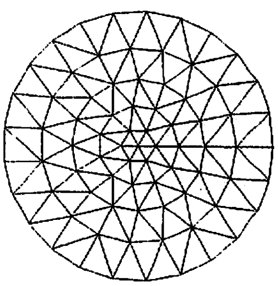


Figure 7. Mesh generation about a fixed point.

Figure 4, are used in the finite element mesh generation routine. The index "is", a two-digit integer number, is used to define the type and the physical characteristic of each line segment. The type of a line segment is classified by the number from 1 to 5, as defined in Table 1. The first row of "is", * in Table 1, is an indicator specifying the physical characteristic of a line segment. The following rules are defined:

1. Straight lines, arbitrary curves, circular arcs and sharp corners (in Figure 4 (4), lines 1-2-3 only) may be real boundaries, such as coastlines. In this case, let the first row of index "is" be "0" and specify the reflection coefficient along them.
2. If bathymetric discontinuity exists, let the first row of index "is" be "1".
3. For the other cases, the first row should be a number except for "0" and "1".
4. The spacing, d11, d31, ..., is used to decide the additional intermediate nodes on each line segment.

3.4 Three Kinds of Particular Mesh

Figures 5-7 show particular mesh generated between a pair of arcs, around a sharp corner and about a fixed point.

3.5 Central Mesh Generation

In the central subregion except for three kinds of particular mesh given in Figures 5-7, the equilateral triangular mesh is generated as much as possible. The reason is that, in the finite element analysis, the numerical error for the equilateral triangular element is the minimum among those for all types of triangles with the same size.

4. Procedures in CATWAVES

4.1 Overview

The best way to understand how to use CATWAVES is to examine it with the aide of an example. This chapter is intended to introduce all the procedures according to Figure 2, and describe the I/O data files, variables, dimensional declarators and the data form in each routine.

The frame for introducing each routine in CATWAVES is as follows:

1. Main routine: Name of the main program.
2. Subroutine: Name of the subprogram.
3. I/O data files: Name of I/O data files and their contents.
4. Variable: Name of variables in the routine.
5. Dimensional declarators: Dimensional size of the array and its meaning.

The user of CATWAVES should specify the I/O date files and, if necessary, the variables according to the instructions appearing on the screen. The dimensional declarator should be defined by the PARAMETER statement in the main program. The termination with error messages will be caused when the dimensional space is not enough. In all the routines the spacial and time units are meter and second, respectively.

4.2 Procedures

■ Routine 1: Segmentation of Bathymetric Data

Purpose

This process is preparation for setting up the bathymetric data and therefore collects raw bathymetric data for each subregion, along the shorelines and depth contours.

Segmentation Processes

You have to digitized bathymetric data along the shorelines and depth contours in the following way:

1. Divide the wave field into several subregions.
2. Number all nodes. At this point, the nodes are intersections of line segments and other necessary points.
3. Number all line segments and subregions.
4. Define line segments.
5. Define subregions by a set of these line segments.
6. Digitize depth data on line segments with suitable spacing.
7. Digitize interior depth data in each subregion with spacing suitable for interpolation of depth contours.
8. Save data for line segments and subregions in "file1" and save interior depth data in "pfile", according to the bathymetric data form, defined in Tables 2 and 3.

Procedure

1. Main routine: damakem.f

Table 2. Form of the file1 = "depth0.&&&".

Variables	Format	Comment etc.
*** comment line ***		
file name = "depth0.&&&&&"		file1
`number of subregions ..... nregn =i5`		nregn
`     "    segments ....... nsegm =i5`		nsegm
*** comment line ***		*** construction of subregions (iteration of "nregn" times)
nss, sz	i5, f15.5	Number of line segments surrounding each subregion, average spacing for interior nodes. (iteration of "nss" times)
[ntr	10i5	Line segment number.
*** comment line ***		*** line segments (iteration of "nsegm" times)
is, rmin	i5, f15.5	Type of a line segment, minimum spacing between nodes. 0: Real boundary. 1: Discontinuity. Others: Temporary boundary.
[jj, xp, yp, hp	i5, 3f15.5	(iteration) If "is" not equals 1: Dummy No., (x, y)-coordinate, water depth. (to finish, xp, yp, hp = 99999.0)
jj, xp, yp, idp	i5, 2f15.5, i5	If "is" equals 1: Dummy No., (x, y)-coordinate, degree of overlapping of discontinuity. (iteration of "idp" times)
[nrg, whh	5x, 4(i5, f10.5)	Region No., water depth. (to finish, xp, yp = 99999.0)

Table 3. Form of the pfiles = "depth01.&&&", "depth02.&&&", ...

Variables	Format	Comment etc.
*** comment line ***		
*** comment line ***		
[xp, yp, hp	5x, 3f15.5	(iteration) (x, y)-coordinate, water depth. (to finish, xp, yp, hp = 99999.0)

2. Subroutine: damake.f
3. I/O data files
   - file1 = "depth0.&&&": Input file where data for line segments and subregions have already been stored.
   - pfile = "depth01.&&&", "depth02.&&&", ... : Input file(s) where digitized interior bathymetric data in each subregion have already been stored.
   - file2 = "depth1.&&&": Output file where the digitized points of bathymetric data will be stored.
4. Variable

nfend	Number of pfile(s).

5. Dimensional declarators

mr	Maximum number of subregions.
ms	Maximum number of line segments.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.

Since, in general, interior depth data along the depth contour lines and necessary points could be stored separately in some files, this routine ask the user to input the number of "pfiles" and the file names.

The forms of input data files are defined in Tables 2 and 3. Notice that:

- The bracket means the range of iteration and the iteration count is in the comment column.
- The line segments surrounding each subregion must be closed.
- For the bathymetric data, the type of line segments is an arbitrary line, hence the index "is" is a single-digit integer number, i.e., "0", "1", ..., only to define physical characteristics (Refer to 3.3).
- Data on line segments are selected by checking whether the distance of adjacent nodes is greater than "rmin".
- For a bathymetric point necessary for segmentation, set the dummy number jj be "99999" and therefore the point is selected even if the distance is less than "rmin".
- Similarly, interior points for each subregion are selected according to the spacing "sz". This routine excludes the unnecessary points that do not belong to any subregion.

▣ Routine 2: Triangulation of Bathymetric Data

Purpose

This routine generates the triangular mesh for digitized bathymetric nodes. Details of bathymetry are finally furnished on the set of triangular elements.

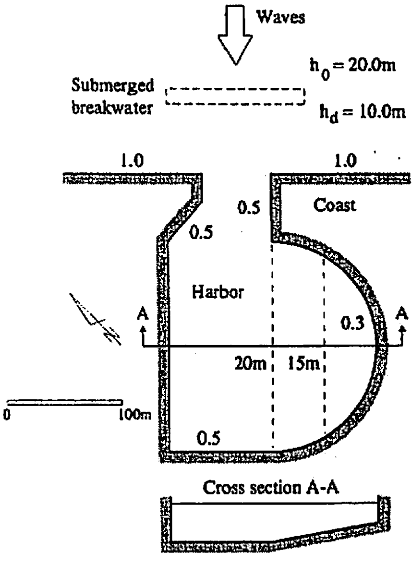
Procedure

1. Main routine: depthm.f
2. Subroutine: depth.f
3. I/O data files
   - file1 = "depth1.&&&": Input file where the digitized points of bathymetric data have already been stored.
   - file2 = "depth2.&&&": Output file where, after triangulation, details of bathymetric data will be stored.
4. Dimensional declarators

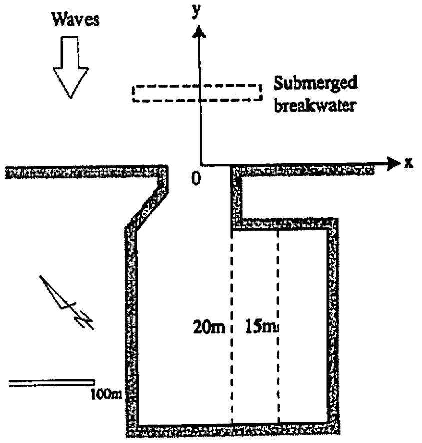
mr	Maximum number of subregions.
ms	Maximum number of line segments.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.

Example of a Harbor Model

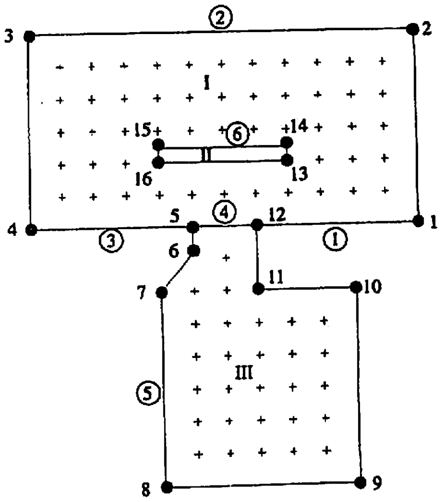
Here, consider an example of an artificial harbor model with a submerged breakwater in front of the harbor mouth, as shown in Figure 8 (1). In the semicircular region of the harbor, the water depth decreases uniformly from 20m to 10m, as shown in the cross section A-A. There is discontinuity along the submerged breakwater where the water depth changes abruptly from 20m to 10m. For simplicity, let the water depth in the remaining region be constant, say, 20m. It is assumed that there is the vertical wall along the shorelines. The reflection coefficients along shorelines, as shown in Figure 8 (1), are assumed to be 1.0, 0.5 and 0.3. Figure 8 (2) shows the coordinate system and the shape of coast-



(1) Artificial harbour.



(2) Coordinate system and bathymetry.



(3) Segmentation of bathymetric data.

Figure 8. An artificial harbor model.

lines for bathymetry that covers the harbor model to interpolate the nodal water depth.

The bathymetry is divided into three subregions, as shown in Figure 8 (3):

(I) the offshore region,
(II) the region on the submerged breakwater and
(III) the region in the harbor.

The nodes and line segments are set up. The cross in all the subregions denotes internal points that assumed to be on the depth contour lines. Notice that, in this example, the wave field on the submerged breakwater is subdivided along the depth discontinuous line (6).

■ Routine 3: Creation of Line Segments

Purpose

This routine creates line segments (straight and arbitrary lines) with desired spacing. The results can be used in the input data file, "gmesh1.&&&&", for finite element mesh generation. Notes that:

- You have to use the coastlines and particular depth contour lines with suitable spacing as the line segments for finite element mesh generation.
- The graphics routine "cross" helps you to generate raw bathymetric data useful for this purpose (Refer to Notes in Routine 9: *Graphics II - Cross Sections*).

Procedure

1. Main routine: segmakem.f
2. Subroutine: segmake.f
3. I/O data files
    - file1 = "&&&&&&": Input file where digitized data for line segments have already been stored.
    - file2 = "&&&&&&": Output file where data for line segments with desired spacing will be stored.
4. Variables

nss	Number of line segments.
nrr	Number of spacing.
rmin	Minimum spacing on each line segment.

5. Dimensional declarator

mr	Maximum number of subregions.

Table 4. Form of the file1 = "&&&&&&".

Variables	Format	Comment etc.
*** comment line ***		
[jj, xp, yp, title	i5, 2f15.5, a	(iterantion of "nss" times) Dummy No., (x, y)-coordinate, comment. (to finish xp, yp = 99999.0)

The form of input data for creation of line segments is defined in Table 4. The bracket means the range of iteration and the iteration count is in the comment column.

■ Routine 4: Finite Element Mesh Generation and Changing Reflection Coefficients

Purpose

The procedure presents two kinds of routine for generating the linear and nonlinear finite element mesh and also the routine for changing reflection coefficients along the shorelines and the boundary of coastal structures.

(1) Finite Element Mesh Generation

This routine generates the triangular element mesh over the wave field of the finite element model and provides you with capabilities of:

(a) changing the element mesh size in each subregion and
(b) making the variable-band width of the coefficient matrix of the system of linear equations small.

Remark that:

- The first demand (a) is realistic and, to do so, the wave field must be divided into subregions.
- The second one (b) comes from computational problems. Since the coefficient matrix of the system is sparse, the smaller the variable-band width of the coefficient matrix, saves the computer storage and, as a result, computing time. This task is accomplished by the procedure: *renumbering of node numbers* included in CATWAVES.
- The algorithm of renumbering of node numbers is similar to a widely-used Cuthill & McKee (1969) ordering, to reduce the variable-band width.
- Details of the generated finite element mesh are prepared for interpolation of the nodal water depth in Routine 5.

Segmentation Processes

As described in Chapter 3: *General Rules for Mesh Generation*, the segmentation processes for finite element mesh generation are:

1. Divide the wave field into several subregions.
2. Number all nodes. At this point, the nodes are intersections of line segments and other necessary points.
3. Number all line segments and subregions.
4. Define line segments.
5. Define subregions by a set of these line segments.
6. Decide spacing on line segments.
7. Decide elements mesh sizes required for subregions.
8. After preparation, save data in "file1", according to the data form of the finite element mesh, defined in Table 5.

Reflection Coefficients

Among the five line segments, straight lines, arbitrary curves, circular arcs and sharp corners may be boundaries along which you have to specify reflection coefficients, according to the following rules:

1. Reflection coefficients at all boundary nodes should be given in advance.
2. Intermediate nodes are automatically created for straight lines, circular arcs and sharp corners.
3. Reflection coefficients at these additional nodes on the boundaries are evaluated as the same value of the former node, e.g., in Figure 4,
   - for a straight line:
     the value at the node 1 along the line 1-2,
     the value at the node 2 along the line 2-3,
     the value at the node 3 along the line 3-4,
   - for a circular arc: the value at the node 1,
   - for a sharp corner: the value at the node 2.
4. For arbitrary curves, you should input reflection coefficients at all the nodes. However, if some of the reflection coefficients are successively the same, it is enough to input the value only at the beginning node (Refer to Table 5).

(4.1) Procedure for Linear Mesh Generation

1. Main routine: gmeshm.f
2. Subroutine: gmesh.f
3. I/O data files
   - file1 = "gmesh1.&&&": Input file where data for line segments and subregions have already been stored.
   - file2 = "gmesh2.&&&": Output file where details of the generated linear element mesh will be stored.
   - file3 = "waves2.&&&": Output file where details of the reflection coefficient will be stored.
   - file4 = "gmesh2.&&&#2": Output file where detail information of the generated linear element mesh will be stored. This file will be created according to the user's request to generate nonlinear finite element mesh.
4. Variables

h0	Constant water depth in the exterior region (in meters).
irenum	Parameter for renumbering of node numbers; 0: No renumbering. 1: Renumbering to reduce the variable-band width of the coefficient matrix of the linear system.
imesh	Parameter for output the file4 = "gmesh2.&&&#2"; 0: No output (default). 1: Output.

5. Dimensional declarators

mr	Maximum number of subregions.
ms	Maximum number of line segments.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.
mk	Maximum number of nodes on straight lines and curves.
mi	Maximum number of layers in the three kinds of particularly generated mesh, i.e., a pair of arcs, a sharp corner and a fixed point.

The form of input data for finite element mesh generation is defined in Table 5. Note that:

- The bracket means the range of iteration and the iteration count is in the comment column.
- Exclude the number of subregions defined by "pair of arcs", "sharp corner" and "fixed point" from the input number of subregions, "nsubr".
- Similarly, exclude the number of nodes on arbitrary curves from the input numbers of nodes, "ninpt", "nbinpt" and "ndinpt".
- In order to satisfy the assumption in Chapter 1: *Background and Structure* for the finite element method, the outer segment that separates the interior region from the exterior region should be:
  (1) a semi-ring for a harbor model or a ring for an islands model, and
  (2) the first line segment should be an arc or a circle.
- To obtain a circle, let the beginning and ending nodes of an arc be the same node.
- The line segments surrounding each subregion must be closed.

(4.2) Procedure for Nonlinear Mesh Generation

Before running this routine, the user should prepare the mesh and wave data files, i.e., "gmesh2.&&&", "waves2.&&&" and "gmesh2.&&&#2", by the linear mesh generation routine "gmesh" in (4-1) and interpolate the nodal water depth in terms of Routine "interp" in (5).

1. Main routine: nlgmeshm.f
2. Subroutine: nlgmesh.f
3. I/O data files
   - file1 = "gmesh2.&&&": Input file where details of the generated linear element mesh and the nodal water depth have already been stored.
   - file2 = "gmesh2.&&&#2": Input file where detail information of the generated linear element mesh have already been stored.
   - file3 = "waves2.&&&": Input file where details of the reflection coefficient for the linear wave analysis have already been stored.
   - file4 = "gmesh3.&&&": Output file where details of the generated element mesh and the nodal water depth for the nonlinear wave analysis will be stored.
   - file5 = "waves3.&&&": Output file where details of the reflection coefficient for the nonlinear wave analysis will be stored.
4. Variables

hc	Critical water depth (in meters); the region with the water depth shallower than this value is assumed to be the region of the nonlinear wave analysis.

Table 5. Form of the file1 = "gmesh1.&&&".

Variables	Format	Comment etc.
*** comment line ***		
file name = "gmesh1.&&&&&&"		file1
number of subregions .... nsubr =i5		nsubr
" input nodes .... ninpt =i5		ninpt
" input boundary nodes .... nbinpt =i5		nbinpt
" input discontinuity nodes .. ndinpt =i5		ndinpt
" segments .... nsegm =i5		nsegm
" pairs of arcs .... nparc =i5		nparc
" sharp corners .... nscor =i5		nscor
" fixed points .... nfpts =i5		nfpts
*** comment line ***		(iteration of "ninpt" times)
[jj, wx, wy	i5, 2f15.5	Dummy No., (x, y)-coordinate.
*** comment line		If nbinpt > 0: (iteration of "nbinpt" times)
[jj, ibw, wrc	2i5, f10.5	Dummy No., boundary node, reflection coefficient.
*** comment line ***		If ndinpt > 0: (iteration of "ndinpt" times)
[jj, idw	2i5	Dummy No., discontinuity node.
*** comment line ***		If nparc > 0: (iteration of "nparc" times)
[kparc1, kparc2	2i5	Line segment numbers of a pair of arcs.
*** comment line ***		*** construction of subregions
		(iteration of "nsubr" times)
nss, sz	i5, f15.5	Number of line segments surrounding each subregion, element mesh size.
		(iteration of "nss" times)
[ncr	10i5	Line segment number.
		*** line segments (iteration of "nsegm" times)
*** (1) straight lines ***		*** comment line
is	i5	Type of a line segment.
ne1, dl1	i5, f10.5	Number of nodes, spacing on a line.
		(iteration of "ne1" times)
[nd1	10i5	Node number.
*** (2) arbitrary curves ***		*** comment line
is	i5	Type of a line segment.
nd21	i5	Beginning node.
		(iteration)
[jj, xp, yp, rc	i5, 3f15.5	Dummy No., (x, y)-coordinate, (if necessary, reflection coefficient). (to finish, xp, yp = 99999.0)
nd22	i5	Ending node.
*** (3) circular arcs ***		*** comment line
is	i5	Type of a line segment.
nd31, nd32, x30, y30, d31	2i5, 3f10.5	Beginning and ending nodes, center coordinate, spacing on an arc.
*** (4) sharp corners ***		*** comment line
is	i5	Type of a line segment
nd41, nd42, nd43, r4, d41, d42	3i5, 3f10.5	Beginning, center and ending nodes, radius of an outer arc, spacing on an inner and outer arcs.
*** (5) fixed points ***		*** comment line
is	i5	Type of a line segment.
nd5, r5, d51, d52	i5, 3f10.5	Center node, radius of an outer circle, spacing on an inner and outer circles.

5. Dimensional declarators

mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.

(2) Changing Reflection Coefficients Along Boundaries

After generation of the finite element mesh, Routine "rchng" in (4-1) can change the reflection coefficients along the shorelines and the boundary of the coastal structures. Notice however that to modify the reflection coefficients in the file "waves3.&&&" for nonlinear wave analysis, you should make new file "waves2.&&&" by using this routine and later create the nonlinear finite element mesh with the aide of Routine "nlgmesh" in (4-2).

Procedure

1. Main routine: rchngm.f
2. Subroutine: rchng.f
3. I/O data files
   - file1 = "gmesh1.&&&": Input file where data for line segments and subregions have already been stored. New reflection coefficients to be modified should be specified in this file.
   - file2 = "gmesh2.&&&": Input file where details of the generated linear element mesh have already been stored.
   - file3 = "waves2.&&&": Input file where details of the reflection coefficient have already been stored.
   - file4 = "waves2.&&&": Output file where details of the modified reflection coefficient will now be stored.
4. Dimensional declarators

ms	Maximum number of line segments.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.
mk	Maximum number of nodes on straight lines and curves.

Note that data for new nodal reflection coefficients desired should be specified in "file1" and the modified reflection coefficients will be stored in the new file "file4".

Example of the Harbor Model

Consider the artificial harbor model, as was shown in Figure 8 (1). Figure 9 shows the result of segmentation of the wave field; the nodes, line segments and subregions being numbered. According to the fourth note in *Procedure for Linear Mesh Generation*, the outer region is the semi-ring defined by a pair of arcs. To illustrate all line segments defined in CATWAVES, there is an arbitrary curve between the nodes 9 and 10, a sharp corner along the shoreline 6-7-8, and a fixed point about the node 13. The others are straight lines.

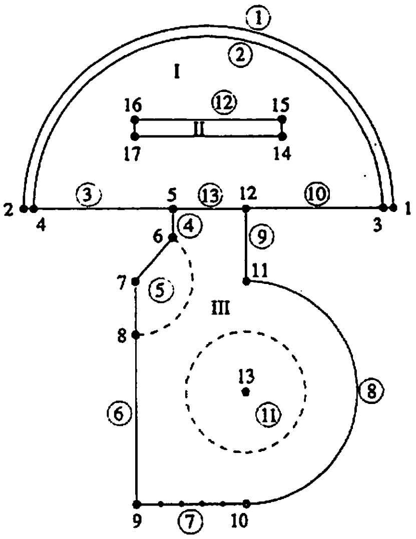


Figure 9. Segmentation of the wave field.

As was seen in Figure 8 (1), the reflection coefficients are assumed to be:

1.0 along lines 2-4-5 and 1-3-12,
0.5 along lines 5-6-7-8-9-10 and 11-12, and
0.3 along the arc 10-11.

Note that the shape of the subregion on the submerged breakwater divided by the depth discontinuous line (12) is identical with the corresponding region of bathymetry in Figure 8 (3).

■ Routine 5: Interpolation of the Water Depth

Purpose

So far, the bathymetric data file "depth2.&&&" and the finite element mesh data file "gmesh2.&&&" have been created. Referring to the bathymetric data, this routine completes the mesh data by linearly interpolating the nodal water depth.

Procedure

1. Main routine: interpm.f
2. Subroutine: interp.f
3. I/O data files
   - file1 = "depth2.&&&": Input file where, after triangulation, details of bathymetric data have already been stored.
   - file2 = "gmesh2.&&&": Input/output file where details of the generated linear element mesh have already been stored. After interpolation, the nodal water depth will be added.
4. Dimensional declarators

mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.

■ Routine 6: Finite Element Analysis - Evaluation of Water Surface Displacement -

Purpose

This routine evaluates the water surface displacement at all the finite element nodes based on the finite element method in terms of the infinite element (Tsutsui, 1999, 2000) and the coupling method (Chen and Mei, 1975; Mei, 1983).

(6.1) Procedure for the Linear Wave Analysis

1. Main routine: femm.f
2. Subroutine: fem.f
3. I/O data files
   - file1 = "gmesh2.&&&": Input file where details of the generated linear element mesh and the nodal water depth have already been stored.
   - file2 = "waves2.&&&": Input/output file where details of the reflection coefficient have already been stored. After calculation, the water surface displacement at all the nodes will be added.
4. Variables

T	Wave period (in seconds).
thi	Incident angle of waves (in degrees), measured anti-clockwise from the positive x-axis, as in Figure 1.

5. Dimensional declarators

mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of nodes on the exterior arc or the exterior circle; This value is an index to the variable-band width of the sparse coefficient matrix of the linear system.
mn3	Maximum number of nonzero elements in the coefficient matrix of the linear system; Since the coefficient matrix is sparse and symmetric, nonzero elements in the right-upper triangular matrix are stored in the one-dimensional array. An estimated value of mn3 is about $0.01\times mn(mn+1)/2$.

6. Solution of the sparse linear systems

Since, in the linear wave analysis, the coefficient matrix of the system of linear algebraic equations is of sparse and symmetric, the so-called skyline method is implemented to solve the system. The numerical results of the water surface displacement, normalized by the incident wave amplitude, are complex values and the wave profile is sinusoidal. So that the nodal wave height relative to the incident wave amplitude is evaluated by taking the absolute value of the water surface displacement.

(6.2) Procedure for the Nonlinear Wave Analysis

1. Main routine: nlfemm.f
2. Subroutine: nlfem.f
3. I/O data files
   - file1 = "gmesh3.&&&": Input file where details of the generated nonlinear element mesh and the nodal water depth have already been stored.
   - file2 = "waves3.&&&": Input/output file where details of the reflection coefficient have already been stored. After nonlinear evaluation, the water surface displacement at all the nodes will be added.
4. Variables

T	Wave period (in seconds).
thi	Incident angle of waves (in degrees), measured anti-clockwise from the positive x-axis, as in Figure 1.
nf	Number of Fourier modes.
a0	Offshore incident wave amplitude (in meters).

5. Dimensional declarators

mnf	Maximum number of Fourier modes.
mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of nodes on the exterior arc or the exterior circle.
mn3	Maximum number of nonzero elements in the coefficient matrix of the linear system; An estimated value of mn3 is about $0.02\times mn(mn+1)/2$.
mn4	Maximum number of unknowns in the final coupled system ($= mnf\times mn$)
mn5	Maximum number of nonzero elements in the coefficient matrix of the final coupled system ($= mnf\times mnf\times mn3$)
mn6	Maximum dimensional size of the one-dimensional array where all the values of entries (fill-in) in the process of Gaussian elimination are stored. Note that the required space, mn6, for LU decomposition is a few times the number of nonzero elements, mn5.

6. Solution of the sparse nonlinear systems

The model equation for nonlinear waves leads, after quasi-linearization of the nonlinear terms, to the system of nonlinear algebraic equations, i.e., the unknown water surface displacement appears in the coefficient matrix of the system. The system is the coupled system of Fourier components specified and then the coefficient matrix is sparse, large and numerically unsymmetric but has the symmetric structure.

There may be some methodologies for solving this nonlinear system. Typical methods are (1) the direct iteration and (2) the numerical integration of the system of ordinary differential equations derived from the original system, in the manner similar to the incremental method in the structural analysis (Zienkiewicz, et al., 1989) (See Appendix). To the method (1), the simple iterative method and the Newton method are available. To the method (2), on the contrary, the predictor-corrector method, the backward Euler method and the Runge-Kutta-type method are available. In all the cases, however, we should solve the unsymmetric system of linear equations with the parameter of the incident wave amplitude.

There exists a number of popular and powerful iterative methods for solving linear systems, e.g., GMRES (Saad, el. al., 1986), CGS (Sonneveld, 1989) and Bi-CGSTAB (van der Vorst, 1992; Tsutsui, et al., 1998c). The numerical values of the coefficient matrix, however, are of complex due to the boundary conditions imposed at infinity and therefore suitable preconditioners for the methods above and convergence properties are unknown.

In this package, therefore, the direct methods for sparse matrices in terms of sparse matrix techniques (Duff, et al., 1986) are worth implementing for the key to solution. Among the public domain software, Howell code MA28 (Duff, 1979) and/or the Y12M (Zlatev, et al., 1981) are recommend to solve the system of sparse unsymmetric linear equations. These are written in Fortran and available from *netlib*[†], although the users obtaining MA28 in this way are still requested to sign a license agreement.

Similar to the linear wave analysis, the numerical results of the water surface displacement, normalized by the incident wave amplitude, are complex values. However, the wave profile is not sinusoidal so that, after calculating the wave profile base on the definition (See Appendix), the nodal wave height is evaluated as the height between the maximum wave crest and the minimum wave trough.

Notice that the routine for the nonlinear wave analysis, "nlfemm", covers the linear wave analysis, too, and therefore the user can work with this routine both for linear and nonlinear wave motion.

■ Routine 7: Evaluation of Wave Ray Trajectory due to Refraction

Purpose

By using the digitized bathymetric data, this routine generates line segments of the wave ray trajectory due to refraction. In order to visualize them combining with bathymetric data, you can use the graphics routine 8: *Graphics I - Plane Views*.

Procedure

1. Main routine: raym.f
2. Subroutine: ray.f
3. I/O data files
   - file1 = "depth2.&&&": Input file where, after triangulation, details of bathymetric data have already been stored.
   - file2 = "ray.&&&": Output file where data for line segments of wave ray trajectory will be stored.
4. Variables

h0	Constant water depth in the exterior region (in meters).
T	Wave period (in seconds).
thi	Incident angle of waves, measured anti-clockwise from the positive x-axis (in degrees).
nray	Number of wave rays.
xr, yr	Starting points of the wave rays in (x, y)-coordinate (in meters).

5. Attribute file for variables
   - Attribute file = "&&&&&": Input file where all the variables have already been stored.
6. Dimensional declarators

mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mr3	Maximum number of wave rays.

Notes:

- If you have not run this routine yet, the attribute file specifying the variables necessary for computation will not exist. However, you can interactively input the variables according to the instructions that appear on the screen, and save them in the specific file.
- The routine traces a wave ray and when the wave reaches to the shoreline, quits computing.

■ Routine 8: Graphics I - Plane Views

Purpose

As the post-processor, the graphics routine plots digitized nodes, finite element nodes, triangular element mesh, contour lines and wave rays.

Options

For ndata = 1; Digitized bathymetric raw data ("depth1.&&&"):
- job = 1: Digitized nodes.

For ndata = 2; Digitized bathymetric data ("depth2.&&&"):
- job = 2: Triangular mesh.
- job = 3: Depth contours.

For ndata = 3; Interpolated bathymetric data and the wave height ("gmesh2.&&&" and "waves2.&&&"):
- job = 4: Finite element nodes.
- job = 5: Triangular element mesh.
- job = 6: Depth contours.
- job = 7: Contours for wave height distribution.

Procedure

1. Main routine: femfigm.f
2. Subroutine: femfig.f
3. I/O data files

(a) For ndata = 1:
- file1 = "depth1.&&&": Input file where the digitized points of bathymetric data have already been stored.
- file4 = "&&&&&": Output file where vector data for figures will be stored.

[†] *netlib* is the site of distribution of mathematical software via electronic mail in the Bell Research Labs.

(b) For ndata = 2:

- file1 = "depth2.&&&": Input file where, after triangulation, details of bathymetric data have already been stored.
- file3 = "ray.&&&": Input file where data for line segments of wave ray trajectory have already been stored.
- file4 = "&&&&&": Output file where vector data for figures will be stored.
- file5 = "cont.&&&": Output file where line segments for contours will be stored and used for bird's-eye views in Routine 10.

(c) For ndata = 3:

- file1 = "gmesh2.&&&": Input file where details of the generated linear element mesh and the nodal water depth have already been stored.
- file2 = "waves2.&&&": Input file where the wave data have already been stored.
- file3 = "ray.&&&": Input file where data for line segments of wave ray trajectory have already been stored.
- file4 = "&&&&&": Output file where vector data for figures will be stored.
- file5 = "cont.&&&": Output file where line segments for contours will be stored and used for bird's-eye views in Routine 10.

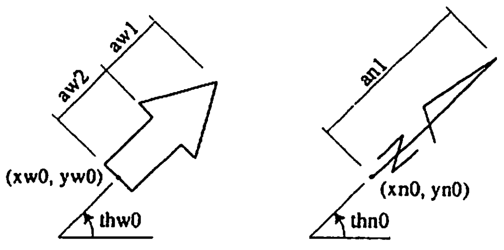
4. Variables

- For general use:

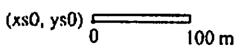
ndata	Type of input data (1, 2 and 3).
job	Option number (1, 2, 3, ..., and 7).
isgn	Index for data output of contours; 0: No output (default). 1: Output when the user wish to use the contour lines of the water depth and/or the wave height distribution for bird's-eye views.
iray	Index for combining wave ray trajectory due to refraction with bathymetric data, only for the cases of ndata = 2, job = 3 and ndata = 3, job = 6; 0: No combining (default). 1: Combining.
ibound	0: No drawing boundaries. 1: Drawing boundaries (default).
ilabel	0: No labeling. 1: Labeling (default).
icoord	0: No drawing the coordinate origin. 1: Drawing the coordinate origin (default).

- For the contours:

mch1	Number of contours for the water depth.
mch2	Number of contours for the wave height.
hight	Values of contours for the water depth (in meters) and wave height.



(1) Wave direction. (2) The north.



(3) Scale.

Figure 10. Definition of the variables for the symbols in the graphics routine I.

- For the symbol of wave direction (See Figure 10 (1)):

iwave	0: No drawing the symbol of wave direction. 1: Drawing the symbol of wave direction (default).
xw0, yw0	Position of the origin of the symbol of wave direction in the (x, y)-coordinate (in meters).
thw0	Angle measured anti-clockwise from the positive x-axis (in degrees).
aw1	Length of an arrow head (in centimeters).
aw2	Length of an arrow (in centimeters).
cthi	Wave direction (characters for the azimuth).

- For the symbol of the north (See Figure 10 (2)):

inorth	0: No drawing the symbol of the north. 1: Drawing the symbol of the north (default).
xn0, yn0	Position of the origin of the symbol of the north in the (x, y)-coordinate (in meters).
thn0	Angle measured anti-clockwise from the positive x-axis (in degrees).
an1	Length of the symbol of the north (in centimeters).

- For the symbol of the scale (See Figure 10 (3)):

iscale	0: No drawing the symbol of the scale. 1: Drawing the symbol of the scale (default).
xs0, ys0	Beginning location of the symbol of the scale in the (x, y)-coordinate (in meters).
nscl	Iteration count of drawing the scale unit.
sunit	Spatial unit of the scale (in a character "m" or "km").

5. Attribute file for variables

- Attribute file = "&&&&&": Input file where all the variables have already been stored.

6. Dimensional declarators

mr	Maximum number of subregions.
ms	Maximum number of line segments.
mm	Maximum number of elements.
mm2	Maximum number of elements in each subregion
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.
mh	Maximum number of contours.
mr3	Maximum number of wave rays.
mn3	Maximum number of nodes on a wave ray.

Notes:

- Refer to Notes in Routine 7 about the attribute file.

Table 6. Form of vector data for plane figures.

Variables	Format and comment etc.
• Drawing line:	
l, lw, lp	'l', 2i3
u, xp, yp, d, xp, yp [, d, xp, yp, ...]	'u', 2f7.3, 'd', 2f7.3 [, 'd',2f7.3, ...]
	lw: Line width lp: Line pattern (fixed to be 0 for solid line) (xp, yp): (x, y)-coordinat "u" denotes pen up and "d", pen down.
• Drawing comment and title	
c, nc, xp, yp, wh, wc, run, rise, slant	'c', i3, 2f7.3, 5f7.2
title	
	nc: Character size (xp, yp): (x, y)-coordinate (wh, wc): Height and width of characters (run, rise): (x, y)-vector of incline of the title slant: Slant of characters title: Characters
• Ending:	
end	'end' End of data

- The variables for the symbols and their origins shown by • are defined in Figure 10.
- For the numerical results of the nonlinear wave analysis, use Routine "nlfemfig" and the I/O data files "gmesh3.&&&" and "waves3.&&&". Variables and dimensional declarators and so on are the same with those in Routine "femfig".

Form of Vector Data

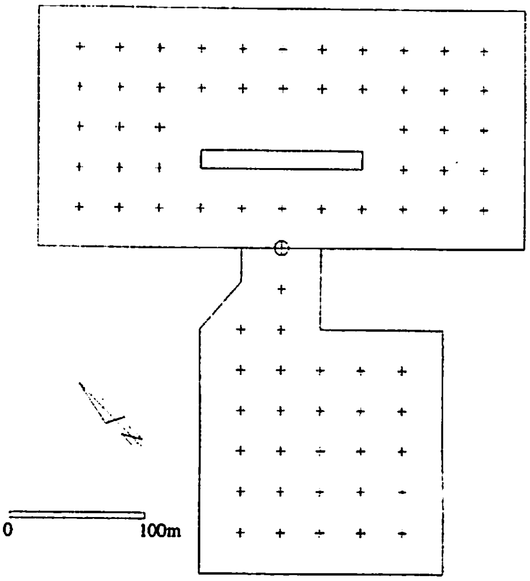
There are two kinds of vector data generated by this routine:

- The first vector data are for convenience of providing for a plotting routine. The form of vector data is defined in Table 6. As was noticed before, the bracket means the range of iteration.
- A viewer, CATFIG, is presented for this data. It is written in the C-language and works with Mac OS.
- The second vector data file, written in the PostScript language, is for commercial-base software that can deal with PS-file. This file is automatically generated and has the characters ".PS" at the end of the vector data file name.

Note that, in the following examples of the three graphics routines, the results for the linear wave analysis are firstly presented and later the results of the nonlinear wave analysis will be shown.

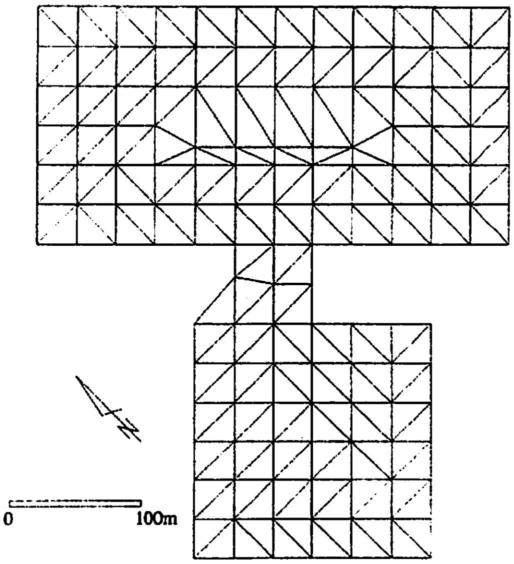
Examples of Graphics I

For the artificial harbor model demonstrated before, you can obtain the following graphic results according to the options.



DIGITIZED NODES
FILE-1 = depth1.dat
Number of nodes: n = 148

Figure 11. Disitized points representing bathymetric nodes (job = 1).



DIGITIZED GRIDS OF WATER DEPTH
FILE-1 = depth2.dat
Number of elements: m = 225
Number of nodes: n = 143

Figure 12. Triangular mesh generated over the digitized bathymetric data (job = 2).

- Job = 1: Figure 11 shows the digitized nodes in case of the parameters ibound = 1, ilabel = 1, and icoord = 1. The line segments showing coastlines and temporary boundaries such as along the submerged breakwater are drawn by the solid line and the interior points, by the cross. The cross with a circle denotes the coordinate origin. Using this figure, you can check whether all the points are digitized with suitable spacing. It is

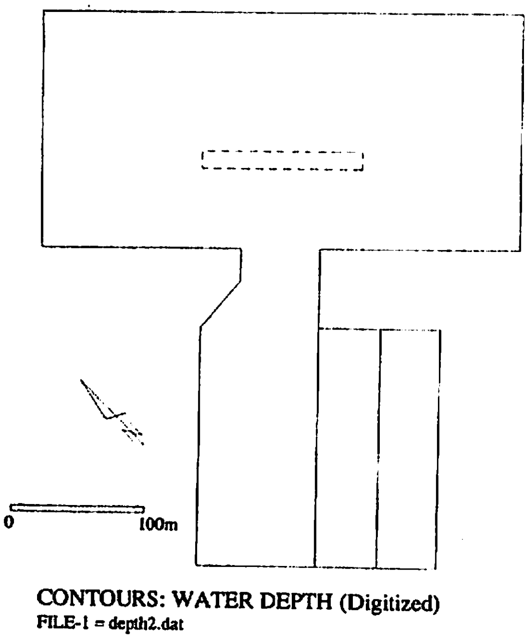


Figure 13. Depth contours generated from the bathymetric data (job = 3).

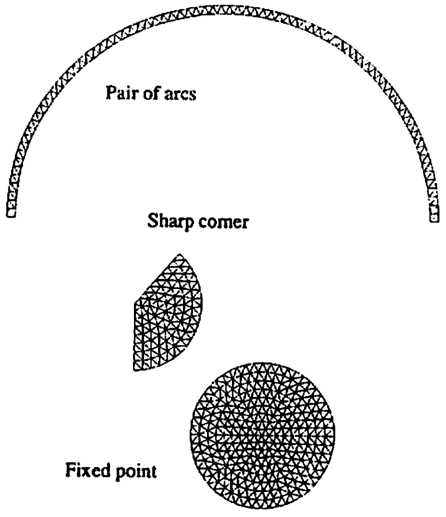


Figure 15. Three kinds of particularly generated mesh in CATWAVES; (1) the pair of arcs, (2) the sharp corner and (3) the fixed point.

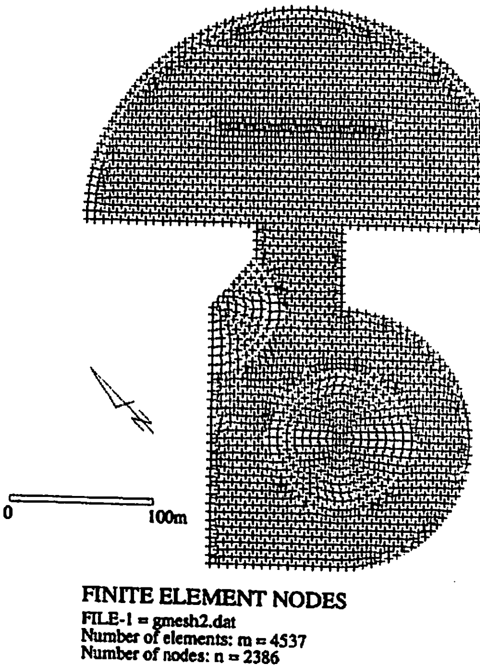


Figure 14. Finite element nodes (job = 4).

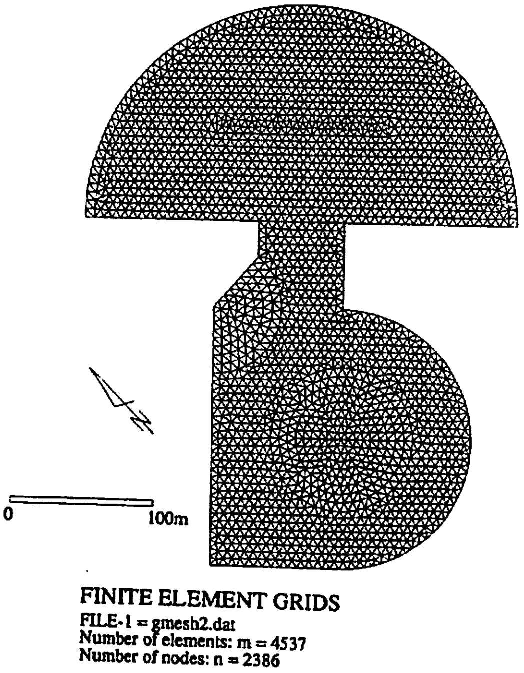


Figure 16. Triangular element mesh for the linear finite element analysis, designed for waves with the period of 10 secs. (job = 5).

noted, comparing with Figure 8 (3), that a few points in front of the submerged breakwater are excluded and then the digitized spacing near the submerged breakwater is unsuitable.

- Job = 2: Figure 12 shows the corresponding triangular mesh that is used to interpolate the nodal water depth.
- Job = 3: The depth contours are shown in Figure 13. Only two contour lines (10, 20m), in this case, are drawn in the harbor because the sea bottom is flat in the other regions.
- Job = 4: Figure 14 shows the finite element nodes indicated by the cross, with capable of checking the nodal spacing.

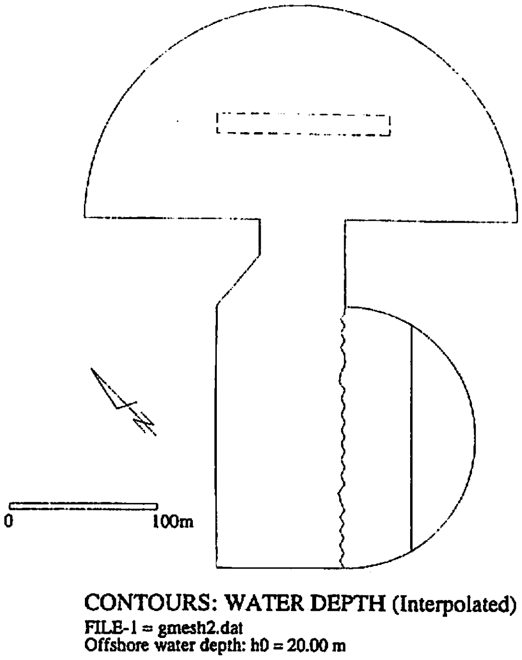
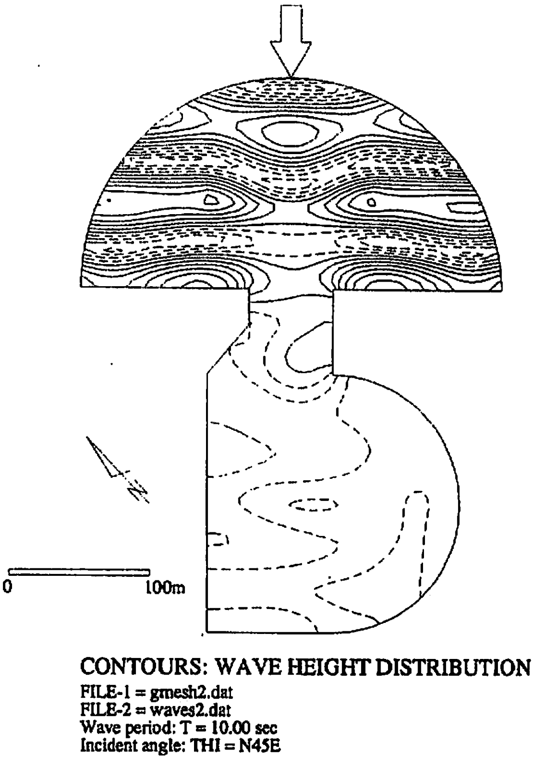
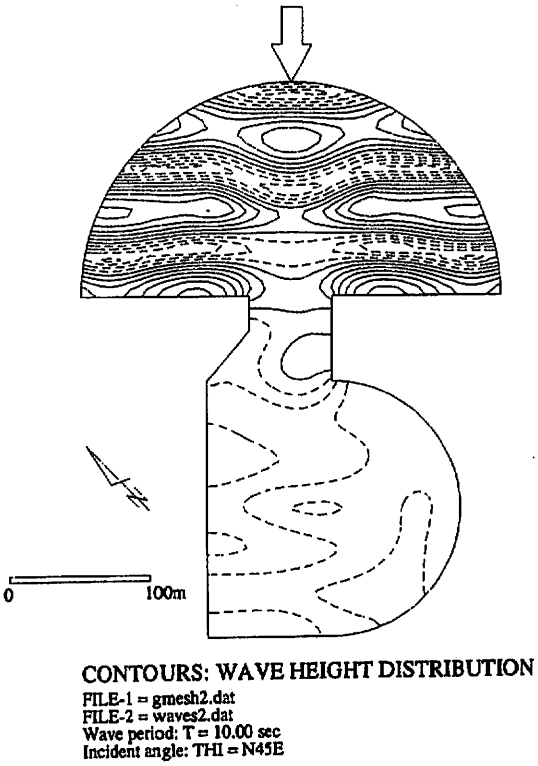


Figure 17. Depth contours for the interpolated bathymetric data (job = 6).

- Job = 5: Figure 15 shows three kinds of particularly generated mesh in this example, defined in Chapter 2: *General Rules for Mesh Generation*. The triangular element mesh for the linear finite element analysis is shown in Figure 16. Note that the central mesh regions I-III in Figure 9, except for three particular mesh shown in Figure 15, are formed by nearly equilateral triangles. This mesh is designed for waves with the period of 10 seconds. The element mesh size is about 1/15-1/18 of the local wave length in the semicircular region of the harbor and 1/20 in the other regions.
- Job = 6: The depth contours for interpolated bathymetric data are shown in Figure 17, where there are two contour lines in the harbor. The contour line for 10m water depth is correct but that for 20m water depth is wrong because it should be a straight line. The reason of obtaining the zigzaz contour comes from the fact that the corresponding region is not subdivided along the depth contours. This is an example of unsuitable subdivision of the bathymetric region of interest. Therefore, this indicates that in order to get correct depth contour lines, the wave field of interest should be subdivided by using the depth contour lines and the shorelines.
- Job = 7: Figure 18 shows the comparison of the contours for the wave height distribution of the linear wave analysis in terms of (1) the infinite element and (2) the coupling method. When the wave height relative to the incident wave height is greater than or equals 1.0, the contours are drawn by the solid line and when less than 1.0, by the broken line. The interval of the contours is 0.2. The thick arrow shows the direction of wave incidence. Though there exists a little discrepancy chiefly nearby the harbor mouth, these results are agree well with each other.

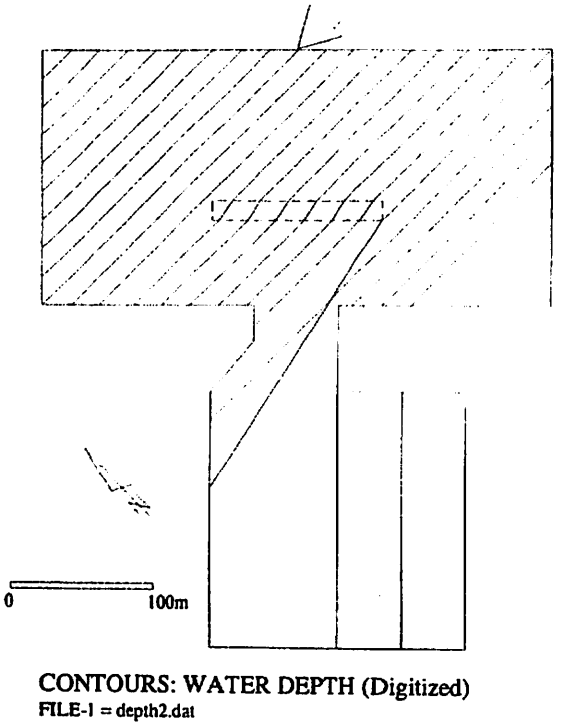


(1) Evaluated in terms of the infinit element.

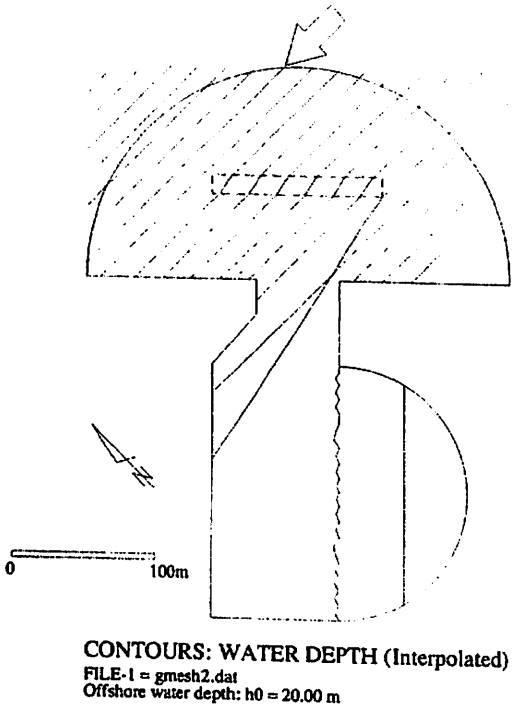


(2) Evaluated in terms of the coupling method.

Figure 18. Contours showing the linear wave height distribution (job = 7).



(1) Combined with batymetric data (job = 3).



(2) Combined with generated element mesh data (job = 6).

Figure 19. Wave rays on the two kinds of depth contours.

The partial standing waves formed in front of the straight shoreline (y = 0) and the submerged breakwater are clearly seen. With the aide of partial wave reflection by the submerged breakwater, the wave field in the harbor seems to become calm compared with the normal condition of wave incidence.

- Jobs = 3, 6: As the further illustration of graphics, Figure 19 shows an example of combination of the wave ray trajectory due to refraction with the depth contours obtained from (1) bathymetric data and (2) the generated element mesh data. The angle of wave incidence, in this case, is 45 degrees. Wave rays are refracted at the edge of the submerged breakwater and when reaches to the shoreline, stop propagation. Combination of the wave ray trajectory with bathymetric data, Figure 19 (1), is usual for checking incidence of waves. On the contrary, combination with generated element mesh is sometime valuable for investigation of wave rays together with the wave height distribution.

■ Routine 9: Graphics II - Cross Sections

Purpose

This routine generates data for the values of water depth and wave height at any specified point and their cross sections, which are available for commercial-base software.

Options

For digitized bathymetric data ("depth2.&&&"):

- job = 1: Evaluation of the water depth at any point.
- job = 2: Cross section of the water depth along the specified line.

For interpolated bathymetric data and the wave height ("gmesh2.&&&" and "waves2.&&&"):

- job = 3: Evaluation of the water depth and wave height at any point.
- job = 4: Cross section of the water depth and wave height along the specified line.

Procedure

1. Main routine: crossm.f
2. Subroutine: cross.f
3. I/O data files

(a) For jobs = 1 and 2:

- file1 = "depth2.&&&": Input file where, after triangulation, details of bathymetric data have already been stored.
- file3 = "cross.&&&": Output file where evaluated values of water depth will be stored.

(b) For jobs = 3 and 4:

- file1 = "gmesh2.&&&": Input file where details of the generated linear element mesh and the nodal water depth have already been stored.
- file2 = "waves2.&&&": Input file where the wave data have already been stored.
- file3 = "cross.&&&": Output file where evaluated values of the water depth and wave height will be stored.

4. Variables

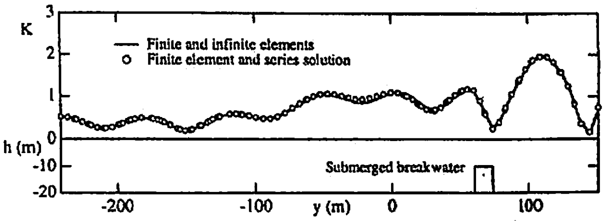
job	Option number (1, 2, 3 and 4).
ninpt	Number of input data for jobs = 1 and 3.
icount	Number of cross sections for jobs = 2 and 4.
cx, cy	• For jobs = 1 and 3, arbitrary points (in meters), at which the water depth and wave height are evaluated. • For jobs = 2 and 4, the first and last points on a straight line (in meters), along which the cross section is evaluated.

5. Attribute file for variables
- Attribute file = "&&&&&&": Input file where the variables have already been stored.

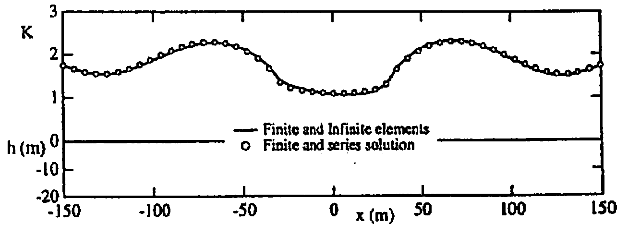
6. Dimensional declarators

mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of nodes.
mn1	Maximum number of boundary nodes.
mn2	Maximum number of interior nodes in each subregion.

Notes:
- Refer to Notes in Routine 7 about the attribute file.
- Cross sections for jobs = 2 and 4 are evaluated along the given straight line from the first-point toward the last-point.
- Data of the cross section for job = 2 can be used for preparation of line segments, being processed in Routine 3 and used in the file "gmesh1.&&&&".
- For the numerical results of the nonlinear wave analysis, use Routine "nlcross" and the I/O data files "gmesh3.&&&&" and "waves3.&&&&". Variables and dimensional declarators and so on are the same with those in Routine "cross".



(1) Along the line x = 0.



(2) Along the line y = 0.

Figure 20. Cross sections of the wave height and water depth (job = 4).

Form of Data

Table 7 shows the form of data for the water depth, wave height and their cross sections.
- The bracket means the range of iteration.
- If there is a boundary on the cross section, the value of water depth will be set to be "99999.0" to make a vertical wall.

Examples of Graphics II

Examples of the cross sections of wave height distribution along the lines x = 0 and y = 0 for job = 4 are shown in Figure 20, where the variables h and K are the water depth (in meters) and the dimensionless wave height relative to the incident wave height, respectively.

- The partial standing waves formed in front of the submerged

Table 7. Form of data for the water depth, wave height and their cross sections.

Variables	Format	Comment etc.
• job = 1: For water depth		
[jj, xp, yp, hp1	i5, 3e14.5	Dummy No., (x, y)-coordinate, water depth.
[jj, xp, yp, hp1, hp2	i5, 4e14.5	If on the discontinuity point: Dummy No., (x, y)-coordinate, water depth#1, water depth#2.
• jobs = 2 & 4: For cross section of water depth		
[jj, xp, yp, hp	i5, 3e14.5	Dummy No., (x, y)-coordinate, water depth.
• job = 3: For wave height and water depth:		
[jj, xp, yp, wh, hp1	i5, 4e14.5	Dummy No., (x, y)-coordinate, wave height, water depth.
[jj, xp, yp, wh, hp1, hp2	i5, 5e14.5	If on the discontinuity point: Dummy No., (x, y)-coordinate, wave height, water depth#1, water depth#2.
• job = 4: For cross section of wave height		
[jj, xp, yp, wh	i5, 3e14.5	Dummy No., (x, y)-coordinate, wave height.

breakwater is clearly seen.

- Two kinds of the finite element solution for the boundary condition at infinity agree well with over the whole region.
- Though, investigating in detail, there is a very little difference in the values of relative wave height nearby the harbor mouth, the results of two solutions show efficiency of the numerical model in CATWAVES.

■ Routine 10: Graphics III - Bird's-Eye Views

Purpose

The bird's-eye view (perspective projection) is created for two types of surface modelling, i.e., using the triangular mesh as polygons and contour lines. It is set up through two processes:

(1) First, set up triangular data for perspective projection by Routine "cset".

(2) Adopting the data created by Routine "cset", the second routine "cperth" generates vector data for graphics of perspective projection.

Options

The first process:

- job = 1: For digitized bathymetric data ("depth2.&&&").
- job = 2: For interpolated bathymetric data ("gmesh2.&&&").
- job = 3: For wave data("waves2.&&&").

The second process:

- itype = 1: For surface modelling by polygons.
- itype = 2: For surface modelling by contour lines.

Procedure for Preparation of Segmental Data

1. Main routine: csetm.f
2. Subroutine: cset.f
3. I/O data files

(a) For job = 1:

- file1 = "depth2.&&&": Input file where, after triangulation, details of bathymetric data have already been stored.
- file3 = "cset.&&&": Output file where triangular data for perspective projection will be stored.

(b) For job = 2:

- file1 = "gmesh2.&&&": Input file where details of the generated linear element mesh and the nodal water depth have already been stored.
- file3 = "cset.&&&": Output file where triangular data for perspective projection will be stored.

(c) For job = 3:

- file1 = "gmesh2.&&&": Input file where details of the generated linear element mesh and the nodal water depth have already been stored.
- file2 = "waves2.&&&": Input file where the wave data have already been stored.
- file3 = "cset.&&&": Output file where triangular data for perspective projection will be stored.

4. Variables

job	Option number (1, 2 and 3).
ibcon	Index for drawing boundary; 1: No drawing boundary. 2: Drawing only boundary. 3: Drawing boundary and vertical wall.

5. Dimensional declarators

mr	Maximum number of subregions.
mm	Maximum number of elements.
mn	Maximum number of node.
mn1	Maximum number of boundary nodes.

Procedure for Bird's-eye Views

1. Main routine: cperthm.f
2. Subroutine: cperth.f
3. I/O data files

(a) For itype = 1:

- file1 = "cset.&&&": Input file where triangular data for bird's-eye views (perspective projection) have already been stored.
- file3 = "&&&&&": Output file where vector data for bird's-eye views will be stored.

(b) For itype = 2:

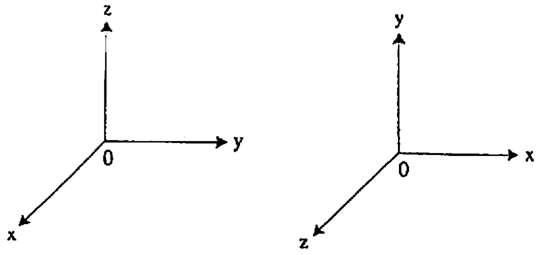
- file1 = "cset.&&&": Input file where triangular data for bird's-eye views (perspective projection) have already been stored.
- file2 = "cont.&&&": Input file where data for line segments of contours have already been stored. This data file of contour lines for bird's-eye views is generated by Routines "femfig" and "nlfemfig" in *Graphics I - Plane Views*.
- file3 = "&&&&&": Output file where vector data for bird's-eye views will be stored.

4. Variables

itype	Type of surface modelling (1 and 2).
(xf, yf, zf)	Point of the viewer's eyes in the world coordinate (in meters).
(xa, ya, za)	Point in the world coordinate, at which the viewer's eyes are directed (in meters).
h	Location of the view plane in the world coordinate (in meters), on which the bird's-eye view is drawn, i.e., distance from the point of the viewer's eyes.
th0	Viewing angle (in degrees).
yfc	Magnification factor in the vertical coordinate; Considering difference of the scale between the horizontal bathymetry measured in meters and the vertical variable, i.e., the water depth (in meters) and the dimensionless wave height, you should magnify the vertical value by the factor of "yfc" to obtain appropriate figures.

5. Dimensional declarators

mm	Maximum number of polygons after triangulation.
mn	Maximum number of nodes.
mn1	Maximum number of boundary segments.



(1) In the finite element analysis. (2) In perspective projection.

Figure 21. Definition of the two systems; (1) the normal coordinate in the finite element analysis and (2) the world coordinate in perspective projection.

Definition of the World Coordinate System

In the finite element analysis the horizontal and vertical axes are (x, y) and z, respectively, usually defined as in Figure 21 (1). In the world coordinate used in the perspective projection, on the contrary, the horizontal axes are (x, z) and the vertical axis is y, as defined in Figure 21 (2). The viewer looks at the object form the point (xf, yf, zf) to the point (xa, ya, za) and then the object is projected on the view plane parallel to the vertical (x, y)-plane and h meters apart from the point of the viewer's eyes.

Note:

- For graphics from the numerical results of the nonlinear wave analysis, use Routines "nlcset" and "nlcperth" and the I/O data files "gmesh3.&&&" and "waves3.&&&". Variables and dimensional declarators and so on are the same with those in Routines "cset" and "cperth".

Form of Vector Data

Similar to Routine 8: *Graphics I - Plane Views*, there are two kinds of vector data.

Examples of Graphics III

According to the options, you can obtain three kinds of perspective projection for bathymetry and wave height distribution.

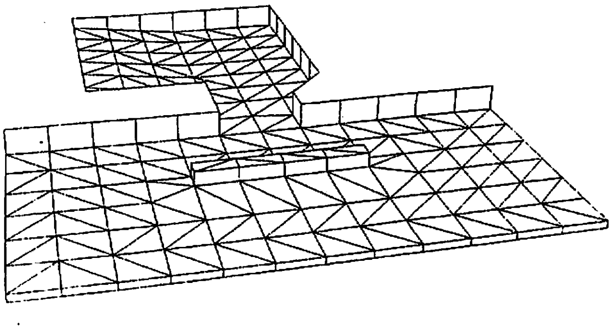


Figure 22. Perspective projection of the digitized bathymetry (job = 1, itype = 1); xf = 700m, yf = 300m, zf = 150m, xa = ya = za = 0m, h = 500m, th0 = 30degrees, yfc = 1.

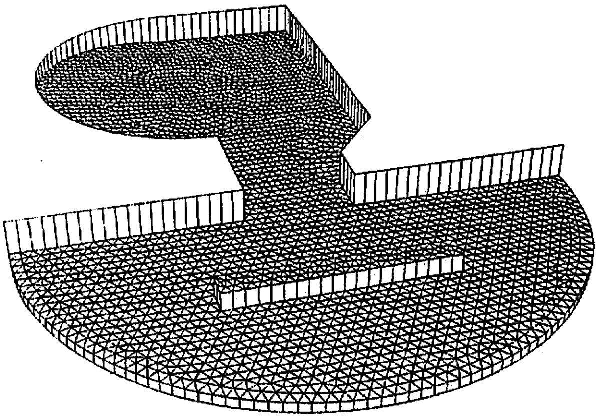
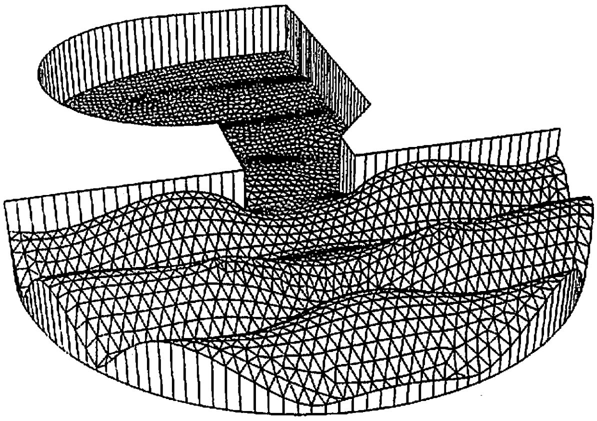
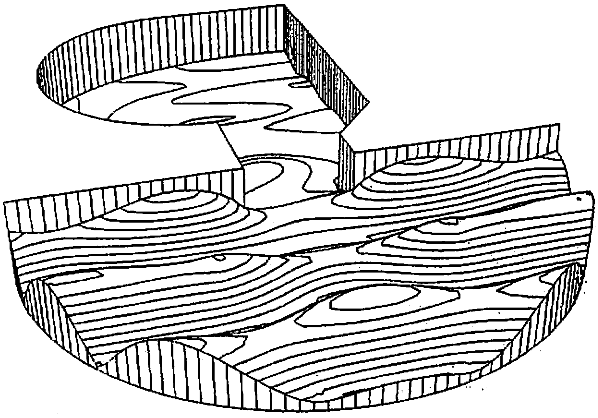


Figure 23. Perspective projection of the bathymetry for the linear finite element analysis (job = 2, itype = 1); xf = 500m, yf = 300m, zf = 150m, xa = ya = za = 0m, h = 500m, th0 = 30degrees, yfc = 1.



(1) Water surface modelling by the triangular mesh (itype = 1).



(2) Water surface modelling by the contour lines (itype = 2).

Figure 24. Perspective projection for the linear wave height distribution (job = 3); same parameters as in Figure 23 except that yfc = 15.

- Jobs = 1, 2: The perspective projection of the digitized bathymetry and that for the finite element analysis are shown in Figures 22 and 23, respectively, in case of the parameter ibcon = 3 and with the magnification factor yfc = 1. By using these figurers the user can confirm the features of bathymetry and the finite element mesh.
- Job = 3: Figure 24 shows the perspective projection for wave height distribution, where the vertical scale is magnified by the factor yfc = 15. In figure (1), the triangular mesh are adopted as polygons for modelling the water surface and, in figure (2), the contour lines are used for modelling the water surface. The partial standing waves are formed in front of the submerged breakwater and the straight shoreline. We can see that the wave height in the harbor is attenuated due to the effects of the submerged breakwater but the wave period seems to be of the same with that of the incident wave. With the aide of these figures, we can easily understand the features of the linear wave height distribution.

5. Examples of Graphics for the Nonlinear Wave Analysis

So far the graphics for the linear wave analysis have been illustrated. Here are examples of graphics of plane views, bird's-eye views and cross sections for the nonlinear wave analysis, showing the differences between the linear and nonlinear wave analyses.

Plain Views

- For the nonlinear wave analysis, we should notice that the size of the finite element mesh is governed by the local wave length of the highest Fourier component. If, for example, three Fourier components are assumed in the analysis, the element mesh size should be about 1/15-1/20 times the local wave length of the third harmonic of the Fourier components, the wave period of which is 1/3 of the incident wave period.
- Job = 5: Figure 25 shows the nonlinear finite element grids that consist of the pair of arcs and the central mesh. The region within the semicircular part of the harbor is subdivided along the depth contours of 10m and 20m. The incident wave period designed is 15 seconds. The regions on the submerged breakwater and in the semicircular part of the harbor are assumed to be the regions for the nonlinear wave analysis, where the fine finite element mesh with 6-nodes are created. Notice, in the figure, that the three intermediate nodes of the 6-nodes triangular element are linked. The element mesh size in the region for the linear wave analysis is 1/20 of the local wave length. In the regions for the nonlinear wave analysis, the element mesh size is about 1/8 of the local wave length for the third harmonic. However, there are three nodes on a side of the 6-nodes triangular element and therefore the distance between adjacent two nodes is about 1/16 of the local wave length.

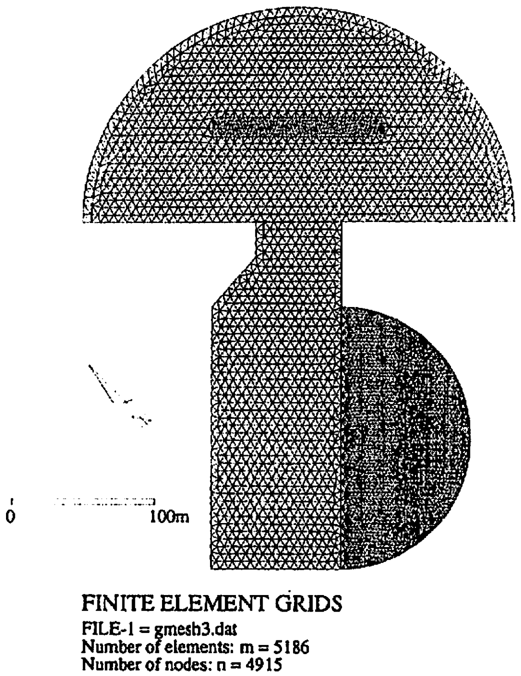


Figure 25. Triangular element mesh for the nonlinear finite element analysis, designed for waves with the period of 15 secs. (job = 5).

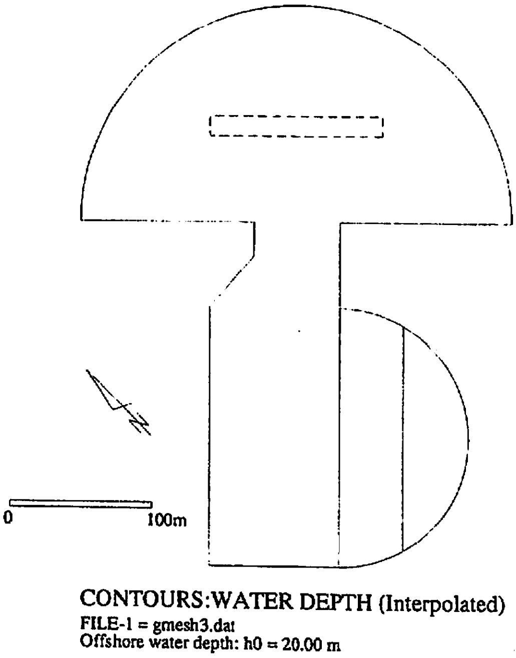
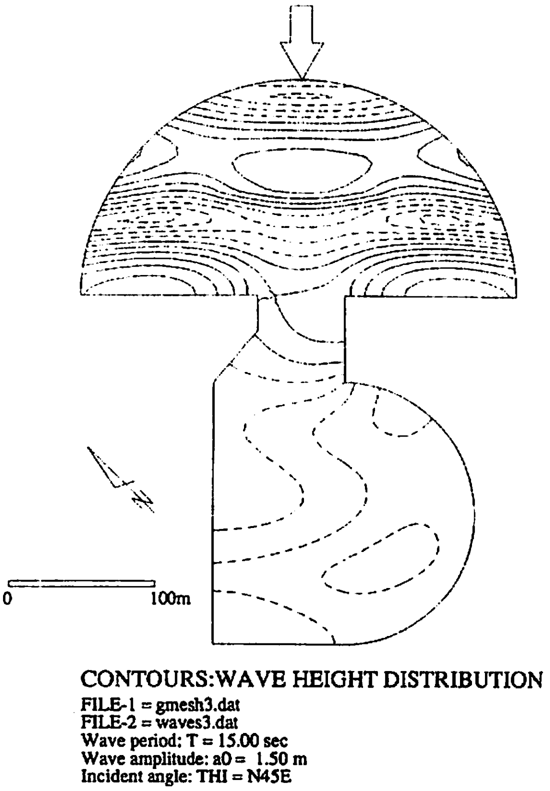
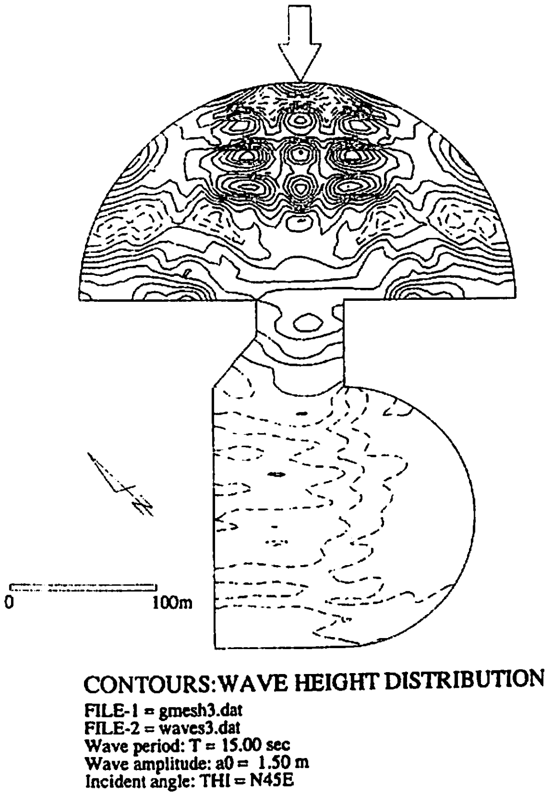


Figure 26. Depth contours generated from the bathymetric data (job = 6).

- Job = 6: The depth contours for interpolated bathymetric data are shown in Figure 26. Since, in this case, the bathymetry is subdivided along the depth contours, the contour lines are parallel and correct (Compare with Figure 17).
- Job = 7: Figure 27 shows the contour lines of the linear and nonlinear wave height distribution, where the incident wave period is 15 seconds and the incident wave amplitude is 1.5 meters. Comparing the results of the linear and nonlinear wave analyses, nonlinear effects are clearly seen in the regions,



(1) Linear wave analysis.



(2) Nonlinear wave analysis.

Figure 27. Contours showing the linear and nonlinear wave height distribution evaluated in terms of the infinit element (job = 7).

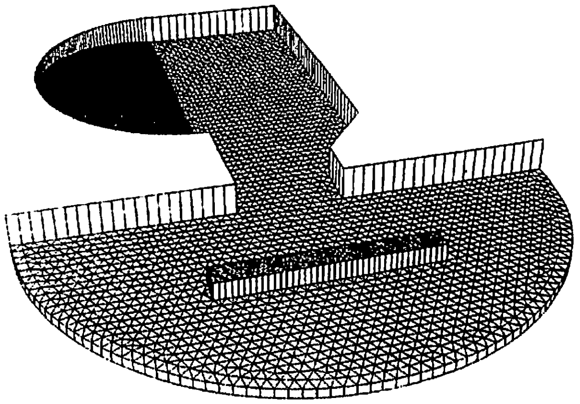
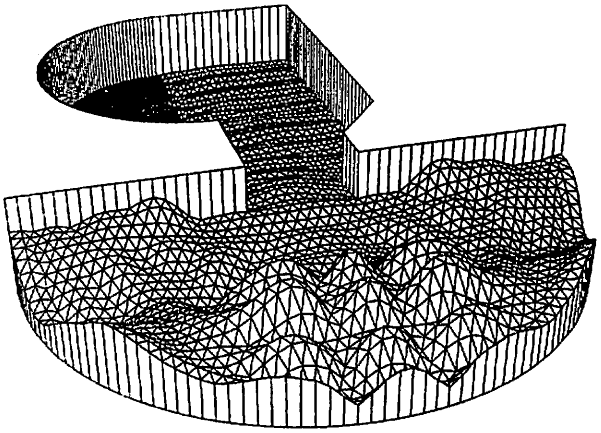
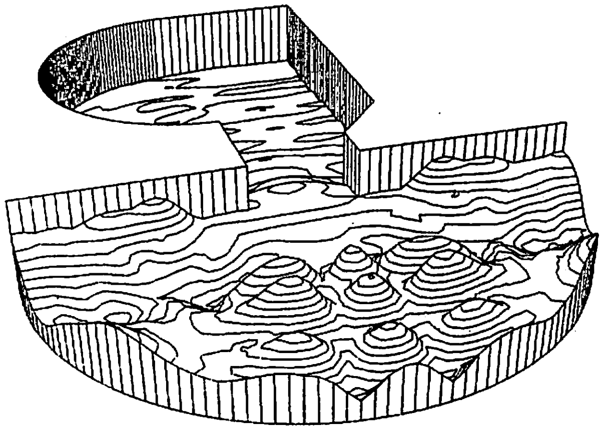


Figure 28. Perspective projection of the bathymetry for the nonlinear finite element analysis (job = 2, itype = 1); same parameters as in Figure 23.

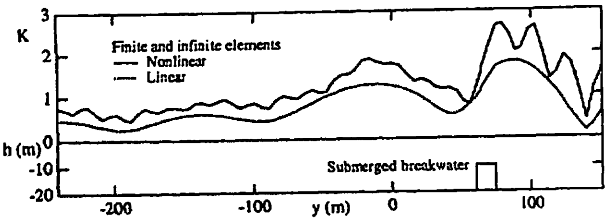


(1) Water surface modelling by the triangular mesh (itype = 1).

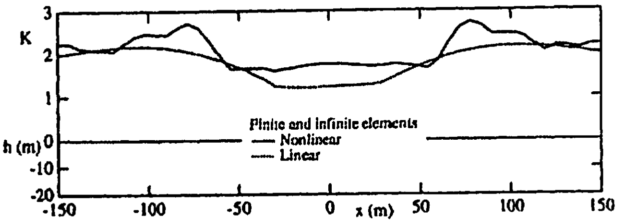


(2) Water surface modelling by the contour lines (itype = 2).

Figure 29. Perspective projection for the nonlinear wave height distribution (job = 3); same parameters as in Figure 23 except that yfc = 15.



(1) Along the line x = 0.



(2) Along the line y = 0.

Figure 30. Cross sections of the wave height and water depth (job = 4).

especially in front of the submerged breakwater and the straight shoreline (y = 0). The higher harmonics are generated as the results of nonlinear wave interaction and they have the wave period shorter than the incident wave period. In front of the sub-merged breakwater, the short-crested waves of the higher harmonics are generated both in the (x, y)-directions. On the contrary, the partial standing waves near the shoreline (y = 0) are deformed due to the higher harmonics, compared with that of the linear standing waves. Furthermore, the wave field in the harbor consists of the higher harmonics and then becomes complex.

Bird's-Eye Views

- Similar to Figures 23 and 24, the bird's-eye views of bathymetry and nonlinear wave height distribution for jobs = 2 and 3 are shown in Figures 28 and 29, respectively, where the parameter ibcon = 3. In Figure 28, the fine mesh generated on the sub-merged breakwater and in the semicircular part of the harbor are seen. Figure 29 shows clearly the behavior of wave motion, especially short-crested waves of the higher harmonics formed in front of the submerged breakwater and of the partial standing waves along the shoreline (y = 0). Furthermore, it is evident that the higher harmonics are excited in the harbor compared with the results of the linear wave analysis, as was shown in Figures 27 and 24.

Cross Sections

- Finally, in order to illustrate an example of nonlinear effects quantitatively, Figure 30 shows the comparison of the linear and nonlinear wave height distribution along the lines x = 0 and y = 0. The higher harmonics generated due to nonlinear wave interaction ride on the linear wave such that the nonlinear wave height becomes larger in the almost whole regions.

6. Concluding Remarks

The structure of CATWAVES - the mesh generation, finite element analyses and graphics routines - and its function have been demonstrated with the aide of a simple example for the artificial harbor model. The results of linear and nonlinear wave analyses show the efficiency of the numerical simulation implemented in CATWAVES. The remainder is introduction of some wave phenomena into the model equation, e.g., wave breaking, wave set-up and wave induced current in shallow water.

References

Berkoff, J. C. W. (1972): Computation of combined refraction-diffraction problem, Proc. 13th Conf. on Coastal Eng., ASCE, Vol.1, pp.471-490.

Chen, H. S. and C. C. Mei (1975): Hybrid-element method for water waves, Proc. of the Modelling Technics Conf.(Modelling 1975), Vol.1, pp.63-81.

Cuthill, E. and J. McKee (1969): Reducing the bandwidth of sparse symmetric matrices, Proc. 24th Nat. Conf. of the Association for Computing Machinery, Brandon Press, New Jersey, pp.157-172.

Duff, I. S. (1979): MA28 - A set of Fortran subroutines for sparse unsymmetric linear equations, Report AERE R8730, HMSO, London.

Duff, I. S., A. M. Erisman and J. K. Reid (1986): Direct Methods for Sparse Matrices, Numerical Mathematics Scientific Computation, Oxford Science Publications, Oxford, 341pp.

Kohler, E. K. (1997): PSPLOT: PostScript for technical drawings - A free Fortran-callable PostScript plotting library, Nova Southeastern Univ., Florida, 36pp.

Lo, S. H. (1985): A new mesh generation scheme for arbitrary planar domains, Inter. Jour. for Numer. Methods. in Eng., Vol.21, pp.1403-1426.

Mei, C. C. (1983): The Applied Dynamics of Ocean Surface Waves, John Wiley and Sons Inc., New York, 740pp.

Saad, Y. and M. H. Schultz (1986): GMRES: A generalizad minimum residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput., Vol.7, pp.856-869.

Sonneveld, P. (1989): CGS: A fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput, Vol.10, pp.36-52.

Sulaiman, D. M ., S. Tsutsui, H. Yoshioka, S. Oshiro, and Y. Tsuchiya (1994): Prediction of the maximum wave on the coral flat, Proc. 24th Inter. Conf. on Coastal Eng., ASCE, pp.6.9-6.23.

Tsutsui, S. (1989): CATWAVES-Wave Analysis System, Centre for Water Research, University of Western Australia, Report No.WP-318-ST, 61pp.

Tsutsui, S. and K. Zamami (1993): Jump condition of energy flax at the line of bathymetric discontinuity and wave breaking on the reef flat, Coastal Eng. in Japan, JSCE, Vol.36, pp.155-175.

Tsutsui, S. (1995): Model equations combining full linear dispersion with long wave nonlinearity, Part II, Bull. of the Faculty of Eng., Univ. of the Ryukyus, Vol.50, pp.45-54 (in Japanese).

Tsutsui, S., K. Suzuyama, and H. Ohki (1996a): Model equation of nonlinear dispersive waves in shallow water and its application to the step-type reef, Proc. Coastal Eng., JSCE, Vol.43., pp.16-21 (in Japanese).

Tsutsui, S., K. Suzuyama and H. Ohki (1996b): Model equations combining full linear dispersion with long wave nonlinearity, Part III, - Nonlinear evolution of waves on the step-type reef, Bull. of the Faculty of Eng., Univ. of the Ryukyus, Vol.52, pp.25-39 (in Japanese).

Tsutsui, S., K. Suzuyama and H. Ohki (1998a): Model equations of nonlinear dispersive waves in shallow water and an application of its simplest version to wave evolution on the step-type reef, Coastal Eng. Jour., JSCE, Vol.40, No.1, pp.41-60.

Tsutsui, S. and Y. Ohki (1998b): Nonlinear wave evolution on the slope- and step-type reefs, Proc. Coastal Eng., JSCE, Vol.45., pp.41-45 (in Japanese).

Tsutsui, S. and H. Ohki (1998c): Model equations combining full linear dispersion with long wave nonlinearity, Part IV, - Application of Bi-CGSTAB to sparse nonsymmetric systems -, Bull. of the Faculty of Eng., Univ. of the Ryukyus, Vol.55, pp.17-25(in Japanese).

Tsutsui, S. (1999a): New infinite element for wave analysis in unbounded coastal domains. Proc. Coastal Eng., JSCE, Vol.46., pp.81-85 (in Japanese).

Tsutsui, S. (1999b): New infinite element for nonlinear wave analyses in unbounded coastal domains, Bull. of the Faculty of Eng., Univ. of the Ryukyus, Vol.58, pp.17-27 (in Japanese).

Tsutsui, S. (2000): New infinite element for nonlinear wave analyses in unbounded coastal domains, part 2, - higher order infinite element and error estimate -, Bull. of the Faculty of Eng., Univ. of the Ryukyus, Vol.59, pp.39-51 (in Japanese).

van der Vorst, H. A. (1992): Bi-CGSTAB - A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., Vol.13, pp.631-644.

Zienkiewicz, O. C., D. W. Kelly and P. Bettess (1977): The coupling of the finite element method and boundary solution procedures, Inter. Jour. for Numer. Methods. in Eng., Vol.11, No.2, pp.355-375.

Zienkiewicz, O. C. and R. L. Taylor (1989): The Finite Elment Method, Fourth Ed., Vol. 1, McGraw-Hill, New York, 648p.

Zlatev, Z., J. Wasniewski and K. Schaumburg (1981): Y12M - Solution of large and sparse systems of linear algebraic equations, Lecture Note in Computer Sci., Vol.121, Springer-Verlag, Berlin, Heifelberg, New York, and Tokyo, 128p.

Appendix A. Model Equation implemented in CATWAVES for the Nonlinear Wave Analysis

The model equation of nonlinear wave motion implemented in CATWAVES is given as follows. All the physical variables are nondimentionalized by using the representative length h_0^*, the time $\sqrt{h_0^*/g}$ and the velocity $\sqrt{gh_0^*}$, where h_0^* is the constant water depth in the exterior region, g is the acceleration of gravity, and the asterisk (*) denotes dimensional variables.

Let the horizontal coordinates, the local water depth, the water surface displacement and the incident wave amplitude be (x^*, y^*), h^*, η^* and a_0^*, respectively, where the coordinate origin being on the still water level. Let also the frequency, the wave number, the wave velocity and the group velocity be ω^*, k^*, c^* and c_g^*, respectively. The dimensionless variables are then defined as:

$$\left.\begin{aligned} (x, y, h, \eta, a_0) &= (x^*, y^*, h^*, \eta^*, a_0^*) / h_0^*, \\ \omega &= \omega^* \sqrt{h_0^*/g}, \\ k &= k^* h_0^*, \\ (c, c_g) &= (c^*, c_g^*) / \sqrt{gh_0^*}. \end{aligned}\right\} \tag{A.1}$$

Assume that the water surface displacement can be expressed as the Fourier series such that

$$\eta(x, y, t) = \frac{1}{2} \sum_n \eta_n(x, y) \exp(-\mathrm{i} n\omega t) \tag{A.2}$$

with $n = \pm 1, \pm 2, \ldots$, where i is the imaginary unit and η_{-n} is the complex conjugate of η_n. The model equation for the n-harmonic of the Fourier components, $\zeta_n = \eta_n / a_0$, is given by

$$\frac{\partial}{\partial x}\left(cc_g \frac{\partial \zeta_n}{\partial x}\right) + \frac{\partial}{\partial y}\left(cc_g \frac{\partial \zeta_n}{\partial y}\right) + (n\omega)^2 \frac{c_g}{c} \zeta_n$$

$$\begin{aligned} &= \frac{\varepsilon}{2} \sum_{l=-\infty}^{\infty} (n^2 - l^2)\omega^2 \frac{c_g}{c} \zeta_l \zeta_{n-l} \\ &- \frac{\varepsilon}{2} \sum_{l=-\infty, l \neq n}^{\infty} \frac{n+l}{n-l} cc_g \left(\frac{\partial \zeta_l}{\partial x} \frac{\partial \zeta_{n-l}}{\partial x} + \frac{\partial \zeta_l}{\partial y} \frac{\partial \zeta_{n-l}}{\partial y} \right) \\ &- \frac{\varepsilon}{2} \sum_{l=-\infty, l \neq n}^{\infty} \frac{2}{l(n-l)\omega^2} c^3 c_g \times \\ &\qquad \times \left(\frac{\partial^2 \zeta_l}{\partial x^2} \frac{\partial^2 \zeta_{n-l}}{\partial y^2} - \frac{\partial^2 \zeta_l}{\partial x \partial y} \frac{\partial^2 \zeta_{n-l}}{\partial x \partial y} \right) \end{aligned} \tag{A.3}$$

(Tsutsui, 1995; Tsutsui et al., 1996, 1998) with $l = \pm 1, \pm 2, \ldots$, and

$$\varepsilon = a_0 / h, \tag{A.4}$$

$$c^2 = \frac{1}{k} \tanh kh, \tag{A.5}$$

$$c_g / c = (1 + 2kh / \sinh 2kh) / 2, \tag{A.6}$$

where the frequency ω, the wave number k, the wave velocity c and the group velocity c_g in Eqs.(A.3), (A.5) and (A.6) are values for the wave of n-harmonic.

When all the nonlinear terms in the right-hand side of Eq. (A.3) are neglected, the model equation reduces to the mild-slope equation. Therefore, the finite element approximation based on the model equation (A.3) covers both linear and nonlinear wave motion.

Assuming the maximum number of Fourier components be N and, after quasi-linearization of the nonlinear terms in Eq.(A.3), discretizing based on the finite element method, we have the coupled system of nonlinear equations:

$$\varphi(x) \equiv A(x, \varepsilon) \cdot x = B, \tag{A.7}$$

where $x = \{\zeta_1, \zeta_2, \ldots, \zeta_N\}^T$ is the N- blocked vector, each block of which consists of unknowns of the number of nodes, A is the $N \times N$- blocked coefficient matrix of the coupled system, and B is the N- blocked given vector. Eqation (A.7) is the system to be solved.

The ordinary differential equation alternative to Eq.(A.7) can be developed as follows: Since the incident wave amplitude is included in the coefficient matrix of Eq.(A.7) as the parameter of the external force, the system (A.7) can be written as

$$\varphi(x, \lambda) \equiv A(x, \varepsilon\lambda) \cdot x = B, \qquad 0 \leq \lambda \leq 1, \tag{A.8}$$

introducing the parameter λ. Increase in the value of this parameter is identical with an increment of the incident wave amplitude. Then the unknown vector x is also the function of λ. Similar to the concept of the incremental method in the field of structural analysis, differentiating Eq.(A.8) with respect to λ leads to the required equation:

$$K_T(x) \frac{dx}{d\lambda} = -\frac{d\varphi}{d\lambda}, \tag{A.9}$$

where $K_T(x) = \partial\varphi / \partial x$ is the Jacobian matrix.