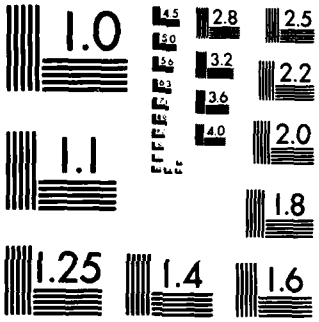
AD-A149 467 EROSION CONTROL OF SCOUR DURING CONSTRUCTION REPORT 6 1/2
FINITE - A NUMERICA..(U) ARMY ENGINEER WATERWAYS
EXPERIMENT STATION VICKSBURG MS HYDRA..
UNCLASSIFIED J R HOUSTON ET AL. SEP 84 WES/TR/HL-80-3-6 F/G 13/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

US Army Corps of Engineers

AD-A149 467

TECHNICAL REPORT HL-80-3

EROSION CONTROL OF SCOUR DURING CONSTRUCTION

Report 6

FINITE — A NUMERICAL MODEL FOR COMBINED REFRACTION AND DIFFRACTION OF WAVES

by

James R. Houston, Lucia W. Chou

Hydraulics Laboratory

DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
PO Box 631
Vicksburg, Mississippi 39180-0631

September 1984

Report 6 of a Series

Approved For Public Release; Distribution Unlimited

DTIC FILE COPY

Prepared for

DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

85 01 09 006

HYDRAULICS LABORATORY

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
Technical Report HL-80-3

2. GOVT ACCESSION NO.
AD-A149467

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
EROSION CONTROL OF SCOUR DURING CONSTRUCTION; Report 6, FINITE--A NUMERICAL MODEL FOR COMBINED REFRACTION AND DIFFRACTION OF WAVES

5. TYPE OF REPORT & PERIOD COVERED
Report 6 of a series

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
James R. Houston
Lucia W. Chou

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
US Army Engineer Waterways Experiment Station
Hydraulics Laboratory
PO Box 631, Vicksburg, Mississippi 39180-0631

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

12. REPORT DATE
September 1984

13. NUMBER OF PAGES
98

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Available from National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A hybrid finite element numerical model (FINITE) is presented that solves a linear mild-slope equation for short-wave propagation. The efficient formulation of the model permits the solution of large problems with relatively small time and memory storage requirements. Although the model solves an equation that is strictly valid only for mild bathymetric variations, ~~the model~~ it can provide reasonable answers for problems where there are rapid depth

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued).

variations (at much lower cost than required by three-dimensional models that are appropriate for problems involving rapid depth variations). The model does not provide a mechanism for energy dissipation, and thus energy loss through wave breaking is simulated by permitting waves to propagate out of the computational region. The program documentation, user guide, and sample problem output, are provided.

PREFACE

The study reported herein was authorized as a part of the Civil Works Research and Development Program by the Office, Chief of Engineers (OCE), US Army. This particular work unit, Erosion Control of Scour During Construction, is part of the Improvement of Operations and Maintenance Techniques (IOMT) Program. Mr. James L. Gottesman was the OCE Technical Monitor for the IOMT Program during preparation and publication of this report.

This study was conducted during the period 1 January 1981 through 31 March 1982 by personnel of the Hydraulics Laboratory of the US Army Engineer Waterways Experiment Station (WES) under the general supervision of Messrs. H. B. Simmons, Chief of the Hydraulics Laboratory; F. A. Herrmann, Jr., Assistant Chief of the Hydraulics Laboratory; R. A. Sager, Chief of the Estuaries Division and IOMT Program Manager; Dr. R. W. Whalin and Mr. C. E. Chatham, former and acting Chiefs of the Wave Dynamics Division, respectively; Mr. D. D. Davidson, Chief of the Wave Research Branch; and Dr. J. R. Houston, Research Hydraulic Engineer and Principal Investigator for the Erosion Control of Scour During Construction work unit. The Wave Dynamics Division was transferred to the Coastal Engineering Research Center (CERC) of the WES on 1 July 1983 under the direction of Dr. Whalin, Chief of the Coastal Engineering Research Center. Computer programming for this study was performed by Ms. L. Chou, Mathematician. This report was prepared by Dr. Houston and Ms. Chou.

Commanders and Directors of WES during the conduct of this investigation and the preparation and publication of this report were COL Nelson P. Conover, CE, and COL Tilford C. Creel, CE. Technical Director was Mr. F. R. Brown.

DTIC
COPY
INSPECTED
1

CONTENTS

	Page
PREFACE	1
CONVERSION FACTORS, US CUSTOMARY TO METRIC (SI) UNITS OF MEASUREMENT	3
PART I: INTRODUCTION	4
Statement of the Problem	4
Purpose of the Study	5
PART II: NUMERICAL MODEL (FINITE)	6
Background	6
Wave Equation	7
Finite Element Solution	8
Verification	11
Comparison with Three-Dimensional Numerical Model	17
Comparison with Laboratory Experiments	22
PART III: INPUT DATA PREPARATION FOR FINITE ELEMENT SOLUTION	26
Data Preparation for Computer Plotting	26
Plotting the Finite Element Grid	33
PART IV: PROGRAM FINITE	44
Data Preparation for Program FINITE	44
Procedure to Run Program FINITE	73
A Sample Application	83
PART V: SUMMARY	94
REFERENCES	95
APPENDIX A: NOTATION	A1

CONVERSION FACTORS, US CUSTOMARY TO METRIC (SI) UNITS OF MEASUREMENT

US customary units of measurement used in this report can be converted to metric (SI) units as follows:

Multiply	By	To Obtain
feet	0.3048	metres
feet per second	0.3048	metres per second
feet-feet per second	0.0929	metres-metres per second
feet per second per second	0.3048	metres per second per second
inches	25.5	millimetres
square feet	0.09290304	square metres

EROSION CONTROL OF SCOUR DURING CONSTRUCTION

FINITE--A NUMERICAL MODEL FOR COMBINED REFRACTION AND DIFFRACTION OF WAVES

PART I: INTRODUCTION

Statement of the Problem

1. It is frequently necessary to construct large engineering works of improvement in the surf and nearshore zone to protect harbor entrances, ~~recrea~~tional beaches, and navigation channels. These structures, usually built from quarried rock or precast concrete, are placed in position by crane or barge. When these major structures are erected in the coastal zone they alter the existing currents. Shallow-water surface gravity waves breaking on the new structure will cause bottom material to be suspended and transported from the region by longshore or other currents that may exist. This removal of material is often not compensated by an influx of additional material, ~~and~~ the result is a scour hole, or erosion, that usually develops along the toe of the structure.

2. In order to ensure structural stability and functional adequacy of the works of improvement, any scour area must be filled with nonerodible material (sufficiently stable to withstand the environmental forces to which it will be subjected). This may result in additional quantities of material being required during construction that can potentially lead to substantial cost overruns. To minimize potential cost increases due to scour during construction, it is necessary to quantify the probability and ultimate extent of potential scour during the scheduled construction period. This is an extremely complex problem and quantification of the probability of potential scour will likely be site-specific.

3. Since waves and wave-induced currents play an important role in producing scour near structures, it is important to predict the wave field in the vicinity of structures. Structures of interest may have complex shapes and the surrounding bathymetry may be very complex. Both long- and short-period waves may attack the structure from many different angles.

Purpose of the Study

4. The purpose of this study is to develop a numerical model that can determine wave fields near arbitrary structures in a region of arbitrary and variable bathymetry. The model must be capable of simulating both long and short waves approaching structures from any arbitrary direction. In addition, the model must be sufficiently efficient to allow application to real coastal engineering problems.

PART II: NUMERICAL MODEL (FINITE)

Background

5. Short waves have wavelengths that are sufficiently short (wavelength to depth ratio less than approximately 20) that propagation speeds are a function of wave frequency. In the nearshore region, short waves with periods from a few seconds to approximately 20 sec play an important role in the movement of sediment and the stability of coastal structures.

6. Equations that govern long-wave propagation over variable depths have been known for some time (Lamb 1932). However, for shorter period waves where frequency dispersion is important, the theory has included only constant depths. Attempts have been made (Pierson 1951; Eckart 1952) to develop a two-dimensional equation for combined refraction and diffraction that would govern short-wave propagation in a region of variable depth; however, the equations developed do not reduce to the appropriate simple refraction equation after neglecting the curvature of the amplitude function and they also do not reduce to the linear long-wave equation in the case of small water depth. In recent years, Boussinesq-type equations also have been used to study the propagation of short-period waves. These equations include terms that govern both frequency dispersion and nonlinearity. Two-dimensional modeling of these equations is difficult since the frequency dispersion term is third order, therefore requiring a third-order accurate numerical scheme. Abbott, Petersen, and Skovgaard (1978) have presented a time-marching, two-dimensional, finite difference numerical model with a third-order accurate implicit solution scheme. However, a time-marching scheme requires significant computational time. In addition, the finite difference method does not realistically model a complex land-water interface as a result of the stair-step representation and has difficulties in allowing scattered waves to propagate out of a finite extent grid, since these scattered waves are not known a priori.

7. Berkhoff (1972) and Schonfeld (1972) have derived a two-dimensional wave equation that governs short-wave propagation over moderately varying depths. Smith and Sprinks (1975) gave a formal derivation of this equation. In the present study, a hybrid finite element model (FINITE) is used to solve this wave equation. The method combines a finite element solution over a finite extent region of variable depth with an analytical solution for a

surrounding infinite region of constant depth.

Wave Equation

9. The propagation of periodic, small amplitude, surface gravity waves over a variable depth seabed of mild slope is governed by the following equation.

$$\nabla \cdot (cc_g \nabla\phi) + \frac{c_g}{c} \omega^2 \phi = 0 \qquad (1)$$

where

∇ = horizontal gradient operator, dimensionless

c = phase velocity, ft/sec* = $(g/k \tanh kh)^{1/2}$

g = gravitational constant, 32.174 ft/sec^2

k = wave number, $2\pi/L$, 1/ft

h = still-water depth, ft

c_g = group velocity, ft/sec = $1/2\ c(1 + G)$

G = $2k\ h/\sinh 2kh$, ft/sec

ϕ = velocity potential defined by $\bar{u} = \nabla\phi$, ft^2/sec

$\bar{u}$ = two-dimensional velocity vector, ft/sec

ω = angular frequency, $2\pi/T$, 1/sec

10. Equation 1 was derived by Berkhoff (1972) and Schonfeld (1972) and is discussed in detail by Jonsson and Brink-Kjaer (1973). This equation governs both refraction and diffraction. It reduces to the well-known "eikonal" equation governing refraction by neglect of the variation of the amplitude function in the horizontal plane. The equation reduces to the diffraction Helmholtz equation in deep or constant-depth water and to the linear long-wave equation in shallow water.

11. Finite element numerical models have been used to solve Equation 1. Berkhoff (1972, 1976) linked a finite element solution of Equation 1 over a variable depth region to a source distribution for a constant-depth outer region. However, as noted by Chen and Mei (1974), Berkhoff (1972) did not use a proper functional with the consequence that his global stiffness matrix was nonsymmetric and thus inconvenient numerically for all but the simplest problems. Bettess and Zienkiewicz (1977) also developed a finite element solution

---

* A table of factors for converting US customary units of measurements to metric (SI) units is presented on page 3.

of Equation 1. However, they used infinite elements to cover the constant-depth outer region. The shape function used in the infinite elements had an exponentially decreasing term in the direction away from the inner region. The choice of a decay length was somewhat arbitrary; but the solutions were not, in general, too sensitive to the exact value. A disadvantage of this technique is that the infinite elements increase the number of equations to be solved. If a problem requires a large number of elements because the region of interest is large and the incident wavelengths are short, this solution can require substantial computational time.

12. Equation 1 also has been solved by a parabolic approximation (Radder 1979). The approximation is derived from splitting the wave field into transmitted and reflected components and then neglecting the reflected components. This approach is applicable to some propagation problems but is not appropriate for problems involving wave interaction with coastal structures such as breakwaters.

Finite Element Solution

13. Equation 1 is solved by program FINITE using a hybrid finite element method originally developed by Chen and Mei (1974) to solve the diffraction Helmholtz equation in a constant-depth region. Space is divided into two regions as shown in Figure 1 (finite inner region A and infinite outer region B). Conventional finite elements are used in the variable depth region A . A single superelement is used to cover the constant-depth infinite region B . Variational principles are used that incorporate the matching conditions between the regular elements and the superelement as natural conditions. Thus a symmetric global stiffness matrix is obtained that is very advantageous for highly complex problems.

14. The variational principle for the boundary value problem requires that the following functional be stationary with respect to arbitrary first variation of the velocity potential ϕ :

$$F(\phi) = \iint_A \frac{1}{2}\left[\frac{cc_g}{g}(\nabla\phi)^2 - \frac{\omega^2 c_g}{gc}\phi^2\right]dA$$

$$+ \int_{\partial B_\infty} \frac{ikcc_g}{2g}\left(\phi - \phi_I\right)^2 dL - \int_{\partial B_\infty} \frac{cc_g \partial\phi_I}{g\partial n}\phi dL \qquad (2)$$

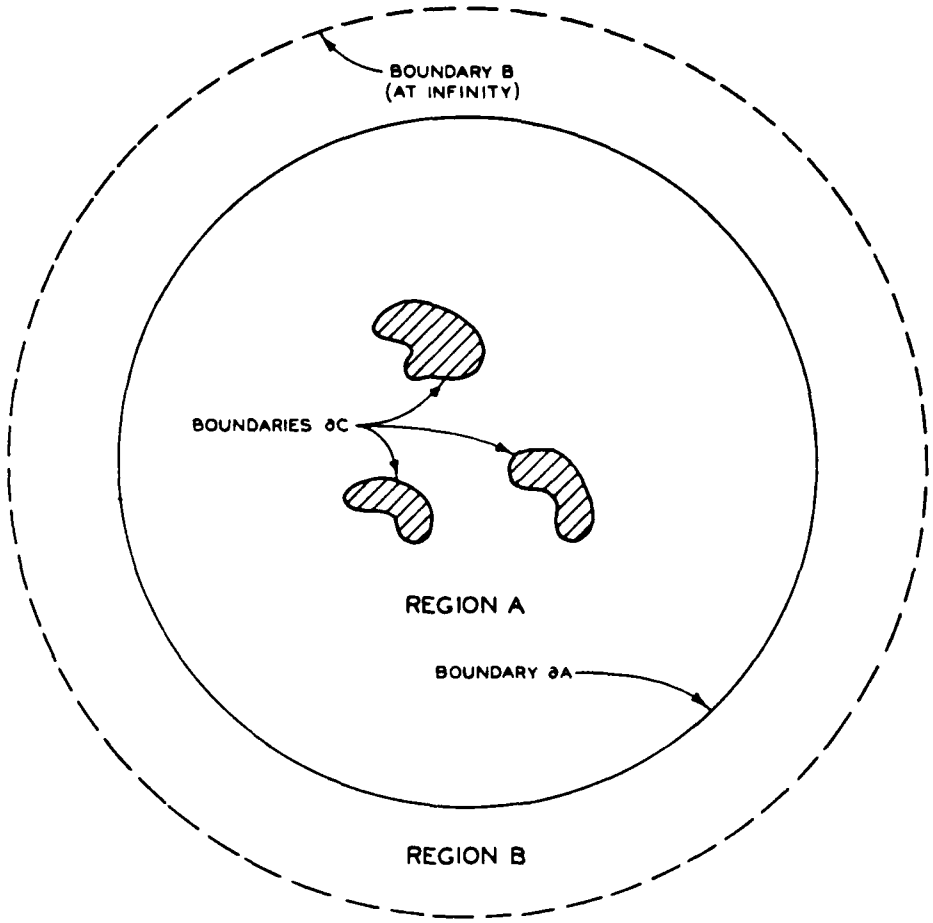


Figure 1. Regions of computation

where ϕ_I and ϕ_R are the incident and reflected wave velocity potentials, respectively; n is a unit normal; and the last two integrals are line integrals at infinity. Analogous to the derivation of Chen and Mei (1974), this functional can be rewritten as follows:

$$F(\phi) = \iint_A \frac{1}{2}\left[\frac{cc_g}{g}\left(\nabla\phi_A\right)^2 - \frac{\omega^2 c_g}{gc}\phi_A^2\right]dA + \int_{\partial A}\frac{1}{2}\frac{cc_g}{g}(\phi_B - \phi_I)\frac{\partial(\phi_B - \phi_I)}{\partial n_A}$$

$$- \int_{\partial A}\frac{cc_g}{g}\phi_A\frac{\partial(\phi_R - \phi_I)}{\partial n_A} - \int_{\partial A}\frac{cc_g}{g}\phi_A\left|\frac{\partial\phi_I}{\partial n_A}\right. \qquad (3)$$

$$+ \int_{\partial A}\frac{cc_g}{g}\phi_I\frac{\partial(\phi_B - \phi_I)}{\partial n_A} + \int_{\partial A}\frac{cc_g}{g}\phi_I\frac{\partial\phi_I}{\partial n_A}$$

where ϕ_B and ϕ_A are the velocity potentials in regions B and A, respectively, and n_A is a unit normal to the boundary separating regions B and A.

15. Note that all integrals are evaluated within region A or along ∂A. Thus the variational principle is a localized one. As discussed by Aranha, Mei, and Yue (1979), this variational principle can be replaced by an equivalent weak formulation.

16. The inner region A is assumed to have a variable depth and to be of finite extent. This region is subdivided into finite elements. Here the elements are triangular with simple linear shape functions. The infinite region B (or semi-infinite if a harbor located along an infinite coastline is of concern) is assumed to have a constant depth and is covered with a single superelement. Since region B has a constant depth, the governing equation is the diffraction Helmholtz equation. An analytical solution for the velocity potential in region B is well known and can be expressed as follows:

$$\phi_B = \sum_{n=0}^{\infty} H_n(kr)(\alpha_n \cos n\Theta + \beta_n \sin n\Theta) \tag{4}$$

where α_n and β_n are constant and unknown coefficients, $H_n(kr)$ is Hankel function of the first kind, and r and Θ are radial and angular variables in polar coordinates. For a semi-infinite region B and a straight infinite coastline, ϕ_B can be expressed as follows:

$$\phi_B = \sum_{n=0}^{\infty} \alpha_n H_n(kr) \cos n\Theta \tag{5}$$

The velocity potentials given in Equations 4 and 5 satisfy the Sommerfeld radiation condition that the scattered waves must behave as outgoing waves at infinity. Thus region B can be considered to be a single superelement with shape function given by Equations 4 and 5.

17. If the shape functions are used to evaluate the integrals of Equation 3 and the functional is extremized with respect to the unknowns, a set of linear algebraic equations is obtained. Of course, the infinite series given by Equations 4 and 5 must be truncated at some finite extent. The number of terms that must be retained depends upon the incident wavelength and may be found by increasing the number of terms until the solution is insensitive to

the addition of further terms. Solution of the boundary value problem thus reduces to the solution of N linear algebraic equations for N unknowns (where N is the number of node points in the finite element discretization plus the number of unknowns in the truncated series). That is,

$$\underset{N \times N}{[K]} \quad \underset{N \times 1}{\{\psi\}} = \underset{N \times 1}{\{Q\}} \tag{6}$$

The symmetric complex coefficient matrix [K] is in general large, sparse, and banded. It can be stored and manipulated in the computer in a packed form (Chen and Mei 1974). The packed form is chosen to be a rectangular array (N variables in length and the semibandwidth in width). Only elements of [K] on and above the diagonal and within the band width need to be stored in the packed form.

18. Although the packed form of [K] greatly reduces the required computer memory, the problems discussed later are so large that even memory requirements of the packed form of [K] are excessive. However, since the symmetric coefficient matrix is positive definite, a solution is possible by elimination methods without pivoting. Without pivoting, elimination performed using one row affects only the triangle of elements within the band below that row. Thus the packed form of [K] can be partitioned into several smaller blocks. Using Gaussian elimination, only two blocks at a time are involved in the reduction and back substitution with the remainder of the blocks kept in peripheral storage. This technique allows the solution of extremely large matrices.

Verification

19. To verify this numerical model, comparisons were made between the finite element calculations and both an analytical and a numerical solution for the interaction of waves with a circular island on a paraboloidal shoal. Figure 2 is a sketch of the problem. Hom-ma (1950) presented the analytical solution to the long-wave equation for plane waves incident upon this island.

20. Many investigators, including Berkhoff (1972, 1976) and Bettess and Zienkiewicz (1977), have compared their finite element solutions of the long-wave equation with Hom-ma's solution. For example, Berkhoff (1972, 1976) used 156 elements to cover the paraboloidal shoal (actually one-half of the

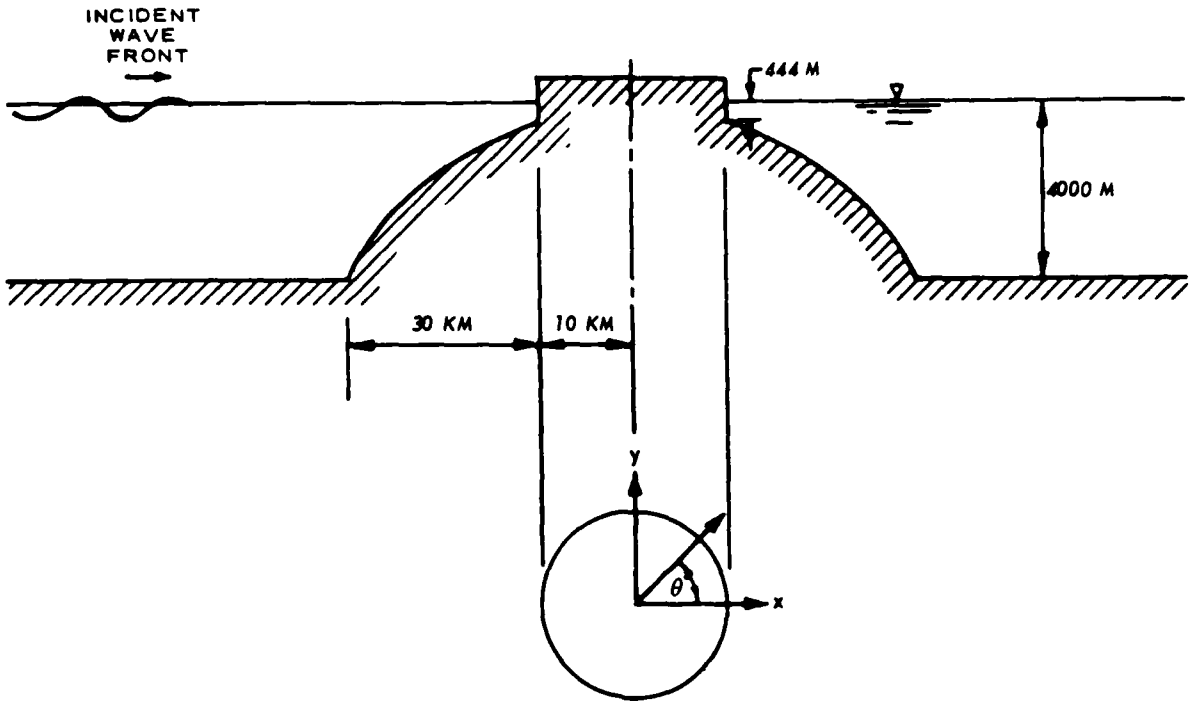


Figure 2. Circular island on a paraboloidal shoal

shoal since by symmetry the solution required calculations for just one-half of the shoal). This number of elements was not sufficient to adequately resolve features of the problem; consequently, Berkhoff's finite element solution differed from Hom-ma's analytical solution by as much as 70 percent at one location. Since Berkhoff did not attempt to use a finer grid, it is likely that his coarse grid required a moderate computational time. The model described in this paper would require less than 0.1 sec of computational time on a CRAY-1 computer to solve Equation 6 using Berkhoff's grid. Bettess and Zienkiewicz (1977) also used a fairly coarse grid similar to Berkhoff's grid.

21. Lautenbacher (1970) used an integral equation solution to solve the long-wave equation for waves interacting with a circular island on a shoal with linear side slopes. He used a coarse circular mesh grid with only 130 points. Because the resulting coefficient matrix was full, the computational time required for a solution was 60 min on an IBM 7094 computer.

22. Figure 3 shows a finite element grid with 2,640 elements used by the model described in this report to solve the problem of the interaction of long waves with a circular island on a paraboloidal shoal (by symmetry only

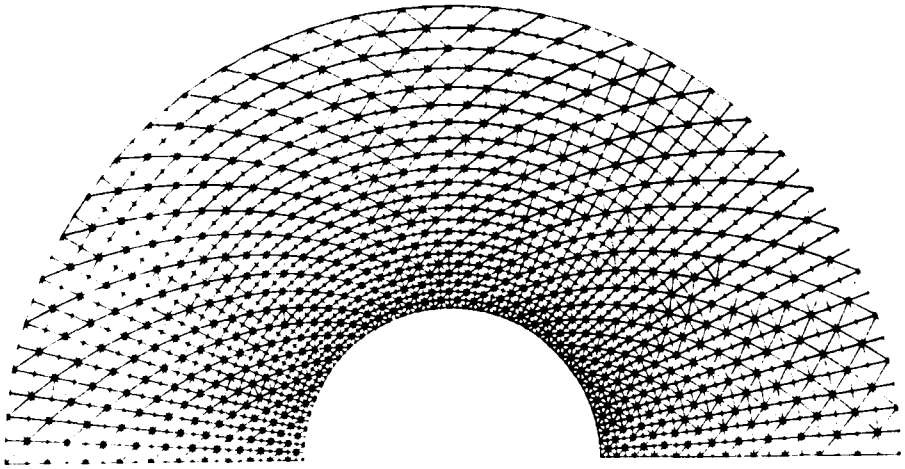


Figure 3. Finite element grid for paraboloidal island (2,640 elements)

half the shoal needs to be considered). Figure 4 shows comparisons between Hom-ma's analytical solution and the finite element model solution for incident waves with five different periods. The agreement is excellent with only slight differences for the 240-sec waves (resulting from lower resolution of the incident wave for shorter period waves). The computational time for solution is approximately 4 sec on a CRAY-1 computer.

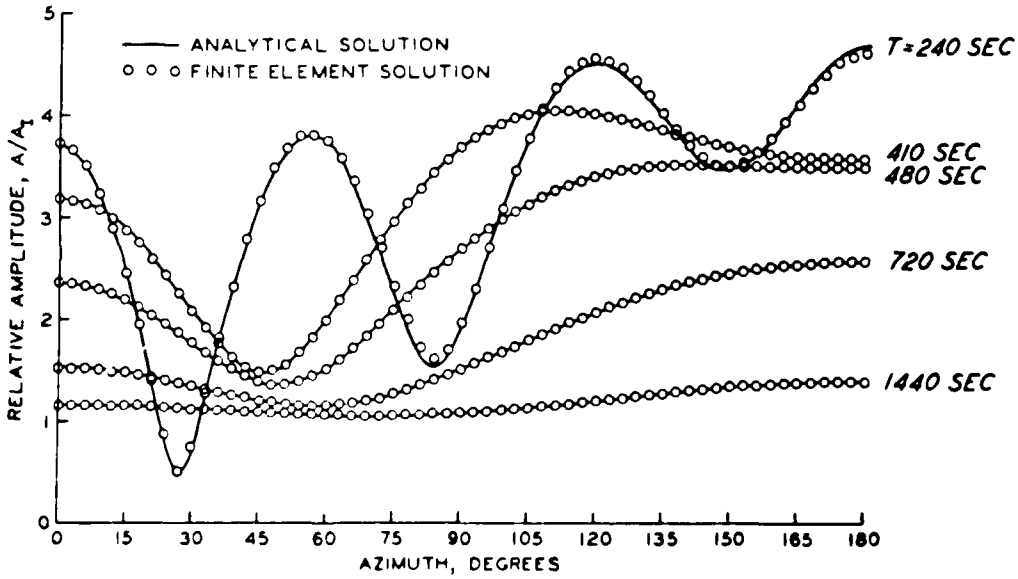


Figure 4. Solutions without dispersion (2,640 element grid)

23. Jonsson, Skovgaard, and Brink-Kjaer (1976) show that for a wave with a 240-sec period interacting with the circular island on a paraboloidal shoal the effect of frequency dispersion is not particularly significant. The ratio of wavelength to water depth for this case is approximately 11. However, for a 120-sec incident wave (wavelength to water depth ratio of less than 5), frequency dispersion is quite significant. In order to maintain a resolution of a 120-sec wave that is approximately equal to that obtained for the 240-sec wave using the 2,640-element grid, it is necessary to reduce element side lengths by a factor of approximately 2. This reduction results in a quadrupling of the number of elements. Figure 5 shows a finite element grid with 10,560 elements used to calculate the interaction of a 120-sec wave with the island.

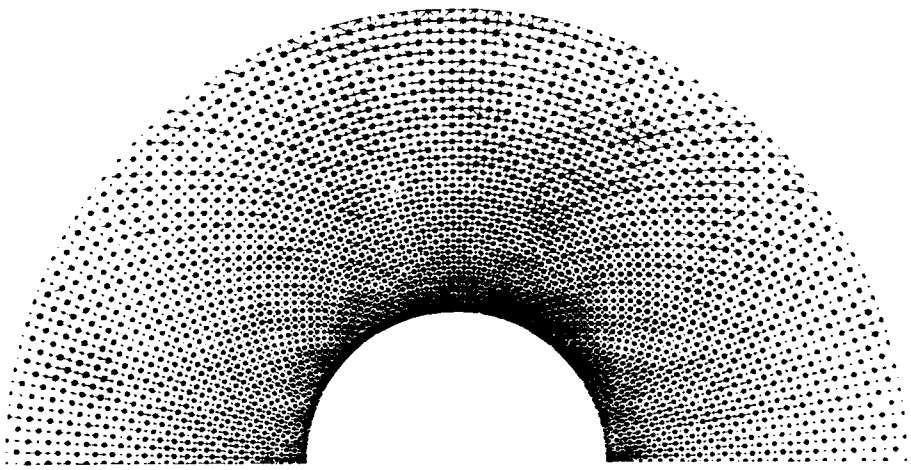


Figure 5. Finite element grid for paraboloidal island (10,560 elements)

24. Figure 6 shows a comparison between the analytical solution of the long-wave equation by Hom-ma (1950) and the finite element solution (grid of Figure 5) for a 120-sec incident wave. Figure 7 also shows a comparison between a numerical solution of Equation 1 using an orthogonal collocation solution and the finite element model solution of this report. In both cases there is excellent agreement. The effect of frequency dispersion is quite significant as shown in Figure 8, where the solutions of the long-wave equation and Equation 1 are overlapped. In fact, the inclusion of frequency dispersion is much more significant a factor than is resolution of the wave form for this particular case. For example, Figure 9 shows a solution including dispersion using the 2,640-element grid. The agreement between the finite element and

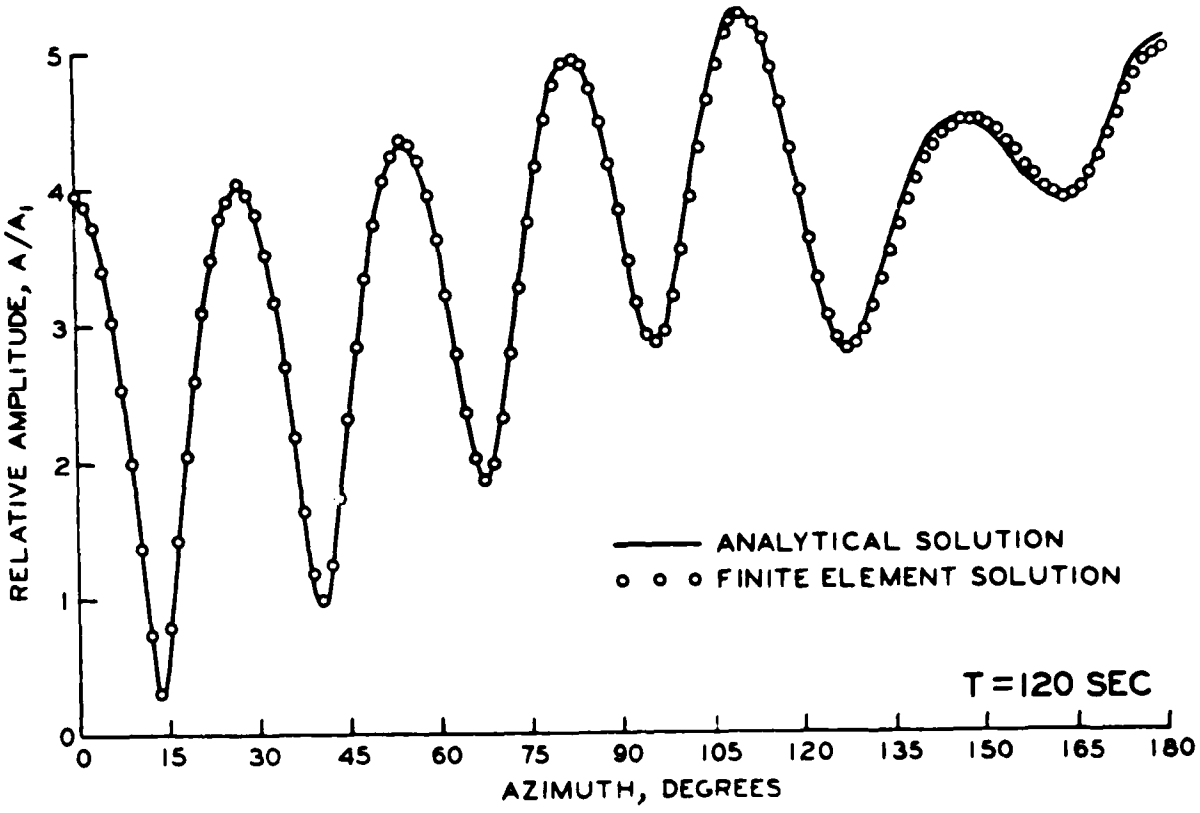


Figure 6. Solutions without dispersion (10,560-element grid)

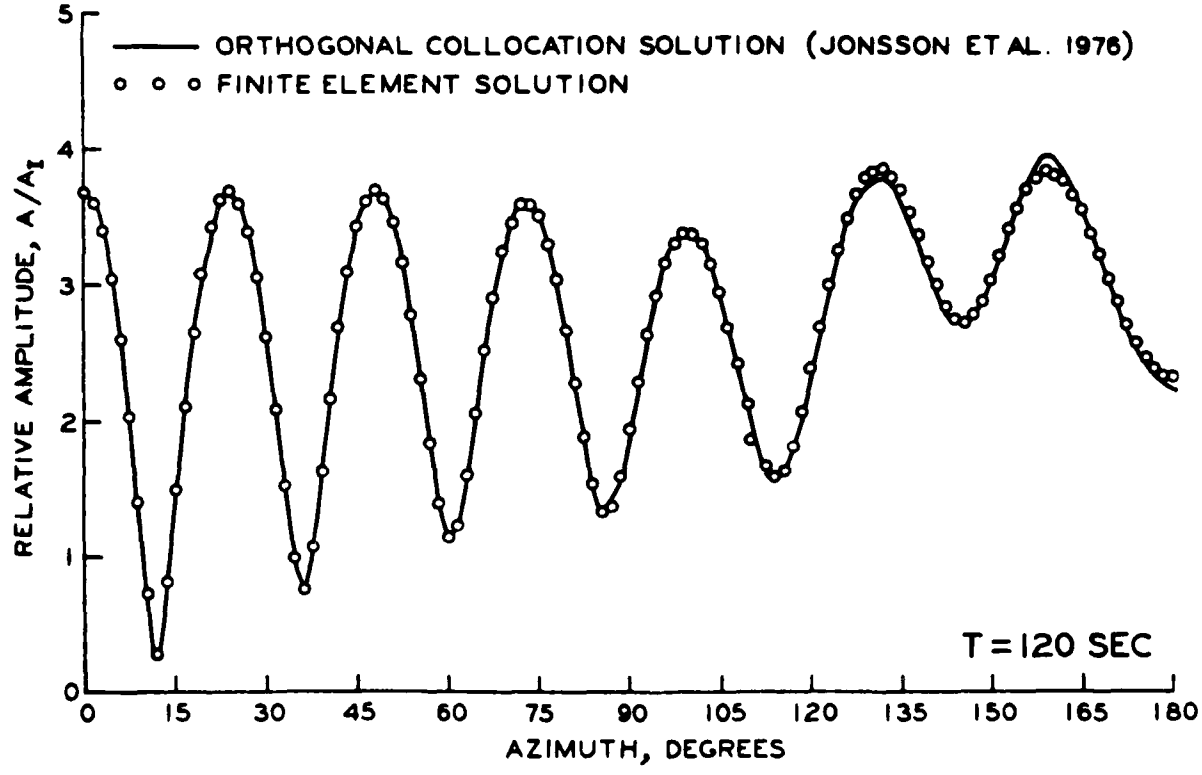


Figure 7. Solutions with dispersion (10,560-element grid)

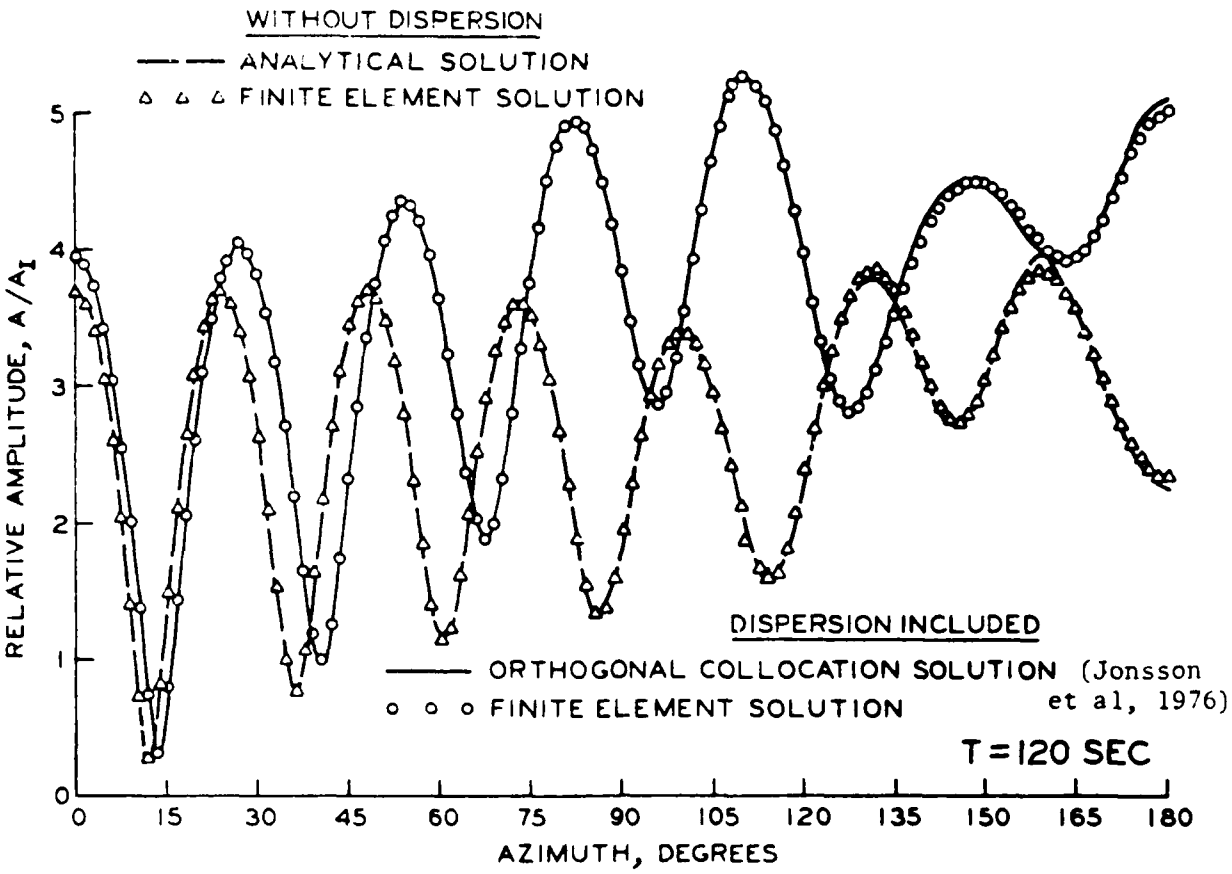


Figure 8. Effect of dispersion

collocation solutions is much better using the 2,640-element grid and including dispersion than is obtained using the 10,560-element grid without dispersion.

25. The computational time requirement to solve a problem using the 10,560-element grid is less than 1 min on a CRAY-1. This is extremely modest considering the very large size of this finite element grid. The computational time of the model is proportional to the number of nodes times the bandwidth of the coefficient matrix squared. This grid has more than 50 times the number of nodes contained in the grid used by Berkhoff (1972, 1976) and a bandwidth approximately 3.3 times greater. Thus the computational time requirement is almost 600 times greater for this grid than for Berkhoff's grid. The computer memory requirements also are very modest as a result of the partitioning of the coefficient matrix. Although the packed form of the coefficient matrix had approximately 1.7 million terms for this problem, only 45,000 terms were in central memory at any given time with the remainder in peripheral storage.

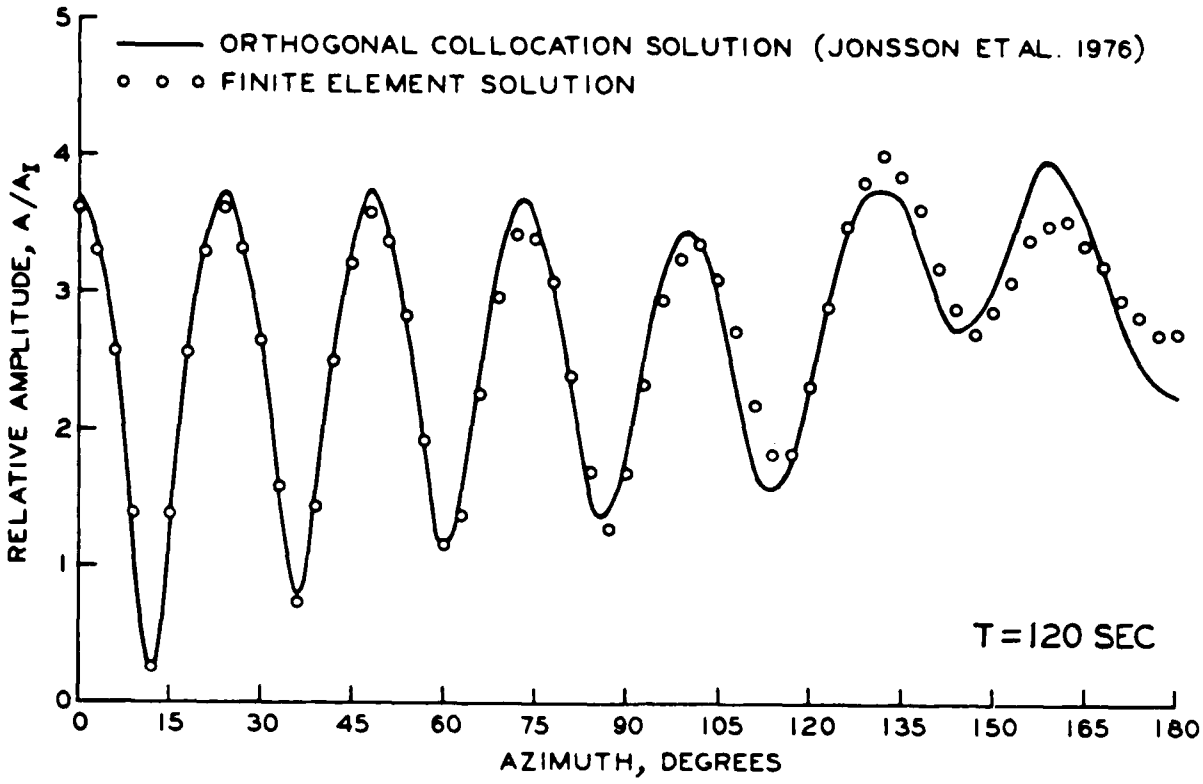


Figure 9. Solutions with dispersion (2,640-element grid)

Comparison with Three-Dimensional Numerical Model

26. Equation 1 was derived assuming that the bathymetry was slowly varying. This mild slope approximation is valid for the slow bathymetric variations of the paraboloidal shoal shown in Figure 2. However, the approximation is not appropriate for many practical problems involving the interaction of waves with man-made structures. For these problems, it is necessary to use a fully three-dimensional model such as the hybrid three-dimensional finite element developed by Yue, Chen, and Mei (1976). Of course, a three-dimensional model requires substantial computational time. Therefore it is of interest to consider how well a two-dimensional mild-slope model compares with a three-dimensional model for a problem where the mild-slope approximation is strongly violated.

27. Yue, Chen, and Mei (1976) used a three-dimensional model to consider the interaction of small amplitude plane waves with an elliptic island on a circular base. Figure 10 illustrates the problem and shows the finite

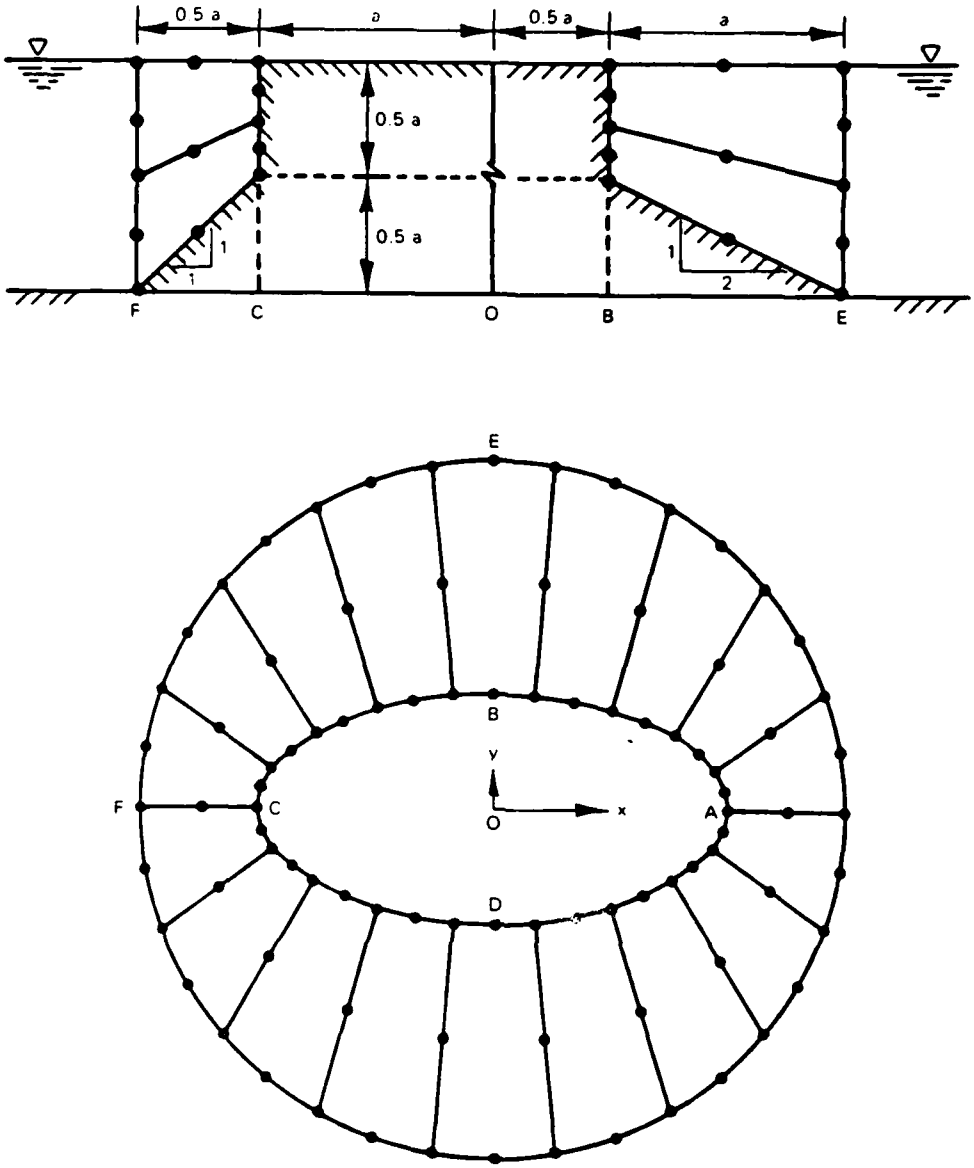


Figure 10. Elliptic island and three-dimensional grid

element grid used by Yue, Chen, and Mei (1976). Bottom side slopes vary from 1V on 1H to 1V on 2H and violate the mild-slope assumption.

28. Figure 11 shows the two-dimensional finite element grid used by the model described in this report to simulate the interaction of small amplitude plane waves with the elliptic island. Figures 12 and 13 show the interaction of waves ($k_o a = 1$ and two incident directions) with the elliptical island. The amplification factors and phases around the elliptical island calculated by the three-dimensional model of Yue, Chen, and Mei (1976) and the

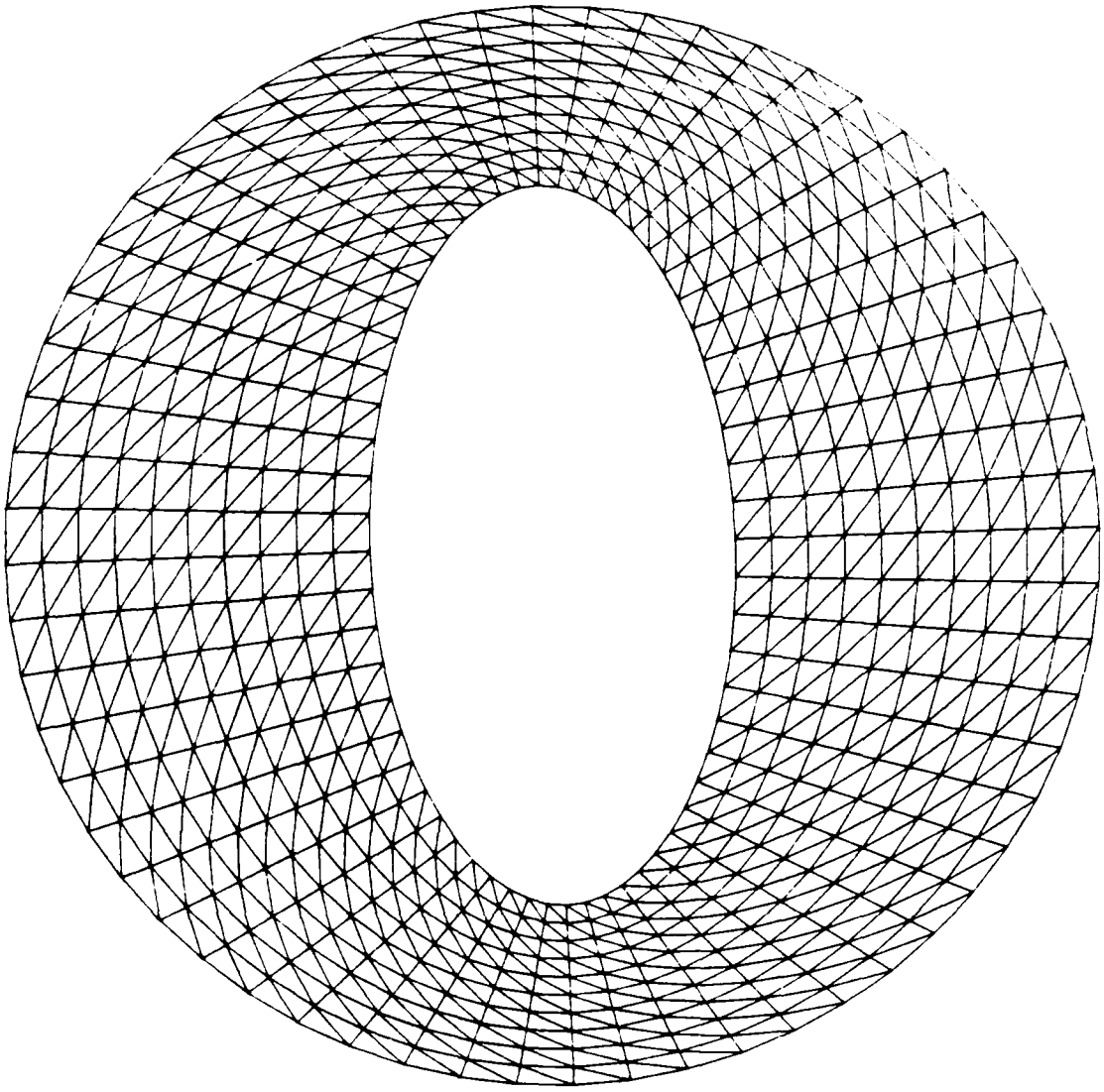


Figure 11. Finite element grid for elliptical island

two-dimensional model described in this report are shown. The maximum difference in amplification factors is not much greater than 10 percent. Similar agreement was found for greater values of $k_o a$.

29. Although the two-dimensional model cannot perfectly reproduce the three-dimensional model results, the difference may be within the accuracy requirements of many engineering applications. Of course, the computational requirements of the two-dimensional model are very modest compared with the requirements of the three-dimensional model. For the problem of waves interacting with the elliptical island, the three-dimensional model required 2.6 min of computational time on an IBM 370/168 computer for each wave period. The

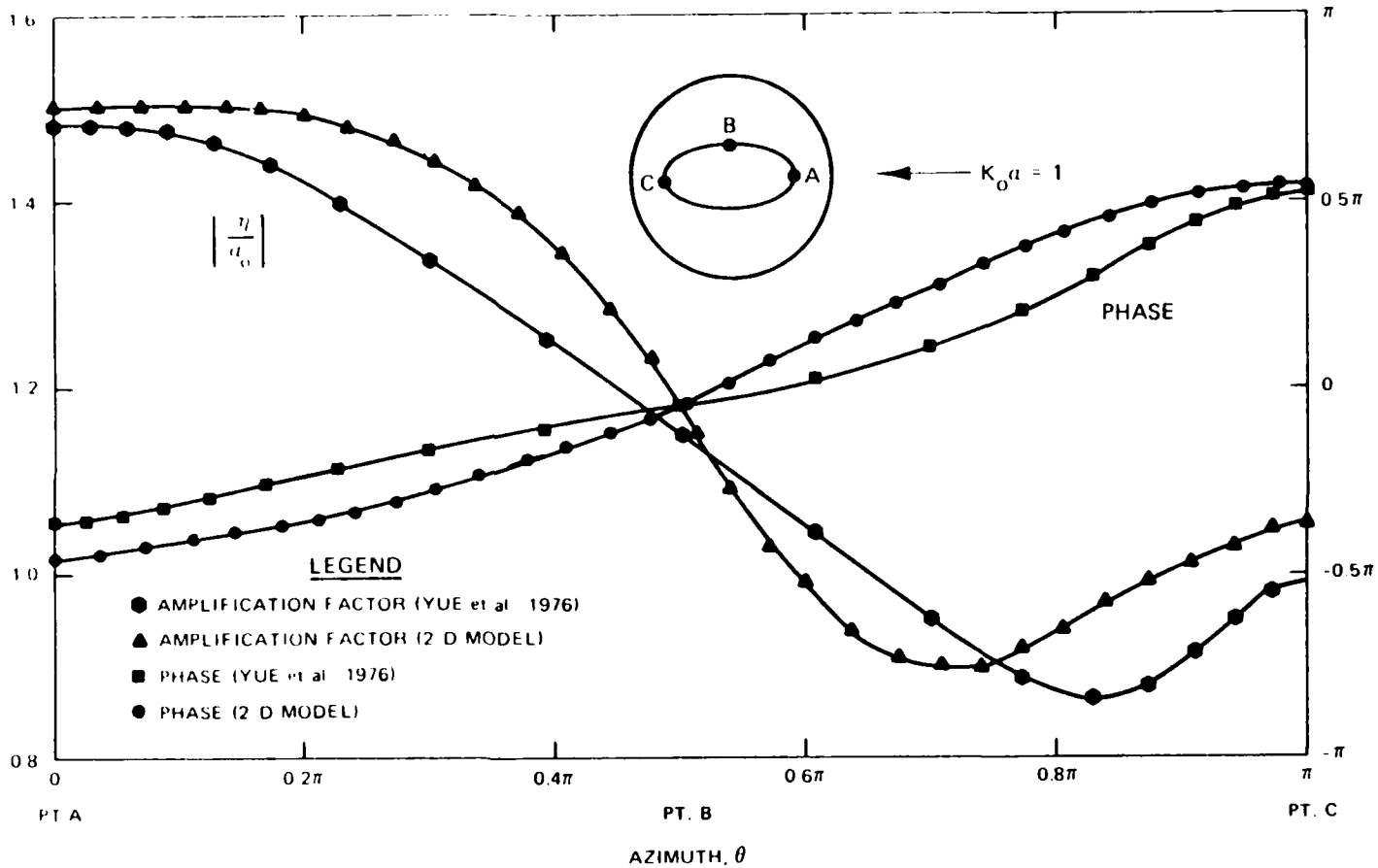


Figure 12. Comparison of model results (incident azimuth 0)

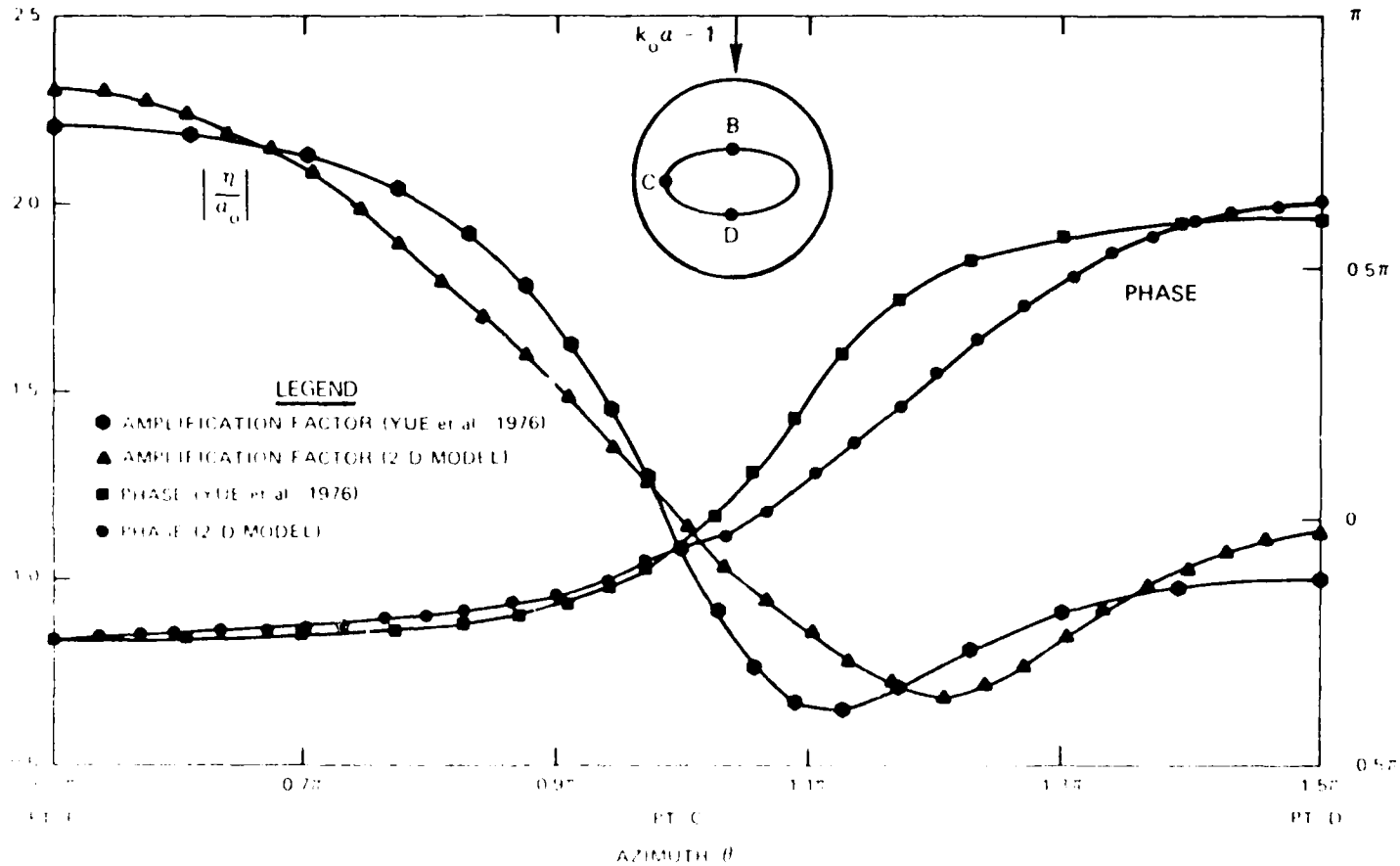


Figure 13. Comparison of model results (incident azimuth 90)

two-dimensional model required less than 2 sec of computational time on a CRAY-1 computer. Part of the difference in computational time is a consequence of the unknown relative speeds of the two computers.

30. The difference in computational time of the two- and three-dimensional models becomes more significant for larger problems. For example, Yue, Chen, and Mei (1976) estimate that approximately 6 hr of computational time on an IBM 370/168 computer would be necessary to calculate the interaction of a wave with a period of 8.5 sec or greater with a particular offshore harbor. If a resolution of 10 grid points per wavelength is maintained, the two-dimensional model would require an estimated computational time of only 0.5 to 1 min to perform the same calculation on a CRAY-1 computer.

Comparison with Laboratory Experiments

31. Putnam and Arthur (1948) performed the pioneering laboratory experiments that considered diffracted waves in the lee of an impermeable breakwater located in constant-depth water. Mobarek (1962) performed similar experiments except that a linear bottom slope was used in the lee of the breakwater and a very small model (72 ft^2) was used. In order to study the phenomenon of combined refraction and diffraction near structures, laboratory experiments were performed by Hales (1980). In these experiments an impermeable breakwater was located perpendicular to a straight coastline. A linear bottom slope extended from some distance in front of the breakwater to the shoreline.

32. The laboratory facility used in these tests covered an area of approximately 2,500 ft^2. The breakwater was 15 ft long and 1 in. thick. The water depth in the facility decreased from 1.0 ft to 0.0 ft over a distance of 20 ft. The sidewalls that laterally bounded the facility were curved to follow wave orthogonals. A plunger-type wave maker was used to generate small amplitude sinusoidal waves that approached the breakwater at an angle. An array of 32 parallel-wire, resistance-type sensors was used to measure the waves. The gages were supported by a large stand so that only the measuring wires of the gages were in the water (i.e., the gages did not require individual feet for support). Information was recorded and analyzed by a minicomputer.

33. One problem in simulating these laboratory tests numerically is that the waves break in the hydraulic model near the shoreline and thus dissipate

their energy. There is no mechanism to dissipate energy in the numerical model described in this paper. However, dissipation can be simulated by allowing waves to continue to propagate out of the problem area. Figure 14 illustrates schematically how this is done. The breakwater and the linear slope are numerically modeled only to the point where breaking occurs. The depth is

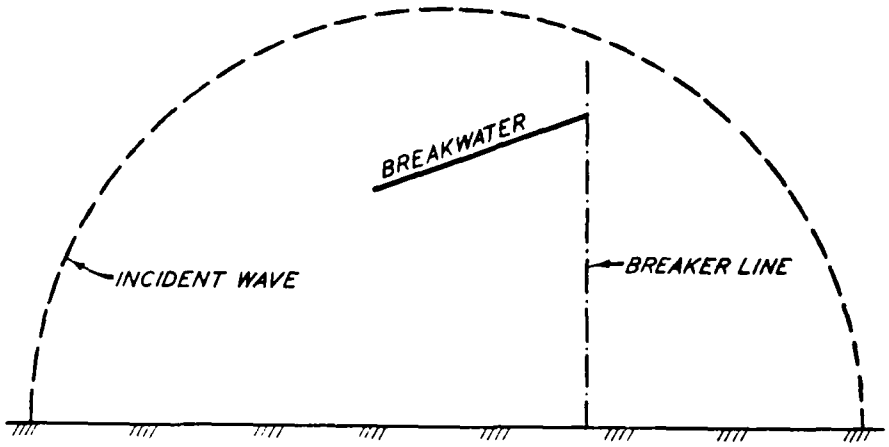


Figure 14. Wave interaction with impermeable breakwater

then increased to the depth of the semi-infinite region surrounding the inner region and the waves are allowed to radiate away from the inner region. Figure 15 shows the finite element grid used for this simulation.

34. Figure 16 shows a typical comparison between the laboratory measurements and the finite element calculations. Also shown is a uniformly valid asymptotic solution derived recently by Liu and Lozano (1979). The solution derived by Liu and Lozano (1979) is in excellent agreement with the laboratory tests. The finite element calculations agree quite well in the shadow zone with the laboratory tests. The agreement is not as good outside the shadow zone. The difference probably is attributable to the artificial increase in depth to allow the waves to radiate from the inner region. This depth transition would cause some energy to reflect back into the inner region.

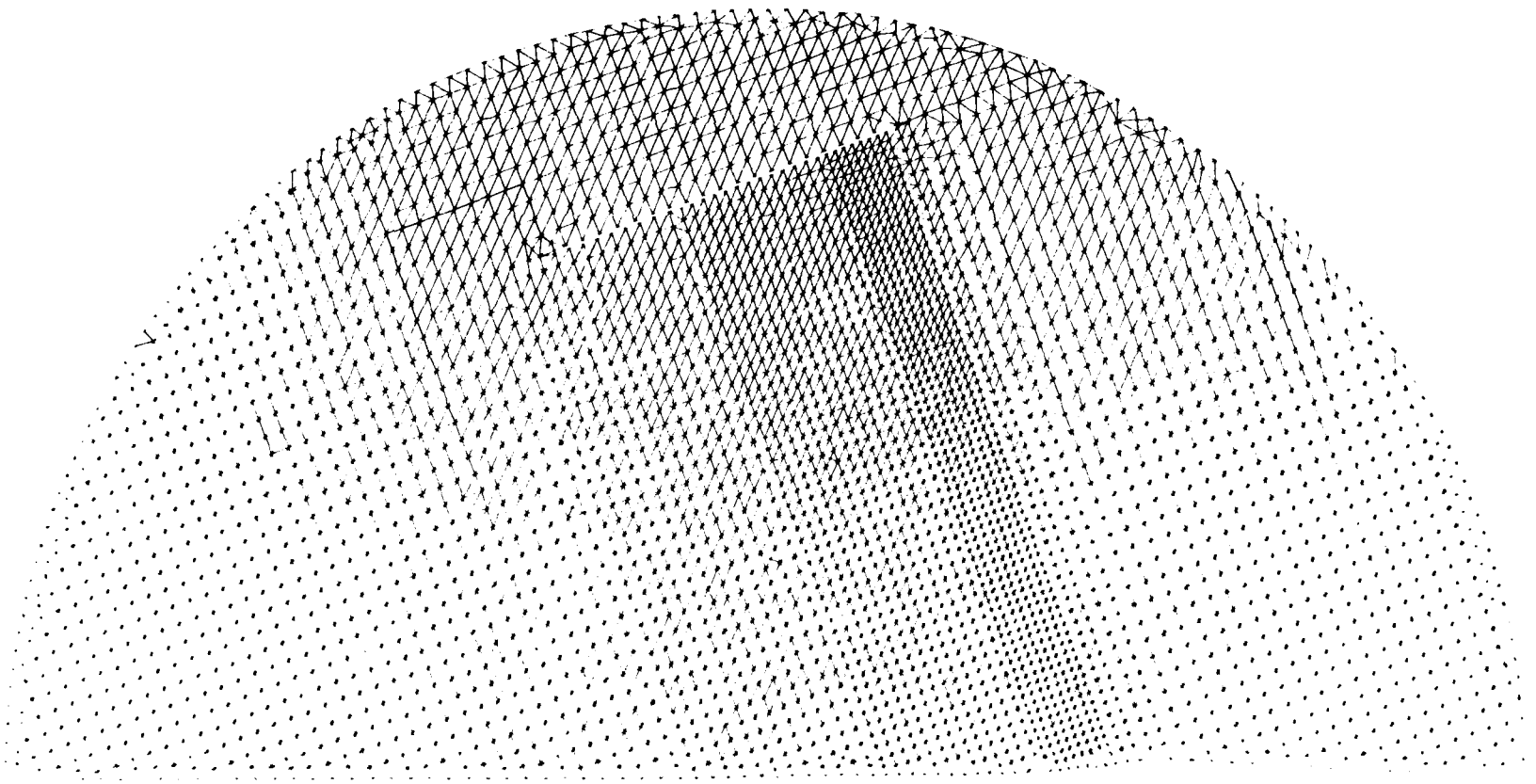


Figure 15. Finite element grid for wave interaction with breakwater

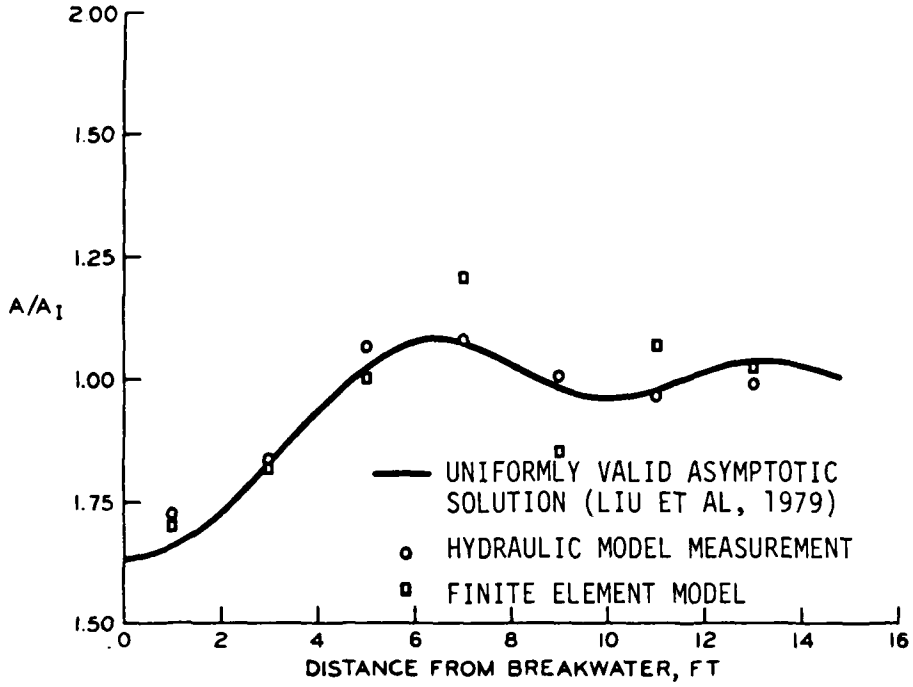


Figure 16. Comparison with laboratory experiments

PART III: INPUT DATA PREPARATION FOR FINITE ELEMENT SOLUTION

35. The following example problem is presented to demonstrate the steps and procedures for preparing a finite element grid to be used for the numerical model solution of combined refraction and diffraction. This example outlines the methodology for data manipulation on the computer facilities at the US Army Engineer Waterways Experiment Station (WES).

Data Preparation for Computer Plotting

36. Assume it has been determined that an area of interest such as outlined in Figure 17 is required to be digitized for computer processing of the hybrid finite element model. The initial determinations after the overall region has been delineated include an orientation so that the elements of the grid can be numbered in continuous fashion. Also, the nodes of the elements should be numbered in a continuous fashion (counterclockwise). The x and y coordinates of each node of the elements are digitized in continuous fashion and recorded for verification later. If the program CONVER is used for converting the measurement of digitized x and y coordinates to prototype units, it is necessary that all x and y values be positive. When the grid has been digitized on magnetic tape, the tape is assigned a number for relocation later.

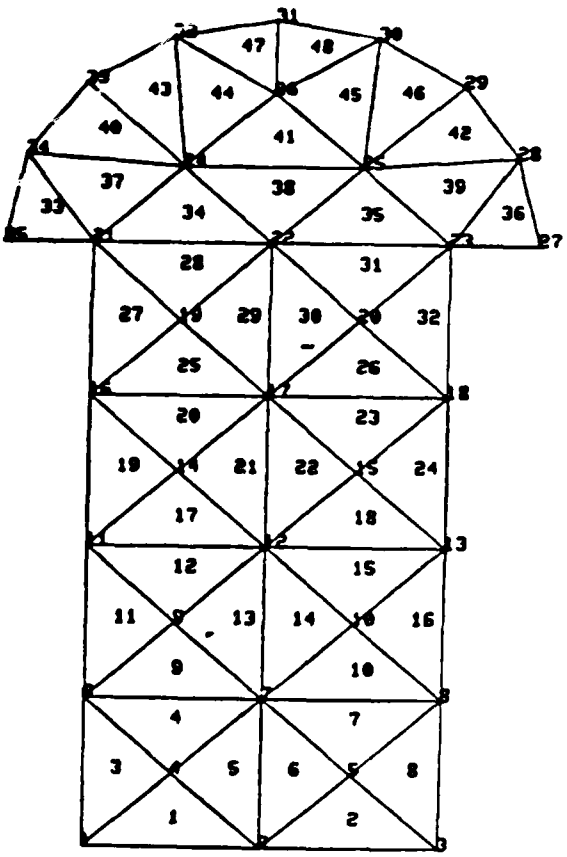


Figure 17. Example problem for demonstrating the preparation of input data for hybrid finite model

37. To convert the digitized ASCII tape to BCD tape, run program DIG1 (Figure 18). The steps for performing such an operation include:

```
 OLD DIG1
*LIST

10$$N
20$:IDENT:USERID,USER NAME
40$:OPTION:FORTRAN
60$:FORTY:ASCII
70C       PROGRAM TO CONVERT ASCII TAPES FROM DIGITIZER TO BCD TAPES
80        CHARACTER*6 X(275),Y(275)
90        CHARACTER*5 B5 /'      '/
100       CHARACTER*1 SGN1(275),SGN2(275)
110       CHARACTER SIGN*1(2) /'+','-'/
120       CHARACTER Z*68 /'
130      1                    '/
135       CALL FMEDIA(3,0)
140       NREC=0
150    10 NRECS=NREC
160       PRINT 900
170   900 FORMAT (1H1)
180       DO 5 I=1,275
190       K=I
200       READ (1,100,END=99) SGN1(I),X(I),SGN2(I),Y(I)
210   100 FORMAT (2(A1,A5))
220       NREC=NREC+1
250       WRITE (3,300) SGN1(I),X(I),SGN2(I),Y(I),Z
260   300 FORMAT (2(A1,A5),A68)
270     5 CONTINUE
280       DO 15 I=1,55
290       N=NRECS+I
300       N1=N+55
310       N2=N1+55
320       N3=N2+55
330       N4=N3+55
340       J1=I+55
350       J2=I+110
360       J3=I+165
370       J4=I+220
380       PRINT 200,N,SGN1(I),X(I),SGN2(1),Y(I),
390      1 N1,SGN1(J1),X(J1),SGN2(J1),Y(J1),
400      2 N2,SGN1(J2),X(J2),SGN2(J2),Y(J2),
410      3 N3,SGN1(J3),X(J3),SGN2(J3),Y(J3),
420      4 N4,SGN1(J4),X(J4),SGN2(J4),Y(J4)
430   200 FORMAT (5(5X,I7,2H# ,2(A1,A5)))
440    15 CONTINUE
450       GO TO 10
460    99 DO 20 I=K,275
470       SGN1(I)=SIGN(2)
480       X(I)=B5
490       SGN2(I)=SIGN(2)
500       Y(I)=B5
510    20 CONTINUE
520       DO 25 I=1,55
530       N=NRECS+I
540       N1=N+55
550       N2=N1+55
560       N3=N2+55
570       N4=N3+55
580       J1=I+55
590       J2=I+110
600       J3=I+165
610       J4=I+220
620       PRINT 200,N,SGN1(I),X(I),SGN2(I),Y(I),
630      1 N1,SGN1(J1),X(J1),SGN2(J1),Y(J1),
640      2 N2,SGN1(J2),X(J2),SGN2(J2),Y(J2),
650      3 N3,SGN1(J3),X(J3),SGN2(J3),Y(J3),
660      4 N4,SGN1(J4),X(J4),SGN2(J4),Y(J4)
670    25 CONTINUE
680       STOP
690       END
700$:EXECUTE
710$:LIMITS:30,15K,,20K
720$:TAPE9:01,X1D,,TAPEND,,ASCII TAPE NAME,,DENB
725$:FFILE:01,NLABEL,ASA9,FIXLNG/4,NSER
730$:FFILE:03,NOSLEW
740$:TAPE9:03,X3D,,,,BCD TAPE NAME
750$:MSG2:SAVE 03,USER NAME,USERID,BCD TAPE NAME
780$:ENDJOB

*
```

Figure 18. Program DIG1 to convert digitized ASCII tape to BCD tape

a. Sign on DPS1 Honeywell System at WES. The telephone numbers for both 300 Baud and 1200 Baud are 2162, 2171, 3561, and 3571.

b. Type in
OLD ROHH314/DIG1,R
CLEAR
SAVE DIG1

c. Type in
EDIT
The system will respond with a "-".

d. Type in "PS:/IDENT/" after the "-".
The system will search and print the line containing the character string "ident" as:
20$:IDENT:USERID,USERNAME

e. Type in
RVS:/USERID,USERNAME/:/YOUR USER ID, YOUR NAME/
The system will replace the old user's ID and user's NAME by the new user's ID and NAME, as desired.

f. Type in
PS:/720/
The system will respond with
720$TAPE9:01,X1D,,TAPENO,,ASCII TAPE NAME,,DEN8

g. Type in
RVS:/TAPENO,,ASCII TAPE NAME,,DEN8/:/ACTUAL TAPE NO,,ACTUAL ASCII TAPE NAME,,DENSITY OF THE TAPE/
where TAPENO has been previously assigned to the digitized magnetic tape. ASCII TAPE NAME can be (1-8) arbitrary alpha-numeric characters. Parameter DEN8 specified 800 density while DEN16 indicates 1600 density.

h. Type in
PS:/740/
The system will respond with
740$:TAPE9:03,X3D,,,,BCD TAPE NAME

i. Type in
RVS:/BCD TAPE NAME/:/ACTUAL BCD TAPE NAME/

j. Type in
PS:/750/
The system will respond with
750$:MSG2:save 03,USER NAME,USERID,BCD TAPE NAME

k. Type in
RVS:/USER NAME,USERID,BCD TAPE NAME/:/YOUR NAME,YOUR ID,ACTUAL BCD TAPE NAME/
(Note: BCD TAPE NAME should be the same name as that used in Line 740.)

l. Type in
RESAVE DIG1
The system will respond with
DATA SAVED - DIG1
End of file

m. Type in
DONE
The system will respond with a "*" (CARDIN MODE) and wait for the next command.

n. Type in
RUN
The system will respond with
SNUMB XXXXXT

38. The BCD tape contents will be saved as a permanent file by running program TP2FIL (Figure 19) in the following manner.

a. Sign on the DPS1 computer at WES

b. Type in
OLD ROHH314/TP2FIL,R
CLEAR
SAVE TP2FIL

c. Type in
EDIT

d. Type in
PS:/IDNET/

e. Replace USERID and USER NAME with actual user's ID and user's NAME
Type in
RVS:/USERID,USER NAME/:/ACTUAL USER'S ID, USER'S NAME/

f. Type in
PS:/60/
The system will respond with
60$:TAPE9:F1,X1D,,BCD TAPENO,,BCD TAPE NAME

g. Type in
RVS:/BCD TAPENO,,BCD TAPE NAME/:/ACTUAL BCD TAPE NUMBER,,ACTUAL BCD TAPE NAME/
(Note: The BCD TAPE NUMBER is obtained from the log file of the printout after program DIG1 has been run, example

```
 OLD TP2FIL
*LIST

10$$N,J
20$:IDENT:USERID,USERNAME
30$:UTILITY
40$:FUTIL:F3,F2,COPY/1R/,HOLD/F2/
50$:FUTIL:F1,F2,COPY/1F/,RWD/F2/
60$:TAPE9:F1,X1D,,BCD TAPENO,,BCDTAPE NAME
70$:FILE:F2,X2S,1L
80$:DATA:F3,,COPY
90INPUT,ASIS
100$:ENDCOPY
110$:PROGRAM:TSCONV
120$:FILE:I$,X2R
130$:FILE:OT,,1L,NEW,PF NAME
140$:ENDJOB

*
```

Figure 19. Program TP2FIL to save BCD tape as a permanent file

shown in Figure 20. The same BCD TAPE NAME should be used as that previously used in DIG1.)

h. Type in
PS:/130/
The system will respond with
130$:FILE:0T,1L,NEW,PF NAME

i. Replace the PF NAME with actual permanent file name
Type in
RVS:/PF NAME/:/ACTUAL PF NAME
(The same PF NAME as BCD TAPE NAME is suggested.)

j. Type in
RESAVE TP2FIL

k. Type in
DONE

l. Type in
RUN
The system will respond with
SNUMB XXXXXT

```
OPERATOR STARTED   WITH #10048 FOR FILE CODE 03 GE  600 BTL GE PHX 10048 10048  0001 84016   000MARTIN8CD
T4UM02 10048 WAS CHAINED TO CONTROL MASTER TAPENAME   ROCRMSC
 * NORMAL TERMINATION      AT 021343 I=5000 SW=000000000000

     START  9.294      LINES    213      PROC  0.0018       I/O   0.001     IU  5   MEMORY   15K
     STOP   9.299      LIMIT  20480      LIMIT 0.3000       LIMIT           CU  5   M*T      343
     SWAP   0.000
     LAPSE  0.005 FC D   TYPE     BUSY    IP/AT    FP/RT    IS/RC RS/RE    ADDRESS T#/PK#

                  9* R MSU500 *     42        4        4       36    36    0-12-20
                  R* R MSU500 *     11        0        0        1     1    0-12-13
                  01 D TAPE-9     1306                0/01      406     0    0-16-05 #U0180
                  03 D TAPE-9      352                0/03       49     0    0-18-05 #10048
                  P*    SYOUT
                  L* R MSU450 *   1290        0        0     1200  1200R   0-08-02 -
                  *L R MSU500 *     60        0        0      500   500R   0-12-13

      LIST     99 LINES
      RC-52   114 LINES


          ACTIVITY COST SUMMARY

    RESOURCE              RESOURCE     BILLING
    DESCRIPTION           USAGE        UNITS

 PROCESSOR (SECS)             6           10
 DISC I/O TIME (SECS)         1            1
 TAPE I/O TIME (SECS)         2            1
```

Figure 20. Example showing how BCD TAPE NUMBER is obtained from log file of printout after program DIG1 is run

39. Digitizing errors may be edited from the tape with the following procedures.

a. Sign on the DPS1 computer at WES

b. Type in
OLD PFNAME
(PFNAME is the permanent file name used in TP2FIL.)

c. Type in
EDIT
The system will respond with a "-" to indicate that it is in the EDIT mode.

d. Use the editor command to correct all errors.

e. Type in
DONE
The system will respond with a "*" and return to the CARDIN mode waiting for the next command.

f. Type in
RESAVE PFNAME
(This should be the same file name as PFNAME.)

40. Program CONVER will convert the digitized values of the x and y coordinates to prototype units (Figure 21).

a. Sign on the DPS1 computer at WES

b. Type in
OLD ROHH314/CONVER,R
CLEAR
SAVE CONVER

c. Type in
EDIT

d. Replace USERID and USER NAME by actual user's ID and NAME
Type in
RVS:/USERID,USERNAME/:/ACTUAL USER'S ID, ACTUAL USER'S NAME/

e. Type in
PS:/PARAMETER/
The system will respond with the statement
150 PARAMETER XMO=0,YMO=0,XM1=12., YM1=0.
(where XMO, YMO, XM1, and YM1 are coordinates of reference points in the prototype units)

f. Type in
RVS:/XMO=OLD VALUE,YMO=OLD VALUE/:/XMO=NEW VALUE,YMO=NEW VALUE/
RVS:/XM1=OLD VALUE,YM1=OLD VALUE/:/XMO=NEW VALUE,YM1=NEW VALUE/

g. Locate and replace with parameter N1 and N2 in Line 160
Type in
PS:/160/
RVS:/N1=OLD VALUE,N2=OLD VALUE/:/N1=NEW VALUE,N2=NEW VALUE/

h. Locate and change the format statement in Line 290
Type in
PS:/290/
RVS:/FORMAT(4(i6,2F7.2))/:/THE DESIRED FORMAT/

```
 OLD CONVER
*LIST

100$:N
110$:IDENT:USERID,USERNAME
120$:OPTION:FORTRAN
125$:USE:.GTLIT
130$:FORTY:NFORM,NLNO
150 PARAMETER XMO=0.,YMO=0.,XM1=12.,YM1=0.
160 PARAMETER N1=1851,N2=3973
165 PARAMETER N3=N2-N1+1
170 DIMENSION XP(N3),YP(N3),NTT(N3)
190 DIST=SQRT((ABS(XM1-XMO))**2.+(ABS(YM1-YMO))**2.)
192 DO 255 I=1,N3
194 NTT(I)=I+N1-1
196 255 CONTINUE
200 READ(1,100) XO,YO
210 READ(1,100) X1,Y1
220 100 FORMAT(2X,F4.0,2X,F4.0)
230 READ(1,100)(XP(I),YP(I),I=1,N3)
240 CALL CONVER(XO,YO,X1,Y1,N1,N2,XP,YP,DIST,XMO,YMO)
270 WRITE(43,200) (NTT(I),XP(I),YP(I),I=1,N3)
275 PRINT 200,(NTT(I),XP(I),YP(I),I=1,N3)
290 200 FORMAT(4(I6,2F7.2))
300 REWIND 1
310 CALL DETACH(1,,)
320 STOP
330 END
340 SUBROUTINE CONVER(XO,YO,X1,Y1,N1,N2,X,Y,DIST,XMO,YMO)
350 DIMENSION X(1),Y(1)
360 HYP=SQRT((Y1-YO)**2+(X1-XO)**2)
370 SCALE=DIST/HYP
380 SINE=((Y1-YO)*SCALE)/DIST
390 COSINE=((X1-XO)*SCALE)/DIST
395 N3=N2-N1+1
400 DO 10 I=1,N3
410 XX=((X(I)-XO)*COSINE+(Y(I)-YO)*SINE)*SCALE+XMO
420 Y(I)=(-(X(I)-XO)*SINE+(Y(I)-YO)*COSINE)*SCALE+YMO
430 X(I)=XX
440 10 CONTINUE
450 RETURN
460 END
470$:EXECUTE
480$:LIMITS:02,16K,,3000
490$:PRMFL:01,R,L,USERID/PF NAME
500$:ENDJOB

*
```

Figure 21. Program CONVER to convert digitized values of x and y coordinates to prototype values

i. Locate and replace the parameter of USERID and PFNAME in Line 490

Type in
PS:/490/
RVS:/USERID/:/ACTUAL USER'S ID/
RVS:/PFNAME/:/
(Same file name that was used in Line 130 in file TP2FIL.)

j. Type in
RUN
The system will respond with
SNUMB XXXXXT

Plotting the Finite Element Grid

41. After the finite element grid data have been prepared according to the above procedures, the grid may be plotted by use of the WES Graphics Compatibility System (GCS) (1979) subroutines. The plot file can be directed either to a Calcomp plotter or to a Tektronix 4014 terminal interactively. This program FNGRID listing is presented in Figure 22.

Steps to run FNGRID

42. Program FNGRID may be run by using the following procedures.

A. Use the XEDIT mode editor to change the total number of nodal points and elements of the grid.

a. Sign on CYBERNET (Control Data Corporation) COMPUTER SYSTEM
Telephone No. 2047 for 300 Baud
Telephone No. 2030 for 1200 Baud

b. Call out the procedure file PLOTF
Type in
GET,PLOTF/UN=CEROMO
SAVE, PLOTF

c. Use XEDIT to make necessary changes
Type in
XEDIT

d. Type in
L/IXY/
The system will respond with
Q,FNGRID.C/IXY/35/*;C/NEL/48/*

e. Change the total number of nodes (IXY) and elements (NEL) to the desired number
Type in
C/35/TOTAL NO. OF NODES/
C/48/TOTAL NO. OF ELEMENTS/

f. Option: If the user does not wish to change data file names FINDAT and FINDAT1 to other names, go to section g. Otherwise,
Type in
C/FINDAT/NEW FILE NAME/
C/FINDAT1/NEW FILE NAME/

g. Type in
E,,RL
(This command will terminate the editing mode and the file PLOTF will be replaced by the edited version.)

```
 OLD,FNGRID
/LIST
      PROGRAM GRIDFL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6,TAPE8,
     *TAPE12,TAPE13,TAPE10,TAPE66,TAPE77,TAPE9,TAPE99)
C *** IXY IS THE MAX NUMBER OF NODES
C *** NEL IS THE MAX NUMBER OF ELEMENTS
C *** N1-N4 IS THE NODE NUMBER OF EACH ELEMENT
C *** DEP IS THE DEPTH AT EACH NODE
C *** IC1=0  PLOT NODE AND ELEMENT NUMBERS
C *** IC1=1  PLOT ELEMENTS NUMBERS.
C *** IC1=2  NO NUMBERS PLOTTED
C *** IC2=0  NO DEPTH DATA OR PLOT
C *** IC2=1  READ AND PLOT DEPTH DATA
C *** XTR,YTR=X AND Y COORDINATES OF THE CENTER OF THE SEMICIRCLE.
C *** ANGG=ANGLE IN RADIANS OF SEMICIRCLE DIAMETER FROM GRID X-AXIS.
C *** RR=RADIUS OF SEMICIRCLE IN FEET
C *** NDEV=0 PLOT FILE IS DIRECTED TO THE CALCOMP PLOTTER.
C *** NDEV=1 PLOT FILE IS DIRECTED TO THE TEKTRONIX 4014 TERMINAL.
C *** NODR=NO. OF POINTS ON THE CIRCUMFERENCE OF THE SEMICIRCLE (OR CIRCLE).
C *** NCL=1 SEMICIRCLE GRID IS USED.
C *** NCL=0 FULL CIRCLE GRID IS USED.
C *** FACT IS THE FACTOR USED TO INCREASE OR DECREASE THE ORIGIONAL PLOT SIZE.
C *** FACT=1.0 IF NDEV=1
C *** WXMAX= THE RIGHT-MOST WINDOW BOUNDARY. (SEE GCS MANUAL)
C *** WXMIN= THE LEFT-MOST WINDOW BOUNDARY. (SEE GCS MANUAL)
C *** WYMAX= THE TOP-MOST WINDOW BOUNDARY. (SEE GCS MANUAL)
C *** WYMIN= THE BOTTOM-MOST WINDOW BOUNDARY. (SEE GCS MANUAL)
C *** XLO= THE X COORDINATE OF THE LOWER LEFT CORNER OF THE
C          FIRST CHARACTER OF THE OUTPUT. (SEE GCS MANUAL)
C *** YLO= THE Y COORDINATE OF THE LOWER LEFT CORNER OF THE
C          FIRST CHARACTER OF THE OUTPUT. (SEE GCS MANUAL)
C *** RTITLE= TITLE OF THE PLOT
C *** NAME1= IDENTIFIER USED FOR THE PROJECT
C *** USE PROC. FILE 'PLOTF' TO SET DESIRED DIMENSIONS.
C
C
      COMMON LDUM(2000)
      DIMENSION X(IXY),Y(IXY),DEP(IXY)
      DIMENSION K(NEL),N1(NEL),N2(NEL),N3(NEL),N4(NEL)
      DIMENSION XAVE(NEL),YAVE(NEL)
      DIMENSION NAME1(10)
      DIMENSION RTITLE(8)
      READ(12,10) NDEV,NODR,NCL,FACT
   10 FORMAT(3I5,F5.1)
      IR1='R1'
      IV1=0
      IV2=1
      CALL SETJCI(IR1,IV1)
      IF(NDEV.EQ.1) CALL SETJCI(IR1,IV2)
      READ(12,15) NAME1
   15 FORMAT(10A6)
      NSPACE=NODR-1
      READ(12,20) SCALE,IC1,IC2
   20 FORMAT(F10.2,2I10)
      READ(12,40) XTR,YTR,ANGG,RR
   40 FORMAT(4F10.2)
      READ(13,50) (X(I),Y(I),I=1,IXY)
   50 FORMAT(3(6X,2F8.3))
      PAI=3.1416
      IF(NCL.EQ.1) DA2=PAI/NSPACE
      IF(NCL.EQ.0) DA2=2.*PAI/NODR
      NNOD=IXY
      NODR2=NNOD-NODR
```

Figure 22. Program FNGRID for plotting the finite element solution (Sheet 1 of 3)

```
      WRITE(8,95)
      WRITE(8,105) NAME1
      WRITE(8,96)
   95 FORMAT(//,10X'*****************************************************')
  105 FORMAT(/,20X,10A6,/)
   96 FORMAT(10X,'*****************************************************',/)
C *** K(I) USE TEMPORARY
      DO 60 I=1,IXY
   60 K(I)=I
      READ(9,70) (DEP(I),I=1,NNOD)
   70 FORMAT(16F5.2)
      WRITE(8,115)
  115 FORMAT(7X,'NODE',11X,'X-COOD',11X,'Y-COOD',11X,'DEPTH')
      WRITE(8,90) (K(I),X(I),Y(I),DEP(I),I=1,IXY)
   90 FORMAT(I10,2X,F15.2,2X,F15.2,2X,F15.2)
      NODB1=IXY-NODB+1
      N=IXY
      M=NEL
      DO 130 I=1,N
      X(I)=X(I)*SCALE
      Y(I)=Y(I)*SCALE
  130    CONTINUE
C *** PRINT INTERMEDIATE DATA
  140 FORMAT(1X,12A6)
      WRITE(8,150)N,M,YR,SCALE
  150 FORMAT(1X,12HNO OF POINTS,I5,/,1X,14HNO OF ELEMENTS,I5,/,1X,16HRAN
     *GE OF Y-COORD,F10.0,/,1X,22HSCALE FACTOR (IN/FEET),E10.3)
C *** CALCULATE ELEMENT CENTROIDS
      WRITE(8,160)
  160 FORMAT(//,7HELEMENT,1X,4HN(1),2X,4HN(2),2X,4HN(3),2X,7HELEMENT,1X,
     *4HN(1),2X,4HN(2),2X,4HN(3),2X,7HELEMENT,1X,4HN(1),2X,4HN(2),2X,4HN
     *(3))
      READ(9,170)  (N1(I),N2(I),N3(I),I=1,NEL)
  170 FORMAT(3(5X,3I5))
  180 FORMAT(4I6,2X,4I6,2X,4I6)
      WRITE(8,130)(I,N1(I),N2(I),N3(I),I=1,NEL)
      WRITE(10,250)(X(I),I=1,IXY)
      WRITE(10,250)(Y(I),I=1,IXY)
      WRITE(10,170)(N1(I),N2(I),N3(I),I=1,NEL)
      WRITE(10,250) (DEP(I),I=1,NNOD)
  250 FORMAT(8F10.2)
C *** READ IN DATA FOR X-RANGE AND Y-RANGE  ***
      READ(12,251) WXMAX,WXMIN,WYMAX,WYMIN
  251 FORMAT(4F10.1)
C *** READ IN DATA TO LABEL THE PLOTC ***
      READ(12,41) XLO,YLO
   41 FORMAT(2F6.2)
      READ(12,42) RTITLE
      WRITE(8,42) RTITLE
   42 FORMAT(8A10)
      DO 35 I=1,M
   35 N4(I)=0
      IF(IC1.GE.2) GO TO 220
      DO 200 I=1,M
      NN1=N1(I)
      NN2=N2(I)
      NN3=N3(I)
      NN4=N4(I)
      XAVE(I)=X(NN1)+X(NN2)+X(NN3)
      YAVE(I)=Y(NN1)+Y(NN2)+Y(NN3)
      IF(N4(I).GT.0) GO TO 210
      XAVE(I)=XAVE(I)/3.
      YAVE(I)=YAVE(I)/3.
      GO TO 200
  210 XAVE(I)=(XAVE(I)+X(NN4))/4.
      YAVE(I)=(YAVE(I)+Y(NN4))/4.
```

Figure 22. (Sheet 2 of 3)

```
  200 CONTINUE
  220 CALL USTART
      CALL UPSET('SPEED',120.)
      CALL UERASE
      CALL UBELL
C     CALL UPSET('TERMINATOR',';')
      IF(NDEV.EQ.1) GO TO 225
      CALL USET('ALTE')
      CALL UERASE
  225 CONTINUE
      CALL USET('DEVICE')
      CALL FACTOR(FACT)
      CALL UDAREA(0.0,14.0,0.5,10.)
      CALL USET('VIRTUAL')
      CALL UWINDO(WXMAX,WXMIN,WYMAX,WYMIN)
      DO 25 I=1,M
      XI=I
C *** MOVE PEN TO FIRST ELEMENT NODE
      M1=N1(I)
      M2=N2(I)
      M3=N3(I)
      M4=N4(I)
      CALL USET('NOAXES')
      CALL USET('LINE')
      CALL UMOVE(X(M1),Y(M1))
      CALL UPEN(X(M2),Y(M2))
      CALL UMOVE(X(M2),Y(M2))
      CALL UPEN(X(M3),Y(M3))
      CALL UMOVE(X(M3),Y(M3))
      IF(M4.EQ.0) GO TO 26
      CALL UPEN(X(M4),Y(M4))
      CALL UMOVE(X(M4),Y(M4))
   26 CALL UPEN(X(M1),Y(M1))
   25 CONTINUE
      IF(IC1.EQ.2) GO TO 27
      DO 275 I=1,M
      ELE=I
      CALL UMOVE(XAVE(I),YAVE(I))
      CALL USET('SMALL')
      CALL USET('INTEGER')
      CALL UPRINT(XAVE(I),YAVE(I),ELE)
  275 CONTINUE
      IF (IC1.GE.1)  GO TO 27
      DO 270 I=1,N
      EN=I
      CALL UMOVE(X(I),Y(I))
      CALL USET('SMALL')
      CALL USET('INTEGER')
      CALL UPRINT(X(I),Y(I),EN)
  270 CONTINUE
   27 CONTINUE
       IF (IC2.EQ.0) GO TO 116
      CALL USET('HARDWARE')
      CALL USET('SMALL')
      DO 117 I=1,NNOD
      CALL USET('REAL')
      CALL UPRINT(X(I),Y(I),DEP(I))
  117 CONTINUE
  116 CONTINUE
C *** LABEL PLOT ***
      CALL USET('VIRTUAL')
      CALL USET('LARGE')
      CALL USET('TEXT')
      CALL UPRINT(XLO,YLO,RTITLE)
      CALL UEND
      WRITE(3,1000)
 1000 FORMAT(1X,'PLOTTING COMPLETE')
      STOP
      END
```

Figure 22. (Sheet 3 of 3)

B. Use procedure file to run program FNGRID.

a. Type in
OLD,PLOTF
BEGIN,RUNPL,PLOTF
The system will respond by printing out the statements which are edited by "Q" statement in file PLOTF

b. Answer the first question of PROGRAM FILE NAME
Type in
FNGRID1

c. Answer the question of TYPES (X,N,B) OF FILE
Type in
S

d. Answer the question of DEVICE
Type in
DR4 (if Calcomp plotter is chosen)
TK4 (if Tektronix 4014 terminal is chosen)
(Note: Computer will begin to plot on the screen if TK4 is chosen; computer will respond with more questions if DR4 is chosen.)

e. Answer the question of USER'S ID
Type in
CEROXX (the user's ID)
The system will respond with
TAPE 99 ROUTED THRU AJZZ123
(where AJZZ123 is the job name of the plot file)

f. Call (Ext 3442) the operator of the COPE terminal at the WES Automatic Data Processing (ADP) Center to retrieve the plot file. (Note: TAPE10 is saved and replaced by file name FINDAT after the run, and will be used as data file to run program FINITE.)

Data preparation information

43. The variable names and format allocations for program FNGRID are as follows.

DATA SET NO. 1

CARD NO. 1

Col.	Format	Variable Name	Description
1 - 5	I5	NDEV	NDEV=0, plot file is directed to the Calcomp plotter NDEV=1, plot file is directed to the Tektronix 4014 terminal
6 - 10	I5	NODR	Number of points on circumference
11 - 15	I5	NCL	NCL=0, full circle grid is used NCL=1, semicircle grid is used

Col.	Format	Variable Name	Description
16 - 20	F5.1	FACT	Factor used to change original plot size. (Note: If NDEV-1, then FACT=1.0.)
			CARD NO. 2
1 - 60	10A6	NAME1	Identifier used for the project
			CARD NO. 3
1 - 10	F10.2	SCALE	SCALE=1 and 2, x and y coordinates of the nodes are in feet and inches, respectively
11 - 20	I10	IC1	IC1=0, plot node numbers and element numbers IC1=1, plot element numbers only IC1=2, no number plotted
21 - 30	I10	IC2	IC2=0, no depth data on plot IC2=1, read and plot depth data
			CARD NO. 4
1 - 10	F10.2	XTR	X - coordinate of the center of semicircle
11 - 20	F10.2	YTR	Y - coordinate of the center of semicircle
21 - 30	F10.2	ANGG	Angle of the semicircle (radians)
31 - 40	F10.2	RR	Radius of the semicircle (feet)
			CARD NO. 5
1 - 10	F10.1	WXMAX	The rightmost window boundary (See GCS 1979 manual)
11 - 20	F10.1	WXMIN	The leftmost window boundary (See GCS 1979 manual)
21 - 30	F10.1	WYMAX	The topmost window boundary (See GCS 1979 manual)
31 - 40	F10.1	WYMIN	The bottommost window boundary (See GCS 1979 manual)
			CARD NO. 6
1 - 6	F6.2	XLO	x-coordinate of the lower left corner of the first character of the output (See GCS 1979 manual)
7 - 12	F6.2	YLO	y-coordinate of the lower left corner of the first character of the output (See GCS 1979 manual)

Col.	Format	Variable Name	Description
			CARD NO. 7
1 - 80	8A10	RTITLE	Title of plot
			DATA SET NO. 2
			CARD NO. 1

(Note: This data set is obtained after the program CONVER is run.)

1 - 6	Blanks		
7 - 14	F8.3	X(1)	x-coordinate of node 1
15 - 22	F8.3	Y(1)	y-coordinate of node 1
23 - 28	Blanks		
29 - 36	F8.3	X(2)	x-coordinate of node 2
37 - 44	F8.3	Y(2)	y-coordinate of node 2
45 - 50	Blanks		
51 - 58	F8.3	X(3)	x-coordinate of node 3
59 - 66	F8.3	Y(3)	y-coordinate of node 3

(Note: Six values on each card. Total number of x and y values should equal the total number of nodes.)

DATA SET NO. 3

CARD NO. 1

1 - 5	F5.1	DEP(1)	Depth in feet at each nodal point
6 - 10	F5.1	DEP(2)	(Note: Sixteen values on each card, and use as many cards as necessary.)
.			
.			
.			
.			
76 - 80	F5.1	DEP(16)	

DATA SET NO. 4

CARD NO. 1

1 - 5	I5	K(1)	Dummy variable used as an index number for the element
6 - 10	I5	N1(1)	First nodal point of element 1
11 - 15	I5	N2(1)	Second nodal point of element 1
16 - 20	I5	N3(1)	Third nodal point of element 1

Col.	Format	Variable Name	Description
21 - 25	I5	K(2)	Dummy variable
26 - 30	I5	N1(2)	First nodal point of element 2
31 - 35	I5	N2(2)	Second nodal point of element 2
36 - 40	I5	N3(2)	Third nodal point of element 3
41 - 45	I5	E(3)	Dummy variable
46 - 50	I5	N1(3)	First nodal point of element 3
51 - 55	I5	N2(3)	Second nodal point of element 3
56 - 40	I5	N3(3)	Third nodal point of element 3

(Note: Twelve values on each card for as many cards as needed. The order of N1, N2, and N3 of each element is chosen by following a counterclockwise direction.)

Example run

44. The finite element grid of Figure 17 will be processed as a typical representative example of the capability of the programs.

A. Use XEDIT mode to set the total number of nodes to 35 and total number of elements to 48 (Figure 23).

```
 GET,PLOTF/UN=CEROMO
/SAVE,PLOTF
/OLD,PLOTF
/XEDIT
 XEDIT 3.1.00
?? L/Q,FNGRID/
Q,FNGRID.C/IXY/30/*;C/NEL/40/*
?? C/30/35/
Q,FNGRID.C/IXY/35/*;C/NEL/40/*
?? C/40/48/
Q,FNGRID.C/IXY/35/*;C/NEL/48/*
?? E,,RL
PLOTF    REPLACED
PLOTF    IS A LOCAL FILE
AEB ,       0.254UNTS.
/
```

Figure 23. Using XEDIT mode to set total number of nodes and elements

B. Run program FNGRID

a. Plot file is directed to the Tektronix 4014 terminal (Figure 24).

```
OLD,PLOTF
/BEGIN,RUNPL,PLOTF,FD1,FD2,FD3.
C *** 35 IS THE MAX NUMBER OF NODES
      DIMENSION X(35),Y(35),DEP(35)
      READ(13,50) (X(I),Y(I),I=1,35)
      NNOD=35
      DO 60 I=1,35
      WRITE(8,90) (K(I),X(I),Y(I),DEP(I),I=1,35)
      NODB1=35-NODR+1
      N=35
      WRITE(10,250)(X(I),I=1,35)
      WRITE(10,250)(Y(I),I=1,35)
END OF FILE
C *** 48 IS THE MAX NUMBER OF ELEMENTS
      DIMENSION K(48),N1(48),N2(48),N3(48),N4(48)
      DIMENSION XAVE(48),YAVE(48)
      N=48
      READ(9,170)  (N1(I),N2(I),N3(I),I=1,48)
      WRITE(8,180)(I,N1(I),N2(I),N3(I),I=1,48)
      WRITE(10,170)(N1(I),N2(I),N3(I),I=1,48)
END OF FILE
 ENTER PROGRAM FILENAME
? FNGRID1
 ENTER S FOR SOURCE FILE WITH NO LINE NUMBERS
 ENTER N FOR SOURCE FILE WITH LINE NUMBERS
 ENTER B FOR BINARY FILE
? S
 DEVICE-
? TK4
```

Figure 24. Directing plot file to Tektronix 4014 terminal

b. Plot file is directed to the Calcomp plotter (Figure 25).

```
OLD,PLOTF
/BEGIN,RUNPL,PLOTF,FD1,FD2,FD3.
C *** 35 IS THE MAX NUMBER OF NODES
      DIMENSION X(35),Y(35),DEP(35)
      READ(13,50) (X(I),Y(I),I=1,35)
      NNOD=35
      DO 60 I=1,35
      WRITE(8,90) (K(I),X(I),Y(I),DEP(I),I=1,35)
      NODB1=35-NODR+1
      N=35
      WRITE(10,250)(X(I),I=1,35)
      WRITE(10,250)(Y(I),I=1,35)
END OF FILE
C *** 48 IS THE MAX NUMBER OF ELEMENTS
      DIMENSION K(48),N1(48),N2(48),N3(48),N4(48)
      DIMENSION XAVE(48),YAVE(48)
      N=48
      READ(9,170)  (N1(I),N2(I),N3(I),I=1,48)
      WRITE(8,180)(I,N1(I),N2(I),N3(I),I=1,48)
      WRITE(10,170)(N1(I),N2(I),N3(I),I=1,48)
END OF FILE
 ENTER PROGRAM FILENAME
? FNGRID1
 ENTER S FOR SOURCE FILE WITH NO LINE NUMBERS
 ENTER N FOR SOURCE FILE WITH LINE NUMBERS
 ENTER B FOR BINARY FILE
? S
 DEVICE-
? DR4
 ENTER YOUR USER NUMBER
? CER0M8
 TAPE99 ROUTED THRU JOB AJZZ324
CALL DOT GRIFFIN (TEL.3442) & SPECIFY 3-BIT SOFTWARE
REVERT.
/
```

Figure 25. Directing plot file to Calcomp plotter terminal

C. Listing of data files (FD1, FD2, FD3) (Figure 26).

```
 OLD,FD1
/LIST
    0     9     1  1.0
SAMPLE PROBLEM
      1.00          0          0
      0.00       0.00       0.00       9.00
     -35.0       35.0      -40.0       15.0
 -8.00-37.00
FINITE ELEMENT GRID\
```

Figure 26a. Listing of data file FD1

```
 OLD,FD2
/LIST
     -8.000 -32.000          0.000 -32.000          8.000 -32.000
     -4.000 -28.000          4.000 -28.000         -8.000 -24.000
      0.000 -24.000          8.000 -24.000         -4.000 -20.000
      4.000 -20.000         -8.000 -16.000          0.000 -16.000
      8.000 -16.000         -4.000 -12.000          4.000 -12.000
     -8.000  -8.000          0.000  -8.000          8.000  -8.000
     -4.000  -4.000          4.000  -4.000         -8.000   0.000
      0.000   0.000          8.000   0.000         -4.000   4.000
      4.000   4.000          0.000   8.000         12.000   0.000
     11.000   4.600          8.500   8.500          4.600  11.100
      0.000  12.000         -4.600  11.100         -8.500   8.500
    -11.100   4.600        -12.000   0.000
```

Figure 26b. Listing of data file FD2

```
 OLD,FD3
/LIST
 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
 0.25 0.25 0.25
        1    2    4         2    3    5         1    4    6
        4    7    6         2    7    4         2    5    7
        5    8    7         3    8    5         6    7    9
        7    8   10         6    9   11         9   12   11
        7   12    9         7   10   12        10   13   12
        8   13   10        11   12   14        12   13   15
       11   14   16        14   17   16        12   17   14
       12   15   17        15   18   17        13   18   15
       16   17   19        17   18   20        16   19   21
       19   22   21        17   22   19        17   20   22
       20   23   22        18   23   20        35   21   34
       21   22   24        22   23   25        23   27   28
       21   24   34        22   25   24        23   28   25
       34   24   33        24   25   26        25   28   29
       24   32   33        24   26   32        25   30   26
       25   29   30        26   31   32        26   30   31
```

Figure 26c. Listing of data file FD3

The computer plot of the finite element grid originally conceived in Figure 17 is presented in Figure 27 after processing by the programs DIG1, TP2FIL, CONVER, and FNGRID.

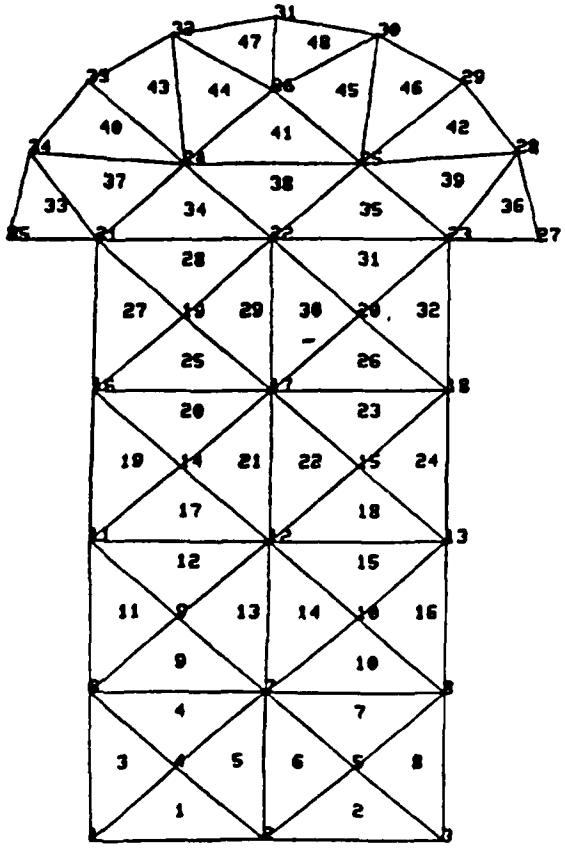


Figure 27. Computer plot of finite element grid

PART IV: PROGRAM FINITE

45. The propagation of periodic, small amplitude surface gravity waves over a variable depth seabed of mild slope is governed by

$$A'' (cc_g Ad) + \frac{c_g}{c} 1^2 d = 0 \qquad (1 \text{ bis})$$

Equation 1 is solved numerically by program FINITE using a hybrid finite element method originally developed by Chen and Mei (1974) to solve the diffraction Helmholtz equation in a constant depth region.

Data Preparation for Program FINITE

46. Two types of data are necessary to run program FINITE: (a) data values defined by parameter statements in the program FINITE and (b) data submitted to the program FINITE by data files. Program FINITE is presented in Figure 28.

Data values defined by parameter statements

47. The following data values are defined by parameter statements in the program FINITE.

Variable Name	Format	Description
NNOD	Integer	Total number of nodes
NELE	Integer	Total number of elements
NODE	Integer	Total number of radiation boundary nodal points
NCS*	Integer	Total number of coefficients in the expansion of radiation domain
ISPT	Integer	Number of selected elevation station (nodes)
ISPTE	Integer	Number of selected elevation station (elements)
NSYSK	Integer	NLGL x NBAND x 2 + 50 (see comment statements in program FINITE)
NSYTP	Integer	NLGL x NBAND + 100 (see comment statements in program FINITE)

(*Note: The value of NCS is not known a priori. Enough terms must be included such that additional terms have a negligible effect on the solution. NCS can be estimated by determining the argument ($2\frac{1}{2}R/L$) of the J Bessel Functions

(Committee for the Calculation of Mathematical Tables 1958). The values of different order J Bessel Functions with this argument can be determined using Bessel Function tables. When the value of a Bessel Function of higher order becomes much less than the lower order Bessel Function of the same argument, NCS is set equal to the order of the higher order Bessel Function. For $h = 1$, $k = 12$, $T = 2$ sec; $l = (\sqrt{gh})T = 11.4$ ft, $(2\pi R/L) = 6.64$. Since $J_{13}(6.64)$ is much less than $J_o(6.64)$, NCS is set equal to 13. The model should then be run with NCS equal to both 13 and 14. If the results are the same within desired tolerances, the value of NCS = 13 is confirmed.)

Data values submitted by data files

48. Data file FINDAT1 contains Data Set No. 1 through Data Set No. 3. Data file FINDAT contains Data Set No. 4 through Data Set No. 7. (FINDAT can be obtained after running the plotting program FNGRID.)

```
 OLD,FINITE
/LIST
*DECK MAIN
      PROGRAM LALB (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE4,
     1TAPE9,TAPE11,TAPE12,TAPE13,TAPE14,TAPE15,TAPE16,TAPE17,TAPE18,
     1TAPE19,TAPE20,TAPE21,TAPE22,TAPE23,TAPE24,TAPE25)
C
CCC  BARBERS POINT HARBOR OSCILLATION STUDY
C
C     ****************************************************************
C     NELE= TOTAL NO. OF ELEMENTS
C     NNOD= TOTAL NO. OF NODES
C     NODR=TOTAL NO.OF RADIATION-BOUNDARY NODAL POINTS
C     NCS =TOTAL NO.OF COEFF. IN THE EXPANSION OF RADIATION DOMAIN
C     NEQT= TOTAL NO. OF SIM. LIN. EQS. =NNOD+NODR
C*****************************************************************
      COMPLEX SYSK,SYSQ,XH,DH,SM
     *,SYTP,STMP,FTEMP
      COMMON/DB/ WK,ALPHA,D1,DA2,DA4,RR,WR,XL,WL,RKHP,HKL
      DIMENSION X(1277),Y(1277),NOD(1277),NCON(2334,3),XJ(25),
     1 XY(25),XH(25),DH(25),ANGL(13),SM(13),DD(2334),NSP(100),IPER(5)
      DIMENSION AF(1297),AT(1297),ID(20),NSPE(100)
      DIMENSION U(2334),V(2334),YC(2334),Z(3)
      COMMON/L1/SYSK(19850)
      COMMON/L2/SYSQ(1300)
      COMMON/L3/STMP(1300)
      COMMON/L4/SYTP(9950)
      DIMENSION FTEMP(9950)
      DIMENSION VJ(105),VY(105)
      DIMENSION NR(3)
      DIMENSION XK(25)
      COMMON /NEW/ NUMBLK,NLG,NLGL,L11,LS
     1 ,L4,L3
      LEVEL 2,SYSK,SYSQ,STMP,SYTP,FTEMP
      EQUIVALENCE (FTEMP(1),SYTP(1))
      DATA (IPER(J),J=1,5)/4HWAVE,4H PER,4HIOD ,4HIN S,4HECS=/
      IPC = 0
      IPMAX = 0
      READ(5,2) ID
    2 FORMAT(20A4)
      WRITE(6,3) ID
    3 FORMAT(//,35X,20A4,////)
      READ(5,5430)NBAND,NNOD,NELE,NODR,NCS,IPR,RR,ALPHA,DM,IBLKLEN
 5430 FORMAT(6I5,F10.2,F10.5,F10.2,1I6)
      NELE3=3*NELE
      NEQT=NNOD+NCS
      NODR1=NODR-1
      NUMBLK = NEQT*NBAND/IBLKLEN + 1
      PAI = 3.141592654
      D1 = 1.0
      DA2 = PAI/NODR1
      DA4 = DA2/2.
 5431 NLG=NEQT/NUMBLK
      NLGL=NEQT-(NUMBLK-1)*NLG
      NLGG=2*NLGL
      IF(NLGG.NBAND.LT.19850)GO TO 5432
      NUMBLK = NUMBLK + 1
      GO TO 5431
 5432 CONTINUE
      WRITE(6,12)NELE,NNOD,NODR,NCS,NEQT,ALPHA,DA2,DA4,RR,D1,DM
   12 FORMAT(/1X,23HTOTAL NO. OF ELEMENTS =,I5,3X,20HTOTAL NO. OF NODES
     1=,I5,3X,22HTOTAL NO. OF SOURCES =,I5,3X,30HFIRST ELEMENT W.R.T. SO
     2URCES =,I5,/1X,30HTOTAL NO. OF SIM. LINEAR EQS =,I5,4X,25HWAVE ANG
```

Figure 28. Program FINITE for solving hybrid finite element model (Sheet 1 of 23)

```
     3LE OF INCIDENCE =,F10.5,4X,29HINCREMENT IN ANG. DIRECTION = 2F10.4
     4,/1X,32HOUTEST EXTEND OF FINITE DOMAIN =,F10.2,2X,29HNORM. DEPTH O
     5F OUTER REGION =,F5.2,5X,31HAVER. DEPTH(FT) OF SEMICIRCLE =,F10.2,
     6//)
C*********************************************************************
C     WK = WAVE NO.
C     ALPHA= WAVE ANGLE OF INCIDENCE
C     D1 - WATER DEPTH OF OUTER REGION
C     DA2 INCREMENT IN ANGULAR DIRECTION
C*********************************************************************
      IF (IPR .NE. 1) GO TO 2013
      PRINT 2010
 2010 FORMAT(50X,28HINPUT DATA CHECK--ECHO PRINT
      PRINT 2014, NUMBLK,NLG,NLGL,IBCALEM,IPR,NBAND
 2014 FORMAT(//,1X,6I6)
 2013 CONTINUE
      READ(5,300)TMAX,TMIN,TDELT,IKK
      IF(IPR.EQ.0)GO TO 3020
      PRINT 300,TMAX,TMIN,TDELT,IKK
  300 FORMAT(3F10.2,I3)
 3020 XK(1) = TMIN
      DO 320 J = 2,IKK
      J1 = J - 1
      XK(J) = XK(J1) + TDELT
  320 CONTINUE
      READ(5,305)IPC,IPMAX,ISPT,IDPC,ISPTE,IDCH,IBUGA,IBUGG,IBUGL
  305 FORMAT(9I5)
       IF(IPR.EQ.0)GO TO 3010
      PRINT 305,IPC,IPMAX,ISPT,IDPC,ISPTE,IDCH,IBUGA,IBUGG,IBUGL
 3010 IF(IPC.EQ.0)GO TO 3000
      READ(5,306)(NSP(J),J=1,ISPT)
      READ(5,306)(NSPE(J),J=1,ISPTE)
  306 FORMAT(16I5)
      IF(IPR.EQ.0)GO TO 3000
      WRITE(6,306)(NSP(J),J=1,ISPT)
      WRITE(6,306)(NSPE(J),J=1,ISPTE)
 3000 IKT = IKK
      DO 99 IK = 1,IKK
      WK = (2.*PAI)/(XK(IK)*SQRT(DH*32.2))
      WRITE(6,73) XK(IK),WK
   73 FORMAT(1H1,//,25X,24HWAVE PERIOD IN SECONDS =,F10.2,24X,9HWAVE NO
     1=,E12.5,//)
      IF( IK .GT. 1) GO TO 98
      CALL INPUT2( X,Y,NOD,NCON
     1 ,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,DD,IDPC,DH,IDCH)
      CALL BAND( NCON
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2)
   98 CONTINUE
      CALL ASEMBK(X,Y,NCON,SYSK,SYSQ,SYTP,NLGL
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,DD,          NELE3
     1 ,NLG,IDPC,IBUGA)
      IF(IBUGA.EQ.0)GO TO 3050
      WRITE(6,350)
  350 FORMAT(//,1X,13HELK ASSEMBLED)
 3050 CONTINUE
      CALL GSPK(SYTP,XJ,XY,XH,DH,ANGL
     1 ,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,VJ,VY,NLGL,IBUGG)
      IF(IBUGG.EQ.0)GO TO 3060
      WRITE(6,355)
  355 FORMAT(//,1X,13HGSPK COMPLETE)
 3060 CONTINUE
      CALL LOAD(SH,SYSQ,XJ,DH,ANGL
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,IBUGL)
      IF(IBUGL.EQ.0)GO TO 3070
      WRITE(6,375)
  375 FORMAT(//,1X,13HLOAD COMPLETE)
```

Figure 28. (Sheet 2 of 23)

```
3070 CONTINUE
     CALL BANSOL (SYSK,SYSC,STMP,NLG,NBAND,NUMBLK,NEQT,
    1 NLGG,SYTP,NLGL,FTEMP)
     WRITE(6,20)
  20 FORMAT(////,45X,40(1HO)/,45X,
    *40H        THE SOLUTION OF THE SYSTEM        /,45X,40(1HO)///
    1/1X,7HELEMENT,5X,13HAMPLIFICATION,19X,8HVELOCITY,14X,7HELEMENT,5X,
    213HAMPLIFICATION,19X,8HVELOCITY,/4X,2HOR,11X,6HFACTOR,16X,
    310HCOMPONENTS,5X,9HMAGNITUDE,5X,2HOR,11X,6HFACTOR,16X,
    410HCOMPONENTS,8X,9HMAGNITUDE,/2X,4HNODE,4X,4HMAG.,8X,5HPHASE,9X,
    51HU,11X,1HV,18X,4HNODE,4X,4HMAG.,8X,5HPHASE,9X,1HU,11X,1HV,///)
     N1 = NNOD+1
     DO 22 I=1,NEQT
     AR=REAL(SYSC(I))
     AI= AIMAG(SYSC(I))
     AT(I) ATAN2(AI,AR)
     AF(I)=SQRT(AR**2+AI**2)
     AF(I)= AF(I)/2.
  22 CONTINUE
     G=32.2
     DO 4014 L=1,NELE
     DO 4008 J=1,3
     NR(J)= NCON(L,J)
4008 CONTINUE
     I1=NR(1)
     I2=NR(2)
     I3=NR(3)
     X1=X(I1)
     X2=X(I2)
     X3=X(I3)
     Y1=Y(I1)
     Y2=Y(I2)
     Y3=Y(I3)
     B1= Y3-Y2
     B2= Y1-Y3
     B3= Y2-Y1
     C1= X2-X3
     C2= X3-X1
     C3= X1-X2
     AREA=0.5*(B1*C2-B2*C1)
     FAC=2.*AREA/G
     LUM= L
     LUM1=3*LUM-2
     LUM2=3*LUM-1
     LUM3=3*LUM
     DV1L1     =-B1/FAC
     DV1L2     =-B2/FAC
     DV1L3     =-B3/FAC
     DV2L1     =-C1/FAC
     DV2L2     =-C2/FAC
     DV2L3     =-C3/FAC
     IF( AREA .GT. 0. ) GO TO 4009
     WRITE(6,4010) AREA,LUM
4010 FORMAT(1X,21HDEBUG ELMK 100, AREA=,E12.5,6H    AT,I5,11H-TH ELEMEN
    1T)
     STOP
4009 CONTINUE
     SYTP(LUM1)= (0.0,1.0)*DV2L1
     SYTP(LUM1)= SYTP(LUM1) + DV1L1
     SYTP(LUM2)= (0.0,1.0)*DV2L2
     SYTP(LUM2)= SYTP(LUM2) + DV1L2
     SYTP(LUM3)= (0.0,1.0)*DV2L3
     SYTP(LUM3)= SYTP(LUM3) + DV1L3
4014 CONTINUE
     K11=0
     SIGMA= 2.*PAI/XK(IK)
```

Figure 28. (Sheet 3 of 23)

```
      DO 250 J=1,NELE
      DO 260  L=1,3
      N=NCON(J,L)
260   Z(L)= AF(N)*2.
      U(J)=0.0
      V(J)=0.0
      DO 270 L=1,3
      K11=K11+1
      DV1=  REAL( SYTF(K11) )
      DV2= AIMAG( SYTF(K11) )
      U(J)= U(J)+DV1*Z(L)
270   V(J)= V(J)+DV2*Z(L)
      U(J)=-U(J)/SIGMA
      V(J)=-V(J)/SIGMA
 250  YC(J)=SQRT(U(J)*U(J)+V(J)*V(J))
      IF(IPC.EQ.0)GO TO 502
      WRITE(6,30)
   30 FORMAT(/,36X,61HELEVATION AND VELOCITY FOR SELECTED ELEVATION STAT
     1IONS(NODES),///)
      DO 500 I=1,ISPT,2
      II   I+1
      J = NSP(I)
      IF(II.GT.ISPT)GO TO 501
      J1 = NSP(II)
      WRITE (6,25) J,AF(J),AT(J),U(J),V(J),YC(J),J1,AF(J1),AT(J1),U(J1),
     1V(J1),YC(J1)
      GO TO 500
  501 WRITE (6,24) J,AF(J),AT(J),U(J),V(J),YC(J)
  500 CONTINUE
  502 NSK2 = 2*NSKIP
      IF(IREDPT.EQ.1)WRITE(6,505)NSKIP
  505 FORMAT(///,39X,32HELEVATION AND VELOCITY AT EVERY ,I3,19H-TH NODE
     1OR ELEMENT,///)
      DO 400 I=1,NNOD,NSK2
      II = I+NSKIP
      IF(II.GT.NNOD)GO TO 410
      WRITE (6,25) I,AF(I),AT(I),U(I),V(I),YC(I),II,AF(II),AT(II),U(II),
     1V(II),YC(II)
      GO TO 400
  410 WRITE (6,24) I,AF(I),AT(I),U(I),V(I),YC(I)
  400 CONTINUE
      NNOD1=NNOD+1
      WRITE(6,23)
      DO 414 I=NNOD1,NEQT,2
      II=I+1
      IF(II.GT.NEQT)GO TO 415
      WRITE (6,25) I,AF(I),AT(I),U(I),V(I),YC(I),II,AF(II),AT(II),U(II),
     1V(II),YC(II)
      GO TO 414
  415 WRITE (6,24) I,AF(I),AT(I),U(I),V(I),YC(I)
  414 CONTINUE
      WRITE(6,26)NSKIP
   26 FORMAT(////,43X,20HVELOCITIES FOR EVERY,I3,22H-TH REMAINING ELEMEN
     1T,///)
      NEQT1=NEQT+1
      NSK3 = 3*NSKIP
      DO 417 I = NEQT1,NELE,NSK3
      J1 = I
      J2 = I + 2*NSKIP
      IF(J2.GT.NELE)J2 = NELE
      WRITE(6,222)(J,U(J),V(J),YC(J),J=J1,J2,NSKIP)
  222 FORMAT(1X,3(I5,1X,3E12.5,2X))
  417 CONTINUE
      IF(IPC.EQ.0)GO TO 520
      WRITE(6,31)
   31 FORMAT(////,42X,            49HVELOCITY FOR SELECTED VELOCITY STATI
```

Figure 28. (Sheet 4 of 23)

```
     1ONS(ELEMENTS),///)
      DO 515 I=1,ISPTE,3
      II = I + 1
      III = I + 2
      J = NSPE(I)
      IF(II.GT.ISPTE)GO TO 516
      J1 = NSPE(II)
      IF(III.GT.ISPTE)GO TO 517
      J2 = NSPE(III)
      WRITE(6,222)J,U(J),V(J),YC(J),J1,U(J1),V(J1),YC(J1),J2,U(J2),V(J2)
     1,YC(J2)
      GO TO 515
  516 WRITE (6,223)J,U(J),V(J),YC(J)
      GO TO 515
  517 WRITE(6,221)J,U(J),V(J),YC(J),J1,U(J1),V(J1),YC(J1)
  515 CONTINUE
  221 FORMAT(1X,2(I5,1X,3E12.5,2X))
  223 FORMAT(1X,1(I5,1X,3E12.5,2X))
 2071 FORMAT(9H SOLUTION)
 2022 FORMAT(4(1X,I4,1X,2F7.3))
 2023 FORMAT(5(2X,I4,1X,F8.3))
   23 FORMAT(/,60X,13HSOURCE POINTS,/)
   24 FORMAT(1X,I5,5E12.5)
   25 FORMAT(1X,2(I5,5E12.5))
  520 IF((IPC.NE.0).OR.(IPMAX.NE.0))CALL CPUNCH(NODPT,IPMAX,AF,U,V,NNOD,
     1NELE,NEQT,XK,IK,NSP,ISPT,IPER,ID,IPC,NSPE,ISPTE,IKT)
   99 CONTINUE
      STOP
      END
*DECK CPUN
      SUBROUTINE CPUNCH(NODPT,IPMAX,AF,U,V,NNOD,NELE,NEQT,XK,IK,NSP,ISPT
     1,IPER,ID,IPC,NSPE,ISPTE,IKT)
C*********************************************************************
C     THIS SUBROUTINE PUNCHES CARDS FOR AMPLIFICATION AND VELOCITY
C*********************************************************************
      DIMENSION AF(NEQT),U(NELE),V(NELE),XK(25),ID(20),NSP(100),IPER(5),
     1AFT(5),NSPE(100),UT1(3),VT1(3)
      WRITE(6,101)
  101 FORMAT( 4H ''C )
      WRITE(6,10)ID
   10 FORMAT(20A4)
      IF(IPC.EQ.0)GO TO 999
      WRITE(6,50)ISPT,ISPTE,IKT
   50 FORMAT(1X,3I5)
      ISP = ISPT/5
      FISPT = FLOAT(ISPT)
      FISP = FISPT/5.0
      IF(FISP.GT.ISP)ISP = ISP+1
      ICOUNT = 0
      J1 = 0
      J2 = 0
      DO 110 I=1,ISP
      ICOUNT = ICOUNT +1
      J1 = J2 + 1
      J2 = J1 + 4
      IF(J2.GT.ISPT)J2 = ISPT
      N1 = 0
      DO 112 J = J1,J2
      N1 = N1 + 1
      K = NSP(J)
  112 AFT(N1) = AF(K)
      WRITE(6,111)ICOUNT,XK(IK),(AFT(N),N=1,N1)
  111 FORMAT(1X,I5,F10.5,5F13.4)
  110 CONTINUE
      ISPE = ISPTE/3
      FISPTE=FLOAT(ISPTE)
```

Figure 28. (Sheet 5 of 23)

```
      FISPE=FISPTE/3.0
      IF(FISPE.GT.ISPE)ISPE=ISPE+1
      K1 = 0
      K2 = 0
      ICT = 0
      DO 120 I=1,ISPE
      ICT = ICT + 1
      K1 = K2 + 1
      K2 = K1 + 2
      IF(K2.GT.ISPTE)K2 = ISPTE
      M1 = 0
      DO 113 K=K1,K2
      M1 = M1 + 1
      J = NSPE(K)
      UT1(M1) = U(J)
  113 VT1(M1) = V(J)
      WRITE(6,114)ICT,XK(IK),(UT1(M),VT1(M),M=1,M1)
  114 FORMAT(1X,I4,F10.5,6F11.4)
  120 CONTINUE
      WRITE(6,150)
  150 FORMAT(1X)
      WRITE(6,102)
  102 FORMAT( 4H ''L )
      IF(IPMAX.NE.0)GO TO 1000
      RETURN
  999 IF(IPMAX.EQ.0)GO TO 2000
 1000 IC = 0
      WRITE(6,1010)(IPER(I),I=1,5),XK(IK)
 1010 FORMAT(1X,5A4,F10.5)
      DO 1100 I=1,NNOD,11
      IC=IC + 1
      L1=I
      L2=L1 + 10
      IF (L2.GT.NNOD)L2 = NNOD
      WRITE(6,20)IC,(AF(L),L=L1,L2)
   20 FORMAT(1X,I4,2X,11F6.3)
 1100 CONTINUE
      DO 1200 I = 1,NELE,5
      IC = IC + 1
      L1 = I
      L2 = L1 + 4
      IF(L2.GT.NELE) L2 = NELE
      WRITE(6,30)IC,(U(L),V(L),L=L1,L2)
   30 FORMAT(1X,I4,2X,5(2F7.3))
 1200 CONTINUE
      WRITE(6,150)
 2000 WRITE(6,102)
      RETURN
      END
*DECK INPT
      SUBROUTINE INPUT2( X,Y,NOD,NCON
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,DD,IDFC,DM,IDCH)
C*********************************************************************
C     THIS READS THE INPUT DATA FOR 3-NODES TRIANGLE ELEMENTS
C*********************************************************************
      COMMON/DB/ WK,ALPHA,D1,DA2,DA4,RR,WR,XL,WL,RKHF,HKL
      DIMENSION X(NNOD),Y(NNOD),NOD(NNOD),NCON(NELE,3),DD(NELE)
C*********************************************************************
C     NOD(I)=EXTERNAL NODE NUMBER. IE, ACTUAL NODE NUMBER ASSIGNED
C     X(I), Y(I)= X-AND Y-COORDINATES OF NOD(I)
C     NCON(M,N)=M-TH-ELEMENT NODAL CONNECTIVITIES@ N=I-J-K
C*********************************************************************
      NODB1=NNOD-NODR+1
      NODB2=NNOD-NODR
      DO 18 I=NODB1,NNOD
      IF(IDFC.NE.0)DD(I)=DM
```

Figure 28. (Sheet 6 of 23)

```
      NOD(I)=I
      II I-NODB1+1
      NA=II
      AAD=(II-1)*DA2
      X(I)= RR*COS(AAD)
      Y(I)= RR*SIN(AAD)
   19 CONTINUE
      CALL GDEPTH(IDPC,DD,NNOD,NODB2,NELE,IDCH)
      DO 315 I=1,NODB2,2
      II=I+1
      IF(I.EQ.NODB2) GO TO 3314
      READ(4,316) NOD(I),X(I),Y(I),NOD(II),X(II),Y(II)
      GO TO 315
 3314 READ(4,3316) NOD(I),X(I),Y(I)
 315  CONTINUE
      IF(IDCH.EQ.1)PRINT 316,(NOD(I),X(I),Y(I),I=1,NNOD)
 316  FORMAT( 2(I10,2F10.2) )
 3316 FORMAT(  I10,2F10.2  )
      IF(IDPC.NE.0)GO TO 400
      DD(I) = DD(I)/DM
      DO 60 I=1,NELE
   60 CONTINUE
      GO TO 70
  400 DO 56 I = 1,NNOD
      DD(I) = DD(I)/DM
   56 CONTINUE
   70 DO 415 L = 1,NELE,4
      L1= L
      L2= L1 +3
      IF(L2.GT.NELE) L2= NELE
 415  READ(4,416) ((NCON(LL,J),J=1,3),LL=L1,L2)
 416  FORMAT( 4(5X,3I5) )
      IF(IDCH.EQ.1)PRINT 416,((NCON(L,J),J=1,3),L=1,NELE)
      RETURN
      END
*DECK DEPTH
      SUBROUTINE GDEPTH(IDPC,DD,NNOD,NODB2:NELE,IDCH)
C******                ***********************                    ************
C     THIS SUBROUTINE READS DEPTH AT EACH GRID NODE FROM CARDS OR USES
C     SPECIAL PROCEDURES TO READ AVER. DEPTH FOR EACH ELEMENT
C******                ***********************                    ************
      DIMENSION DD(NELE)
      IF(IDPC.NE.0)GO TO 1000
C     SPECIAL PROCEDURE FOR INPUT OF AVER DEPTH FOR EACH ELEMENT
      DO 100 I = 1,121
      DD(I) = 24.
  100 CONTINUE
      DO 101 I=122,125
      DD(I) = 24.
  101 CONTINUE
      DO 102 I=126,318
      DD(I) = 24.
  102 CONTINUE
      DO 103 I=319,336
      DD(I) = 24.
  103 CONTINUE
      DO 104 I=337,360
      DD(I) = 29.5
  104 CONTINUE
      DO 105 I=361,371
      DD(I) = 24.
  105 CONTINUE
      DO 106 I=372,385
      DD(I) = 40.
  106 CONTINUE
      DO 107 I=386,529
```

Figure 28. (Sheet 7 of 23)

```
      DD(I)=21.
  107 CONTINUE
      DO 108 I=530,2232
      DD(I) = 38.
  108 CONTINUE
      DO 109 I = 2233,2333
      DD(I) = 42.
  109 CONTINUE
      DD(2334) = 38.
      IF(IDCH.EQ.1)PRINT 2010,(DD(I),I=1,NELE)
 2010 FORMAT(//,(12F10.3))
      GO TO 2000
C     READ DEPTH AT EACH NODE FROM CARDS OF PERM FILE
  527 FORMAT(8(I5,F5.0))
 1000  READ(4,527)(I,DD(I),I=1,NODB2)
      IF(IDCH.EQ.1)PRINT 2010,(DD(I),I=1,NNOD)
 2000 RETURN
      END
*DECK BAND
      SUBROUTINE BAND( NCON
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2)
      DIMENSION NCON(NELE,3)
C*********************************************************************
C     THIS SUBROUTINE DETERMINES THE BANDWIDTH FOR TRIANGLE ELEMENTS
C*********************************************************************
      MPS=  NODR+NCS
      KMAX=1
      DO 74 L=1,NELE
      IMAX=NCON(L,1)
      IMIN=NCON(L,1)
      DO 72 K=1,3
      IF(NCON(L,K) .LE. IMIN) IMIN=NCON(L,K)
      IF(NCON(L,K) .GE. IMAX) IMAX=NCON(L,K)
   72 CONTINUE
      KK=IMAX-IMIN
      IF(KK .GT. KMAX) KMAX=KK
   74 CONTINUE
      MBAND=KMAX+1
      NBAND=AMAX0(MBAND,MPS)
       WRITE(6,84)NBAND,MBAND,MPS
   84 FORMAT(////1X,40(1HO),/12H BAND WIDTH=,I5,/27H BAND WIDTH W.R.T. E
     1LEMENT=,I5,/34H BAND WIDTH W.R.T. SOURCE + BDARY=,I5,/1X,40(1HO),/
     2)
      RETURN
      END
*DECK ASBK
      SUBROUTINE ASEMBK(X,Y,NCON,SYSK,SYSQ
     1 ,ST,NGL
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,DD,          NELE3
     1 ,NLGG,IDPC,IBUG)
C*********************************************************************
C     THIS SUBROUTINE ASSEMBLES ELK(I,J) ---ELEMENT--- INTO SYSK(I,J) --
C     - SYSTEM ---.
C     ELK= ELEMENT STIFFNESS MATRIX 3 BY 3
C     SYSK= SYSTEM STIFFNESS MATRIX NEQT BY NBAND
C*********************************************************************
      COMPLEX SYSK,SYSQ,ST
      COMMON/DB/WK,ALPHA,D1,DA2,DA4,RR,WP,XL,WL,RKHP,HKL
      DIMENSION X(NNOD),Y(NNOD),NCON(NELE,3),SYSQ(NEQT)
      COMMON /NEW/ NUMBLK,NLG,NLGL,L11,LQ,L4,L3
      DIMENSION NR(3),ELK(3,3),DD(NELE)
      DIMENSION NR(3)
      DIMENSION ST(NGL,NBAND)
C   CHANGES
      DIMENSION SYSK(NLGG,NBAND)
      LEVEL 2,SYSK,SYSQ,ST
```

Figure 28. (Sheet 8 of 23)

```
      DO 122 I=1,NEQT
  122 SYSQ(I)=(0.0,0.0)
      L2=0
C     CHANGES
      L3=NLG*NBAND*2
      LQ=NLG*2
      DO 888 IQ=1,NUMBLK
      LEN=NLG
      IF (IQ .EQ. NUMBLK) LEN=NLGL
      L2=L2+LEN
      L1=(IQ-1)*NLG+1
      L11=L1-1
      DO 12 I=1,LEN
C      CHANGES MADE
      DO 12 J=1,NBAND
      IF (IQ .NE. NUMBLK) GO TO 7176
      ST(I,J)=(0.0,0.0)
      GO TO 12
 7176 CONTINUE
      SYSK(I,J)= (0.0,0.0)
   12 CONTINUE
      DO 25 L=1,NELE
      IGO=1
      DO 14 J=1,3
      MR(J)=0
      NR(J)= NCON(L,J)
      IF ((NR(J).LT.L1).OR.(NR(J).GT.L2)) GO TO 14
      IGO=2
      MR(J)=1
   14 CONTINUE
      IF(IBUG.EQ.0)GO TO 1000
      WRITE(6,10)(NCON(L,J),J=1,3)
   10 FORMAT(1X,4H****,3I10,4H****)
 1000 GO TO (25,44),IGO
   44 I1=NR(1)

      I2=NR(2)
      I3=NR(3)
      X1=X(I1)
      X2=X(I2)
      X3=X(I3)
      Y1=Y(I1)
      Y2=Y(I2)
      Y3=Y(I3)
      IF(IDFC.NE.0)GO TO 210
      D9 = DD(L)
      GO TO 200
  210 D11=DD(I1)
      D12=DD(I2)
      D13=DD(I3)
      D9=(D11+D12+D13)/3.
      IF(L.GE.1435.AND.L.LE.1571)D9=0.5
C*****************                                        ****************
C************* SPECIAL STATEMENTS MAY BE ADDED TO COMPENSATE FOR   *********
C*************  AVERAGING PROCEDURE, FOR EXAMPLE, AS ABOVE--   *************
C***  IF(L.GE.1435.AND.L.LE.1571) D9 = 0.5                    **********
C*****************                                                 *********
  200 DIN = 1.0
       CALL ELMK(X1,X2,X3,Y1,Y2,Y3,WK,D9,DIN,ELK,AREA,L,          NELE3)
      IF(IBUG.EQ.0)GO TO 1050
      WRITE(6,15)((ELK(M,N),N=M,3),M=1,3)
   15 FORMAT(6E15.6)
 1050 DO 20 I=1,3
      DO 18 J=I,3
      IF( NR(J)-NR(I) .GE. 0 ) GO TO 16
      IF (NR(J) .EQ. 0) GO TO 18
```

Figure 28. (Sheet 9 of 23)

```
      LR=NR(I)-NR(J)+1
      J1=NR(J)-L11
      IF (IQ .NE. NUMBLK) GO TO 7177
      ST(J1,LR)=ST(J1,LR)+ELK(I,J)
      GO TO 18
 7177 CONTINUE
      SYSK(J1,LR)=SYSK(J1,LR)+ELK(I,J)
      GO TO 18
   16 IF (NR(I) .EQ. 0) GO TO 18
      LS=NR(J)-NR(I)+1
      J2=NR(I)-L11
      IF (IQ .NE. NUMBLK) GO TO 7178
      ST(J2,LS)=ST(J2,LS)+ELK(I,J)
      GO TO 18
 7178 CONTINUE
      SYSK(J2,LS)=SYSK(J2,LS)+ELK(I,J)
   18 CONTINUE
   20 CONTINUE
   25 CONTINUE
      IF (IQ .EQ. NUMBLK) GO TO 888
      L4=IQ+10
      CALL ECWR(SYSK,L4,L3,IERR)
      IF(IERR.EQ.1)GO TO 999
  888 CONTINUE
  544 RETURN
  999 PRINT 777
  777 FORMAT(20H WRITE ERROR IN ABSK  )
      STOP
      END
*DECK GSPK
      SUBROUTINE GSPK(SYSK,XJ,XY,XH,DH,ANGL
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,VJ,VY,NLGG,IBUG)
C       CHANGES MADE
C*********************************************************************
C     THIS SUBROUTINE OBTAINS MATRIX K2 + K3, AND STORES IN THEIR
C     CORRESPONDING ADDRESSES IN SYSK(I,J)
C*********************************************************************
      COMPLEX XH,DH,TM,SYSK
      COMMON/DB/ WK,ALPHA,D1,DA2,DA4,RR,WR,XL,WL,RKHP,HKL
      DIMENSION SYSK(NLGG,NBAND),XJ(NCS),XY(NCS),XH(NCS),DH(NCS)
     1,ANGL(NODR1)
      DIMENSION VJ(1),VY(1)
      LEVEL 2,SYSK
      COMMON /NEW/ NUMBLK,NLG,NLGL,L11,LS
     1,L4,L3
C*********************************************************************
C     OBTAIN BESSEL FCN XJ,XY,XH,DH
C*********************************************************************
      WR= WK*RR
      XL= RR*DA2
      WL=WK*XL
      DO 12 I=1,NCS
      II=I-1
      FNU=0.
      CALL BESSY(WR,FNU,II,VJ,VY)
      XJ(I)=VJ(I)
      XY(I)=VY(I)
      IF(IBUG.EQ.0)GO TO 12
      WRITE(6,20) XJ(I),XY(I)
   12 CONTINUE
      DO 15 I=1,NCS
      XN= FLOAT(I-1)/WR
      XH(I)=XJ(I)+(0.0,1.0)*XY(I)
      IF( I .NE. 1) GO TO 9
      DH(1)=-XJ(2)-(0.0,1.0)*XY(2)
      GO TO 10
```

Figure 28. (Sheet 10 of 23)

```
    9 DH(I)=(XJ(I-1)-XN*XJ(I))+(0.0,1.0)*(XY(I-1)-XN*XY(I))
   10 CONTINUE
      IF(IBUG.EQ.0)GO TO 15
      WRITE(6,20) XH(I),DH(I)
   15 CONTINUE
   20 FORMAT(9H GSPK BY=,4E15.6)
 1520 FORMAT (20H BLOCK ERROR IN GSPK  ,I5)
C*********************************************************************
C     STORE THE UPPER TRIANGLE ELEMENT OF K2 IN CORRESPONDING ADDRESSES
C     IN SYSK(I,J)
C*********************************************************************
      RKHP= 3.14159265*WR*D1
      DO 38 I=1,NCS
      I1=NNOD+I-L11
      NLLL=1
      IF (I1.LE.0 .OR. I1.GT.NLGG) GO TO 1551
      CX=0.5
      IF(I.EQ.1) CX=1.0
      SYSK(I1,1)=SYSK(I1,1)+CX*RKHP*XH(I )*DH(I )
   38 CONTINUE
C*********************************************************************
C     STORE K3 IN CORRESPONDING ADDRESS IN SYSK(I,J)
C*********************************************************************
      DO 40 I=1,NODR1
   40 ANGL(I)=(I-1)*DA2+DA4
      HKL=WL*D1/2.
      N1=NNOD-NODR
      DO 48 I=1,NODR
      II=N1+I-L11
      NLLL=2
      IF (II.LE.0 .OR. II.GT.NLGG) GO TO 1551
      I1=I-1
      NODRI=NODR+1-I
      DO 48 J=1,NCS
      JJ=NODRI+J
      IF( J .GT. 1) GO TO 42
      TM=HKL*DH(1)
      IF(I.GT.1.AND.I.LT.NODR) TM=2.0*TM
      GO TO 46
   42 J2=J-1
      IF(I.LE.1.OR.I.GE.NODR) GO TO 51
      A1=ANGL(I)*J2
      A2=ANGL(I1)*J2
      TM=HKL*DH(J)*(COS(A2)+COS(A1))
      GO TO 46
   51 IF(I.EQ.NODR) GO TO 52
      A1=ANGL(I)*J2
      TM=HKL*DH(J)*COS(A1)
      GO TO 46
   52 A2=ANGL(I1)*J2
      TM=HKL*DH(J)*COS(A2)
   46 SYSK(II,JJ)=SYSK(II,JJ)-TM
   48 CONTINUE
      IF (NUMBLK .EQ. 1) RETURN
      L5=NLGG*NBAND*2
      L6=L4+1
      CALL ECWR(SYSK,L6,L5,IERR)
      IF(IERR.EQ.1)GO TO 2000
      IF(IBUG.EQ.0)GO TO 3000
      PRINT 1018, (SYSK(I,1),SYSK(I,2),I=1,NLGG,4)
 1018 FORMAT (1X,4G14.5)
 3000 RETURN
 2000 PRINT 999
  999 FORMAT (20H WRITE ERROR IN GSPK  )
      STOP
 1551 PRINT 1520, NLLL
```

Figure 28. (Sheet 11 of 23)

```
      STOP
      END
*DECK LOAD
      SUBROUTINE LOAD(SM,SYSQ,XJ,DH,ANGL
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,IBUG)
C*********************************************************************
C     THIS SUBROUTINE GIVES LOADING TERMS @ SYSQ(I)
C*********************************************************************
      COMPLEX DUMY,SM,SYSQ,DH,TM,CM,DUM2
      COMMON/DB/ WK,ALPHA,D1,DA2,DA4,RR,WR,XL,WL,RKHP,HKL
      DIMENSION SM(NODR1),SYSQ(NEQT),DH(NCS),XJ(NCS),ANGL(NODR1)
      LEVEL 2,SYSQ
      DWL=0.5*D1*WL
C*********************************************************************
C     CALCULATE LOADING TERMS W.R.T. NODAL POTENTIAL + STORE IN SM(I)
C     AND THEN IN SYSQ(I)
C*********************************************************************
      DO 12 I=1,NODR1
      THETA= ANGL(I)
      CX=COS(THETA-ALPHA)
      AUG=WR*CX
      DUMY=COS(AUG)+(0.0,1.0)*SIN(AUG)
      C2=COS(THETA+ALPHA)
      AU2=WR*C2
      DUM2=COS(AU2)+(0.0,1.0)*SIN(AU2)
      SM(I)=DWL*(0.0,1.0)*(CX*DUMY+C2*DUM2)
   12 CONTINUE
      IF(IBUG.EQ.0)GO TO 100
      WRITE(6,8)(SM(I),I=1,NODR)
    8 FORMAT(//,(7E15.7))
  100 N1 = NNOD-NODR
      DO 16 I=1,NODR
      I1=N1+I
      IF(I.EQ.1) GO TO 14
      IF(I.EQ.NODR) GO TO 15
      II=I-1
      SYSQ(I1)=SYSQ(I1) +SM(I)+SM(II)
      GO TO 16
   14 SYSQ(I1)=SYSQ(I1)+SM(1)
      GO TO 16
   15 SYSQ(I1)=SYSQ(I1)+SM(NODR1)
   16 CONTINUE
      IF(IBUG.EQ.0)GO TO 110
      WRITE(6,8)(SYSQ(I),I=N1,NNOD)
C*********************************************************************
C     CALCULATE LOADING TERMS W.R.T. SOURCES AND STORE IN SYSQ(I)
C*********************************************************************
  110 RKHP2=2.0*RKHP
      DO 40 I=1,NCS
      NNI=NNOD+I
      II=I-1
      AI=ALPHA*II
      CM=(0.0,1.0)**II
      TM=CM*XJ(I)*DH(I)*COS(AI)
      SYSQ(NNI)=SYSQ(NNI)-RKHP2*TM
   40 CONTINUE
      RETURN
      END
*DECK ELMT
      SUBROUTINE ELMK(X1,X2,X3,Y1,Y2,Y3,WK,D,DIN,ELK,AREA,LUM,
     1NELE3)
C*********************************************************************
C     GENERATION OF TRIANGLE ELEMENT MATRIX ELK
C     DIN=1.0 IF IN OUTER DOMAIN? D=D1, HELMOTZ EQ.
C     DIN=0.0 IF UNDER FLOATING BODY? D=D2, LAPLACE EQ.
C*********************************************************************
```

Figure 28. (Sheet 12 of 23)

```
      DIMENSION ELK(3,3)
      DO 8 I=1,3
      DO 8 J=1,3
    8 ELK(I,J)= 0.
      B1= Y3-Y2
      B2= Y1-Y3
      B3= Y2-Y1
      C1= X2-X3
      C2= X3-X1
      C3= X1-X2
      AREA=0.5*(B1*C2-B2*C1)
      G=32.2
      FAC=2.*AREA/G
      LUM1=3*LUM-2
      LUM2=3*LUM-1
      LUM3=3*LUM
      IF( AREA .GT. 0. ) GO TO 10
      WRITE(6,100) AREA,LUM
  100 FORMAT(1X,23H DEBUG ELMK 100, AREA =,E12.5,5H   AT,I5,11H-TH ELEME
     1NT)
      STOP
   10 A4=4.*AREA
      WKVD=WK
      WK=WK/SQRT(D)
      WKA= DIN*WK*WK*AREA/12.
      ELK(1,1)= ((B1*B1+C1*C1)/A4-2.*WKA)*D
      ELK(1,2)= ((B1*B2+C1*C2)/A4-   WKA)*D
      ELK(1,3)= ((B1*B3+C1*C3)/A4-   WKA)*D
C     ELK(2,1)=ELK(1,2)
      ELK(2,2)= ((B2*B2+C2*C2)/A4-2.*WKA)*D
      ELK(2,3)= ((B2*B3+C2*C3)/A4-   WKA)*D
C     ELK(3,1)=ELK(1,3)
C     ELK(3,2)=ELK(2,3)
      ELK(3,3)= ((B3*B3+C3*C3)/A4-2.*WKA)*D
      WK=WKVD
      RETURN
      END
*DECK BESY
      SUBROUTINE BESSY(X,FNU,N,BJ,BY)
C     BESSY NYU MATH UTILITY SUBROUTINE FOR BESSEL Y 3/15/64 F. RAGUSA
      DIMENSION BJ(1),BY(1)
C     DIMENSION BJ(1),BY(1)
      EQUIVALENCE(FN,MF)
C
      XSAVE=X
      M1=N
      MF=131070
      BJ(1)=FN
      CALL BESSJ(X,FNU,FN,BJ)
      BY(1)=FN
      YNU=BY(1)
      N=M1
      BY(2)=X
      X=XSAVE
C
      NN=IABS(N)
      N1=NN-1
      CONST=2.0/X
      PI=.314159265E+01
C
      IF(X-10.0) 30,19,19
   30 X10=X+25.0
      K10=X10
      N10=NN+10
      M= MAX0(K10,N10)
C
```

Figure 28. (Sheet 13 of 23)

```
      IF (X - 1.0) 1, 1, 3
    1 KP=172.69388/(3.6888795-ALOG(X))
      M=MINO(M,KP)
      GO TO 2
    3 KP=39.0*X**.3333333
      M=MINO(M,KP)
    2 M=M/2
      K=2*M+1
      ARG=FNU*PI
      GARG=GAMMA(1.0+FNU,0.)**2
C
C     COMPUTE GAMMA ZERO EQU. 15,PG. 5 , IF NU=0 USE EQU. 16.
C
   14 IF(FNU) 15,16,15
   15 TERM =(1.0/PI)* CONST **(2.0*FNU)
      GAM1=COS(ARG)/SIN(ARG) - TERM*(GARG/FNU)
C     COMPUTE GAMMA ONE FOR NU NOT EQUAL ZERO.
      GAM2=2.0 * TERM * GARG * (FNU+2.0) / (1.0-FNU)
      GO TO 10
C
C     COMPUTE GAMMA ZERO AND ONE FOR NU EQUAL ZERO.
   16 TLOG = ALOG(X/2.0)
      A=.577215664E+00
      PIH=2.0/PI
      GAM1=PIH*(A+TLOG)
      GAM2=4.0/PI
C
C     COMPUTE GAMMA, YNU, AND BY(1), AND BY(2)
C     EQUATIONS 15,17, AND 18.
C
   10 BY(2) =0.0
      BY(1)=0.0
      GAM3=0.0
      E1=-(1.0/PI)*CONST **(1.0+2.0*FNU) * GARG
      BY(2)=E1*BJ(1)+BY(2)
      E1=GAM1-GAM2/2.0
      BY(2)=E1*BJ(2)+BY(2)
      YNU=GAM1*BJ(1)
      TXNU=3.0*FNU/X
      AB=ABS(BJ(1)) - .000005
      I2=1
      MP1=M+1
C
      DO 11 I=2,MP1
      I2 = I2 + 2
      FI = I
      FIM = I-1
      FI2 = 2*I
      DENOM = FI*(FI-FNU)*(FNU+FI2-2.0)
      GAM3 = (FNU+FI2)*(2.0*FNU+FIM)*(FNU+FIM)/DENOM
      GAM3 = -GAM3*GAM2
      YNU = GAM2*BJ(I2)+YNU
      IF(AB)18,18,23
C     WHEN J(NU) IS NEAR ZERO COMPUTE BY(2) FROM EQ. 18,ELSE EQ. 17.
C
   18 E1=TXNU*GAM2
      BY(2)=E1*BJ(I2)+BY(2)
      IF(I2-K) 25,110,110
   25 E1=(GAM2-GAM3)/2.0
      BY(2)=E1*BJ(I2+1)+BY(2)
   23 GAM1=GAM2
   11 GAM2=GAM3
C
  110 BY(1)=YNU
      IF(AB) 19,19,17
   17 BY(2)=(YNU*BJ(2)-2.0/(PI*X))/BJ(1)
```

Figure 28. (Sheet 14 of 23)

```
C
C     IF N=0 OR 1 GO OUT ALL BY S COMPUTED.
   19 IF(N) 50,49,20
   20 IF(N-1) 49,49,21
C
C     COMPUTE Y(N+1) BY RECURRENCE
C
   21 DO 22 I=1,N1
      FI1=I
   22 BY(I+2)=CONST*(FI1+FNU)*BY(I+1)-BY(I)
C
   49 RETURN
   50 IF(FNU-.500000) 54,53,54
   53 BY(2)=BJ(1)
      IF(NN-1) 49,61,60
   60 ARG=-1.0
      BY(3)=ARG*BJ(2)
      BJ(2)=CONST*FNU*BJ(1)-BJ(2)
      ARG=-ARG
      IF (NN-2) 49,51,56
   56 DO 55 I=2,N1
      BY(I+2)=ARG*BJ(I+1)
   55 ARG=-ARG
      GO TO 51
   54 BY(2)=CONST*FNU*BY(1)-BY(2)
      BJ(2)=CONST*FNU*BJ(1)-BJ(2)
      IF(NN-1) 49,49,51
   51 FRAC=FNU
      DO 52 I=1,N1
      FRAC=FRAC-1.0
      BJ(I+2)=CONST*FRAC*BJ(I+1)-BJ(I)
      IF(FNU-.50000) 58,52,58
   58 BY(I+2)=CONST*FRAC*BY(I+1)-BY(I)
   52 CONTINUE
      GO TO 49
   61 BJ(2)=CONST*FNU*BJ(1)-BJ(2)
      GO TO 49
       END
*DECK GAMA
      FUNCTION GAMMA(U,V)
      DIMENSION B(8)
      B(1)= .035868343
      B(2)=-.193527818
      B(3)= .482199394
      B(4)=-.756704078
      B(5)= .918206857
      B(6)=-.897056937
      B(7)= .988205891
      B(8)=-.577191652
      A=U
      X=V
      XN=15.0
    1 IF(A)4,2,3
    2 IF(X-1.0)4000,4000,5000
    4 Z = AINT(A)
      Y=ABS(Z-A)
      EPS=3.0E-8
      IF(Y-EPS)21,21,5
   21 A=Z
      GO TO 22
    5 Y=1.0-Y
      IF(Y-EPS)6,6,3
    6 A=Z-1.0
   22 IF(X-1.0)4000,4000,5000
    3 IF(X)7,1000,7
    7 IF(X-1.0)9,9,8
```

Figure 28. (Sheet 15 of 23)

```
    8 IF (A) 5000, 10, 10
   10 IF((X/A)-1.0)1000,5000,5000
    9 IF (ABS(A) -1.) 11,1000,1000
   11 IF(X-.1)1000,12,12
   12 IF(X-.2)13,14,14
   13 XN=140.0
      GO TO 5000
   14 IF(X-.4)15,16,16
   15 XN=80.0
      GO TO 5000
   16 IF(X-.6)17,18,18
   17 XN=60.0
      GO TO 5000
   18 IF(X-.8)19,20,20
   19 XN=40.0
      GO TO 5000
   20 XN=20.0
 5000 XFT=X
      N=XN
      DO 5001 I=1,N
      XFT=(XN/XFT)+1.0
      XFT=((XN-A)/XFT)+X
 5001 XN=XN-1.0
      TEM = (ALOG(X)*A)-X
      GO TO 6000
 4000 SUM=0.0
      ANS=XFT
      XFT = EXP(TEM)/XFT
      EPS=1.0E-8
      XM=1.0
      XMT=1.0
      EX=-1.0
      TEM=X
 4001 Y=TEM/(XMT*XM)
 4011 SUM=SUM+EX*Y
      IF (ABS(Y)-EPS) 4003,4002,4002
 4002 TEM=TEM*X
      XM=XM+1.0
      XMT=XMT*XM
      EX=-EX
      GO TO 4001
 4003 E=-ALOG(X)-.57721566-SUM
      IF(A)4005,4004,4005
 4004 ANS=E
      GO TO 6000
 4005 IF(A+2.0)4010,4009,4008
 4008 ANS=-E+EXP(-X)/X
      GO TO 6000
 4009 SUM=(1.0/X)-(1.0/(X*X))
      ANS=(E-EXP(-X)*SUM)*.5
      GO TO 6000
 4010 EX=-1.0
      N=-A-1.0
      XMT=1.0
      SUM=0.0
      TEM=X
      DO 4006 I=1,N
      TEM=TEM*X
      XM=I+1
      Y=XMT/TEM
      SUM=SUM+EX*Y
      EX=-EX
 4006 XMT=XMT*XM
      SUM=SUM+1.0/X
      Z=E-EXP(-X)*SUM
      Y=ABS(A)
```

Figure 28. (Sheet 16 of 23)

```
      EX=-1.0
      XMT=1.0
      N=Y
      DO 4007 I=2,N
      XM=I
      XMT=XMT*XM
 4007 EX=-EX
      ANS=Z*EX/XMT
      GO TO 6000
 1000 IF(A-1.0)1001,1003,1004
 1004 IF(A-2.0)1003,1003,1002
 1003 AT=A-1.0
      ATM=B(1)*AT
      DO 1005 I=2,8
 1005 ATM=(ATM+B(I))*AT
      ANS=ATM+1.0
      GO TO 2000
 1001 IF(A)1007,1006,1006
 1006 BN=A
      BS=A
      GO TO 1010
 1007 BN=A
      BS=A
 1011 IF(BS)1009,1010,1010
 1009 BS=BS+1.0
      BN=BS*BN
      GO TO 1011
 1010 ATM=BS*B(1)
      DO 1012 I=2,8
 1012 ATM=(ATM+B(I))*BS
      ANS=(ATM+1.0)/BN
      GO TO 2000
 1002 BN=A-1.0
      BS=BN
 1013 IF(2.0-BS)1015,1014,1014
 1015 BS=BS-1.0
      BN=BS*BN
      GO TO 1013
 1014 BS=BS-1.0
      ATM=BS*B(1)
      DO 1016 I=2,8
 1016 ATM=(ATM+B(I))*BS
      ANS=(ATM+1.0)*BN
 2000 IF(X)2001,6000,2001
 2001 W=1.0
      XM=1.0
      SUM=1.0
      EPS=1.0E-8
 2002 W=(X/(A+XM))*W
      SUM=SUM+W
      XM=XM+1.0
      IF (ABS(W)-EPS) 2003,2002,2002
 2003 T=A*ALOG(X)
      T=EXP(T-X)/A
      ANS=ANS-(T*SUM)
 6000 GAMMA=ANS
      RETURN
      END
*DECK BESJ
      SUBROUTINE BESSJ(X,FNU,FN,F)
C     BESSJ NYU MATH UTILITY BESSEL J SUBROUTINE 3/15/64 F. RAGUSA
      DIMENSION F(1)
      EQUIVALENCE(FN,MF)
C
      N=FN
      NN=IABS(N)
```

Figure 28. (Sheet 17 of 23)

```
C     WHEN BESJ HAS BEEN CALLED BY BESY F(1) WILL CONTAIN INTERGER 131070
      FM=F(1)
      IY=1
      IF(KF-131070) 2,1,2
    1 IY=2
    2 CONST=2.0/X
C
      IF(X-50.0) 6,8,8
    6 X10=X+25.0
      K10=X10
      N10=NN+10
   40 M=MAX0(K10,N10)
C
      IF(X-1.0) 3,3,41
    3 KP=(172.69388/(3.6888795-ALOG(X)))-1.0
      M=MIN0(M,KP)
      GO TO 4
   41 KP=39.0*X**.3333333
      M=MIN0(M,KP)
    4 M=M/2
   43 K=2*M+1
      K2=K+1
      K3=K+2
      F(K2)=1.0E-35
      F(K3)=0.0
C
      DO5 L=1,K
      I=K+1-L
      FLI=I
    5 F(I)=CONST*(FLI+FNU)*F(I+1)-F(I+2)
C
C     FIND ALPHA EITHER FROM EQUA. 12,PAGE 4 WHEN X IS LESS THAN 8.0
C     OR FROM BESSEL FUNCT. FOR LARGE ARGUMENTS WHEN X IS =OR GREATER.
C
      IF(X-10.0) 7,8,8
    7 PHI=FNU+2.0
      ALF=PHI*F(3)+F(1)
      MO=3
C
      DO15 I=2,M
      MO=MO+2
      FM2=2*I
      FM1=I-1
      FI=I
      TEMP=((FNU+FM2)*(FNU+FM1))/(FI*(FNU+FM2-2.0))*PHI
      PHI=TEMP
   15 ALF=PHI*F(MO)+ALF
C
      ALF=CONST**FNU*GAMMA(FNU+1.0,0.)*ALF
C
      I1=1
C
C     FIND J(N) EQUATION 7,PAGE 3,WHEN X LESS THAN 10.0
C     OR J(2),J(3),...,J(N)=F(2)/ALPHA,...,F(N)/ALPHA WHERE
C     ALPHA=F(1)/J(1) AND J(1) A*COS(PHI) FROM PATH 8.
C
   16 DO17 I=I1,K
   17 F(I)=F(I)/ALF
C     TEST FOR NEGATIVE N AND RECOMPUTE FS BY RECURSION.
      IF(N) 18,22,22
   18 IF(IY-2) 19,22,19
   19 F(2)=CONST*FNU*F(1)-F(2)
      IF(NN-1) 22,22,20
   20 FRAC=FNU
      N1=NN+1
      DO 21 L=3,N1
```

Figure 28. (Sheet 18 of 23)

```
      FRAC=FRAC-1.0
   21 F(L)=CONST*FRAC*F(L-1)-F(L-2)
C
C     IF ENTRY WAS FROM BESY RETURN A VALUE FOR Y(1) IN
C     PLACE OF N.IY WILL BE EQUAL TO 2 IN THIS CASE.
C
   22 GO TO (24,23),IY
   23 FN Y1
      X=Y2
   24 RETURN
C
C     PATH 8. WHEN X IS GREATER THAN OR =TO 10.0
C     COMPUTE J(1) FROM M.GOLDSTEIN PAPER BESSEL
C     FUNCTIONS FOR LARGE ARGUMENTS.
C
    8 MOUNT=1
      GNU=FNU
      C0 =.25
      C1 =.15625
      C2 =-.375
      C3 =.1171875
      C4 =-1.15625
      C5 =1.875
      C6 =.952148437E-01
      C7 =-2.38671875
      C8 =14.2265625
      C9 =-19.68750
      C10=.809326171E-01
      C11=-.410058593E+01
      C12=.582246093E+02
      C13=-277.87500
      C14=354.3750
      C15=.416666666E-01
      C16=-.25
      C17=.125E-1
      C18=-.350
      C19=.558035718E-03
      C20=- .424107142E+00
      C21=3.60267857
      C22=-5.625
      C23=.30381944E-02
      C24=-.486111
      C25=.102864583E+02
      C26=-58.0
      C27=78.75
    9 AL1=GNU**2-.25
      A2 =C0*AL1
      A4 =C1*AL1
      A4=(A4+C2)*AL1
      A6=C3*AL1
      A6=(A6+C4)*AL1
      A6=(A6+C5)*AL1
      A8=C6*AL1
      A8=(A8+C7 )*AL1
      A8=(A8+C8 )*AL1
      A8=(A8+C9 )*AL1
      A10 C10*AL1
      A10=(A10+C11)*AL1
      A10=(A10+C12)*AL1
      A10=(A10+C13)*AL1
      A10=(A10+C14)*AL1
C
      PI=.314159265E+01
      TS=1.0/X
      T2=TS**2
C
```

Figure 28. (Sheet 19 of 23)

```
      B=A10*T2+AB
      B=B*T2+A6
      B=B*T2+A4
      B=B*T2+A2
      BNU=B*T2+1.0
C
      ANU=BNU/SQRT(.5*PI*X)
C
C     PAGE 20, EQUATION 12 TO GET PHI ZERO
C
      A2=.5*AL1
      A4=C15*AL1
      A4=(A4+C16)*AL1
      A6=C17*AL1
      A6=(A6+C18)*AL1
      A6=(A6+.75)*AL1
      A8=C19*AL1
      A8=(A8+C20)*AL1
      A8=(A8+C21)*AL1
      A8=(A8+C22)*AL1
      A10=C23*AL1
      A10=(A10+C24)*AL1
      A10=(A10+C25)*AL1
      A10=(A10+C26)*AL1
      A10=(A10+C27)*AL1
C
      B=A10*T2+A8
      B=B*T2+A6
      B=B*T2+A4
      B=B*T2+A2
      TPHI=B*T2+1.0
      PHI=TPHI*X-(GNU+.5)*(PI/2.0)
      COP=COS(PHI)
      SIP=SIN(PHI)
C
      F1=ANU*COP
      Y1=ANU*SIP
C
      IF(KOUNT-1) 10,10,11
C
   10 FSAVE=F1
      YSAVE=Y1
      GNU=FNU+1.0
      KOUNT=2
      GO TO 9
C
   11 F2=F1
      Y2=Y1
      F1=FSAVE
      Y1=YSAVE
C
      IF(X-50.0) 110,111,111
  110 IF (ABS(F1) - ABS(F2)) 12,12,13
  111 F(1)=F1
      F(2)=F2
      IF(N) 18,22,112
  112 IF(N-1) 22,22,113
  113 N1=NN+1
      FLI=0.0
      DO 114 I=3,N1
      FLI=FLI+1.0
  114 F(I)=CONST*(FNU+FLI)*F(I-1)-F(I-2)
      GO TO 22
C
   12 ALF=F(2)/F2
      GO TO 14
```

Figure 28. (Sheet 20 of 23)

```
C
   13 ALF=F(1)/F1
   14 F(1)=F1
      F(2)=F2
      I1=3
      GO TO 16
       END
*DECK BANDS
      SUBROUTINE BANSOL(A,D,B,NN,MM,NUMBLK,NEQT,LLL,E,NLGL,F)
      COMPLEX F
      DIMENSION F(NN,MM)
      COMPLEX A,B,D,C,E
      DIMENSION A(LLL,MM),D(1),B(1),E(NLGL,MM)
      LEVEL 2,A,D,B,E,F
      NNN=LLL/2
      NQ=NN*MM
      NBL=NN
      MQ=NQ*2
      MQQ=MQ
      NBAND=MM
      NBLQ=NN
      NB=0
      GO TO 150
  100 NB=NB+1
      IF (NUMBLK-NB-1) 103,104,101
  104 MQQ=NNN*MM*2
      NBLQ=NNN
      GO TO 101
  103 NBL=NNN
C CHANGES
  101 DO 125 N=1,NBL
      NM=NN+N
      B(N)=B(NM)
      B(NM)=0.0
      DO 125 M=1,MM
      A(N,M)=A(NM,M)
  125 A(NM,M)=0.0
C   NO CHANGES
      IF (NUMBLK -NB) 150,200,150
  150 NQQ=NB+11
      IF (NBLQ-NN) 109,110,109
  110 CALL FILL (A,F,B,D,NB,NN,NNN,NBAND,LLL,NQQ,MQQ)
      GO TO 126
  109 CALL FILL (A,E,B,D,NB,NN,NNN,
     1 NBAND,LLL,NQQ,MQQ)
  126 IF (NB) 200,100,200
  200 DO 300 N=1,NBL
      IF (CABS(A(N,1)) .LT. 1.0E-30) GO TO 999
  225 B(N)=B(N)/A(N,1)
      DO 275 L=2,MM
      IF (A(N,L)) 230,275,230
  230 C=A(N,L)/A(N,1)
      I=N+L-1
      J=0
      DO 250 K=L,MM
      J=J+1
  250 A(I,J)=A(I,J)-C*A(N,K)
      B(I)=B(I)-A(N,L)*B(N)
      A(N,L)=C
  275 CONTINUE
  300 CONTINUE
      IF (NUMBLK-NB) 375,405,375
  375 NQZ=(NB-1)+11
      NBNN=NB*NN
      DO 127 N=1,NN
      NC=NBNN-NN+N
```

Figure 28. (Sheet 21 of 23)

```
      D(NC)=B(N)
      DO 127 M=1,NBAND
  127 F(N,M)=A(N,M)
      CALL ECWR(F,NQZ,MQ,IERR)
      IF (IERR .EQ. 1) GO TO 888
      GO TO 100
  405 CONTINUE
  400 DO 450 M=1,NBL
      N=NBL+1-M
      DO 425 K=2,MM
      L=M+K-1
  425 B(N)=B(N)-A(N,K)*B(L)
      NM=N+NNN
      B(N+NN)=B(N)
  450 A(NM,NB)=B(N)
      NB=NB-1
      NBL=NN
      IF (NB) 475,500,475
  475 NQL=(NB-1)+11
      CALL ECRD(F,NQL,MQ,IERR)
      IF (IERR .EQ. 1) GO TO 777
      DO 136 N=1,NN
      NC=NB*NN-NN+N
      B(N)=D(NC)
      DO 136 M=1,NBAND
  136 A(N,M)=F(N,M)
      GO TO 400
  500 K=0
      DO 600 NB=1,NUMBLK
      IF (NB .EQ. NUMBLK) NBL=NNN
      DO 600 N=1,NBL
      NM=N+NNN
      K=K+1
  600 D(K)=A(NM,NB)
  555 RETURN
  777 N=1
  444 PRINT 666,N
  666 FORMAT (10H ERROR NO.,I10)
      STOP
  888 N=2
      GO TO 444
  999 PRINT 998,A(N,1),N
  998 FORMAT (13H DIAG TROUBLE  ,5X,2G13.5,I5)
      STOP
      END
*DECK FILL
      SUBROUTINE FILL (A,E,B,D,NB,NM,NN,NBD,LLL,NQQ,MQQ)
      COMPLEX A,B,D,E
      DIMENSION A(LLL,NBD),B(1),D(1),E(NN,NBD)
      LEVEL 2,A,E,B,D
      NBN=NB*NM
      CALL ECRD(E,NQQ,MQQ,IER)
      IF (IER .EQ. 1) GO TO 999
      DO 1 N=1,NN
      NC=NBN+N
      NNN=NN+N
      B(NNN)=D(NC)
      DO 1 M=1,NBD
    1 A(NNN,N)=E(N,M)
      RETURN
  999 N=3
      PRINT 666,N
  666 FORMAT (10H ERROR NO.  ,I5)
      STOP
      END
*DECK ECWR
```

Figure 28. (Sheet 22 of 23)

```
      SUBROUTINE ECWR(S,L1,L2,IERR)
      DIMENSION S(1)
      LEVEL 2,S
      IERR = 0
      REWIND L1
      WRITE(L1) (S(I),I=1,L2)
      IF(UNIT(L1).GE.0.)IERR = 1
      RETURN
      END
*DECK ECRD
      SUBROUTINE ECRD(S,L1,L2,IERD)
      DIMENSION S(1)
      LEVEL 2,S
      IERD = 0
      REWIND L1
      READ(L1) (S(I),I=1,L2)
      IF((EOF(L1).NE.0).OR.(IOCHEC(L1).NE.0))IERD = 1
      RETURN
      END
```

Figure 28. (Sheet 23 of 23)

DATA SET NO. 1

CARD NO. 1

Col.	Format	Variable Name	Description
1 - 80	20A4	ID	This card contains alpha-numeric title that appears in the output for job identification

CARD NO. 2

1 - 5	I5	NBAND	NBAND equals either the maximum difference between two nodes or the sum of NCS and NODR, whichever is greater.
6 - 10	I5	IPR	IPR=1, input data check - echo printed IPR=2, echo not printed
11 - 20	F10.2	RR	Radius of the semicircle (feet)
21 - 30	F10.5	ALPHA	Wave angle of incidence
31 - 40	F10.5	DM	Water depth of the node on the semicircle
41 - 46	I6	IBLKLEN	The value of IBLKLEN should not exceed 31000

CARD NO. 3

1 - 5	I5	IPC	IPC=0, the elevation and velocity at selected elevation stations are omitted IPC=1, the values are printed
6 - 10	I5	IPMAX	IPMAX=0, subroutine CPUNCH will not be called
11 - 15	I5	IDPC	IDPC=0, special procedure for input of average depths for each element is used IDPC=1, read depth at each grid node from cards or from the data files
16 - 20	I5	IDCH	IDCH=1, element number, nodal point, and depth of each element and the x and y coordinates of each node are printed
21 - 25	I5	IBUGA	IBUGA=1, ELK (element stiffness matrix 3 x 3) are printed

Col.	Format	Variable Name	Description
26 - 30	I5	IBUGG	IBUGG=1, SYSK(I,1) and SYSK(I,2) are printed
31 - 35	I5	IBUGL	IBUGL=1, SYSO(I) are printed
36 - 40	I5	IDISP	IDISP=1, dispersion included
41 - 45	I5	ICIRC	ICIRC=0, for semicircle grid ICIRC=1, for circular grid
46 - 50	I5	NC	NC=0, selected elevation stations (nodes) will not be punched
		CARD NO. 4	
1 - 10	F10.2	TMIN	The minimum wave period (seconds)
11 - 20	F10.2	TDELT	The increment of wave period (seconds)
21 - 25	I5	IKK	The number of times of the increment
		DATA SET NO. 2	
		CARD NO. 1	
1 - 5	I5	NSP(1)	Selected elevation station number (nodes), starting from NSP(1) to NSP(ISPT)
6 - 10	I5	NSP(2)	Fourteen values on each card and continuing to the next card as needed
.			
.			
.			
.			
66 - 70	I5	NSP(14)	
		DATA SET NO. 3	
		CARD NO. 1	
1 - 5	I5	NSPE(1)	Selected elevation station number (element), starting from NSPE(1) to NSPE(ISPTE)
6 - 10	i5	NSPE(2)	Fourteen values on each card and continuing to the next card as needed

Col.	Format	Variable Name	Description
.			
.			
.			
66 - 70	I5	NSPE(14)	

DATA SET NO. 4

CARD NO. 1

1 - 10	F10.2	X(1)	x-coordinate at each nodal point, starting from X(1) to X(NNOD)
11 - 20	F10.2	X(2)	Eight values on each card and continuing to the next card as needed
.			
.			
.			
.			
71 - 80	F10.2	X(8)	

DATA SET NO. 5

CARD NO. 1

1 - 10	F10.2	Y(1)	y-coordinate at each nodal point, starting from Y(1) to Y(NNOD)
11 - 20	F10.2	Y(2)	Eight values on each card and continuing to the next card as needed
.			
.			
.			
.			
71 - 80	F10.2	Y(8)	

Col.	Format	Variable Name	Description
		DATA SET NO. 6	
		CARD NO. 1	
1 - 5	Blanks		
6 - 10	I5	NCON(1,1)	Node number of the first nodal point of element 1
11 - 15	I5	NCON(1,2)	Node number of the second nodal point of element 1
16 - 20	I5	NCON(1,3)	Node number of the third nodal point of element 1
21 - 25	Blanks		
26 - 30	I5	NCON(2,1)	Node number of the first nodal point of element 2, etc.
31 - 35	I5	NCON(2,2)	Nine values on each card, starting from NCON(1,1), NCON(1,2), and NCON(1,3) to NCON(NELE,1), NCON(NELE,2), and NCON(NELE,3), respectively, continuing to the next card as needed
.			
.			
.			
.			
56 - 60	I5	NCON(3,3)	
		DATA SET NO. 7*	
		CARD NO. 1	
1 - 10	F10.2	DD(1)	Depth at each nodal point, starting from DD(1) to DD(NNOD)
11 - 20	F10.2	DD(2)	Eight values on each card and continuing to the next card as needed
.			
.			
.			
.			
71 - 80	F10.2	DD(8)	

*Note: If IDPC=0, Data Set No. 7 is omitted. Depth of each node should be specified by user in the program FINITE.

Procedure to Run Program FINITE

49. To set the correct values for the parameter statements in program FINITE, the following procedure should be followed.

a. Sign on the Cybernet Computer System
Telephone No. 2047 for 300 Baud
Telephone No. 2030 for 1200 Baud

b. Type in
GET,FINITE/UN=CEROMO
SAVE,FINITE

c. Use XEDIT to set correct values for the parameters in the parameter statements
Type in
OLD,FINITE
XEDIT

d. Locate the first parameter statement
Type in
L/PARAMETER/
The system will respond with

PARAMETER(NNOD=35,NELE=48,NODR=9,NCS=7,ISPT=35,ISPTE=48)

e. Change old values in the parameter statement to desired values
Type in
C/NNOD=35,NELE=48,NODR=9/NNOD=new value,NELE=new value,NODR=new value/
C/NCS=7,ISPT=35,ISPTE=48/NCS=new value,ISPT=new value,ISPTE=new value/

f. Locate the second parameter statement
Type in
N1
The system will respond with

PARAMETER(NSYSK=1394,NSYTP=772,NSTMP=84)

g. Change the values in parameter statement to desired values
Type in
C/NSYSK=1394,NSYTP=772/NSYSK=new value,NSYTP=new value/
C/NSTMP=84/NSTMP=new value/

h. Replace program file FINITE by editted version
Type in
E,,RL

50. Establish an "AFFILE" on the CYBER 205 System. An "AFFILE" enables the Corps of Engineers (KOE) user to transmit files from KOE to the CYBER 205, and vise versa.

a. Type in
GET,AFJCL/UN=CEROMO
SAVE,AFJCL

b. Use XEDIT to make necessary changes
Type in
OLD,AFJCL
XEDIT

c. Replace 205USER with actual user's ID of the CYBER 205
Type in
C/205USER/USER'S 205 ID/

d. Replace all PASSWORDS with the actual user's PASSWORD
Type in
C/PASSWORD/ACTUAL USER'S PASSWORD/*

e. Replace all KOEUSER with actual user's ID of the KOE
Type in
T
This command will bring the pointer to the top of the file
Type in
C/KOEUSER/ACTUAL KOE USER'S ID/*

f. Replace file AFJCL with the edited version
Type in
E,,RL

g. To execute file AFJCL
Type in
GET,ADYJOB/UN=CATHDDC
BEGIN,,ADYJOB,AFJCL
The system will respond with

ALL DONE. USER JOB NAME IS COEXXXX

(Note: The status of the job can then be tracked by using "LINK,ENQUIRE." When the job is complete, output will return to the users QGET queue and can be accessed by using the "QLIST" command.)

51. Establish file "QLIST", which enables the KOE users to retrieve CYBER 205 dayfiles and outputs.

a. Type in
GET,QLIST/UN=CEROMO
SAVE,QLIST

b. Use XEDIT to make necessary changes
Type in
OLD,QLIST
XEDIT

c. Replace old UID with new UID
Type in
C/UID=HASH/UID=XXXX/
(XXXX is the user's index hash. It can be obtained from "ENQUIRE(B)." XXXX are the first four characters of the job name that the system assigned to the user.)

d. Replace KOEUSER with user's ID
Type in
C/KOEUSER/USER'S ID/

e. Replace old header with user's header
   Type in
   C/HEADER1/USER'S HEADER/
   C/HEADER2/USER'S HEADER/
   C/HEADER3/USER'S HEADER/

f. Replace file QLIST with the edited version
   Type in
   E,,RL

52. Create an Update Library

a. Type in
   GET,FINUP1/UN=CEROMO
   BEGIN,RUNF,FINUP1
   The system will respond with

   15.56.20 SUBMIT COMPLETE. JOB NAME IS AJZZXXX

b. Check dayfile
   Type in
   OLD,DAY1
   LIST

53. Establish an Update Correction File

a. Type in
   GET,FINDIR/UN=CEROMO
   SAVE,FINDIR

b. Use XEDIT to make necessary changes and replace the file FINDIR with the edited version

54. Run program FINITE with Update Correction File FINDIR (Figure 29).

a. Type in
   GET,FINI/UN=CEROMO
   SAVE,FINI

b. Use XEDIT to make necessary changes
   Type in
   OLD,FINI
   XEDIT

c. Replace 205USER with the CYBER 205 user's ID
   Type in
   C/205USER/CYBER 205 USER'S ID/

d. Replace PASSWORD with user's PASSWORD
   Type in
   C/PASSWORD/USER'S PASSWORD/

e. Replace all KOEUSER with user's KOE USER ID
   Type in
   C/KOEUSER/USER'S KOE USER ID/*

f. Change data file name (optional)*
   Type in
   C/FINDAT1/NEW FILE NAME/
   C/FINDAT/NEW FILE NAME/

(*See Note, page 83)

```
 OLD,FINDIR
/LIST
*IDENT  PARAIS
*D MAIN.2
      PROGRAM FINIT(TAPE4=TAPE4,TAPE8=TAPE8,OUTPUT,TAPE6=OUTPUT,
*D MAIN.3,MAIN.4
     1TAPE9=TAPE9,TAPE11=TAPE11,TAPE12=TAPE12,
     2TAPE13=TAPE13,TAPE14=TAPE14,TAPE15=TAPE15)
*D MAIN.6
CCC   SAMPLE PROBLEM
*D MAIN.8
C*********************************************************************
*D MAIN.14
C*********************************************************************
*D MAIN.13
C     NEQT= TOTAL NO. OF SIM. LIN. EQS. = NNOD+NCS
C     IKK = TOTAL NO. OF CASAS TO BE COMPUTED
C     ISPT= TOTAL NO. OF SELECTD ELEVATION STATION WITH RESPECT TO NODES
C     ISPTE= TOTAL NO. OF SELECTED ELEVATION STATION WITH RESPECT TO ELEMENTS
C     XK= WAVE PERIOD
C     SYSQ,STMP,AT,AF HAVE DIMENSION OF NEQT.
C     X,Y,NOD HAVE DIMENSION OF NNOD.
C     XJ,XY,XH,HD HAVE DIMENSION OF NCS.
C     ANGL,SN HAVE DIMENSION OF NODR.
C     DD,V,U,YC,NCON HAVE DIMENSION OF NELE.
C     NR AND Z HAVE THE DIEMNSION OF THE NO. OF THE NODAL POINTS IN EACH CELL
C     NSP HAS THE DIMENSION OF IPST
C     NSPE HAS THE DIMENSION OF IPSTE
C     IPER HAS THE DIMENSION OF 5
C     XK HAS THE DIMENSION OF IKK
C     AP HAS THE DIMENSION OF ISPT+1
C     SYSK HAS THE DIMENSION OF NLGLXNBANDX2+50
C     SYTP AND FTEMP HAVE DIMENSION OF NLGLXNBAND+100
C     NUMBLK=NEQT*NBAND/IBLKLEN+1
C     NLG = NEQT/NUMBLK
C     NLGL = NEQT-(NUMBLK - 1)*NLG
C     NSTMP = NLG+NLGL
*D MAIN.18,MAIN.21
C
      PARAMETER(NNOD=35,NELE=48,NODR=9,NCS=7,ISPT=35,ISPTE=48)
      PARAMETER(NSYSK=1394,NSYTP=772,NSTMP=84)
      DIMENSION X(NNOD),Y(NNOD),NOD(NNOD),XJ(NCS),
     *XY(NCS),XH(NCS),DH(NCS),ANGL(NODR),SN(NODR),NSP(100),IPER(5)
      DIMENSION                    ID(20),NSPE(100)
      DIMENSION Z(3)
*D MAIN.22,MAIN.24
      COMMON/L1/SYSK(NSYSK)
      COMMON/L2/SYSQ(NSTMP)
      COMMON/L3/STMP(NSTMP)
      COMMON/L4/DD(NELE)
      COMMON/L5/AT(NNOD+NCS)
      COMMON/L6/AF(NNOD+NCS)
      COMMON/L7/SYTP(NSYTP)
      COMMON/L9/U(NELE),V(NELE),YC(NELE)
      COMMON/L8/NCON(NELE,3)
      DIMENSION FTEMP(NSYTP)
*D MAIN.25,MAIN.28
      DIMENSION VJ(100),VY(100)
      DIMENSION NR(3)
      DIMENSION AP(100)
*D MAIN.32
*D MAIN.35,MAIN.37
```

Figure 29. Program FINDIR, update correction file for program FINITE (Sheet 1 of 7)

```
   READ (8,2) ID
*D MAIN.41
      READ(8,5435) NBAND,IFR,RR,ALPHA,DM,IBLKLEN
 5435 FORMAT(2I5,F10.2,F10.5,F10.2,1I6)
*D MAIN.42
*I MAIN.46
      WRITE(6,403) NUMBLK
  403 FORMAT(1X,'NUMBLK =',I5)
*D MAIN.48
      D1=1.
      READ(8,305)IFC,IFMAX,IDFC,IDCH,IBUGA,IBUGG,IBUGL,IDISP,
     1ICIRC,NC
  305 FORMAT(10I5)
      IF(ICIRC.NE.1)GO TO 3010
C FOR CIRCULAR REGION NCS2=(NCS+1)/2 = TOTAL NO. OF COEFF. IN RAD. DOMAIN
      NCS2=(NCS+1)/2
      NCK2=2*NCS2
      IF(NCK2.NE.NCS) GO TO 3010
      WRITE(6,877)
  877 FORMAT(1X,33HNCS MUST BE ODD FOR CIRCULAR GRID)
      STOP
 3010 IF(IFR.EQ.0)GO TO 876
  876 CONTINUE
*D MAIN.54
      IF(NLGG*NBAND.LE.NSYSK) GO TO 5432
*D MAIN.58
      NNELE=NELE
      NNNOD=NNOD
      NNODR=NODR
      NNCS=NCS
      WRITE(6,12) NNELE,NNNOD,NNODR,NNCS,NEQT,ALPHA,DA2,DA4,RK,D1,DM
*I MAIN.72
      WRITE(6,2010)
*D MAIN.73
*D MAIN.75
      WRITE(6,2014) NUMBLK,NLG,NLGL,IBLKLEN,IFR,NBAND
      IF(NLG.GE.NBAND) GO TO 878
      WRITE(6,879)
  879 FORMAT(1X,30HNLG MUST BE GREATER THAN NBAND)
      STOP
  878 CONTINUE
*D MAIN.78
      READ(8,300) TMIN,TDELT,IKK
*D MAIN.80
      WRITE(6,300)TMIN,TDELT,IKK
*D MAIN.81
  300 FORMAT(2F10.2,I5)
*I MAIN.82
      IF(IKK.EQ.1) GO TO 321
*I MAIN.86
  321 CONTINUE
*D MAIN.87,MAIN.89
      IF(ICIRC.EQ.1) DA2=2.*PAI/NODR
      IF(ICIRC.EQ.1) DA4=DA2/2.
*D MAIN.90,MAIN.93
      IF(IFC.EQ.0) GO TO 3000
      READ(8,306)(NSF(J),J=1,ISPT)
      READ(8,306) (NSFE(J),J=1,ISPTE)
C     LEVEL 2,U,V,YC
C     LEVEL 2,NCON,DD,AF,AT
*D MAIN.94
  306 FORMAT(14I5)
*I MAIN.100
      T=XK(IK)
      G=32.2
      IF(IDISP.EQ.0) GO TO 77
```

Figure 29. (Sheet 2 of 7)

```
   78 TM=G*WK*TANH(WK*DM)
      T1=SQRT(TM)
      T2=2.*PAI/T1
      T3=XK(IK)-T2
      T4=ABS(T3)
      IF(T4.LE.0.01) GO TO 76
      DK =-2.*PAI*T3/(XK(IK)*XK(IK)*SQRT(G*DM))
      WK=WK+DK
      GO TO 78
   76 XN=.5*(1.+2.*WK*DM/SINH(2.*WK*DM))
      D1=(XN*4.*PAI*PAI)/(T*T*G*WK*WK)
      WRITE(6,74) D1
   74 FORMAT(//,25X,44HDISPERSION INCLUDED. D1 REPLACED BY C*CG/G =,F10.
     12)
   77 CONTINUE
*I MAIN.101
   73 FORMAT(//,25X,24HWAVE PERIOD IN SECONDS =,F10.2,24X,9HWAVE NO =,E
     112.5,//)
*D MAIN.102,MAIN.103
*D MAIN.106
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,DD,IDPC,DM,IDCH,IDISP)
*D MAIN.112
     1,NLG,IDPC,IBUGA,DM,IDISP,T)
*D MAIN.118
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,VJ,VY,NLGL,IBUGG,ICIRC)
*D MAIN.124
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,IBUGL,ICIRC)
*D MAIN.131,MAIN.138
*D MAIN.145
      IF(ICIRC.NE.1) AF(I)=AF(I)/2.
*I MAIN.191
C     WRITE(6,37) (I,AF(I),AT(I),I=1,NNOD)
C  37 FORMAT(6(I5,2F7.2))
*D MAIN.197
  260 Z(L)=AF(N)
*I MAIN.212
      JP=-1
*I MAIN.213
      JP=JP+2
*D MAIN.218,MAIN.219
      WRITE(6,25) J,AF(J),AT(J),J1,AF(J1),AT(J1)
  501 JPP=JP+1
      AP(JP)=AF(J)
      IF(II.GT.ISPT) GO TO 503
      AP(JPP)=AF(J1)
*D MAIN.221
  503 WRITE(6,24) J,AF(J),AT(J)
*I MAIN.222
      IF(NC.EQ.0) GO TO 103
      WRITE(6,101)
  101 FORMAT(4H CCC)
  103 WRITE(6,201)
  201 FORMAT(//)
      WRITE(6,200) (AP(II),II=1,ISPT)
  200 FORMAT(13F6.2)
*D MAIN.223
  502 CONTINUE
*D MAIN.224,MAIN.234
*D MAIN.240,MAIN.241
      WRITE(6,25) I,AF(I),AT(I),II,AF(II),AT(II)
*D MAIN.243
  415 WRITE(6,24) I,AF(I),AT(I)
*D MAIN.245,MAIN.255
  222 FORMAT(3(I5,1X,3E12.5,1X))
*D MAIN.257
      IF(NC.EQ.0) GO TO 104
```

Figure 29. (Sheet 3 of 7)

```
  WRITE (6,102)
  102 FORMAT(4H [CL)
  104 IF(IPC.EQ.0) GO TO 99
*D MAIN.278,MAIN.280
*D MAIN.282,MAIN.285
   24 FORMAT(1X,I5,2E12.5)
   25 FORMAT(1X,2(I5,E12.5,2X,E12.5))
C     IF(IPMAX.NE.0) CALL CPUNCH(NODPT,IPMAX,AF,U,V,NNOD,NELE,NEQT,XK,IK,
C    1NSP,ISPT,IPER,ID,IPC,NSPE,ISPTE,IKT)
*D INPT.3
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,DD,IDPC,DM,IDCH,IDISP)
*I INPT.8
C     LEVEL 2,NCON,DD
*D INPT.16,INPT.37
      READ(4,200) (X(I),I=1,NNOD)
      READ(4,200) (Y(I),I=1,NNOD)
  200 FORMAT(8F10.2)
C     DO 1000 I=1,NNOD
C     X(I)=X(I)*.1110
C1000 Y(I)=Y(I)*.1110
C     DO 647 I=1055,NNOD
C     II=I-NODB1+1
C     IF(I.LT.1100) AAD=(I-1055)*DA2+.5*DA2
C     IF(I.GE.1100) AAD=(II-1)*DA2
C     RS=160000.
C     IF(I.LT.1100) RS=140000.
C     X(I)=-RS*COS(AAD+ANGG)
C     Y(I)=-RS*SIN(AAD+ANGG)
C 647 CONTINUE
  101 FORMAT(3(5X,3I5))
      READ(4,101) ((NCON(I,J),J=1,3),I=1,NELE)
C     DO 83 I=1,NELE
C  83 DD(I)=6.*DD(I)
      CALL GDEPTH(IDPC,DD,NNOD,NODB2,NELE,IDCH)
      IF(IDCH.EQ.1) GO TO 430
      GO TO 450
  430 PRINT 316,(I,X(I),Y(I),I=1,NNOD)
      DO 440 I=1,NELE
      PRINT 420, I,(NCON(I,J),J=1,3)
  440 CONTINUE
  316 FORMAT(2(I10,2F10.2))
  420 FORMAT(1X,6(I5,1X,3I5))
  450 IF(IDISP.GT.0) GO TO 70
      IF(IDPC.NE.0) GO TO 400
*I INPT.38
      DD(I)=DD(I)/DM
*D INPT.44,INPT.48
   70 CONTINUE
*D INPT.49
  416 FORMAT(4(4X,4I5))
*D INPT.50
      IF(IDCH.EQ.1) PRINT 416, (L,(NCON(L,J),J=1,3),L=1,NELE)
*I DEPTH.7
C     LEVEL 2,DD
*I DEPTH.9
      DO 10 I=1,NELE
   10 DD(I)=0.25
*D DEPTH.10,DEPTH.40
*D DEPTH.43,DEPTH.46
1000  READ(4,980) (DD(I),I=1,NNOD)
  980 FORMAT(8F10.2)
      IF(IDCH.EQ.1)PRINT 2010, (DD(I),I=1,NNOD)
*D DEPTH.47
*I BAND.4
C     LEVEL 2,NCON
*D ASBK.5
```

Figure 29. (Sheet 4 of 7)

```
     1,NLGG,IDPC,IBUG,DM,IDISP,T)
*D ASBK.21
C     LEVEL 2,SYSK,SYSQ,ST,NCON,DD
*D ASBK.73
C     IF(L.GE.1435.AND.L.LE.1571)D9=0.5
*D ASBK.80
      CALL ELMK(X1,X2,X3,Y1,Y2,Y3,WK,D9,DIR,ELK,AREA,L,NELE3,DM,IDISP,T)
*D ASBK.84
 1050 DO 20 I=1,3
*I ASBK.106
      L4=IQ+9
*D ASBK.113
  999 WRITE(6,777)
*D GSPK.3
     1,NELE,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,VJ,VY,NLGG,IBUG,ICIRC)
*D GSPK.14
*D GSPK.12
     1,ANGL(NODR)
*I GSPK.22
      CALL BESSY(WR,0.,NCS,VJ,VY)
      NDUM=NCS
      IF(ICIRC.EQ.1) NDUM=NCS2
      IF(IBUG.EQ.0) GO TO 13
      WRITE(6,21)
   21 FORMAT(//,1X,'XJ=',10X,'XY=')
*D GSPK.23
   13 DO 12 I=1,NDUM
*D GSPK.23,GSPK.26
*I GSPK.31
      IF(IBUG.EQ.0) GO TO 14
      WRITE(6,22)
   22 FORMAT(//,1X,'XH=',10X,'DH=')
*D GSPK.32
   14 DO 15 I=1,NDUM
*D GSPK.43
   20 FORMAT(9H GSPK BY=,4E15.8)
*D GSPK.50
      DO 38 I=1,NCS
      III=I/2+1
*I GSPK.55
      ICE=I
      IF(ICIRC.EQ.1) ICE=III
      IF(ICIRC.EQ.1) CX=2.*CX
*D GSPK.56
      SYSK(I1,1)=SYSK(I1,1)+CX*RKHF*XH(ICE)*DH(ICE)
*I GSPK.60
      CT=1.
      IF(ICIRC.EQ.1) CT=2.
      ND=NODR1
      IF(ICIRC.EQ.1) ND=NODR
*D GSPK.61
      DO 40 I=1,ND
*I GSPK.69
      IF(I1.EQ.0.AND.ICIRC.EQ.1) I1=NODR
*I GSPK.71
      IF(ICIRC.EQ.1) GO TO 935
      GO TO 936
  935 J2=J/2
      J1=J2+1
      A1=ANGL(I)*J2
      A2=ANGL(I1)*J2
  936 CONTINUE
*D GSPK.74
      TM=CT*HKL*DH(1)
      IF(ICIRC.EQ.1  GO TO 46
*I GSPK.77
```

Figure 29. (Sheet 5 of 7)

```
      IF (ICIRC .EQ. 1) GO TO 937
      GO TO 938
  937 IF(MOD(J,2).EQ.1) GO TO 44
      TM=HKL*DH(J1)*(COS(A2)+COS(A1))
      GO TO 46
  938 CONTINUE
*I GSPK.88
      GO TO 46
   44 TM=HKL*DH(J1)*(SIN(A2)+SIN(A1))
*D GSPK.91
*D GSPK.97
C     WRITE(6,1018) (SYSK(I,1),SYSK(I,2),I=1,NLGG)
*D GSPK.98
C1018 FORMAT(1X,4G14.5)
*D GSPK.100
 2000 WRITE(6,999)
*D GSPK.103
 1551 WRITE(6,1520) NLLL
*D LOAD.3
     1,NELS,NNOD,NODR,NCS,NEQT,NBAND,NCS2,NODR1,IBUG,ICIRC)
*D LOAD.9
      DIMENSION SM(NODR),SYSQ(NEQT),DH(NCS),XJ(NCS),ANGL(NODR)
*D LOAD.10
*I LOAD.15
      NDUM=NODR1
      IF(ICIRC.EQ.1) NDUM=NODR
*D LOAD.16
      DO 12 I=1,NDUM
*I LOAD.23
      IF(ICIRC.EQ.1) DUM2=(0.0,0.0)
*I LOAD.31
      IF(ICIRC.EQ.1) GO TO 52
*I LOAD.33
   52 CONTINUE
*I LOAD.34
      IF(II.EQ.0.AND.ICIRC.EQ.1) II=NODR
*I LOAD.49
      IF(ICIRC.EQ.1) II=I/2
      IJ=II+1
*I LOAD.51
      IF(ICIRC.EQ.1) GO TO 555
      GO TO 556
  555 IF(I.EQ.1) GO TO 557
      IF(MOD(I,2).EQ.1) GO TO 18
      TM=CM*XJ(IJ)*DH(IJ)*COS(AI)
      GO TO 20
  557 TM=XJ(1)*DH(1)
      GO TO 20
   18 TM=CM*XJ(IJ)*DH(IJ)*SIN(AI)
      GO TO 20
  556 CONTINUE
*I LOAD.52
   20 CONTINUE
*D ELMT.3
     1NELE3,DN,IDISP,T)
*I ELMT.25
C     WRITE(6,500) X1,X2,X3,Y1,Y2,Y3
*I ELMT.30
      XK1=WK/SQRT(D)
      IF(IDISP.NE.1) GO TO 300
      XK1=XK1*SQRT(DN)
      PI=3.1416
  150 TM=G*XK1*TANH(XK1*D)
      T1=SQRT(TM)
      T2=2.*PI/T1
      T3=T-T2
```

Figure 29. (Sheet 6 of 7)

```
      T4=ABS(T-T2)
      IF(T4.LE..01) GO TO 200
      DK=-2.*PI*T3/(T*T*SQRT(G*D))
      XK1=XK1+DK
      GO TO 150
  200 CONTINUE
      XN=.5*(1.+2.*XK1   *D/SINH(2.*XK1*D))
      XD=(XN*4.*PI*PI)/(T*T*G*XK1*XK1)
      GO TO 400
  300 XD=D
  400 CONTINUE
*D ELMT.31,ELMT.33
      WKA=DIN*XK1*XK1*AREA/12.
      SAV=D
      D=XD
*D ELMT.43
      D=SAV
*I BESY.11
      FN=N
*D BESY.80
      GAM3=-GAM3*GAM2
*D BESJ.38
    5 F(I)=CONST*(FLI+FNU)*F(I+1)-F(I+2)
*D BESJ.76
   21 F(L)=CONST*FRAC*F(L-1)-F(L-2)
*D BESJ.202
  114 F(I)=CONST*(FNU+FLI)*F(I-1)-F(I-2)
*D GAMA.53,GAMA.56
      XFT=EXP(TEM)/XFT
      ANS=XFT
      GO TO 6000
 4000 SUM=0.0
*I BANDS.5
      DIMENSION Q(100,100),R(100,100),ML(100),NL(100)
*D BANDS.7
*D BANDS.44
      IF(REAL(A(N,L))) 230,275,230
*D BANDS.62
      F(N,M)=A(N,M)
  127 CONTINUE
*D BANDS.74
      A(NM,NB)=B(N)
  450 CONTINUE
*D BANDS.85
      A(N,M)=F(N,M)
  136 CONTINUE
*D BANDS.93
      D(K)=A(NM,NB)
  600 CONTINUE
*D BANDS.96
  444 WRITE(6,666)N
*D BANDS.101,BANDS.102
  999 WRITE(6,998) A(N,1),N,NB
  998 FORMAT(13H DIAG TROUBLE  ,5X,2G13.5,2I5)
*D FILL.17
      WRITE(6,666)N
*D FILL.5
      DIMENSION Q(100,100),R(100,100),NL(100),ML(100)
*D ECWR.4
*D ECWR.7,ECWR.9
      BUFFER OUT(L1,0)(S(1),S(L2))
      IF(UNIT(L1))10,20,30
   30 WRITE(6,40)
   40 FORMAT(1X,'PARITY ERROR')
   20 IERR=1
   10 RETURN
*D ECRD.4
*D ECRD.7,ECRD.9
      BUFFER IN(L1,0)(S(1),S(L2))
      IF (UNIT(L1))10,20,30
   30 WRITE(6,40)
   40 FORMAT(1X,'PARITY ERROR')
   20 IERD=1
   10 RETURN
```

Figure 29. (Sheet 7 of 7)

(*Note: File FINDAT1 contains Data Set No. 1 through Data Set No. 3. File FINDAT contains Data Set No. 4 through Data Set No. 7. File FINDAT can be obtained at the completion of the run of the plotting program FNGRID. If that is the case, the new file name replacing FINDAT should correspond with the file name that was used to save TAPE10 in the procedure file PLOTF.)

g. Replace file FINI with the edited version
   Type in
   E,,RL

h. To execute file FINI
   Type in
   SUBMIT,FINI

i. To check job status
   Type in
   LINK,ENQUIRE,PD=MODD,PT=HHMM
   (PD=MODD, where MO is the month from 01 to 12, and DD is the date from 01 to 31. PT=HHMM, where HH is the hour from 01 to 24, and MM is the minute from 01 to 60. Example: To generate status information for a job received since 9:00 a.m. on 31 May, type in
   LINK,ENQUIRE,PD=0531,PT=0900
   The last three characters of the run JOB NAME will not stay the same as in initiated job names. The system assigns different names at different stages that transmit a job from KOE to CYBER 205, or vice versa. If the job output is available at KOE, then continue to the next step.)

j. To execute "QLIST"
   Type in
   OLD,QLIST
   BEGIN,,QLIST,XXX,LSV,DSV
   (where XXX are the last three characters of the job name that appeared in the statement "JOB NAME IS IN USER QGET QUEUE." LSV causes the output to be saved with file name HASHXXX. DSV causes the dayfile to be saved with file name DAYXXX.

k. To list output and dayfile on the user's terminal
   Type in
   OLD,FILE NAME
   LIST

l. To direct output and dayfile to a fast printer at COPE terminal
   Type in
   GET,LFN=FILE NAME
   ROUTE,LFN,DC=PR,UN=COPE USER'S ID
   (where LFN is the local file name, and FILE NAME is the permanent file name)

A Sample Application

55. In this sample application to demonstrate the execution of program

FINITE on the CYBER 205 COMPUTER SYSTEM, the values of the parameters in the parameter statements stay unchanged.

a. Obtain the source file FINITE
   Type in
   GET,FINITE/UN=CEROMO
   /SAVE,FINITE

b. Establish AFFILE "AF205" on CYBER 205

   1. To execute file AFJCL

```
 GET,AFJCL/UN=CEROMO
/SAVE,AFJCL
/OLD,AFJCL
/XEDIT
 XEDIT 3.1.00
?? C/205USER/052010/
USER(U=052010,PA=PASSWORD)
?? C/PASSWORD/IODINE/*
USER(U=052010,PA=IODINE)
--EOR--
USER(KOEUSER,IODINE,KOE)
END OF FILE
?? T
?? C/KOEUSER/CEROMB/*
CHARGE(CEROEGC,CEROMB)
--EOR--
USER(CEROMB,IODINE,KOE)
CHARGE(CEROEGC,CEROMB)
END OF FILE
?? E,,RL
AFJCL    REPLACED
AFJCL    IS A LOCAL FILE
AEB ,       0.257UNTS.
/GET,ADYJOB/UN=CATHDDC
/BEGIN,,ADYJOB,AFJCL
 ALL DONE.  USER JOB NAME IS COE429
$REVERT.
/
```

   2. Use "LINK,ENQUIRE" to check job status

```
LINK,ENQUIRE,JN=COE429
         STATUS FOR CEROMB /KOE

DATE TIME    USER      SYSTEM    FILE   FILE
MMDD HHMM  JOB NAME  JOB NAME  TYPE  STATUS

0522 1020  COE429    AJZZ305   NO    ROUTING INITIATED TO KAA
0522 1244  COE429    SENDLOT   TO    ARRIVED AT ADY
0522 1245  COE429    AUUALOO   NO    ARRIVED AT KAA
0522 1245  COE429    AUUALOO   TO    AUUALOO LINKING TO ADY
0522 1246  COE429    AJZZLOT   WT    OUTPUT AVAILABLE AT KOE/T
LINK COMPLETE.
/
```

3. Use "QLIST" to access the output

```
 OLD,QLIST
/BEGIN,,QLIST,LQT
  11.44.47 RESOURCE(JCAT=P6,TL=200)
  11.44.49 CHARGE, CEROEGC,CEROMB
  11.44.49  ALL DONE
  11.44.49 TV,10+.
  11.44.49 PURGE(AF205)
  11.44.50  ALL DONE
  11.44.50 TV,4+.
  11.44.50 COPY,INPUT,AF205.
  11.44.51        9 WORDS OF FILE  INPUT    COPIED TO FILE AF205
  11.44.51  ALL DONE
  11.44.51 DEFINE,AF205.
  11.44.51 EXISTING LOCAL FILE MADE PERMANENT
  11.44.51  ALL DONE
  11.44.51    CHARGE,CEROEGC ,CEROMB                          3.982 SBUS
  11.44.51 SYSTEM BILLING UNITS (SBU)                         3.982
  11.44.51 USER CPU TIME (SECS)                                .194
  11.44.51 SYSTEM CPU TIME (SECS)                              .693
  11.44.51 USER MEMORY USAGE (PAGE*SECS)                     18.731
  11.44.51 USER AVERAGE WORKING SET SIZE (PAGES)             96
  11.44.51 NUMBER OF VIRTUAL SYSTEM REQUESTS                235
  11.44.51 NUMBER OF SMALL PAGE FAULTS                       73
  11.44.51 NUMBER OF DISK I/O REQUESTS                       11
  11.44.51 NUMBER OF DISK SECTORS TRANSFERRED                11
  11.44.51 $$COMPLETE$$
END OF FILE
/
```

c. Establish the file "QLIST"

```
 GET,QLIST/UN=CEROMO
/SAVE,QLIST
/OLD,QLIST
/XEDIT
 XEDIT 3.1.00
?? C/HASH/AJZZ/
.PROC,QLIST,FID,LSV=N/Y,DSV=N/Y,DSP=N/Y,UID=AJZZ,UN=USERNUM,AUN=CHENDDC,
?? C/USERNUM/CEROMB/
.PROC,QLIST,FID,LSV=N/Y,DSV=N/Y,DSP=N/Y,UID=AJZZ,UN=CEROMB,AUN=CHENDDC,
?? C/HEADER1/LUCYCHOU/
H1=LUCYCHOU,H2=HEADER2,H3=HEADER3.
?? C/HEADER2/CERC/
H1=LUCYCHOU,H2=CERC,H3=HEADER3.
?? C/HEADER3/WES/
H1=LUCYCHOU,H2=CERC,H3=WES.
?? E,,RL
QLIST   REPLACED
QLIST   IS A LOCAL FILE
AEB ,      0.254UNTS.
/
```

d. *Create an Update Library*

```
 GET,FINUP1/UN=CEROMO
/SAVE,FINUP1
/BEGIN,RUNF,FINUP1
 15.56.41. SUBMIT COMPLETE.  JOBNAME IS AJZZFOT
$REVERT.CCL
/ENQUIRE,JN=FOT
 AJZZFOT  NOT FOUND.
/OLD,DAY1
/LIST
 15.56.41.FINIJOB,CM320000,P6.
 15.56.41.ABF , INPUT  ,   0.002KIODB,01.
 15.56.41.USER,CEROM8,,KOE.
 15.56.41.ABG , P6.
 15.56.42.CHARGE,CEROEGC,CEROM8.
 15.56.43.GET,FINITE.
 15.56.44.UPDATE(I=FINITE,N=LIBLC)
 15.56.44.UPDATE CREATION RUN
 15.56.44.CREATING NEW PROGRAM LIBRARY
 15.56.46. UPDATE COMPLETE.
 15.56.46.SAVE,LIBLC=LIBLC.
 15.56.47.DAYFILE,DAY1.
/
```

e. Establish an Update Correction File "FINDIR". (In the sample run, file FINDIR is used without change.)

Type in

```
GET,FINDIR/UN=CEROMO
/SAVE,FINDIR
/
```

f. To execute program FINITE with Update Correction file FINDIR

1. *Execute file FINI*

```
 GET,FINI/UN=CEROMO
/SAVE,FINI
/OLD,FINI
/XEDIT
 XEDIT 3.1.00
?? C/205USER/052010/
USER(U 052010,PW=PASSWORD)ADY
?? C/PASSWORD/IODINE/
USER(U-052010,PW=IODINE)ADY
?? C/LOEUSER/CEROM8/
CHARGE(CEROEGC,CEROM8)
GETKOE,COMPF/DD=C6,UN=CEROM8.
LINK,REPLACE(LIST LIST205/UN CEROM8,FM=KOE,DD=C6,AF=AF205)
LINK,REPLACE(FNMAP=MAP205/UN=CEROM8,FM=KOE,DD=C6,AF=AF205)
LINK,REPLACE(LIST LIST205/UN=CEROM8,FM=KOE,DD=C6,AF=AF205)
LINK,REPLACE(FNMAP=MAP205/UN=CEROM8,FM=KOE,DD=C6,AF=AF205)
END OF FILE
?? E,,RL
FINI     REPLACED
FINI     IS A LOCAL FILE
AEB ,       0.255UNTS.
/SUBMIT,FINI
 14.24.02. SUBMIT COMPLETE.  JOBNAME IS AJZZWOA
/
```

2. Check job status

```
LINK,ENQUIRE,PD=0531,PT=0100
          STATUS FOR CEROH8 /KOE

DATE TIME    USER      SYSTEM   FILE   FILE
MMDD HHMM  JOB NAME  JOB NAME  TYPE  STATUS

0531 1424  COESSS    AJZZWQA   NO    ROUTING INITIATED TO KAA
0531 1433  COESSS    LUCYGIJ   TO    ARRIVED AT ADY
0531 1433  COESSS    AUUAGIH   NO    ARRIVED AT KAA
0531 1434  COESSS    AUUAGIH   TO    AUUAGIH LINKING TO ADY
0531 1439  COESSS    AJZZGIJ   WT    OUTPUT AVAILABLE AT KOE/T
LINK COMPLETE.
/
```

3. To retrieve job output and save it as a permanent file
   Type in
   OLD,QLIST
   /BEGIN,,QLIST,GIJ,LSV,DSV

4. To route the output to the remote COPE terminal
   Type in
   GET,OUTLUCY=AJZZGIJ
   /ROUTE,OUTLUCY,DC=PR,UN=CERORC
   The system will respond with
   ROUTE COMPLETE
   /

5. A lis.ing of the sample application output from the COPE terminal includes the following:

```
/OLD,AJZZGIJ
/LIST

                                    SAMPLE PROBLEM


NUMBER=     1

TOTAL NO. OF ELEMENTS      48   TOTAL NO. OF NODES      35   TOTAL NO. OF SOURCES       9   FIRST ELEMENT W.R.T. SOURCES      7
TOTAL NO. OF SIM. LINEAR EQS      42    WAVE ANGLE OF INCIDENCE      4.71229    INCREMENT IN ANG. DIRECTION        .3927      .1963
OUTEST EXTEND OF FINITE DOMAIN        12.00     NORM. DEPTH OF OUTER REGION   1.00     AVER. DEPTH(FT) OF SEMICIRCLE        1.00

                                                  INPUT DATA CHECK--ECHO PRINT

    1     42     42 20000      1     16
  141.76        .00     1
   1     2     3     4     5     6     7     8     9    10    11    12    13    14
  15    16    17    18    19    20    21    22    23    24    25    26    27    28
  29    30    31    32    33    34    35
   1     2     3     4     5     6     7     8     9    10    11    12    13    14
  15    16    17    18    19    20    21    22    23    24    25    26    27    28
  29    30    31    32    33    34    35    36    37    38    39    40    41    42
  43    44    45    46    47    48

                        WAVE PERIOD IN SECONDS =     141.76                        WAVE NO     .78108E-02


0000000000000000000000000000000000000000
BAND WIDTH     16
BAND WIDTH W.R.T. ELEMENT     15
BAND WIDTH W.R.T. SOURCE + BDARY=     16
0000000000000000000000000000000000000000


XJ            XY
GSFK BY    .99780487E+00 -.15760292E+01
GSFK BY    .46813616E-01 -.68809307E+01
GSFK BY    .10973647E-02 -.14524819E+03
GSFK BY    .17145838E-04 -.61916854E+04
GSFK BY    .20070738E-06 -.39620639E+06
GSFK BY=   .18832465E-08 -.33810568E+08
GSFK BY=   .14710523E-10 -.36068268E+10
```

```
XH=          DH=
GSFN BY    .99780487E+00 -.15760292E+01 -.46813646E-01  .68809307E+01
GSFN BY    .46813646E-01 -.68809307E+01  .49835375E+00  .[illegible]
GSFN BY    .10973647E-02 -.14521319E+03  .23395250E-01  .3[illegible]
GSFN BY    .17115838E-04 -.61916854E+04  .54858189E-03  .1[illegible]
GSFN BY    .20090738E-06 -.39620639E+06  .85719772E-05  .16902[illegible]
GSFN BY    .19832465E-08 -.33810563E+08  .10044631E-06  .13032153E+10
GSFN BY    .14710523E-10 -.36068268E+10  .94157400E-09  .23085190E+12

GSFN COMPLETE

                                ELEVATION AND VELOCITY FOR SELECTED ELEVATION STATIONS [illegible]

   1  .12310E+01    .18324E-01    2  .12310E+01    .18324E-01
   3  .12310E+01    .18324E-01    4  .12278E+01    .18324E-01
   5  .12278E+01    .18324E-01    6  .12214E+01    .18324E-01
   7  .12214E+01    .18324E-01    8  .12214E+01    .18324E-01
   9  .12086E+01    .18324E-01   10  .12086E+01    .18324E-01
  11  .11927E+01    .18324E-01   12  .11927E+01    .18324E-01
  13  .11927E+01    .18324E-01   14  .11707E+01    .18325E-01
  15  .11707E+01    .18324E-01   16  .11455E+01    .18325E-01
  17  .11455E+01    .18324E-01   18  .11455E+01    .18323E-01
  19  .11145E+01    .18326E-01   20  .11144E+01    .18323E-01
  21  .10713E+01    .18323E-01   22  .10896E+01    .18332E-01
  23  .10713E+01    .18310E-01   24  .10599E+01    .1831[illegible]E-01
  25  .10598E+01    .18336E-01   26  .10408E+01    .18370E-01
  27  .10349E+01    .18264E-01   28  .10310E+01    .18[illegible]E-01
  29  .10255E+01    .18332E-01   30  .10220E+01    .18383E-01
  31  .10205E+01    .18409E-01   32  .10220E+01    .18394E-01
  33  .10257E+01    .18352E-01   34  .10307E+01    .18310E-01
  35  .10351E+01  .18291E-01

1.23  1.23  1.23  1.23  1.23  1.22  1.22  1.22  1.21  1.21  1.19  1.19  1.19
1.17  1.17  1.15  1.15  1.15  1.11  1.11  1.07  1.09  1.07  1.06  1.06  1.04
1.03  1.03  1.03  1.02  1.02  1.02  1.03  1.03  1.04

                                                        SOURCE POINTS

  36  .18312E-01    .15891E+01   37  .30876E-05    .13680E+01
  38  .27817E-01    .15839E+01   39  .30395E-08    .1[illegible]E+01
  40  .19159E-08    .15893E+01   41  .23125E-12   -.15533E+01
  42  .22165E-13  .15873E+01
```

VELOCITY FOR SELECTED VELOCITY STATIONS(ELEMENTS)

1	.26390E-05	.58102E+00	.58102E+00	2	.26390E-05	.58103E+00	.58103E+00	3	-.29003E+00	.87106E+00	.91808E+00
4	.78724E-05	.11611E+01	.11611E+01	5	.29004E+00	.87106E+00	.91808E+00	6	-.29003E+00	.87106E+00	.91808E+00
7	.78724E-05	.11611E+01	.11611E+01	8	.29004E+00	.87107E+00	.91809E+00	9	.78724E-05	.23141E+01	.23141E+01
10	.78724E-05	.23141E+01	.23141E+01	11	-.28549E+00	.25996E+01	.26152E+01	12	.44286E-04	.28851E+01	.28851E+01
13	.28555E+00	.25996E+01	.26153E+01	14	-.28549E+00	.25996E+01	.26152E+01	15	.44374E-04	.28852E+01	.28852E+01
16	.28555E+00	.25997E+01	.26153E+01	17	.44286E-04	.40110E+01	.40110E+01	18	.44374E-04	.40112E+01	.40112E+01
19	-.27654E+00	.42876E+01	.42965E+01	20	-.14623E-04	.45641E+01	.45641E+01	21	.27656E+00	.42875E+01	.42964E+01
22	-.27626E+00	.42875E+01	.42964E+01	23	.52785E-03	.45643E+01	.45643E+01	24	.27684E+00	.42880E+01	.42969E+01
25	-.14623E-04	.56450E+01	.56450E+01	26	.52785E-03	.56462E+01	.56462E+01	27	-.10953E+01	.67403E+01	.68287E+01
28	-.16641E+01	.61715E+01	.63919E+01	29	-.56877E+00	.50762E+01	.51080E+01	30	.57051E+00	.50762E+01	.51082E+01
31	.16670E+01	.61728E+01	.63939E+01	32	.10970E+01	.67427E+01	.68314E+01	33	-.65808E+01	.19785E+01	.68717E+01
34	-.16641E+01	.37376E+01	.40913E+01	35	.16670E+01	.37486E+01	.41026E+01	36	.66083E+01	.20523E+01	.69197E+01
37	-.25926E+01	.46662E+01	.53380E+01	38	.69743E-02	.54087E+01	.54087E+01	39	.25908E+01	.46721E+01	.53426E+01
40	-.27527E+01	.27718E+01	.39064E+01	41	.69743E-02	.34567E+01	.34567E+01	42	.27527E+01	.27811E+01	.39153E+01
43	-.17968E+01	.37276E+01	.41381E+01	44	-.38340E+00	.38471E+01	.38662E+01	45	.38510E+00	.38348E+01	.38541E+01
46	.18238E+01	.37133E+01	.41370E+01	47	-.48914E+00	.36902E+01	.37225E+01	48	.48259E+00	.36902E+01	.37216E+01

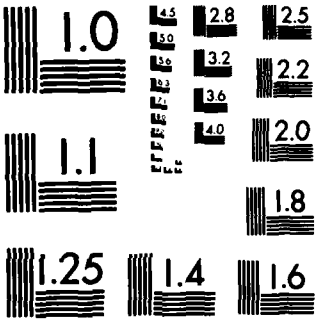
```
1$$  STUP  $$
1 13.28.40 R592C          V592C          052010  REL21B   I      LUCYGIJ          05/31/84
  13.28.40 RESOURCE(JCAT=P6,TL=100)
  13.28.44 CHARGE, CEROEGC,CEROM8
  13.28.44  ALL DONE
  13.28.44 PATTACH,LINKPOOL.
  13.28.45 POOL    LINKPOOL ATTACHED
  13.28.45  ALL DONE
  13.28.45 TV,10+.
  13.28.45 ***
  13.28.45 ***GETKOE,COMPF/DD=C6,UN=CEROM8.
  13.28.45 ***
  13.28.45 MFLINK(COMPF,ST=KAA,DD=C6,
  13.28.45 JCS='*********************',
  13.28.45 '*****************')
  13.28.46  WAITING FOR CONNECTED STATUS.
  13.28.52
  13.28.52   PTFS  - USER(COEAPPL,)
  13.28.52
  13.28.52   PTFS  - ATTACH(COE556/NA)
  13.28.55
  13.28.55   PTFS  - PF REQUEST COMPLETE.
  13.28.56  ALL DONE
  13.28.56 ***
  13.28.56 ***GETKOE(TAPE8=FINDAT1)
  13.28.56 ***
  13.28.56 MFLINK(TAPE8,ST=KAA,DD=C6,
  13.28.56 JCS='*********************',
  13.28.56 '*****************')
  13.28.57  WAITING FOR CONNECTED STATUS.
  13.29.04
  13.29.04   PTFS  - USER(COEAPPL,)
  13.29.04
  13.29.04   PTFS  - ATTACH(COE557/NA)
  13.29.05
  13.29.05   PTFS  - PF REQUEST COMPLETE.
  13.29.06  ALL DONE
```

```
13.29.06 ***
13.29.06 ***GETROE(TAPE4=FINDAT)
13.29.06 ***
13.29.06 MFLINK(TAPE4,ST=KAA,DD=C6,
13.29.06 JCS='*********************',
13.29.06 '*****************')
13.29.07  WAITING FOR CONNECTED STATUS.
13.29.14
13.29.14   PTFS  - USER(COEAPPL,)
13.29.14
13.29.14   PTFS  - ATTACH(COE553/NA)
13.29.16
13.29.16   PTFS  - PF REQUEST COMPLETE.
13.29.17  ALL DONE
13.29.17 FORTRAN(I COMPF,L=LIST)
13.29.18 FORTRAN 2.1.5 CYCLE L592C     BUILT 05/01/84 16:21
13.29.18 COMPILING FINIT
13.29.24        NO ERRORS
13.29.24 COMPILING CPUNCH
13.29.24        NO ERRORS
13.29.24 COMPILING INPUT2
13.29.24        NO ERRORS
13.29.24 COMPILING GDEPTH
13.29.24        NO ERRORS
13.29.24 COMPILING BAND
13.29.24        NO ERRORS
13.29.24 COMPILING ASEMBK
13.29.24        NO ERRORS
13.29.24 COMPILING GSPK
13.29.25        NO ERRORS
13.29.25    FILE BINARY   EXTENDED, NEW LENGTH =      32
13.29.25 COMPILING LOAD
13.29.25        NO ERRORS
13.29.25 COMPILING ELMK
13.29.25        NO ERRORS
13.29.25 COMPILING BESSY
13.29.25        NO ERRORS
13.29.25 COMPILING GAMMA
13.29.25        NO ERRORS
13.29.25 COMPILING BESSJ
13.29.26        NO ERRORS
13.29.26 COMPILING BANSOL
13.29.26        NO ERRORS
13.29.26 COMPILING FILL
13.29.26        NO ERRORS
13.29.26 COMPILING ECWR
13.29.26        NO ERRORS
13.29.26 COMPILING ECRD
13.29.26        NO ERRORS
13.29.26      1.944 SECONDS COMPILATION TIME
13.29.26  ALL DONE
13.29.26 LOAD,L FNMAP.
13.29.27  LOAD R2.1 CYCLE L592C
13.29.37  ALL DONE
13.29.37 GO(TAPE6 OUTPUT)
13.29.41  **  STOP  **
13.29.41  ALL DONE
13.29.41 LINK,REPLACE(LIST LIST205/UN=CEROM8,FM=KOF,DD=C6,AF=AF205)
13.29.41  LINK PHASE II  VERSION LLINK50  BUILT  05/01/84        14.47.59
```

```
13.29.44 REPLACE(LIST205/UN CEROHB.
13.29.44 L3882,P6.  KOE JOB  LLINK50 3882
13.29.51  WAITING FOR CONNECTED STATUS.
13.30.03  TRANSFER ABORTED
13.30.10  ALL DONE
13.30.10 LINK,REPLACE(FNMAP-MAP205/UN-CEROHB,FM KOE,DD C6,AF AF205)
13.30.11  LINK PHASE II  VERSION LLINK50  BUILT  05/01/84        14.47.59
13.30.11 REPLACE(MAP205/UN CEROHB.
13.30.12 L3883,P6.  KOE JOB  LLINK50 3883
13.30.19  WAITING FOR CONNECTED STATUS.
13.30.31  TRANSFER ABORTED
13.30.36  ALL DONE
13.30.36 NFLINK(NULL,ST KAA,DD=C6,
13.30.36 JCS-'*****************',
13.30.36 '*****************')
13.30.40  WAITING FOR CONNECTED STATUS.
13.30.49
13.30.49   PTFS  - USER(COEAPPL,)
13.30.49
13.30.49   PTFS  - PURGE(COE558/NA)
13.30.49
13.30.49   PTFS  - PF REQUEST COMPLETE.
13.30.50  ALL DONE
13.30.50 NFLINK(NULL,ST KAA,DD=C6,
13.30.50 JCS-'*****************',
13.30.50 '*****************')
13.30.52  WAITING FOR CONNECTED STATUS.
13.30.58
13.30.58   PTFS  - USER(COEAPPL,)
13.30.58
13.30.58   PTFS  - PURGE(COE557/NA)
13.30.58
13.30.58   PTFS  - PF REQUEST COMPLETE.
13.31.00  ALL DONE
13.31.00 NFLINK(NULL,ST KAA,DD=C6,
13.31.00 JCS-'*****************',
13.31.00 '*****************')
13.31.01  WAITING FOR CONNECTED STATUS.
13.31.08
13.31.08   PTFS  - USER(COEAPPL,)
13.31.08
13.31.08   PTFS  - PURGE(COE556/NA)
13.31.08
13.31.08   PTFS  - PF REQUEST COMPLETE.
13.31.09  ALL DONE
13.31.09    CHARGE,CEROEGC ,CEROHB                          71.367 SBUS
13.31.09 SYSTEM BILLING UNITS (SBU)                         71.367
13.31.09 USER CPU TIME (SECS)                                7.632
13.31.09 SYSTEM CPU TIME (SECS)                              5.233
13.31.09 USER MEMORY USAGE (PAGE*SECS)                    1742.860
13.31.09 USER AVERAGE WORKING SET SIZE (PAGES)             228
13.31.09 NUMBER OF VIRTUAL SYSTEM REQUESTS                1907
13.31.09 NUMBER OF SMALL PAGE FAULTS                       850
13.31.09 NUMBER OF DISK I/O REQUESTS                       29.
13.31.09 NUMBER OF DISK SECTORS TRANSFERRED                62.
13.31.10 $$COMPLETE$$
```

AD-A149 467 EROSION CONTROL OF SCOUR DURING CONSTRUCTION REPORT 6 2/2
FINITE - A NUMERICA..(U) ARMY ENGINEER WATERWAYS
EXPERIMENT STATION VICKSBURG MS HYDRA..
UNCLASSIFIED J R HOUSTON ET AL. SEP 84 WES/TR/HL-80-3-6 F/G 13/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

6. Listing of Data File FINDAT1

```
OLD,FINDAT1
/LIST
SAMPLE PROBLEM
   16    1     12.00    4.71229      1.00 20000
    1    0     0    0    0    1    0    0    0    0
    141.76     0.00    1
    1    2    3    4    5    6    7    8    9   10   11   12   13   14
   15   16   17   18   19   20   21   22   23   24   25   26   27   28
   29   30   31   32   33   34   35
    1    2    3    4    5    6    7    8    9   10   11   12   13   14
   15   16   17   18   19   20   21   22   23   24   25   26   27   28
   29   30   31   32   33   34   35   36   37   38   39   40   41   42
   43   44   45   46   47   48
/
```

7. Listing of Data File FINDAT

```
OLD,FINDAT
LIST
   -8.00      0.00      8.00     -4.00      4.00     -8.00      0.00      8.00
   -4.00      4.00     -8.00      0.00      8.00     -4.00      4.00     -8.00
    0.00      8.00     -4.00      4.00     -8.00      0.00      8.00     -4.00
    4.00      0.00     12.00     11.00      8.50      4.60      0.00     -4.60
   -8.50    -11.10    -12.00
  -32.00    -32.00    -32.00    -28.00    -28.00    -24.00    -24.00    -24.00
  -20.00    -20.00    -16.00    -16.00    -16.00    -12.00    -12.00     -8.00
   -8.00     -8.00     -4.00     -4.00      0.00      0.00      0.00      4.00
    4.00      8.00      0.00      4.60      8.50     11.10     12.00     11.10
    8.50      4.60      0.00
       1    2    4         2    3    5         1    4    6
       4    7    6         2    7    4         2    5    7
       5    8    7         3    8    5         6    7    9
       7    8   10         6    9   11         9   12   11
       7   12    9         7   10   12        10   13   12
       6   13   10        11   12   14        12   13   15
      11   14   16        14   17   16        12   17   14
      12   15   17        15   18   17        13   18   15
      16   17   19        17   18   20        16   19   21
      19   22   21        17   22   19        17   20   22
      20   23   22        18   23   20        35   21   34
      21   22   24        22   23   25        23   27   28
      21   24   34        22   25   24        23   28   25
      34   24   33        24   25   26        25   28   29
      24   32   33        24   26   32        25   30   26
      25   29   30        26   31   32        26   30   31
     .25       .25       .25       .25       .25       .25       .25       .25
     .25       .25       .25       .25       .25       .25       .25       .25
     .25       .25       .25       .25       .25       .25       .25       .25
     .25       .25       .25       .25       .25       .25       .25       .25
     .25       .25       .25
```

PART V: SUMMARY

56. A two-dimensional finite element numerical simulation model (FINITE) was developed that calculates wave heights under combined refraction and diffraction of both long and short waves approaching structures from any arbitrary direction. The wave equation solved governs the propagation of periodic, small amplitude surface gravity waves over a variable depth seabed of mild slope. The efficient formulation of the model permits the solution of large problems with relatively small time and memory storage requirements. A computational scheme is employed that allows the solution of practical problems that typically require large computational grids.

57. Although the model solves an equation that is strictly valid only for mild bathymetric variations, the model can provide reasonable answers for problems where there are rapid depth variations (at much lower cost than required by three-dimensional models that are appropriate for problems involving rapid depth variations). Comparisons are presented between the finite element model calculations and an analytical solution, a two-dimensional numerical solution, a three-dimensional numerical solution, and measurements from laboratory studies. Because the finite element FINITE does not provide a mechanism for energy dissipation, energy loss through wave breaking can be simulated only by permitting waves to propagate out of the computational region. The program documentation, user guide, and sample problem output are provided.

REFERENCES

Abbott, M. B., Petersen, H. M., and Skovgaard, O. 1978. "On the Numerical Modeling of Short Waves in Shallow Water," Journal of Hydraulic Research, Vol 16, p 173.

Aranha, J. A., Mei, C. C., and Yue, K. K. P. 1979. "Some Properties of Hybrid Element Methods for Water Waves," International Journal of Numerical Methods in Engineering, Vol 14, p 1627.

Berkhoff, J. C. W. 1972. "Computation of Combined Refraction-Diffraction," Proceedings, Thirteenth International Conference on Coastal Engineering, Vancouver, British Columbia, Canada, Vol 1, pp 471-490.

_____. 1976. "Mathematical Models for Simple Harmonic Linear Water Waves," Technical Report No. 163, Delft Hydraulics Laboratory, Delft, Netherlands.

Bettess, P., and Zienkiewicz, O. C. 1977. "Diffraction and Refraction of Surface Waves Using Finite and Infinite Elements," International Journal of Numerical Methods in Engineering, Vol 2, p 1271.

Chen, H. S., and Mei, C. C. 1974. "Oscillations and Wave Forces in an Offshore Harbor," Technical Report No. 190, Massachusetts Institute of Technology, Cambridge, Mass.

Committee for the Calculation of Mathematical Tables. 1958. "British Association for the Advancement of Science, Mathematical Tables, Bessel Functions, Part I, Functions of Orders Zero and Unity," University Press, Cambridge.

Eckart, C. 1952. "The Propagation of Gravity Waves from Deep to Shallow Water," Circular No. 521, National Bureau of Standards, Washington, D. C.

Hales, L. Z. 1980 (Sep). "Erosion Control of Scour During Construction; Experimental Measurements of Refraction, Diffraction, and Current Patterns near Jetties," Technical Report HL-80-3, Report 3, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Hom-ma, S. 1950. "On the Behavior of Seismic Sea Waves Around Circular Islands," Geophysical Magazine, Vol 21, p 199.

Jonsson, I. G., and Brink-Kjaer, O. A. 1973. "A Comparison Between Two Reduced Wave Equations for Gradually Varying Depth," Technical Report No. 31, Technical University of Denmark.

Jonsson, I. G., Skovgaard, O., and Brink-Kjaer, O. A. 1976. "Diffraction and Refraction Calculations for Waves Incident on an Island," Journal of Marine Research, Vol 34, p 469.

Lamb, H. 1932. Hydrodynamics, Cambridge University Press.

Lautenbacher, C. C. 1970. "Gravity Wave Refraction by Islands," Journal of Fluid Mechanics, Vol 41, p 655.

Liu, P. L. F., and Lozano, C. J. 1979. "Combined Wave Refraction and Diffraction," Proceedings, Coastal Structures 1979, Vol 2, p 978.

Mobarek, I. 1962. "Effect of Bottom Slope on Wave Diffraction," Technical Report No. HEL-1-1, University of California, Berkeley, Calif.

Pierson, W. J. 1951. "The Interpretation of Cross Orthogonals in Wave Refraction Phenomena," Technical Memorandum No. 21, US Army Engineer Beach Erosion Board, Washington, D. C.

Putnam, J. A., and Arthur, R. S. 1948. "Diffraction of Water Waves by Breakwaters," Transactions, American Geophysical Union, Vol 29, p 481.

Radder, A. C. 1979. "On the Parabolic Equation Method for Water Wave Propagation," Journal of Fluid Mechanics, Vol 95, p 159.

Schonfeld, J. 1972. "Propagation of Two-Dimensional Short Waves," Delft University of Technology.

Smith, R., and Sprinks, T. 1975. "Scattering of Surface Waves by a Conical Island," Journal of Fluid Mechanics, Vol 72, p 373.

US Army Engineer Waterways Experiment Station. 1979 (Oct) "Graphics Compatibility System (GCS), Programmer's Reference Manual," Miscellaneous Paper 0-79-5, Vicksburg, Miss.

Yue, D. K. P., Chen, H. S., and Mei, C. C. 1976. "A Hybrid Finite Element Method for Calculating Three-Dimensional Water Wave Scattering," Technical Report No. 215, Massachusetts Institute of Technology, Cambridge, Mass.

APPENDIX A: NOTATION

a	Wave amplitude, ft
A	Numerical computational region, dimensionless
A	Wave amplitude, ft
A_I	Incident wave amplitude, ft
B	Numerical computational region, dimensionless
c	Wave phase velocity $(g/k \tanh kh)^{1/2}$, ft/sec
c_g	Wave group velocity $(1/2c(1 + G))$, ft/sec
$F(\phi)$	Mathematical functional, dimensionless
g	Gravitational constant, 32.174, ft/sec^2
G	Computational parameter $(2\ k\ h/\sinh 2\ kh)$, ft/sec
h	Still-water depth, ft
H_n	Hankel function of the first kind, dimensionless
i	$(-1)^{1/2}$, dimensionless
k	Wave number, $2\pi/L$, 1/ft
k_o	Incident wave number, $2\pi/L_o$, ft
K	Symmetric complex coefficient matrix, dimensionless
L	Arbitrary wavelength, ft
L_o	Incident wavelength, ft
n	Unit normal, dimensionless
n_A	Unit normal to the boundary separating computational regions A and B , dimensionless
N	Number of node points in the finite element discretization, dimensionless
r	Radial variables, ft
T	Wave period, sec
$\bar{u}$	Two-dimensional velocity vector, ft/sec
α_n	Constant coefficients in Hankel functions, dimensionless
β_n	Constant coefficients in Hankel functions, dimensionless
Θ	Angular variable in polar coordinates, deg
ϕ	Velocity potential defined by $\bar{u} = \nabla\phi$, ft^2/sec
ϕ_A	Velocity potential in region A , ft^2/sec
ϕ_B	Velocity potential in region B , ft^2/sec
ϕ_I	Incident wave velocity potential, ft^2/sec
ϕ_R	Reflected wave velocity potential, ft^2/sec

ω Angular frequency, $2\pi/T$, 1/sec

∇ Horizontal gradient operator, dimensionless

•U. S GOVERNMENT PRINTING OFFICE: 1984-544-035 19156

END

FILMED

2-85

DTIC