1

MicroD

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents

**BOUNDARY ELEMENT ANALYSIS OF EARTHQUAKE
INDUCED HYDRODYNAMIC PRESSURES
IN A WATER RESERVOIR**

by

ALEXANDER MACIEJ JABŁOŃSKI

B.Sc., M.Eng. - Technical University of Cracow

M.Sc. - University of Illinois

**A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Civil Engineering**

November 1988

**Department of Civil Engineering
Faculty of Engineering
Carleton University
Ottawa, Canada**

Copyright © Alexander Maciej Jablonski, 1988

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-51118-4

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis titled:

**BOUNDARY ELEMENT ANALYSIS OF EARTHQUAKE
INDUCED HYDRODYNAMIC PRESSURES
IN A WATER RESERVOIR**

submitted by ALEXANDER MACIEJ JABŁOŃSKI, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil Engineering.

Thesis Supervisor

Chairman,

Department of Civil Engineering

November 1988

Department of Civil Engineering

Faculty of Engineering

Carleton University

Ottawa, Canada

ABSTRACT

The seismic analysis of concrete gravity and arch dams is affected by the hydrodynamic pressures in the water reservoir. Classical solutions for such pressures can be obtained only for a very idealized geometry; in all other cases a numerical solution must be attempted. The boundary element method (BEM) may prove to be an efficient method for a numerical solution of the problem.

In this work, BEM formulations are derived for the hydrodynamic pressures arising in a gravity dam-reservoir-foundation system. Both 2D and 3D cases are treated. The formulations are based on the respective mathematical models which are governed by two-dimensional or three-dimensional Helmholtz equations with appropriate boundary conditions.

In a BEM formulation, only the boundary of the reservoir needs to be discretized. However, for infinite reservoirs, loss of energy due to pressure waves moving away towards infinity strongly influences response. Since it is not possible to discretize an infinite extent, the radiation damping due to outgoing waves must be accounted for incorporating a special boundary condition at the far end. In a similar manner, the loss of the energy due to absorption of waves by a flexible bottom of the reservoir and banks can be accounted for by a special condition along these boundaries. Both of the special boundary conditions are successfully incorporated in 2D and 3D BEM formulations.

Numerical results are obtained and compared with available classical solutions or results obtained by other researchers. Convergence of numerical results with the size and number of boundary elements is studied. Some consideration is given to the size of an element in relation to the excitation frequency. In the 2D formulation, results obtained from a constant boundary element discretization are compared with those obtained from a linear boundary element discretization. Since no special advantage is noticed in the use of the linear elements as compared to constant elements, only constant elements are used in the 3D formulation.

It is concluded that the direct boundary element method is an effective tool for the evaluation of the hydrodynamic pressures in finite and infinite dam-reservoir-foundation

systems subjected to harmonic types of motion. The method can be easily extended to the analysis of response to any random type of motion with the use of Fast Fourier Transformation technique.

A summary of the conclusions derived as well as recommendations for future research are presented.

ACKNOWLEDGEMENTS

The author wishes to express his deep gratitude and appreciation to his supervisor, Dr. J.L. HUMAR, Professor of Civil Engineering, for his help, encouragement and advice throughout the course of this study.

Thanks are due to Dr. A.P.S. SELVADURAI, Professor and Chairman of the Department of Civil Engineering for his help and to the other members of the Ph.D. Committee for their helpful comments and discussions.

The author wishes to thank all his professors, academic chaplains, friends and fellow-students not only from Carleton University, but also from the 400-year old Bartłomiej Nowodworski Memorial College in Cracow, Poland, the Technical University of Cracow, Poland, the University of Technology in Helsinki, Finland, the University of Illinois at Chicago, USA, and the University of Ottawa, as all of them have contributed to his knowledge.

This research would not have been possible without the scholarships awarded to the author by CARLETON UNIVERSITY and by the Ministry of Colleges and Universities of Ontario in form of the Ontario Graduate Scholarship.

Grateful thanks are also due to the author's family and friends for their support and encouragement during his studies in the New World.

Special thanks are due to my wife, JOANNA and to my son JAN OLAF DOMINIK for their encouragement and patience throughout the time of this study.

CONTENTS

VOLUME 1

Abstract	iii
Acknowledgements	v
Contents	vi
1. Introduction to the Problem	1
1.1 General Comments	1
1.2 Classification of Dam-Reservoir-Foundation Problems	3
1.3 Seismic Aspects of the Design of Dams	4
1.3.1 Gravity Dams	4
1.3.2 Arch Dams	5
1.4 Review of Literature	5
1.4.1 <i>Historical Perspective on the Seismic Analysis of Dam-Reservoir Systems</i>	5
1.4.2 <i>Current Approaches in the Seismic Analysis of Dam-Reservoir Systems</i>	6
1.5 Scope of Present Study and Format of Thesis	9
1.5.1 <i>The Problem</i>	10
1.5.2 <i>Scope of Present Study</i>	11
1.5.3 <i>Format of Thesis</i>	13
2. Formulation of a 2D Model for Gravity Dam-Reservoir Foundation System	15
2.1 General Comments	15
2.2 Mathematical Model of System and Ground Motion	16
2.3 Equations of Motion and Boundary Conditions	17
2.4 Modelling of Foundation Damping	20
2.5 Modelling of Infinite Radiation	23
3. Boundary Element Formulation for 2D Reservoirs	27
3.1 General Comments	27
3.2 Formulation of Boundary Integral Equations in 2D Case	27

3.3	Constant Boundary Element Formulation	29
3.4	Linear Boundary Element Formulation	31
3.5	Fundamental Solution in 2D Case	33
3.6	Evaluation of Boundary Integrals	34
3.6.1	<i>Evaluation of Constant Boundary Integrals</i>	35
3.6.2	<i>Evaluation of Linear Boundary Integrals</i>	36
3.6.3	<i>Treatment of Corner Nodes in Linear Boundary Element Formulation</i>	38
3.7	Incorporation of Infinite Radiation and Foundation Damping Conditions in 2D BEM	42
3.8	Boundary Element Superposition Formulation in 2D Case	44
3.9	Infinite Radiation Condition in 2D BESM	48
4.	Hydrodynamic Pressures in 2D Reservoirs	52
4.1	General Comments	52
4.2	Hydrodynamic Pressure in 2D Finite Reservoirs	52
4.3	Hydrodynamic Pressure in 2D Infinite Reservoirs	61
4.3.1	<i>Effects of Infinite Radiation and Foundation Damping in Rectangular Infinite Reservoirs</i>	61
4.3.2	<i>Comparison of the Results of Constant Boundary Element and Linear Boundary Element Formulations</i>	75
4.3.3	<i>Effects of the Location of the Far Boundary</i>	88
4.3.4	<i>Effects of Inclined Dam Face in Rectangular Infinite Reservoirs</i>	95
4.3.5	<i>Effects of Sloped Bottom in Rectangular Infinite Reservoirs</i>	98
4.4	Convergence of the Numerical Results with the Size and Number of Boundary Elements	102
4.4.1	<i>Size of an Element in Relation to the Excitation Frequency</i>	102
4.4.2	<i>Effect of the Number of Boundary Elements</i>	107
4.4.3	<i>Effect of the Number of Finite Elements in Modelling of Radiation Damping</i>	115
4.5	Comparison of Superposition Method and Boundary Element Method Results in Selected Cases	116
5.	Formulation of the Mathematical Model for a 3D Reservoir Bounded by an Arch Dam	123
5.1	General Comments	123
5.2	Mathematical Model of System and Ground Motion	124
5.3	Equations of Motion and Boundary Conditions	125

5.4	Damping due to Flexibility of the Foundation and the Banks	128
5.5	Modelling of the Infinite Region of the 3D Reservoir	128
6.	Constant Boundary Element Formulation for 3D Reservoirs	134
6.1	General Comments	134
6.2	Formulation of Boundary Integral Equations in 3D Case	135
6.3	Constant Boundary Element Formulation	136
6.4	Fundamental Solution in 3D Case	138
6.5	Evaluation of Constant Boundary Integrals	139
6.6	Incorporation of Infinite Radiation and Foundation and Banks Damping in 3D BEM	143
6.7	Boundary Element Superposition Formulation for Finite 3D Reservoirs	146
7.	Hydrodynamic Pressures in 3D Reservoirs	150
7.1	General Comments	150
7.2	Hydrodynamic Pressure in 3D Finite Reservoirs by BEM	151
7.3	Hydrodynamic Pressure in 3D Infinite Reservoirs by BEM	163
7.3.1	<i>Effects of Infinite Radiation and Foundation Damping in 3D Rectangular Infinite Reservoirs</i>	<i>163</i>
7.3.2	<i>Effects of Infinite Radiation and Foundation in 3D Infinite Reservoirs Impounded by an Arch Dam</i>	<i>179</i>
7.4	Convergence of the Numerical Results with the Size and Number of the Boundary Elements	211
7.4.1	<i>Size of an Element in Relation to the Excitation Frequency</i>	<i>211</i>
7.4.2	<i>Effects of the Number of Finite Elements in Modelling of Radiation Damping</i>	<i>219</i>
7.5	Comparison of Superposition Method and Boundary Element Method Results in Selected Cases	222
8.	Summary, Conclusions and Recommendations for Future Research	225
8.1	Summary	225
8.2	Conclusions	227
8.3	Recommendations for Future Research	230
	References	231
	Appendices	238

A	Notations	238
B	Foundation Damping Condition Derivation	242
C	Development of Boundary Condition for Radiation Damping in 2D Case	247
D	Derivation of the Transformation Matrix T_p for Constant Boundary Elements	253
E	Development of Boundary Condition for Radiation Damping in 3D Case	255

VOLUME 2

1.	User's Guide to Boundary Element Programs	1
1.1	General Comments	1
1.2	Constant Element Program for 2D Reservoirs - BEMC.2D	1
1.2.1	<i>Description of the BEMC.2D Program</i>	<i>1</i>
1.2.2	<i>Input Data</i>	<i>5</i>
1.2.3	<i>Output</i>	<i>6</i>
1.2.4	<i>Example</i>	<i>6</i>
1.2.5	<i>Listing of the BEMC.2D Program</i>	<i>11</i>
1.3	Linear Element Program for 2D Reservoirs - BEML.2D	57
1.3.1	<i>Description of the BEML.2D Program</i>	<i>57</i>
1.3.2	<i>Input Data</i>	<i>59</i>
1.3.3	<i>Output</i>	<i>60</i>
1.3.4	<i>Example</i>	<i>60</i>
1.3.5	<i>Listing of the BEML.2D Program</i>	<i>65</i>
1.4	Constant Element Program for 3D Reservoirs - BEMC.3D	110
1.4.1	<i>Description of the BEMC.3D Program</i>	<i>110</i>
1.4.2	<i>Input Data</i>	<i>113</i>
1.4.3	<i>Output</i>	<i>115</i>
1.4.4	<i>Example</i>	<i>116</i>
1.4.5	<i>Listing of the BEMC.3D Program</i>	<i>124</i>
2.	User's Guide to Superposition Programs	183
2.1	General Comments	183
2.2	Superposition Program for 2D Finite Reservoirs - SUP.2D	183
2.2.1	<i>Description of the SUP.2D Program</i>	<i>183</i>
2.2.2	<i>Input Data</i>	<i>185</i>

2.2.3	Output	185
2.2.4	Example	186
2.2.5	Listing of the SUP.2D Program	190
2.3	Superposition Program for 3D Finite Reservoirs - SUP.3D	198
2.3.1	Description of the SUP.3D Program	198
2.3.2	Input Data	199
2.3.3	Output	199
2.3.4	Example	200
2.3.5	Listing of the SUP.3D Program	206
	References	213

VOLUME 1

CHAPTER 1

INTRODUCTION TO THE PROBLEM

1.1 General Comments

Dams are major engineering structures involving allocation of huge financial and material resources. They range in size and complexity of construction from low earth embankment dams to massive earth or concrete structures. It is possible to classify dams into certain broad categories which include: earth dams, concrete gravity dams and concrete arch dams.

The consequences of the failure of a major dam are of such proportions that dams must be designed and constructed with the utmost care. The design of dams should take into account all the forces imposed by the natural causes and the environment. Among these forces are those due to earthquake motion. The analysis of a dam for earthquake forces is complex. This complexity is further increased by the vibrations of the reservoir impounded by the dam. These vibrations give rise to hydrodynamic pressures in the reservoir which interact with the dam and modify the response of the latter. Additional factors that must be taken into account are: loss of energy in water waves propagating to infinity and interaction between the foundation soil and the dam structure and between the reservoir and the soil. Even when simplifying assumptions and idealizations are made, closed form mathematical solutions are not possible and numerical methods of analysis must be used. During recent years the FEM has been successfully applied to such analytical solution. Boundary element method promises to be another effective numerical technique for the solution of this problem.

This study focuses on the analysis of hydrodynamic pressures caused in a reservoir impounded by a concrete gravity dam or arch dam due to seismic vibrations using boundary

element method. The pressures are affected by the type of motion and the reservoir geometry, that is the shape of the dam face and the bottom. Typical geometries are illustrated by sections of some existing dams shown in Fig. 1.1.

Fig. 1.1 Typical Sections of Existing Gravity and Arch Dams (1,2)

- A. Hoover Dam, Nevada-Arizona, U.S., Height about 725 ft (221 m).**
- B. Grand Coulee Dam, Washington State, U.S., Height 550 ft (168 m).**
- C. Fontana Dam, Tennessee, U.S., Height 450 ft, (137 m).**
- D. Studen Kladenetz Dam, Bulgaria, Height 220 ft, (67.5 m).**
- E. Sakuma Dam, Japan, Height 460 ft, (140 m).**
- F. Koyna Dam, India, Height 338 ft, (103 m).**

1.2 Classification of the Dam-Reservoir-Foundation Problems

The design of large dams is a challenging field in engineering. Dams represent substantial modification to the environment in which they are constructed and are, in turn, affected by the forces of environment. Seismic activity is one of the natural environmental factors to be taken into account. A dam-reservoir-foundation system may be subjected to two types of earthquakes: natural earthquakes in seismic zones and, in some cases, reservoir induced earthquakes. Irrespective of the cause of the earthquake, the dynamic response of the dam is affected by the presence of impounded water.

A statistical study by Haws and Reilly (3) in 1981 reveals that although a substantial number of dams (including concrete gravity dams) have been subjected to earthquakes, relatively few have collapsed completely. Nevertheless, an understanding of the forces imposed on a dam due to earthquake motion is crucial to a rational design.

Improved understanding of the effect of the impounded water on the dynamics of concrete dams is considered essential in the development of reliable procedures for analyses and design of concrete gravity and arch dams (4). In some countries including USA, India and Japan design standards have been developed which take into account the hydrodynamic pressures due to earthquakes (5,6,7,8). The seismic aspects of design of dams contained in the U.S. Standards are summarized in the following section.

In an analytical study of the seismic response of a dam-reservoir-foundation system the following classification of the problem may be useful.

1. Linear Interaction of Dam-Reservoir-Foundation Systems.
 - 1.1 Hydrodynamic pressures on rigid dams.
 - 1.2 Hydrodynamic pressures on flexible dams.
 - 1.3 Dynamic response of dams including foundation damping and reservoir interaction.
 - 1.4 Treatment of infinite boundaries in dam-reservoir-foundation systems.
2. Nonlinear Interaction of Dam-Reservoir-Foundation Systems.
 - 2.1 Effect of surface waves in dam-reservoir-foundation systems.
 - 2.2 Dynamic response of dams due to large vibrations of reservoirs.
 - 2.3 Effect of nonlinear behavior of foundation soil.

3. Local Effects on Interaction of Dam-Reservoir-Foundation Systems.

3.1 Effect of nonlinear behavior of concrete.

3.2 Effect of landslides due to earthquakes.

3.3 Local damage of dam, foundation or embankments.

The above list, although incomplete, gives a general view of the complexity of the analysis of the interaction of a dam-reservoir-foundation system. The problems studied here belongs to the linear class of interaction problems.

1.3 Seismic Aspects of the Design of Dams

A general overview of the seismic aspects of design of dams is presented here in brief. Gravity dams are discussed first. This is followed by a discussion of arch dams. Discussions are focussed on provisions of the U.S. Standards on the design of dams.

1.3.1 Gravity Dams

Traditionally, gravity dams have been designed and analyzed by simple procedures outlined in the old U.S. Standards (10,11,12). New design criteria published in middle seventies by the U.S. Bureau of Reclamation (5,6,9) did not change this early trend. These Standards require that the overturning and sliding stability of the dam monolith be checked together with stresses in the monolith calculated by elementary beam theory. Earthquake forces are treated as static forces and combined with hydrostatic and gravity loads. Neither the interaction between dam and reservoir nor between the foundation and reservoir is recognized. No account is taken of the characteristics of the earthquake motion. Two types of earthquake induced static lateral loads are considered: forces calculated by multiplying the weight of the dam by a constant seismic coefficient ranged from 0.05 to 0.10, forces due to dynamic water pressures, in excess of the hydrostatic pressure. The hydrodynamic pressure is calculated from seismic and pressure coefficients derived on the basis of assumption that the dam is rigid and water incompressible. Thus, traditional criteria focus on satisfying the overturning and sliding stability of the dam and neglect interaction within a dam-reservoir-foundation system.

1.3.2 Arch Dams

Seismic design procedures for arch dams follow those for gravity dams. An earlier standard published by the U.S. Bureau of Reclamation in 1965 (12) disregards completely the 'vibratory response' of a dam and water due to an earthquake motion. The lateral forces due to earthquakes are taken as equal to the product of a seismic coefficient and the weight of the dam. Forces due to water pressures are calculated in terms of the seismic coefficient and the pressure coefficient. The pressure coefficient is based on an assumption that the dam is rigid and straight, and that the water is incompressible. It is required that the design values of compressive and tensile stresses within the dam body fall below maximum allowable values. The interaction effects within dam-reservoir-foundation system are not recognized.

1.4 Review of Literature

A general review of the literature related to the topic of the thesis is presented here. The material is divided into two subsections. In the first subsection a historical perspective of the problem is described. A presentation of the current approaches used in the solution of the problem follows.

1.4.1 Historical Perspective on the Seismic Analysis of Dam-Reservoir Systems

Earliest work on dams subjected to earthquake ground motion concentrated on the analysis of hydrodynamic pressures in reservoirs of simple rectangular geometries. The problem was first studied by Westergaard in his work on hydrodynamic pressures on rigid dams in 1933 (13). He considered a rigid dam with a vertical upstream face resting on a rigid foundation. Water was assumed to be compressible and surface waves were neglected. For excitation frequencies lower than the fundamental frequency of the reservoir, the hydrodynamic pressure was found to be opposite to the horizontal ground acceleration. Its effect was modelled by an added mass of the water moving with the dam, the magnitude of

which was dependent on the ground motion frequency. If the compressibility of water was neglected, the added mass was found to be frequency independent.

In the earlier investigations (14,15,16,17,18,19,20) the dam was treated as rigid. The hydrodynamic pressure acting on vertical or inclined face of the rigid dam due to horizontal or vertical excitations were determined. Although in some studies the compressibility of water was taken into account, the interaction effects were neglected or ignored. Chopra (18) and Kotsubo (16) found that for excitation frequencies higher than the fundamental frequency of the reservoir, the effect of the hydrodynamic pressures would be represented, in addition to the added mass concept, by an equivalent damping due to energy dissipation by waves travelling away from the dam face.

Zienkiewicz and Newton (20) introduced a modelling of an infinite fluid domain by infinite elements in the two dimensional case. Their work was followed by a number of studies that used various types of infinite elements especially in offshore and harbour engineering. Later studies by Chakrabarti and Chopra (21) and also by Sharan (22) accounted for the flexibility of the dam and took interaction effects into account.

In the studies cited in the foregoing paragraph, the dam was usually represented as an assemblage of finite elements and the reservoir was modelled as a continuum with simple boundaries for which classical solutions could be obtained (21,22). In the late seventies Chwang and Housner (23,24) employed analytical techniques to calculate the hydrodynamic pressures on rigid dams with inclined upstream face due to an earthquake motion. In more recent studies irregular boundaries has been considered by modelling the entire dam-reservoir system by finite elements considering displacements as the unknowns for the dam and the pressures as the unknowns for the reservoir (25).

The important conclusions drawn from the studies that considered water to be compressible and dam as flexible were that the earthquake type of motion could cause significant pressures in excess of the hydrostatic pressures, the compressibility of water could not be neglected in analysis, and the deformations of the dam substantially altered the pressures.

1.4.2 Current Approaches in the Seismic Analysis of Dam-Reservoir Systems

In the eighties, the solutions to the problem of seismic response of dam-reservoir systems

were extended to include the foundation interaction and the finite element approach to the solution of the dam-reservoir-foundation interaction became well established due to the work of a group at the University of California at Berkeley (23,26,27,28).

Finite element discretization cannot however be applied in the case of an infinite water reservoir. In such a case, the energy loss in the outgoing waves has been modelled by assuming that beyond a certain length upstream of the dam the reservoir has a uniform rectangular shape. Finite element discretization is limited to the irregular region of the reservoir. For the regular and infinite region a continuum or a one-dimensional finite element solution is employed. Compatibility of pressures and pressure gradients is then enforced at the interface of the regular and irregular regions of the reservoir to complete the solutions. This method was first proposed by Hall and Chopra (26,27).

A second source of energy dissipation in a reservoir is partial absorption of the hydrodynamic pressure waves impinging on the bottom of the reservoir. The flexible foundation soil also modifies the free field ground motion at the base of the dam. These effects can be considered by modelling the foundation material as an elastic continuum or by using finite elements to represent the flexible foundation (22,25).

The fluid-foundation interaction effect has also been accounted for in an approximate manner by using a simplified boundary condition at the interface which models the energy dissipated by partial absorption of the incident pressure waves (26,28). In recent studies, Lotfi, Roesset and Tassoulas (29,30) have suggested that the accuracy of the simplified one dimensional model to represent the wave absorption at the reservoir bottom is uncertain and have suggested its modification. They have also proposed an alternative approach based on a layered foundation stratum assumption and the introduction of a special interface condition between the soil and the fluid within a so-called fluid-soil hyperelement.

Studies carried out by Fenves and Chopra on the previously established simplified boundary condition at the bottom of the reservoir have led to further modifications and simplifications of the finite element approach (31,32,33,34,35,36).

A three-dimensional model for the analysis of an arch dam-reservoir system was first proposed by Hall and Chopra (28). In this FEM approach the reservoir was divided into two parts: finite irregular region and infinite channel of constant cross-section. This approach was a subject of further studies by Porter, Fok and Chopra (37,38,39,40). These studies

included absorption effects and foundation flexibility. A review of the earthquake analysis including design and safety evaluation of concrete dams has recently been presented by Chopra (41).

The reliability of the analytical procedures based on the FEM and developed by Chopra et al. during last twenty years has been established by comparison of analytical results with limited field observation, especially related to Koyna Dam during the Koyna earthquake in a region of India which was considered to be stable and nearly non-seismic (42,43). Analytical results from the FEM analysis were also found to be consistent with the limited number of responses measured during forced vibration tests on Pine Flat Dam near Fresno, California (44).

The considerable computational effort involved in the application of the FEM and the difficulty in modelling infinite domains have given impetus to new studies based on the application of the boundary element method (BEM), to the response analysis of dam-reservoir-foundation system. The BEM method had earlier been applied to many problems in mathematical and engineering physics (45,46,47). Cruse and Rizzo were first to show the immense power of this approach in elastostatics and elastodynamics (48,49).

Recently, the boundary element method has been applied to the analysis of hydrodynamic forces in two-dimensional reservoirs by Hanna and Humar (50,51), Liu and Cheng (52), Liu (78), and Humar and Jablonski (53,54). Wepf and Wolf have also used the BEM in the time domain analysis of the hydrodynamic forces in two-dimensional infinite reservoirs (55). The three dimensional finite arch dam-reservoir vibration problem was solved by Elkhodary (56) and by Humar and Jablonski (57), the latter by the so-called boundary element superposition method (BESM). The three-dimensional boundary element method was also used by Bičkovsky (58) in the treatment of seismic vibrations of a regular, rectangular, finite water reservoir. Humar and Jablonski have recently used the constant boundary element method in the analysis of hydrodynamic forces in a three-dimensional infinite reservoir impounded by an arch dam (59).

The boundary element method has also been applied to the solutions of problems involving magnetic fields (79) and sound radiation (80).

Recently, Hung (81) and Wang have used the finite difference method for a solution of the nonlinear hydrodynamic pressures in a 2D infinite reservoir taking into account the

water dynamic viscosity and the surface wave effect. In an independent research Soong and Murtha have used 8-node quadratic boundary elements for the solution of a 3D rectangular reservoir with an infinite radiation condition, included in a manner similar to that described in the present thesis (82).

1.5 Scope of Present Study

A brief description of the problem investigated in this work is presented here. This is followed by a description of the scope of the study and the format of the thesis.

1.5.1 The Problem

Fig. 1.2 Gravity Dam-Reservoir-Foundation Vibration

Fig. 1.2 shows a typical two-dimensional model of the dam-reservoir-foundation system. The response analysis of this model is a subject of study in this work. The three components of the system are first treated separately and then coupled by taking into account the interaction forces. The water reservoir can have any irregular shape and may extend to

infinity. The foundation may be modelled as rigid when it consists of bedrock or as flexible when it consists of alluvial soil. The complete system is subjected to random excitation due to an earthquake. This work focuses on an analytical solution of the response of a dam-reservoir-foundation system to a horizontal or vertical harmonic excitation. The dam is treated as a rigid in the present work. However, extensions of the methods presented to a flexible dam are straightforward. Also the excitation is assumed to be harmonic rather than irregular as in an earthquake. Extension of the present work to cover the case of irregular motion is easily accomplished by an analysis in the frequency domain.

A three-dimensional arch dam-reservoir-foundation system is represented in Fig. 1.3. It is the second system studied in this work. The arch dam shown in the figure is circular in section, but the method proposed in this study can handle any irregular shape. The dam impounds a reservoir that may be finite or infinite. Response is obtained for horizontal harmonic excitation in two directions as well as for a vertical harmonic excitation.

Other researchers have obtained BEM solutions for both 2D and 3D infinite reservoirs by omitting the far boundary (50,56). Although the procedure may give reasonable estimates of the hydrodynamic pressures, the results are sensitive to the length of the reservoir and convergence is not easily achieved, particularly for excitation frequencies greater than the fundamental frequency of the reservoir. In the present work, radiation damping caused by the outgoing waves in an infinite reservoir is accounted for by incorporating the same special condition at the far end as was first developed for the FEM solution (26). The foundation damping is also accounted for by applying a special simplified condition developed by other researchers (26,27,28). However, in a boundary element analysis, it is straightforward to incorporate other kind of representation for foundation medium, such as for example, an elastic half space or a layered stratum.

1.5.2 Proposed Scope of Study

As mentioned earlier, this study focuses on the use of the boundary element method for the seismic analysis of a dam-reservoir-foundation system. The system itself is considered to be linear. The dam is assumed to be perfectly rigid, extension to the case of a flexible dam is however straightforward. The foundation can either be rigid or flexible. In the latter

Fig. 1.3 Arch Dam-Reservoir-Foundation Vibration

case, a special boundary condition is introduced to account for energy loss due to wave absorption. The water in the reservoir is considered to be non-viscous but compressible. Small vibration theory is applied.

The objective of this study is to demonstrate that the BEM is an effective tool in the seismic analysis of dam-reservoir-foundation systems and may prove to be a more efficient alternative to the FEM particularly for the 3D case. The main goals of research reported here were achieved in several stages. Each of these stages is described below.

1. Analytical formulation of the response of a 2D model of a dam-reservoir-foundation system subjected to a horizontal and/or a vertical harmonic motion using the classical boundary element method (BEM) or its variation, the so-called boundary element superposition method (BESM). This aspect of the work can be further subdivided into:

- 1.1 A constant boundary element formulation for a 2D water reservoir.
 - 1.2 A linear boundary element formulation for a 2D water reservoir.
 - 1.3 An investigation of the effects of infinite radiation using a special boundary condition based on classical solution or one-dimensional finite element discretization of the far region.
 - 1.4 An investigation of the effect of foundation damping.
 - 1.5 An application of the boundary element superposition method to the solution of vibration problem of 2D reservoir subject to harmonic excitation.
2. Analytical formulation of the response of a 3D model of an arch dam-reservoir-foundation system subjected to a horizontal and/or a vertical harmonic motion using classical boundary element method (BEM) or its variation the boundary element superposition method (BESM). This stage of the work can be further subdivided into:
 - 2.1 A constant boundary element formulation for a 3D water reservoir.
 - 2.2 An investigation of the effects of infinite radiation using a special boundary condition for an infinite regular shaped channel.
 - 2.3 An investigation of the effects of damping along the foundation or banks.
 - 2.4 An application of the boundary element superposition method to the solution of the 3D case.
 3. The following problems related to the analytical formulations have also been considered in the thesis:
 - 3.1 Effect of type and size of the boundary element used for only the 2D case.
 - 3.2 Effect of the location of far boundary in the case of infinite water reservoirs.
 - 3.2 The effects of a sloping reservoir bed and an inclined dam face in the 2D case.
 - 3.4 Studies related to the application of the boundary element superposition method, such as for example, the effect of the location of a congruent boundary.

To meet these goals three main computer programs have been developed: (1) for a 2D constant element formulation, (2) for a 2D linear element formulation and (3) for a 3D constant element formulation. Several supporting programs have been developed and

tested. They include, for example, programs for the application of the boundary element superposition method.

1.5.3 Format of Thesis

The thesis constitutes of two volumes: Volume 1 and Volume 2. Volume 1 consists of 8 chapters and 4 appendices. Volume 2 contains details of computer programs as well as user's manuals.

Volume 1

Chapter 1 highlights the importance of seismic analysis and design of dams and introduces the problems of dam-reservoir-foundation interaction involved in the analysis. This chapter also presents a literature survey which consists of review of the historical perspective and more recent approaches in the analysis. Towards the end of Chapter 1, the problem treated in the thesis is briefly described. This is followed by a presentation of the scope of study and the format of the thesis.

In Chapter 2, the formulation of a 2D model to represent a gravity dam-reservoir-foundation system is described. The equations of motion for vibration under the effect of ground motion are derived. It is shown that for harmonic excitation they reduce to a Helmholtz wave equation. Appropriate boundary conditions are formulated. Also, details of modelling to represent damping in foundation and infinite radiation are presented.

Chapter 3 is devoted to the boundary element formulation for 2D water reservoirs. Two different formulations are presented: a constant element formulation and a linear element formulation. Then a detailed description is presented of the infinite radiation condition and foundation damping condition. Chapter 3 also presents details of the boundary element superposition formulation for a 2D reservoir. An infinite radiation condition derived from a classical solution of the infinite region is also described. It is shown that this condition can be effectively incorporated in both the direct BEM and the superposition analysis in the latter case provided foundation damping in the far region need not be accounted for.

In Chapter 4, numerical results obtained from a series of 2D analyses are presented.

Results obtained from each of the two alternative formulations are compared against each other as well as with classical solutions and results obtained by other researchers.

Chapter 5 presents the formulation of a 3D model to represent an arch dam-reservoir-foundation system. The governing equations are derived and it is shown that for a harmonic excitation they reduce to a 3D wave equation. Modelling of damping in foundation and banks as well as modelling of infinite radiation is described.

In Chapter 6, a constant boundary element formulation of the 3D case is presented. Development of boundary conditions to account for damping in flexible foundations and banks and infinite radiation is also presented. In addition, this Chapter contains a description of the boundary element superposition method for the solution of the 3D case. The difficulties involved in the use of this method are also described.

Chapter 7 presents the results of analytical studies of 3D case made with the direct BEM. Numerical results obtained are compared with classical solutions and results obtained by other researchers.

A summary of the work carried out in presented study and the conclusions drawn are contained in Chapter 8. Several recommendations for future research are also presented in that Chapter.

Appendices cover notation used and details of several mathematical derivations not contained in the main body of the thesis.

Volume 2

Volume 2 consists of three sections: user's guide to boundary element programs, user's guide to superposition programs and listing of the programs developed.

CHAPTER 2

FORMULATION OF A 2D MODEL FOR A GRAVITY DAM-RESERVOIR -FOUNDATION SYSTEM

2.1 General Comments

A concrete gravity dam is designed to be stable under its own load when standing freely and subjected to hydrostatic and hydrodynamic pressures due to water. The center of gravity of the dam, is low enough that the stabilizing moment provided by the self weight more than offsets the overturning moment due to water pressure and the lateral dynamic forces of an earthquake. Anchoring to the foundation or abutments is not necessary. The dam cross-section is usually kept constant throughout the length of the dam.

The development of a 2D model of a gravity dam-reservoir-foundation system subjected to ground motion is introduced in this chapter. The assumptions made in the modelling are stated in detail and the governing equations are presented. It is shown that for a harmonic excitation the governing equations reduce to a two-dimensional wave equation or the Helmholtz equation. Boundary conditions applicable to 2D case are presented. Finally, methods of modelling the foundation damping and infinite radiation through special boundary conditions are described. The derivations of these conditions closely parallels that presented by Hall and Chopra (26,28). Reference is made to other options for modelling foundation damping, however, the details are not developed. These options include the method proposed by Abdalla (60), which is based on the development of a foundation impedance matrix for an elastic half plane or soil layer and could be used in the derivation of a more representative foundation boundary condition.

The formulation presented in this chapter is applicable to the analysis of gravity dams. A similar approach has been independently used in work by Wepf and Wolf (55) in their

time-domain dam-reservoir interaction analysis based on the boundary element method.

2.2 Mathematical Model of System and Ground Motion

A concrete gravity dam, the impounded water reservoir and the foundation together constitutes a dam-reservoir-foundation system. The concrete gravity dam is usually modelled as a two-dimensional system in which planar vibrations of the individual monolith of the dam are considered (Fig. 2.1). The interaction between monoliths of the dam is neglected. Each monolith is assumed to be in a plane stress state. The two-dimensional idealization is not capable of providing a solution for response to cross-stream component of ground motion. Analysis of response is thus possible only for a upstream-downstream or a vertical component of the ground motion. The dam is assumed to be linear and elastic.

Fig. 2.1 A 2D Dam-Reservoir-Foundation System Subjected to Upstream-Downstream Ground Harmonic Motion

The reservoir may have a short length or extend to a large distance from the dam face. In the former case, the reservoir is treated as finite, in the latter it is considered infinite for the purpose of analysis.

The base of the dam and the bottom of the reservoir together called the foundation may undergo a prescribed acceleration history due to the vertical and/or horizontal components of ground motion, in x and/or y direction respectively. Because in this work analysis is restricted to the case of a rigid dam, dam-foundation interaction is not considered. Reservoir-foundation interaction is, however, accounted for through a special boundary condition. The foundation can be rigid or flexible. In the case of a rigid foundation, there is no interaction between the dam and the foundation or the water reservoir and the foundation. A flexible foundation will interact both with the dam and the reservoir.

2.3 Equations of Motion and Boundary Conditions

The following assumptions are made in the formulation of the equations of motion of the water in the reservoir:

1. Water is compressible and non-viscous.
2. The effect of free surface waves is negligible. The associated errors were previously reported to be small (28).
3. The motion of water particles is limited to small amplitudes.

With the above assumptions, the following set of Navier-Stokes equations govern the motion of water:

$$\frac{\partial^2 u}{\partial t^2} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x} \quad (2.1a)$$

$$\frac{\partial^2 v}{\partial t^2} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial y} \quad (2.1b)$$

$$\frac{\partial^2 \bar{p}}{\partial x^2} + \frac{\partial^2 \bar{p}}{\partial y^2} = \frac{1}{c^2} \frac{\partial^2 \bar{p}}{\partial t^2} \quad (2.1c)$$

in which, x and y = the cartesian coordinates as shown in Fig. 2.1; u and v = respectively

the x and y displacements of a particle of water; t = time variable; ρ = mass of water per unit volume; $\bar{p}$ = the hydrodynamic pressure; and c = the velocity of sound in water.

For a harmonic motion of the dam or the foundation, the hydrodynamic pressure can be represented by $\bar{p} = p(x, y, \omega) e^{i\omega t}$ where ω is the exciting frequency. Substitution into Eq. 2.1c then gives:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + k^2 p = 0 \quad (2.2)$$

where $k = \omega/c$ is the wave number.

Equation 2.2 is known as a two-dimensional Helmholtz equation. It is a second order partial differential equation belonging to the elliptical class of equations with constant coefficients. Solutions to this equation have been reported in the literature for simple geometries (61,62,63).

The following boundary conditions are specified in the 2D case:

1. Along surface of the reservoir

Since the effect of surface waves is neglected, the hydrodynamic pressure should vanish at the free surface:

$$\bar{p}(x, H, t) = 0 \quad (2.3)$$

which gives

$$p(x, H) = 0 \quad (2.4)$$

2. Along the dam face

The following condition is applied along the dam face. Assuming that the dam itself is treated as rigid.

$$\frac{\partial \bar{p}}{\partial n}(s, t) = -\frac{w}{g} a_n(s, t) e^{i\omega t} \quad (2.5)$$

in which w is the specific gravity of water, s is the coordinate along the dam face, and a_n is the component of harmonic exciting acceleration amplitude along the outward normal to the reservoir at the dam-reservoir interface. If the dam is treated as rigid, so that its acceleration is the same as that of the foundation on which it rests, and if the horizontal acceleration of ground is $e^{i\omega t}$,

$$a_n = -\cos \alpha \quad (2.6)$$

in which α is the angle the dam face makes with the vertical. Boundary condition given by Eq. 2.5 can thus be expressed as:

$$q = \frac{\partial p}{\partial n}(s, \omega) = \frac{w}{g} \cos \alpha \quad (2.7)$$

Similarly, for a vertical harmonic excitation, the following boundary condition is applicable.

$$q = \frac{\partial p}{\partial n}(s, \omega) = \frac{w}{g} \sin \alpha \quad (2.8)$$

3. Along the reservoir bottom

For a rigid type of foundation, as for example a bedrock, there is no interaction between the water and the foundation, and the boundary condition along the reservoir bottom is

$$\frac{\partial \bar{p}}{\partial n}(s', t) = -\frac{w}{g} a_n(s', t) \quad (2.9)$$

in which s' is the coordinate along the reservoir bottom, and $a_n(s')$ is the excitation amplitude along the outward normal.

For a flexible foundation, the boundary condition is adapted to include, in an appropriate manner, the effect of wave absorption. A special boundary condition developed originally by Hall and Chopra (26,28) is used here. A detailed description of this condition is presented in the next section.

4. Along the far boundary

For a finite reservoir, the condition at the far end depends on whether that boundary is fixed or free to move. An infinite reservoir is modelled as being comprised of two sections, an irregular section of finite length and an infinite section with a uniform rectangular section. The far boundary is located at the end of the irregular region of the reservoir. Beyond this boundary, the reservoir has a regular rectangular shape extending to infinity. The energy loss due to radiation damping in the infinite region is accounted for by imposing a special boundary condition. The derivation of this special condition closely parallels that presented by Hall and Chopra (28) and is presented in the one of subsequent sections. Some studies on a similar type of the non-reflecting condition have been recently presented by Sharan (64) and also by Wepf and Wolf (55). However neither of these two latter formulations covers a treatment of the foundation damping in the infinite section of the reservoir.

2.4 Modelling of Foundation Damping

In the case where a flexible foundation provides a damping mechanism, the special boundary condition is adapted to include, in an approximate manner, the effect of wave absorption. The derivation of this condition is based on a consideration of a one-dimensional fluid-foundation system and an assumption that energy is dissipated by a partial absorption of the incident compression waves travelling only in the direction normal to the reservoir-foundation interface. The formulation is then assumed to apply to a two-dimensional fluid-foundation system.

The resulting condition has the following form (26,28):

$$q = \frac{\partial p}{\partial n}(s', \omega) = -\frac{w}{g} a_n(s') - i\omega\gamma p(s', \omega) \quad (2.10)$$

in which s' = the coordinate along the reservoir bottom, $\gamma = w/(w_r c_r)$ = the foundation damping coefficient, w_r = specific gravity of the foundation material, c_r = compression wave velocity in the foundation material and $a_n(s')$ = the harmonic excitation amplitude along the outward normal. For a horizontal excitation

$$a_n(s') = \sin \phi \quad (2.11)$$

while for a vertical excitation

$$a_n(s') = -\cos \phi \quad (2.12)$$

in which ϕ is the angle that the reservoir bottom makes with the horizontal.

The second term on the right hand side of Eq. 2.10 represents the effect of foundation damping. For rigid foundation, $\gamma = 0$ and the damping is absent. Foundation damping can also be defined in terms of a wave reflection coefficient α_r which is related to γ as follows:

$$\gamma = \frac{1}{c} \frac{1 - \alpha_r}{1 + \alpha_r} \quad (2.13)$$

in which c = velocity of sound in water.

Details of the derivation of special condition represented by Eq. 2.10 and the relationship between γ and α_r are presented in Appendix B. However, a brief discussion of the underlying principle is given below.

As shown in Fig. 2.2, the foundation can be considered as a system of thin columns of infinite lengths, which extend in the direction normal to the reservoir-foundation interface. The foundation motion is modelled by the axially travelling compression waves in the columns. It is assumed that each of these columns vibrate independently and that there is no interaction between columns in the transverse direction. Compatibility is maintained between normal displacement and stress across the reservoir-foundation interface. The damping coefficient for the foundation stratum depends on the physical properties of these columns: E_r , w_r , c_r .

The foundation model of Fig. 2.2 provides an idealization of the real foundation. It has been successfully used in the past by the other researchers in studies of reservoir-foundation systems in which a FEM technique is used in the analysis (26,28).

Recently, the resulting aforementioned condition has been criticized by Lotfi, Roeset and Tassoulas on the ground that it tends to overestimate the damping provided by the flexible foundation (30). Lotfi has derived a modified version of this condition and also has

Fig. 2.2 An Idealization of the Foundation in the 2D System Model

condition based on a layered foundation stratum assumption (29). Lotfi's special interface condition between the soil and the fluid is introduced within so-called fluid-solid hyperelement leading to the solution of an eigenvalue problem associated with this condition.

The effect of foundation damping can also be accounted for through a more detailed representation of the foundation soil. This is achieved by deriving an impedance matrix for the soil medium using a boundary element technique. Details of development of the impedance matrix have been provided by Abdalla (60). The impedance matrix is in fact, a so-called dynamic stiffness matrix for an elastic half plane or a soil layer and could be used in the derivation of a more representative foundation boundary condition.

The simplified boundary condition (Eq. 2.10) is used to represent foundation damping in this work. However any other condition can be accommodated in the BEM formulation, but it would require further studies.

2.5 Modelling of Infinite Radiation

The infinite section of the reservoir is shown in Fig. 2.3. The radiation boundary condition is derived using a one-dimensional finite element discretization of the infinite region (26,28). The details of the derivation are given in the references cited. For the sake of completeness, they are presented in Appendix C and summarized here.

Fig. 2.3 An Infinite but Regular Section of a 2D Reservoir

- horizontal excitation

When the excitation is only a horizontal ground motion, the hydrodynamic pressure in the infinite section is governed by Eq. 2.2 with the following boundary conditions:

$$\begin{aligned}
 \frac{\partial p}{\partial x}(0, y, \omega) &= q_f \\
 \frac{\partial p}{\partial y}(x, 0, \omega) &= i\omega\gamma p(x, 0, \omega) \\
 p(x, H, \omega) &= 0
 \end{aligned}
 \tag{2.14}$$

in which q_f is the pressure derivative $\frac{\partial p}{\partial n}$ in the finite region at the interface of the finite and infinite regions of the reservoir.

The governing Equation 2.2 can be solved by a separation of variables

$$p = p_x p_y \quad (2.15)$$

in which p_x satisfies the equation

$$\frac{d^2 p_x}{dx^2} - \kappa^2 p_x = 0 \quad (2.16)$$

and p_y satisfies the equation

$$\frac{d^2 p_y}{dy^2} + \lambda^2 p_y = 0 \quad (2.17)$$

with the boundary conditions

$$\begin{aligned} p_y &= 0 & \text{at } y &= H \\ \frac{dp_y}{dy} &= i\omega\gamma p_y & \text{at } y &= 0 \end{aligned} \quad (2.18)$$

In Eqs. 2.16 and 2.17, κ = the separation constant and λ is given by

$$\lambda^2 = k^2 + \kappa^2 \quad (2.19)$$

A finite element solution of Eq. 2.17 gives

$$[\mathbf{A} + i\omega\gamma \mathbf{B}] \mathbf{p} = \lambda^2 \mathbf{F} \mathbf{p} \quad (2.20)$$

where $\mathbf{A}$ and $\mathbf{F}$ are assembled matrices whose individual components are given in Appendix C and $\mathbf{B}$ is an $M \times M$ matrix whose only non-zero term is the last element on the diagonal with value equal to $p_y(0)$. As stated earlier, $\mathbf{p}$ is the vector of the pressure values at the nodes. The surface node is excluded from the formulation because the pressure value is known to be zero there.

Equation 2.20 represents an eigenvalue problem which for non-zero γ leads to complex eigenvalues and eigenvectors. If the matrix of M eigenvectors obtained from Eq. 2.14 is represented by Ψ , the following relationship is satisfied:

$$\Psi^T [A + i\omega\gamma B] \Psi = A \quad (2.21)$$

in which A = a diagonal matrix of the M eigenvalues $\lambda_1^2, \lambda_2^2, \dots, \lambda_M^2$ (real or complex), and the eigenvectors have been normalized to be orthonormal w.r.t F so that

$$\Psi^T F \Psi = I \quad (2.22)$$

The solution to Eq. 2.16 can now be written as

$$p_z = e^{-\kappa_n z} \quad (2.23)$$

in which $\kappa_n = \sqrt{\lambda_n^2 - k^2}$.

When κ is real valued, it is taken with a positive sign to ensure that p_z tends to zero as $z \rightarrow \infty$. When κ is imaginary, both of its parts (real and imaginary) should be positive. A positive real part will imply that the wave amplitude decays with increasing z . A positive imaginary part will account for the fact the waves move away towards infinity and no reflected waves are present.

The general solution for pressure vector p has a form

$$p = \sum_{n=1}^M \eta_n \psi_n e^{-\kappa_n z} \quad (2.24)$$

or

$$p = \Psi E \eta \quad (2.25)$$

in which η = a vector of modal coordinates and E = is a diagonal matrix of terms $e^{-\kappa_n z}$. Taking the origin at the far boundary and differentiating Eq. 2.25

$$q = -\Psi K \eta \quad (2.26)$$

in which K is a diagonal matrix with elements $\kappa_1, \kappa_2, \dots, \kappa_M$.

- vertical excitation

When the excitation is a vertical ground motion, the hydrodynamic pressure is still governed by Eq. 2.2 with the boundary conditions

$$\begin{aligned}
\frac{\partial p}{\partial x}(0, y, \omega) &= 0 \\
\frac{\partial p}{\partial y}(x, 0, \omega) &= -\frac{w}{g} a_y + i\omega\gamma p(x, 0, \omega) \\
p(x, H, \omega) &= 0
\end{aligned} \tag{2.27}$$

The problem is one dimensional in y coordinate and x variation is absent so the governing equation becomes

$$\frac{d^2 p}{dy^2} + k^2 p = 0 \tag{2.28}$$

A numerical solution of governing equation with the finite element discretization gives

$$[A + i\omega\gamma B - k^2 F] p = \frac{w}{g} d \tag{2.29}$$

in which A , B , F are the same matrices as described in Eq. 2.20 and d is a vector with one non-zero term equal to an acceleration a_y corresponding to the base node b . Equation 2.29 can be solved for p by using an eigenvector expansion in which the complex valued eigenvalues λ and eigenvectors Ψ are determined by solving Eq. 2.20. The vector of nodal pressures at the far end is given by

$$p = \frac{w}{g} \Psi K^{-1} \Psi^T d \tag{2.30}$$

Combining Eqs. 2.25 and 2.30, the total pressure at the finite element nodes on the far boundary is obtained as

$$p = \Psi \eta + \frac{w}{g} \Psi K^{-1} \Psi^T d \tag{2.31}$$

An incorporation of the above derived condition in the two-dimensional boundary element formulation is presented in the next chapter. As an alternative to the finite element solution a classical solution can be used to derive the far boundary condition. A detailed description of the classical solution for an infinite rectangular in shape water domain subjected to small vibration motion is presented in reference 28.

CHAPTER 3

BOUNDARY ELEMENT FORMULATION FOR 2D RESERVOIRS

3.1 General Comments

The application of the direct boundary element method (BEM) to the analysis of small vibrations of a 2D reservoir is described in this Chapter. Two alternative boundary element formulations are presented: one using constant boundary elements, and the other using linear boundary elements. Evaluation of respective boundary integrals is described in detail. This is followed by a presentation of the infinite radiation condition and the simplified damping condition which when incorporated in the BEM formulation accounts for energy loss due respectively to waves moving away to infinity and wave absorption at the flexible reservoir bottom. Towards the end of this chapter, the boundary element superposition procedure for the 2D case is introduced and, the special form of infinite radiation condition applicable to 2D BESM is presented. Finally, the difficulties encountered in the application of the BESM are summarized.

3.2 Formulation of Boundary Integral Equations in 2D Case

The motion of the water in a 2D reservoir is governed by the two-dimensional Helmholtz equation also known as the reduced wave equation (see Eq. 2.2).

For a numerical solution of the problem represented by Eq. 2.2 a weighted residual technique is applied in which the governing equation is satisfied in an average sense over the domain. This leads to the following equation:

$$\iint_A \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + k^2 p \right) p^* dx dy = 0 \quad (3.1)$$

in which p^* is an assumed weighting function and A is the water domain under consideration.

Repeated partial integration of Eq. 3.1 and an application of the Green's theorem in 2D space gives:

$$\iint_A \left(\frac{\partial^2 p^*}{\partial x^2} + \frac{\partial^2 p^*}{\partial y^2} + k^2 p^* \right) p dx dy = \int_S p q^* ds - \int_S p^* q ds \quad (3.2)$$

in which $q^* = \frac{\partial p^*}{\partial n}$, $q = \frac{\partial p}{\partial n}$, S = boundary of the water reservoir, and n = the outward normal to the boundary.

The essence of the boundary integral equation method lies in an appropriate choice of the weighting function. The chosen function permits elimination of the area integral on the L.H.S. of Eq. 3.2. The desired function has to satisfy the following equation:

$$\frac{\partial^2 p_i^*}{\partial x^2} + \frac{\partial^2 p_i^*}{\partial y^2} + k^2 p_i^* = \Delta^i \quad (3.3)$$

in which Δ^i = a Dirac Delta function centered at point i having coordinates (x_i, y_i) .

Substitution of Eq. 3.3 into Eq. 3.2 gives

$$c_i p_i = \int_S p q_i^* ds - \int_S p_i^* q ds \quad (3.4)$$

in which $c_i = 1$ for a point inside the water reservoir, $c_i = \frac{1}{2}$ for a point on the smooth boundary of the water reservoir, $c_i = 0$ for a point outside the water reservoir, and p_i = the hydrodynamic pressure at point i .

When i is located at a corner on the boundary, as shown in Fig. 3.2, the value of c_i is given by (65)

$$c_i = 1 - \frac{\beta}{2\pi} \quad (3.5)$$

where β is the external angle at the corner.

Equation 3.4 is known as the boundary integral equation associated with the given Helmholtz equation. It provides a means of determining the unknown pressure at any point within a two-dimensional water reservoir in terms of the integrals of pressure and pressure gradient values evaluated on the reservoir boundary only. While some of these boundary values may be specified, other must be determined before Eq. 3.4 can be effectively used. In the boundary element formulation, Eq. 3.4 itself is used to determine the unknowns on the boundary. To achieve this, the boundary of the water domain is divided into a series of segments (elements) and the parameters p and q are assumed to vary in an assumed, prescribed manner over a given element. The manner in which this variation takes place determines the details of formulation. Two such formulation will be described here, namely: the constant boundary element formulation and the linear boundary element formulation.

3.3 Constant Boundary Element Formulation

The constant boundary element formulation for the two-dimensional Helmholtz equation has been described by other researchers. However, the main points of this formulation are summarized here for the sake of completeness.

In the constant boundary element formulation, the segments (elements) are straight lines, and p and q are assumed to be constant and equal to their values at the midpoint of the segment (element). This midpoint is referred to as a node. A constant boundary element model of the two-dimensional water reservoir is presented in Fig. 3.1.

Assuming that the boundary of the water reservoir is divided into N straight line boundary elements (Fig. 3.1), Eq. 3.4 takes for a given point i on the boundary the following discretized form:

$$-\frac{1}{2} p_i + \sum_{j=1}^N p_j \int_{s_j} q_i^* ds_j = \sum_{j=1}^N q_j \int_{s_j} p_i^* ds_j \quad (3.6)$$

in which p_j and q_j = the values of these variables over the j th element; and s_j = the boundary of that element.

Introducing the following notations:

Fig. 3.1 Constant Boundary Element Discretization of a 2D Reservoir

$$H_{ij} = \int_{s_j} q_i^* ds_j \quad i \neq j \quad (3.7a)$$

$$H_{ii} = -\frac{1}{2} + \int_{s_i} q_i^* ds_i \quad (3.7b)$$

$$G_{ij} = \int_{s_j} p_i^* ds_j \quad (3.7c)$$

Equation 3.6 can be written as

$$\sum_{j=1}^N p_j H_{ij} = \sum_{j=1}^N q_j G_{ij} \quad (3.8)$$

$$i = 1, 2, \dots, N$$

Corresponding to N values of i , N equations of the type of Eq. 3.8 can be written.

Equation 3.6 or 3.8 together with Eqs. 3.7 constitutes the constant boundary element formulation for the two-dimensional Helmholtz equation. Details of the evaluation of the integrals involved in these equations will be presented in one of the following sections.

3.4 Linear Boundary Element Formulation

In the linear boundary element formulation, the segments (elements) are again straight lines similar to those in the constant boundary element case, but parameters p and q are assumed to vary linearly over a segment and are functions of their values at the end points of the segment (element). The end points of each element are called the nodes. Equation 3.4 can be rewritten for a point i located at a node on the discretized boundary, and takes the form:

$$-\left(1 - \frac{\beta}{2\pi}\right) p_i + \sum_{j=1}^N \int_{s_j} p q_i^* ds_j = \sum_{j=1}^N \int_{s_j} p_i^* q ds_j \quad (3.9)$$

in which s_j is the boundary of the j th element.

A linear boundary element discretization model is shown in Fig. 3.2 and the j th element of the model is shown in Fig. 3.3.

Fig. 3.2 Linear Boundary Element Discretization of a 2D Reservoir

Referring to Fig. 3.3, the variation of p and q over an element can be expressed

$$\begin{aligned} p &= [\phi_1 \quad \phi_2] \begin{Bmatrix} p_j \\ p_{j+1} \end{Bmatrix} \\ q &= [\phi_1 \quad \phi_2] \begin{Bmatrix} q_j \\ q_{j+1} \end{Bmatrix} \end{aligned} \quad (3.10)$$

in which ϕ_1 and ϕ_2 are the shape functions given by:

$$\begin{aligned} \phi_1 &= 1/2 (1 - \xi) \\ \phi_2 &= 1/2 (1 + \xi) \end{aligned} \quad (3.11)$$

and ξ is a coordinate along the boundary element.

Substitution in Eq. 3.9 for p and q from Eq. 3.10 gives

$$\begin{aligned} - \left(1 - \frac{\beta}{2\pi}\right) p_i + \sum_{j=1}^N \int_{s_j} [\phi_1 \quad \phi_2] \begin{Bmatrix} p_j \\ p_{j+1} \end{Bmatrix} q_i^* ds_j \\ = \sum_{j=1}^N \int_{s_j} [\phi_1 \quad \phi_2] \begin{Bmatrix} q_j \\ q_{j+1} \end{Bmatrix} p_i^* ds_j \end{aligned} \quad (3.12)$$

$i = 1, 2, \dots, N$

Introducing the following notations:

$$\int_{s_j} \phi_1 q_i^* ds_j = h_{ij}^{(1)} \quad (3.13a)$$

$$\int_{s_j} \phi_2 q_i^* ds_j = h_{ij}^{(2)} \quad (3.13b)$$

$$\int_{s_j} \phi_1 p_i^* ds_j = g_{ij}^{(1)} \quad (3.13c)$$

$$\int_{s_j} \phi_2 p_i^* ds_j = g_{ij}^{(2)} \quad (3.13d)$$

Equation 3.12 can be written in the following compact form:

$$\begin{aligned} - \left(1 - \frac{\beta}{2\pi}\right) p_i + \sum_{j=1}^N (h_{ij}^{(1)} p_j + h_{ij}^{(2)} p_{j+1}) \\ = \sum_{j=1}^N (g_{ij}^{(1)} q_j + g_{ij}^{(2)} q_{j+1}) \end{aligned} \quad (3.14)$$

$i = 1, 2, \dots, N$

Assembly of the N equations represented by Eqs. 3.8 for the constant element formulation or by Eqs. 3.14 for the linear element formulation gives, in the matrix form:

$$\mathbf{H} \mathbf{p} = \mathbf{G} \mathbf{q} \quad (3.15)$$

in which $\mathbf{H}$ and $\mathbf{G}$ are each matrices of size $N \times N$ whose elements H_{ij} and G_{ij} , respectively are given by Eq. 3.7 for the constant element formulation and by the following set of expressions for the linear element formulation:

$$H_{ij} = h_{ij}^{(1)} + h_{(i,j-1)}^{(2)} \quad i \neq j \quad (3.16a)$$

$$H_{ii} = -\left(1 - \frac{\beta}{2\pi}\right) + h_{ii}^{(1)} + h_{(i,i-1)}^{(2)} \quad (3.16b)$$

$$G_{ij} = g_{ij}^{(1)} + g_{(i,j-1)}^{(2)} \quad (3.16c)$$

3.5 Fundamental Solution in 2D Case

The solution of Eq. 3.3 is commonly referred to as the fundamental solution and it is obtained by the so-called Green's function approach (62,66,67).

$$p^* = \frac{1}{4} Y_0(kr) \quad (3.17)$$

in which Y_0 = Bessel function of zero order and second kind; r = the radial distance variable given by $r = \sqrt{(x - x_i)^2 + (y - y_i)^2}$, where the point with coordinates (x_i, y_i) is referred to as the source point.

The fundamental solution given by Eq. 3.17, has been used by several researchers to study problems concerning wave propagation in bounded domains (50,62,68). For infinite domains, the solution $Y_0(kr)$ which includes waves reflected from infinity is unacceptable from a physical point of view and a solution $p^* = \frac{i}{4} H_0^{(2)}(kr)$ must be used, where $H_0^{(2)}$ is the Hankel function of zero order and second kind (51,66). In the present study, however, an infinite reservoir is represented by a finite part and an infinite part, in which the infinite part is modelled as a continuum or by one-dimensional finite elements. Boundary element

discretization is restricted to the finite part of the reservoir and the $Y_0(kr)$ solution is seen to be appropriate in this case.

3.6 Evaluation of Boundary Integrals

The boundary integrals in Eqs. 3.7 and 3.13 are calculated by substituting for p^* and q^* from the fundamental solution given by Eq. 3.17. Details of the evaluation of these integrals are presented in the following paragraphs.

3.6.1 Evaluation of Constant Boundary Integrals

The evaluation of constant element boundary integrals has been fully described by Hanna and Humar (50,51). Some details are, however, presented here for the sake of completeness.

The term H_{ij} in Eq. 3.7a is given by

$$\begin{aligned} H_{ij} &= \int_{s_j} q_i^* ds_j \\ &= \frac{k}{4} \int_{s_j} Y_0'(kr) \cos \theta ds_j \end{aligned} \quad (3.18)$$

in which $Y_0'(kr)$ = the first derivative of $Y_0(kr)$ w.r.t. kr , and θ = the angle between the outward normal n , and the radius vector r , as shown in Fig. 3.1.

The integral in Eq. 3.18 can now be evaluated by any numerical method, for example, a four-point Gauss quadrature formula, giving:

$$H_{ij} = \frac{kl_j}{8} \sum_{m=1}^4 Y_0'(kr)_m (\cos \theta)_m w_m \quad (3.19)$$

w_m = the weighting function used in the Gauss quadrature, and l_j = the length of the j th element.

For the special case (Eq. 3.7b) of $i = j$

$$H_{ii} = \left\{ \frac{kl_j}{8} \sum_{m=1}^4 Y_0'(kr)_m (\cos \theta)_m w_m \right\} - \frac{1}{2} \quad (3.20)$$

The first of the two terms in the expression for H_{ii} vanishes because the radius vector r is orthogonal to the outward normal n so that $\cos \theta = 0$. Equation 3.20 therefore reduces to

$$H_{ii} = -\frac{1}{2} \quad (3.21)$$

The term G_{ij} is also obtained by numerical integration and is given by

$$G_{ij} = \frac{l_j}{8} \sum_{m=1}^4 Y_0(kr)_m w_m \quad (3.22)$$

A special procedure must be used in the evaluation of G_{ii} because of the presence of a singularity. In this particular case, the term G_{ii} is calculated analytically by using the following approximation of the fundamental solution.

$$Y_0(kr) \simeq \frac{2}{\pi} \ln(kr) \quad r \rightarrow 0 \quad (3.23)$$

Substituting this value in the expression for G_{ii} and referring to Fig. 3.3

$$\begin{aligned} G_{ii} &= \frac{1}{2\pi} \int_{s_i} \ln(kr) ds_i \\ &= \frac{l_i}{2\pi} \int_0^{+1} \ln\left(k\xi \frac{l}{2}\right) d\xi \\ &= \frac{l_i}{2\pi} \left\{ \ln\left(\frac{kl_i}{2}\right) - 1 \right\} \end{aligned} \quad (3.24)$$

Equation 3.24 will give fairly accurate values when $\frac{kl}{2}$ is small. The accuracy of G_{ii} can be improved by numerically integrating the residual $\frac{1}{4} \left\{ Y_0(kr) - \frac{2}{\pi} \ln(kr) \right\}$, say by the Gauss quadrature method.

For $\frac{kl}{2} = 0.18$, which corresponds to an exciting frequency of 53.3 rad/s, a compression wave velocity 1440 m/s and an element length of 10 m, the integration of the residual yields a value that is less than 4% of the total value of G_{ii} .

Fig. 3.3 Coordinate System Used in the Evaluation of Constant Boundary Integrals

3.6.2 Evaluation of Linear Boundary Integrals

The evaluation of linear boundary integrals is presented in this subsection. The integrals represented by Eqs. 3.13 are calculated by substituting the appropriate expressions for shape functions ϕ_1 and ϕ_2 and using the fundamental solution given by Eq. 3.17.

$$\begin{aligned}
 h_{ij}^{(1)} &= \int_{s_j} \phi_1 q_i^* ds_j \\
 &= \frac{kl_j}{16} \int_{-1}^{+1} (1 - \xi) Y_0'(kr) \cos \theta d\xi
 \end{aligned} \tag{3.25}$$

The notations used in Eq. 3.25 are represented in Fig. 3.4.

Application of the Gauss 4-point quadrature formula for the integration of Eq. 3.25 gives

$$h_{ij}^{(1)} = \frac{kl_j}{16} \sum_{m=1}^4 (1 - \xi_m) Y_0'(kr)_m (\cos \theta)_m w_m \tag{3.26a}$$

Fig. 3.4 Coordinate System Used in the Evaluation of Linear Boundary Integrals

In a similar manner

$$h_{ij}^{(2)} = \frac{kl_j}{16} \sum_{m=1}^4 (1 + \xi_m) Y_0'(kr)_m (\cos \theta)_m w_m \quad (3.26b)$$

$$g_{ij}^{(1)} = \frac{l_j}{16} \sum_{m=1}^4 (1 - \xi_m) Y_0(kr)_m w_m \quad (3.26c)$$

$$g_{ij}^{(2)} = \frac{l_j}{16} \sum_{m=1}^4 (1 + \xi_m) Y_0(kr)_m w_m \quad (3.26d)$$

For the special case of H_{ii} , the terms $h_{ii}^{(1)}$ and $h_{i,i-1}^{(2)}$ vanish because of the orthogonality of r and n so that $\cos \theta = 0$. This gives the following reduced expression for H_{ii} :

$$H_{ii} = -1 + \frac{\beta}{2\pi} \quad (3.27)$$

In the evaluation of G_{ii} , singular kernels arise both when $j = i$ and $j = i + 1$. Integrals containing singular kernels are evaluated analytically using the approximation for $Y_0(kr)$ given by Eq. 3.23. Term G_{ii} is first expressed as

$$G_{ii} = g_{ii}^{(1)} + g_{i,i-1}^{(2)} \quad (3.28)$$

Then referring to Fig. 3.4,

$$g_{ii}^{(1)} = \frac{1}{2\pi} \int_0^{l_i} \left(1 - \frac{x}{l_i}\right) \ln(kx) dx \quad (3.29a)$$

$$= \frac{l_i}{4\pi} (\ln kl_i - 1.5) \quad (3.29b)$$

In a similar manner:

$$g_{i,i-1}^{(2)} = \frac{l_{i-1}}{4\pi} (\ln kl_{i-1} - 1.5) \quad (3.30)$$

Also,

$$G_{i,i+1} = g_{i,i+1}^{(1)} + g_{ii}^{(2)} \quad (3.31)$$

in which term $g_{ii}^{(2)}$ containing a singular integrand is evaluated as follows:

$$\begin{aligned} g_{ii}^{(2)} &= \frac{1}{2\pi} \int_0^{l_i} x \ln(kx) dx \\ &= \frac{l_i}{4\pi} (\ln kl_i - 0.5) \end{aligned} \quad (3.32)$$

Again, the accuracy of integrals containing singular kernels can be improved by numerically integrating the residual $\frac{1}{4} \{Y_0(kr) - \frac{2}{\pi} \ln(kr)\}$ using Gauss quadrature.

3.6.3 Treatment of Corner Nodes in Linear Boundary Element Formulation

In the direct boundary element method described in the preceding paragraphs, two parameters are associate with each node: a pressure and pressure derivative. One of these parameters is known while other needs to be determined by solving the boundary element equations.

In the linear boundary element formulation, difficulties sometime arise at a corner node. Referring to Fig. 3.5 it would be seen that at a corner node n , there is only one pressure value but two pressure gradients corresponding to the two boundary normals. Assuming

that the nodes are numbered counterclockwise direction these gradients may be referred to as $q_n^{(-)}$ and $q_n^{(+)}$.

If the pressure at node n is the specified quantity then it would appear that these are two unknowns at the node. As a result for each node of the type of n these would be one too few equations in the boundary element formulation. One possible means of avoiding this difficulty is to define two nodes very close to each other and on either side of the corner as shown in Fig. 3.5 (69). This will add one more equation to the boundary formulation to match the one extra unknown.

Fig. 3.5 A Corner Node in the Linear Boundary Element Formulation

In the case of a 2D reservoir-dam system a corner node arrives at the junction of dam face and top surface of the reservoir. However, in this case $q_n^{(+)}$ is given, $p_n = 0$ and $q_n^{(-)}$ is thus the only unknown. An extra node is not therefore required but the boundary equation formulation should be slightly revised to account for two gradients $q_n^{(-)}$ and $q_n^{(+)}$.

The i th equation given by Eq. 3.14 will get modified to

$$\begin{aligned}
& - \left(1 - \frac{\beta}{2\pi}\right) p_i + \sum_{j=1}^N \left(h_{ij}^{(1)} p_j + h_{ij}^{(2)} p_{j+1} \right) \\
& = \sum_{\substack{j=1 \\ j \neq n-1 \\ j \neq n}}^N \left(g_{ij}^{(1)} q_j + g_{ij}^{(2)} q_{j+1} \right) \\
& + g_{i,n-1}^{(1)} q_{n-1} + g_{i,n-1}^{(2)} q_n^{(-)} \\
& + g_{in}^{(1)} q_n^{(+)} + g_{in}^{(2)} q_{n+1}
\end{aligned} \tag{3.33}$$

and the assembled equations will become

$$\mathbf{H} \mathbf{p} = \mathbf{G} \mathbf{q} + \hat{\mathbf{g}}_n q_n^{(+)} \tag{3.34}$$

in which $\hat{\mathbf{g}}_n$ is a vector given by

$$\hat{\mathbf{g}}_n = \begin{pmatrix} g_{1n}^{(1)} \\ g_{2n}^{(1)} \\ \vdots \\ g_{Nn}^{(1)} \end{pmatrix} \tag{3.35}$$

and the expression for G_{in} gets modified to

$$G_{in} = g_{i,n-1}^{(2)} \tag{3.36}$$

It should be noted that since $q_n^{(+)}$ is specified $\hat{\mathbf{g}}_n q_n^{(+)}$ simply adds a known vector in the equations.

A somewhat similar condition applies to the corner node at the interface of dam face and reservoir bottom (Fig. 3.6). In this case both $q_m^{(-)}$ and $q_m^{(+)}$ are specified and hence only the pressure is unknown.

The i th equation becomes

Fig. 3.6 A Corner Node at the Interface of Dam Face and Reservoir Bottom

$$\begin{aligned}
 & - \left(1 - \frac{\beta}{2\pi} \right) p_i + \sum_{j=1}^N \left(h_{ij}^{(1)} p_j + h_{ij}^{(2)} p_{j+1} \right) \\
 & = \sum_{\substack{j=1 \\ j \neq m-1 \\ j \neq m}}^N \left(g_{ij}^{(1)} q_j + g_{ij}^{(2)} q_{j+1} \right) \\
 & + g_{i,m-1}^{(1)} q_{m-1} + g_{i,m-1}^{(2)} q_m^{(-)} \\
 & + g_{im}^{(1)} q_m^{(+)} + g_{im}^{(2)} q_{m+1}
 \end{aligned} \tag{3.37}$$

The assembled equations become

$$\mathbf{H} \mathbf{p} = \mathbf{G} \mathbf{q} + \hat{\mathbf{g}}_{m-1} q_m^{(-)} + \hat{\mathbf{g}}_m q_m^{(+)} \tag{3.38}$$

where

$$\hat{\mathbf{g}}_{m-1} = \begin{pmatrix} g_{1,m-1}^{(2)} \\ g_{2,m-1}^{(2)} \\ \vdots \\ g_{N,m-1}^{(2)} \end{pmatrix} \tag{3.39}$$

$$\hat{g}_m = \begin{pmatrix} g_{1,m}^{(1)} \\ g_{2,m}^{(1)} \\ \vdots \\ g_{N,m}^{(1)} \end{pmatrix} \quad (3.40)$$

and

$$G_{im} = 0 \quad (3.41)$$

The nodes at the interface of the reservoir bottom and far end, as well as the reservoir surface and far end, are treated by splitting them into two nodes (Fig. 3.7).

Fig. 3.7 Corner Nodes at Far End of the Reservoir

3.7 Incorporation of Infinite Radiation and Foundation Damping Conditions

The derivation of the eigenvalue problem associated with the modelling of the infinite reservoir was presented in detail in Section 2.5. The special boundary condition to account for the damping provided by the flexible foundation was presented in Section 2.4.

The incorporation of these two conditions in the two-dimensional BEM is described in this section.

To match the solutions at the interface of the finite and infinite region, and to apply the condition along the reservoir bottom Eq. 3.15 is expressed in a partitioned form as follows:

$$\begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad (3.42)$$

in which p_1, q_1 represent, respectively, the pressures and pressure derivatives at the nodes on the reservoir surface and along the dam face; p_2, q_2 represent the same quantities on the reservoir bottom while p_3, q_3 represent them at the nodes on the far boundary.

Based on the previously presented derivation, the pressure gradient on the flexible bottom is given by

$$q_2 = -\frac{w}{g} a_n - i\omega\gamma p_2 \quad (3.43)$$

in which a_n = vector of prescribed acceleration components along the normal to reservoir bottom and the term $i\omega\gamma p_2$ accounts for foundation damping.

The following two expressions represent the infinite radiation condition in the 2D case:

$$p_3 = \Psi\eta + \frac{w}{g} \Psi K^{-2} \Psi^T d \quad (3.44a)$$

$$q_3 = \Psi K \eta \quad (3.44b)$$

where Ψ, K, η are the same quantities as described in Section 2.5. The second term in p_3 represents the effect of the vertical ground excitation along the infinite section. When this excitation is absent, d is zero and the corresponding term vanishes. Eigenvector Ψ are real valued if there is no foundation damping. When damping is present Ψ will be complex valued.

When linear boundary elements are used, p_3 and q_3 are directly derived from Eqs. 3.44. For constant boundary elements the nodal values of constant boundary elements must be related to the finite element nodal values of Eqs. 3.44. For ease of reference the finite

element values are denoted as $\hat{p}$ and $\hat{q}$. The relationship between p_3 , q_3 and $\hat{p}$ and $\hat{q}$ is given by

$$p_3 = T_p \hat{p} \quad (3.45a)$$

$$q_3 = T_p \hat{q} \quad (3.45b)$$

where T_p is the transformation matrix between boundary element nodal values and finite element nodal values at the far boundary. For linear elements T_p is an identity matrix, I. For constant boundary elements any rational method can be used to derive T_p . One possible method is described in Appendix D.

Substitution of Eqs. 3.45, 3.44 and 3.43 in Eqs. 3.42 gives

$$\begin{pmatrix} H_{11} & H_{12} + i\omega\gamma G_{12} & H_{13}T_p\Psi + G_{13}T_p\Psi K \\ H_{21} & H_{22} + i\omega\gamma G_{22} & H_{23}T_p\Psi + G_{23}T_p\Psi K \\ H_{31} & H_{32} + i\omega\gamma G_{32} & H_{33}T_p\Psi + G_{33}T_p\Psi K \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_2 \\ \eta \end{pmatrix} = \begin{pmatrix} G_{11}q_1 - \frac{\omega}{g}G_{12}a_n - \frac{\omega}{g}H_{13}T_p\Psi K^{-2}\Psi^T d \\ G_{21}q_1 - \frac{\omega}{g}G_{22}a_n - \frac{\omega}{g}H_{23}T_p\Psi K^{-2}\Psi^T d \\ G_{31}q_1 - \frac{\omega}{g}G_{32}a_n - \frac{\omega}{g}H_{33}T_p\Psi K^{-2}\Psi^T d \end{pmatrix} \quad (3.46)$$

Equations 3.46 represent the boundary element formulation in the 2D case, which include infinite radiation condition and foundation damping and is valid for the constant boundary elements as well as for the linear boundary elements. Equations 3.46 contain N unknowns values of p and q . After suitable rearrangement these equations can be solved for the unknowns.

3.8 Boundary Element Superposition Formulation in 2D Case

A variation of the boundary element method (BEM) is the so-called boundary element superposition method (BESM). The superposition methods have been used to solve certain problems of elastostatics and plate bending (70,71). The superposition method is found to be less successful when boundary conditions change rapidly or a singularity is present. The motion of water reservoir in the two-dimensional case is governed by a Helmholtz equation and thus belongs to a class of potential problems where potential and potential

gradient do not exhibit singularity. The superposition method was, therefore, expected to be successful in this application at the earliest stage of this study. The computational efforts in the application of the superposition method are significantly lower in volume than those required for the BEM. However, it was discovered that the results obtained were quite sensitive to the location of the so-called congruent boundary. This introduced another level of uncertainty in the accuracy of the results and detracted significantly from the efficiency of the numerical method. Similar difficulties have been noted by other researchers (73). The results of the research concerning the BESM are presented in brief here (57).

Fig. 3.8 A Finite Domain Embedded in an Infinite Domain

In applying the BESM it is assumed that the domain D under consideration is embedded in an infinite domain (Fig. 3.8). Since solutions of the governing equation for infinite domain are easily available, they can be used to find the pressures and pressure derivatives due to any prescribed sources within the domain. These solutions will, however, violate the conditions on the boundary. In order to rectify this situation, it is assumed that a number of additional point sources exist in the infinite domain but external to D .

The strengths of these sources can be so selected that the boundary conditions on S are satisfied. Then the superposition of infinite solutions due to internal and external sources will be equivalent to the solution for finite domain and only internal sources.

In practical solution, boundary conditions are satisfied at a selected number of points say N along the boundary. The additional sources of unknown strengths are located along the boundary external to the domain. The external boundary is referred to as the congruent boundary (Fig. 3.8). The number of unknown sources is also N . The congruent boundary is generally taken to be parallel to the real boundary and located at an arbitrary distance D from it.

Fig. 3.9 A Rigid Gravity Dam Storing a Finite Water Reservoir
- Boundary Element Superposition Formulation

Referring to Fig. 3.9, which shows the boundary element discretization of a finite reservoir, the pressure p_i^* , and pressure derivative $\frac{\partial p_i^*}{\partial n}$, arising at a point i due to a point source of strength Q_j located on the congruent boundary, are given by

$$p_i^* = Q_j G(kr) \quad (3.47)$$

$$\frac{\partial p_i^*}{\partial n} = Q_j G'(kr) \cos \theta \quad (3.48)$$

where n = outward normal, θ = the angle between the radius vector r and the normal n , and $G'(kr)$ is the derivative of $G(kr)$. Function $G(kr)$ is referred to as the fundamental

solution of the governing Helmholtz equation (see Fig. 3.3). For a finite reservoir, either of the following fundamental solutions may be used:

$$G(kr) = \frac{1}{4}Y_0(kr) \quad (3.49)$$

$$G(kr) = \frac{1}{4}H_0^{(2)}(kr) \quad (3.50)$$

in which $Y_0(kr)$ = Bessel function of zero order and second kind, and $H_0^{(2)}(kr)$ = Hankel function of zero order and second kind. The solution given by Eq. 3.50 which contains only outgoing waves may be preferable for infinite reservoirs (50,51).

The total hydrodynamic pressure and pressure derivative at i are obtained by summing over the point sources.

$$p_i = \sum_j Q_j G(kr) \quad (3.51)$$

$$\frac{\partial p_i}{\partial n} = \sum_j Q_j G'(kr) \quad (3.52)$$

By satisfying the given mixed boundary valued problem at a number of selected points on reservoir boundary, a set of simultaneous equations in Q_j is obtained. These equations are solved for the unknown source strengths Q_j , which can then be used to find the pressure and pressure derivative at any point within the reservoir or on its boundary.

If the number of boundary nodes over which the given boundary conditions are to be satisfied is selected as N , the complete set of equations can be expressed as

$$C Q = b \quad (3.53)$$

where

$$b^T = [b_1 \ b_2 \ b_3 \ \dots \ b_N] = \text{prescribed boundary conditions}$$

$$Q^T = [Q_1 \ Q_2 \ Q_3 \ \dots \ Q_N] = \text{unknown point source strengths}$$

$$C = N \times N \text{ influence matrix}$$

When pressure is prescribed, $b_i = p_i$ and $C_{ij} = G(kr_{ij})$. When pressure derivative is prescribed, $b_i = \frac{\partial p_i}{\partial n}$ and $C_{ij} = G'(kr_{ij}) \cos \theta$.

Equations 3.53 constitute the boundary element superposition formulation in the case of the 2D water reservoir.

3.9 Infinite Radiation Condition in 2D BESM

An infinite radiation condition for the two-dimensional BESM is presented here for the case of a rigid foundation. The infinite radiation condition is based on the classical continuum solution which can be found in reference 28. The procedure presented is similar to that investigated by Wolf and Darbe (73) in their search for the dynamic-stiffness matrix of soil by the boundary element method.

The continuum solution reported in reference 28 can be rewritten in the following way:

$$p(x, y, \omega) = \frac{w}{g} \frac{2}{H} \sum_{n=1}^{\infty} \frac{I_n}{\kappa_n} \cos \lambda_n y e^{-\kappa_n x} \quad (3.54)$$

where $I_n = \int_0^H a_z(y) \cos \lambda_n y dy$, $\kappa_n = \sqrt{\lambda_n^2 - \frac{\omega^2}{c^2}}$, and $\lambda_n = \frac{(2n-1)\pi}{2H}$.

Referring to Fig. 3.10 and assuming that the interface between the finite and infinite regions is discretized by means of a finite number of constant boundary elements, Eq. 3.54 takes the following form for k th element:

$$p_k(y, \omega) = \frac{2w}{gH} \sum_{n=1}^{\infty} \frac{I_n}{\kappa_n} \cos \lambda_n y_k \quad (3.55)$$

The integral I_n has the following form:

$$I_n = \int_0^H a_z(y) \cos \lambda_n y dy \quad (3.56)$$

in which $a_z(y) = -\frac{1}{\rho} \frac{\partial p}{\partial z}$ is the acceleration at the far boundary and $\rho = \frac{w}{g}$, the mass density of water.

Fig. 3.10 Discretization of the Far Boundary

On substituting for $a_x(y)$, Eq. 3.56 becomes

$$I_n = -\frac{1}{\rho} \int_0^H \frac{\partial p}{\partial x} \cos \lambda_n y \, dy \quad (3.57)$$

Since $q = \frac{\partial p}{\partial x}$ is assumed to be constant over an element the integrand in Eq. 3.57 reduces to

$$\begin{aligned} -\rho I_n &= \int_0^{\Delta y} q_1 \cos \lambda_n y \, dy + \int_{\Delta y}^{2\Delta y} q_2 \cos \lambda_n y \, dy + \dots + \int_{(m-1)\Delta y}^H q_m \cos \lambda_n y \, dy \\ &= \frac{q_1}{\lambda_n} \sin \lambda_n y \Big|_0^{\Delta y} + \frac{q_2}{\lambda_n} \sin \lambda_n y \Big|_{\Delta y}^{2\Delta y} + \dots + \frac{q_m}{\lambda_n} \sin \lambda_n y \Big|_{(m-1)\Delta y}^H \end{aligned} \quad (3.58)$$

Thus, Eq. 3.55 can be rewritten in the following form:

$$\begin{aligned} p_k(y, \omega) &= -\frac{2}{H} \sum_{n=1}^{\infty} \frac{\int_0^H q \cos \lambda_n y \, dy}{\kappa_n} \cos \lambda_n y_k \\ &= -\frac{2}{H} \sum_{n=1}^{\infty} \frac{\left(\frac{q_1}{\lambda_n} \sin \lambda_n y \Big|_0^{\Delta y} + \frac{q_2}{\lambda_n} \sin \lambda_n y \Big|_{\Delta y}^{2\Delta y} + \dots + \frac{q_m}{\lambda_n} \sin \lambda_n y \Big|_{(m-1)\Delta y}^H \right)}{\kappa_n} \cos \lambda_n y_k \end{aligned} \quad (3.59)$$

After, several algebraic operations the interface condition can be set in the following matrix form:

$$\mathbf{p} = \mathbf{S} \mathbf{q} \quad (3.60)$$

where $\mathbf{p}$ and $\mathbf{q}$ are vectors of pressures and pressure normal derivatives at the interface. Matrix $\mathbf{S}$ is a symmetric matrix of size $m \times m$ whose elements are given by

$$S_{ij} = -\frac{2}{H} \sum_{n=1}^{\infty} \frac{(\sin \lambda_n i \Delta y - \sin \lambda_n (i-1) \Delta y)}{\lambda_n \kappa_n} \cos \lambda_n y_j \quad (3.61)$$

Each term of the matrix $\mathbf{S}$ constitutes a convergent infinite series.

Infinite condition of the form given by Eq. 3.60 is valid provided damping in the infinite region is neglected. The condition can be used either with the derived boundary element method formulation or the superposition method formulation.

Referring to Fig. 3.11 let these be n_1 nodes on the surface of the reservoir and the dam face, n_2 nodes along the foundation bottom and n_3 nodes on the far end. Eqs. 3.60 can now be expressed in a partitioned form as

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} \begin{pmatrix} \mathbf{Q} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{q}_2 \\ \mathbf{p}_3 \end{pmatrix} \quad (3.62)$$

Known values $\mathbf{b}_1$ consist of pressures and pressure derivatives on the reservoir surface and dam face, $\mathbf{q}_2$ are the pressure derivatives along the reservoir bottom and $\mathbf{p}_3$ are the pressures on the far end.

Foundation damping may exist in the finite region only. The foundation condition for the finite region and the far end radiation condition are introduced as follows:

$$\mathbf{p}_3 = \mathbf{S} \mathbf{q}_3 = \mathbf{S} \mathbf{B}_3 \mathbf{Q} \quad (3.63a)$$

$$\mathbf{q}_2 = \mathbf{F}_d \mathbf{p}_2 = \mathbf{F}_d \mathbf{A}_2 \mathbf{Q} \quad (3.63b)$$

where matrix $\mathbf{S}$ is defined by Eq. 3.61, matrix $\mathbf{F}_d$ is a diagonal matrix with terms equal to $i\omega\gamma$ and represents the foundation damping and matrices $\mathbf{A}$ and $\mathbf{B}$ are given by

Fig. 3.11 Break-Down of the Boundary Conditions in the Two-Dimensional BESM

$$A_{ij} = G(kr_{ij}) \quad (3.64a)$$

$$B_{ij} = G'(kr_{ij}) \cos \theta \quad (3.64b)$$

Substitution of Eqs. 3.63 and 3.64 into Eqs. 3.62 gives

$$\begin{pmatrix} C_1 \\ B_2 - F_d A_2 \\ A_3 - S B_3 \end{pmatrix} (Q) = \begin{pmatrix} b_1 \\ 0 \\ 0 \end{pmatrix} \quad (3.65)$$

Equations 3.65 are complex whenever F_d is non-zero i.e. foundation damping exist. They are complex also when the Hankel function fundamental solution is used. These equations constitute the boundary element superposition formulation for the two-dimensional Helmholtz equation. Equations 3.65 contain N values of unknown source strengths. Once the source strengths have been determined they can be used to find the pressure and pressure gradient at any point within the reservoir or on its boundary.

CHAPTER 4

HYDRODYNAMIC PRESSURES IN 2D RESERVOIRS

4.1 General Comments

The development of the direct boundary element and boundary element superposition formulation for the analysis of hydrodynamic pressures in a 2D infinite or finite reservoir was presented in Chapters 2 and 3. For the numerical implementation of the analysis procedure a series of computer programs were developed. All of the programs were designed to accept an excitation which may vary from element to element.

This chapter presents the results of analytical studies of the hydrodynamic pressures in 2D reservoirs subjected to a horizontal or vertical harmonic motion. At the beginning of this chapter, results relating to 2D reservoirs are presented. They are followed by results of a more extensive study of 2D infinite reservoirs. Effects of reservoir geometry, such as an inclined dam face and a sloped bottom as well as of infinite radiation are studied. Convergence behaviour and the manner in which it is altered by the number and size of constant elements, location of the far boundary and by the number of finite elements used to model a radiation condition, are examined. Finally, a comparison of the results obtained by boundary element superposition method and the boundary element method is presented for a few selected cases.

4.2 Hydrodynamic Pressures in 2D Finite Reservoirs by BEM

A major goal of the research reported here was to develop a BEM solution procedure for calculating the hydrodynamic pressure in a general 2D water reservoir that included the

effect of infinite radiation and foundation damping. Thus, a majority of studies reported here are on infinite reservoirs. Some selected results relating to finite reservoirs are presented for the sake of completeness.

In order to be able to compare the results with known classical solutions, two simple rectangular reservoirs are considered (one with a reservoir length H and the other with length $2H$ where H is the height of reservoir). The constant boundary element models for the two reservoirs are shown in Fig. 4.1.

Fig. 4.1 Two Finite Reservoirs Models - with
 (a) 60 Constant Boundary Elements
 (b) 40 Constant Boundary Elements

The following data is used in the computation of the results presented: height of the reservoir, $H = 100 \text{ m}$; length of the reservoir $L = 2H$ or H ; velocity of sound in water, $c = 1440 \frac{\text{m}}{\text{s}}$; the mass density of water, $\rho = 1 \frac{\text{tonne}}{\text{m}^3}$.

The constant boundary element model consists of 40 constant boundary elements for a reservoir length $L = H = 100 \text{ m}$ and 60 constant boundary elements for $L = 2H = 200 \text{ m}$.

The variations of hydrodynamic pressure along the dam face for the longer reservoir, $L = 2H = 200 \text{ m}$, for three exciting frequencies given by $\frac{\omega}{\omega_1} = 0.4, 1.2, 2.0$ and a coefficient of reflection $\alpha_R = 1.0$ (rigid foundation case), caused by a unit horizontal harmonic excitation of the dam, are plotted in Figs. 4.2a, 4.2b, and 4.2c. Similar results for the shorter reservoir, $L = H = 100 \text{ m}$, are plotted in the Figs. 4.3a, 4.3b and 4.3c. It should be noted that in conformity with the assumption of constant pressure in an element, the constant boundary element results should be truly represented as a stepped variation. However, in the graphs presented here a smooth curve has been fitted through the values at the centre nodes of the individual elements. Such a smoothing of data improves the match between the numerical results and the true variation of pressure and it has been adopted throughout this thesis.

The results from the BEM analysis are compared with classical solution for the rigid case and show relatively good agreement. The agreement is expected to improve with an increase in the number of elements used in the analysis. It will be noted that for each exciting frequency, whether less than the fundamental frequency or greater than it, the pressure values are significantly affected by the length of the reservoir. It is evident that for a finite reservoir the reservoir shape will be expected to have a significant effect on the hydrodynamic pressures.

It should be noted that in the absence of any damping, the pressure values are entirely in phase with the excitation and the imaginary part is absent.

Fig. 4.2a Distribution of the Total Hydrodynamic Pressure on a Gravity
 Dam Face due to Harmonic Horizontal Excitation
 in a Finite Reservoir - $\frac{\omega}{\omega_1} = 0.4$, $\frac{L}{H} = 2.0$

Fig. 4.2b Distribution of the Total Hydrodynamic Pressure on a Gravity
 Dam Face due to Harmonic Horizontal Excitation
 in a Finite Reservoir - $\frac{\omega}{\omega_1} = 1.2$, $\frac{L}{H} = 2.0$

Fig. 4.2c Distribution of the Total Hydrodynamic Pressure on a Gravity
 Dam Face due to Harmonic Horizontal Excitation
 in a Finite Reservoir - $\frac{\omega}{\omega_1} = 2.0$, $\frac{L}{H} = 2.0$

Fig. 4.3a Distribution of the Total Hydrodynamic Pressure on a Gravity
 Dam Face due to Harmonic Horizontal Excitation
 in a Finite Reservoir - $\frac{\omega}{\omega_1} = 0.4$, $\frac{L}{H} = 1.0$

Fig. 4.3b Distribution of the Total Hydrodynamic Pressure on a Gravity
 Dam Face due to Harmonic Horizontal Excitation
 in a Finite Reservoir - $\frac{\omega}{\omega_1} = 1.2$, $\frac{L}{H} = 1.0$

Fig. 4.3c Distribution of the Total Hydrodynamic Pressure on a Gravity
Dam Face due to Harmonic Horizontal Excitation
in a Finite Reservoir - $\frac{\omega}{\omega_1} = 2.0$, $\frac{L}{H} = 1.0$

4.3 Hydrodynamic Pressures in 2D Infinite Reservoirs by BEM

As stated earlier two alternative formulations have been developed for the analysis of reservoirs: a constant element formulation and a linear element formulation. In both cases, an infinite reservoir is represented by a finite region of regular or irregular boundary, as the case may be, and an infinite region of uniform rectangular section. The infinite region is modelled by one-dimensional finite element discretization. The modelling leads to an eigenvalue problem, which is solved by means of the QZ algorithm originally developed by Moler and Stewart (74), and later upgraded at the Argonne Research Laboratory U.S. and adapted to run on Apollo computer as part of the present work.

4.3.1 Effects of Infinite Radiation and Foundation Damping in Rectangular Infinite Reservoir

Selected results of analytical studies carried out on infinite reservoirs are presented here. In order to be able to compare the results with known classical solutions, a reservoir with simple boundaries is considered (Fig. 4.4a). The following data is used in the computation of the results shown: height of the reservoir, $H = 100 \text{ m}$; velocity of sound in water, $c = 1440 \frac{\text{m}}{\text{s}}$; the mass density of water, $\rho = 1 \frac{\text{tonne}}{\text{m}^3}$. The far end is placed at a distance of 200 m from the dam face.

The constant boundary element model of the considered reservoir is shown in Fig. 4.4a and the linear boundary element model in Fig. 4.4b. The constant element model consists of 60 boundary elements, while the linear model has 62 elements. In both cases 10 finite elements are used to model the infinite radiation condition.

The total hydrodynamic force acting on the dam face and resulting from a unit horizontal harmonic acceleration of the dam is plotted in Figs. 4.5a, 4.5b and 4.5c, as a function of the frequency ratios $\frac{\omega}{\omega_1}$, where ω_1 is the first natural frequency of the reservoir, given by $\omega_1 = \frac{\pi c}{2H}$. The results presented in Fig. 4.5a show the real part of the hydrodynamic force; Fig. 4.5b shows the imaginary part. The absolute value of the force is shown in Fig. 4.5c.

Fig. 4.4a The Constant Boundary Element Model with 60 Nodes

Fig. 4.4b The Linear Boundary Element Model with 62 Nodes

The results presented in Figs. 4.5 have been obtained from a linear boundary element formulation. Full reflection of incident waves at the reservoir bottom is represented by a coefficient of reflection, $\alpha_R = 1.0$. Partial damping due to wave absorption in the alluvial deposit at the bottom of the reservoir is represented by two values of α_R smaller than 1, namely $\alpha_R = 0.50$ and $\alpha_R = 0.85$. Complete absorption of the waves or full damping case is represented by $\alpha_R = 0$. The results obtained from a classical solution for the rigid foundation case ($\alpha_R = 1.0$) are also shown.

An examination of the results shows that foundation damping significantly modifies the response near resonance. However, away from the resonance, the effect of damping is not so pronounced.

The hydrodynamic forces on the dam produced by a unit vertical harmonic motion are presented in Figs. 4.6. Figure 4.6a shows the real part of the forces, Fig. 4.6b shows the imaginary part while Fig. 4.6c shows the absolute value. As in the case of the horizontal acceleration, the hydrodynamic force is plotted as a function of the frequency ratios $\frac{\omega}{\omega_1}$ for four different conditions of foundation damping. The classical solution for a rigid foundation case, is also presented. Comparison between the classical and the boundary element results is very good. Foundation damping reduces the response significantly, particularly near regions of resonance.

Hall and Chopra (28) and Fenves and Chopra (38) have obtained analytical results for the same set of data using FEM and classical solutions. To retain clarity the FEM and classical solutions for values of α_R other than 1.0 are not included in Figs. 4.5 and Figs. 4.6. They are, in fact, in good agreement with the BEM results. As one example, BEM results for $\alpha_R = 0.85$ and $\alpha_R = 1.0$ are compared with corresponding FEM results and classical solutions reported in references (28) and (38), in Figs. 4.7 and 4.8 for horizontal and vertical excitations respectively. The close comparison is evident from these Figures.

For the rigid foundation case, the graph of the hydrodynamic forces caused by vertical excitation exhibits discontinuity near resonance, the force value changing from $+\infty$ to $-\infty$ as the excitation frequency increases from below natural frequency to above it. With foundation damping, the discontinuities are changed to inflexion points located at the points of resonance. Also, for the rigid foundation case there is no imaginary part in the hydrodynamic force, a result which is in agreement with the classical solution.

REAL PART

HORIZONTAL EXCITATION

Fig. 4.5a Real Part of the Total Hydrodynamic Force on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir - Linear BEM with 62 Nodes

IMAGINARY PART

HORIZONTAL EXCITATION

Fig 4.5b Imaginary Part of the Total Hydrodynamic Force on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir - Linear BEM with 62 Nodes

ABSOLUTE VALUE

HORIZONTAL EXCITATION

Fig. 4.5c Absolute Value of the Total Hydrodynamic Force on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir - Linear BEM with 62 Nodes

REAL PART

Fig. 4.6a Real Part of the Total Hydrodynamic Force on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir - Linear BEM with 62 Nodes

IMAGINARY PART

VERTICAL EXCITATION

Fig. 4.6b Imaginary Part of the Total Hydrodynamic Force on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir - Linear BEM with 62 Nodes

Fig. 4.6c Absolute Value of the Total Hydrodynamic Force on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir - Linear BEM with 62 Nodes

Fig. 4.7a Comparison of Absolute Values of the Total Hydrodynamic Forces on a Gravity Dam Face due to Harmonic Horizontal Excitation by BEM, FEM and Classical Solution $\cdot \alpha_R = 1.00$

Fig. 4.7b Comparison of Absolute Values of the Total Hydrodynamic Forces on a Gravity Dam Face due to Harmonic Horizontal Excitation by BEM, FEM and Classical Solution - $\alpha_R = 0.85$

ABSOLUTE VALUE

VERTICAL EXCITATION

Fig. 4.8a Comparison of Absolute Values of the Total Hydrodynamic Forces on a Gravity

Dan. Face due to Harmonic Vertical Excitation by BEM, FEM and Classical

Solution - $\alpha_R = 1.00$

ABSOLUTE VALUE

Fig. 48b Comparison of Absolute Values of the Total Hydrodynamic Forces on a Gravity Dam Face due to Harmonic Vertical Excitation by BEM, FEM and Classical Solution - $\alpha_R = 0.85$

4.3.2 Comparison of the Results of Constant Boundary Element and Linear Element Formulations

The two models used for a comparison of the results obtained from a constant and linear boundary element analysis have already been described in Subsection 4.3.1 and presented in Figs. 4.4a and 4.4b, respectively.

The results presented here show the variation of hydrodynamic pressure along the dam face for both horizontal and vertical harmonic excitation. The results are for one particular frequency namely $\frac{\omega}{\omega_1} = 1.2$. Three different types of damping are considered with following values of coefficient of reflection α_R : 0.5, 0.75 and 1.0. Figures 4.9 and 4.11 show the distribution of pressure values obtained from a constant element formulation while Figs. 4.10 and 4.12 show the results from a linear element formulation. For the purpose of comparison, classical solutions for the rigid foundation case are also presented. The actual shape function distributions which for a constant element case would show stepped graphs have not been used in Figs. 4.9 to 4.11. The curves are based on the midnode values for the constant boundary element formulation and end node values for the linear boundary element formulation respectively. The calculations of total hydrodynamic forces acting on the dam face are based on the assumptions of linear variation between nodal values for both the constant boundary element case and the linear boundary element case. This type of treatment improves accuracy of the results obtained from a constant boundary element representation.

The results obtained from the constant and the linear boundary element formulations are very similar and are in good agreement with the available classical solutions. Foundation damping generally reduces the pressures. For the particular frequency studied ($\frac{\omega}{\omega_1} = 1.2$), the imaginary part of the hydrodynamic pressure representing out-of-phase response is dominant for horizontal excitation while the real part or in-phase response is dominant in the case of vertical excitation.

From the results presented here, it is evident that the constant boundary elements provide as much accuracy as a linear boundary element formulation and the latter does not offer any significant advantage over the former.

Fig. 4.9a Real Part of the Hydrodynamic Pressure on a Gravity
Dam Face due to Harmonic Horizontal Excitation in
an Infinite Reservoir by Constant BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.9b Imaginary Part of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir by Constant BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.9c Total Absolute Value of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir by Constant BEM - $\frac{w}{\epsilon_1} = 1.2$

Fig. 4.10a Real Part of the Hydrodynamic Pressure on a Gravity
Dam Face due to Harmonic Horizontal Excitation
in an Infinite Reservoir by Linear BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.10b Imaginary Part of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir by Linear BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.10c Total Absolute Value of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Horizontal Excitation in an Infinite Reservoir by Linear BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.11a Real Part of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir by Constant BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.11b Imaginary Part of the Hydrodynamic Pressure on a Gravity
Dam Face due to Harmonic Vertical Excitation in
an Infinite Reservoir by Constant BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.11c Total Absolute Value of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir by Constant BEM - $\frac{\omega}{\omega_1} = 1.2$

2

Micred

Fig. 4.12a Real Part of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir by Linear BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.12b Imaginary Part of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir by Linear BEM - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.12c Total Absolute Value of the Hydrodynamic Pressure on a Gravity Dam Face due to Harmonic Vertical Excitation in an Infinite Reservoir by Linear BEM - $\frac{\omega}{\omega_1} = 1.2$

4.3.3 Effects of the Location of the Far Boundary

As stated earlier, in modelling an infinite reservoir an artificial boundary must be created at some distance from the dam face. The region of the reservoir beyond this boundary is treated as having a uniform rectangular section extending to infinity. It would be of interest to study the effect of the location of this boundary.

- The effect of the location of the far boundary in case of a rectangular infinite reservoir.

Six reservoirs with different length have been studied: $\frac{H}{L} = 0.5, 1.0, 1.5, 2.5, 3.0$, for four chosen frequencies of excitation in upstream-downstream direction only: $\frac{\omega}{\omega_1} = 0.8, 1.2, 1.6$ and 2.0 . The number of finite elements used to model infinite radiation is kept constant in all reservoir models and is equal to 10. The schematic view of these six models is presented in Fig. 4.13.

Fig. 4.13 Six Rectangular Infinite Reservoirs Models -
with Different Location of the Far End

The results obtained are presented in four Tables: Table 4-1a for $\frac{\omega}{\omega_1} = 0.8$, Table 4-2a for $\frac{\omega}{\omega_1} = 1.2$, Table 4-3a for $\frac{\omega}{\omega_1} = 1.6$ and Table 4-4a for $\frac{\omega}{\omega_1} = 2.0$. The results from classical solutions are presented in separate Tables: Table 4-1b for $\frac{\omega}{\omega_1} = 0.8$, Table 4-2b for $\frac{\omega}{\omega_1} = 1.2$,

Table 4-3b for $\frac{\omega}{\omega_1} = 1.6$ and Table 4-4b for $\frac{\omega}{\omega_1} = 2.0$.

The BEM results show the difference in the absolute values being generally less than 5% w.r.t. classical solutions. It is apparent from the results presented here that when the model represents the geometry faithfully, as it does in this case, the artificial far boundary need not be located too far from the dam. Thus moving the far boundary away from $0.5H$ to $1.0H$ or farther does not necessarily improve the results.

TABLE 4-1a

**Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Location of the Far End - $\frac{\omega}{\omega_1} = 0.8$**

y	30 CBE $L = 0.5 H$	40 CBE $L = 1.0 H$	50 CBE $L = 1.5 H$	60 CBE $L = 2.0 H$	70 CBE $L = 2.5 H$	80 CBE $L = 3.0 H$
95.	1.6931	1.7033	1.5652	1.6682	1.6665	1.6781
85.	4.2535	4.2796	3.9283	4.1901	4.1856	4.2147
75.	6.3009	6.3411	5.7969	6.2022	6.1953	6.2402
65.	8.0090	8.0621	7.3396	7.8778	7.8683	7.9278
55.	9.4384	9.5033	8.6162	9.2709	9.2652	9.3381
45.	10.6150	10.6910	9.6526	10.4260	10.4120	10.4970
35.	11.5500	11.6350	10.4610	11.3350	11.3200	11.4160
25.	12.2480	12.3420	11.0450	12.0110	11.9940	12.1000
15.	12.7070	12.8100	11.4030	12.4500	12.4320	12.5470
5.	12.8970	13.0080	11.5000	12.6230	12.6040	12.7270

TABLE 4-2a

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Location of the Far End - $\frac{\omega}{\omega_1} = 1.2$

30 Elem. L = 0.5 H				40 Elem. L = 1.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.6656	-0.8701	1.0955	0.7713	-0.9041	1.1883
85.	0.9465	-2.7480	2.9064	1.2600	-2.9535	3.2110
75.	0.8025	-4.5384	4.6088	1.3115	-4.9205	5.0923
65.	0.4501	-6.2105	6.2268	1.1411	-6.7608	6.8564
55.	-0.0010	-7.7289	7.7289	0.8560	-8.4296	8.4729
45.	-0.4776	-9.0597	9.0723	0.5267	-9.8849	9.8989
35.	-0.9255	-10.1730	10.2150	0.2052	-11.0900	11.0919
25.	-1.3038	-11.0430	11.1197	-0.0695	-12.0150	12.0152
15.	-1.5854	-11.6520	11.7594	-0.2714	-12.6340	12.6369
5.	-1.7801	-11.9860	12.1175	-0.4107	-12.9280	12.9345
50 Elem. L = 1.5 H				60 Elem. L = 2.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.7457	-0.8970	1.1664	0.7371	-0.8985	1.1621
85.	1.1181	-2.9111	3.1184	1.1153	-2.9047	3.1114
75.	1.1825	-4.8418	4.9841	1.1350	-4.8262	4.9579
65.	0.9653	-6.6476	6.7173	0.9000	-6.6231	6.6840
55.	0.6377	-8.2854	8.3099	0.5562	-8.2530	8.2717
45.	0.2712	-9.7153	9.7191	0.1757	-9.6766	9.6782
35.	-0.0813	-10.9020	10.9023	-0.1884	-10.8590	10.8606
25.	-0.3804	-11.8150	11.8241	-0.4764	-11.7710	11.7806
15.	-0.5994	-12.4320	12.4464	-0.7212	-12.3900	12.4110
5.	-0.7483	-12.7340	12.7560	-0.8728	-12.6980	12.7280
70 Elem. L = 2.5 H				80 Elem. L = 3.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.6899	-1.0218	1.0241	0.7658	-0.8699	1.1590
85.	0.9433	-3.2560	3.3899	1.2591	-2.8315	3.0988
75.	0.7625	-5.3897	5.4434	1.3172	-4.7127	4.8933
65.	0.3729	-7.3832	7.3926	1.1538	-6.4725	6.5745
55.	-0.1100	-9.1923	9.1930	0.8748	-8.0683	8.1156
45.	-0.6087	-10.7750	10.7922	0.5501	-9.4600	9.4768
35.	-1.0651	-12.0950	12.1418	0.2311	-10.6150	10.6175
25.	-1.4350	-13.1200	13.1982	-0.0441	-11.5020	11.5021
15.	-1.6875	-13.8280	13.9306	-0.2499	-12.0990	12.1016
5.	-1.8277	-14.2020	14.3191	-0.3977	-12.3060	12.3124

TABLE 4-1b

**Hydrodynamic Pressure by Classical Solution
for Infinite Rectangular Reservoir - $\frac{\omega}{\omega_1} = 0.8$**

y	Real Part
95.	1.8193
85.	4.3667
75.	6.4003
65.	8.0994
55.	9.5705
45.	10.6930
35.	11.6250
25.	12.3210
15.	12.7850
5.	13.0170

TABLE 4-2b

**Hydrodynamic Pressure by Classical Solution
for Infinite Rectangular Reservoir - $\frac{\omega}{\omega_1} = 1.2$**

y	Real Part	Imag. Part	Abs. Value
95.	0.7521	0.9772	1.2331
85.	1.1920	2.9077	3.1425
75.	1.1809	4.7665	4.9106
65.	0.9526	6.5080	6.5773
55.	0.6155	8.0892	8.1126
45.	0.2409	9.4713	9.4744
35.	-0.1185	10.6200	10.6207
25.	-0.4223	11.5070	11.5147
15.	-0.6411	12.1111	12.1279
5.	-0.7554	12.4170	12.4399

TABLE 4-3a

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Location of the Far End - $\frac{\omega}{\omega_1} = 1.6$

	30 Elem. L = 0.5 H			40 Elem. L = 1.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.7790	-0.4652	0.9073	0.7666	-0.4670	0.8977
85.	1.2761	-1.5345	1.9958	1.2387	-1.5169	1.6914
75.	1.3179	-2.2564	2.6131	1.2570	-2.5242	1.9584
65.	1.1188	-3.5290	3.7021	1.0360	-3.4666	3.6181
55.	0.7894	-4.4042	4.4744	0.6869	-4.3215	4.3310
45.	0.4064	-5.1668	5.1828	0.2866	-5.0677	5.0758
35.	0.0290	-5.7963	5.7964	-0.1053	-5.6866	5.6876
25.	-0.2967	-6.2758	6.2828	-0.4424	-6.1627	6.1786
15.	-0.5384	-6.5919	6.6138	-0.6922	-6.4838	6.5206
5.	-0.7016	-6.7328	6.7692	-0.8601	-6.6393	6.6948
	50 Elem. L = 1.5 H			60 Elem. L = 2.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.7568	-0.4769	0.8945	0.7738	-0.4652	0.9029
85.	1.2068	-1.5300	1.9487	1.2657	-1.5230	1.9803
75.	1.2039	-2.5374	2.8085	1.3034	-2.5396	2.8544
65.	0.9632	-3.4788	3.6097	1.1011	-3.4911	3.6606
55.	0.5960	-4.3328	4.3736	0.7688	-4.3542	4.4215
45.	0.1798	-5.0790	5.0822	0.3832	-5.1068	5.1211
35.	-0.2254	-5.6996	5.7039	0.0032	-5.7298	5.7298
25.	-0.5730	-6.1793	6.2058	-0.3250	-6.2068	6.2153
15.	-0.8301	-6.5069	6.5596	-0.5693	-6.5253	6.5501
5.	-1.0020	-6.6726	6.7474	-0.7356	-6.6735	6.7139
	70 Elem. L = 2.5 H			80 Elem. L = 3.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.8073	-0.4622	0.9302	0.7262	-0.5339	0.9005
85.	1.3610	-1.5224	2.0421	1.0810	-1.6529	1.9750
75.	1.4561	-2.5426	2.9300	0.9843	-2.7139	2.8869
65.	1.3066	-3.4982	3.7342	0.6546	-3.7020	3.7594
55.	1.0223	-4.3649	4.4830	0.2068	-4.5980	4.6026
45.	0.6792	-5.1203	5.1652	-0.2784	-5.3833	5.3905
35.	0.3359	-5.7447	5.7545	-0.7386	-6.0412	6.0862
25.	0.0378	-6.2215	6.2216	-1.1248	-6.5582	6.6540
15.	-0.1829	-6.5376	6.5402	-1.4019	-6.9244	7.0649
5.	-0.3320	-6.6808	6.6890	-1.5729	-7.1336	7.3049

TABLE 4-3b

Hydrodynamic Pressure by Classical Solution
for Infinite Rectangular Reservoir - $\frac{\omega}{\omega_1} = 1.6$

y	Real Part	Imag. Part	Abs. Value
95.	0.7770	-0.5190	0.9344
85.	1.2568	-1.5443	1.9911
75.	1.2660	-2.5315	2.8304
65.	1.0375	-3.4564	3.6087
55.	0.6820	-4.2962	4.3500
45.	0.2769	-5.0302	5.0378
35.	-0.1181	-5.6404	5.6416
25.	-0.4562	-6.1116	6.1286
15.	-0.7015	-6.4324	6.4705
5.	-0.8303	-6.5948	6.6469

TABLE 4-4b

Hydrodynamic Pressure by Classical Solution
for Infinite Rectangular Reservoir - $\frac{\omega}{\omega_1} = 2.0$

y	Real Part	Imag. Part	Abs. Value
95.	0.8186	0.3743	0.9001
85.	1.3664	1.1136	1.7627
75.	1.4122	1.8255	2.3080
65.	1.1854	2.4925	2.7600
55.	0.8002	3.0981	3.1998
45.	0.3427	3.6274	3.6435
35.	-0.1155	4.0674	4.0690
25.	-0.5149	4.4072	4.4372
15.	-0.8085	4.6385	4.7083
5.	-0.9636	4.7556	4.8525

TABLE 4-4a

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Location of the Far End - $\frac{\omega}{\omega_1} = 2.0$

30 Elem. L = 0.5 H				40 Elem. L = 1.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.8064	-0.3320	0.8720	0.8064	-0.3390	0.8748
85.	1.3517	-1.0858	1.7338	1.3479	-1.0953	1.7368
75.	1.4115	-1.8112	2.2962	1.4033	-1.8204	2.2985
65.	1.1966	-2.4914	2.7639	1.1834	-2.4987	2.7648
55.	0.8212	-3.1090	3.2156	0.8028	-3.1142	3.2160
45.	0.3712	-3.6481	3.6669	0.3471	-3.6515	3.6680
35.	-0.0823	-4.0944	4.0952	-0.1122	-4.0974	4.0989
25.	-0.4806	-4.4360	4.4620	-0.5163	-4.4408	4.4707
15.	-0.7797	-4.6636	4.7283	-0.8209	-4.6730	4.7445
5.	-0.9766	-4.7685	4.8675	-1.0227	-4.7857	4.8937
50 Elem. L = 1.5 H				60 Elem. L = 2.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.8115	-0.3343	0.8777	0.8065	-0.3374	0.8742
85.	1.3661	-1.0960	1.7514	1.3501	-1.0940	1.7377
75.	1.4345	-1.8288	2.3243	1.4082	-1.8201	2.3013
65.	1.2269	-2.5151	2.7984	1.1912	-2.4992	2.7686
55.	0.8574	-3.1375	3.2525	0.8135	-3.1150	3.2195
45.	0.4112	-3.6797	3.7026	0.3608	-3.6520	3.6698
35.	-0.0458	-4.1276	4.1278	-0.0956	-4.0967	4.0978
25.	-0.4389	-4.4693	4.4908	-0.4969	-4.4378	4.5165
15.	-0.7399	-4.6955	4.7534	-0.7990	-4.6666	4.7345
5.	-4.7968	-4.7968	4.8881	-0.9986	-4.7745	4.8778
70 Elem. L = 2.5 H				80 Elem. L = 3.0 H		
y	Real P.	Imag. P.	Abs. V.	Real P.	Imag. P.	Abs. V.
95.	0.7858	-0.3907	0.8776	0.8073	-0.3325	0.8731
85.	1.2372	-1.2052	1.7272	1.3543	-1.0834	1.7343
75.	1.1980	-1.9752	2.3101	1.4156	-1.8051	2.2940
65.	0.8854	-2.6890	2.8310	1.2014	-2.4810	2.7566
55.	0.4183	-3.3325	3.3599	0.8259	-3.0947	3.2030
45.	-0.1132	-3.8925	3.8941	-0.2784	-3.6306	3.6499
35.	-0.6342	-4.3557	4.4032	-0.7386	-4.0751	4.0759
25.	-1.0821	-4.7177	4.8402	-1.1248	-4.4167	4.4431
15.	-1.4092	-4.9677	5.1637	-0.2499	-4.6462	4.7125
5.	-1.6081	-5.1024	5.3498	-0.9903	-4.7547	4.8567

4.3.4 Effects of Inclined Dam Face on the Response of Rectangular Infinite Reservoirs

Reservoirs with four different angles of inclination of the dam face namely 2.5° , 5° , 10° , and 15° from the vertical have been studied. Hydrodynamic pressures are obtained in the reservoirs for three different frequencies. A schematic view of the four reservoir models is presented in Fig. 4.14.

Fig. 4.14 Four Rectangular Infinite Reservoir Models with Different Angle of Inclination of Dam Face

TABLE 4-5a

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Inclination of Dam Face - $\frac{\omega}{\omega_1} = 0.4$

y	Classical Solution	60 CBE 0°	OUTI1 2.5°	OUTI2 5°	OUTI3 10°	OUTI4 15°
95.	1.4356	1.3469	1.2795	1.2160	1.1012	0.9999
85.	3.2370	3.1454	3.0357	2.9327	2.7494	2.5898
75.	4.5532	4.4696	4.3347	4.2085	3.9877	3.7996
65.	5.5865	5.5078	5.3535	5.2099	4.9622	4.7561
55.	6.4120	6.3346	6.1635	6.0045	5.7338	5.5127
45.	7.0646	6.9884	6.8011	6.6270	6.3333	6.0968
35.	7.5689	7.4913	7.2872	7.0968	6.7776	6.5222
25.	7.9376	7.8564	7.6336	7.4241	7.0741	6.7939
15.	8.1791	8.0889	7.8456	7.6104	7.2204	6.9056
5.	8.2987	8.1634	7.9179	7.6198	7.1727	6.8078

TABLE 4-5b

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Inclination of Dam Face - $\frac{w}{w_1} = 1.2$

Force Part	y	Classical Solution	60 CBE 0°	OUT11 2.5°	OUT12 5°	OUT13 10°	OUT14 15°
Real Part	95.	0.7521	0.7371	0.7808	0.8133	0.8514	0.8580
	85.	1.1920	1.1529	1.3345	1.4953	1.7617	1.9564
	75.	1.1809	1.1350	1.4357	1.7113	2.1942	2.5842
	65.	0.9526	0.8999	1.2962	1.6660	2.3330	2.8979
	55.	0.6155	0.5562	1.0228	1.4634	2.2733	2.9795
	45.	0.2409	0.1757	0.6875	1.1742	2.0813	2.8879
	35.	-0.1185	-0.1884	0.3456	0.8524	1.8099	2.6721
	25.	-0.4223	-0.4964	0.0346	0.5410	1.5030	2.3753
	15.	-0.6411	-0.7212	-0.2101	0.2713	1.1947	2.0329
	5.	-0.7554	-0.8728	-0.3719	0.0352	0.8789	1.6418
Imaginary Part	95.	-0.9772	-0.8985	-0.8267	-0.7522	-0.6094	-0.4778
	85.	-2.9077	-2.9047	-2.7815	-2.6403	-2.3446	-2.0372
	75.	-4.7665	-4.8262	-4.6910	-4.5239	-4.1583	-3.7566
	65.	-6.5080	-6.6231	-6.4977	-6.3291	-5.9466	-5.5076
	55.	-8.0892	-8.2530	-8.1507	-7.9963	-7.6353	-7.1994
	45.	-9.4713	-9.6766	-9.6046	-9.4742	-9.1550	-8.7564
	35.	-10.6700	-10.8590	-10.8200	-10.7180	-10.4570	-10.1130
	25.	-11.5070	-11.7710	-11.7630	-11.6910	-11.4220	-11.2120
	15.	-12.1110	-12.3900	-12.4080	-12.3620	-12.2200	-12.2120
	5.	-12.4170	-12.6960	-12.7360	-12.7040	-12.6060	-12.4390
Absolute Value	95.	1.2331	1.1621	1.1372	1.1079	1.0470	0.9821
	85.	3.1425	3.1251	3.0851	3.0343	2.9327	2.8245
	75.	4.9106	4.9579	4.9058	4.8368	4.7017	4.5596
	65.	6.5773	6.6840	6.6257	6.5447	6.3879	6.2235
	55.	8.1126	8.2717	8.2146	8.1291	7.9646	7.7916
	45.	9.4744	9.6782	9.6292	9.5468	9.3886	9.2203
	35.	10.6207	10.8606	10.8255	10.7497	10.6125	10.4601
	25.	11.5147	11.7815	10.8200	11.7035	11.5899	11.4606
	15.	12.1279	12.4110	12.4098	12.3650	12.2783	12.1739
	5.	12.4399	12.7279	12.7414	12.7040	12.6356	12.5469

TABLE 4-5c

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
with Different Inclination of Dam Face - $\frac{w}{w_1} = 2.4$

Force Part	y	Classical Solution	60 CBE 0°	OUT11 2.5°	OUT12 5°	OUT13 10°	OUT14 15°
Real Part	95.	0.8992	0.8893	0.8835	0.8820	0.8592	0.7842
	85.	1.5791	1.5877	1.6326	1.6887	1.7681	1.7493
	75.	1.7031	1.7421	1.8303	1.9348	2.1084	2.1565
	65.	1.4855	1.5506	1.6685	1.8047	2.0407	2.1417
	55.	1.0456	1.1300	1.2620	1.4109	1.6721	1.8002
	45.	0.4839	0.5806	0.7107	0.8531	1.1019	1.2281
	35.	-0.1043	-0.0033	0.1092	0.2276	0.4296	0.5267
	25.	-0.6329	-0.5362	-0.4546	-0.3755	-0.2484	-0.2038
	15.	-0.9482	-0.9482	-0.9070	-0.8803	-0.8487	-0.8758
Imaginary Part	5.	-1.2413	-1.2162	-1.1973	-1.2590	-1.3371	-1.4520
	95.	-0.2971	-0.2637	-0.2603	-0.2541	-0.2295	-0.1769
	85.	-0.8841	-0.8647	-0.8867	-0.9040	-0.9025	-0.8007
	75.	-1.4493	-1.4437	-1.4905	-1.5343	-1.5723	-1.4508
	65.	-1.9788	-1.9866	-2.0467	-2.1084	-2.1766	-2.0430
	55.	-2.4595	-2.4790	-2.5366	-2.6017	-2.6765	-2.5236
	45.	-2.8797	-2.9077	-2.9467	-2.9992	-3.0511	-2.8635
	35.	-3.2290	-3.2612	-3.2685	-3.2945	-3.2957	-3.0560
	25.	-3.4988	-3.5298	-3.4982	-3.4892	-3.4204	-3.1150
Absolute Value	15.	-3.6325	-3.7060	-3.6361	-3.5912	-3.4473	-3.0724
	5.	-3.7754	-3.7811	-3.6871	-3.6100	-3.4045	-2.9724
	95.	0.9471	0.9275	0.9210	0.9179	0.8893	0.8039
	85.	1.8097	1.8079	1.8321	1.9154	1.9860	1.9238
	75.	2.2363	2.2626	2.3604	2.4699	2.6301	2.5991
	65.	2.4743	2.5201	2.6406	2.7753	2.9836	2.9598
	55.	2.6725	2.7244	2.8332	2.9596	3.1559	3.0999
	45.	2.9201	2.9651	3.0312	3.1182	3.2440	3.1157
	35.	3.2307	3.2612	3.2703	3.3023	3.3234	3.1010
	25.	3.5556	3.5703	3.5270	3.5093	3.4294	3.1217
	15.	3.8026	3.8739	3.7475	3.6975	3.5502	3.2358
	5.	3.9742	3.9719	3.8766	3.8232	3.6577	3.3081

The location of the far end is kept constant in all four models. The number of constant boundary elements is approximately the same (straight dam - 60 elements, dam with $\alpha = 15^\circ$ - 57 elements). The hydrodynamic pressures caused by a unit horizontal harmonic excitation are shown in Table 4-5a for $\frac{\omega}{\omega_1} = 0.4$, in Table 4-5b for $\frac{\omega}{\omega_1} = 1.2$ and in Table 4-5.c for $\frac{\omega}{\omega_1} = 2.4$. They show a decrease in hydrodynamic pressure with an increase of the angle of inclination of dam face. However, the overall change in hydrodynamic pressure is small.

4.3.5 Effects of Sloped Bottom in Rectangular Infinite Reservoirs

Reservoirs with four different angles of inclination of the reservoir bottom namely 2.5° , 5° , 10° and 15° from the horizontal have been studied: for three chosen frequencies. Hydrodynamic pressures were obtained for a horizontal harmonic acceleration of the dam face and the foundation. The excitation has a component normal to the sloped bottom. The schematic view of these four models is presented in Fig. 4.15.

Fig. 4.15 Four Rectangular Infinite Reservoir Models with Different Angles of Inclination of the Reservoir Bottom

The location of the far end is kept constant in all four models. The number of constant

boundary elements is the same in all models.

The results obtained are presented in three Tables: Table 4-6a, 4-6b and 4-6c. Table 4-6a shows the results for $\frac{\omega}{\omega_1} = 0.4$, Table 4-6b for $\frac{\omega}{\omega_1} = 1.2$ and Table 4-6c for $\frac{\omega}{\omega_1} = 2.4$. For a frequency of $\frac{\omega}{\omega_1} = 0.4$, the hydrodynamic pressures increase with slope but only moderately. For a frequency of $\frac{\omega}{\omega_1} = 1.2$, the increase in the hydrodynamic pressures with increasing inclination is more dramatic. For a frequency of $\frac{\omega}{\omega_1} = 2.4$, the hydrodynamic pressures, in fact, decrease with increasing inclination, although the change is quite modest. In conclusion, it can be stated that for $\alpha_R = 1.0$, the hydrodynamic pressures due to horizontal excitation may be significantly affected by the geometry of the reservoir bottom, but there is no recognized trend in the effect due to the slope of the bottom.

TABLE 4-6a

**Hydrodynamic Pressure in Infinite Rectangular Reservoirs with
Different Angles of Inclination of a Sloped Bottom - $\frac{\omega}{\omega_1} = 0.4$**

y	Classical Solution	60 CBE 0°	OUTB1 2.5°	OUTB2 5°	OUTB3 10°	OUTB4 15°
95.	1.4356	1.3469	1.3512	1.3558	1.3663	1.3786
85.	3.2370	3.1454	3.1596	3.1749	3.2094	3.2501
75.	4.5532	4.4696	4.4942	4.5206	4.5800	4.6502
65.	5.5865	5.5078	5.5434	5.5817	5.6678	5.7696
55.	6.4120	6.3346	6.3824	6.4338	6.5493	6.6860
45.	7.0646	6.9884	7.0500	7.1161	7.2650	7.4409
35.	7.5689	7.4913	7.5689	7.6522	8.2894	8.0605
25.	7.9376	7.8564	7.9530	8.0567	8.6250	8.5631
15.	8.1791	8.0889	8.2089	8.3375	8.8222	8.9609
5.	8.2987	8.1634	8.3121	8.4707	8.8222	9.2285

TABLE 6-5b

Hydrodynamic Pressure in Infinite Rectangular Reservoirs with
Different Angles of Inclination of a Sloped Bottom - $\frac{\omega}{\omega_1} = 1.2$

Force Part	y	Classical Solution	60 CBF 0°	OUTB1 2.5°	OUTB2 5°	OUTB3 10°	OUTB4 15°
Real Part	95.	0.7521	0.7371	0.3620	-0.2896	-5.6404	9.9509
	85.	1.1920	1.1529	-0.0581	-2.1641	-19.4720	30.0983
	75.	1.1809	1.1350	-0.8722	-4.3647	-33.0840	50.6580
	65.	0.9526	0.8999	-1.8443	-6.6226	-45.9410	68.7440
	55.	0.6155	0.5562	-2.8466	-8.7765	-57.6120	84.8950
	45.	0.2409	0.1757	-3.7895	-10.7070	-67.7390	98.7750
	35.	-0.1185	-0.1884	-4.6056	-12.3220	-12.3220	110.1000
	25.	-0.4223	-0.4964	-5.2441	-13.5530	-13.5530	118.6700
	15.	-0.6411	-0.7212	-5.6718	-14.3550	-14.3550	124.3600
	5.	-0.7554	-0.8728	-5.8971	-14.7330	-88.1830	127.1200
Imaginary Part	95.	-0.9772	-0.8985	-0.9535	-1.0855	-3.2034	-0.4142
	85.	-2.9077	-2.9047	-3.0835	-3.5114	-10.3660	-1.3389
	75.	-4.7665	-4.8262	-5.1227	-5.8325	-17.2100	-2.2206
	65.	-6.5080	-6.6231	-7.0278	-7.9988	-23.5810	-3.0385
	55.	-8.0892	-8.2530	-8.7535	-9.9580	-29.3200	-3.7717
	45.	-9.4713	-9.6766	-10.2580	-11.6620	-34.2850	-4.4018
	35.	-10.6700	-10.8590	-11.5030	-13.0600	-38.3540	-4.9141
	25.	-11.5070	-11.7710	-12.4600	-14.1440	-41.4360	-5.2982
	15.	-12.1110	-12.3900	-13.1060	-14.8650	-43.4720	-5.5486
	5.	-12.4170	-12.6980	-13.4220	-15.2140	-44.4340	-5.6653
Absolute Value	95.	1.2331	1.1621	1.0199	1.1235	6.4866	9.9192
	85.	3.1425	3.1251	3.0840	4.1247	22.0593	30.1281
	75.	4.9106	4.9579	5.1964	7.2848	37.2926	50.7066
	65.	6.5773	6.6840	7.2658	10.3846	51.6395	68.8111
	55.	8.1126	8.2717	9.2047	13.2736	64.6437	84.9787
	45.	9.4744	9.6782	10.9356	15.8317	75.9212	98.8730
	35.	10.6207	10.8590	12.1971	18.7586	85.1563	110.2096
	25.	11.5147	11.7710	13.5186	19.5892	92.1148	118.7882
	15.	12.1279	12.3900	14.2806	20.6648	96.6540	124.4837
	5.	12.4399	12.6980	14.6603	21.1784	98.7452	127.2461

TABLE 4-6c

Hydrodynamic Pressure in Infinite Rectangular Reservoirs with
Different Angles of Inclination of a Sloped Bottom - $\frac{\omega}{\omega_1} = 2.4$

Force Part	y	Classical Solution	60 CBE 0°	OUTB1 2.5°	OUTB2 5°	OUTB3 10°	OUTB4 15°
Real Part	95.	0.8992	0.8893	0.8716	0.8522	0.8579	0.7839
	85.	1.5791	1.5877	1.5385	1.4877	1.4808	1.2548
	75.	1.7031	1.7421	1.6701	1.5984	1.5849	1.2291
	65.	1.4855	1.5506	1.4684	1.3906	1.3842	0.9318
	55.	1.0456	1.1300	1.0531	0.9868	1.0069	0.4996
	45.	0.4839	0.5806	0.5257	0.4895	0.5581	0.0421
	35.	-0.1043	-0.0033	-0.0203	-0.0088	0.1298	-0.3488
	25.	-0.6329	-0.5362	-0.5015	-0.4286	-0.2026	-0.6034
	15.	-0.9482	-0.9482	-0.8529	-0.7104	-0.3886	-0.6820
Imaginary Part	5.	-1.2413	-1.2162	-1.0590	-0.8487	-0.4414	-0.6184
	95.	-0.2971	-0.2637	-0.2492	-0.2328	-0.1808	-0.0909
	85.	-0.8841	-0.8647	-0.8211	-0.7703	-0.6118	-0.3177
	75.	-1.4493	-1.4437	-1.3787	-1.3013	-1.0528	-0.5596
	65.	-1.9788	-1.9866	-1.9115	-1.8187	-1.5036	-0.8196
	55.	-2.4595	-2.4790	-2.4071	-2.3121	-1.9580	-1.0947
	45.	-2.8797	-2.9077	-2.8522	-2.7682	-2.4026	-1.3759
	35.	-3.2290	-3.2612	-3.2329	-3.1705	-2.8168	-1.6470
	25.	-3.4988	-3.5298	-3.5347	-3.5001	-3.1739	-1.8859
Absolute Value	15.	-3.6825	-3.7060	-3.7434	-3.7361	-3.4418	-2.0662
	5.	-3.7754	-3.7811	-3.8415	-3.8529	-3.5822	-2.1574
	95.	0.9471	0.9275	0.9065	0.8834	0.8768	0.7891
	85.	1.8097	1.8079	1.7439	1.6753	1.6022	1.2944
	75.	2.2363	2.2626	2.1656	2.0611	1.9027	1.3505
	65.	2.4743	2.5201	2.4104	2.2894	2.0437	1.2409
	55.	2.6725	2.7244	2.6274	2.5139	2.2017	1.2033
	45.	2.9201	2.9651	2.9002	2.8111	2.4714	1.3765
	35.	3.2307	3.2612	3.2330	3.1705	2.8198	1.6835
	25.	3.5556	3.5703	3.5701	3.5262	3.1803	1.9801
	15.	3.8026	3.8739	3.8393	3.8030	3.4637	2.1758
	5.	3.9742	3.9719	3.9848	3.9453	3.6093	2.2443

4.4 Convergence of the Numerical Results with the Size and Number of Boundary Elements

The following three studies relating to the convergence of numerical results are described in this Section: the influence of the size of an element in relation to the excitation frequency, the influence of the number of boundary elements and finally, the influence of the number of one-dimensional finite elements used in modelling the infinite region of the reservoir located beyond the artificial boundary.

4.4.1 Size of Element in Relation to the Excitation Frequency

It is apparent that the length of the element used would significantly influence the accuracy of the results obtained. Considering that the hydrodynamic pressure response is a wave propagation phenomenon in which several waves of different wave lengths are propagating simultaneously, the size of the element should be a function of the wave of smallest wave length so that a constant or linear boundary element can adequately represent the pressure variation. Since the wave length of a harmonic wave is equal to the wave velocity divided by the frequency in cycles per second, the shortest wave length will correspond to the largest frequency. For an earthquake motion the limit of highest frequency excitation is between 50 to 100 cps which for a reservoir height of 100 m corresponds to $\frac{\omega}{\omega_1}$ from 13.89 to 27.78. The associated wave lengths vary from 28.80 to 14.40 m. From a practical point of view limit of significant excitation may be taken as say 40 cps or $\frac{\omega}{\omega_1} = 11.11$ with a wave length of 36 m. For such an excitation frequency, and considering that these should be about 8 to 10 elements for a wave, the length of an element would work out to 4.5 m. It is evident that the influence of the size of element on the accuracy of the numerical computations is an important point of study.

The following test has been carried out to check the influence of the size of boundary elements in relation to the excitation frequency. Two models have been studied: one with 60 constant boundary elements each of the length $l = 10$ m, and the other with 35 constant boundary elements, 25 of which have a length of $l = 20$ m, the remaining 10 elements on the dam face retaining the length of 10 m. The excitation frequencies used in the analysis

are: $90.45 \frac{1}{g}$ and $180.96 \frac{1}{g}$, corresponding to $\frac{\omega}{\omega_1} = 4.0$ and $\frac{\omega}{\omega_1} = 8.0$, respectively and wave lengths of 100 m and 50 m respectively.

The results obtained are presented in Table 4-7 for $\frac{\omega}{\omega_1} = 4.0$ and $\frac{\omega}{\omega_1} = 8.0$. The results of a classical solution are also shown for comparison. The rigid foundation case is considered ($\alpha_R = 1.0$). Figures 4.16a and 4.16b are based on the data presented in Table 4-7 referred to above.

The results presented in Figs. 4.16a and 4.16b show clearly that the accuracy of the results depends on the element size. For $\frac{\omega}{\omega_1} = 4.0$ corresponding to $\lambda = 100$ m, there will be 10 elements 10 m long each within a wave length; a number which can be considered adequate. With 20 m long elements, there are only 5 elements in a wave length and the accuracy of results is bound to suffer. This is borne out by comparing the BEM results in Table 4-7 and Fig. 4.16a with classical results. The results corresponding to element size of 20 m are not too good, although still reasonably close to the exact results.

For $\frac{\omega}{\omega_1} = 8.0$ corresponding to $\lambda = 50$ m, there will be 5, 10 m elements in a wave length. The results obtained with this coarse division are likely to have significant error. With 20 m long elements there are only $2\frac{1}{2}$ elements per wave length. Such division would be unable to represent the present variation within a wave length and the results obtained with such elements would be grossly in error. These conclusions are borne out by the results presented in Table 4.7 and Fig. 4.16b.

In summary, it can be stated that for obtained reasonable accuracy, the length of the largest element in the discretization should be no more than about $\frac{1}{10}$ the smallest wave length of the harmonic component of the excitation.

TABLE 4-7

Effect of the Ratio of Element Length to Wave Length - $\alpha_R = 1.0$

Force Part	y	$\lambda_w = 100 \text{ m}, \frac{w}{w_1} = 4.0$			$\lambda_w = 50 \text{ m}, \frac{w}{w_1} = 8.0$		
		Class. Sol.	60 CBE $10l = \lambda_w$	30 CBE $5l = \lambda_w$	Class. Sol.	60 CBE $5l = \lambda_w$	30 CBE $2.5l = \lambda_w$
Real Part	95.	0.6325	0.6222	0.7974	0.4269	0.5003	0.5228
	85.	0.7796	0.7549	0.8037	0.1658	0.2656	0.1608
	75.	0.4528	0.4447	0.4515	-0.2090	-0.1190	-0.3086
	65.	0.0332	0.0174	0.0450	-0.1749	-0.1378	-0.3568
	55.	-0.3476	-0.3110	-0.2055	0.0953	0.0981	-0.0101
	45.	-0.4758	-0.4208	-0.1975	0.1804	0.2085	0.3281
	35.	-0.3658	-0.2975	0.0529	-0.0062	0.0889	0.3832
	25.	-0.0959	-0.0201	0.4292	-0.1643	-0.0256	0.2526
	15.	0.0005	0.2774	0.7545	-0.0681	0.0625	0.1664
	5.	0.3899	0.4349	0.8021	0.1260	0.1682	0.1228
Imaginary Part	95.	-0.4104	-0.3850	-0.6034	-0.4289	-0.3215	-0.3088
	85.	-1.1741	-1.1757	-1.5734	-1.0325	-0.9614	-0.6548
	75.	-1.7781	-1.7938	-2.2902	-1.1026	-1.0742	-0.8654
	65.	-2.1524	-2.1679	-2.6617	-0.8175	-0.8185	-0.9117
	55.	-2.2731	-2.2772	-2.6751	-0.5951	-0.6446	-0.8235
	45.	-2.1661	-2.1556	-2.3903	-0.6700	-0.7572	-0.6461
	35.	-1.9007	-1.8808	-1.9201	-0.9040	-0.9334	-0.4545
	25.	-1.5726	-1.5547	-1.4001	-0.9989	-0.8793	-0.3554
	15.	-1.2829	-1.2777	-0.9545	-0.8550	-0.6416	-0.4033
	5.	-1.1146	-1.1218	-0.6543	-0.6613	-0.5420	-0.4905
Absolute Value	95.	0.7540	0.7317	1.0000	0.6051	0.5947	0.6072
	85.	1.4094	1.3972	1.7668	1.0457	0.9974	0.6743
	75.	1.8348	1.8481	2.3344	1.1222	1.0808	0.9188
	65.	2.1527	2.1680	2.6621	0.8360	0.8300	0.9790
	55.	2.2995	2.2983	2.6830	0.6027	0.6520	0.8235
	45.	2.2177	2.1963	2.3984	0.6939	0.7854	0.7246
	35.	1.9356	1.9042	1.9208	0.9040	0.9376	0.5945
	25.	1.5755	1.5548	1.4644	1.0123	0.8796	0.4361
	15.	1.2985	1.3075	1.2167	0.8577	0.6446	0.4363
	5.	1.1808	1.2032	1.0351	0.6732	0.5675	0.5057

Fig. 4.16a Effect of the Ratio of Element Length to Wave Length
 60 CBE - $10 l = \lambda_w$ and 30 CBE - $5 l = \lambda_w$

Fig. 4.16b Effect of the Ratio of Element Length to Wave Length
 60 CBE - $5l = \lambda_w$ and 30 CBE - $2.5l = \lambda_w$

4.4.2 The Effect of the Number of Boundary Elements

The results presented in the previous subsection show that the element size should be about $\frac{1}{10}$ the smallest wave length. It needs to be studied whether a finer discretization of the boundary would significantly improve the accuracy. Another aspect that requires consideration is whether the discretization of the far boundary which models the infinite section of the reservoir needs to be finer. These aspects are studied in this subsection with the help of several numerical solutions.

Four different models of the same 2D infinite reservoir are used in the analysis. In this case, the far boundary is situated 200 m from the straight dam face. The first model consists of 60 constant boundary elements and has been already used in Subsection 4.3.2 (see Fig. 4.4a). The other three models are shown in Figs. 4.17a, b and c. The 36-constant boundary element model (Fig. 4.17a) is, in fact, based on the 60-constant boundary element model. The numbers of elements on the dam face and the far end are the same in both models; but the 30 element model has fewer elements on the surface and the reservoir bottom (8 elements only). The 31-constant boundary element model (Fig. 4.17b) is similar to the 36 element model except that in the former there are only 5 elements on the far boundary. Finally, the 30-constant boundary element model is derived from the 31 element model by dropping 1 element from the far boundary (Fig. 4.17c).

The hydrodynamic forces on the dam face due to a horizontal harmonic excitation of the dam were calculated using BEM and four models described above. The results are presented in the following Tables:

Table 4-8a for excitation frequency $\frac{\omega}{\omega_1} = 1.2$ and rigid foundation $\alpha_R = 1.0$,

Table 4-8b for excitation frequency $\frac{\omega}{\omega_1} = 2.4$ and rigid foundation $\alpha_R = 1.0$,

Table 4-8c for excitation frequency $\frac{\omega}{\omega_1} = 1.2$ and flexible foundation $\alpha_R = 0.85$,

Table 4-8d for excitation frequency $\frac{\omega}{\omega_1} = 2.4$ and flexible foundation $\alpha_R = 0.85$.

The results for the rigid case, $\alpha_R = 1.0$ are compared against classical solutions. Very little difference exists in the results of both, real and imaginary parts of the hydrodynamic pressures obtained from the four models. Even with 30 elements in all and only 4 elements on the far boundary the results are quite close to classical solutions.

The behaviour for flexible foundation is similar, the difference between the results

obtained from the four models being quite small. The results tend to confirm the conclusion that the element length of $\leq \frac{\lambda}{10}$ is a sufficient requirement for obtaining reasonable accuracy.

Fig. 4.17a 36 Node Constant Boundary Element Model of an Infinite Reservoir

Fig. 4.17b 31 Node Constant Boundary Element Model of an Infinite Reservoir

Fig. 4.17c 30 Node Constant Boundary Element Model of an Infinite Reservoir

TABLE 4-8a

Hydrodynamic Pressures in Infinite Rectangular Reservoirs
Effect of the Number of Boundary Elements

$$L = 2 H, \frac{\omega}{\omega_1} = 1.2, \alpha_R = 1.0$$

Force Part	y	Classical Solution	60 CBE	36 CBE	31 CBE	30 CBE
Real Part	95.	0.7521	0.7371	0.7376	0.7439	0.7469
	85.	1.1920	1.1529	1.1509	1.1563	1.1550
	75.	1.1809	1.1350	1.1256	1.1282	1.1212
	65.	0.9526	0.8999	0.8822	0.8816	0.8689
	55.	0.6155	0.5562	0.5299	0.5269	0.5093
	45.	0.2409	0.1757	0.1416	0.1374	0.1163
	35.	-0.1185	-0.1884	-0.2291	-0.2322	-0.2548
	25.	-0.4223	-0.4964	-0.5410	-0.5406	-0.5622
	15.	-0.6411	-0.7212	-0.7639	-0.7565	-0.7743
	5.	-0.7554	-0.8728	-0.8842	-0.8654	-0.8754
Imaginary Part	95.	-0.9772	-0.8985	-0.8932	-0.8967	-0.9004
	85.	-2.9077	-2.9047	-2.9149	-2.9249	-2.9345
	75.	-4.7665	-4.8262	-4.8609	-4.8770	-4.8918
	65.	-6.5080	-6.6231	-6.6841	-6.7058	-6.7255
	55.	-8.0892	-8.2530	-8.3390	-8.3659	-8.3900
	45.	-9.4713	-9.6766	-9.7842	-9.8158	-9.8441
	35.	-10.6700	-10.8590	-10.9840	-11.0190	-11.0510
	25.	-11.5070	-11.7710	-11.9070	-11.9470	-11.9820
	15.	-12.1110	-12.3900	-12.5330	-12.5760	-12.6150
	5.	-12.4170	-12.6960	-12.8450	-12.8910	-12.9320
Absolute Value	95.	1.2331	1.1621	1.1584	1.1651	1.1687
	85.	3.1425	3.1251	3.1339	3.1452	3.1536
	75.	4.9106	4.9579	4.9895	5.0058	5.0186
	65.	6.5773	6.6840	6.7421	6.7635	6.7814
	55.	8.1126	8.2717	8.3558	8.3825	8.4054
	45.	9.4744	9.6782	9.7852	9.8168	9.8418
	35.	10.6207	10.8606	10.9864	11.0214	11.0653
	25.	11.5147	11.7815	11.9193	11.9592	11.9952
	15.	12.1279	12.4110	12.5562	12.5987	12.6387
	5.	12.4399	12.7279	12.8754	12.9200	12.9616

TABLE 4-8b

Hydrodynamic Pressures in Infinite Rectangular Reservoirs
Effect of the Number of Boundary Elements

$$L = 2 H, \frac{\omega}{\omega_1} = 2.4, \alpha_R = 1.0$$

Force Part	y	Classical Solution	60 CBE	36 CBE	31 CBE	30 CBE
Real Part	95.	0.8992	0.8893	0.8665	0.8762	0.8833
	85.	1.5791	1.5877	1.5339	1.5634	1.5821
	75.	1.7031	1.7421	1.6599	1.7076	1.7363
	65.	1.4855	1.5506	1.4451	1.5084	1.5453
	55.	1.0456	1.1300	1.0057	1.0813	1.1244
	45.	0.4839	0.5806	0.4407	0.5240	0.5722
	35.	-0.1043	-0.0033	-0.1571	-0.0604	-0.1862
	25.	-0.6329	-0.5362	-0.7035	-0.6142	-0.5635
	15.	-0.9482	-1.1282	-1.1628	-1.0417	-0.9914
	5.	-1.2413	-1.2162	-1.3896	-1.3091	-1.2599
Imaginary Part	95.	-0.2971	-0.2637	-0.2661	-0.2478	-0.2376
	85.	-0.8841	-0.8647	-0.8446	-0.8100	-0.7925
	75.	-1.4493	-1.4437	-1.3963	-1.3526	-1.3323
	65.	-1.9788	-1.9866	-1.9120	-1.8657	-1.8468
	55.	-2.4595	-2.4790	-2.3812	-2.3385	-2.3250
	45.	-2.8797	-2.9077	-2.7937	-2.7601	-2.7553
	35.	-3.2290	-3.2612	-3.1407	-3.1200	-3.1263
	25.	-3.4988	-3.5298	-3.4150	-3.4092	-3.4276
	15.	-3.6825	-3.7060	-3.6128	-3.6214	-3.6514
	5.	-3.7754	-3.7811	-3.7406	-3.7608	-3.8001
Absolute Value	95.	0.9471	0.8893	0.9065	0.9147	0.9147
	85.	1.8097	1.8079	1.7511	1.7695	1.7695
	75.	2.2363	2.2626	2.1691	2.1784	2.0403
	65.	2.4743	2.5201	2.3967	2.3992	2.4080
	55.	2.6725	2.7244	2.5849	2.5764	2.5826
	45.	2.9201	2.9651	2.8282	2.8094	2.8141
	35.	3.2307	3.2612	3.1446	3.1206	3.1318
	25.	3.5556	3.5703	3.4867	3.4641	3.4736
	15.	3.8026	3.8739	3.7953	3.7682	3.7836
	5.	3.9742	3.9719	3.9904	3.9821	4.0035

TABLE 4-8c

Hydrodynamic Pressures in Infinite Rectangular Reservoirs
Effect of Number of Boundary Elements

$$L = 2H, \frac{\omega}{\omega_1} = 1.2, \alpha_R = 0.85$$

Force Part	y	60 CBE	36 CBE	31 CBE	30 CBE
Real Part	95.	0.8950	0.9074	0.9132	0.9158
	85.	1.6621	1.7011	1.7080	1.7081
	75.	1.9621	2.0331	2.0396	2.0359
	65.	2.0290	2.1095	2.1151	2.1076
	55.	1.9290	2.0262	2.0312	2.0203
	45.	1.7333	1.8442	1.8494	1.8361
	35.	1.4864	1.6078	1.6144	1.6000
	25.	1.2218	1.3508	1.3605	1.3467
	15.	0.9626	1.0991	1.1141	1.1030
	5.	0.6983	0.8640	0.8875	0.8819
Imaginary Part	95.	-0.8791	-0.8744	-0.8776	-0.8809
	85.	-2.8420	-2.8518	-2.8638	-2.8745
	75.	-4.7199	-4.7528	-4.7734	-4.7913
	65.	-6.4723	-6.5302	-6.5589	-6.5834
	55.	-8.0573	-8.1388	-8.1748	-8.2054
	45.	-9.4360	-9.5379	-9.5802	-9.6161
	35.	-10.5750	-10.6930	-10.7400	-10.7810
	25.	-11.4470	-11.5760	-11.6280	-11.6710
	15.	-12.0340	-12.1690	-12.2230	-12.2690
	5.	-12.3230	-12.4620	-12.5170	-12.5640
Absolute Value	95.	1.2545	1.2601	1.2665	1.2707
	85.	3.2923	3.3206	3.3345	3.3437
	75.	5.1153	5.1694	5.1909	5.2059
	65.	6.7829	6.8625	6.8915	6.9125
	55.	8.2850	8.3872	8.4234	8.4504
	45.	9.5939	9.7146	9.7571	9.7898
	35.	10.6789	10.8132	10.8606	10.8991
	25.	11.5120	11.6545	11.7073	11.7484
	15.	12.0724	12.2185	12.2737	12.3185
	5.	12.3428	12.4919	12.5484	12.5949

TABLE 4-8d

Hydrodynamic Pressures in Infinite Rectangular Reservoirs
Effect of Number of Boundary Elements

$$L = 2 H, \frac{\omega}{\omega_1} = 2.4, \alpha_R = 0.85$$

Force Part	y	60 CBE	36 CBE	31 CBE	30 CBE
Real Part	95.	0.9114	0.8892	0.8972	0.9025
	85.	1.6585	1.6051	1.6310	1.6452
	75.	1.8550	1.7734	1.8158	1.8377
	65.	1.6953	1.5912	1.6478	1.6757
	55.	1.2925	1.1716	1.2390	1.2710
	45.	0.7438	0.6105	0.6849	0.7193
	35.	0.1408	-0.0017	0.07576	0.1106
	25.	-0.4326	-0.5828	-0.5063	-0.4728
	15.	-0.9074	-1.0640	-0.9918	-0.9603
	5.	-1.2599	-1.4049	-1.3404	-1.3118
Imaginary Part	95.	-0.3002	-0.3058	-0.2893	-0.2787
	85.	-0.9766	-0.9669	-0.9346	-0.9138
	75.	-1.6128	-1.5826	-1.5408	-1.5138
	65.	-2.1855	-2.1339	-2.0885	-2.0591
	55.	-2.6748	-2.6042	-2.5609	-2.5329
	45.	-3.0663	-2.9817	-2.9455	-2.9222
	35.	-3.3523	-3.2609	-3.2356	-3.2194
	25.	-3.5319	-3.5371	-3.4307	-3.4232
	15.	-3.6111	-3.5371	-3.5302	-3.5396
	5.	-3.5999	-3.5690	-3.5812	-3.5903
Absolute Value	95.	0.9596	0.9404	0.9427	0.9446
	85.	1.9247	1.8739	1.8798	1.8819
	75.	2.4581	2.3769	2.3814	2.3809
	65.	2.7659	2.6618	2.6603	2.6548
	55.	2.9707	2.8556	2.8449	2.8198
	45.	3.1552	3.0436	3.0241	3.0094
	35.	3.3555	3.2609	3.2365	3.2213
	25.	3.5583	3.4919	3.4678	3.4557
	15.	3.7234	3.6937	3.6669	3.6675
	5.	3.8140	3.8356	3.8238	3.8224

4.4.3 The Effect of the Number of Finite Elements in Modelling of Radiation Damping

For the excitation frequency greater than ω_1 the numerical results of the analysis of any 2D infinite reservoir depends strongly on the modelling of the radiation damping condition at the far boundary. In the previous subsections, it was shown that the far boundary of the rectangular reservoir, need not be located far from the dam in order to obtain reasonable accuracy. The results presented in Subsection 4.4.2 indicated that even fairly coarse division of the far boundary did not introduce any appreciable error in the hydrodynamic pressure values on the dam face, as long as the element size was not more than about $\frac{1}{10}$ the smallest wave length.

TABLE 4-9

**Comparison of First Four Eigenfrequencies in 2D Case
for Different Finite Element Meshes**

	Class. Sol.	1-d. FEM 2 Elem.	1-d. FEM 4 Elem.	1-d. FEM 8 Elem.	1-d. FEM 10 Elem.
λ_1^2	0.000247	0.000260	0.000250	0.000247	0.000247
λ_2^2	0.002221	0.003169	0.002487	0.002286	0.002262
λ_3^2	0.006168	-	0.008207	0.006678	0.006492
λ_4^2	0.012090	-	0.017163	0.014081	0.013350
ω_1	22.62 $\frac{1}{sec}$	23.20 $\frac{1}{sec}$	22.77 $\frac{1}{sec}$	22.66 $\frac{1}{sec}$	22.64 $\frac{1}{sec}$
ω_2	67.86 $\frac{1}{sec}$	81.06 $\frac{1}{sec}$	68.84 $\frac{1}{sec}$	68.84 $\frac{1}{sec}$	68.49 $\frac{1}{sec}$
ω_3	113.10 $\frac{1}{sec}$	-	130.45 $\frac{1}{sec}$	117.67 $\frac{1}{sec}$	116.02 $\frac{1}{sec}$
ω_4	158.34 $\frac{1}{sec}$	188.65 $\frac{1}{sec}$	170.87 $\frac{1}{sec}$	170.87 $\frac{1}{sec}$	166.38 $\frac{1}{sec}$
Mesh Sketch

To obtain a direct measure of the effect of discretization used on the far boundary of the reservoir, the first four eigenvalues together with the respective eigenfrequencies were calculated, for 4 different one-dimensional meshes. The results obtained are presented in Table 4.9 where the classical solution is also shown for comparison. Even with a 2-element mesh, the difference between the first frequency obtained from the numerical solution and that obtained from a continuum solution is remarkably small. For 8-element and 10-element meshes the differences from the classical results are very small, both in the eigenvalues and the eigenfunctions.

4.5 Comparison of Superposition Method and Boundary Element Method Results in Selected Cases

The boundary element superposition method (BESM) was developed and used in the early stage of this research. Due to the difficulties encountered in the application of BESM, especially in the choice of the location of so-called congruent boundary, this method was later abandoned. More detailed description of the type of difficulties in the application of BESM was presented earlier.

A limited number of results from BESM are presented here for the case of a reservoirs with rigid boundaries. In this cases the BESM results show good agreement with the classical solution.

The finite reservoir model used in the studies is shown in Fig. 4.18. The rigid dam is assumed to undergo a horizontal harmonic motion in the upstream-downstream direction due to a ground acceleration of $e^{i\omega t}$. The hydrodynamic pressure distributions on the upstream face of the straight dam are calculated for two excitation frequencies: $\frac{\omega}{\omega_1} = 0.4$ and $\frac{\omega}{\omega_1} = 1.2$. The results are shown in Figs. 4.19a and 4.19b. The superposition method values were obtained for a congruent boundary location - 40 m outside the reservoir boundary. A change in the distance of the congruent boundary does influence the results, and for irregular geometries where classical solution does not exist the choice of the location of congruent boundary represents a real problem.

Fig. 4.19a Distribution of the Total Hydrodynamic Pressure on a Gravity Dam Face in a Finite Reservoir by SUP - $\frac{\omega}{\omega_1} = 0.4$

Fig. 4.19b Distribution of the Total Hydrodynamic Pressure on a Gravity Dam Face in a Finite Reservoir by SUP - $\frac{\epsilon}{\epsilon_1} = 1.2$

Fig. 4.20a Distribution of the Total Hydrodynamic Pressure on a Gravity Dam Face in an Infinite Reservoir by SUP - $\frac{\omega}{\omega_1} = 0.4$

Fig. 4.20b Distribution of the Real Part of the Total Hydrodynamic Pressure on a Gravity Dam Face in an Infinite Reservoir by SUP - $\frac{\omega}{\omega_1} = 1.2$

Fig. 4.20c Distribution of the Imaginary Part of the Total Hydrodynamic Pressure on a Gravity Dam Face in an Infinite Reservoir by SUP - $\frac{\omega}{\omega_1} = 1.2$

CHAPTER 5

FORMULATION OF THE MATHEMATICAL MODEL FOR A 3D RESERVOIR BOUNDED BY AN ARCH DAM

5.1 General Comments

When the river valley is narrow and surrounded by a natural banks of rock or firm soil and the foundation consists of solid bedrock, an arch dam is usually the preferred choice for impounding the river. In comparison to a concrete gravity dam, the analysis and design of an arch dam is more complex, mainly because of the three-dimensional geometry of the system. Among the many forces that must be considered in the design of an arch dam are the forces caused by an earthquake and the resulting hydrodynamic forces exerted by the reservoir water.

The seismic analysis and design of an arch dam has not been studied in as much detail as of a concrete gravity dam because of the complexity in the dynamic analysis of a three-dimensional system. However, in more recent years the finite element technique has successfully been applied to the analysis of arch dam-reservoir systems (28,38). Also, some preliminary work has been reported on the application of the boundary element technique (56,57). Further work on the analysis of hydrodynamic pressures in a reservoir impounded by an arch dam is reported in this work.

This chapter presents the development of 3D model of the reservoir for analysis by the boundary element technique. The assumptions made in modelling the 3D reservoir are described in detail. It is stated that, for a harmonic excitation the equations governing the reservoir vibration reduce to a three-dimensional wave equation. Boundary conditions applicable to the 3D case are presented and special conditions to account for foundation damping as well as radiation damping in an infinite reservoir are described.

From the formulation presented in this chapter it would be apparent that as in the 2D case BEM reduces the dimensionality of the 3D problem by one. This should result in significant saving in the computational effort involved in the analysis.

5.2 Mathematical Model of System and Ground Motion

An arch dam-reservoir-foundation system consists of the dam, the impounded water reservoir and the foundation together with the banks of the reservoir. Such a system is shown in Fig. 5.1. The arch dam usually has a curved but regular shape, such as, for example cylindrical, conical or paraboloidal thick shell. The ground motion considered in the analysis may be in any of the following three directions:

1. An upstream-downstream (x) direction along the central cross-sectional plane of the dam.
2. The cross-stream (y) direction, which is perpendicular to the above mentioned plane of the dam.
3. The vertical (z) direction.

It is assumed that, even though the ground motion varies with time, at any one instant it is identical throughout the reservoir bottom and banks. As for the 2D case, the dam is assumed to be linear and elastic.

The reservoir may have a short length or may extend in the form of a channel to a large distance from the dam face. In the first case, the reservoir is treated as finite, in the second it is considered infinite for the purpose of analysis.

The analytical studies reported here are restricted to the case of a rigid arch dam; thus the dam-foundation interaction is not considered. Reservoir-foundation interaction is, however, accounted for through a special boundary condition. The foundation and reservoir banks can be rigid or flexible. In the case of rigid foundation and banks, there is no interaction between the dam and the foundation or the water reservoir and the foundation. A flexible foundation will, however, interact with the dam and the reservoir.

Fig. 5.1 An Arch Dam-Reservoir-Foundation System

5.3 Equations of Motion and Boundary Conditions

Assumptions made in the analysis of 3D case are identical to these for the 2D case and are listed below.

1. Water is compressible but non-viscous.
2. Effect of surface waves is assumed to be negligible.
3. Only small amplitude motion is considered.

With these assumptions, the motion of the water in the three-dimensional water reservoir is governed by the following set of Navier-Stokes equations:

$$\frac{\partial^2 u}{\partial t^2} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x} \quad (5.1a)$$

$$\frac{\partial^2 v}{\partial t^2} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial y} \quad (5.1b)$$

$$\frac{\partial^2 w}{\partial t^2} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial z} \quad (5.1c)$$

$$\frac{\partial^2 \bar{p}}{\partial x^2} + \frac{\partial^2 \bar{p}}{\partial y^2} + \frac{\partial^2 \bar{p}}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 \bar{p}}{\partial t^2} \quad (5.1d)$$

in which, x , y and z = the cartesian coordinates as shown in Fig. 5.1; u , v and w = respectively the x , y and z components of the displacement of a particle of water; t = time variable; ρ = mass density of water; $\bar{p}$ = the hydrodynamic pressure; and c = the velocity of sound in water.

If the excitation is harmonic, the hydrodynamic pressure can be represented by $\bar{p} = p(x, y, z, \omega) e^{i\omega t}$ where ω is the exciting frequency. Substitution into Eq. 5.1d gives:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} + k^2 p = 0 \quad (5.2)$$

where $k = \omega/c$ is the wave number.

Equation 5.2 is known as a three-dimensional Helmholtz equation. It is a second order partial differential equation in the 3D space, and belongs to the elliptical class of equations with constant coefficients.

The boundary conditions applicable to a general three-dimensional geometry are as follows:

1. Along the dam-fluid interface

$$\frac{\partial p}{\partial n}(s, r, w) = -\frac{w}{g} a_n(s, r) \quad (5.3)$$

w = specific weight of water; s, r = coordinates over the dam-fluid interface; $a_n(s, r)$ = normal component of dam acceleration; n = outward normal to the interface.

2. Along surface of the reservoir

Since the effect of surface waves is neglected, the hydrodynamic pressure should vanish at the free surface giving

$$p(x, y, H, \omega) = 0 \quad (5.4)$$

3. Along the reservoir bottom

For a rigid type of foundation, there is no interaction between the water and the foundation, and the boundary condition along the reservoir bottom is given by

$$\frac{\partial p}{\partial n}(s', r', \omega) = 0 \quad (5.5)$$

where $s', r' =$ coordinates over the reservoir bottom and n is the outward normal to reservoir-foundation interface.

For a flexible foundation, a boundary condition similar to that used in the 2D case is applied (26,28). This condition is described in the next section.

4. Along the far boundary

For a finite reservoir, the condition at the far end depends on whether that boundary is fixed or free to move. An infinite reservoir is modelled as being comprised of two regions, finite region which may be irregular and an infinite channel of uniform cross-section. In this case the so-called far boundary is located at the interface of the irregular and regular regions. The energy loss due to radiation damping in the infinite region is accounted for through a special boundary condition similar to one used in the 2D case. The derivation of this special condition closely parallels that presented by Hall and Chopra (28) and is presented in the one of subsequent sections. The two-dimensional finite element technique is used to solve an eigenvalue problem associated with small vibrations of an infinite channel. In an independent research, Kandasamy has recently used the two-dimensional boundary element technique to solve a similar eigenvalue problem and then coupled its solution with the FEM analysis of the gravity dam (75).

5.4 Damping due to Flexibility of the Foundation and Banks

The damping due to a flexible foundation and reservoir banks is introduced by a special boundary condition that takes into account, in an approximate manner, the effect of wave absorption. The derivation of this condition for the 3D case is identical to that for the 2D case. It is based on a foundation model consisting of thin columns of infinite length extending outward in a normal direction from the reservoir floor and sides (see Section 2.4 and Appendix B) and the assumption that one-dimensional wave motion is propagated along the length of these columns without interaction with the adjacent columns. The final form of the boundary condition derived on the basis of this assumption can be expressed as

$$\frac{\partial p}{\partial n}(s', r', \omega) = -\frac{w}{g}a_n(s', r') - i\omega\gamma p(s', r', \omega) \quad (5.6)$$

where γ = the damping coefficient (described in Appendix B); $a_n(s', r')$ = the acceleration of the reservoir floor and sides; ω = the excitation frequency.

The effect of foundation damping can be accounted for through a more detailed representation of the foundation soil, i.e. by deriving an impedance matrix for the soil medium using a boundary element technique (60). However, only the simplified treatment of foundation damping as represented by Eq. 5.6 is used in this work.

5.5 Modelling of the Infinite Region of 3D Reservoir

The infinite region of the water reservoir is modelled as an infinite channel of uniform, constant cross-section as shown in Fig. 5.2. The region shown in Fig. 5.2 extends to infinity along the x axis with uniform $y - z$ cross-section.

The 3D Helmholtz equation 5.2 governs the reservoir motion in the infinite channel as well. The boundary conditions are as given below:

Fig. 5.2 Infinite but Regular Region of a 3D Reservoir

- horizontal excitation along x -axis (downstream-upstream motion)

- Along the interface of the finite and infinite region

$$\frac{\partial p}{\partial x}(0, y, z, \omega) = q_f \quad (5.7a)$$

- Along the banks

$$\frac{\partial p}{\partial n}(x, r', \omega) = -i\omega\gamma p(x, r', \omega) \quad (5.7b)$$

- Along the upper surface of the reservoir

$$p(x, H, z, \omega) = 0 \quad (5.7c)$$

- Along the reservoir bottom

$$\frac{\partial p}{\partial n}(x, 0, \omega) = -i\omega\gamma p(x, 0, \omega) \quad (5.7d)$$

- excitation perpendicular to the x -axis (cross-stream motion)

- Along the interface of the finite and infinite region

$$\frac{\partial p}{\partial x}(0, y, z, \omega) = q_f \quad (5.8a)$$

- Along the banks and reservoir bottom

$$\frac{\partial p}{\partial n}(x, r', \omega) = -\frac{w}{g} a_n^i(r') - i\omega\gamma p(x, r', \omega) \quad (5.8b)$$

- Along the upper surface of the reservoir

$$p(x, H, z, \omega) = 0 \quad (5.8c)$$

The term q_f from Eq. 5.7a is the pressure derivative $\frac{\partial p}{\partial n}$ in the finite region at the interface of the finite and infinite regions of the reservoir.

The governing equation 5.2 can be solved by a separation of variables

$$p = p_x p_{yx} \quad (5.9)$$

in which p_x satisfies the equation

$$\frac{d^2 p_x}{dx^2} - \kappa^2 p_x = 0 \quad (5.10)$$

and p_{yx} satisfies the equation

$$\frac{\partial^2 p_{yx}}{\partial y^2} + \lambda^2 p_{yx} = 0 \quad (5.11)$$

where $\lambda^2 = \frac{\omega^2}{c^2} + \kappa^2$

Equation 5.11 is solved by a two-dimensional finite element discretization following procedure similar to that given in reference (26) and included in Appendix E for the sake of completeness.

Such a solution gives

$$[\mathbf{A} + i\omega\gamma \mathbf{B} - \lambda^2 \mathbf{F}] \mathbf{p}_{yx} = \frac{\omega}{g} \mathbf{d} \quad (5.12)$$

where $\mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{d}$ are obtained by assembling the corresponding element matrices shown in Appendix E. As stated earlier $\mathbf{p}_{yx}$ is the vector of the pressure values at the nodes. The surface nodes are excluded from the formulation because the pressure values are known to be zero there. Equation 5.12 is valid for both horizontal excitations (downstream-upstream and cross stream) and also for a vertical excitation, which is represented by the R.H.S. of this equation.

For a purely horizontal excitation Eq. 5.12 reduces to

$$[\mathbf{A} + i\omega\gamma \mathbf{B}] \mathbf{p}_{yx} = \lambda^2 \mathbf{F} \mathbf{p}_{yx} \quad (5.13)$$

Equations 5.12 and 5.13 have the same form as Eqs. 2.20 and 2.29 obtained for a one-dimensional finite element model of the infinite region of a 2D reservoir.

Equation 5.13 represents an eigenvalue problem which for non-zero γ leads to complex eigenvalues and eigenvectors. If the matrix of M eigenvectors obtained from Eq. 5.13 is represented by Ψ , the following relationship is satisfied:

$$\Psi^T [A + i\omega\gamma B] \Psi = \Lambda \quad (5.14)$$

in which Λ = a diagonal matrix of the M eigenvalues $\lambda_1^2, \lambda_2^2, \dots, \lambda_M^2$ (real or complex), and the eigenvectors have been normalized to be orthonormal w.r.t. F so that

$$\Psi^T F \Psi = I \quad (5.15)$$

The solution to Eq. 5.10 can now be written as

$$p_z = e^{-\kappa_n z} \quad (5.16)$$

in which $\kappa_n = \sqrt{\lambda_n^2 - k^2}$.

The general solution for pressure vector p_{yz} has a form

$$p_{yz} = \sum_{n=1}^M \eta_n \psi_n e^{-\kappa_n z} \quad (5.17)$$

or

$$p_{yz} = \Psi E \eta \quad (5.18)$$

in which η = a vector of nodal coordinates and E = is a diagonal matrix of terms $e^{-\kappa z}$. Taking the origin at the far boundary and differentiating Eq. 5.18

$$q_{yz} = \Psi K \eta \quad (5.19)$$

in which K = a diagonal matrix with elements $\kappa_1, \kappa_2, \dots, \kappa_M$.

Except for the subscripts yz , formulae 5.18 and 5.19 are the same as those obtained in the one-dimensional model.

The finite element equations for a vertical excitation obtained in the one-dimensional model are equally applicable to the present two-dimensional model of the infinite channel. The only difference lies in the assembled matrices A , B , F and d .

In the case of vertical excitation, the vector of nodal pressures at the far end is given by

$$p_{ys} = \frac{w}{g} \Psi K^{-1} \Psi^T d \quad (5.20)$$

and the total pressure is obtained by adding Eqs. 5.18 and 5.20

$$p_{ys} = \Psi \eta + \frac{w}{g} \Psi K^{-1} \Psi^T d \quad (5.21)$$

Incorporation of the above derived condition in the three-dimensional boundary element formulation follows lines similar to those in the two-dimensional formulation presented earlier in Chapter 3, since the boundary element formulation given by Eq. 5.21 is exactly the same as that given by Eq. 2.31 (see Section 2.5). However, a modified mesh of the boundary elements at the far boundary is required in order to match the constant boundary element pressure values with the nodal values of pressure in 4-node quadrilateral finite elements. The details will be presented in the next chapter.

CHAPTER 6

CONSTANT BOUNDARY ELEMENT FORMULATION FOR 3D RESERVOIRS

6.1 General Comments

The application of the direct boundary element method (BEM) to the analysis of small vibrations of a 3D water reservoir is described in this chapter. The results presented in Chapter 4 for the 2D case indicate that the response estimates obtained from an equal number of constant elements and linear elements are very similar and the use of linear elements offers no particular advantage in the problem being solved. It is likely that a similar situation will exist in the 3D analysis. Therefore, it was decided to use a simple constant 4-point quadrilateral boundary element in the 3D analysis.

As evident from the presentation of the 2D case, the BEM reduces the dimensionality of the problem by one. This is of course true in the 3D case as well. The fundamental solution for the 3D case is, in fact, simpler than for the 2D case. These factors are likely to make the 3D BEM analysis computationally efficient.

Evaluation of constant boundary integrals is described in detail in this chapter. This is followed by a presentation of the infinite radiation condition and simplified damping condition both of which have been adapted from Hall and Chopra (26,28). Towards the end of the chapter, the boundary element superposition procedure for the analysis of 3D reservoirs of finite extent is introduced (57) and some analytical results obtained from it are presented.

6.2 Formulation of Boundary Integral Equations in 3D Case

The motion of the water in a 3D reservoir is governed by the three dimensional Helmholtz equation also known as the reduced wave equation, given by Eq. 5.2.

As in a two-dimensional case, a weighted residual technique is applied in which the governing equation is satisfied in an average sense over the domain V . This leads to the following equation:

$$\iiint_V \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} + k^2 p \right) p^* dV = 0 \quad (6.1)$$

in which p^* is the assumed weighting function. By application of the Green's theorem to the integrals in Eq. 6.1 the latter can be reduced to the following form (56):

$$\begin{aligned} & \iiint_V \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} + k^2 p \right) p^* dV \\ &= \iiint_V \left(\frac{\partial^2 p^*}{\partial x^2} + \frac{\partial^2 p^*}{\partial y^2} + \frac{\partial^2 p^*}{\partial z^2} + k^2 p^* \right) p dV \\ & - \left[\iint_S p \frac{\partial p^*}{\partial x} l_x dS + \iint_S p \frac{\partial p^*}{\partial y} l_y dS + \iint_S p \frac{\partial p^*}{\partial z} l_z dS \right] \\ & + \left[\iint_S p^* \frac{\partial p}{\partial x} l_x dS + \iint_S p^* \frac{\partial p}{\partial y} l_y dS + \iint_S p^* \frac{\partial p}{\partial z} l_z dS \right] \end{aligned} \quad (6.2)$$

where l_x, l_y, l_z are the direction cosines of boundary surface S of domain V .

Equation 6.2 may be expressed as

$$\begin{aligned} & \iiint_V \left(\frac{\partial^2 p^*}{\partial x^2} + \frac{\partial^2 p^*}{\partial y^2} + \frac{\partial^2 p^*}{\partial z^2} + k^2 p^* \right) p dx dy dz \\ &= \iint_S p \frac{\partial p^*}{\partial n} dS - \iint_S p^* \frac{\partial p}{\partial n} dS \end{aligned} \quad (6.3)$$

The R.H.S. of Eq. 6.3 contains surface integrals over the boundary S only. By an appropriate choice of function p^* , the integration over the domain V appearing on the L.H.S. of Eq. 6.3 can be avoided. The desired weighting function p^* is chosen to satisfy the following equation:

$$\frac{\partial^2 p^*}{\partial x^2} + \frac{\partial^2 p^*}{\partial y^2} + \frac{\partial^2 p^*}{\partial z^2} + k^2 p^* = \Delta^i \quad (6.4)$$

in which Δ^i = a Dirac Delta function centered at point i having coordinates (x_i, y_i, z_i) .

The solution of Eq. 6.4 is called the fundamental solution and will be described later.

Substitution of Eq. 6.4 into Eq. 6.3 gives

$$c_i p_i = \iint_S p q_i^* dS - \iint_S p_i^* q dS \quad (6.6)$$

in which $c_i = 1$ for a point inside the water reservoir, $c_i = \frac{1}{2}$ for a point on the smooth boundary of the water reservoir, $c_i = 0$ for a point outside the water reservoir, p_i = the hydrodynamic pressure at point i , $q = \frac{\partial p}{\partial n}$ and $q^* = \frac{\partial p^*}{\partial n}$.

Equation 6.6 is called the boundary integral equation associated with the given Helmholtz equation in the 3D space. It provides a means of determining the unknown pressure at any point within the three-dimensional domain V of the reservoir in terms of integrals of pressure and pressure derivative values evaluated on the boundary S only. While some of these boundary values are specified, others must be determined before Eq. 6.6 can be used. In the boundary element formulation, Eq. 6.6 itself is used to determine the unknowns on the boundary of the reservoir. To solve this problem, the boundary S of the water domain V is divided into a series of two-dimensional boundary elements. The parameters p and q are assumed to vary in a chosen manner over a given element. The manner in which this variation takes place determines the details of formulation. In this study, a constant boundary element formulation is used.

6.3 Constant Boundary Element Formulation

The constant boundary element formulation for the three-dimensional Helmholtz equation has previously been used by Elkhodary (56) in his analysis of the three-dimensional finite water reservoirs subjected to small harmonic vibrations. The elements used are 4-

point quadrilateral elements. A node is defined at the center of each element. The values p and q are assumed to be constant over each element and equal to their values at the mid-node of the element. The corner points of the element are used to define its geometry. Equation 6.6 for a given point i , on the boundary S has the following discretized form:

$$-\frac{1}{2} p_i + \sum_{j=1}^N p_j \iint_{S_j} q_i^* dS_j = \sum_{j=1}^N q_j \iint_{S_j} p_i^* dS_j \quad (6.7)$$

where p_j and q_j are the pressure gradient, respectively over the j th element and S_j represents the surface of that element.

A constant boundary element model of the three-dimensional water reservoir is presented in Fig. 6.1

Fig. 6.1 Constant Boundary Element Discretization of a 3D Reservoir

The following notations are now introduced in the formulation.

$$H_{ij} = \iint_{S_j} q_i^* dS_j \quad i \neq j \quad (6.8a)$$

$$H_{ii} = -\frac{1}{2} + \iint_{S_i} q_i^* dS_i \quad (6.8b)$$

$$G_{ij} = \iint_{S_j} p_i^* dS_j \quad (6.8c)$$

Using the notations of Eqs. 6.8, Eq. 6.7 may be expressed in the following alternative form:

$$\sum_{j=1}^N p_j H_{ij} = \sum_{j=1}^N q_j G_{ij} \quad (6.9)$$

$$i = 1, 2, \dots, N$$

Corresponding to n values of i , N equations of the type of Eq. 6.9 can be written. Assembly of these n equations gives, in the matrix form

$$\mathbf{H} \mathbf{p} = \mathbf{G} \mathbf{q} \quad (6.10)$$

in which $\mathbf{H}$ and $\mathbf{G}$ are each matrices of size $N \times N$ whose elements H_{ij} and G_{ij} are given by Eqs. 6.8.

Equation 6.7 together with Eqs. 6.8 constitutes the constant boundary element formulation for the three-dimensional Helmholtz equation. Details of the evaluation of the integrals involved in these equations will be presented in one of the following sections.

6.4 Fundamental Solution in 3D Case

The solution of Eq. 6.4 is commonly referred to as the fundamental solution. It can be obtained by so-called Green's function approach (56,66,67) and is given by

$$\begin{aligned}
 p^* &= -\frac{1}{4\pi r} e^{-ikr} \\
 &= -\frac{1}{4\pi r} [\cos kr - i \sin kr] \\
 &= -\frac{\cos kr}{4\pi r} + \frac{\sin kr}{4\pi r} i
 \end{aligned} \tag{6.10}$$

in which r = the radial distance variable given by $r = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$, the point with coordinates (x_i, y_i, z_i) is referred to as the source point.

The fundamental solution given by Eq. 6.10, has been used by several researchers to study small vibrations of the three-dimensional finite reservoirs (56,57). This type of the fundamental solution is acceptable for both infinite and finite reservoirs, because it does not include waves reflected from infinity.

It should be mentioned here that other possibilities exist for the selection of a fundamental solution. For example, Tsai and Lee (76) have used the following fundamental solution which satisfies the free surface condition

$$p'' = \frac{1}{4\pi} \left(\frac{e^{-ikr}}{r} - \frac{e^{-ikr'}}{r'} \right) \tag{6.11}$$

where r' = the distance between the field point and the mirror image point.

Research on the most appropriate fundamental solution for both 2D and 3D cases is still continuing (55,76).

6.5 Evaluation of Constant Boundary Integrals

The constant boundary element integrals in Eqs. 6.8 are evaluated by substituting for p^* and q^* from the fundamental solution given by Eq. 6.10.

The term H_{ij} in Eq. 6.8a is given by

$$\begin{aligned}
H_{ij} &= \iint_{S_j} q_i^* dS_j \\
&= \iint_{S_j} \frac{1}{4\pi r^2} (\cos kr + kr \sin kr) \cos \theta dS_j \\
&\quad + i \iint_{S_j} \frac{1}{4\pi r^2} (-\sin kr + kr \cos kr) \cos \theta dS_j
\end{aligned} \tag{6.12}$$

in which θ = the angle between the outward normal n , and the radius vector r , as shown in Fig. 6.1.

Transformation of Eq. 6.12 into local coordinates system shown in Fig. 6.2 gives

$$\begin{aligned}
H_{ij} &= \int_{-1}^{+1} \int_{-1}^{+1} \frac{\cos \theta}{4\pi r^2} (\cos kr + kr \sin kr) |J| d\xi_1 d\xi_2 \\
&\quad + i \int_{-1}^{+1} \int_{-1}^{+1} \frac{\cos \theta}{4\pi r^2} (-\sin kr + kr \cos kr) |J| d\xi_1 d\xi_2
\end{aligned} \tag{6.13}$$

Using a Gauss quadrature formula for calculations of both terms results in

$$H_{ij} = \frac{1}{4\pi} \sum_{n=1}^4 \sum_{m=1}^4 \left(\frac{\cos \theta}{r^2} \right)_{nm} \left[(\cos kr + kr \sin kr) + i(-\sin kr + kr \cos kr) \right]_{nm} |J| \omega_{nm} \tag{6.14}$$

where ω_{nm} = the weighting function used in the Gauss quadrature; $|J|$ = the Jacobian of transformation between (x, y, z) coordinates system and (ξ_1, ξ_2) coordinates system.

When $i = j$, H_{ii} is given by Eq. 6.8b. Since the outward normal is perpendicular to the element, $\theta = \frac{\pi}{2}$ and $\cos \theta = 0$. This causes the second term in Eq. 6.8b to vanish and the expression for H_{ii} reduces to

$$H_{ii} = -\frac{1}{2} \tag{6.15}$$

The term G_{ij} is also obtained by numerical integration and is given by

$$G_{ij} = -\frac{1}{4\pi} \sum_{n=1}^4 \sum_{m=1}^4 \left[(\cos kr + i \sin kr) \frac{\cos \theta}{r} \right]_{nm} |J| \omega_{nm} \tag{6.16}$$

Fig. 6.2 Collapsed Quadrilateral Element

A special procedure for calculating G_{ii} is adapted from Elkhodary (56), in which the concept of a so-called collapsed quadrilateral element is used.

In a region containing the singular point and an infinitesimal circle around it, the value of G_{ii} can be shown to be zero. The evaluation of G_{ii} over the rest of element is achieved by the use of the above mentioned collapsed quadrilateral elements (Fig. 6.2).

If N_i is the assumed shape function, the respective partial derivatives w.r.t. η_1 and ξ_1 , are

$$\frac{\partial N_i}{\partial \eta_1} = \frac{\partial N_i}{\partial \xi_1} \frac{\partial \xi_1}{\partial \eta_1} + \frac{\partial N_i}{\partial \xi_2} \frac{\partial \xi_2}{\partial \eta_1} \quad (6.17a)$$

$$\frac{\partial N_i}{\partial \xi_1} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi_1} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi_1} + \frac{\partial N_i}{\partial z} \frac{\partial z}{\partial \xi_1} \quad (6.17b)$$

Similarly $\frac{\partial N_i}{\partial \xi_2}$ could be written. Two Jacobians must be calculated. $|J|$ is the Jacobian relating the global coordinates system (x, y, z) and the local coordinates system (ξ_1, ξ_2) . $|G|$ is the Jacobian relating the two local systems of coordinates (ξ_1, ξ_2) and (η_1, η_2) .

The respective transformation relationships from η system to ξ system are

$$\xi_1 = \psi_1 \xi_{1(1)} + \psi_2 \xi_{1(2)} \quad (6.18a)$$

$$\xi_2 = \psi_1 \xi_{2(1)} + \psi_2 \xi_{2(2)} \quad (6.18b)$$

where

$$\psi_1 = \frac{1}{4} (1 + \eta_1) (1 - \eta_2) \quad (6.19a)$$

$$\psi_2 = \frac{1}{4} (1 - \eta_1) (1 + \eta_2) \quad (6.19b)$$

and $\xi_{1(1)}$ and $\xi_{1(2)}$ are the same coordinates of points (1) and (2) respectively. Points (3) and (4) are chosen to lay always at the origin of ξ_1, ξ_2 system of coordinates, i.e. $\xi_{(3)} = \xi_{(4)} = 0$.

The Jacobian matrix of transformation, G_t is given by

$$G_t = \begin{pmatrix} \frac{\partial \psi_1}{\partial \eta_1} & \frac{\partial \psi_2}{\partial \eta_1} \\ \frac{\partial \psi_1}{\partial \eta_2} & \frac{\partial \psi_2}{\partial \eta_2} \end{pmatrix} \begin{pmatrix} \xi_{1(1)} & \xi_{2(1)} \\ \xi_{1(2)} & \xi_{2(2)} \end{pmatrix} \quad (6.20)$$

The Jacobian $|G|$ relating the two local systems of coordinates (ξ_1, ξ_2) and (η_1, η_2) can be obtained from

$$|G| = \det G_t \quad (6.21)$$

The transformation relationships from ξ system of coordinates to the global coordinates system (x, y, z) are

$$x = \phi_1 x_{(1)} + \phi_2 x_{(2)} + \phi_3 x_{(3)} + \phi_4 x_{(4)} \quad (6.22a)$$

$$y = \phi_1 y_{(1)} + \phi_2 y_{(2)} + \phi_3 y_{(3)} + \phi_4 y_{(4)} \quad (6.22b)$$

$$z = \phi_1 z_{(1)} + \phi_2 z_{(2)} + \phi_3 z_{(3)} + \phi_4 z_{(4)} \quad (6.22c)$$

where the shape functions are given by

$$\phi_1 = \frac{1}{4} (1 + \xi_1) (1 - \xi_2) \quad (6.23a)$$

$$\phi_2 = \frac{1}{4} (1 + \xi_1) (1 + \xi_2) \quad (6.23b)$$

$$\phi_3 = \frac{1}{4} (1 - \xi_1) (1 + \xi_2) \quad (6.23c)$$

$$\phi_4 = \frac{1}{4} (1 - \xi_1) (1 - \xi_2) \quad (6.23d)$$

The Jacobian $|J|$ relating the global system of coordinates (x, y, z) to the local system (ξ_1, ξ_2) can be obtained from

$$|J| = \sqrt{J_1^2 + J_2^2 + J_3^2} \quad (6.24)$$

where

$$J_1 = \frac{\partial y}{\partial \xi_1} \frac{\partial z}{\partial \xi_2} - \frac{\partial z}{\partial \xi_1} \frac{\partial y}{\partial \xi_2} \quad (6.25a)$$

$$J_2 = \frac{\partial z}{\partial \xi_1} \frac{\partial x}{\partial \xi_2} - \frac{\partial x}{\partial \xi_1} \frac{\partial z}{\partial \xi_2} \quad (6.25b)$$

$$J_3 = \frac{\partial x}{\partial \xi_1} \frac{\partial y}{\partial \xi_2} - \frac{\partial y}{\partial \xi_1} \frac{\partial x}{\partial \xi_2} \quad (6.25a)$$

6.6 Incorporation of Infinite Radiation Condition and Foundation and Banks Damping.

The derivation of the eigenvalue problem associated with a finite element model of the infinite region of a 3D reservoir was presented in detail in Section 5.5. The special boundary condition to account for the damping provided by the flexible foundation was presented in Section 5.4. The incorporation of these two conditions in the three-dimensional BEM formulation is described in this section.

To match the solutions at the interface of the finite and infinite region, and to apply the condition along the reservoir bottom and banks Eq. 6.10 is expressed in a partitioned form as follows:

$$\begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad (6.26)$$

in which p_1 , q_1 represent, respectively, the pressures and pressure derivatives at the nodes on the reservoir surface and along the dam face, p_2 , q_2 represent the same quantities on the reservoir bottom and banks while p_3 , q_3 represent them at the nodes on the far boundary.

As discussed in Section 5.5, the pressure gradient on the flexible bottom and banks is given by

$$q_2 = -\frac{\omega}{g} a_n - i \omega \gamma p_2 \quad (6.27)$$

in which a_n = a vector which describes the acceleration components along the normal to the reservoir bottom or the bank and the second term accounts for a simplified version of foundation damping effect.

The following two expressions represent the infinite radiation condition in the 3D case. The expressions are identical to those previously obtained for the 2D case.

$$p_3 = \Psi \eta + \frac{\omega}{g} \Psi K^{-1} \Psi^T d \quad (6.28a)$$

$$q_3 = \Psi K \eta \quad (6.28b)$$

where Ψ, K, η are the same quantities as described in Section 5.5. The second term in p_3 represents the effect of the vertical excitation along the infinite section. When this excitation is absent, d is zero and the corresponding term vanishes. Eigenvectors Ψ are real valued if there is no foundation damping. When damping is present Ψ is complex valued.

When a constant boundary element formulation is used in the 3D infinite case, difficulty arises in the evaluation of p_3 and q_3 . Values of pressure and pressure derivatives at the interface of the finite and infinite region obtained by a finite element formulation of the infinite region must match those of the far boundary of the finite region. Difficulty arises because while the finite element nodes are located at the vertices of a 4-node quadrilateral element, the constant boundary element nodes are at the geometric centers of these elements. One possible manner in which this difficulty can be resolved is to define different set of elements for the boundary element discretization and the finite element discretization. By proper selection of the element boundaries it is possible to match most of the nodes in

the two formulations. An example of such a mesh division is shown in Fig. 6.3. Nodes 6, 7, 10 and 11 are identical in two nodes. Nodes 1 through 4, 5, 8, 9, 12 and 13 through 16 do not match exactly, but the p and q values for the BEM and FEM sets are assumed to be identical. Also the pressure at nodes 4, 8, 12 and 16 is assumed to be zero in both cases.

Fig. 6.3 Regular (FEM) and Modified (BEM) Meshes at Far Boundary

With the boundary element and finite elements on the far boundary matched as discussed, substitution of Eqs. 6.23 and 6.22 into Eqs. 6.21 gives

$$\begin{pmatrix} H_{11} & H_{12} + i\omega\gamma G_{12} & H_{13}\Psi + G_{13}\Psi K \\ H_{21} & H_{22} + i\omega\gamma G_{22} & H_{23}\Psi + G_{23}\Psi K \\ H_{31} & H_{32} + i\omega\gamma G_{32} & H_{33}\Psi + G_{33}\Psi K \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_2 \\ \eta \end{pmatrix} = \begin{pmatrix} G_{11}q_1 - \frac{\omega}{g}G_{12}a_n - \frac{\omega}{g}H_{13}\Psi K^{-2}\Psi^T d \\ G_{21}q_1 - \frac{\omega}{g}G_{22}a_n - \frac{\omega}{g}H_{23}\Psi K^{-2}\Psi^T d \\ G_{31}q_1 - \frac{\omega}{g}G_{32}a_n - \frac{\omega}{g}H_{33}\Psi K^{-2}\Psi^T d \end{pmatrix} \quad (6.29)$$

Equations 6.29 are similar to those obtained in the 2D case, the difference lies in the computation of individual terms in the matrices. Equations 6.29 which represent the constant boundary element formulation for the 3D case, including infinite radiation condition

and foundation damping contain N unknown values of p and q . After suitable rearrangement these equations can be solved for the unknowns.

6.7 Boundary Element Superposition Formulation for Finite 3D Reservoirs

The boundary element superposition method (BESM) was introduced earlier for the 2D case. The difficulties encountered with the BESM were discussed at the end of Chapter 3 (Section 3.8). Similar difficulties arise in the use of the BESM for the 3D case. However, results of the research concerning the use of this method for finite 3D water reservoirs are presented here in brief (57).

In applying the BESM it is assumed that the three-dimensional domain $\mathcal{F}$ under consideration is embedded in an infinite domain (Fig. 6.4). Since solutions of the governing equation for infinite domain are easily available, they can be used to find pressures and pressure derivatives due to any prescribed sources within the domain. These solutions will, however, violate the conditions on the reservoir boundary. In order to rectify this situation, it is assumed that a number of additional point sources exist in the infinite domain but external to $\mathcal{F}$.

The strengths of these sources can be selected so that the boundary conditions on S are satisfied. Then the superposition of infinite solutions due to internal and external sources will be equivalent to the solution for finite domain and only internal sources.

In practical solution, similar to the 2D case, boundary conditions are satisfied at a selected number of points say N along the boundary S . The additional sources of unknown strengths are located along the boundary external to the domain. The external boundary is referred to as the congruent boundary (Fig. 6.4). The number of unknown sources is also N . The congruent boundary is generally taken to be parallel to the real boundary and located at an arbitrary distance D from it.

Referring to Fig. 6.5, the pressure p^* , and the pressure derivative $\frac{\partial p^*}{\partial n}$, arising at point i due to a point source of strength Q_j located on the congruent boundary, are given by

Fig. 6.4 A Finite 3D Domain Embedded in an Infinite Domain

$$p^* = Q_j G(kr) \quad (6.30)$$

$$\frac{\partial p^*}{\partial n} = Q_j G'(kr) \cos \theta \quad (6.31)$$

where n = outward normal, θ = the angle between the radius vector r and the normal n , and $G'(kr)$ is the derivative of $G(kr)$ w.r.t. the argument kr . The function $G(kr)$ is referred to be the fundamental solution of the governing Helmholtz equation.

The fundamental solution selected is

$$G(kr) = -\frac{1}{4\pi r} e^{-ikr} \quad (6.32)$$

A boundary element discretization of the three-dimensional water reservoir is represented in Fig. 6.5.

The total hydrodynamic pressure and pressure derivative at point i are obtained by summing over the point sources

$$p_i = \sum_j \frac{Q_j}{4} \left[-\frac{\cos kr_{ij}}{\pi r_{ij}} + i \frac{\sin kr_{ij}}{\pi r_{ij}} \right] \quad (6.33)$$

Fig. 6.5 An Arch Dam Storing a Finite Water Reservoir
- Boundary Element Superposition Analysis

$$\frac{\partial p_i}{\partial n} = \sum_j \frac{Q_j}{4} \left[\frac{\cos kr_{ij} + k r_{ij} \sin kr_{ij}}{\pi r^2} + i \frac{-\sin kr_{ij} + k r_{ij} \cos kr_{ij}}{\pi r^2} \right] \cos \theta \quad (6.34)$$

By satisfying the given mixed boundary valued problem at a number of selected points on reservoir boundary, set of simultaneous equations in Q_j is obtained. These equations are solved for the unknown source strengths Q_j . Once the source strengths Q_j are determined, they may be used to obtain the pressure and pressure derivative at any point within the reservoir or on its surface.

The complete set of equations can be expressed as

$$C Q = b \quad (6.35)$$

where

$b^T = [b_1 \ b_2 \ b_3 \ \dots \ b_N] =$ prescribed boundary conditions at N selected points

$Q^T = [Q_1 \ Q_2 \ Q_3 \ \dots \ Q_N] =$ unknown point source strengths

$C = N \times N$ influence matrix

When pressure is prescribed, $b_i = p_i$ and $C_{ij} = G(kr_{ij})$. When pressure derivative is prescribed, $b_i = \frac{\partial p_i}{\partial n}$ and $C_{ij} = G'(kr_{ij}) \cos \theta$.

Equations 6.35 constitute the boundary element superposition formulation in the case of the 3D finite water reservoir.

As discussed for the 2D case, the results obtained from the BESM are found to be sensitive to the location of the congruent boundary. This introduces another level of uncertainty in the analysis and the BESM loses its advantage. Research on the method was therefore not pursued any further. Some numerical results obtained from the BESM for the 3D case are, however, presented at the end of the next chapter.

CHAPTER 7

HYDRODYNAMIC PRESSURES IN 3D RESERVOIRS

7.1 General Comments

The formulations of direct boundary element and boundary element superposition methods for the analysis of hydrodynamic pressures in a 3D finite and infinite reservoir were presented in Chapters 5 and 6. For the numerical implementation of the analysis procedure a series of computer programs were developed. This chapter presents the results of analytical studies of the hydrodynamic pressures in 3D reservoirs subjected to an upstream-downstream, cross stream and vertical harmonic motion. At the beginning of this chapter, results relating to 3D finite reservoirs impounded by a straight gravity dam or an arch dam are presented and compared with available classical results. This is followed by the presentation of the study of 3D rectangular, infinite reservoirs. The objective of this latter study is to verify that the results from the analysis of a 3D infinite rectangular reservoir for upstream-downstream and vertical motion are compatible with the results from a 2D infinite rectangular reservoir. Such a comparison provides an independent check on the accuracy and validity of the computer algorithms and the boundary element formulation used. Next, the numerical procedure is used to analyze an infinite 3D reservoir impounded by an arch dam. Convergence behaviour and the manner in which it is affected by the number of finite elements used to model the radiation condition for an infinite channel, are examined. Finally, a comparison of the results obtained by boundary element superposition method and boundary element method results, is presented for a 3D finite reservoir only.

7.2 Hydrodynamic Pressures in 3D Finite Reservoirs by BEM

As in the analysis of the 2D cases, the major task of this research was to develop a BEM solution procedure for calculating the hydrodynamic pressures in a general 3D water reservoir that included the effect of infinite radiation and foundation and bank damping. Some selected results relating to finite reservoirs are presented here for sake of completeness.

In order to be able to compare the results with known classical solutions only reservoirs of regular geometry are studied. At first, a 3D finite rectangular reservoir is considered (Fig. 7.1). The constant boundary element model of this reservoir is shown in Fig. 7.2.

Fig. 7.1 A 3D Rectangular Finite Reservoir

For an upstream-downstream or vertical ground motion that does not vary along the y direction the results from the analysis of this 3D reservoir should be identical within discretization errors to those obtained for a 2D reservoirs in the $x - z$ plane having similar dimensions.

A 3D reservoir impounded by a simple arch dam is studied (Fig. 7.3) next. The constant boundary element model of that reservoir is shown in Fig. 7.4.

Fig. 7.2 The Constant Boundary Element Model of 3D Rectangular Finite Reservoir - $N = 119$ elements

Fig. 7.3 A 3D Finite Reservoir Impounded by an Arch Dam

The following data is used in the computation of the results presented for both dams: velocity of sound in water, $c = 1440 \text{ m/s}$; the mass density of water, $\rho = 1 \frac{\text{tonne}}{\text{m}^3}$. The geometric data is included in Figs. 7.1, 7.2, 7.3 and 7.4.

Fig. 7.4a The Constant Boundary Element Model of a 3D
Finite Reservoir - Reservoir's Top and Bottom
 $N = 143$ Constant Boundary Elements

The variation of hydrodynamic pressure along the straight dam face for the reservoir described in Figs. 7.1 and 7.2 due to a unit harmonic acceleration of the dam face in upstream-downstream direction, for exciting frequencies given by $\omega/\omega_1 = 0.4, 1.2, 2.0$ and a coefficient of reflection, $\alpha_R = 1.0$ (rigid foundation and banks case), are plotted in Figs. 7.5a, 7.5b and 7.5c respectively. The results from the 3D BEM analysis are compared with 2D classical solutions and also 2D BEM results. The results presented here show the relatively good agreement with 2D classical solutions as well with 2D BEM results obtained for constant boundary element formulation. This comparison provides a partial verification of the validity of the formulation and the computer algorithm. The variation of hydrodynamic pressure acting on the simple arch dam face (Figs. 7.3 and 7.4), for the exciting frequencies used earlier, due to a unit harmonic excitation in upstream-downstream direction and a rigid foundation and banks, are plotted in Figs. 7.6a, 7.6b and 7.6c. Classical solutions are available only for a radially infinite reservoir. As expected they are not the same as those for a finite reservoir. For $\omega < \omega_1$, however, finite and infinite reservoir values are similar, as is the case for $\omega = 0.4 \omega_1$.

As was found in the 2D case, the pressure values are significantly affected by the geometry of the reservoir. In absence of damping, the pressure values are entirely in phase with excitation frequency and the imaginary part should not exist.

Fig. 7.4b The Constant Boundary Element Model of a 3D Finite Reservoir

Reservoir's Banks and Dam Face

$N = 143$ Constant Boundary Elements

Fig. 7.5a Distribution of the Total Hydrodynamic Pressure on a Gravity Dam
 due to Upstream-Downstream Harmonic Motion
 in a 3D Finite Reservoir - $\frac{\omega}{\omega_1} = 0.4$

Fig. 7.5b Distribution of the Total Hydrodynamic Pressure on a Gravity Dam
 due to Upstream-Downstream Harmonic Motion
 in a 3D Finite Reservoir - $\frac{\varepsilon}{\varepsilon_1} = 1.2$

Fig. 7.5c Distribution of the Total Hydrodynamic Pressure on a Gravity Dam
 due to Upstream-Downstream Harmonic Motion
 in a 3D Finite Reservoir - $\frac{w}{w_1} = 2.0$

Fig. 7.6a Absolute Value of the Total Hydrodynamic Pressure on an Arch Dam
due to Upstream-Downstream Harmonic Motion
in a 3D Finite Reservoir - $\frac{w}{\epsilon_1} = 0.4$

Fig. 7.6b Absolute Value of the Total Hydrodynamic Pressure on an Arch Dam
due to Upstream-Downstream Harmonic Motion
in a 3D Finite Reservoir - $\frac{\omega}{\omega_1} = 1.2$

Fig. 7.6c Absolute Value of the Total Hydrodynamic Pressure on an Arch Dam
due to Upstream-Downstream Harmonic Motion
in a 3D Finite Reservoir - $\frac{\omega}{\omega_1} = 2.0$

7.3 Hydrodynamic Pressures in 3D Infinite Reservoirs by BEM

As stated earlier in the formulation of boundary element method for the analysis of 3D reservoirs, an infinite reservoir is represented by a finite region of regular or irregular geometry, as the case may be, and an infinite channel of constant cross section. The infinite region is modelled by two-dimensional finite element discretization. The modelling leads to an eigenvalue problem which is solved, as in the 2D case, by means of the QZ algorithm originally developed by Moler and Stewart (74).

7.3.1 Effects of Infinite Radiation and Foundation Damping on the Hydrodynamic Pressure in 3D Infinite Rectangular Reservoirs

Fig. 7.7 A 3D Rectangular Infinite Reservoir

Selected results of analytical studies carried out on infinite reservoirs, are presented here. To be able to obtain some verification of the formulation and the computer algorithm, a rectangular reservoir shown in Fig. 7.7 is considered first. The following data is used in the computation of the results shown: height of the reservoir, $H = 100 \text{ m}$; velocity of sound in water, $c = 1440 \text{ m/s}$; the mass density of water, $\rho = 1 \frac{\text{tonne}}{\text{m}^3}$. The far end is placed at a distance of 100 m from the dam face.

The constant boundary element model of the reservoir being studied is shown in Fig. 7.8.

Fig. 7.8a The Constant Boundary Element Model of
a 3D Infinite Rectangular Reservoir
Dam, Banks and the Far End

**Fig. 7.8b The Constant Boundary Element Model of
a 3D Infinite Rectangular Reservoir
Reservoir's Top and Bottom**

The model consists of 119 boundary elements. The finite element model of the far boundary is presented in Fig. 7.9. In this model, 15 four-node linear isoparametric quadrilateral elements are used to represent the infinite channel.

Fig. 7.9 The Finite Element Model of the Far End

The results presented here show the variations of hydrodynamic pressure along the dam face for the two types of harmonic motion: upstream-downstream and vertical motion. Four different foundation damping levels are considered with the following values of coefficient of reflection: $\alpha_R = 0.0, 0.5, 0.75$ and 1.0 . The excitation frequencies have been chosen to be: $\omega/\omega_1 = 0.4, 1.2$ and 2.0 . The results for upstream-downstream harmonic motion are presented in Figs. 7.10 throughout 7.11 and for the vertical harmonic motion in Figs. 7.12 throughout 7.13, respectively. The real and imaginary parts of the hydrodynamic pressure together with their respective absolute values are shown separately. The 2D classical results and 2D BEM results for $\alpha_R = 1.0$ are also shown for comparison (for horizontal and vertical motions). The results show very good agreement with classical results. As stated earlier, for upstream-downstream and vertical motion that do not vary in the cross-stream direction, the 3D rectangular reservoir is, in fact, identical to a 2D case. Good comparison between the foregoing 3D results and the 2D classical and BEM results provides a partial verification of the formulation and computer algorithm.

Fig. 7.10a Real Part of the Hydrodynamic Pressure on a Gravity
Dam due to Upstream-Downstream Harmonic Motion
in a 3D Infinite Rectangular Reservoir

$$-\frac{\epsilon_1}{\epsilon} = 0.4$$

Fig. 7.10b Imaginary Part of the Hydrodynamic Pressure on a Gravity Dam due to Upstream-Downstream Harmonic Motion in a 3D Infinite Rectangular Reservoir

$$- \frac{\epsilon_1}{\epsilon_1} = 0.4$$

Fig. 7.10c Absolute Value of the Hydrodynamic Pressure on a Gravity
Dam due to Upstream-Downstream Harmonic Motion
in a 3D Infinite Rectangular Reservoir

$$\frac{\omega}{\omega_1} = 0.4$$

Fig. 7.11a Real Part of the Hydrodynamic Pressure on a Gravity
Dam due to Upstream-Downstream Harmonic Motion
in a 3D Infinite Rectangular Reservoir

$$- \frac{\omega}{\omega_1} = 1.2$$

Fig. 7.11b Imaginary Part of the Hydrodynamic Pressure on a Gravity
Dam due to Upstream-Downstream Harmonic Motion
in a 3D Infinite Rectangular Reservoir

$$-\frac{\omega}{\omega_1} = 1.2$$

Fig. 7.11c Absolute Value of the Hydrodynamic Pressure on a Gravity Dam due to Upstream-Downstream Harmonic Motion in a 3D Infinite Rectangular Reservoir

$$\frac{\omega}{\omega_1} = 1.2$$

Fig. 7.12a Real Part of the Hydrodynamic Pressure on a Gravity
Dam due to Vertical Harmonic Motion
in a 3D Infinite Rectangular Reservoir

$$- \frac{\omega}{\omega_1} = 0.4$$

Fig. 7.12b Imaginary Part of the Hydrodynamic Pressure on a Gravity
Dam due to Vertical Harmonic Motion
in a 3D Infinite Rectangular Reservoir

$$-\frac{\epsilon}{\epsilon_1} = 0.4$$

Fig. 7.12c Absolute Value of the Hydrodynamic Pressure on a Gravity Dam due to Vertical Harmonic Motion in a 3D Infinite Rectangular Reservoir

$$\frac{w}{\omega_1} = 0.4$$

Fig. 7.13a Real Part of the Hydrodynamic Pressure on a Gravity Dam due to Vertical Harmonic Motion in a 3D Infinite Rectangular Reservoir

$$-\frac{\epsilon_1}{c_1} = 1.2$$

Fig. 7.13b Imaginary Part of the Hydrodynamic Pressure on a Gravity Dam due to Vertical Harmonic Motion in a 3D Infinite Rectangular Reservoir

$$\frac{\omega}{\omega_1} = 1.2$$

Fig. 7.13c Absolute Value of the Hydrodynamic Pressure on a Gravity Dam due to Vertical Harmonic Motion in a 3D Infinite Rectangular Reservoir

$$- \frac{\omega}{\omega_1} = 1.2$$

7.3.2 Effects of Infinite Radiation and Foundation Damping on the Hydrodynamic Pressure in 3D Infinite Reservoirs Impounded by an Arch Dam

The effects of infinite radiation and foundation damping on the hydrodynamic pressures in a 3D reservoir impounded by an arch dam is presented here. The reservoir model used in the study is shown in Fig. 7.19. The following data is used in the computation: height of the reservoir, $H = 100 \text{ m}$; radius of the circular section of the arch dam, $R = 90 \text{ m}$; length of the section, $L = 110 \text{ m}$; velocity of sound in water, $c = 1440 \text{ m/s}$; the mass density of water, $\rho = 1 \frac{\text{tonne}}{\text{m}^3}$. The far end is placed at the distance equal to projection of $L = 110 \text{ m}$ on the axis of symmetry of the dam and reservoir.

The constant boundary element model of the reservoir studied is shown in Fig. 7.19. The model consists of 260 four-node constant quadrilateral boundary elements. Also, 30 four-node linear isoparametric quadrilateral finite elements are used to model the infinite channel (Fig. 7.20).

The results presented here show the variations of hydrodynamic pressure along the dam face for three types of harmonic motion: upstream-downstream, cross-stream and vertical motion. Two different values of coefficient of reflection $\alpha_R = 0.75$ and $\alpha_R = 1.0$ are considered. The excitation frequencies selected in the analysis are $\omega/\omega_1 = 0.4$, 1.2 and 2.0. The results are presented in three set of Tables: Tables 7-1 for $\omega/\omega_1 = 0.4$, Tables 7-2 for $\omega/\omega_1 = 1.2$ and Tables 7-3 for $\omega/\omega_1 = 2.0$. Graphical representations of the pressures due to upstream-downstream motion are shown in Figs. 7.21, 7.22 and 7.23. The absolute values of the hydrodynamic pressures are presented. The 3D classical results obtained by Porter and Chopra (37,77) for a radially infinite reservoir are also shown for comparison. The classical results should be compared with the BEM results for no foundation damping or $\alpha_R = 1.0$. Most of the results show a qualitative similarity in two sets of response, however significant numerical differences exist presumably because of the difference in the geometry of the infinite part.

**Fig. 7.19a The Constant Boundary Element Model
of a 3D Reservoir Impounded by
an Arch Dam - Reservoir's Top**

Fig. 7.19b The Constant Boundary Element Model
of a 3D Reservoir Impounded by
an Arch Dam - Reservoir's Bottom

3

MICROD

Fig. 7.19c The Constant Boundary Element Model of a 3D Reservoir Impounded by an Arch Dam - Dam, Banks and the Far End

Fig. 7.20 The Finite Element Model of the Far End

The results of the analysis for cross-stream motion are presented in three sets of Tables: Tables 7-4 for $\omega/\omega_1 = 0.4$, Tables 7-5 for $\omega/\omega_1 = 1.2$ and Tables 7-6 for $\omega/\omega_1 = 2.0$. Graphical representations are shown in Figs. 7.24, 7.25 and 7.26. The results of BEM for the upstream-downstream and cross-stream motions are different from the classical infinite case reported by Porter and Chopra (77). The geometry of the infinite part of the reservoir seems to have significant effect on the results.

Finally the hydrodynamic pressures due to a vertical harmonic motion are presented in Tables: Tables 7-7 for $\omega/\omega_1 = 0.4$, Tables 7-8 for $\omega/\omega_1 = 1.2$ and Tables 7-9 for $\omega/\omega_1 = 2.0$. Figures 7.27, 7.28 and 7.29 show graphical representation of the results. The results of BEM for the vertical motion are much similar to those obtained for the radially infinite reservoir (77).

In both types of motions referred to above two different coefficients of reflection are used: $\alpha_R = 1.0$ (rigid case) and $\alpha_R = 0.75$ (flexible foundation case with 75% of waves are freely reflected at the foundation and the banks of the reservoir). Generally, the foundation and banks damping reduce the response, especially around the resonance regions.

TABLE 7-1a

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Upstream-Downstream Motion**

$$-\frac{\omega}{\omega_1} = 0.4 \text{ at } 0^\circ$$

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	1.3541	1.3502	real part = absolute value
	0.7	2.7905	2.7785	
	0.5	3.6879	3.6681	
	0.3	4.2330	4.2058	
	0.1	4.4484	4.4414	
Imaginary Part	0.9	very small	-0.0347	none
	0.7		-0.1079	
	0.5		-0.1785	
	0.3		-0.1785	
	0.1		-0.3138	
Absolute Value	0.9	1.3541	1.3506	1.31
	0.7	2.7905	2.7806	2.84
	0.5	3.6224	3.6697	3.71
	0.3	4.2330	4.2130	4.15
	0.1	4.4484	4.4525	4.48

TABLE 7-1b

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Upstream-Downstream Motion**

$$-\frac{w}{\omega_1} = 0.4 \text{ at } 45^\circ$$

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	1.6113	1.6044	real part = absolute value
	0.7	3.5700	3.5488	
	0.5	4.8073	4.7723	
	0.3	5.5449	5.4970	
	0.1	5.8467	5.7877	
Imaginary Part	0.9	very small	-0.0716	none
	0.7		-0.2116	
	0.5		-0.3367	
	0.3		-0.4462	
	0.1		-0.5423	
Absolute Value	0.9	1.6113	1.6060	1.97
	0.7	3.5700	3.5551	4.15
	0.5	4.8073	4.7842	5.35
	0.3	5.5449	5.5151	6.12
	0.1	5.8467	5.8130	6.55

TABLE 7-2a

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Upstream-Downstream Motion**

$$- \frac{\varepsilon}{\varepsilon_1} = 1.2 \text{ at } 0^\circ$$

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.4422	0.8432	not available
	0.7	-0.0326	1.1692	
	0.5	-0.7904	1.1018	
	0.3	-1.4774	0.6530	
	0.1	-1.9191	0.1838	
Imaginary Part	0.9	-1.4605	-1.1258	not available
	0.7	-4.4533	-3.4105	
	0.5	-6.9897	-5.3318	
	0.3	-8.8300	-6.7069	
	0.1	-9.7784	-7.3974	
Absolute Value	0.9	1.5260	1.4066	1.31
	0.7	4.4534	3.6053	3.49
	0.5	7.0342	5.4444	5.24
	0.3	8.9527	6.7386	6.77
	0.1	9.9649	7.3997	7.64

TABLE 7-2b

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Upstream-Downstream Motion**

$$- \frac{u}{\varepsilon_1} = 1.2 \text{ at } 45^\circ$$

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.9312	1.1520	not available
	0.7	1.4358	2.0917	
	0.5	1.3759	2.3289	
	0.3	1.1334	2.1749	
	0.1	0.9108	1.7936	
Imaginary Part	0.9	-1.5742	-1.2678	not available
	0.7	-4.7696	-3.8042	
	0.5	-7.4905	-5.9389	
	0.3	-9.4656	-7.4752	
	0.1	-10.4910	-8.2637	
Absolute Value	0.9	1.8290	1.7130	1.52
	0.7	4.9810	4.3413	4.04
	0.5	7.6158	6.3413	6.00
	0.3	9.5332	7.7852	7.34
	0.1	10.5305	8.4756	8.19

TABLE 7-3a

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Upstream-Downstream Motion**

$$-\frac{\epsilon}{\epsilon_1} = 2.0 \text{ at } 0^\circ$$

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	1.4458	1.0397	not available
	0.7	2.9437	1.7200	
	0.5	3.7640	1.8275	
	0.3	4.1693	1.6785	
	0.1	4.2924	1.4450	
Imaginary Part	0.9	-0.1057	-0.2105	not available
	0.7	-0.3320	-0.6173	
	0.5	-0.5262	-0.9408	
	0.3	-0.6718	-1.1629	
	0.1	-0.7544	-1.2985	
Absolute Value	0.9	1.4497	1.0608	0.87
	0.7	2.9624	1.8274	1.53
	0.5	3.8006	2.0554	1.79
	0.3	4.8231	2.0419	1.86
	0.1	4.3582	1.9427	1.75

TABLE 7-3b

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Upstream-Downstream Motion**

$$-\frac{u}{\omega_1} = 2.0 \text{ at } 45^\circ$$

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.2687	0.6880	not available
	0.7	-0.6441	0.6207	
	0.5	-2.0480	-0.0977	
	0.3	-3.3596	-0.9783	
	0.1	-4.1820	-1.6872	
Imaginary Part	0.9	-0.4252	-0.2105	not available
	0.7	-1.2625	-0.6173	
	0.5	-1.9664	-0.9408	
	0.3	-2.4690	-1.1629	
	0.1	-2.7222	-1.2985	
Absolute Value	0.9	0.5030	0.9156	1.09
	0.7	1.4173	1.8615	2.18
	0.5	2.8006	2.7855	3.23
	0.3	4.1693	3.2158	3.93
	0.1	4.9899	3.5721	4.59

Fig. 7.21 Absolute Value of the Hydrodynamic Pressure on an Arch Dam due to Upstream-Downstream Harmonic Motion in a 3D Infinite Reservoir

$\frac{\epsilon}{\epsilon_1} = 0.4$

Fig. 7.22 Absolute Value of the Hydrodynamic Pressure on an Arch Dam due to Upstream-Downstream Harmonic Motion in a 3D Infinite Reservoir

$-\frac{z}{H} = 1.2$

Fig. 7.23 Absolute Value of the Hydrodynamic Pressure on an Arch Dam due to Upstream-Downstream Harmonic Motion in a 3D Infinite Reservoir

$$-\frac{\epsilon}{\epsilon_1} = 2.0$$

TABLE 7-4a

Hydrodynamic Pressure on an Arch Dam
 Impounding an Infinite Reservoir
 Cross-Stream Motion - $\frac{u}{c_1} = 0.4$
 at 22.5°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.6131	0.6109	real part = absolute value
	0.7	1.3473	1.3405	
	0.5	1.8336	1.8224	
	0.3	2.1365	2.1211	
	0.1	2.2650	2.2457	
Imaginary Part	0.9	very small	-0.0212	none
	0.7		-0.0662	
	0.5		-0.1086	
	0.3		-0.1481	
	0.1		-0.1852	
Absolute Value	0.9	0.6131	0.6113	0.25
	0.7	1.3473	1.3421	0.38
	0.5	1.8336	1.8367	0.35
	0.3	2.1365	2.1263	0.28
	0.1	2.2650	2.2533	0.25

TABLE 7-4b
Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Cross-Stream Motion - $\frac{u}{c_1} = 0.4$
at 45°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	1.2948	1.2905	real part = absolute value
	0.7	2.7780	2.7647	
	0.5	3.6706	3.6490	
	0.3	4.1824	4.1530	
	0.1	4.3800	4.3436	
Imaginary Part	0.9	very small	-0.0489	none
	0.7		-0.1435	
	0.5		-0.2268	
	0.3		-0.2993	
	0.1		-0.3642	
Absolute Value	0.9	1.2948	1.2914	0.25
	0.7	2.7780	2.7684	0.57
	0.5	3.6706	3.6560	0.78
	0.3	4.1824	4.1638	0.97
	0.1	4.3800	4.3588	1.12

TABLE 7-5a

Hydrodynamic Pressure on an Arch Dam
 Impounding an Infinite Reservoir
 Cross-Stream Motion - $\frac{\omega}{\epsilon_1} = 1.2$
 at 22.5°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.6740	0.6721	not available
	0.7	1.5165	1.4962	
	0.5	2.0742	1.9966	
	0.3	2.4122	2.2196	
	0.1	2.5459	2.1732	
Imaginary Part	0.9	-0.1153	-0.6764	not available
	0.7	-3.4500	-2.0540	
	0.5	-5.4871	-3.2326	
	0.3	-6.9321	-4.1140	
	0.1	-7.6886	-4.6218	
Absolute Value	0.9	0.6838	0.9535	1.16
	0.7	3.8141	2.5412	2.85
	0.5	5.8661	3.7995	4.22
	0.3	7.3398	4.9316	5.17
	0.1	8.0991	5.1072	5.80

TABLE 7-5b

Hydrodynamic Pressure on an Arch Dam
 Impounding an Infinite Reservoir
 Cross-Stream Motion - $\frac{\omega}{\omega_1} = 1.2$
 at 45°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	1.4705	1.3713	not available
	0.7	3.2655	2.9519	
	0.5	4.8649	3.8211	
	0.3	4.9965	4.1798	
	0.1	5.2370	4.1119	
Imaginary Part	0.9	-1.6148	-1.0176	not available
	0.7	-4.8899	-3.0551	
	0.5	-7.6805	-4.7921	
	0.3	-9.7081	-6.0833	
	0.1	-10.7680	-6.8327	
Absolute Value	0.9	2.1840	1.7075	1.26
	0.7	5.8770	4.2482	3.31
	0.5	8.8342	6.1290	5.38
	0.3	10.9184	7.3809	6.53
	0.1	11.9740	7.9745	7.17

TABLE 7-6a

Hydrodynamic Pressure on an Arch Dam
 Impounding an Infinite Reservoir
 Cross-Stream Motion - $\frac{u}{\omega_1} = 2.0$
 at 22.5°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	-0.0149	0.0947	not available
	0.7	-0.5972	-0.2644	
	0.5	-1.2907	-0.7749	
	0.3	-1.8892	-1.2520	
	0.1	-2.2497	-1.5668	
Imaginary Part	0.9	-0.0781	-0.1349	not available
	0.7	-0.2385	-0.3962	
	0.5	-0.3708	-0.5646	
	0.3	-0.4623	-0.6100	
	0.1	-0.5022	-0.5324	
Absolute Value	0.9	0.0795	0.1648	0.74
	0.7	0.6431	0.4763	1.42
	0.5	1.3429	0.9877	2.06
	0.3	1.9449	1.3927	2.32
	0.1	2.3012	1.6548	2.53

TABLE 7-6b

Hydrodynamic Pressure on an Arch Dam
 Impounding an Infinite Reservoir
 Cross-Stream Motion - $\frac{\omega}{\omega_1} = 2.0$
 at 45°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.7543	0.7820	not available
	0.7	1.0116	1.1025	
	0.5	0.6990	0.8231	
	0.3	0.2261	0.3306	
	0.1	-0.1340	-0.1224	
Imaginary Part	0.9	-0.4074	-0.4484	not available
	0.7	-1.2109	-1.2953	
	0.5	-1.8897	-1.9226	
	0.3	-2.3803	-2.2826	
	0.1	-2.6416	-2.3997	
Absolute Value	0.9	0.8573	0.9014	0.16
	0.7	1.5778	1.7010	0.27
	0.5	2.0148	2.0913	0.37
	0.3	2.3910	2.3064	0.40
	0.1	2.3947	2.4028	0.42

Fig. 7.24 Absolute Value of the Hydrodynamic Pressure on an Arch Dam due to Cross-Stream Harmonic Motion in a 3D Infinite Reservoir

$$\frac{\epsilon_1}{\epsilon_2} = 0.4$$

Fig. 7.25 Absolute Value of the Hydrodynamic Pressure on an Arch
Dam due to Cross-Stream Harmonic Motion
in a 3D Infinite Reservoir
- $\frac{\omega}{\omega_1} = 1.2$

Fig. 7.26 Absolute Value of the Hydrodynamic Pressure on an Arch
Dam due to Cross-Stream Harmonic Motion
in a 3D Infinite Reservoir
- $\frac{\epsilon}{\epsilon_1} = 2.0$

TABLE 7-7a
Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Vertical Motion - $\frac{u}{u_1} = 0.4$
at 0°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.6980	0.6875	real part = absolute value
	0.7	2.1902	2.1574	
	0.5	3.6442	3.5903	
	0.3	5.0434	4.9699	
	0.1	6.3546	6.3649	
Imaginary Part	0.9	very small	-0.0810	none
	0.7		-0.2546	
	0.5		-0.4214	
	0.3		-0.5778	
	0.1		-0.7192	
Absolute Value	0.9	0.6980	0.6922	0.70
	0.7	2.1902	2.1724	2.16
	0.5	3.6442	3.6150	3.51
	0.3	5.0434	5.0034	4.87
	0.1	6.3546	6.3060	6.55

TABLE 7-7b
Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Vertical Motion - $\frac{u}{u_1} = 0.4$
at 45°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	0.6969	0.6810	real part = absolute value
	0.7	2.1821	2.1334	
	0.5	3.6340	3.5544	
	0.3	5.0232	4.9161	
	0.1	6.3038	6.3824	
Imaginary Part	0.9	very small	-0.1093	none
	0.7		-0.3394	
	0.5		-0.5574	
	0.3		-0.7529	
	0.1		-0.9130	
Absolute Value	0.9	0.6969	0.6897	0.70
	0.7	2.1821	2.1602	2.16
	0.5	3.6340	3.5978	3.51
	0.3	5.0232	4.9734	4.87
	0.1	6.3038	6.2412	6.55

TABLE 7-8a

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir**

Vertical Motion - $\frac{\omega}{\omega_1} = 1.2$

at 0°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	-2.1118	-1.4572	not available
	0.7	-6.3662	-4.4069	
	0.5	-9.7351	-6.7379	
	0.3	-11.7800	-8.1432	
	0.1	-12.2100	-8.4229	
Imaginary Part	0.9	0.0207	-1.0981	not available
	0.7	0.0927	-3.3011	
	0.5	0.1579	-5.0814	
	0.3	0.2093	-6.2288	
	0.1	0.2409	-6.5972	
Absolute Value	0.9	2.1119	1.8246	1.83
	0.7	6.3669	5.5082	5.14
	0.5	9.7364	8.4392	7.71
	0.3	11.7828	10.2523	9.26
	0.1	12.2124	10.6990	9.34

TABLE 7-8b

Hydrodynamic Pressure on an Arch Dam

Impounding an Infinite Reservoir

Vertical Motion - $\frac{\omega}{\omega_1} = 1.2$ at 45°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	-2.1336	-1.5178	not available
	0.7	-6.4077	-4.5664	
	0.5	-9.8144	-6.9779	
	0.3	-11.8910	-8.4087	
	0.1	-12.3590	-8.6583	
Imaginary Part	0.9	0.0145	-0.7688	not available
	0.7	0.0875	-2.3010	
	0.5	0.1594	-3.5272	
	0.3	0.2174	-4.2863	
	0.1	0.2520	-4.4671	
Absolute Value	0.9	2.1336	1.7014	1.83
	0.7	6.4083	5.1134	5.14
	0.5	9.8157	7.8187	7.71
	0.3	11.8930	9.4381	9.26
	0.1	12.3616	9.7427	9.34

TABLE 7-9a

**Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir**

Vertical Motion - $\frac{\omega}{\omega_1} = 2.0$

at 0°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	-0.5817	-0.5771	not available
	0.7	-1.6140	-1.5996	
	0.5	-2.0551	-2.0301	
	0.3	-1.7559	-1.7191	
	0.1	-0.8363	-0.7864	
Imaginary Part	0.9	-0.0011	0.0103	not available
	0.7	0.0086	0.0400	
	0.5	0.0158	0.0620	
	0.3	0.0171	0.0860	
	0.1	0.0112	0.1051	
Absolute Value	0.9	0.5817	0.5772	0.61
	0.7	1.6140	1.6001	1.83
	0.5	2.0552	2.0310	1.92
	0.3	1.7560	1.7706	1.57
	0.1	0.8364	0.7934	0.61

TABLE 7-9b

Hydrodynamic Pressure on an Arch Dam
Impounding an Infinite Reservoir
Vertical Motion - $\frac{\omega}{\omega_1} = 2.0$
at 45°

Force Part	z/H	3D BEM $\alpha_R = 1.0$	3D BEM $\alpha_R = 0.75$	3D Inf. Class.
Real Part	0.9	-0.5250	-0.5081	not available
	0.7	-1.4395	-1.3941	
	0.5	-1.7907	-1.7281	
	0.3	-1.4364	-1.3680	
	0.1	-0.5145	-0.4453	
Imaginary Part	0.9	0.0205	0.0482	not available
	0.7	0.0725	0.1316	
	0.5	0.1161	0.1607	
	0.3	0.1435	0.1318	
	0.1	0.1506	0.0804	
Absolute Value	0.9	0.5254	0.5104	0.61
	0.7	1.4413	1.4003	1.83
	0.5	1.7945	1.7356	1.92
	0.3	1.4436	1.3743	1.57
	0.1	0.5361	0.4525	0.61

Fig. 7.27 Absolute Value of the Hydrodynamic Pressure on an Arch
Dam due to Vertical Harmonic Motion
in a 3D Infinite Reservoir
- $\frac{\omega}{\omega_1} = 0.4$

Fig. 7.28 Absolute Value of the Hydrodynamic Pressure on an Arch Dam due to Vertical Harmonic Motion in a 3D Infinite Reservoir

$$\frac{\epsilon}{\epsilon_1} = 1.2$$

Fig. 7.29 Absolute Value of the Hydrodynamic Pressure on an Arch
Dam due to Vertical Harmonic Motion
in a 3D Infinite Reservoir
- $\frac{\omega}{\omega_1} = 2.0$

7.4 Convergence of the Numerical Results with Size and the Number of Boundary Elements in 3D Models

The following studies relating to the convergence of numerical results are described briefly in this section: the influence of the size of an element in relation to the excitation frequency, the effect of the number of boundary elements, and the effect of the number of two-dimensional finite elements used in modelling the infinite channel located beyond the transmitting boundary.

7.4.1 Effect of the Size of the Element in Relation to the Excitation Frequency

It is apparent that as for a 2D case, the size of the elements used in a 3D model would significantly influence the accuracy of the numerical results obtained. As evident from the discussion of this subject for the 2D case (Subsection 4.4.1), the size of the element should be a fraction of the smallest wave length so that a constant or linear boundary element can adequately represent the pressure variation in the reservoir. The wave phenomenon aspects of the hydrodynamic motion of the reservoir, has been already covered in Chapter 4 and will not be repeated here. However, in the 3D case, a new measure for representing the size of a two-dimensional constant boundary element is required. The proposed measure, adopted here, is the size of the larger diagonal of each quadrilateral element.

According to Subsection 4.4.1, the limit of the highest possible excitation frequency is between 50 and 100 cps, which for a reservoir height of 100 m corresponds to $\frac{\omega}{\omega_1}$ from 13.89 to 27.78. The associated wave length will vary from 28.80 m to 14.40 m. From a practical point of view the limit of significant excitation frequency due to an earthquake motion may be taken as say 40 cps with a wave length of 36.0 m. For such an excitation frequency, and considering that there should be about 8 to 10 elements in one wave length, the diagonal of an element should be no more than 4.4 m. The largest element used in the 3D rectangular reservoir model has a diagonal equal to 28.28 m. Thus, the analysis is expected to give reasonable results up to the wave length $\lambda_w = 282.80$ m, which corresponds to frequency ratio $\omega/\omega_1 = 1.4$. In comparison, the analysis presented in Section 7.3 has covered $\omega/\omega_1 = 0.4, 1.2$ and 2.0.

Fig. 7.30a The Constant Boundary Element Model of a 3D Infinite Rectangular Reservoir - $N = 158$ Elements - Dam, Banks and Far End

The results corresponding to the excitation frequencies: $\omega/\omega_1 = 0.4, 1.2, 2.0, 2.6$ and 3.2 are presented in Tables 7-10, 7-11, 7-12, 7-13 and 7-14. They can be compared with results obtained using a 3D model with less number of elements and also with appropriate classical solutions.

The results obtained from a model with higher number of elements, $N = 158$, do not show an improvement when compared to those obtained from a model with $N = 119$, because the increase from 119 to 158 is not substantial enough.

TABLE 7-10

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
Comparison of Two Models (119 Elements and 158 Elements)
Upstream-Downstream Motion - $\frac{u}{\omega_1} = 0.4$

Force Part	z/H	BEM 119 Elem. $\alpha_R = 1.0$	BEM 119 Elem. $\alpha_R = 0.75$	BEM 158 Elem. $\alpha_R = 1.0$	BEM 158 Elem. $\alpha_R = 0.75$	2D Class. Sol. $\alpha_R = 1.0$
Real Part	0.9	2.3738	2.3533	2.3001	2.2802	2.4217
	0.7	4.9281	4.8646	4.9006	4.8379	5.0997
	0.5	6.5285	6.4259	6.5173	6.4157	6.7580
	0.3	7.4915	7.3559	7.4933	7.3588	7.7695
	0.1	7.8671	7.7063	7.9122	7.7527	8.2540
Imaginary Part	0.9		-0.1168		-0.1142	
	0.7	very	-0.3597	very	-0.3569	none
	0.5	small	-0.5776	small	-0.5747	
	0.3		-0.7611		-0.7580	
	0.1		-0.9080		-0.9051	
Absolute Value	0.9		2.3562		2.2830	
	0.7	the same	4.8779	the same	4.8510	the same
	0.5	as real	6.4518	as real	6.4414	as real
	0.3		7.3952		7.3977	
	0.1		7.7596		7.8053	

TABLE 7-11

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
 Comparison of Two Models (119 Elements and 158 Elements)
 Upstream-Downstream Motion - $\frac{u}{c_1} = 1.2$

Force Part	z/H	BEM 119 Elem. $\alpha_R = 1.0$	BEM 119 Elem. $\alpha_R = 0.75$	BEM 158 Elem. $\alpha_R = 1.0$	BEM 158 Elem. $\alpha_R = 0.75$	2D Class. Sol. $\alpha_R = 1.0$
Real Part	0.9	1.0817	1.8982	1.0554	1.8379	1.0556
	0.7	0.9051	3.3854	0.9802	3.3861	1.0859
	0.5	0.1053	3.3925	0.2436	3.3962	0.4292
	0.3	-0.7241	3.9524	-0.5387	4.0044	-0.2794
	0.1	-1.3081	3.5923	-1.0603	3.6962	-0.7122
Imaginary Part	0.9	-1.8886	-1.2657	-1.8722	-1.2431	-1.9485
	0.7	-5.7468	-3.8517	-5.7798	-3.8391	-5.6547
	0.5	-9.0317	-6.0459	-9.1072	-6.0420	-8.8074
	0.3	-11.4210	-7.6458	-11.5270	-7.6500	-11.0980
	0.1	-12.6668	-8.5205	-12.7980	-8.5379	-12.3020
Absolute Value	0.9	2.1764	2.2815	2.1492	2.2188	2.2161
	0.7	5.8176	5.1280	5.8623	5.1190	5.7580
	0.5	9.0323	7.2123	9.1104	6.9311	8.8178
	0.3	11.4439	8.6070	11.5396	8.6347	11.1063
	0.1	12.7342	9.2468	12.8418	9.3036	12.3226

TABLE 7-12

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
Comparison of Two Models (119 Elements and 158 Elements)
Upstream-Downstream Motion - $\frac{\omega}{\omega_1} = 2.0$

Force Part	s/H	BEM 119 Elem. $\alpha_R = 1.0$	BEM 119 Elem. $\alpha_R = 0.75$	BEM 158 Elem. $\alpha_R = 1.0$	BEM 158 Elem. $\alpha_R = 0.75$	2D Class. Sol. $\alpha_R = 1.0$
Real Part	0.9	1.2867	1.4456	1.2234	1.3811	1.1819
	0.7	1.4103	1.8815	1.3914	1.8646	1.3248
	0.5	0.6861	1.3729	0.6847	1.3750	0.5760
	0.3	-0.2140	0.5317	-0.1977	0.5512	-0.3258
	0.1	-0.8871	-0.2830	-0.8215	-0.2163	-0.9048
Imaginary Part	0.9	-0.6926	-0.7611	-0.6729	-0.7436	-0.7462
	0.7	-2.0941	-2.2579	-2.0784	-2.2500	-2.1657
	0.5	-3.2809	-3.4082	-3.2712	-3.4097	-3.3731
	0.3	-4.1393	-4.1007	-4.1352	-4.1109	-4.2504
	0.1	-4.5794	-4.3291	-4.5893	-4.3594	-4.7116
Absolute Value	0.9	1.4613	1.6337	1.3962	1.5686	1.3977
	0.7	2.5247	2.9391	2.5011	2.9222	2.5388
	0.5	3.3519	3.6743	3.3421	3.6765	3.4219
	0.3	4.1448	4.1350	4.1399	4.1477	4.2629
	0.1	4.6645	4.3383	4.6622	4.3648	4.7977

TABLE 7-13

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
Comparison of Two Models (119 Elements and 158 Elements)
Upstream-Downstream Motion - $\frac{\omega}{\omega_1} = 2.6$

Force Part	z/H	BEM 119 Elem. $\alpha_R = 1.0$	BEM 119 Elem. $\alpha_R = 0.75$	BEM 158 Elem. $\alpha_R = 1.0$	BEM 158 Elem. $\alpha_R = 0.75$	2D Class. Sol. $\alpha_R = 1.0$
Real Part	0.9	1.5362	1.5778	1.4714	1.5074	1.4851
	0.7	1.9103	2.0745	1.9123	2.0582	1.9412
	0.5	0.9516	1.2965	0.9981	1.3074	0.9850
	0.3	-0.4698	0.0398	-0.3807	0.0746	-0.4350
	0.1	-1.5556	-1.0415	-1.4062	-0.9625	-1.4307
Imaginary Part	0.9	-0.5199	-0.7520	-0.4996	-0.7296	-0.5386
	0.7	-1.5612	-2.1195	-1.5503	-2.1099	-1.5630
	0.5	-2.4267	-2.9027	-2.4338	-2.9123	-2.4344
	0.3	-3.0325	-3.0637	-3.0606	-3.0954	-3.0675
	0.1	-3.3201	-2.8849	-3.3777	-2.9018	-3.4003
Absolute Value	0.9	1.6218	1.7478	1.5539	1.6747	1.5797
	0.7	2.4671	2.9658	2.4618	2.9474	2.4922
	0.5	2.6066	3.1791	2.6305	3.1923	2.6261
	0.3	3.0687	3.0639	3.0842	3.0963	3.0982
	0.1	3.6665	2.8852	3.6587	3.0573	3.6890

TABLE 7-14

Hydrodynamic Pressure in Infinite Rectangular Reservoirs
Comparison of Two Models (119 Elements and 158 Elements)
Upstream-Downstream Motion - $\frac{\omega}{\omega_1} = 3.2$

Force Part	z/H	BEM 119 Elem. $\alpha_R = 1.0$	BEM 119 Elem. $\alpha_R = 0.75$	BEM 158 Elem. $\alpha_R = 1.0$	BEM 158 Elem. $\alpha_R = 0.75$	2D Class. Sol. $\alpha_R = 1.0$
Real Part	0.9	0.6945	1.2151	0.6513	1.1531	0.6911
	0.7	-0.1169	1.1053	-0.0898	1.1027	0.1498
	0.5	-0.5635	0.4081	-0.5226	0.4147	-0.3503
	0.3	-0.1273	-0.0862	-0.0940	-0.0886	-0.1711
	0.1	0.4309	-0.3263	-0.4810	-0.3168	0.2365
Imaginary Part	0.9	-1.1546	-1.2348	-1.1187	-1.2041	-1.5479
	0.7	-2.8974	-3.1130	-3.0473	-3.1133	-3.6765
	0.5	-3.0612	-3.3001	-3.2230	-3.3327	-3.6706
	0.3	-1.9919	-2.1219	-2.0776	-2.1793	-2.0351
	0.1	-0.9119	-0.8449	-0.9343	-0.9296	-0.4813
Absolute Value	0.9	1.3474	1.7324	1.3539	1.6672	1.6952
	0.7	2.8997	3.3034	3.0486	3.3028	3.6796
	0.5	3.1126	3.3252	3.2661	3.3584	3.6873
	0.3	1.9960	2.1236	2.0797	2.1811	2.0423
	0.1	1.0086	0.9057	1.0508	0.9821	0.5363

7.4.2 Effect of the Number of Finite Elements in Modelling of Radiation Damping

For excitation frequencies greater than ω_1 , the numerical results of any 2D infinite reservoir depends strongly on the modelling of radiation damping condition. This problem is slightly more complicated in the 3D case, when the far end does not constitute a single line like in the 2D case, but is a uniform cross section of an infinite channel.

To obtain a direct measure of the effect of discretization used on the far end of the reservoir, the first four eigenvalues together with the respective eigenfrequencies were calculated for four different two-dimensional finite element meshes of the simple rectangular cross section of the infinite channel. The results obtained are presented in Table 7-15 where the classical solution is also shown for comparison. The classical solution for a rectangular cross section of the infinite channel was presented by Fok and Chopra (38). The steady improvement of results with an increase of the number of two-dimensional elements is clearly seen. Even for very rough mesh of 4 elements only, the results are quite good.

The effect of the chosen finite element mesh of the far boundary of the reservoir is also apparent from an independent study. The hydrodynamic pressures due to a harmonic upstream-downstream motion calculated for three different meshes are listed in Table 7-16. The classical solution is also presented for comparison. From these results, it is evident that the variation of the hydrodynamic pressure on the dam face depends upon the modelling of an infinite channel. Better results are obtained for more refined mesh, however even the 9-element mesh seems adequate.

TABLE 7-15

Comparison of First Four Eigenfrequencies in 3D Case
for Different Finite Element Meshes

	Class. Sol.	2-d. FEM 4 Elem.	2-d. FEM 16 Elem.	2-d. FEM 64 Elem.
λ_{11}^s	0.000685	0.000721	0.000694	0.000688
λ_{11}^a	0.003426	0.004054	0.003579	0.003465
λ_{21}^s	0.006143	no match	0.006909	0.006349
λ_{22}^s	0.008885	0.008802	0.009794	0.009126
ω_{11}^s	$37.66 \frac{1}{sec}$	$38.67 \frac{1}{sec}$	$37.97 \frac{1}{sec}$	$37.78 \frac{1}{sec}$
ω_{11}^a	$84.29 \frac{1}{sec}$	$91.69 \frac{1}{sec}$	$86.15 \frac{1}{sec}$	$84.76 \frac{1}{sec}$
ω_{21}^s	$112.87 \frac{1}{sec}$	-	$119.69 \frac{1}{sec}$	$114.74 \frac{1}{sec}$
ω_{22}^s	$135.73 \frac{1}{sec}$	$135.10 \frac{1}{sec}$	$142.51 \frac{1}{sec}$	$137.56 \frac{1}{sec}$
Mesh Type

TABLE 7-16

Effect of Number of Elements Modelling Far Boundary
on Hydrodynamic Pressures - Upstream-Downstream Motion

$$\frac{\varepsilon}{\varepsilon_1} = 2.0$$

Force Part	z/H	BEM 119 Elem. $\alpha_R = 1.0$	BEM 107 Elem. $\alpha_R = 1.0$	BEM 104 Elem. $\alpha_R = 1.0$	2D Class. Sol.
Real Part	0.9	1.2867	1.2940	1.3072	1.1819
	0.7	1.4103	1.4320	1.4710	1.3248
	0.5	0.6861	0.7190	0.7769	0.5757
	0.3	-0.2140	-0.1736	-0.1042	-0.3258
	0.1	-0.8871	-0.8427	-0.7679	-0.9048
Imaginary Part	0.9	-0.6926	-0.6847	-0.6665	-0.7462
	0.7	-2.0941	-2.0694	-2.0130	-2.1657
	0.5	-3.2809	-3.2395	-3.1465	-3.3731
	0.3	-4.1393	-4.0833	-3.9607	-4.2504
	0.1	-4.5794	-4.5153	-4.3764	-4.7116
Absolute Value	0.9	1.4613	1.4640	1.4673	1.3977
	0.7	2.5247	2.5165	2.4932	2.5388
	0.5	3.3519	3.3183	3.2410	3.4219
	0.3	4.1448	4.0870	3.9621	4.2628
	0.1	4.6645	4.5933	4.4433	4.7977
	Mesh Type

7.5 Comparison of Superposition Method and Boundary Element Results in 3D Finite Reservoir

The boundary element superposition method (BESM) was developed in the early stage of this research for both 2D and 3D cases. Detailed description of the formulation as well as difficulties involved in the application of the BESM to the analysis of hydrodynamic problem were presented earlier. Because of these difficulties the study of this method was not pursued. A limited number of results from the BESM are presented here for the case of a finite 3D reservoir, impounded by an arch dam. The dam and reservoir boundaries are considered to be rigid. The finite reservoir model used in these studies is the same as shown in Fig. 7.19.

The rigid dam is assumed to undergo a horizontal upstream-downstream motion due to a ground motion acceleration of $e^{i\omega t}$. The hydrodynamic pressure distributions on the upstream face of the arch dam are calculated for two excitation frequencies: $\omega/\omega_1 = 0.4$ and $\omega/\omega_1 = 1.2$. The results are shown in Figs. 7.31a and 7.31b. The superposition values were obtained for a congruent boundary located 60 m outside the reservoir boundary. The distance of the congruent boundary, does influence the results. The results presented are compared to existing classical solution for radially infinite reservoir (37,77). Classical results for 3D finite, irregular reservoir are not available. For excitation frequencies $\omega < \omega_1$, the pressures in finite and infinite reservoirs are similar, but for $\omega > \omega_1$, the two are quite different.

The computations involved in the application of the superposition method are less than those required for the direct BEM because the former method does not involve integration over the elements.

Fig. 7.31a Absolute Value of the Hydrodynamic Pressure on an Arch Dam due to Upstream-Downstream Harmonic Motion in a 3D Finite Reservoir by SUP

$$-\frac{\omega_1}{\omega_0} = 0.4, D = 60 \text{ m}$$

Fig. 7.31b Absolute Value of the Hydrodynamic Pressure on an Arch
Dam due to Upstream-Downstream Harmonic Motion
in a 3D Finite Reservoir by SUP
- $\frac{\epsilon}{\epsilon_1} = 1.2$, $D = 60$ m

CHAPTER 8

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

8.1 Summary

The seismic response of concrete gravity and arch dams is affected by the hydrodynamic pressures in the water reservoir. Classical solutions for hydrodynamic pressures can be obtained only for very idealized geometry; in all other cases, a numerical solution must be attempted. Finite element method has been successfully applied in the past to a solution of the problem. More recently, boundary element solutions have been developed for certain 2D and 3D problems. The main objective of this study was to obtain more effective boundary element formulations for both the 2D and 3D dam-reservoir-foundation systems subjected to harmonic ground motion. In the work presented here, the dam has been considered as rigid but the techniques described are easily extended to a flexible dam.

This thesis presents the details of boundary element methods (BEM) developed for the analysis of hydrodynamic pressures in 2D and 3D reservoirs. The presentation begins with an introductory section on the classification of dam-reservoir-foundation problems and the seismic aspects of the design of gravity and arch dams. This is followed by a review of literature related to the boundary element method and the dam-reservoir-foundation problem. Formulation of BEM for a 2D mathematical model of a dam-reservoir-foundation system is presented next. Two alternative approaches are described in detail: one based on constant boundary elements and the other based on linear boundary elements. The reduction of governing differential equations to a two-dimensional Helmholtz equation and the choice of appropriate fundamental solution to transform the domain problem to a two-dimensional boundary problem are discussed. The definition of the mixed-boundary value

problem is then completed by introducing the boundary conditions. It is shown that in a BEM solution, only the boundary needs to be discretized. The infinite radiation condition which represents the energy loss in the outgoing waves in the reservoir is incorporated in the BEM by dividing the reservoir into two regions: one finite and irregular and the other infinite but with a uniform width with constant cross section. For the latter region a continuum or finite element solution can be obtained. This solution is then coupled with the boundary element solution applied to the finite irregular region of the reservoir. Compatibility of pressures and pressure gradients is then imposed at the interface of the regular and irregular regions, referred to as the transmitting boundary, to complete the solution. The foregoing procedure for modelling radiation damping through a transmitting boundary is similar to one used by other researchers in a finite element solution.

Numerical results are obtained and compared with available classical solutions and FEM results reported by other researchers. Convergence of the numerical results with the size and number of boundary elements is then studied. Some studies of the interrelationship between the size of an element and the excitation frequency are also carried out. Also, some analytical results are presented to demonstrate the effects of the number of elements used to model the finite region and the number of finite elements used in modelling of the infinite radiation damping.

The studies referred to in the foregoing paragraphs are based on what is commonly known as the direct boundary element method. A variation of BEM referred to as the boundary element superposition method presents certain computational advantage over the direct BEM. Limited studies based on the 2D boundary element superposition formulation are presented. It is noted that the method is not yet practical due to the uncertainties introduced by the choice of the location of the so-called congruent boundary.

In the second part of this work, a 3D boundary element formulation is developed for the analysis of hydrodynamic pressures in a reservoir impounded by an arch dam. Four-node constant quadrilateral boundary elements are used in this formulation. The infinite radiation and the foundation and banks damping are included in a 3D mathematical model of the arch dam-reservoir-foundation system in a manner very similar to that used for the 2D case. For the analysis of an infinite reservoir, the latter is divided into two regions: an irregular but finite region and an infinite channel of constant cross section. The infinite channel is modelled by 4-node linear isoparametric finite elements, while BEM is applied

to the finite region. The infinite channel solution is then coupled with boundary element solution of the finite region by enforcing the compatibility of pressure and pressure gradients at the interface.

Numerical results are obtained and compared with available classical solution results for a radially infinite reservoir. Limited convergence studies similar to these reported for the 2D case, are also carried out.

Finally, some results based on the 3D boundary element superposition formulation are presented for the case of a 3D finite reservoir.

To implement the analytical procedures described above several computer programs were developed. Program listings and user's manuals and details are given in Volume 2. Three programs based on the BEM were developed in this research: the constant element program for 2D reservoirs - BEMC.2D, the linear element program for 2D reservoirs - BEML.2D and the constant element program for 3D reservoirs - BEMC.3D. Two programs based on the BESM are also presented: the superposition program for 2D finite reservoirs - SUP.2D and the superposition program for 3D reservoirs - SUP.3D. A number of supplementary programs were developed to obtain classical solutions for both horizontal and vertical harmonic motions.

8.2 Conclusions

As a results of the research reported here, the following conclusions can be drawn. The conclusions are divided into three categories: (i) those related to 2D reservoir analysis, (ii) those related to 3D reservoir analysis, (iii) those of a general nature.

(i) 2D reservoir analysis

- 1) The boundary element method (BEM) is seen to be an effective procedure for evaluating the hydrodynamic pressures and pressure gradients caused in a 2D reservoir by the harmonic motion of the dam and/or the reservoir bottom in either a horizontal or a vertical direction or both.

- 2) When boundary element discretization is restricted only to the finite, irregular part of the reservoir, a real fundamental solution $Y_0(kr)$ is seen to be appropriate.
- 3) When compared to the finite element method (FEM), BEM offers the advantage of reducing the dimensionality of the problem by one. In the 2D case, however, the resulting saving in the computation time is offset by the time consumed in a numerical evaluation of the Bessel's functions appearing in the fundamental solution.
- 4) The matrices involved in the BEM solution are unsymmetric and fully populated and must be recomputed for each excitation frequency while those of FEM (except for the transmitting boundaries) are frequency independent.
- 5) The results obtained from an equal number of constant and linear elements are very similar. This indicates that in the present analysis, linear element formulation offer no particular advantage and constant element formulation which is simpler and involves less computational effort is adequate.
- 6) The location of the transmitting boundary which divides a reservoir into finite, irregular part and infinite, regular part must be selected with care. The results of the analysis seem to depend strongly on the geometrical shape of the irregular part. If the irregular part is nearly rectangular the results are less dependent on the location of the transmitting boundary.

(ii) 3D reservoir analysis

- 1) The boundary element method (BEM), is an effective procedure for evaluating the hydrodynamic pressures and pressure gradients caused in a 3D reservoir by the harmonic motion of dam/or the reservoir bottom and banks. The motion could be upstream-downstream, cross stream or vertical.
- 2) As in the 2D case, BEM reduces the dimensionality by one in the 3D case as well.
- 3) In comparison to the 2D case the fundamental solution for 3D case is simpler and is given by $p^* = -\frac{1}{4\pi r} e^{-ikr}$. However, the exponential fundamental solution is a complex function and includes real as well as imaginary parts.
- 4) When compared to the finite element solution, the matrices involved in the BEM

are of a smaller order. However, they are both unsymmetric and fully populated and must be, as in the 2D case, recomputed for each excitation frequency.

(iii) General conclusions

- 1) The infinite radiation condition first developed by other researchers (26,28) for the finite element method can be successfully used in both 2D and 3D boundary element formulations.
- 2) The simplified procedure to account for foundation and bank damping, again developed earlier for FEM (26,28), also works successfully in both 2D and 3D boundary element formulations.
- 3) To achieve reasonable accuracy, the overall length of one-dimensional boundary elements used in the 2D analysis as well as the larger diagonal of a two-dimensional element used in the 3D analysis must not exceed $1/10$ of the shortest wave length. This factor would specify also a number of elements required to solve a given problem.
- 4) The method presented here can be easily extended to treat a flexible dam, but a full substructure type of analysis must be carried out in that case.
- 5) Results are presented here for harmonic excitation. The analytical technique, however can be also extended to the case of the random earthquake motion using the Fast Fourier Transformation technique.

8.3 Recommendation for Future Research

The following recommendations are made for future research:

- 1) Implementation of the BEM to analyze a flexible dam for an arbitrary ground motion taking into account the foundation-dam interaction would be of interest. Following such implementation comparative studies can be carried out on the efficiencies of the BEM and FEM methods.

- 2) The proposed method of analysis can have applications in several other areas using two-dimensional and three-dimensional Helmholtz equations as the governing equations, e.g. in harbour engineering, offshore engineering and other areas dealing with the dynamic analysis of liquid retaining structures, submerged structures and floating structures; electrical engineering problems dealing with electrostatics and electromagnetism; sound engineering problems which are governed by the same reduced wave equation. However, the different natures of above phenomena require a separate study in each case with regard to the suitability of the method.
- 3) A drawback of the BEM is the frequency dependence of the fundamental solution which requires the formulation of the boundary matrices for each exciting frequency. The use of frequency independent fundamental solution is therefore of significant interest.
- 4) The effect of foundation and banks damping can be evaluated by a more detailed representation through the development of the foundation impedance matrices of the soil using the boundary element method.
- 5) The effect of the location of the transmitting boundary, in both 2D and 3D cases, should be studied further.
- 6) It would be valuable to supplement analytical studies such as these by experimental studies in the field as well as on scaled models.
- 7) Studies incorporating nonlinear behaviour of foundation soil and nonlinear behaviour of concrete should be of interest.

REFERENCES

1. Wahlstrom, E.E., "Dams, Dam Foundations and Reservoir Sites", *Elsevier Scientific, Amsterdam*, 1974.
2. Chopra, A.K., Chakrabarti, P., "Analysis of the Earthquake Performance of Koyna Dam", *Bull. of the Indian Society of Earthquake Technology*, Vol. 9, 1972, pp. 49-60.
3. Haws, E.T., Reilly, N., "Dams, Natural and Induced Earthquakes and the Environment", *Dams and Earthquake, Proceedings of a Conference Held at the Institution of Civil Engineers, London, U.K.*, 1981, pp. 253-260.
4. Mineiro, A.J.C., "Seismic Safety Criteria", *Proceedings of the International Conference on Safety of Dams, Coimbra, Portugal*, 1984, pp. 565-574.
5. U.S. Bureau of Reclamation. "Design of Gravity Dams", *Denver, Colorado, U.S. Government Printing Office*, 1976.
6. U.S. Bureau of Reclamation. "Design of Arch Dams", *Denver, Colorado, U.S. Government Printing Office*, 1977.
7. Japanese National Committee on Large Dams, "Design Criteria for Dams", *Tokyo*, 1971.
8. Indian Standard, "Criteria for Earthquake Resistant Design of Structures", *Third Revision, IS: 1893 - 1975*.
9. U.S. Bureau of Reclamation, "Design Criteria for Concrete Arch and Concrete Gravity Dams", *Engineering Monograph No. 19, Washington, D.C., U.S. Government Printing Office*, 1974.
10. U.S. Army Corps of Engineers, "Gravity Dam Design", *Design Manual EM 1110-2-2200, Washington, D.C.*, 1958.
11. U.S. Bureau of Reclamation, "Gravity Dams", *In Concrete Dams (Chapter 2), Design Standards No. 2, Denver, Colorado, Department of Interior*, 1966.
12. U.S. Bureau of Reclamation, "Arch Dams", *In Concrete Dams (Chapter 1), Design Standards No. 2, Denver, Colorado, Department of Interior*, 1965.

13. Westergaard, H.M., "Water Pressures on Dams during Earthquake", *Transactions, ASCE*, Vol. 98, 1933, pp. 418-433.
14. Zagnar, C.N., Hoefeli, R.J., "Electric Analog Indicates Effect of Horizontal Earthquake Shock on Dams", *Civil Engineering*, Vol. 22, 1952, pp. 278-279.
15. Zagnar, C.N., "Hydrodynamic Pressures on Dams due to Horizontal Earthquakes", *Proceedings of the Society for Experimental Stress Analysis*, Vol. 10, 1953, pp. 93-102.
16. Kotsubo, S., "Dynamic Water Pressures on Dams due to Irregular Earthquakes", *Memoirs, Faculty of Engineering, Kyushu, Iukuoka, Japan*, Vol. 18, 1959.
17. Bustamante, J.I., Flores, A., "Water Pressure on Dams Subjected to Earthquakes", *Journal of the Engineering Mechanics Division, ASCE*, Vol. 92, 1966, pp. 115-127.
18. Chopra, A.K., "Hydrodynamic Pressures on Dams due to Horizontal Earthquakes", *Journal of the Engineering Mechanics Division, ASCE*, Vol. 93, 1967, pp. 205-223.
19. Flores, A., Herrera, I., Lozano, C., "Hydrodynamic Pressure Generated by Vertical Earthquake Component", *Proceedings, Fourth World Conference on Earthquake Engineering, Santiago, Chile*, 1969.
20. Zienkiewicz, O.C., Newton, R.Z., "Coupled Vibration of Structure Submerged in a Compressible Fluid", *Proceedings of the International Symposium on Finite Element Techniques, Stuttgart, West Germany*, 1969.
21. Chakrabarti, P., Chopra, A.K., "Earthquake Analysis of Gravity Dams Including Hydrodynamic Interaction", *Earthquake Engineering and Structural Dynamics*, Vol. 2, 1973, pp. 143-160.
22. Sharan, K.S., "Earthquake Response of Dam-Reservoir-Foundation-Systems", *Ph.D. Thesis Submitted to the University of Waterloo, Waterloo, Canada*, 1978.
23. Chwang, A.T., Housner, G.W., "Hydrodynamic Pressures on Sloping Dams During Earthquakes, Part 1. Momentum Method", *Journal of Fluid Mechanics*, Vol. 87, 1978, pp. 335-341.
24. Chwang, A.T., "Hydrodynamic Pressures on Sloping Dams During Earthquakes, Part 2. Exact Theory", *Journal of Fluid Mechanics*, Vol. 87, 1978, pp. 343-348.

25. Chopra, A.K., Chakrabarti, P., "Earthquake Analysis of Concrete Gravity Dams Including Dam-Water-Foundation Rock Interaction", *Earthquake Engineering and Structural Dynamics*, Vol. 9, 1981, pp. 363-383.
26. Hall, J.F., Chopra, A.K., "Two Dimensional Dynamic Analysis of Concrete Gravity and Embankment Dams Including Hydrodynamic Effects", *Earthquake Engineering and Structural Dynamics*, Vol. 10, 1982, pp. 305-332.
27. Hall, J.F., Chopra, A.K., "Hydrodynamic Effects in the Dynamic Response of Concrete Gravity Dams", *Earthquake Engineering and Structural Dynamics*, Vol. 10, 1982, pp. 333-345.
28. Hall, J.F., Chopra, A.K., "Earthquake Analysis and Response of Concrete Gravity Dams", Report No. UCB/EERC-80/39, Earthquake Engineering Research Center, University of California, Berkeley, California, 1980.
29. Lotfi, V., "Analysis of the Response of Dams to Earthquakes", Ph.D. Thesis Submitted to the University of Texas at Austin, Austin, Texas, 1985.
30. Lotfi, V., Roesset, J.M., Tassoulas, J.L., "A Technique for the Analysis of the Response of Dams to Earthquakes", *Earthquake Engineering and Structural Dynamics*, Vol. 15, 1987, pp. 463-490.
31. Fenves, G., Chopra, A.K., "Effects of Reservoir Bottom Absorption on Earthquake Response of Concrete Gravity Dams", *Earthquake Engineering and Structural Dynamics*, Vol. 11, 1983, pp. 809-829.
32. Fenves, G., Chopra, A.K., "Earthquake Analysis of Concrete Gravity Dams Including Reservoir Bottom Absorption and Dam-Water-Foundation Rock Interaction", *Earthquake Engineering and Structural Dynamics*, Vol. 12, 1984, pp. 663-680.
33. Fenves, G., Chopra, A.K., "Earthquake Analysis and Response of Concrete Gravity Dams", Report No. UCB/EERC-84/10, Earthquake Engineering Research Center, University of California, Berkeley, California, 1984.
34. Fenves, G., Chopra, A.K., "Effects of Reservoir Bottom Absorption Effects and Dam-Water-Foundation Rock Interaction on Frequency Response Functions for Concrete Gravity Dams", *Earthquake Eng. and Structural Dynamics*, Vol. 13, 1985, pp. 13-31.

35. Fenves, G., Chopra, A.K., "Simplified Earthquake Analysis of Concrete Gravity Dams: Combined Hydrodynamic and Foundation Interaction Effects", *Journal of Engineering Mechanics Division, ASCE, Vol. 111, 1985*, pp. 715-735.
36. Fenves, G., Chopra, A.K., "Simplified Earthquake Analysis of Concrete Gravity Dams: Separate Hydrodynamic and Foundation Interaction Effects", *Journal of Engineering Mechanics Division, ASCE, Vol. 111, 1985*, pp. 736-756.
37. Porter, C.S., Chopra, A.K., "Hydrodynamic Effects in Dynamic Response of Simple Arch Dams", *Earthquake Engineering and Structural Dynamics, Vol. 10, 1982*, pp. 417-431.
38. Fok, K.L., Chopra, A.K., "Earthquake Analysis and Response of Concrete Arch Dams", *Report No. UCB/EERC -85/07, Earthquake Engineering Research Center, University of California, Berkeley, California, 1985*.
39. Fok, K.L., Chopra, A.K., "Earthquake Analysis of Arch Dams Including Dam-Water Interaction, Reservoir Boundary Absorption, and Foundation Flexibility", *Earthquake Engineering and Structural Dynamics, Vol. 14, 1986*, pp. 155-184.
40. Fok, K.L., Chopra, A.K., "Hydrodynamic and Foundation Flexibility Effects in Earthquake Response of Arch Dams", *Journal of Structural Engineering, ASCE, Vol. 112, 1986*, pp. 1810-1828.
41. Chopra, A.K., Chakrabarti, P., "The Earthquake Experience at Koyna Dam and Stresses in Concrete Gravity Dams", *Earthquake Engineering and Structural Dynamics, Vol. 1, 1972*, pp. 151-164.
42. Chopra, A.K., Chakrabarti, P., "The Koyna Earthquake and Damage to Koyna Dam", *Bulletin of the Seismological Society of America, Vol. 63, 1973*, pp. 381-397.
43. Dixon, R., Liaw, C.-Y., Chopra, A.K., "Mathematical Models for the Dynamic Analysis of Concrete Gravity Dams", *Earthquake Engineering and Structural Dynamics, Vol. 3, 1975*, pp. 249-258.
44. Chopra, A.K., "Earthquake Analysis, Design and Safety Evaluation of Concrete Dams", *Proceedings, the 5th Canadian Conference on Earthquake Engineering, Ottawa, 1987*.
45. Banerjee, P.K., Butterfield, R., ed., "Developments in Boundary Element Method - 1", *Applied Science Publishers Ltd., London, 1979*.

46. Zienkiewicz, O.C., Lewis, R.W., Stagg, K.G., editors, "Numerical Methods in Offshore Engineering", *John Wiley & Sons, Chichester, 1978.*
47. Jawsom, M.A., Sym, G.T., "Integral Equation Methods in Potential Theory and Elastostatics", *Academic Press, London, 1977.*
48. Cruse, T.A., Rizzo, F.J., "A Direct Formulation and Numerical Solution of the General Transient Elastodynamic Problem", Part I and part II, *Journal of Mathematical Analysis and Applications, Vol. 22, 1968, pp. 249-259 and pp. 341-355.*
49. Rizzo, F.J., "An Integral Equation Approach to Boundary Value Problem of Classical Elastostatics", *Quarterly of Applied Mathematics, Waterloo, Canada, 1967, pp. 83-95.*
50. Hanna, Y.G., "Application of Boundary Element Method to Certain Problems in Structural Dynamics", *M.Eng. Thesis Submitted to Carleton University, Ottawa, 1980.*
51. Hanna, Y.G., Humar, J.L., "Boundary Element Analysis of Fluid Domain", *Journal of Engineering Mechanics Division, ASCE, Vol. 108, 1982, pp. 436-450.*
52. Liu, P.L.-F., Cheng A.H.-D., "Boundary Solutions for Fluid-Structure Interaction", *Journal of Hydraulic Engineering, ASCE, Vol. 110, 1984, pp. 51-64.*
53. Humar, J.L., Jablonski, A.M., "Boundary Element Analysis of the Hydrodynamic Forces in a 2D Reservoir", *Proceedings of the 5th Canadian Conference on Earthquake Engineering, Ottawa, Canada, 1986.*
54. Humar, J.L., Jablonski, A.M., "Boundary Element Reservoir Model for Seismic Analysis of Gravity Dams", *Paper to be Published in Earthquake Engineering and Structural Dynamics, 1988.*
55. Wepf, D.H., Wolf, J.P., "Time Domain Dam-Reservoir Interaction Analysis Based on Boundary Elements", *Proceedings of China-U.S. Workshop on Earthquake Behaviour of Arch Dams, Beijing, China, 1987.*
56. Elkhodary, M.A., "Boundary Element Analysis of Hydrodynamic Pressure on Arch Dams", *M.Eng. Thesis Submitted to Carleton University, 1984.*
57. Humar, J.L., Jablonski, A.M., "Boundary Element Superposition Analysis of Reservoir Motion", *Proceedings of the 8th European Conference on Earthquake Engineering, Lisbon, Portugal, 1986.*

58. Bičkovski, V., "Mathematical Modelling of Arch Dams Under High Seismicity Conditions", *Proceedings of the 8th European Conference on Earthquake Engineering, Lisbon, Portugal, 1986*.
59. Humar, J.L., Jablonski, A.M., "Boundary Element Analysis of Hydrodynamic Pressure on an Arch Dam", *Paper to be Published in the Proceedings of the 9th World Congress on Earthquake Engineering, Tokyo, Japan, 1988*.
60. Abdalla, M.H.M., "Dynamic Soil-Structure Analysis by Boundary Element Method", *Ph.D. Thesis Submitted to Carleton University, Ottawa, Canada, 1984*.
61. De Mey, G., "Calculation of Eigenvalues of Helmholtz Equation", *International Journal for Numerical Methods in Engineering, Vol. 10, 1977, pp. 59-66*.
62. De Mey, G., "A Simplified Integral Equation Method for Calculation of Helmholtz Equation", *International Journal for Numerical Methods in Engineering, Vol. 11, 1977, pp. 1340-1342*.
63. Arfhan, G., "Mathematical Methods for Physicists", *Academic Press Inc., Orlando, Florida, 1985*.
64. Sharan, S.K., "A Non-Reflecting Boundary in Fluid-Structure Interaction", *Computers & Structures, Vol. 26, 1987, pp. 811-846*.
65. Brebbia, C.A., "Boundary Element Method for Engineers", *Pentech Press, Devon, England, 1977*.
66. Greenberg, M.D., "Application of Green's Functions in Science and Engineering", *Hall Inc., Englewood Cliffs, New Jersey, 1971*.
67. Collins, R.E., "Mathematical Methods for Physicists and Engineers", *Reinhold Physics Textbook Series, Reinhold Book Corp., New York, 1968*.
68. Hutchinson, J.R., "Determination of Membrane Vibrational Characteristics by the Boundary Integral Equation Method", *Recent Advances in Boundary Element Methods, C. Brebbia, ed., Pentech Press, London, England, 1978, pp. 301-316*.
69. Banerjee, P.K. Butterfield, R., "Boundary Element Methods in Engineering and Science", *McGraw-Hill Book Co. (U.K.) Ltd., Maidenhead, Berkshire, England, 1981*.
70. Burgess, G., Mahajerin, E., "A Comparison of the Boundary Element and Superposition Methods", *Computers & Structures, Vol. 19, 1984, pp. 697-705*.

71. Burgess, G., Mahajerin, E., "A Numerical Method for Laterally Loaded Thin Plates", *Computers in Applied Mechanics and Engineering*, Vol. 49, 1985, pp. 1-5.
72. Beer, G., Comment on "A Comparison of the Boundary Element and Superposition Methods", *Computers & Structures*, Vol. 23, 1986, pp.459.
73. Wolf, J.P., Darbre, G.R., "Dynamic-Stiffness Matrix of Soil by the Boundary Element Method: Conceptual Aspects", *Earthquake Engineering and Structural Dynamics*, Vol. 12, 1984, pp. 385-400.
74. Moler, C.B., Stewart, G.W., "An Algorithm for Generalized Matrix Eigenvalues Problems", *Journal of Numerical Analysis, SIAM*, Vol. 10, 1973, pp. 241-265.
75. Kandasamy, K., "Three-Dimensional Dynamic Analysis of Gravity Dam and Reservoir Systems", *Ph.D. Thesis Submitted to The John Hopkins University, Baltimore, Maryland, 1986*.
76. Tsai, Ch.-Sh., Lee, G.C., "Arch Dam-Fluid Interactions: by FEM-BEM and Substructure Concept", *International Journal for Numerical Methods in Engineering*, Vol. 24, 1987, pp. 2367-2388.
77. Porter, C.S., Chopra, A.K., "Dynamic Response of Simple Arch Dams Including Hydrodynamic Interaction", *Report No. UCB/EERC-80/17, Earthquake Engineering Research Center, University of California, Berkeley, California, 1980*.
78. Liu, P.L.-F., "Hydrodynamic Pressures on Rigid Dams During Earthquakes", *Journal of Fluid Mechanics*, Vol. 165, 1986, pp. 131-145.
79. Lean, M.H., Wexler, A., "Accurate Field Computation with the Boundary Element Method", *IEEE Transactions on Magnetics*, Vol. Mag-18, 1982, pp. 331-335.
80. Seybert, A.F., Soenarko, B., Rizzo, F.J., Shippy, D.J., "Application of the BIE Method to Sound Radiation Problems Using an Isoparametric Element", *Journal of Vibration, Acoustics, Stress and Reliability in Design, ASME*, Vol. 106, 1984, pp. 414-420.
81. Hung, T.-K., Wang, M.-H., "Nonlinear Hydrodynamic Pressure on Rigid Dams", *Journal of Engineering Mechanics, ASCE*, Vol. 113, 1987, pp. 482-489.
82. Soong, T.W., Murtha, J.P., "Boundary Integral Equation Approach to Earthquake Induced Hydrodynamic Forces", *Proceedings of the 5th Conference on Computing in Civil Engineering: Microcomputers to Supercomputers, Alexandria, Virginia, 1988*.

APPENDIX A

NOTATIONS

- A** a water domain in the 2D case
- A** the assembled $M \times M$ matrix in the formulation of the 2D eigenvalue problem (see Appendix C) or in the formulation of the 3D eigenvalue problem (see Appendix E)
- a_n the harmonic excitation amplitude along the outward normal n
- a_x the harmonic excitation amplitude along the x - direction
- a_y the harmonic excitation amplitude along the y - direction
- B** the assembled $M \times M$ matrix whose only non-zero term is the last element on the diagonal with value equal to $p_v(0)$
- b** a vector of unknown source strengths
- C** the influence matrix with elements C_{ij}
- c velocity of sound in water ($1440 \frac{m}{s}$)
- c_r the compression wave velocity in the foundation stratum
- c_i the constant whose value depends on the location of point i
- D** the distance between congruent and real boundaries
- D** a vector with one non-zero term equal to the vertical acceleration of the base node b
- D** a water domain in the 2D SUP formulation
- E** a diagonal matrix of terms $e^{k_n z}$
- E_r Young's modulus of the foundation stratum
- F** the assembled $M \times M$ matrix in the formulation of the 2D eigenvalue problem (see Appendix C) or in the formulation of the 3D eigenvalue problem (see Appendix E)
- F_d** a diagonal matrix with terms equal to $i\omega\gamma$

- $\mathcal{F}$ a water domain in the 3D SUP formulation
- G matrix containing elements G_{ij}
- G_t Jacobian transformation matrix relating two local coordinate systems (ξ_1, ξ_2) and (η_1, η_2)
- $|G|$ Jacobian of transformation given by $|G| = \det G_t$
- g acceleration due to gravity $= 9.81 \frac{m}{s^2}$
- $\hat{g}_n$ a vector in the linear boundary element formulation given by Eq. 3.35
- $\hat{g}_{m-1}$ a vector in the linear boundary element formulation given by Eq. 3.39
- $\hat{g}_m$ a vector in the linear boundary element formulation given by Eq. 3.40
- H height of the water reservoir
- $H_0^{(2)}(kr)$ the Hankel function of zero order and second kind
- H matrix containing elements H_{ij}
- i imaginary number $= \sqrt{-1}$
- I the identity matrix
- I_n the integral given by Eq. 3.56 required in the far end condition formulation
- $|J|$ Jacobian relating the global system of coordinates (x, y, z) to the local system (ξ_1, ξ_2)
- k the wave constant $= \frac{\omega}{c}$
- K a diagonal matrix with elements $\kappa_1, \kappa_2, \dots, \kappa_M$
- l_j the length of the j th element
- M the order of the eigenvalue problem
- N the number of the boundary equations
- n the outward normal to the reservoir's boundary
- p hydrodynamic pressure in excess of the hydrostatic pressure
- p^* the fundamental solution for the Helmholtz equation
- q hydrodynamic pressure gradient
- Q_j point source strength at point j

Q	a vector of unknown source strengths
q^*	derivative of p^* with respect to the outward normal n
R	radius of the upstream face of an arch dam
r	radial distance in the 2D or 3D case
s	coordinate along the dam face
s', r'	coordinates along the reservoir bottom and sides
s_j	the boundary of the j th element in the 2D case
S	the boundary of the 2D or 3D water domain
S_j	the boundary of the j th element in the 3D case
$\bar{S}$	the boundary of the 2D or 3D water domain in the SUP formulation
t	time variable
T_p	the transformation matrix between boundary element nodal values and finite element nodal values at the far boundary
u, v, w	components of the displacement of a particle of water in the cartesian coordinates
V	a three-dimensional water domain
w	the specific gravity of water
w_m	the weighting function used in the Gauss quadrature scheme
w_r	the specific gravity of the foundation soil or rock
x, y, z	orthogonal cartesian coordinate system (Fig. 1.2 and Fig. 1.3)
$Y_0(kr)$	the Bessel function of zero order and second kind
α	the angle the dam face makes with the vertical
α_R	the wave reflection coefficient
β	the external angle at a corner of the domain
γ	the foundation damping coefficient (see Appendix B)
Δ^i	Dirac Delta function centered at point $i(x_i, y_i)$
Δ_ν	the length of one element at the far boundary

- $\{\eta\}$ a vector of modal coordinates
 η_n the n th modal coordinate
 θ the angle between the outward normal n and the vector radius r
 κ the separation constant
 λ the constant given by $\kappa^2 + k^2$
 Λ the diagonal matrix of the M eigenvalues $\lambda_1^2, \lambda_2^2, \dots, \lambda_M^2$
 ξ the local coordinate of an element in the 2D formulation
 ξ_1, ξ_2 local coordinate system given by Eq. 6.18
 ρ specific mass of water
 ϕ the angle that reservoir's bottom makes with the horizontal
 $\phi_1, \phi_2, \phi_3, \phi_4$ shape functions required for the transformation from the (x, y, z) system to the (ξ_1, ξ_2) system
 ψ_n the n th eigenvector function
 ψ_1, ψ_2 the shape functions required for the transformation from the (ξ_1, ξ_2) system to the (η_1, η_2) system
 Ψ the matrix of M eigenvectors
 ω the circular frequency of the harmonic ground motion

APPENDIX B

FOUNDATION DAMPING CONDITION DERIVATION

A one-dimensional reservoir-foundation system was first introduced by Hall and Chopra (26,28). A detailed presentation of the interface interaction condition derived by them is given here. The limitations of this approach are also indicated.

Fig. B.1 One-Dimensional Reservoir-Foundation System (28)

The one-dimensional reservoir-foundation system consists of two adjacent half-spaces: fluid half-space and foundation soil half-space (Fig. B.1). The problem is assumed to be one-dimensional, in y -direction only. There is no variation of any kind in x -direction. The excitation is initiated by an incident displacement wave travelling upward through the

foundation.

- water reservoir

For a harmonic excitation, the resulting pressure in the reservoir is governed by a one-dimensional Helmholtz equation:

$$\frac{d^2 p}{dy^2} + k^2 p = 0 \quad (B-1)$$

in which $k = \frac{\omega}{c}$ is the wave constant.

On the reservoir-foundation interface pressure $p(y, \omega)$ must satisfy the following boundary condition:

$$\frac{dp}{dy}(y, \omega) = -\frac{\omega}{g} \frac{\partial^2 u}{\partial t^2} \quad (B-2)$$

But since $u = e^{i\omega t}$, Eq. B-2 reduces to

$$\frac{dp}{dy}(y, \omega) = \rho \omega^2 u(y, \omega); \quad y = 0 \quad (B-3)$$

- foundation

Displacements within the foundation soil are also governed by a one-dimensional Helmholtz equation:

$$\frac{d^2 u}{dy^2} + k_r^2 u = 0 \quad (B-4)$$

where $k_r = \frac{\omega}{c_r}$ and c_r = compression wave velocity in the foundation soil.

The general solution to Eq. B-4 is

$$u(y, \omega) = A(\omega) e^{i \frac{k_r}{c_r} y} + B(\omega) e^{-i \frac{k_r}{c_r} y} \quad (B-5)$$

where $A(\omega)$ is the unknown amplitude of the reflected displacement wave and $B(\omega)$ is the specified amplitude of the incident displacement wave.

First consider the case when there is no water in the reservoir so that the acceleration measured at the ground is the free-field acceleration. A relationship between the ampli-

tude $B(\omega)$ of the incident wave and the free-field acceleration a_y can be derived from the consideration that at $y = 0$ the stress and hence the strain in the foundation must be zero.

$$\frac{du}{dy} = 0 \quad \text{at } y = 0 \quad (B - 6)$$

From Eq. B-5:

$$\frac{du}{dy} = A i \frac{\omega}{c_r} e^{i \frac{\omega}{c_r} y} - B i \frac{\omega}{c_r} e^{-i \frac{\omega}{c_r} y} \quad (B - 7)$$

$$A i \frac{\omega}{c_r} \times 1 = B i \frac{\omega}{c_r} \times 1 \quad (B - 8)$$

Thus,

$$A(\omega) = B(\omega) \quad (B - 9)$$

Substituting Eq. B-9 into Eq. B-5 one can write:

$$u(y, \omega, t) = \left(B(\omega) e^{i \frac{\omega}{c_r} y} + B(\omega) e^{-i \frac{\omega}{c_r} y} \right) e^{i \omega t} \quad (B - 10)$$

Differentiating twice w.r.t. time variable t

$$\frac{d^2 u}{dt^2} = -\omega^2 \left(B(\omega) e^{i \frac{\omega}{c_r} y} + B(\omega) e^{-i \frac{\omega}{c_r} y} \right) e^{i \omega t} \quad (B - 11)$$

At $y = 0$:

$$\left. \frac{d^2 u}{dt^2} \right|_{y=0} = -\omega^2 (B(\omega) + B(\omega)) e^{i \omega t} \quad (B - 12)$$

or

$$\left. \frac{d^2 u}{dt^2} \right|_{y=0} = -2B(\omega) \omega^2 e^{i \omega t} \quad (B - 13)$$

Equation B-13 thus gives:

$$a_y = -\omega^2 2B(\omega) \quad (B - 14)$$

or

$$B(\omega) = -\frac{a_y}{2\omega^2} \quad (B-15)$$

Thus, an upward propagating wave has an acceleration amplitude equal to $\frac{1}{2}a_y$, which doubles in its value upon reflection at a free boundary at $y = 0$.

When water is present in the reservoir, the incident wave amplitude will still remain $B(\omega)$, however the amplitude of the reflected wave will be altered.

$$\begin{aligned} u(y, \omega) &= D(\omega)e^{i\frac{\omega}{c_r}y} + B(\omega)e^{-i\frac{\omega}{c_r}y} \\ &= D(\omega)e^{i\frac{\omega}{c_r}y} - \frac{a_y}{2\omega^2}e^{-i\frac{\omega}{c_r}y} \end{aligned} \quad (B-16)$$

An expression for $D(\omega)$ can be obtained by noting that the pressure at the reservoir-foundation interface must equal the normal stress in the foundation soil.

$$p(0, \omega) = -E_r \epsilon_y \quad (B-17a)$$

or

$$p(0, \omega) = -E_r \frac{du}{dy}(0, \omega) \quad (B-17b)$$

where $E_r = \frac{w_r}{g}c_r^2$ is the effective modulus of elasticity of the foundation soil; w_r = the specific gravity of foundation soil; g = acceleration due to gravity.

On combining Eqs. B-16 and B-17 one can find a solution for D , the seeking amplitude of the reflected wave:

$$\left. \frac{du}{dy} \right|_{y=0} = +\frac{i\omega}{c_r}D(\omega) + \frac{i\omega a_y}{2c_r\omega^2} \quad (B-18)$$

$$p(0, \omega) = -\frac{w_r}{g}c_r^2 \left(\frac{i\omega}{c_r}D(\omega) + \frac{i\omega a_y}{2c_r\omega^2} \right) \quad (B-19)$$

Equation B-19 gives the following value for D :

$$D(\omega) = - \left(\frac{a_y}{2\omega^2} + \frac{g}{w_r c_r} \frac{1}{i\omega} p(0, \omega) \right) \quad (B - 20)$$

Substitution of Eqs. B-20 in B-16 gives

$$u(y, \omega) = - \left(\frac{a_y}{2\omega^2} + \frac{g}{w_r c_r} \frac{1}{i\omega} p(0, \omega) \right) e^{i \frac{y}{c_r} \omega} - \frac{a_y}{2\omega^2} e^{-i \frac{y}{c_r} \omega} \quad (B - 21)$$

On combining Eq. B-3 and B-21

$$\frac{dp}{dy}(0, \omega) = - \frac{w}{g} a_y + \frac{w}{w_r c_r} i\omega p(0, \omega) \quad (B - 22)$$

or

$$\frac{dp}{dy}(0, \omega) = - \frac{w}{g} a_y + \gamma i\omega p(0, \omega) \quad (B - 23)$$

where $\gamma = \frac{w}{w_r c_r}$, the damping coefficient; w = specific gravity of water; w_r = specific gravity of foundation soil; c_r = compressive wave velocity in the foundation soil.

If there is no acceleration of the bottom floor of the reservoir, i.e. $a_y = 0$, condition B-23 has a simplified form:

$$\frac{dp}{dy}(0, \omega) = i\omega \gamma p(0, \omega) \quad (B - 24)$$

The reflection coefficient can be found from the following expression (26)

$$\alpha_R = \frac{1 - \gamma c}{1 + \gamma c} = \frac{1 - \frac{w}{w_r c_r}}{1 + \frac{w}{w_r c_r}} \quad (B - 25)$$

The reservoir-foundation interface condition derived above does not consider two-dimensional type of excitation in both half-spaces. It considers only a one-dimensional form of excitation in y -direction due to incident displacement wave travelling in the direction normal to the interface. Recently this condition has been criticized by Lofti, Roesset and Tassoulas on the ground that it tends to overestimate the damping due to flexible foundation (30).

APPENDIX C

DEVELOPMENT OF BOUNDARY CONDITION FOR RADIATION DAMPING IN 2D CASE

The hydrodynamic pressure p in the infinite region of the 2D reservoir can be expressed as $p = p_x p_y$ where the pressure components satisfies the following separated equation (see Section 2.5 and reference 26)

$$\frac{d^2 p_y}{dy^2} + \lambda^2 p_y = 0 \quad (C-1)$$

with the boundary conditions

$$\begin{aligned} p_y &= 0 & \text{at } y &= H \\ \frac{dp_y}{dy} &= i\omega\gamma p_y & \text{at } y &= 0 \end{aligned} \quad (C-2)$$

In Eqs. C-1 and C-2, κ = the separation constant and λ is given by

$$\lambda^2 = k^2 + \kappa^2 \quad (C-3)$$

Equation C-1 is solved by a one-dimensional finite element discretization. To do this let Eq. C-1 be satisfied in a weighted sense so that

$$\int_0^H \left(\frac{d^2 p_y}{dy^2} + \lambda^2 p_y \right) \delta p_y dy = 0 \quad (C-4)$$

The first term under integral in Eq. C-4 can be reduced by partial integration as follows

$$\int_0^H \frac{d^2 p_y}{dy^2} \delta p_y dy = \frac{dp_y}{dy} \delta p_y \Big|_0^H - \int_0^H \frac{dp_y}{dy} \frac{d}{dy} (\delta p_y) dy \quad (C-5)$$

Substitution of the boundary conditions from Eq. C-2 gives

$$\int_0^H \frac{d^2 p_y}{dy^2} dy = -i\omega\gamma p_y(0) \delta p_y(0) - \int_0^H \frac{dp_y}{dy} \delta \left(\frac{dp_y}{dy} \right) dy \quad (C-6)$$

On substituting Eq. C-6 into Eq. C-4 the following relationship is obtained:

$$\delta \left\{ \int_0^H \left[-\frac{1}{2} \left(\frac{dp_y}{dy} \right)^2 + \frac{1}{2} \lambda^2 p_y^2 \right] dy - \frac{1}{2} i\omega\gamma p_y^2(0) \right\} = 0 \quad (C-7)$$

or

$$\delta \chi = 0 \quad (C-8)$$

where χ is a functional given by

$$\chi = \int_0^H \left[\frac{1}{2} \left(\frac{dp_y}{dy} \right)^2 - \frac{1}{2} \lambda^2 p_y^2 \right] dy + \frac{1}{2} i\omega\gamma p_y^2(0) \quad (C-9)$$

The far boundary is now divided into a number of finite elements and the functional χ is expressed in a discrete form as

$$\chi = \sum_{n=1}^M \left\{ \int_n \left[\frac{1}{2} \left(\frac{dp_y}{dy} \right)^2 - \frac{1}{2} \lambda^2 p_y^2 \right] dy + \frac{1}{2} i\omega\gamma p_y^2(0) \right\} \quad (C-10)$$

In a finite element discretization the system variables are assumed to vary in a prescribed manner within an element. A linear variation of p_y is assumed in the present formulation. The far boundary is divided into a number of linear finite elements (see Fig. C.1). In each element variable y is related to the natural coordinate ξ by the expression

$$y = \begin{pmatrix} \frac{1}{2}(1-\xi) & \frac{1}{2}(1+\xi) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (C-11)$$

where y_1 and y_2 are the nodal values of y .

Equation C-11 can be expressed as

$$y = \begin{pmatrix} N_1 & N_2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (C-12)$$

Fig. C.1 Linear Discretization of the Far Boundary and One Linear Element

or as

$$y = N y^e \quad (C - 13)$$

where N_1 and N_2 are the shape functions shown in Fig. C.2, N is the row vector of N_i and y^e is the column vector of the nodal values.

Using the same shape functions to describe the variation p_y within an element

$$p_y = N p_v \quad (C - 14)$$

Substitution of Eq. C-14 in Eq. C-10 gives

$$\chi = \sum_{n=1}^M \left[\int_{-1}^{+1} \left(\frac{1}{2} p_v^T \frac{dN^T}{dy} \frac{dN}{dy} p_v - \frac{1}{2} \lambda^2 p_v^T N^T N p_v \right) \frac{dy}{d\xi} d\xi + \frac{1}{2} i \omega \gamma p_v^2(0) \right] \quad (C - 15)$$

The minimization of the functional χ required by Eq. C-8 can be expressed as

$$\frac{d\chi}{d\{p_y\}} \delta\{p_y\} = 0 \quad (C-16)$$

Fig. C.2 Linear Shape Functions

or as

$$\frac{d\chi}{d\{p_y\}} = 0 \quad (C-17)$$

Using Eq. C-15 for χ , Eq. C-17 gives

$$\sum_{n=1}^M \left[\int_{-1}^{+1} (B_n^T B_n p_y - \lambda^2 N^T N p_y) \frac{dy}{d\xi} d\xi + 2\frac{1}{2} i\omega\gamma p_y^2(0) \right] = 0 \quad (C-18)$$

where

$$B_n = \frac{dN}{dy} \quad (C-19)$$

Individual terms within Eq. C-18 can be reduced as follow

$$\begin{aligned} \int_{-1}^{+1} \mathbf{N}^T \mathbf{N} \frac{dy}{d\xi} d\xi &= \int_{-1}^{+1} \begin{pmatrix} \frac{1}{2}(1-\xi) \\ \frac{1}{2}(1+\xi) \end{pmatrix} \begin{pmatrix} \frac{1}{2}(1-\xi) & \frac{1}{2}(1+\xi) \end{pmatrix} \frac{l}{2} d\xi \\ &= \frac{l}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \end{aligned} \quad (C-20)$$

where l = the length of one finite element.

Differentiation of the shape function vector gives

$$\frac{d\mathbf{N}^T}{dy} = \frac{d\mathbf{N}^T}{d\xi} \frac{d\xi}{dy} = \frac{2}{l} \frac{d}{d\xi} \begin{pmatrix} \frac{1}{2}(1-\xi) \\ \frac{1}{2}(1+\xi) \end{pmatrix} = \frac{2}{l} \begin{pmatrix} -\frac{1}{2} \\ +\frac{1}{2} \end{pmatrix} \quad (C-21)$$

Thus

$$\begin{aligned} \int_{-1}^{+1} \frac{d\mathbf{N}^T}{dy} \frac{d\mathbf{N}}{dy} dy &= \int_{-1}^{+1} \frac{2}{l} \begin{pmatrix} -\frac{1}{2} \\ +\frac{1}{2} \end{pmatrix} \frac{2}{l} \begin{pmatrix} -\frac{1}{2} & +\frac{1}{2} \end{pmatrix} \frac{dy}{d\xi} d\xi \\ &= \frac{4}{l^2} \int_{-1}^{+1} \begin{pmatrix} -\frac{1}{2} \\ +\frac{1}{2} \end{pmatrix} \begin{pmatrix} -\frac{1}{2} & +\frac{1}{2} \end{pmatrix} \frac{l}{2} d\xi \\ &= \frac{1}{l} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \end{aligned} \quad (C-22)$$

The following notations are introduced at this stage:

$$\mathbf{F}^e = \int_{y^1}^{y^2} \mathbf{N}^T \mathbf{N} dy = l \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} \quad (C-23a)$$

$$\mathbf{A}^e = \int_{y^1}^{y^2} \mathbf{B}_n^T \mathbf{B}_n dy = \frac{1}{l} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (C-23b)$$

Equation C-18 can now be written as follows:

$$\sum_{n=1}^M \left[\frac{1}{l} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \mathbf{p}_v - \lambda^2 l \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} \mathbf{p}_v + i \omega \gamma \mathbf{p}_v(0) \right] = 0 \quad (C-24)$$

or in a matrix form as

$$[\mathbf{A} + i\omega\gamma \mathbf{B}] \mathbf{p} = \lambda^2 \mathbf{F} \mathbf{p} \quad (C - 25)$$

where $\mathbf{A}$ and $\mathbf{F}$ are assembled matrices whose individual components are given by Eq. 2.40 and $\mathbf{B}$ is an $M \times M$ matrix whose only non-zero term is the last element on the diagonal with value equal to $p_v(0)$. As stated earlier $\mathbf{p}_v$ is the vector of the pressure values at the nodes. The surface node is excluded from the formulation because the pressure value is known to be zero there.

APPENDIX D

DERIVATION OF THE TRANSFORMATION MATRIX T_p FOR CONSTANT BOUNDARY ELEMENTS

Matrix T_p transforms element side nodal values of pressures and pressure gradients to midpoint nodal values (Fig. D.1).

Fig. D.1 A Finite Element Discretization of the Far Boundary

The following is the assumed relationship between the two sets of values

$$\begin{aligned}
 p_1 &= \frac{\hat{p}_1 + \hat{p}_2}{2} \\
 p_2 &= \frac{\hat{p}_2 + \hat{p}_3}{2} \\
 p_3 &= \frac{\hat{p}_3 + \hat{p}_4}{2} \\
 p_4 &= \frac{\hat{p}_4}{2}
 \end{aligned}
 \tag{D - 1}$$

where $\hat{p}$ represents the pressure values at the two ends of an element and p represents the pressure value at the midpoint of the element. Note that $\hat{p}_5 = 0$ on the free surface.

Eq. D-1 can be written in the matrix form

$$p = T_p \hat{p} \tag{D - 2}$$

where T_p is given by

$$T_p = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \tag{D - 3}$$

The relationship between the two sets of pressure gradient values is assumed to have the same form, although some error is associated with this assumption due to the fact that the pressure gradient on the free surface $\hat{q}_5$ is not equal to zero.

$$q = T_p \hat{q} \tag{D - 4}$$

APPENDIX E

DEVELOPMENT OF BOUNDARY CONDITION FOR RADIATION DAMPING IN 3D CASE

The hydrodynamic pressure p in the infinite channel of the 3D reservoir can be expressed as $p = p_x p_{yz}$ where the pressure components p_{yz} satisfies the following separated equation (see Section 5.5 and reference 28)

$$\frac{\partial^2 p_{yz}}{\partial y^2} + \frac{\partial^2 p_{yz}}{\partial z^2} + \lambda^2 p_{yz} = 0 \quad (E-1)$$

where $\lambda^2 = \frac{\omega^2}{c^2} + \kappa^2$

Equation E-1 is solved by a two-dimensional finite element discretization. In order to derive the finite element equations Eq. E-1 is satisfied in a weighted sense, so that

$$\iint_A \left(\frac{\partial^2 p_{yz}}{\partial y^2} + \frac{\partial^2 p_{yz}}{\partial z^2} + \lambda^2 p_{yz} \right) \delta p_{yz} dydz = 0 \quad (E-2)$$

where the two-dimensional domain A , represents a regular x-section of the infinite channel. Using Green's theorem the integrals of the first terms in Eq. E-2 can be reduced as follows

$$\iint_A \frac{\partial^2 p_{yz}}{\partial y^2} \delta p_{yz} dydz = - \int_S \frac{\partial p_{yz}}{\partial y} \delta p_{yz} dz - \iint_A \frac{\partial p_{yz}}{\partial y} \left(\frac{\partial \delta p_{yz}}{\partial y} \right) dydz \quad (E-3a)$$

$$\iint_A \frac{\partial^2 p_{yz}}{\partial z^2} \delta p_{yz} dydz = \int_S \frac{\partial p_{yz}}{\partial z} \delta p_{yz} dz - \iint_A \frac{\partial p_{yz}}{\partial z} \left(\frac{\partial \delta p_{yz}}{\partial z} \right) dydz \quad (E-3b)$$

Substitution of Eqs. E-3a and E-3b in Eq. E-2 gives

$$\iint_A \left[-\frac{\partial p_{yz}}{\partial y} \frac{\partial}{\partial y} (\delta p_{yz}) - \frac{\partial p_{yz}}{\partial z} \frac{\partial}{\partial z} (\delta p_{yz}) + \lambda^2 p_{yz} \delta p_{yz} \right] dy dz$$

$$- \int_S \frac{\partial p_{yz}}{\partial y} \delta p_{yz} dz + \int_S \frac{\partial p_{yz}}{\partial z} \delta p_{yz} dy = 0 \quad (E-4)$$

Since

$$\int_S \frac{\partial p_{yz}}{\partial n} \delta p_{yz} dS = - \int_S \frac{\partial p_{yz}}{\partial y} \delta p_{yz} dz + \int_S \frac{\partial p_{yz}}{\partial z} \delta p_{yz} dy \quad (E-5)$$

Equation E-4 can be further reduced to the following form:

$$\delta \left\{ \frac{1}{2} \iint_A \left[\left(\frac{\partial p_{yz}}{\partial y} \right)^2 + \left(\frac{\partial p_{yz}}{\partial z} \right)^2 - \lambda^2 p_{yz}^2 \right] dy dz \right\} - \int_S \frac{\partial p_{yz}}{\partial n} \delta p_{yz} dS = 0 \quad (E-6)$$

On the boundary S of the channel cross-section

$$\frac{\partial p_{yz}}{\partial n} = -i\omega\gamma p_{yz} - \frac{w}{g} a_n \quad (E-7)$$

where a_n = the acceleration along the outward normal to boundary S , γ = the damping coefficient and ω = the excitation frequency.

Using Eq. E-7 the last integral in Eq. E-6 may be rewritten as

$$\int_S \frac{\partial p_{yz}}{\partial n} \delta p_{yz} dS = \int_S \left(-i\omega\gamma p_{yz} - \frac{w}{g} a_n \right) \delta p_{yz} dS$$

$$= -\delta \int_S \left(\frac{1}{2} i\omega\gamma p_{yz}^2 + \frac{w}{g} a_n p_{yz} \right) dS \quad (E-8)$$

On substituting Eq. E-8 in Eq. E-6 the following relationship is obtained:

$$\delta \left\{ \frac{1}{2} \iint_A \left[\left(\frac{\partial p_{yz}}{\partial y} \right)^2 + \left(\frac{\partial p_{yz}}{\partial z} \right)^2 - \lambda^2 p_{yz}^2 \right] dy dz \right.$$

$$\left. + \int_S \left(\frac{1}{2} i\omega\gamma p_{yz}^2 + \frac{w}{g} a_n p_{yz} \right) dS \right\} = 0 \quad (E-9)$$

or

$$\delta\chi = 0 \quad (E-10)$$

where χ is a functional given by

$$\begin{aligned} \chi = & \frac{1}{2} \iint_A \left[\left(\frac{\partial p_{yz}}{\partial y} \right)^2 + \left(\frac{\partial p_{yz}}{\partial z} \right)^2 - \lambda^2 p_{yz}^2 \right] dydz \\ & + \frac{1}{2} i\omega\gamma \int_S p_{yz}^2 dS + \frac{w}{g} \int_S a_n p_{yz} dS \end{aligned} \quad (E-11)$$

The far boundary, which is also a cross-section of the infinite channel, is now divided into a number of finite elements and the functional χ is expressed in a discrete form as

$$\begin{aligned} \chi = & \frac{1}{2} \sum_{n=1}^M \left\{ \iint_{A_n} \left[\left(\frac{\partial p_{yz}}{\partial y} \right)^2 + \left(\frac{\partial p_{yz}}{\partial z} \right)^2 - \lambda^2 p_{yz}^2 \right] dydz \right\} \\ & + \frac{1}{2} i\omega\gamma \sum_{n=1}^{N_s} \int_{S_n} p_{yz}^2 dS_n + \frac{w}{g} \sum_{n=1}^{N_s} \int_{S_n} p_{yz} a_n dS_n \end{aligned} \quad (E-12)$$

where summation is taken over all the elements and A_n and S_n represent respectively the area and boundary of an individual element.

In the FEM representation p_{yz} is expressed in terms of nodal values and assumed shape functions

$$p_{yz} = \mathbf{N} \mathbf{p}_{yz} \quad (E-13)$$

where $\mathbf{N}$ = a row vector of shape functions, and $\mathbf{p}_{yz}$ the vector of the nodal values.

The following notations are also introduced

$$\frac{\partial}{\partial y} (\mathbf{N} \mathbf{p}_{yz}) = \frac{\partial \mathbf{N}}{\partial y} \mathbf{p}_{yz} = \mathbf{B}_1 \mathbf{p}_{yz} \quad (E-14a)$$

$$\frac{\partial}{\partial z} (\mathbf{N} \mathbf{p}_{yz}) = \frac{\partial \mathbf{N}}{\partial z} \mathbf{p}_{yz} = \mathbf{B}_2 \mathbf{p}_{yz} \quad (E-14b)$$

where $\mathbf{B}_1 = \frac{\partial \mathbf{N}}{\partial y}$ and $\mathbf{B}_2 = \frac{\partial \mathbf{N}}{\partial z}$

Using Eqs. E-13 and E-14, the functional in Eq. E-12 becomes

$$\begin{aligned}
\chi = & \sum_{n=1}^M \left(\frac{1}{2} \iint_{A_n} \mathbf{p}_{yz}^T \mathbf{B}_1^T \mathbf{B}_1 \mathbf{p}_{yz} dydz + \frac{1}{2} \iint_{A_n} \mathbf{p}_{yz}^T \mathbf{B}_2^T \mathbf{B}_2 \mathbf{p}_{yz} dydz \right. \\
& \left. - \frac{1}{2} \lambda^2 \iint_{A_n} \mathbf{p}_{yz}^T \mathbf{N}^T \mathbf{N} \mathbf{p}_{yz} dydz \right) + \sum_{n=1}^{N_s} \left(\frac{1}{2} i\omega\gamma \int_{S_n} \mathbf{p}_{yz}^T \mathbf{N}^T \mathbf{N} \mathbf{p}_{yz} dS_n \right. \\
& \left. + \frac{w}{g} \int_{S_n} a_n \mathbf{p}_{yz}^T \mathbf{N}^T dS_n \right)
\end{aligned} \quad (E-15)$$

The minimization of the functional χ can now be expressed

$$\frac{d\chi}{d(\mathbf{p}_{yz})} \delta(\mathbf{p}_{yz}) = 0 \quad (E-16)$$

or as

$$\frac{d\chi}{d(\mathbf{p}_{yz})} = 0 \quad (E-17)$$

On substituting for χ from Eq. E-15, Eq. E-17 gives

$$\begin{aligned}
& \sum_{n=1}^M \left[\iint_{A_n} (\mathbf{B}_1^T \mathbf{B}_1 + \mathbf{B}_2^T \mathbf{B}_2) dydz \mathbf{p}_{yz} - \lambda^2 \iint_{A_n} \mathbf{N}^T \mathbf{N} dydz \mathbf{p}_{yz} \right] \\
& + \sum_{n=1}^{N_s} \left(i\omega\gamma \int_{S_n} \mathbf{N}^T \mathbf{N} dS_n \mathbf{p}_{yz} + \frac{w}{g} \int_{S_n} \mathbf{N}^T a_n dS_n \right) = 0
\end{aligned} \quad (E-18)$$

In order to derive the shape functions $\mathbf{N}$, the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$, the assumed variation of pressure over an element must be determined. In the present study, a 4-node isoparametric, quadrilateral element is used.

To derive the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$, the assumed variation of pressure p_{yz} over an element must be determined. A linear variation of the pressure variable p_{yz} is assumed. The global coordinates y, z are also assumed to vary linearly in the transformation to the mapped ξ, η system as shown in Fig. E.1.

With these assumptions the pressure p_{yz} can be expressed in terms of its nodal values as follows:

Fig. E.1 Unmapped and Mapped 4-Node Quadrilateral Isoparametric Element

$$p_{yx} = \sum_{i=1}^4 N_i(\xi, \eta) p_i \quad (E-19)$$

$$z = \sum_{i=1}^4 N_i(\xi, \eta) z_i \quad (E-20)$$

$$y = \sum_{i=1}^4 N_i(\xi, \eta) y_i \quad (E-21)$$

where the shape functions N are given by

$$N_i(\xi, \eta) = \frac{1}{4} (1 + \xi \xi_i)(1 + \eta \eta_i) \quad (E-22)$$

$i = 1, 2, 3, 4$

Before the integrals in Eqs. E.15, can be evaluated matrices N , B_1 , B_2 must be determined. The row vector of shape functions is given by

$$N = \begin{pmatrix} N_1 & N_2 & N_3 & N_4 \end{pmatrix} \quad (E - 23)$$

where $N_i(\xi, \eta)$ are described by Eqs. E-22.

The derivatives $\frac{\partial N}{\partial x}$ and $\frac{\partial N}{\partial y}$ in matrices B_1 and B_2 are obtained as follows:

$$\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \eta} & \frac{\partial x}{\partial \eta} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} \end{pmatrix} = [J] \begin{pmatrix} \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} \end{pmatrix} \quad (E - 24)$$

Inverting Equation E-24

$$\begin{pmatrix} \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} \end{pmatrix} = \frac{1}{|J|} \begin{pmatrix} \frac{\partial x}{\partial \eta} & -\frac{\partial x}{\partial \xi} \\ -\frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \xi} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} \quad (E - 25)$$

where $|J| = \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta}$. Matrix $[J]$ is called the Jacobian matrix and the determinant is referred to as the Jacobian.

Also from Eqs. E-20, E-21 and E-22

$$\frac{\partial y}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i = \sum_{i=1}^4 \frac{1}{4} (1 + \xi_i)(1 + \eta \eta_i) y_i \quad (E - 26a)$$

$$\frac{\partial y}{\partial \eta} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i = \sum_{i=1}^4 \frac{1}{4} (1 + \xi \xi_i)(1 + \eta_i) y_i \quad (E - 26b)$$

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} z_i = \sum_{i=1}^4 \frac{1}{4} (1 + \xi_i)(1 + \eta \eta_i) z_i \quad (E - 26c)$$

$$\frac{\partial x}{\partial \eta} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} z_i = \sum_{i=1}^4 \frac{1}{4} (1 + \xi \xi_i)(1 + \eta_i) z_i \quad (E - 26d)$$

Matrices B_1 and B_2 are now obtained from

$$B_1 = \frac{\partial N}{\partial y} = \frac{1}{|J|} \left(\frac{\partial z}{\partial \eta} \frac{\partial N}{\partial \xi} - \frac{\partial z}{\partial \xi} \frac{\partial N}{\partial \eta} \right) \quad (E - 27a)$$

$$B_2 = \frac{\partial N}{\partial x} = \frac{1}{|J|} \left(-\frac{\partial y}{\partial \eta} \frac{\partial N}{\partial \xi} + \frac{\partial y}{\partial \xi} \frac{\partial N}{\partial \eta} \right) \quad (E - 27b)$$

Introducing the notations

$$c_i(\xi, \eta) = \frac{\partial z}{\partial \eta} \frac{\partial N_i}{\partial \xi} - \frac{\partial z}{\partial \xi} \frac{\partial N_i}{\partial \eta} \quad (E - 28a)$$

$$d_i(\xi, \eta) = -\frac{\partial y}{\partial \eta} \frac{\partial N_i}{\partial \xi} + \frac{\partial y}{\partial \xi} \frac{\partial N_i}{\partial \eta} \quad (E - 28b)$$

expressions for matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ can be written in the following compact form:

$$\mathbf{B}_1 = \frac{1}{|J|} [c_1 \quad c_2 \quad c_3 \quad c_4] \quad (E - 29a)$$

$$\mathbf{B}_2 = \frac{1}{|J|} [d_1 \quad d_2 \quad d_3 \quad d_4] \quad (E - 29b)$$

Before the integrals in Eq. E-18 can be evaluated the differentials $dy \, dz$ must be expressed in terms of ξ, η . This is easily accomplished by noting that the following relationship holds

$$dy \, dz = |J| \, d\xi \, d\eta \quad (E - 30)$$

where $|J|$ is the Jacobian already derived.

The integrals in Eq. E-18 can now be evaluated by any numerical method. A four point Gauss quadrature is used in this work. Equation E-18 can be rewritten in the following matrix form:

$$[\mathbf{A} + i\omega\gamma \mathbf{B} - \lambda^2 \mathbf{F}] \mathbf{p}_{yz} = \frac{w}{g} \mathbf{d} \quad (E - 31)$$

**BOUNDARY ELEMENT ANALYSIS OF EARTHQUAKE
INDUCED HYDRODYNAMIC PRESSURES
IN A WATER RESERVOIR**

by

ALEXANDER MACIEJ JABŁOŃSKI

B.Sc., M.Eng. - Technical University of Cracow

M.Sc. - University of Illinois

**A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Civil Engineering**

November 1988

**Department of Civil Engineering
Faculty of Engineering
Carleton University
Ottawa, Canada**

Copyright © Alexander Maciej Jablonski, 1988

CONTENTS

VOLUME 2

1. User's Guide to Boundary Element Programs	1
1.1 General Comments	1
1.2 Constant Element Program for 2D Reservoirs - BEMC.2D	1
1.2.1 Description of the BEMC.2D Program	1
1.2.2 Input Data	5
1.2.3 Output	6
1.2.4 Example	6
1.2.5 Listing of the BEMC.2D Program	11
1.3 Linear Element Program for 2D Reservoirs - BEML.2D	57
1.3.1 Description of the BEML.2D Program	57
1.3.2 Input Data	59
1.3.3 Output	60
1.3.4 Example	60
1.3.5 Listing of the BEML.2D Program	65
1.4 Constant Element Program for 3D Reservoirs - BEMC.3D	110
1.4.1 Description of the BEMC.3D Program	110
1.4.2 Input Data	113
1.4.3 Output	115
1.4.4 Example	116
1.4.5 Listing of the BEMC.3D Program	124
2. User's Guide to Superposition Programs	183
2.1 General Comments	183
2.2 Superposition Program for 2D Finite Reservoirs - SUP.2D	183
2.2.1 Description of the SUP.2D Program	183
2.2.2 Input Data	185
2.2.3 Output	185
2.2.4 Example	186
2.2.5 Listing of the SUP.2D Program	190
2.3 Superposition Program for 3D Finite Reservoirs - SUP.3D	198
2.3.1 Description of the SUP.3D Program	198

2.3.2	<i>Input Data</i>	199
2.3.3	<i>Output</i>	199
2.3.4	<i>Example</i>	200
2.3.5	<i>Listing of the SUP.3D Program</i>	206
References		213

VOLUME 2

CHAPTER 1

USER'S GUIDE TO BOUNDARY ELEMENT PROGRAMS

1.1 General Comments

To meet objectives of this thesis a number of computer programs had to be developed. In this chapter three programs based on the boundary element method are presented. At the beginning of this chapter, the computer program BEMC.2D based on the constant boundary element 2D formulation is briefly described. A simple example, including input and output files, is also presented. Next, the computer program BEML.2D based on the linear boundary element 2D formulation together with an example, is described. Finally, the computer program BEMC.3D based on the constant boundary element 3D formulation is described. It is followed by an example. The listings of these three programs are also included. All programs are written in FORTRAN77.

1.2 Constant Element Program for 2D Reservoirs - BEMC.2D

The constant element program for 2D reservoirs - BEMC.2D is based on the formulation presented in detail in Chapters 2 and 3 of Volume 1.

1.2.1 Description of the BEMC.2D Program

This FORTRAN computer program can be used to calculate the hydrodynamic pressure on a rigid dam storing a two-dimensional finite or infinite water reservoir and subjected to

a horizontal or/and vertical harmonic motion. The program uses the constant boundary elements. The pressures and the pressure derivatives are assumed to have constant values over the element and equal to their values at the mid-node of each element.

There are no limitations on the shape of the dam or the finite reservoir. The infinite reservoir, however, has to be divided to two parts: one finite and irregular and other regular but infinite. The upstream surface of the dam, the surface, the bottom and the upstream end of the reservoir (in case of the infinite reservoir this end is an interface between irregular and regular parts), are to be covered by a constant boundary element mesh.

The main program defines the limiting number of boundary elements that it can handle. The number is set to 101 as shown in the program listing. The main program calls the eight following subroutines:

INPUT - to read all data and print the first part of the output.

EIGEN - to set and calculate the complex eigenvalue problem in case of infinite radiation with foundation damping included.

EIGENR - to set and to calculate the real eigenvalue problem in case of infinite radiation only (undamped case).

FMAT - to form matrices **G** and **H** and rearrange them according to the boundary conditions on the top of the reservoir.

ADDFBC - to modify the matrix **H** for the boundary conditions on the bottom of the reservoir and at its far end, and to compute the vector of known values of the R.H.S. of the system of equations. This subroutine forms the system of equations for the damped case.

ADDUND - to form the system of equations and apply all boundary conditions required for the undamped case.

SLNPD - to solve the system of equations by Gauss elimination.

OUTPUT - to reorder the vector of computed unknowns and print out the rest of the output file.

The subroutine **EIGEN** calls the subroutine

FSFEM - to compute the real and imaginary parts of the eigenvector matrix and at the same time to create the real and imaginary parts of the eigenvalue vector.

The subroutine FSFEM calls the following subroutines:

- SET - to set the complex matrices for the eigenvalue problem, which governs the damped vibrations in the infinite part of the reservoir: $(\mathbf{A} + i\omega\gamma \mathbf{B}) \mathbf{p} = \lambda^2 \mathbf{F} \mathbf{p}$.
- CQZHES - this subroutine is a complex analogue of the first step of the QZ algorithm for solving generalized matrix eigenvalue problems (1). This subroutine accepts a pair of complex general matrices and reduces one of them to upper Hessenberg form with real (and non-negative) subdiagonal elements and the other to upper triangular form using unitary transformations.
- CQZVAL - this subroutine is a complex analogue of steps 2 and 3 of the QZ algorithm for solving generalized matrix eigenvalue problems. This subroutine accepts a pair of complex matrices, one of them in upper Hessenberg form and the other in upper triangular form. The Hessenberg matrix must further have real subdiagonal elements. It reduces the Hessenberg matrix to triangular form using unitary transformations while maintaining the triangular form of the other matrix and making its diagonal elements real and non-negative. It then returns quantities whose ratios give the generalized eigenvalues.
- CQZVEC - this subroutine is a complex analogue of the fourth step of the QZ algorithm for solving generalized matrix eigenvalue problems. This subroutine accepts a pair of complex matrices in upper triangular form, where one of them further must have real diagonal elements. It computes the eigenvectors of the triangular problem and transforms the results back to the original system. This is last step of QZ algorithm originally developed by Stewart and Moler (1) and modified by B.S. Garbov from Applied Mathematics Division, Argonne National Laboratory, Illinois, U.S.A.
- NORMA - this subroutine normalizes the eigenvectors obtained from the QZ algorithm w.r.t. matrix $\mathbf{F}$.

The subroutine EIGENR calls the following subroutines:

- SETR - to set the real matrices for the eigenvalue problem, which governs the undamped vibrations in the infinite part of the reservoir: $\mathbf{A} \mathbf{p} = \lambda^2 \mathbf{F} \mathbf{p}$.
- JACOBI - this subroutine uses the generalized Jacobi method to solve the generalized real matrix eigenvalue problem (2). It calculates the normalized eigenvectors and eigenvalues.

The subroutine FMAT calls the following subroutines:

INTE - to compute the **G** and **H** matrix elements except those on the diagonal by means of numerical integration along the boundary elements. The numerical integration uses the 4-point Gauss quadrature scheme.

INLO - to calculate the diagonal elements of matrix **G**.

The subroutine INTE calls the subroutine:

BFUNCT - to compute the values of Bessel function $Y_0(kr)$ and its derivative by means of series representation of the Bessel functions (3,4).

Subroutines ADDFBC, ADDUND and OUTPUT call the subroutine:

TP - to calculate the transformation matrix $\mathbf{T}_p$ for adjustment of linearly formed far end condition into constant element formulation.

1.2.2 Input Data.

The following table shows the order of presentation of data in BEMC.2D program.

TABLE 1-2

Input Data for the BEMC.2D Program

Card Group	No. of Cards	Format	FORTTRAN variables
A	1	18A4	TITLE
B1	1	General Format	N = number of boundary elements N0 = last elem. number at the top near the dam N1 = last elem. number at the base of the dam N2 = last elem. number at the base near far end NOUT = 1, flag for detailed output
B2	1	General Format	HY = height of the dam C = velocity of sound in water ALFA = wave reflection coefficient DF = vertical acceleration of the first node at the far end
C	N	General Format	X, Y = coordinates of nodal points
D	N1-N0	General Format	FID = boundary conditions on the dam face
E	N2-N1	General Format	FIB = boundary conditions on the foundation base
F	No. of cases	General Format	OMEGA = excitation frequency

The node numbers must start from the element at the top of the reservoir adjacent to the far end and numbered in counterclockwise direction looking into the reservoir scheme.

1.2.3 Output Data.

The program prints out the input data as well as the hydrodynamic pressure value at each boundary element mid-node. The real and imaginary parts of both, the pressure and its derivative are given. The real and imaginary parts of the total pressure acting on the dam face as well as its absolute maximum are also given. Intermediate results, especially for an eigenvalue problem are possible to be obtained.

1.2.4 Example.

For the purpose of illustration, a simple example is presented in this subsection.

A two-dimensional infinite reservoir impounded by a straight gravity dam is subjected to a horizontal harmonic motion with excitation frequency equal to 27.144 rad/s ($\omega/\omega_1 = 1.2$). The reservoir is modelled using 20 constant boundary element mesh as shown in Fig. 1.1. The far end is modelled by 5 linear one-dimensional finite elements. The flexible bottom of the reservoir has the coefficient of reflection $\alpha_R = 0.75$ (25% of waves is absorbed in the bottom). The reservoir dimensions are also shown in Fig.1.1.

Fig. 1.1 The Constant Boundary Element Model of Infinite 2D Reservoir - $N = 20$

First, the input file is listed and it is followed by the output file.

• Input file - TEST2DC

2D INFINITE RESERVOIR BY THE BEMC

20 5 10 15 1

100. 1440. 0.75 0.0

100.0 100.0

80.0 100.0

60.0 100.0

40.0 100.0

20.0 100.0

0.0 100.0

0.0 80.0

0.0 60.0

0.0 40.0

0.0 20.0

0.0 0.0

20.0 0.0

40.0 0.0

60.0 0.0

80.0 0.0

100.0 0.0

100.0 20.0

100.0 40.0

100.0 60.0

100.0 80.0

0.101937

0.101937

0.101937

0.101937

0.101937

0.0

0.0

0.0

0.0

0.0

27.144

• Output file - OUT2DC

DATA

NUMBER OF BOUNDARY ELEMENTS 20

ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DAM 5

ELEMENT NUMBER AT BASE OF DAM 10

ELEMENT NUMBER AT BOTTOM NEAR FAR END 15

FLAG FOR DETAILED OUTPUT 1

HEIGHT OF DAM 0.10000E+03

VELOCITY OF SOUND IN WATER 0.14400E+04

WAVE REFLECTION COEFFICIENT 0.75000E+00

FOUNDATION DAMPING COEFFICIENT 0.99206E-04

COORDINATES OF THE NODAL POINTS

POINT	X	Y
1	0.1000000E+03	0.1000000E+03
2	0.8000000E+02	0.1000000E+03
3	0.6000000E+02	0.1000000E+03
4	0.4000000E+02	0.1000000E+03
5	0.2000000E+02	0.1000000E+03
6	0.0000000E+00	0.1000000E+03
7	0.0000000E+00	0.8000000E+02
8	0.0000000E+00	0.6000000E+02
9	0.0000000E+00	0.4000000E+02
10	0.0000000E+00	0.2000000E+02
11	0.0000000E+00	0.0000000E+00
12	0.2000000E+02	0.0000000E+00
13	0.4000000E+02	0.0000000E+00
14	0.6000000E+02	0.0000000E+00
15	0.8000000E+02	0.0000000E+00
16	0.1000000E+03	0.0000000E+00
17	0.1000000E+03	0.2000000E+02
18	0.1000000E+03	0.4000000E+02
19	0.1000000E+03	0.6000000E+02
20	0.1000000E+03	0.8000000E+02

BOUNDARY CONDITIONS ON THE DAM FACE

NODE	PRESCRIBED VALUE
1	0.1019370E+00
2	0.1019370E+00
3	0.1019370E+00
4	0.1019370E+00
5	0.1019370E+00

BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM

NODE	PRESCRIBED VALUE
1	0.0000000E+00
2	0.0000000E+00
3	0.0000000E+00
4	0.0000000E+00
5	0.0000000E+00

EXCITING FREQUENCY 0.27144E+02

WAVE CONSTANT 0.18850E-01

THE DETERMINANT =0.14892186E-06

PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE

BOUNDARY ELEMENTS

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			

0.90000E+02	0.10000E+03	0.00000E+00	0.89824E-01
0.70000E+02	0.10000E+03	0.00000E+00	0.49400E-01
0.50000E+02	0.10000E+03	0.00000E+00	0.15315E-01
0.30000E+02	0.10000E+03	0.00000E+00	-0.26358E-01
0.10000E+02	0.10000E+03	0.00000E+00	-0.14717E+00

IMAGINARY PART

0.90000E+02	0.10000E+03	0.00000E+00	0.13186E+00
0.70000E+02	0.10000E+03	0.00000E+00	0.14909E+00
0.50000E+02	0.10000E+03	0.00000E+00	0.16890E+00
0.30000E+02	0.10000E+03	0.00000E+00	0.17841E+00
0.10000E+02	0.10000E+03	0.00000E+00	0.19382E+00

PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE

BOUNDARY ELEMENTS

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			

0.00000E+00	0.90000E+02	0.17458E+01	0.10194E+00
0.00000E+00	0.70000E+02	0.31245E+01	0.10194E+00
0.00000E+00	0.50000E+02	0.35240E+01	0.10194E+00
0.00000E+00	0.30000E+02	0.33719E+01	0.10194E+00
0.00000E+00	0.10000E+02	0.28334E+01	0.10194E+00

IMAGINARY PART

0.00000E+00	0.90000E+02	-0.16679E+01	0.00000E+00
0.00000E+00	0.70000E+02	-0.53425E+01	0.00000E+00
0.00000E+00	0.50000E+02	-0.84477E+01	0.00000E+00
0.00000E+00	0.30000E+02	-0.10672E+02	0.00000E+00
0.00000E+00	0.10000E+02	-0.11799E+02	0.00000E+00

PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM

BOUNDARY ELEMENTS

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			
0.10000E+02	0.00000E+00	0.16605E+01	-0.32045E-01
0.30000E+02	0.00000E+00	-0.49325E+00	-0.30824E-01
0.50000E+02	0.00000E+00	-0.25256E+01	-0.28306E-01
0.70000E+02	0.00000E+00	-0.42966E+01	-0.24809E-01
0.90000E+02	0.00000E+00	-0.56882E+01	-0.20663E-01
IMAGINARY PART			
0.10000E+02	0.00000E+00	-0.11900E+02	-0.44716E-02
0.30000E+02	0.00000E+00	-0.11446E+02	0.13283E-02
0.50000E+02	0.00000E+00	-0.10511E+02	0.68012E-02
0.70000E+02	0.00000E+00	-0.92130E+01	0.11570E-01
0.90000E+02	0.00000E+00	-0.76732E+01	0.15318E-01

PRESSURE AND PRESSURE DERIVATIVE ON FAR END

BOUNDARY ELEMENTS

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			
0.10000E+03	0.10000E+02	-0.61733E+01	-0.50168E-01
0.10000E+03	0.30000E+02	-0.52148E+01	-0.56586E-01
0.10000E+03	0.50000E+02	-0.39367E+01	-0.49475E-01
0.10000E+03	0.70000E+02	-0.24687E+01	-0.29665E-01
0.10000E+03	0.90000E+02	-0.80881E+00	-0.14894E-01
IMAGINARY PART			
0.10000E+03	0.10000E+02	-0.68642E+01	0.76798E-01
0.10000E+03	0.30000E+02	-0.64996E+01	0.75312E-01
0.10000E+03	0.50000E+02	-0.53018E+01	0.60397E-01
0.10000E+03	0.70000E+02	-0.33995E+01	0.35292E-01
0.10000E+03	0.90000E+02	-0.11197E+01	0.49394E-02

TOTAL PRESSURE ON DAM FACE, REAL PART 0.29199E+03

TOTAL PRESSURE ON DAM FACE, IMAGINARY PART -0.75858E+03

TOTAL PRESSURE ON DAM FACE, ABSOLUTE VALUE 0.81283E+03

Note: The values of pressure and pressure derivative are normalized with respect to the water specific gravity w . These values are to be divided by the dam height H to be normalized w.r.t. hydrostatic pressure $w H$.

1.2.5 Listing of the BEMC.2D Program.

User: jablonski

Node: 16156

Date: Sat Nov 12 21:00:06 1988

File: bemc.2d.ftn

Pages: 46

PROGRAM BEMC.2D.I.FTN

PROGRAM FOR THE SOLUTION OF A TWO-DIMENSIONAL RESERVOIR
PROBLEM WITH ONE INFINITE BOUNDARY AND DAMPING INCLUDED
BY CONSTANT BOUNDARY ELEMENT METHOD
SPECIAL B.C. AT FAR BOUNDARY THROUGH FEM, DAMPING INCLUDED

PROGRAMMER: ALEXANDER M. JABLONSKI

DATE: JULY 9, 1987,
SUBSTANTIALLY MODIFIED AUGUST 27, 1987

CENTER: CASCADE, CARLETON UNIVERSITY

NOTE: THIS PROGRAM IS BASED ON PROGRAM BEML.2D.I.FTN

N = NUMBER OF BOUNDARY ELEMENTS, READ THROUGH INPUT
X = ARRAY CONTAINING X COORDINATES OF THE NODES
Y = ARRAY CONTAINING Y COORDINATES OF THE NODES
XM = ARRAY CONTAINING X COORDINATES OF THE MIDPOINTS OF THE ELEMENTS
YM = ARRAY CONTAINING Y COORDINATES OF THE MIDPOINTS OF THE ELEMENTS
G = MATRIX IN THE BOUNDARY ELEMENT EQUATION
H = MATRIX IN THE BOUNDARY ELEMENT EQUATION
FID= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS
ON THE DAM FACE
FIB= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS
ON THE BASE
DFI = RIGHT HAND SIDE VECTOR IN BOUNDARY INTEGRAL EQUATION;
AFTER SOLUTION, IT CONTAINS THE VALUES AT THE BOUNDARY
THAT WERE PREVIOUSLY UNKNOWN
LEC = UNIT NO. CONTAINING INPUT
IMP = UNIT NO. ON WHICH OUTPUT IS WRITTEN
NX = 2*N = DIMENSION OF THE SYSTEM OF EQUATIONS
NU = DIMENSION OF H AND G MATRICES IN THE MAIN PROGRAM
ND = DIMENSION OF THE SPECIAL BOUNDARY CONDITION MATRIX AT
FAR BOUNDARY IN THE MAIN PROGRAM
D = VALUE OF THE DETERMINANT OF COEFFICIENT MATRIX
CONST = OMEGA/C
OMEGA = EXCITING FREQUENCY
C = VELOCITY OF SOUND IN WATER
CONST1=OMEGA
CONST2=FOUNDATION DAMPING COEFFICIENT GAMMA
AFR=REAL PART OF EIGENVECTOR SI
AFI=IMAGINARY PART OF EIGENVECTOR SI
AKR=REAL PART OF KAPPA MATRIX
AKI=IMAGINARY PART OF KAPPA MATRIX
N0=LAST ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DA.
N1=LAST ELEMENT NUMBER AT THE BASE OF THE DAM
N2=LAST ELEMENT NUMBER AT THE BASE NEAR FAR END
MF=NUMBER OF ELEMENTS ON BASE
ME=NUMBER OF ELEMENTS ON THE FAR BOUNDARY
HY=HEIGHT OF THE RESERVOIR

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON,MAIN/ CONST,N,LEC,IMP,NU,ND

```

COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
DOUBLE PRECISION X(101),Y(101),G(200,200),H(200,200),
&DFI(200),FID(50),FIB(50),HOLD(200,200),XM(101),YM(101)
REAL AFI(20,20),AFR(20,20),AKR(20),AKI(20)
LOGICAL FLAG
LEC=2
IMP=3
ND=20
NU=200
PI=3.141592653
OPEN (LEC,FORM='FORMATTED',STATUS='OLD',FILE='TEST2DC')
OPEN (IMP,FORM='FORMATTED',STATUS='NEW',FILE='OUT2DC')
OPEN (6,FORM='FORMATTED',STATUS='NEW',FILE='EIGEN2DC')

C
C   READ DATA FROM INPUT FILE
C
CALL INPUT(X,Y,FID,FIB,N0,N1,N2,NOUT,C,DF)
NX=2*N
5 WRITE(IMP,300)
READ(LEC,*,END=15)OMEGA
CONST=OMEGA/C
WRITE(IMP,350)OMEGA,CONST
CONST1=OMEGA

C
C   COMPUTE EIGENVALUES AND EIGENVECTORS FOR THE FAR BOUNDARY
C
IF (ME.GT.0) THEN
IF (CONST2.GT.1.0E-08) THEN
FLAG=.TRUE.
CALL EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI)
ELSE
CALL EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI,IMP,FLAG)
END IF
write(6,2001)
2001 format(' ')

*   DO 10 I=1,ME
*   WRITE(6,2010)AKR(I),AKI(I)
* 10 CONTINUE
*   write (6,2001)
*   do 2003 i=1,me
*   write(6,2010) (afr(i,j),j=1,me)
* 2003 continue
*   write(6,2001)
*   END IF

C
C   FORM SYSTEM OF EQUATIONS
C
C
CALL FMAT(X,Y,XM,YM,G,H,HOLD,FID,FIB,DFI,N0,N1)
*   do 2000 i=1,n
*   write(6,2010) (h(i,j),j=1,n)
* 2000 continue
*   write(6,2001)
*   write(6,2001)
*   do 2002 i=1,n
*   write(6,2010) (g(i,j),j=1,n)
* 2002 continue
C
C   MODIFY H MATRIX FOR BOUNDARY CONDITIONS AT BASE AND FAR END

```

4

MicroD

```

C      AND COMPUTE DFI
C
      IF (CONST2.GT.1.0E-08) THEN
      CALL ADDDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,DFI,FID,FIB,DF)
      ELSE
      CALL ADDUND(H,G,HOLD,AFR,AKR,AKI,N0,N1,N2,DFI,FID,FIB,FLAG,DF)
      END IF
*      write(6,2001)
*      write(6,2001)
*      do 2005 i=1,nx
*      write(6,2010) (h(i,j),j=1,nx)
* 2005 continue
*      write(6,2001)
*      write(6,2001)
*      write(6,2010) (dfi(i),i=1,nx)
C
C
C      SOLVE SYSTEM OF EQUATIONS
C
      IF (FLAG) THEN
      CALL SLNPD(H,DFI,D,NX,NU,IMP)
      ELSE
      CALL SLNPD(H,DFI,D,N,NU,IMP)
      END IF
*      DO 999 I=1,N
*      WRITE(IMP,*)DFI(I)
* 999 CONTINUE
*      IF (FLAG) THEN
*      DO 1000 I=1,N
*      WRITE(IMP,*)DFI(I+N)
* 1000 CONTINUE
*      END IF
C
C
C      OUTPUT RESULTS
C
      CALL OUTPUT(XM,YM,FID,FIB,DFI,AFR,AFI,AKR,AKI,
* N0,N1,N2,NOUT,FLAG,DF)
      GO TO 5
2010 format(6e13.5)
300 format(//,1X,70('*'),//)
350 format(1X,'EXCITING FREQUENCY',E14.5,/,1X,
*'WAVE CONSTANT',E14.5,/)
15 STOP
END
C
C
C      -----
C
C      SUBROUTINE INPUT(X,Y,FID,FIB,N0,N1,N2,NOUT,C,DF)
C      *****
C
C      READS INPUT DATA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION X(1),Y(1),FID(1),FIB(1),C
      CHARACTER*4 TITLE(18)
      WRITE (IMP,100)
C

```

```

C   READ TITLE
C
C   READ (LEC,150)TITLE
C
C   READ BASIC PARAMETERS
C
C   READ (LEC,*)N,N0,N1,N2,NOUT
C   READ (LEC,*)HY,C,ALF,DF
C   CONST2=((1-ALF)/(1+ALF))*(1/C)
C
C   WRITE (IMP,300)N,N0,N1,N2,NOUT
C   WRITE (IMP,350)HY,C,ALF,CONST2
C
C
C   READ COORDINATES OF NODAL POINTS OF THE
C   BOUNDARY ELEMENTS IN ARRAY X AND Y
C
C   WRITE (IMP,500)
C   DO 10 I=1,N
C   READ (LEC,*)X(I),Y(I)
10  WRITE (IMP,700)I,X(I),Y(I)
C
C
C   READ BOUNDARY CONDITION ON THE DAM FACE
C
C   WRITE (IMP,760)
C   WRITE (IMP,800)
C   ILAST=N1-N0
C   DO 20 I=1,ILAST
C   READ (LEC,*)FID(I)
C   WRITE (IMP,750)I,FID(I)
20  CONTINUE
C
C
C   READ BOUNDARY CONDITION ON THE BASE OF THE RESERVOIR
C
C   WRITE (IMP,770)
C   WRITE (IMP,800)
C   ILAST=N2-N1
C   DO 30 I=1,ILAST
C   READ (LEC,*)FIB(I)
C   WRITE (IMP,750)I,FIB(I)
30  CONTINUE
C   ME=N-N2
C   MF=N2-N1
C
100  FORMAT(1X,60(' '))
150  FORMAT(18A4)
300  FORMAT(1X,'DATA',//,1X,'NUMBER OF BOUNDARY ELEMENTS',
*   I3,/,1X,'ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DAM',I3,/,
*   1X,'ELEMENT NUMBER AT BASE OF DAM',I3,/,1X,
*   'ELEMENT NUMBER AT BOTTOM NEAR FAR END',I3,/,1X,
*   'FLAG FOR DETAILED OUTPUT',I3,/)
350  FORMAT(1X,'HEIGHT OF DAM',E14.5,/,1X,
*   'VELOCITY OF SOUND IN WATER',E14.5,/,1X,
*   'WAVE REFLECTION COEFFICIENT',E14.5,/,1X,
*   'FOUNDATION DAMPING COEFFICIENT',E14.5//)
400  FORMAT(2(2X,E14.7,2X))
500  FORMAT(' COORDINATES OF THE NODAL POINTS',
*   //,4X,'POINT',10X,'X',18X,'Y'/)

```

```

700 FORMAT(5X,I3,2(5X,E14.7))
800 FORMAT(5X,'NODE',6X,
  *'PRESCRIBED VALUE'//)
750 FORMAT(5X,I3,8X,E14.7)
760 FORMAT(///' BOUNDARY CONDITIONS ON THE DAM FACE',//)
770 FORMAT(///' BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM',//)
C
C
      RETURN
      END
C
C
      -----
C
      SUBROUTINE EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI)
C
      *****
      DOUBLE PRECISION CONST,CONST1,CONST2,HY
      REAL AFI(ND,ND),AFR(ND,ND),AKR(NM),AKI(ND)
C
      DY=SNGL(HY/ME)
      T=SNGL(CONST)
      T1=SNGL(CONST1)
      T2=SNGL(CONST2)
C
      CALL FSFEM(AFI,AFR,ME,ND,T,T1,T2,DY,AKR,AKI)
C
      RETURN
      END
C
C
      -----
C
      SUBROUTINE FSFEM(AFI,AFR,N,NM,T,T1,T2,DY,AKR,AKI)
C
      *****
C
      AFR= REAL PART OF THE EIGENVECTOR MATRIX
      AFI= IMAGINARY PART OF EIGENVECTOR MATRIX
      AKR=REAL PART OF KAPPA MATRIX DIAGONAL
      AKI=IMAGINARY PART OF KAPPA MATRIX DIAGONAL
      N=ME= NO. OF ELEMENTS ON THE FAR BOUNDARY
      NM=ND= MAX. DIMENSION OF TNE EIGENVECTOR MATRIX
      DY= PASSED LENGTH OF AN ELEMENT ( DP IN THE MAIN PROGRAM )
      T=CONST= OMEGA/C ( DP IN THE MAIN PROGRAM )
      T1=CONST1= OMEGA ( DP IN THE MAIN PROGRAM )
      T2=CONST2=GAMMA
C
      REAL A( 20, 20),B( 20, 20),Z( 20, 20),BHOLD( 20, 20),
$          AHOLD( 20, 20),AHOLDI( 20, 20),BHOLDI( 20,20),
$          ALFR( 20),ALFI( 20),BETA( 20),
$          EPS1, AI(20,20),BI(20,20),ZI(20,20)
      REAL AMAX1
      COMPLEX FI(20,20),XM(20,20)
      COMPLEX XKAP(20),XK2(20)
      REAL AFR(NM,NM),AFI(NM,NM),AKR(NM),AKI(NM)
      INTEGER ERROR
      DIMENSION INDEX(20)
      DATA IWRITE/6/
C
      CALL SET(DY,T1,T2,N,NM,A,AI,B,BI)
      DO 717 I=1,N
      DO 718 J=1,N
      AHOLD(I,J)=A(I,J)

```

```

      BHOLD(I,J)=B(I,J)
      AHOLDI(I,J)=AI(I,J)
      BHOLDI(I,J)=BI(I,J)
718 CONTINUE
717 CONTINUE
      WRITE(IWRITE,20)
      $      N
20  FORMAT(30H1THE FULL MATRIX A OF ORDER,
      $      I4,22H IS (PRINTED BY ROWS)/)
      DO 30 I = 1,N
30  WRITE(IWRITE,40)
      $      (A(I,J),AI(I,J),J=1,N)
40  FORMAT(5(2F8.4,1HI,3X))
      WRITE(IWRITE,41)
      $      N
41  FORMAT(30H0THE FULL MATRIX B OF ORDER,
      $      I4,22H IS (PRINTED BY ROWS)/)
      DO 42 I = 1,N
42  WRITE(IWRITE,43)
      $      (B(I,J),BI(I,J),J=1,N)
43  FORMAT(5(2F8.4,1HI,3X))

C
C
C
C
C
      EIGENVALUES AND EIGENVECTORS USING CQZVAL AND CQZVEC

      EPS1 = 0.0
      CALL CQZHES(NM,N,A,AI,B,BI,.TRUE.,Z,ZI)
      CALL CQZVAL(NM,N,A,AI,B,BI,EPS1,ALFR,ALFI,BETA,
      $      .TRUE.,Z,ZI,ERROR)
      CALL CQZVEC(NM,N,A,AI,B,BI,ALFR,ALFI,BETA,Z,ZI)
      DO 719 I=1,N
      DO 720 J=1,N
      A(I,J)=AHOLD(I,J)
      B(I,J)=BHOLD(I,J)
      AI(I,J)=AHOLDI(I,J)
      BI(I,J)=BHOLDI(I,J)
720 CONTINUE
719 CONTINUE
      WRITE(IWRITE,292)
292  FORMAT(/15X,7HALFR(I),19X,7HALFI(I),19X,7HBETA(I))
      DO 295 I = 1,N
      WRITE(IWRITE,293)
      $      I,ALFR(I),ALFI(I),BETA(I)
293  FORMAT(I2,3(1PE23.6,3X))
295  CONTINUE
      WRITE(IWRITE,300)
300 FORMAT(/14X,35HCOMPUTED EIGENVALUE AND EIGENVECTOR,20X)

C
      DO 510 K = 1, N
      BETA(K) = AMAX1(BETA(K),1.0E-50)
      ALFR(K) = ALFR(K) / BETA(K)
      ALFI(K) = ALFI(K) / BETA(K)

C
C
C
      ONE EIGENVECTOR.

340  WRITE(IWRITE,350)
      $      K,ALFR(K),ALFI(K),(Z(I,K),
      $      ZI(I,K),I=1,N)

```

```

350     FORMAT(/I2,1P2E23.6/(5X,2E23.6))
510 CONTINUE
    DO 610 K=1,N
    DO 620 L=1,N
    FI(K,L)=CMPLX(Z(K,L),ZI(K,L))
620 CONTINUE
610 CONTINUE
    DO 611 I=1,N
    WRITE(IWRITE,*) (FI(I,J),J=1,N)
611 CONTINUE
    CALL NORMA(FI,B,N,NM,XM)
    DO 81 I=1,N
    WRITE(IWRITE,*) (FI(I,J),J=1,N)
    81 CONTINUE
    DO 613 I=1,N
    DO 612 J=1,N
    AFR(I,J)=REAL(FI(I,J))
    AFI(I,J)=AIMAG(FI(I,J))
612 CONTINUE
613 CONTINUE
    DO 171 I=1,N
    ALFR(I)=ALFR(I)-T*T
    XK2(I)=CMPLX(ALFR(I),ALFI(I))
    XKAP(I)=CSQRT(XK2(I))
    WRITE(IWRITE,*) XKAP(I)
    AKR(I)=REAL(XKAP(I))
    AKI(I)=AIMAG(XKAP(I))
171 CONTINUE
    RETURN
    END

C
C -----
C
SUBROUTINE EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI
1 ,IMP,FLAG)
C *****
C DOUBLE PRECISION CONST,CONST1,CONST2,HY
C REAL AFR(ND,ND),AKR(ND),AKI(ND)
C REAL ALFR(20),A(20,20),F(20,20)
C LOGICAL FLAG

DO 10 I=1,ME
AKR(I)=0
AKI(I)=0
10 CONTINUE
DY=SNGL(HY/ME)
T=SNGL(CONST)
T1=SNGL(CONST1)
T2=SNGL(CONST2)
FLAG=.FALSE.
CALL SETR(DY,ME,ND,A,F)
CALL JACOBI(A,F,ALFR,AFR,ME,ND)
DO 30 I=1,ME
TEMP=ALFR(I)-T*T
TX=TEMP
IF(TEMP.LT.0) THEN
FLAG=.TRUE.
AKI(I)=SQRT(-TX)
ELSE IF(TEMP.GT.0) THEN
AKR(I)=SQRT(TX)

```



```

C      BELOW THE FIRST SUBDIAGONAL HAVE BEEN SET TO ZERO, AND THE
C      SUBDIAGONAL ELEMENTS HAVE BEEN MADE REAL (AND NON-NEGATIVE),
C
C      B HAS BEEN REDUCED TO UPPER TRIANGULAR FORM.  THE ELEMENTS
C      BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,
C
C      Z=(ZR,ZI) CONTAINS THE PRODUCT OF THE RIGHT HAND
C      TRANSFORMATIONS IF MATZ HAS BEEN SET TO .TRUE.
C      OTHERWISE, Z IS NOT REFERENCED.
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      ***** INITIALIZE Z *****
C      IF (.NOT. MATZ) GO TO 10
C
C      DO 3 I = 1, N
C
C          DO 2 J = 1, N
C              ZR(I,J) = 0.0
C              ZI(I,J) = 0.0
C      2  CONTINUE
C
C          ZR(I,I) = 1.0
C      3  CONTINUE
C      ***** REDUCE B TO UPPER TRIANGULAR FORM WITH
C      TEMPORARILY REAL DIAGONAL ELEMENTS *****
C      10 IF (N .LE. 1) GO TO 170
C          NM1 = N - 1
C
C          DO 100 L = 1, NM1
C              L1 = L + 1
C              S = 0.0
C
C              DO 20 I = L, N
C                  S = S + ABS(BR(I,L)) + ABS(BI(I,L))
C      20  CONTINUE
C
C              IF (S .EQ. 0.0) GO TO 100
C              RHO = 0.0
C
C              DO 25 I = L, N
C                  BR(I,L) = BR(I,L) / S
C                  BI(I,L) = BI(I,L) / S
C                  RHO = RHO + BR(I,L)**2 + BI(I,L)**2
C      25  CONTINUE
C
C              R = SQRT(RHO)
C              XR = CABS(CMPLX(BR(L,L),BI(L,L)))
C              IF (XR .EQ. 0.0) GO TO 27
C              RHO = RHO + XR * R
C              U1 = -BR(L,L) / XR
C              U1I = -BI(L,L) / XR
C              YR = R / XR + 1.0
C              BR(L,L) = YR * BR(L,L)
C              BI(L,L) = YR * BI(L,L)
C              GO TO 28
C
C

```

```

27   BR(L,L) = R
    U1 = -1.0
    U1I = 0.0
C
28   DO 50 J = L1, N
    T = 0.0
    TI = 0.0
C
    DO 30 I = L, N
    T = T + BR(I,L) * BR(I,J) + BI(I,L) * BI(I,J)
    TI = TI + BR(I,L) * BI(I,J) - BI(I,L) * BR(I,J)
30   CONTINUE
C
    T = T / RHO
    TI = TI / RHO
C
    DO 40 I = L, N
    BR(I,J) = BR(I,J) - T * BR(I,L) + TI * BI(I,L)
    BI(I,J) = BI(I,J) - T * BI(I,L) - TI * BR(I,L)
40   CONTINUE
C
    XI = U1 * BI(L,J) - U1I * BR(L,J)
    BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
    BI(L,J) = XI
50   CONTINUE
C
    DO 80 J = 1, N
    T = 0.0
    TI = 0.0
C
    DO 60 I = L, N
    T = T + BR(I,L) * AR(I,J) + BI(I,L) * AI(I,J)
    TI = TI + BR(I,L) * AI(I,J) - BI(I,L) * AR(I,J)
60   CONTINUE
C
    T = T / RHO
    TI = TI / RHO
C
    DO 70 I = L, N
    AR(I,J) = AR(I,J) - T * BR(I,L) + TI * BI(I,L)
    AI(I,J) = AI(I,J) - T * BI(I,L) - TI * BR(I,L)
70   CONTINUE
C
    XI = U1 * AI(L,J) - U1I * AR(L,J)
    AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
    AI(L,J) = XI
80   CONTINUE
C
    BR(L,L) = R * S
    BI(L,L) = 0.0
C
    DO 90 I = L1, N
    BR(I,L) = 0.0
    BI(I,L) = 0.0
90   CONTINUE
C
100  CONTINUE
C   ***** REDUCE A TO UPPER HESSENBERG FORM WITH REAL SUBDIAGONAL
C   ELEMENTS, WHILE KEEPING B TRIANGULAR *****
C   DO 160 K = 1, NM1

```

```

      K1 = K + 1
C ***** SET BOTTOM ELEMENT IN K-TH COLUMN OF A REAL *****
      IF (AI(N,K) .EQ. 0.0) GO TO 105
      R = CABS(CMPLX(AR(N,K),AI(N,K)))
      U1 = AR(N,K) / R
      U1I = AI(N,K) / R
      AR(N,K) = R
      AI(N,K) = 0.0
C
      DO 103 J = K1, N
        XI = U1 * AI(N,J) - U1I * AR(N,J)
        AR(N,J) = U1 * AR(N,J) + U1I * AI(N,J)
        AI(N,J) = XI
103  CONTINUE
C
      XI = U1 * BI(N,N) - U1I * BR(N,N)
      BR(N,N) = U1 * BR(N,N) + U1I * BI(N,N)
      BI(N,N) = XI
105  IF (K .EQ. NM1) GO TO 170
      NK1 = NM1 - K
C ***** FOR L=N-1 STEP -1 UNTIL K+1 DO -- *****
      DO 150 LB = 1, NK1
        L = N - LB
        L1 = L + 1
C ***** ZERO A(L+1,K) *****
        S = ABS(AR(L,K)) + ABS(AI(L,K)) + AR(L1,K)
        IF (S .EQ. 0.0) GO TO 150
        U1 = AR(L,K) / S
        U1I = AI(L,K) / S
        U2 = AR(L1,K) / S
        R = SQRT(U1*U1+U1I*U1I+U2*U2)
        U1 = U1 / R
        U1I = U1I / R
        U2 = U2 / R
        AR(L,K) = R * S
        AI(L,K) = 0.0
        AR(L1,K) = 0.0
C
      DO 110 J = K1, N
        XR = AR(L,J)
        XI = AI(L,J)
        YR = AR(L1,J)
        YI = AI(L1,J)
        AR(L,J) = U1 * XR + U1I * XI + U2 * YR
        AI(L,J) = U1 * XI - U1I * XR + U2 * YI
        AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110  CONTINUE
C
      XR = BR(L,L)
      BR(L,L) = U1 * XR
      BI(L,L) = -U1I * XR
      BR(L1,L) = -U2 * XR
C
      DO 120 J = L1, N
        XR = BR(L,J)
        XI = BI(L,J)
        YR = BR(L1,J)
        YI = BI(L1,J)
        BR(L,J) = U1 * XR + U1I * XI + U2 * YR

```

```

      BI(L,J) = U1 * XI - U1I * XR + U2 * YI
      BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
      BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
120  CONTINUE
C ***** ZERO B(L+1,L) *****
      S = ABS(BR(L1,L1)) + ABS(BI(L1,L1)) + ABS(BR(L1,L))
      IF (S .EQ. 0.0) GO TO 150
      U1 = BR(L1,L1) / S
      U1I = BI(L1,L1) / S
      U2 = BR(L1,L) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      BR(L1,L1) = R * S
      BI(L1,L1) = 0.0
      BR(L1,L) = 0.0
C
      DO 130 I = 1, L
        XR = BR(I,L1)
        XI = BI(I,L1)
        YR = BR(I,L)
        YI = BI(I,L)
        BR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        BI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        BR(I,L) = U1 * YR - U1I * YI - U2 * XR
        BI(I,L) = U1 * YI + U1I * YR - U2 * XI
130  CONTINUE
C
      DO 140 I = 1, N
        XR = AR(I,L1)
        XI = AI(I,L1)
        YR = AR(I,L)
        YI = AI(I,L)
        AR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        AI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        AR(I,L) = U1 * YR - U1I * YI - U2 * XR
        AI(I,L) = U1 * YI + U1I * YR - U2 * XI
140  CONTINUE
C
      IF (.NOT. MATZ) GO TO 150
C
      DO 145 I = 1, N
        XR = ZR(I,L1)
        XI = ZI(I,L1)
        YR = ZR(I,L)
        YI = ZI(I,L)
        ZR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        ZI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        ZR(I,L) = U1 * YR - U1I * YI - U2 * XR
        ZI(I,L) = U1 * YI + U1I * YR - U2 * XI
145  CONTINUE
C
150  CONTINUE
C
160 CONTINUE
C
170 RETURN
C ***** LAST CARD OF CQZHES *****
      END

```

```

C -----
C
C SUBROUTINE CQZVAL(NM,N,AR,AI,BR,BI,EPS1,ALFR,ALFI,BETA,
X                      MATZ,ZR,ZI,IERR)
C *****
C

```

```

C   INTEGER I,J,K,L,N,EN,K1,K2,LL,L1,NA,NM,ITS,KM1,LM1,
X       ENM2,IERR,LOR1,ENORN
C   REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ALFR(N),ALFI(N),
X       BETA(N),ZR(NM,N),ZI(NM,N)
C   REAL R,S,A1,A2,EP,SH,U1,U2,XI,XR,YI,YR,ANI,A1I,A33,A34,A43,A44,
X       BNI,B1I,B33,B44,SHI,U1I,A33I,A34I,A43I,A44I,B33I,B44I,
X       EPSA,EPSB,EPS1,ANORM,BNORM,B3344,B3344I
C   REAL SQRT,CABS,ABS
C   INTEGER MAX0
C   LOGICAL MATZ
C   COMPLEX Z3
C   COMPLEX CSQRT,CMLPX
C   REAL REAL,AIMAG

```

```

C
C   THIS SUBROUTINE IS A COMPLEX ANALOGUE OF STEPS 2 AND 3 OF THE
C   QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
C   SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART,
C   AS MODIFIED IN TECHNICAL NOTE NASA TN E-7305(1973) BY WARD.
C

```

```

C   THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES, ONE OF THEM
C   IN UPPER HESSENBERG FORM AND THE OTHER IN UPPER TRIANGULAR FORM,
C   THE HESSENBERG MATRIX MUST FURTHER HAVE REAL SUBDIAGONAL ELEMENTS.
C   IT REDUCES THE HESSENBERG MATRIX TO TRIANGULAR FORM USING
C   UNITARY TRANSFORMATIONS WHILE MAINTAINING THE TRIANGULAR FORM
C   OF THE OTHER MATRIX AND FURTHER MAKING ITS DIAGONAL ELEMENTS
C   REAL AND NON-NEGATIVE. IT THEN RETURNS QUANTITIES WHOSE RATIOS
C   GIVE THE GENERALIZED EIGENVALUES. IT IS USUALLY PRECEDED BY
C   CQZHS AND POSSIBLY FOLLOWED BY CQZVEC.
C

```

```

C   ON INPUT-
C

```

```

C       NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C       ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C       DIMENSION STATEMENT,
C

```

```

C       N IS THE ORDER OF THE MATRICES,
C

```

```

C       A=(AR,AI) CONTAINS A COMPLEX UPPER HESSENBERG MATRIX
C       WITH REAL SUBDIAGONAL ELEMENTS,
C

```

```

C       B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,
C

```

```

C       EPS1 IS A TOLERANCE USED TO DETERMINE NEGLIGIBLE ELEMENTS.
C       EPS1 = 0.0 (OR NEGATIVE) MAY BE INPUT, IN WHICH CASE AN
C       ELEMENT WILL BE NEGLECTED ONLY IF IT IS LESS THAN ROUNDOFF
C       ERROR TIMES THE NORM OF ITS MATRIX. IF THE INPUT EPS1 IS
C       POSITIVE, THEN AN ELEMENT WILL BE CONSIDERED NEGLIGIBLE
C       IF IT IS LESS THAN EPS1 TIMES THE NORM OF ITS MATRIX. A
C       POSITIVE VALUE OF EPS1 MAY RESULT IN FASTER EXECUTION,
C

```

BUT LESS ACCURATE RESULTS,

MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS
ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING
EIGENVECTORS, AND TO .FALSE. OTHERWISE,

Z=(ZR,ZI) CONTAINS, IF MATZ HAS BEEN SET TO .TRUE., THE
TRANSFORMATION MATRIX PRODUCED IN THE REDUCTION
BY CQZHE, IF PERFORMED, OR ELSE THE IDENTITY MATRIX.
IF MATZ HAS BEEN SET TO .FALSE., Z IS NOT REFERENCED.

ON OUTPUT

A HAS BEEN REDUCED TO UPPER TRIANGULAR FORM. THE ELEMENTS
BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,

B IS STILL IN UPPER TRIANGULAR FORM, ALTHOUGH ITS ELEMENTS
HAVE BEEN ALTERED. IN PARTICULAR, ITS DIAGONAL HAS BEEN SET
REAL AND NON-NEGATIVE. THE LOCATION BR(N,1) IS USED TO
STORE EPS1 TIMES THE NORM OF B FOR LATER USE BY CQZVEC,

ALFR AND ALFI CONTAIN THE REAL AND IMAGINARY PARTS OF THE
DIAGONAL ELEMENTS OF THE TRIANGULARIZED A MATRIX,

BETA CONTAINS THE REAL NON-NEGATIVE DIAGONAL ELEMENTS OF THE
CORRESPONDING B. THE GENERALIZED EIGENVALUES ARE THEN
THE RATIOS ((ALFR+I*ALFI)/BETA),

Z CONTAINS THE PRODUCT OF THE RIGHT HAND TRANSFORMATIONS
(FOR BOTH STEPS) IF MATZ HAS BEEN SET TO .TRUE.,

IERR IS SET TO
ZERO FOR NORMAL RETURN,
J IF AR(J,J-1) HAS NOT BECOME
ZERO AFTER 50 ITERATIONS.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARROW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

IERR = 0
***** COMPUTE EPSA, EPSB *****
ANORM = 0.0
BNORM = 0.0

DO 30 I = 1, N
ANI = 0.0
IF (I .NE. 1) ANI = ABS(AR(I,I-1))
BNI = 0.0

DO 20 J = I, N
ANI = ANI + ABS(AR(I,J)) + ABS(AI(I,J))
BNI = BNI + ABS(BR(I,J)) + ABS(BI(I,J))

20 CONTINUE

IF (ANI .GT. ANORM) ANORM = ANI
IF (BNI .GT. BNORM) BNORM = BNI

30 CONTINUE

```

      IF (ANORM .EQ. 0.0) ANORM = 1.0
      IF (BNORM .EQ. 0.0) BNORM = 1.0
      EP = E131
      IF (EP .GT. 0.0) GO TO 50
C     ***** COMPUTE ROUNDOFF LEVEL IF EPS1 IS ZERO *****
      EP = 1.0
40    EP = EP / 2.0
      IF (1.0 + EP .GT. 1.0) GO TO 40
50    EPSA = EP * ANORM
      EPSB = EP * BNORM
C     ***** REDUCE A TO TRIANGULAR FORM, WHILE
C     KEEPING B TRIANGULAR *****
      LOR1 = 1
      ENORN = N
      EN = N
C     ***** BEGIN QZ STEP *****
60    IF (EN .EQ. 0) GO TO 1001
      IF (.NOT. MATZ) ENORN = EN
      ITS = 0
      NA = EN - 1
      ENM2 = NA - 1
C     ***** CHECK FOR CONVERGENCE OR REDUCIBILITY.
C     FOR L=EN STEP -1 UNTIL 1 DO -- *****
70    DO 80 LL = 1, EN
        LM1 = EN - LL
        L = LM1 + 1
        IF (L .EQ. 1) GO TO 95
        IF (ABS(AR(L,LM1)) .LE. EPSA) GO TO 90
80    CONTINUE
C
90    AR(L,LM1) = 0.0
C     ***** SET DIAGONAL ELEMENT AT TOP OF B REAL *****
95    B11 = CABS(CMPLX(BR(L,L),BI(L,L)))
      IF (B11 .EQ. 0.0) GO TO 98
      U1 = BR(L,L) / B11
      U1I = BI(L,L) / B11
C
      DO 97 J = L, ENORN
        XI = U1 * AI(L,J) - U1I * AR(L,J)
        AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
        AI(L,J) = XI
        XI = U1 * BI(L,J) - U1I * BR(L,J)
        BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
        BI(L,J) = XI
97    CONTINUE
C
      BI(L,L) = 0.0
98    IF (L .NE. EN) GO TO 100
C     ***** 1-BY-1 BLOCK ISOLATED *****
      ALFR(EN) = AR(EN,EN)
      ALFI(EN) = AI(EN,EN)
      BETA(EN) = B11
      EN = NA
      GO TO 60
C     ***** CHECK FOR SMALL TOP OF B *****
100   L1 = L + 1
      IF (B11 .GT. EPSB) GO TO 120
      BR(L,L) = 0.0
      S = ABS(AR(L,L)) + ABS(AI(L,L)) + ABS(AR(L1,L))
      U1 = AR(L,L) / S

```



```

      U1I = AI(L,L) / S
      U2 = AR(L1,L) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      AR(L,L) = R * S
      AI(L,L) = 0.0
C
      DO 110 J = L1, ENORN
        XR = AR(L,J)
        XI = AI(L,J)
        YR = AR(L1,J)
        YI = AI(L1,J)
        AR(L,J) = U1 * XR + U1I * XI + U2 * YR
        AI(L,J) = U1 * XI - U1I * XR + U2 * YI
        AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
        XR = BR(L,J)
        XI = BI(L,J)
        YR = BR(L1,J)
        YI = BI(L1,J)
        BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        BR(L,J) = U1 * XR + U1I * XI + U2 * YR
        BI(L,J) = U1 * XI - U1I * XR + U2 * YI
        BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110 CONTINUE
C
      LM1 = L
      L = L1
      GO TO 90
C
      ***** ITERATION STRATEGY *****
120 IF (ITS .EQ. 50) GO TO 1000
      IF (ITS .EQ. 10) GO TO 135
C
      ***** DETERMINE SHIFT *****
      B33 = BR(NA,NA)
      B33I = BI(NA,NA)
      IF (CABS(CMPLX(B33,B33I)) .GE. EPSB) GO TO 122
      B33 = EPSB
      B33I = 0.0
122 B44 = BR(EN,EN)
      B44I = BI(EN,EN)
      IF (CABS(CMPLX(B44,B44I)) .GE. EPSB) GO TO 124
      B44 = EPSB
      B44I = 0.0
124 B3344 = B33 * B44 - B33I * B44I
      B3344I = B33 * B44I + B33I * B44
      A33 = AR(NA,NA) * B44 - AI(NA,NA) * B44I
      A33I = AR(NA,NA) * B44I + AI(NA,NA) * B44
      A34 = AR(NA,EN) * B33 - AI(NA,EN) * B33I
      X - AR(NA,NA) * BR(NA,EN) + AI(NA,NA) * BI(NA,EN)
      A34I = AR(NA,EN) * B33I + AI(NA,EN) * B33
      X - AR(NA,NA) * BI(NA,EN) - AI(NA,NA) * BR(NA,EN)
      A43 = AR(EN,NA) * B44
      A43I = AR(EN,NA) * B44I
      A44 = AR(EN,EN) * B33 - AI(EN,EN) * B33I - AR(EN,NA) * BR(NA,EN)
      A44I = AR(EN,EN) * B33I + AI(EN,EN) * B33 - AR(EN,NA) * BI(NA,EN)
      SH = A44
      SHI = A44I
      XR = A34 * A43 - A34I * A43I

```

```

      XI = A34 * A43I + A34I * A43
      IF (XR .EQ. 0.0 .AND. XI .EQ. 0.0) GO TO 140
      YR = (A33 - SH) / 2.0
      YI = (A33I - SHI) / 2.0
      Z3 = CSQRT(CMPLX(YR**2-YI**2+XR,2.0*YR*YI+XI))
      U1 = REAL(Z3)
      U1I = AIMAG(Z3)
      IF (YR * U1 + YI * U1I .GE. 0.0) GO TO 125
      U1 = -U1
      U1I = -U1I
125  Z3 = (CMPLX(SH,SHI) - CMPLX(XR,XI) / CMPLX(YR+U1,YI+U1I))
      X  / CMPLX(B3344,B3344I)
      SH = REAL(Z3)
      SHI = AIMAG(Z3)
      GO TO 140
C      ***** AD HOC SHIFT *****
135  SH = AR(EN,NA) + AR(NA,ENM2)
      SHI = 0.0
C      ***** DETERMINE ZEROth COLUMN OF A *****
140  A1 = AR(L,L) / B11 - SH
      A1I = AI(L,L) / B11 - SHI
      A2 = AR(L1,L) / B11
      ITS = ITS + 1
      IF (.NOT. MATZ) LOR1 = L
C      ***** MAIN LOOP *****
      DO 260 K = L, NA
        K1 = K + 1
        K2 = K + 2
        KM1 = MAX0(K-1,L)
C      ***** ZERO A(K+1,K-1) *****
        IF (K .EQ. L) GO TO 170
        A1 = AR(K,KM1)
        A1I = AI(K,KM1)
        A2 = AR(K1,KM1)
170  S = ABS(A1) + ABS(A1I) + ABS(A2)
        U1 = A1 / S
        U1I = A1I / S
        U2 = A2 / S
        R = SQRT(U1*U1+U1I*U1I+U2*U2)
        U1 = U1 / R
        U1I = U1I / R
        U2 = U2 / R
C
      DO 180 J = KM1, ENORN
        XR = AR(K,J)
        XI = AI(K,J)
        YR = AR(K1,J)
        YI = AI(K1,J)
        AR(K,J) = U1 * XR + U1I * XI + U2 * YR
        AI(K,J) = U1 * XI - U1I * XR + U2 * YI
        AR(K1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(K1,J) = U1 * YI + U1I * YR - U2 * XI
        XR = BR(K,J)
        XI = BI(K,J)
        YR = BR(K1,J)
        YI = BI(K1,J)
        BR(K,J) = U1 * XR + U1I * XI + U2 * YR
        BI(K,J) = U1 * XI - U1I * XR + U2 * YI
        BR(K1,J) = U1 * YR - U1I * YI - U2 * XR
        BI(K1,J) = U1 * YI + U1I * YR - U2 * XI

```

```

180    CONTINUE
C
    IF (K .EQ. L) GO TO 240
    AI(K,KM1) = 0.0
    AR(K1,KM1) = 0.0
    AI(K1,KM1) = 0.0
C ***** ZERO B(K+1,K) *****
240    S = ABS(BR(K1,K1)) + ABS(BI(K1,K1)) + ABS(BR(K1,K))
    U1 = BR(K1,K1) / S
    U1I = BI(K1,K1) / S
    U2 = BR(K1,K) / S
    R = SQRT(U1*U1+U1I*U1I+U2*U2)
    U1 = U1 / R
    U1I = U1I / R
    U2 = U2 / R
    IF (K .EQ. NA) GO TO 245
    XR = AR(K2,K1)
    AR(K2,K1) = U1 * XR
    AI(K2,K1) = -U1I * XR
    AR(K2,K) = -U2 * XR
C
245    DO 250 I = LOR1, K1
        XR = AR(I,K1)
        XI = AI(I,K1)
        YR = AR(I,K)
        YI = AI(I,K)
        AR(I,K1) = U1 * XR + U1I * XI + U2 * YR
        AI(I,K1) = U1 * XI - U1I * XR + U2 * YI
        AR(I,K) = U1 * YR - U1I * YI - U2 * XR
        AI(I,K) = U1 * YI + U1I * YR - U2 * XI
        XR = BR(I,K1)
        XI = BI(I,K1)
        YR = BR(I,K)
        YI = BI(I,K)
        BR(I,K1) = U1 * XR + U1I * XI + U2 * YR
        BI(I,K1) = U1 * XI - U1I * XR + U2 * YI
        BR(I,K) = U1 * YR - U1I * YI - U2 * XR
        BI(I,K) = U1 * YI + U1I * YR - U2 * XI
250    CONTINUE
C
    BI(K1,K1) = 0.0
    BR(K1,K) = 0.0
    BI(K1,K) = 0.0
    IF (.NOT. MATZ) GO TO 260
C
    DO 255 I = 1, N
        XR = ZR(I,K1)
        XI = ZI(I,K1)
        YR = ZR(I,K)
        YI = ZI(I,K)
        ZR(I,K1) = U1 * XR + U1I * XI + U2 * YR
        ZI(I,K1) = U1 * XI - U1I * XR + U2 * YI
        ZR(I,K) = U1 * YR - U1I * YI - U2 * XR
        ZI(I,K) = U1 * YI + U1I * YR - U2 * XI
255    CONTINUE
C
260    CONTINUE
C ***** SET LAST A SUBDIAGONAL REAL AND END QZ STEP *****
    IF (AI(EN,NA) .EQ. 0.0) GO TO 70
    R = CABS(CMPLX(AR(EN,NA),AI(EN,NA)))

```

```

U1 = AR(EN,NA) / R
U1I = AI(EN,NA) / R
AR(EN,NA) = R
AI(EN,NA) = 0.0
C
DO 270 J = EN, ENORN
    XI = U1 * AI(EN,J) - U1I * AR(EN,J)
    AR(EN,J) = U1 * AR(EN,J) + U1I * AI(EN,J)
    AI(EN,J) = XI
    XI = U1 * BI(EN,J) - U1I * BR(EN,J)
    BR(EN,J) = U1 * BR(EN,J) + U1I * BI(EN,J)
    BI(EN,J) = XI
270 CONTINUE
C
GO TO 70
C
***** SET ERROR -- BOTTOM SUBDIAGONAL ELEMENT HAS NOT
***** BECOME NEGLIGIBLE AFTER 50 ITERATIONS *****
1000 IERR = EN
C
***** SAVE EPSB FOR USE BY CQZVEC *****
1001 IF (N .GT. 1) BR(N,1) = EPSB
RETURN
C
***** LAST CARD OF CQZVAL *****
END
C
C
-----
SUBROUTINE CQZVEC(NM,N,AR,AI,BR,BI,ALFR,ALFI,BETA,ZR,ZI)
*****
INTEGER I,J,K,M,N,EN,II,JJ,NA,NM,NN
REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ALFR(N),ALFI(N),
X BETA(N),ZR(NM,N),ZI(NM,N)
REAL R,T,RI,TI,XI,ALMI,ALMR,BETM,EPSB
REAL CABS
COMPLEX Z3
COMPLEX CMPLX
REAL REAL,AIMAG
C
C
C
C
THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FOURTH STEP OF THE
QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.
C
THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES IN UPPER
TRIANGULAR FORM, WHERE ONE OF THEM FURTHER MUST HAVE REAL DIAGONAL
ELEMENTS. IT COMPUTES THE EIGENVECTORS OF THE TRIANGULAR PROBLEM
AND TRANSFORMS THE RESULTS BACK TO THE ORIGINAL COORDINATE SYSTEM.
IT IS USUALLY PRECEDED BY CQZHES AND CQZVAL.
C
ON INPUT-
C
NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,
C
N IS THE ORDER OF THE MATRICES,
```

```

C      A=(AR,AI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,
C
C      B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX WITH REAL
C      DIAGONAL ELEMENTS.  IN ADDITION, LOCATION BR(N,1) CONTAINS
C      THE TOLERANCE QUANTITY (EPSB) COMPUTED AND SAVED IN CQZVAL,
C
C      ALFR, ALFI, AND BETA ARE VECTORS WITH COMPONENTS WHOSE
C      RATIOS ((ALFR+I*ALFI)/BETA) ARE THE GENERALIZED
C      EIGENVALUES.  THEY ARE USUALLY OBTAINED FROM CQZVAL,
C
C      Z=(ZR,ZI) CONTAINS THE TRANSFORMATION MATRIX PRODUCED IN THE
C      REDUCTIONS BY CQZHES AND CQZVAL, IF PERFORMED.
C      IF THE EIGENVECTORS OF THE TRIANGULAR PROBLEM ARE
C      DESIRED, Z MUST CONTAIN THE IDENTITY MATRIX.
C
C      ON OUTPUT-
C
C      A IS UNALTERED,
C
C      B HAS BEEN DESTROYED,
C
C      ALFR, ALFI, AND BETA ARE UNALTERED,
C
C      Z CONTAINS THE EIGENVECTORS.  EACH EIGENVECTOR IS NORMALIZED
C      SO THAT THE MODULUS OF ITS LARGEST COMPONENT IS 1.0 .
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      IF (N .LE. 1) GO TO 1001
C      EPSB = BR(N,1)
C      ***** FOR EN=N STEP -1 UNTIL 2 DO -- *****
C      DO 800 NN = 2, N
C        EN = N + 2 - NN
C        NA = EN - 1
C        ALMR = ALFR(EN)
C        ALMI = ALFI(EN)
C        BETM = BETA(EN)
C      ***** FOR I=EN-1 STEP -1 UNTIL 1 DO -- *****
C        DO 700 II = 1, NA
C          I = EN - II
C          R = 0.0
C          RI = 0.0
C          M = I + 1
C
C          DO 610 J = M, EN
C            T = BETM * AR(I,J) - ALMR * BR(I,J) + ALMI * BI(I,J)
C            TI = BETM * AI(I,J) - ALMR * BI(I,J) - ALMI * BR(I,J)
C            IF (J .EQ. EN) GO TO 605
C            XI = T * BI(J,EN) + TI * BR(J,EN)
C            T = T * BR(J,EN) - TI * BI(J,EN)
C            TI = XI
C          605      R = R + T
C                  RI = RI + TI
C          610      CONTINUE
C
C      T = ALMR * BETA(I) - BETM * ALFR(I)
C      TI = ALMI * BETA(I) - BETM * ALFI(I)

```

```

                IF (T .EQ. 0.0 .AND. TI .EQ. 0.0) T = EPSB
                Z3 = CMPLX(R,RI) / CMPLX(T,TI)
                BR(I,EN) = REAL(Z3)
                BI(I,EN) = AIMAG(Z3)
700      CONTINUE
C
800      CONTINUE
C      ***** END BACK SUBSTITUTION.
C      TRANSFORM TO ORIGINAL COORDINATE SYSTEM.
C      FOR J=N STEP -1 UNTIL 2 DO -- *****
      DO 880 JJ = 2, N
        J = N + 2 - JJ
        M = J - 1
C
        DO 880 I = 1, N
C
          DO 860 K = 1, M
            ZR(I,J) = ZR(I,J) + ZR(I,K) * BR(K,J) - ZI(I,K) * BI(K,J)
            ZI(I,J) = ZI(I,J) + ZR(I,K) * BI(K,J) + ZI(I,K) * BR(K,J)
860      CONTINUE
C
880      CONTINUE
C      ***** NORMALIZE SO THAT MODULUS OF LARGEST
C      COMPONENT OF EACH VECTOR IS 1 *****
      DO 950 J = 1, N
        T = 0.0
C
        DO 930 I = 1, N
          R = CABS(CMPLX(ZR(I,J),ZI(I,J)))
          IF (R .GT. T) T = R
930      CONTINUE
C
        DO 940 I = 1, N
          ZR(I,J) = ZR(I,J) / T
          ZI(I,J) = ZI(I,J) / T
940      CONTINUE
C
950      CONTINUE
C
1001     RETURN
C      ***** LAST CARD OF CQZVEC *****
      END
C
C
C
C
SUBROUTINE NORMA(FI,G,N,NM,XM)
C      *****
C
      COMPLEX FI(NM,NM),XM(NM,NM),TEMP(20),SUM1,SUM2
      COMPLEX XI(20,20),X(20,20),X1,X2
      REAL G(NM,NM)
      DATA IWRITE/6/
C
      DO 30 I=1,N
        DO 40 J=1,N
          SUM1=(0.0,0.0)
          DO 50 K=1,N
            SUM1=SUM1+CMPLX(G(J,K))*FI(K,I)
50      CONTINUE

```

```

      TEMP(J)=SUM1
40  CONTINUE
      SUM2=(0.0,0.0)
      DO 41 J=1,N
      SUM2=SUM2+FI(J,I)*TEMP(J)
41  CONTINUE
      SUM2=CSQRT(SUM2)
      DO 51 J=1,N
      FI(J,I)=FI(J,I)/SUM2
51  CONTINUE
30  CONTINUE
C
C  FOR CHECK CARRY OUT [FIT]*[G]*[FI]=[I]
C
      DO 100 I=1,N
      DO 110 J=1,N
      X1=(0.0,0.0)
      DO 120 K=1,N
      X1=X1+FI(K,I)*CMPLX(G(K,J))
120  CONTINUE
      X(I,J)=X1
110  CONTINUE
100  CONTINUE
      DO 130 I=1,N
      DO 140 J=1,N
      X2=(0.0,0.0)
      DO 150 K=1,N
      X2=X2+X(I,K)*FI(K,J)
150  CONTINUE
      XI(I,J)=X2
140  CONTINUE
130  CONTINUE
      DO 161 I=1,N
      WRITE(IWRITE,*) (XI(I,J):J=1,N)
161  CONTINUE
      RETURN
      END
C
C  -----
C
      SUBROUTINE SET(DY,T1,T2,N,NM,H,BO,G,BI)
      *****
C  SET CALCULATES MATRIX [A]p=[H]+i[BO]
C  AND MATRIX [B]p=[G]
      REAL H(NM,NM),BO(NM,NM),G(NM,NM)
      REAL BI(NM,NM)
      DO 10 I=1,N
      I1=I-1
      I2=I+1
      DO 20 J=1,N
      IF(I.EQ.1.AND.J.EQ.1) THEN
      H(I,J)=1.0
      G(I,J)=1.0
      ELSE IF (I.NE.1.AND.I.EQ.J) THEN
      H(I,J)=2.0
      G(I,J)=2.0
      ELSE IF (J.EQ.I1.OR.J.EQ.I2) THEN
      H(I,J)=-1.0
      G(I,J)=0.5

```

```

ELSE
  H(I,J)=0.0
  G(I,J)=0.0
  END IF
  H(I,J)=H(I,J)/DY
  G(I,J)=G(I,J)*(DY/3)
20 CONTINUE
10 CONTINUE
  DO 30 I=1,N
  DO 40 J=1,N
    IF(I.EQ.1.AND.J.EQ.1) THEN
      BO(I,J)=T2*T1
    ELSE
      BO(I,J)=0.0
    END IF
  40 CONTINUE
  30 CONTINUE
  DO 80 I=1,N
  DO 90 J=1,N
    BI(I,J)=0.0
  90 CONTINUE
  80 CONTINUE
  RETURN
  END

C
C -----
C
C SUBROUTINE SETR(DY,ME,ND,A,F)
C *****
C
  REAL A(ND,ND),F(ND,ND)
  DO 9 I=1,ME
  DO 8 J=1,ME
    A(I,J)=0.
    F(I,J)=0.
  8 CONTINUE
  9 CONTINUE
    A(1,1)=1/DY
    A(1,2)=-1./DY
    F(1,1)=DY/3
    F(1,2)=.5*DY/3
    DO 10 I=2,ME-1
      A(I,I)=2/DY
      F(I,I)=2*DY/3
      A(I,I-1)=-1/DY
      A(I,I+1)=-1/DY
      F(I,I-1)=.5*DY/3
      F(I,I+1)=.5*DY/3
  10 CONTINUE
    A(ME,ME)=2/DY
    A(ME,ME-1)=-1/DY
    F(ME,ME)=2*DY/3
    F(ME,ME-1)=.5*DY/3
  RETURN
  END

C
C -----
C
C SUBROUTINE FMAT(X,Y,XM,YM,G,H,HOLD,FID,FIB,DFI,N0,N1)
C *****

```



```

C
C      FMAT COMPUTES THE G AND H MATRICES AND FORMS
C      THE SYSTEMS OF EQUATIONS AX=F
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION X(1),Y(1),G(NU,NU),H(NU,NU),
1 FID(1),FIB(1),LFI(1),HOLD(NU,NU),XM(1),YM(1)
C
      NX=2*N
      DO 10 I=1,NX
      DFI(I)=0
      DO 11 J=1,NX
      G(I,J)=0.0
      H(I,J)=0.0
11 CONTINUE
10 CONTINUE
C
      X(N+1)=X(1)
      Y(N+1)=Y(1)
C
      DO 12 I=1,N
      XM(I)=(X(I)+X(I+1))/2
      YM(I)=(Y(I)+Y(I+1))/2
12 CONTINUE
C
      DO 20 I=1,N
      DO 30 J=1,N
      IF(I-J) 40,45,40
40 CALL INTE(XM(I),YM(I),X(J),Y(J),X(J+1),Y(J+1),A,B)
      H(I,J)=A
      G(I,J)=B
      GO TO 30
45 CALL INLO(X(J),Y(J),X(J+1),Y(J+1),B)
      G(I,J)=B
      H(I,J)=-0.5
30 CONTINUE
20 CONTINUE
C
C ARRANGE THE SYSTEM OF EQUATIONS
C
      DO 150 J=1,N0
      DO 145 I=1,N
      CH=G(I,J)
      G(I,J)=-H(I,J)
      H(I,J)=-CH
145 CONTINUE
150 CONTINUE
C
      DO 160 I=1,N
      DO 170 J=1,N
      HOLD(I,J)=H(I,J)
170 CONTINUE
160 CONTINUE
C
      RETURN
      END
C
C -----

```

```

C
C      SUBROUTINE INTE (XP,YP,X1,Y1,X2,Y2,H,G)
C      *****
C
C      THIS SUBROUTINE COMPUTES THE VALUES OF THE H AND GMATRIX
C      OFF DIAGONAL ELEMENTS BY MEANS OF NUMERICAL INTEGRATION
C      ALONG AN ELEMENT
C      USING 4-POINT GAUSS QUADRATURE
C      RA=DISTANCE FROM THE POINT UNDER CONSIDERATION
C      TO THE INTEGRATION POINT ON THE BOUNDARY ELEMENT
C
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C      DOUBLE PRECISION XCO(4),YCO(4),GI(4),OME(4),
C      &LEN,K
C      K=CONST
C      GI(1)=0.86113631
C      GI(2)=-GI(1)
C      GI(3)=0.33998104
C      GI(4)=-GI(3)
C      OME(1)=0.34785485
C      OME(2)=OME(1)
C      OME(3)=0.65214515
C      OME(4)=OME(3)
C      AX=(X2-X1)/2
C      BX=(X2+X1)/2
C      AY=(Y2-Y1)/2
C      BY=(Y2+Y1)/2
C      IF (AX.EQ.0) THEN
C          DIST=DABS (XP-X1)
C      ELSE
C          TA=AY/AX
C          DIST=DABS ( (TA*XP-YP+Y1-TA*X1) /DSQRT (TA**2+1) )
C      END IF
C      SIG=(X1-XP) * (Y2-YP) - (X2-XP) * (Y1-YP)
C      IF (SIG.LT.0) DIST=-DIST
C      H=C
C      G=0
C      LEN=DSQRT (AX*AX+AY*AY)
C      DO 40 I=1,4
C          XCO(I)=AX*GI(I)+BX
C          YCO(I)=AY*GI(I)+BY
C          RA=DSQRT ( (XP-XCO(I)) **2+(YP-YCO(I)) **2)
C          CON=K*RA
C          CALL BFUNCT (CON,Y0X,Y1X)
C          G=G+Y0X*OME(I) *LEN*0.25
C 40 H=H+Y1X*OME(I) *DIST*K*(0.25/RA) *LEN
C      RETURN
C      END
C
C      -----
C
C      SUBROUTINE INLO (X1,Y1,X2,Y2,G)
C      *****
C
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C

```

```

C
C   THIS SUBROUTINE COMPUTES THE VALUES OF THE DIAGONAL
C   ELEMENTS OF THE G MATRIX
C

```

```

AX=(X2-X1)/2
AY=(Y2-Y1)/2
SR=DSQRT(AX**2+AY**2)
XXX=SR*CONST
G=(SR/3.1415927)*(DLOG(XXX)-1.0)
RETURN
END

```

```

C
C
C

```

```

SUBROUTINE BFUNCT(XX,Y0X,Y1X)

```

```

*****

```

```

C
C   IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
REAL*8 XULT0,XOSH0,XOSH1
INTEGER XF,Z0,Z
C   BESSEL FUNCTION CALCULATIONS
IF(XX-15.9)1000,1000,2000
1000 A=0.6366197724
B=0.5772156646
XULT0=1.0
SIGMA=0.0
TY0=0.0
TY1=0.0
DO 7 I=1,500
XF=I
Z=-1.0
Z0=XF+1.0
SINY0=Z**(XF)
SINY1=Z**(Z0)
XOSH0=(XX/2.0)**(2*XF)
XOSH1=(XX/2.0)**(2*XF-1.0)
XULT0=XULT0*XF
XULT1=XULT0/XF
SIGMA=SIGMA+(1.0/XF)
VALU0=DLOG(XX/2.0)+B-SIGMA
VALU1=VALU0+(1.0/(2.0*XF))
TRMY0=((SINY0*XOSH0)/(XULT0**2))*(VALU0)
TRMY1=((SINY1*XOSH1)/(XULT0*XULT1))*(VALU1)
TY0=TY0+TRMY0
TY1=TY1+TRMY1
ERY0=DABS(TRMY0/TY0)
ERY1=DABS(TRMY1/TY1)
IF(ERY0-ERY1)1,1,2
1 ER=ERY1
GO TO 10
2 ER=ERY0
10 IF(ER-.001)11,11,7
7 CONTINUE
11 Y0X=A*(DLOG(XX/2.0)+B+TY0)
Y1X=A*((1.0/XX)-TY1)
GO TO 100
2000 AH=3.1415926536/4.0
BH=XX-AH

```

```

CH=XX+AH
S1H=SIN(BH)
S2H=SIN(CH)
DHY0=S2H/(8.0*XX)
DHY1=S1H*3.0/(8.0*XX)
HH=XX-(3.0*AH)
S3H=SIN(HH)
EH=2.0/(3.1415926536*XX)
FH=DSQRT(EH)
XMY0=S1H-DHY0
XMY1=S3H+DHY1
Y0X=XMY0*FH
Y1X=-XMY1*FH
100 RETURN
END

C
C -----
C
SUBROUTINE ADDDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,
1 DFI,FID,FIB,DF)
C *****
C
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
DOUBLE PRECISION H(NU,NU),G(NU,NU),T(100,20)
DOUBLE PRECISION FID(NU),FIB(NU),DFI(NU),HOLD(NU,NU)
REAL UL(20,20),VB(20,20),VL(20,20),UB(20,20)
COMPLEX XKAP(20),PSI(20,20),TEMP1(20,20),TEMP2(20,20)
COMPLEX SUM3,DX(20)
REAL DXR(20),DXI(20)
REAL ULT(20,20),VLT(20,20),UBT(20,20),VBT(20,20)
REAL AFRT(20,20),AFIT(20,20),TP(20,20)

C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C
IF (ME.GT.0) THEN
DO 20 J=1,ME
DO 10 I=1,ME
UL(I,J)=AFR(I,J)*AKR(J)
VB(I,J)=AFI(I,J)*AKI(J)
VL(I,J)=AFI(I,J)*AKR(J)
UB(I,J)=AFR(I,J)*AKI(J)
10 CONTINUE
20 CONTINUE

C
C CREATE MATRIX TP WHICH TRANSFORMS VECTORS OF PRESSURES AND THEIR
C DERIVATIVES AT FAR END FROM LINEAR FORMULATION INTO CONSTANT ONE
C ( FROM NODAL POINTS INTO MIDPOINTS )
C
CALL TRANSP(TP)

C
C
DO 21 I=1,ME
DO 22 J=1,ME

```

```

    TEM1=0.0
    TEM2=0.0
    TEM3=0.0
    TEM4=0.0
    DO 23 K=1,ME
    TEM1=TEM1+TP(I,K)*UL(K,J)
    TEM2=TEM2+TP(I,K)*VB(K,J)
    TEM3=TEM3+TP(I,K)*VL(K,J)
    TEM4=TEM4+TP(I,K)*UB(K,J)
23  CONTINUE
    ULT(I,J)=TEM1
    VBT(I,J)=TEM2
    VLT(I,J)=TEM3
    UBT(I,J)=TEM4
22  CONTINUE
21  CONTINUE
C
    DO 41 I=1,ME
    DO 31 J=1,ME
    ULT(I,J)=ULT(I,J)-VBT(I,J)
    VLT(I,J)=VLT(I,J)+UBT(I,J)
31  CONTINUE
41  CONTINUE
C
    DO 61 I=1,ME
    DO 62 J=1,ME
    TEM1=0.0
    TEM2=0.0
    DO 63 K=1,ME
    TEM1=TEM1+TP(I,K)*AFR(K,J)
    TEM2=TEM2+TP(I,K)*AFI(K,J)
63  CONTINUE
    AFRT(I,J)=TEM1
    AFIT(I,J)=TEM2
62  CONTINUE
61  CONTINUE
C
    DO 70 I=1,N
    DO 60 J=1,ME
    TEM1=0.0
    TEM2=0.0
    DO 50 K=1,ME
    KK=K+N2
    TEM1=TEM1+H(I,KK)*DBLE(AFRT(K,J))
    TEM2=TEM2+G(I,KK)*DBLE(ULT(K,J))
50  CONTINUE
    T(I,J)=TEM1+TEM2
60  CONTINUE
70  CONTINUE
C
    DO 75 I=1,N
    DO 72 J=1,ME
    JJ=J+N2
    H(I,JJ)=T(I,J)
72  CONTINUE
75  CONTINUE
C
C  IMAGINARY PART
C
C

```

```

DO 120 I=1,N
DO 110 J=1,ME
TEM1=0.0
TEM2=0.0
DO 100 K=1,ME
KK=K+N2
TEM1=TEM1+G(I, KK)*DBLE(VLT(K, J))
TEM2=TEM2+H(I, KK)*DBLE(AFIT(K, J))
100 CONTINUE
T(I, J)=TEM1+TEM2
110 CONTINUE
120 CONTINUE
C
DO 140 I=1,N
DO 130 J=1,ME
H(N+I, J+N2)=T(I, J)
130 CONTINUE
140 CONTINUE
END IF
C
IF(CONST2.GT.1.E-08) THEN
TEMP=CONST1*CONST2
DO 90 I=1,N
DO 80 J=N1+1,N2
H(N+I, J)=G(I, J)*TEMP
80 CONTINUE
90 CONTINUE
END IF
C
C FILL IN THE REST OF H MATRIX
C
C
DO 160 I=1,N
DO 150 J=1,N
H(I, J+N)=--H(I+N, J)
H(I+N, J+N)=H(I, J)
150 CONTINUE
160 CONTINUE
C
C FORM RIGHT HAND VECTOR
C
DO 180 I=1,N
DO 170 J=N0+1,N1
DFI(I)=DFI(I)+G(I, J)*FID(J-N0)
170 CONTINUE
180 CONTINUE
C
DO 200 I=1,N
DO 190 J=N1+1,N2
DFI(I)=DFI(I)+G(I, J)*FIB(J-N1)
190 CONTINUE
200 CONTINUE
C
IF(DF.GT.0.0) THEN
DO 211 I=1,ME
XKAP(I)=CMPLX(AKR(I), AKI(I))
DO 212 J=1,ME
PSI(I, J)=CMPLX(AFRT(I, J), AFIT(I, J))
212 CONTINUE
211 CONTINUE

```



```

      IF (DABS(C) - .000001) 1, 1, 3
1 DO 7 J=K1, NX
C
C   TRY TO INTERCHANGE ROWS TO GET NON ZERO
C   DIAGONAL COEFFICIENT
C
      IF (DABS(A(J,K)) - 0.000001) 7, 7, 5
5 DO 6 LL=K, NX
  C=A(K, LL)
  A(K, LL)=A(J, LL)
6 A(J, LL)=C
  C=B(K)
  B(K)=B(J)
  B(J)=C
  C=A(K, K)
  GO TO 3
7 CONTINUE
8 WRITE(IMP, 2) K
  D=0.0
  GO TO 300
C
C   DIVIDE ROW BY DIAGONAL COEFFICIENT
C
3 C=A(K, K)
  DO 4 J=K1, NX
4 A(K, J)=A(K, J)/C
  B(K)=B(K)/C
C
C   ELIMINATE UNKNOWN X(K) FROM ROW I
C
  DO 10 I=K1, NX
  C=A(I, K)
  DO 9 J=K1, NX
9 A(I, J)=A(I, J)-C*A(K, J)
10 B(I)=B(I)-C*B(K)
100 CONTINUE
C
C   COMPUTE LAST UNKNOWN
C
      IF (DABS(A(NX, NX)) - 0.000001) 8, 8, 101
101 B(NX)=B(NX)/A(NX, NX)
C
C   APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING
C   UNKNOWN
C
  DO 200 LL=1, N1
  K=NX-LL
  K1=K+1
  DO 200 J=K1, NX
200 B(K)=B(K)-A(K, J)*B(J)
C
C   COMPUTE VALUE OF DETERMINATE
  D=1.0
  DO 250 I=1, NX
250 D=D*A(I, I)
  WRITE(IMP, 333) D
333 FORMAT('THE DETERMINANT =', E14.8)
  2 FORMAT('**** SINGULARITY IN ROW ****', I5)
300 RETURN
  END

```



```

C
C -----
C
C SUBROUTINE JACOBI(AS,BM,EIGV,X,ME,ND)
C *****
C
C REAL EIGV(ND),X(ND,ND), BM(ND,ND)
C REAL AS(ND,ND),D(150)
C
C AS=A : STIFFNESS MATRIX
C BM=F : MASS MATRIX
C NT=ME : TOTAL DOF
C X : FINAL EIGENVECTOR
C EIGV: FINAL EIGENVALUE
C IFPR=0 :FOR PRINTOUT OF INTERM. RESULTS
C IFPR=1 :FOR PRINTOUT OF INTERM. RESULTS
C
C CONSTANSTS
C
C NSMAX = 15
C RTOL = 10.**(-12)
C IFPR = 0
C
C
C NT= ME
C
C INITIALIZE EIGENVALUE AND EIGENVECTOR MATRICES
C
C DO 10 I = 1,NT
C IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 9
C WRITE (*,2020)
C STOP
9 D(I) = AS(I,I)/BM(I,I)
C EIGV(I) = D(I)
10 CONTINUE
C DO 30 I = 1,NT
C DO 20 J = 1,NT
20 X(I,J) = 0.
30 X(I,I) = 1.
C IF (NT.EQ. 1) THEN
C WRITE (*,*) ' EIGENVALUE : ',EIGV(1)
C RETURN
C ELSE
C ENDIF
C
C INITIALIZE SWEEP COUNTER AND EIGEN ITERATION
C
C NSWEEP = 0
C NR = NT - 1
40 NSWEEP = NSWEEP + 1
C IF (IFPR .EQ. 1) WRITE (*, 2000) NSWEEP
C
C CHECK IF PRESENT OFF-DIAGONAL ELEMENTS ARE LARGE ENOUGH TO
C REQUIRE ZEROING
C
C EPS = (.01**NSWEEP)**2
C DO 210 J = 1,NR
C JJ = J + 1
C DO 210 K = JJ,NT
C EPTOLA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))

```

```

EPTOLB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
IF((EPTOLA .LT.EPS) .AND. (EPTOLB .LT. EPS)) GOTO 210
C
C   IF ZEROING IS REQUIRED CALCULATE THE ROT MAT ELM CA ,CG
C
AKK = AS(K,K) * BM(J,K) - BM(K,K) * AS(J,K)
AJJ = AS(J,J) * BM(J,K) - BM(J,J) * AS(J,K)
AB = AS(J,J) * BM(K,K) - AS(K,K) * BM(J,J)
CHECK = (AB*AB + 4.* AKK * AJJ)/4.
IF (CHECK .LT. 0.) THEN
    WRITE (*,2020)
    STOP
ELSE
ENDIF
60 SQCH = SQRT(CHECK)
D1 = AB/2. + SQCH
D2 = AB/2. - SQCH
DEN = D1
IF (ABS(D2) .GT. ABS(D1)) THEN
DEN = D2
ELSE
ENDIF
70 IF(DEN .EQ. 0.) THEN
CA = 0.
CG = -AS(J,K)/AS(K,K)
ELSE
80 CA = (AKK)/(DEN)
CG = - (AJJ)/(DEN)
ENDIF
C
C   PERFORM GENERALIZED ROT TO ZERO THE PRESENT OFF DG. ELM
C
90 IF (NT .EQ. 2) GOTO 190
100 JP1 = J + 1
JM1 = J - 1
KP1 = K + 1
KM1 = K - 1
IF (JM1 .LT. 1) GOTO 130
110 DO 120 I = 1, JM1
AJ = AS(I,J)
BJ = BM(I,J)
AK = AS(I,K)
BK = BM(I,K)
AS(I,J) = AJ + CG * AK
BM(I,J) = BJ + CG * BK
AS(I,K) = AK + CA * AJ
120 BM(I,K) = BK + CA * BJ
130 IF (KP1 .GT. NT) GOTO 160
140 DO 150 I = KP1, NT
AJ = AS(J,I)
BJ = BM(J,I)
AK = AS(K,I)
BK = BM(K,I)
AS(J,I) = AJ + CG * AK
BM(J,I) = BJ + CG * BK
AS(K,I) = AK + CA * AJ
150 BM(K,I) = BK + CA * BJ
160 IF(JP1 .GT. KM1) GOTO 190
170 DO 180 I = JP1, KM1
AJ = AS(J,I)

```

```

      BJ = BM(J,I)
      AK = AS(I,K)
      BK = BM(I,K)
      AS(J,I) = AJ + CG * AK
      BM(J,I) = BJ + CG * BK
      AS(I,K) = AK + CA * AJ
180    BM(I,K) = BK + CA * BJ
190    AK = AS(K,K)
      BK = BM(K,K)
      AS(K,K) = AK + 2.* CA * AS(J,K) + CA * CA * AS(J,J)
      BM(K,K) = BK + 2.* CA * BM(J,K) + CA * CA * BM(J,J)
      AS(J,J) = AS(J,J) + 2.* CG * AS(J,K) + CG * CG * AK
      BM(J,J) = BM(J,J) + 2.* CG * BM(J,K) + CG * CG * BK
      AS(J,K) = 0.
      BM(J,K) = 0.

C
C    UPDATE EIGENVALUE MATRIX AFTER EACH ROTATION
C
      DO 200 I = 1,NT
      XJ = X(I,J)
      XK = X(I,K)
      X(I,J) = XJ + CG * XK
      X(I,K) = XK + CA * XJ
200    CONTINUE
210    CONTINUE

C
C    UPDATE EIGENVALUE AFTER EACH SWEEP
C
      DO 220 I = 1,NT
      IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 220
      WRITE (*,2020)
      STOP
220    EIGV(I) = AS(I,I)/BM(I,I)
      IF (IFPR .EQ. 0) GOTO 230
      WRITE (*,2030)
      WRITE (*,710) (EIGV(I), I = 1,NT)

C
C    CHECK FOR CONVERGENCE
C
230    DO 240 I = 1,NT
      TOL = RTOL * D(I)
      DIF = ABS(EIGV(I) - D(I))
      IF (DIF .GT. TOL) GOTO 280
240    CONTINUE

C
C    CHECK ALL OFF DIAG, TO SEE IF ANOTHER SWEEP IS REQD.
C
      EPS = RTOL ** 2
      DO 250 J = 1,NR
      JJ = J + 1
      DO 253 K = JJ,NT
      EPSA = (AS(J,K)*AS(J,K)) / (AS(J,J)*AS(K,K))
      EPSB = (BM(J,K)*BM(J,K)) / (BM(J,J)*BM(K,K))
      IF ((EPSA .LT. EPS) .AND. (EPSB .LT. EPS)) GOTO 253
      GOTC 280
253    CONTINUE
250    CONTINUE

C
C    FILL BOTTOM TRIANG OF RESLT MATRICES AND SCALE EG. VECT.
C

```

```

255 DO 260 I = 1,NT
    DO 260 J = 1,NT
    AS(J,I) = AS(I,J)
260 BM(J,I) = BM(I,J)
    DO 270 J = 1,NT
    BB = SQRT(BM(J,J))
    DO 270 K = 1,NT
270 X(K,J) = X(K,J)/BB
C
C      SORT      EIGENVALUE AND VECTOR
C
510 KOUNT = 0
    DO 520 I = 1,NT-1
    IF (EIGV(I) .GT. EIGV(I+1)) THEN
    KOUNT = 1
    CHNG = EIGV(I)
    EIGV(I) = EIGV(I+1)
    EIGV(I+1) = CHNG
    DO 530 J = 1,NT
    CHNG = X(J,I)
    X(J,I) = X(J,I+1)
    X(J,I+1) = CHNG
530 CONTINUE
    ENDIF
520 CONTINUE
    IF (KOUNT .EQ. 1) GOTO 510
    IF (IFPR .EQ. 0) GOTO 276
    WRITE (*,960)
    DO 275 K = 1,NT
    WRITE (*,710) (X(K,J), J = 1,NT)
275 CONTINUE
276 RETURN
C
C      UPDATE D MATRIX AND START NEW SWEEP, IF ALLOWED
C
280 DO 290 I = 1,NT
290 D(I) = EIGV(I)
    IF (NSWEEP .LT. NSMAX) GOTO 40
    GOTO 255
710 FORMAT (10(E12.5,1X))

910 FCRMAT (//, ' ** STIFFNESS MATRIX **',/)
920 FORMAT (9(G10.4,1X)/
           5X,9(G10.4,1X))
930 FORMAT (//, ' ** MASS MATRIX **',/)
960 FORMAT (//, ' FINAL EIGEN VECTOR OF MODAL SYSTEM',/)
2000 FORMAT (//, ' SWEEP CYCLE IN JACOBI ' ,I4,/)
2020 FORMAT (10X'ERROR SOLUTION STOP', /
           5X, ' MATRICES NOT POSITIVE DEFINITE')
2030 FORMAT (' CURRENT EIGENVALUE IN *JACOBI* ARE',/)
300 RETURN
    END
C
C      -----
C
SUBROUTINE OUTPUT(XM,YM,FID,FIB,DFI,AFR,AFI,AKR,AKI,
* N0,N1,N2,NOUT,FLAG,DF)
*****
C
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)

```

```

COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
DOUBLE PRECISION XM(1),YM(1),FID(1),FIB(1),DFI(1)
REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
REAL UL(20,20),UB(20,20),VL(20,20),VB(20,20)
DOUBLE PRECISION PRA(20),PIA(20),QRA(20),QIA(20)
COMPLEX XKAP(20),PSI(20,20),TEMP1(20,20),TEMP2(20,20)
COMPLEX SUM3,DX(20),PSIT(20,20)
REAL TEMP1R(20,20),TEMP2R(20,20)
REAL DXR(20),DXI(20),SUM3R
REAL ULT(20,20),UBT(20,20),VLT(20,20),VBT(20,20)
REAL AFRT(20,20),AFIT(20,20),TP(20,20)
LOGICAL FLAG

C
C
RES=0.
WRITE(IMP,500)

C
C
WRITE RESULTS FOR RESERVOIR SURFACE

C
IF(NOUT.EQ.1) THEN
WRITE(IMP,600)
WRITE(IMP,330)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 10 I=1,N0
WRITE(IMP,200) XM(I),YM(I),RES,DFI(I)
10 CONTINUE
IF(FLAG) THEN
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
DO 20 I=1,N0
WRITE(IMP,200) XM(I),YM(I),RES,DFI(I+N)
20 CONTINUE
END IF
END IF

C
C
WRITE RESULTS FOR DAM FACE

C
WRITE(IMP,600)
WRITE(IMP,340)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 30 I=N0+1,N1
WRITE(IMP,200) XM(I),YM(I),DFI(I),FID(I-N0)
30 CONTINUE
IF(FLAG) THEN
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
DO 40 I=N0+1,N1
WRITE(IMP,200) XM(I),YM(I),DFI(N+I),RES
40 CONTINUE

```

```

      END IF
C
C      WRITE RESULTS FOR RESERVOIR BOTTOM
C
      IF (NOUT.EQ.1) THEN
        WRITE (IMP, 600)
        WRITE (IMP, 350)
        WRITE (IMP, 600)
        WRITE (IMP, 100)
        WRITE (IMP, 600)
        WRITE (IMP, 300)
        WRITE (IMP, 600)
        DO 50 I=N1+1, N2
          QR=FIB(I-N1)
          IF (CONST2.GT.1.0E-08) THEN
            QR=QR+CONST1*CONST2*DFI(I+N)
          END IF
          WRITE (IMP, 200) XM(I), YM(I), DFI(I), QR
50      CONTINUE
          IF (FLAG) THEN
            WRITE (IMP, 600)
            WRITE (IMP, 320)
            WRITE (IMP, 600)
            DO 60 I=N1+1, N2
              QI=0
              IF (CONST2.GT.1.0E-08) THEN
                QI=QI-CONST1*CONST2*DFI(I)
              END IF
              WRITE (IMP, 200) XM(I), YM(I), DFI(I+N), QI
60      CONTINUE
            END IF
C
C      WRITE RESULTS FOR THE FAR END
C
      IF (ME.GT.0) THEN
        IF (CONST2.GT.1.0E-08) THEN
          DO 80 J=1, ME
            DO 70 I=1, ME
              UL(I, J)=AFR(I, J)*AKR(J)
              VB(I, J)=AFI(I, J)*AKI(J)
              VL(I, J)=AFI(I, J)*AKR(J)
              UB(I, J)=AFR(I, J)*AKI(J)
70      CONTINUE
80      CONTINUE
C
      CALL TRANSP(TP)
C
      DO 81 I=1, ME
        DO 82 J=1, ME
          TEM1=0.0
          TEM2=0.0
          TEM3=0.0
          TEM4=0.0
          DO 83 K=1, ME
            TEM1=TEM1+TP(I, K)*UL(K, J)
            TEM2=TEM2+TP(I, K)*VB(K, J)
            TEM3=TEM3+TP(I, K)*VL(K, J)
            TEM4=TEM4+TP(I, K)*UB(K, J)
83      CONTINUE
          ULT(I, J)=TEM1

```

```

      VBT(I,J)=TEM2
      VLT(I,J)=TEM3
      UBT(I,J)=TEM4
82  CONTINUE
81  CONTINUE
C
      DO 105 I=1,ME
      DO 90 J=1,ME
      ULT(I,J)=ULT(I,J)-VBT(I,J)
      VLT(I,J)=VLT(I,J)+UBT(I,J)
90  CONTINUE
105 CONTINUE
      ELSE IF (FLAG.AND.CONST2.LE.1.0E-08) THEN
C
      CALL TRANSP(TP)
C
      DO 120 J=1,ME
      DO 110 I=1,ME
      UL(I,J)=AFR(I,J)*AKR(J)
      VL(I,J)=AFR(I,J)*AKI(J)
110 CONTINUE
120 CONTINUE
      DO 121 I=1,ME
      DO 122 J=1,ME
      TEM1=0.0
      TEM2=0.0
      DO 123 K=1,ME
      TEM1=TEM1+TP(I,K)*UL(K,J)
      TEM2=TEM2+TP(I,K)*VL(K,J)
123 CONTINUE
      ULT(I,J)=TEM1
      VLT(I,J)=TEM2
122 CONTINUE
121 CONTINUE
      END IF
C
      IF (FLAG) THEN
      DO 125 I=1,ME
      SUM1=0
      SUM2=0
      DO 124 J=1,ME
      SUM1=SUM1+ULT(I,J)*DFI(N2+J)-VLT(I,J)*DFI(N2+J+N)
      SUM2=SUM2+VLT(I,J)*DFI(N2+J)+ULT(I,J)*DFI(N2+J+N)
124 CONTINUE
      QRA(I)=-SUM1
      QIA(I)=-SUM2
125 CONTINUE
      END IF
C
C
      CALL TRANSP(TP)
C
      DO 126 I=1,ME
      DO 127 J=1,ME
      TEM1=0.0
      TEM2=0.0
      DO 128 K=1,ME
      TEM1=TEM1+TP(I,K)*AFR(K,J)
      TEM2=TEM2+TP(I,K)*AFI(K,J)
128 CONTINUE

```

```

      AFRT(I,J)=TEM1
      AFIT(I,J)=TEM2
127  CONTINUE
125  CONTINUE
C
      IF(CONST2.GT.1.0E-08) THEN
      IF,DF.GT.0.0) THEN
      DO 111 I=1,ME
      XKAP(I)=CMPLX(AKR(I),AKI(I))
      DO 112 J=1,ME
      PSIT(I,J)=CMPLX(AFRT(I,J),AFIT(I,J))
      PSI(I,J)=CMPLX(AFR(I,J),AFI(I,J))
112  CONTINUE
111  CONTINUE
      DO 113 I=1,ME
      DO 114 J=1,ME
      TEMP1(I,J)=PSIT(I,J)/(XKAP(J)*XKAP(J))
114  CONTINUE
113  CONTINUE
      DO 115 I=1,ME
      DO 116 J=1,ME
      SUM3=(0.0,0.0)
      DO 117 K=1,ME
      SUM3=SUM3+TEMP1(J,K)*PSI(I,K)
117  CONTINUE
      TEMP2(J,I)=SUM3
115  CONTINUE
115  CONTINUE
      DO 118 I=1,ME
      DX(I)=TEMP2(I,1)*SNGL(DF)
118  CONTINUE
      DO 119 I=1,ME
      DXR(I)=REAL(DX(I))
      DXI(I)=AIMAG(DX(I))
119  CONTINUE
      END IF
      DO 140 I=1,ME
      SUM1=0
      SUM2=0
      DO 130 J=1,ME
      SUM1=SUM1+AFRT(I,J)*DFI(N2+J)-AFIT(I,J)*DFI(N2+J+N)
      SUM2=SUM2+AFIT(I,J)*DFI(N2+J)+AFRT(I,J)*DFI(N2+J+N)
130  CONTINUE
      IF(DF.GT.0.0) THEN
      PRA(I)=SUM1+DXR(I)
      PIA(I)=SUM2+DXI(I)
      ELSE
      PRA(I)=SUM1
      PIA(I)=SUM2
      END IF
140  CONTINUE
      ELSE IF(CONST2.LE.1.0E-08.AND.FLAG) THEN
      IF(DF.GT.0.0) THEN
      DO 141 I=1,ME
      DO 142 J=1,ME
      IF(AKI(J).GT.0.0) THEN
      TEMP1R(I,J)=-AFRT(I,J)/(AKI(J)*AKI(J))
      ELSE
      TEMP1R(I,J)=AFRT(I,J)/(AKR(J)*AKR(J))
      END IF

```



```

142 CONTINUE
141 CONTINUE
    DO 145 I=1,ME
    DO 146 J=1,ME
    SUM3R=0.0
    DO 147 K=1,ME
    SUM3R=SUM3R+TEMP1R(J,K)*AFR(I,K)
147 CONTINUE
    TEMP2R(J,I)=SUM3R
146 CONTINUE
145 CONTINUE
    DO 148 I=1,ME
    DXR(I)=TEMP2R(I,1)*SNGL(DF)
148 CONTINUE
    END IF
    DO 160 I=1,ME
    SUM1=0
    SUM2=0
    DO 150 J=1,ME
    SUM1=SUM1+AFRT(I,J)*DFI(N2+J)
    SUM2=SUM2+AFRT(I,J)*DFI(N2+J+N)
150 CONTINUE
    IF(DF.GT.0.0) THEN
    IF(FLAG) THEN
    PRA(I)=SUM1+DXR(I)
    PIA(I)=SUM2
    ELSE
    PRA(I)=SUM1+DXR(I)
    PIA(I)=SUM2+DXI(I)
    END IF
    ELSE
    PRA(I)=SUM1
    PIA(I)=SUM2
    END IF
160 CONTINUE
    END IF
C
C
    IF(.NOT.FLAG) THEN
    IF(DF.GT.0.0) THEN
    DO 311 I=1,ME
    DO 312 J=1,ME
    TEMP1R(I,J)=AFRT(I,J)/(AKR(J)*AKR(J))
312 CONTINUE
311 CONTINUE
    DO 313 I=1,ME
    DO 314 J=1,ME
    SUM3R=0.0
    DO 315 K=1,ME
    SUM3R=SUM3R+TEMP1R(J,K)*AFR(I,J)
315 CONTINUE
    TEMP2R(J,I)=SUM3R
314 CONTINUE
313 CONTINUE
    DO 316 I=1,ME
    DXR(I)=TEMP2R(I,1)*SNGL(DF)
316 CONTINUE
    END IF
    DO 180 I=1,ME
    SUM1=0.

```

```

      DO 170 J=1,ME
      SUM1=SUM1+AFRT(I,J)*DFI(N2+J)
170  CONTINUE
      IF(DF.GT.0.0) THEN
      PRA(I)=SUM1+DXR(I)
      ELSE
      PRA(I)=SUM1
      END IF
180  CONTINUE
C
C
      DO 205 J=1,ME
      DO 190 I=1,ME
      UL(I,J)=AFRT(I,J)*AKR(J)
190  CONTINUE
205  CONTINUE
      DO 220 I=1,ME
      SUM1=0
      DO 210 J=1,ME
      SUM1=SUM1+UL(I,J)*DFI(N2+J)
210  CONTINUE
      QRA(I)=-SUM1
220  CONTINUE
      END IF
C
C
      WRITE(IMP,600)
      WRITE(IMP,360)
      WRITE(IMP,600)
      WRITE(IMP,100)
      WRITE(IMP,600)
      WRITE(IMP,300)
      WRITE(IMP,600)
      DO 230 I=1,ME
      K1=N2+I
      WRITE(IMP,200) XM(K1), YM(K1), PRA(I), QRA(I)
230  CONTINUE
      IF(FLAG) THEN
      WRITE(IMP,600)
      WRITE(IMP,320)
      WRITE(IMP,600)
      DO 240 I=1,ME
      K1=N2+I
      WRITE(IMP,200) XM(K1), YM(K1), PIA(I), QIA(I)
240  CONTINUE
      END IF
      END IF
      END IF
C
C
      COMPUTE TOTAL PPESSURE ON DAM
C
      NE=N1-N0
      DY=HY/NE
      TPR=DFI(N1)
      DO 250 I=N0+1,N1-1
      TPR=TPR+DFI(I)
250  CONTINUE
      TPR=DY*TPR
      WRITE(IMP,600)
      WRITE(IMP,370) TPR

```

```

C
C
      IF (FLAG) THEN
      TPI=DFI(N1+N)
      DO 260 I=N0+1,N1-1
      TPI=TPI+DFI(I+N)
260  CONTINUE
      TPI=TPI*DY
      WRITE(IMP,600)
      WRITE(IMP,380) TPI
      TPR=DSQRT(TPR**2+TPI**2)
      WRITE(IMP,600)
      WRITE(IMP,390) TPR
      WRITE(IMP,600)
      END IF

C
C
320  FORMAT(1X,'IMAGINARY PART')
200  FORMAT(4(4X,E14.5))
100  FORMAT(1X,'BOUNDARY  ELEMENTS',//
      * ,10X,'X',17X,'Y',15X,'PRESSURE',5X,'PRESSURE DERIVATIVE')
300  FORMAT(1X,'REAL PART',)
330  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE'
      *)
340  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE')
350  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM')
360  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON FAR END')
370  FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, REAL PART',E14.5)
380  FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, IMAGINARY PART',E14.5)
390  FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, ABSOLUTE VALUE',E14.5)
500  FORMAT(1X,70(' '))
600  FORMAT(1X,' ')
      RETURN
      END

C
C
-----
C
      SUBROUTINE ADDUND(H,G,HOLD,AFR,AKR,AKI,N0,N1,N2,DFI,FID,FIB,
      *FLAG,DF)
C
C
      *****
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/  HY,PI,ME,MF,CONST1,CONST2
      REAL AFR(ND,ND),AKR(ND),AKI(ND)
      DOUBLE PRECISION H(NU,NU),G(NU,NU),T(100,20)
      DOUBLE PRECISION FID(NU),FIB(NU),DFI(NU),HOLD(NU,NU)
      REAL U(20,20),V(20,20)
      REAL TEMP1(20,20),TEMP2(20,20)
      REAL SUM3,DXR(20)
      REAL UT(20,20),VT(20,20),AFRT(20,20),TP(20,20)
      LOGICAL FLAG

C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C
      IF (ME.GT.0) THEN

```

```

DO 20 J=1,ME
DO 10 I=1,ME
U(I,J)=AFR(I,J)*AKR(J)
10 CONTINUE
20 CONTINUE
C
CALL TRANSP(TP)
C
DO 21 I=1,ME
DO 22 J=1,ME
TEM1=0.0
DO 23 K=1,ME
TEM1=TEM1+TP(I,K)*U(K,J)
23 CONTINUE
UT(I,J)=TEM1
22 CONTINUE
21 CONTINUE
C
IF (FLAG) THEN
DO 40 J=1,ME
DO 30 I=1,ME
V(I,J)=AFR(I,J)*AKI(J)
30 CONTINUE
40 CONTINUE
C
DO 41 I=1,ME
DO 42 J=1,ME
TEM1=0.0
DO 43 K=1,ME
TEM1=TEM1+TP(I,K)*V(K,J)
43 CONTINUE
VT(I,J)=TEM1
42 CONTINUE
41 CONTINUE
END IF
C
C
DO 51 I=1,ME
DO 52 J=1,ME
TEM1=0.0
DO 53 K=1,ME
TEM1=TEM1+TP(I,K)*AFR(K,J)
53 CONTINUE
AFRT(I,J)=TEM1
52 CONTINUE
51 CONTINUE
C
DO 70 I=1,N
DO 60 J=1,ME
SUM1=0
SUM2=0
DO 50 K=1,ME
KK=K+N2
SUM1=SUM1+H(I,KK)*DBLE(AFRT(K,J))
SUM2=SUM2+G(I,KK)*DBLE(UT(K,J))
50 CONTINUE
T(I,J)=SUM1+SUM2
60 CONTINUE
70 CONTINUE
C

```

```

      DO 75 I=1,N
      DO 72 J=1,ME
      JJ=J+N2
      H(I,JJ)=T(I,J)
72  CONTINUE
75  CONTINUE
C
C  IMAGINARY PART
C
      IF (FLAG) THEN
      DO 120 I=1,N
      DO 110 J=1,ME
      SUM1=0
      DO 100 K=1,ME
      KK=K+N2
      SUM1=SUM1+G(I, KK) *DBLE (VT(K, J) )
100  CONTINUE
      T(I, J) =SUM1
110  CONTINUE
120  CONTINUE
C
      DO 140 I=1,N
      DO 130 J=1,ME
      H(N+I, J+N2) =T(I, J)
130  CONTINUE
140  CONTINUE
C
C  FILL IN THE REST OF H MATRIX
C
C
      DO 160 I=1,N
      DO 150 J=1,N
      H(I, J+N) =-H(I+N, J)
      H(I+N, J+N) =H(I, J)
150  CONTINUE
160  CONTINUE
      END IF
      END IF
C
C  FORM RIGHT HAND VECTOR
C
      DO 180 I=1,N
      DO 170 J=N0+1, N1
      DFI(I) =DFI(I) +G(I, J) *FID(J-N0)
170  CONTINUE
180  CONTINUE
C
      DO 200 I=1,N
      DO 190 J=N1+1, N2
      DFI(I) =DFI(I) +G(I, J) *FIB(J-N1)
190  CONTINUE
200  CONTINUE
C
      IF (DF.GT.0.0) THEN
      DO 210 I=1,ME
      DO 220 J=1,ME
      IF (AKI(J).GT.0.0) THEN
      TEMP1(I, J) =-AFRT(I, J) / (AKI(J) *AKI(J) )
      ELSE
      TEMP1(I, J) =AFRT(I, J) / (AKR(J) *AKR(J) )

```

```

      END IF
220  CONTINUE
210  CONTINUE
      DO 230 I=1,ME
      DO 240 J=1,ME
      SUM3=0.0
      DO 250 K=1,ME
      SUM3=SUM3+TEMP1(J,K)*AFR(I,K)
250  CONTINUE
      TEMP2(J,I)=SUM3
240  CONTINUE
230  CONTINUE
      DO 260 I=1,ME
      DXR(I)=-TEMP2(I,1)*SNGL(DF)
260  CONTINUE
      DO 290 I=1,N
      DO 300 J=1,ME
      K1=J-ME+N
      DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))
300  CONTINUE
290  CONTINUE
      END IF
      RETURN
      END

C
C -----
C
      SUBROUTINE TRANSP(TP)
C *****
C
C TRANSP COMPUTES THE MATRIX OF TRANSFORMATION TP
C FOR ADJUSTMENT OF LINEARLY FORMED FAR END CONDITION
C INTO CONSTANT ELEMENT FORMULATION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      REAL TP(20,20)
C
      DO 10 I=1,ME
      DO 20 J=1,ME
      TP(I,J)=0.0
20  CONTINUE
10  CONTINUE
C
      DO 30 I=1,ME-1
      N1=I+1
      DO 40 J=I,N1
      TP(I,J)=0.5
40  CONTINUE
30  CONTINUE
      TP(ME,ME)=0.5
      DO 50 I=1,ME
      WRITE(6,*)(TP(I,J),J=1,ME)
50  CONTINUE
      RETURN
      END
C
C
C***** THIS IS THE LAST CARD OF THE CONSTANT BEM PROGRAM*****

```

1.3 Linear Element Program for 2D - BEML.2D

The linear element program for 2D reservoirs - BEML.2D is based on the formulation presented in detail in Chapters 2 and 3 of Volume 1.

1.3.1 Description of the BEML.2D Program.

The linear element program was developed before the constant element program. It can be used to calculate the hydrodynamic pressure on a rigid dam storing a two-dimensional finite or infinite reservoir and subjected to a horizontal or/and vertical harmonic motion. The program uses linear boundary elements. The pressures and pressure derivatives are assumed to vary linearly along the elements.

Similar to the BEMC.2D program, there are no limitations on the shape of the dam, the finite reservoir or the finite portion of an infinite reservoir. The infinite portion of the reservoir has a uniform, rectangular shape. The upstream dam face, the surface, the bottom and the far end of the reservoir, are to be covered by a linear boundary element mesh.

The main program defines the limiting number of boundary nodes it can handle. The number is set to 101, as shown in the listing, but can be easily changed to accommodate more elements. The main program calls the eight following subroutines:

- INPUT - to read all data and print the first part of the output.
- EIGEN - to set and calculate the complex eigenvalue problem in case of infinite radiation with foundation damping included.
- EIGENR - to set and calculate the real eigenvalue problem in case of infinite radiation only (undamped case).
- FMAT - to form matrices G and H and rearrange them according to the boundary conditions on the reservoir top.
- ADDFBC - to modify the matrix H for boundary conditions on the bottom of the reservoir and its far end. This subroutine computes also the vector of known values of the R.H.S.

of the system and forms the system of linear equations for damped case.

ADDUND - to form the system of linear equations and apply all boundary conditions for the undamped case.

SLNPD - to solve the system of linear equations by Gauss elimination.

OUTPUT - to reorder the vector of computed unknowns and print out the rest of the output file.

The subroutine **EIGEN** calls, similar to the **BEMC.2D** program, the subroutine:

FSFEM - to compute the real and imaginary parts of the eigenvector matrix and create the real and imaginary parts of the eigenvector.

The subroutine **FSFEM** uses the **QZ** algorithm (1), in the same form as one incorporate in the **BEMC.2D** program. The subroutines of this algorithm: **CQZHES**, **CQZVAL**, **CQZVEC** and the subroutines **SET** and **NORMA** have been already described in the previous section.

The subroutines **EIGENR** calls, similar to the constant element program, two subroutines: **SETR** and **JACOBI**. The descriptions of these subroutines are included in Subsection 1.2.1.

The subroutine **FMAT** calls the following subroutines:

INTE - to compute the **G** and **H** matrix elements except those on the diagonal using 4-point Gauss quadrature integration scheme.

INLOG - to compute the diagonal elements of **G** matrix.

INLOH - to compute the diagonal elements of **H** matrix.

1.3.2 Input Data.

The following table shows the order of presentation of data in BEML.2D program.

TABLE 1-2

Input Data for the BEML.2D Program

Card Group	No. of Cards	Format	FORTTRAN variables
A	1	18A4	TITLE
B1	1	General Format	N = number of boundary elements N0 = node number at the top of the dam N1 = node number at the base of the dam N2 = node number at the far end of the base NOUT = 1, flag for detailed output
B2	1	General Format	HY = height of the dam C = velocity of sound in water ALFA = wave reflection coefficient DF = vertical acceleration of the first node at the far end
C	N	General Format	X, Y = coordinates of nodal points
D	N1-N0+1	General Format	FID = boundary conditions on the dam face
E	N2-N1+1	General Format	FIB = boundary conditions on the foundation base
F	No of cases	General Format	OMEGA = excitation frequency

The node numbers must start from the element at the top of the reservoir adjacent to the far end and to be given in counterclockwise direction looking into the reservoir scheme.

1.3.3 Output.

The programs prints out the input data as well as the hydrodynamic pressure value at each boundary node. The real and imaginary parts of the total pressure and pressure derivative are given. The real and imaginary parts of the total pressure acting on the dam face and its absolute maximum are also given. Intermediate results, e.g. for an eigenvalue problem, are possible to be obtained.

1.3.4 Example.

For the purpose of illustrations, a simple example is presented here.

A two-dimensional infinite reservoir is solved using 22 linear boundary elements. The far end is modelled by 5 linear one-dimensional finite elements. The system is subjected to a horizontal harmonic excitation with a frequency equal to 27.144 rad/s ($\omega/\omega_1 = 1.2$). The bottom of the reservoir is a flexible soil with reflection coefficient $\alpha_R = 0.75$. The model of the reservoir is shown in Fig. 1.2.

Fig. 1.2 The Linear Boundary Element Model of Infinite 2D Reservoir - $N = 22$

First, the input file is showed and it is followed by the output file.

First, the input file is showed and it is followed by the output file.

• Input file - TEST2DC

```

2D INFINITE RESERVOIR THE BEML
22 7 12 17 1
100. 1440.0 0.75 0.0
  100.0      99.9
  99.9      100.0
  80.0      100.0
  60.0      100.0
  40.0      100.0
  20.0      100.0
   0.0      100.0
   0.0      80.0
   0.0      60.0
   0.0      40.0
   0.0      20.0
   0.0       0.0
  20.0       0.0
  40.0       0.0
  60.0       0.0
  80.0       0.0
  99.9       0.0
 100.0       0.1
 100.0      20.0
 100.0      40.0
 100.0      60.0
 100.0      80.0
  0.101937
  0.101937
  0.101937
  0.101937
  0.101937
  0.101937
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
27.144

```

• Output file - OUT2DC

DATA

NUMBER OF BOUNDARY ELEMENTS 22

NODE NUMBER AT TOP OF RESERVOIR AT DAM 7

NODE NUMBER AT BASE OF DAM 12

NODE NUMBER AT FAR END OF BOTTOM 17

FLAG FOR DETAILED OUTPUT 1

HEIGHT OF DAM 0.10000E+03

VELOCITY OF SOUND IN WATER 0.14400E+04

WAVE REFLECTION COEFFICIENT 0.75000E+00

FOUNDATION DAMPING COEFFICIENT 0.99206E-04

COORDINATES OF THE NODAL POINTS

POINT	X	Y
1	0.1000000E+03	0.9990000E+02
2	0.9990000E+02	0.1000000E+03
3	0.8000000E+02	0.1000000E+03
4	0.6000000E+02	0.1000000E+03
5	0.4000000E+02	0.1000000E+03
6	0.2000000E+02	0.1000000E+03
7	0.0000000E+00	0.1000000E+03
8	0.0000000E+00	0.8000000E+02
9	0.0000000E+00	0.6000000E+02
10	0.0000000E+00	0.4000000E+02
11	0.0000000E+00	0.2000000E+02
12	0.0000000E+00	0.0000000E+00
13	0.2000000E+02	0.0000000E+00
14	0.4000000E+02	0.0000000E+00
15	0.6000000E+02	0.0000000E+00
16	0.8000000E+02	0.0000000E+00
17	0.9990000E+02	0.0000000E+00
18	0.1000000E+03	0.1000000E+00
19	0.1000000E+03	0.2000000E+02
20	0.1000000E+03	0.4000000E+02
21	0.1000000E+03	0.6000000E+02
22	0.1000000E+03	0.8000000E+02

BOUNDARY CONDITIONS ON THE DAM FACE

NODE	PRESCRIBED VALUE
1	0.1019370E+00
2	0.1019370E+00
3	0.1019370E+00
4	0.1019370E+00
5	0.1019370E+00
6	0.1019370E+00

BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM

NODE	PRESCRIBED VALUE
1	0.0000000E+00
2	0.0000000E+00
3	0.0000000E+00
4	0.0000000E+00
5	0.0000000E+00
6	0.0000000E+00

EXCITING FREQUENCY 0.27144E+02

WAVE CONSTANT 0.18850E-01

THE DETERMINANT =0.27708077E-08

PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE

BOUNDARY NODES

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			
0.10000E+03	0.99900E+02	0.00000E+00	-0.56750E+00
0.99900E+02	0.10000E+03	0.00000E+00	0.71722E+00
0.80000E+02	0.10000E+03	0.00000E+00	0.10185E-01
0.60000E+02	0.10000E+03	0.00000E+00	0.68296E-01
0.40000E+02	0.10000E+03	0.00000E+00	0.21610E-01
0.20000E+02	0.10000E+03	0.00000E+00	-0.13161E-01
0.00000E+00	0.10000E+03	0.00000E+00	-0.19939E+00

IMAGINARY PART

0.10000E+03	0.99900E+02	0.00000E+00	-0.49237E+00
0.99900E+02	0.10000E+03	0.00000E+00	0.61458E+00
0.80000E+02	0.10000E+03	0.00000E+00	0.48243E-01
0.60000E+02	0.10000E+03	0.00000E+00	0.14147E+00
0.40000E+02	0.10000E+03	0.00000E+00	0.14791E+00
0.20000E+02	0.10000E+03	0.00000E+00	0.16040E+00
0.00000E+00	0.10000E+03	0.00000E+00	0.17169E+00

PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE

BOUNDARY NODES

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			
0.00000E+00	0.10000E+03	0.00000E+00	0.10194E+00
0.00000E+00	0.80000E+02	0.19385E+01	0.10194E+00
0.00000E+00	0.60000E+02	0.22974E+01	0.10194E+00
0.00000E+00	0.40000E+02	0.20414E+01	0.10194E+00
0.00000E+00	0.20000E+02	0.14982E+01	0.10194E+00
0.00000E+00	0.00000E+00	0.88148E+00	0.10194E+00

IMAGINARY PART

0.00000E+00	0.10000E+03	0.00000E+00	0.00000E+00
0.00000E+00	0.80000E+02	-0.30784E+01	0.00000E+00
0.00000E+00	0.60000E+02	-0.58998E+01	0.00000E+00
0.00000E+00	0.40000E+02	-0.81133E+01	0.00000E+00
0.00000E+00	0.20000E+02	-0.95215E+01	0.00000E+00
0.00000E+00	0.00000E+00	-0.10076E+02	0.00000E+00

PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM

BOUNDARY NODES

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			
0.00000E+00	0.00000E+00	0.88148E+00	-0.27134E-01
0.20000E+02	0.00000E+00	-0.10105E+01	-0.26134E-01
0.40000E+02	0.00000E+00	-0.29378E+01	-0.24146E-01
0.60000E+02	0.00000E+00	-0.46344E+01	-0.21288E-01
0.80000E+02	0.00000E+00	-0.59729E+01	-0.17823E-01
0.99900E+02	0.00000E+00	-0.57076E+01	-0.11829E-01
IMAGINARY PART			
0.00000E+00	0.00000E+00	-0.10076E+02	-0.23737E-02
0.20000E+02	0.00000E+00	-0.97048E+01	0.27210E-02
0.40000E+02	0.00000E+00	-0.89667E+01	0.79111E-02
0.60000E+02	0.00000E+00	-0.79053E+01	0.12480E-01
0.80000E+02	0.00000E+00	-0.66187E+01	0.16084E-01
0.99900E+02	0.00000E+00	-0.43926E+01	0.15376E-01

PRESSURE AND PRESSURE DERIVATIVE ON FAR END

BOUNDARY NODES

X	Y	PRESSURE	PRESSURE DERIVATIVE
REAL PART			
0.10000E+03	0.10000E+00	-0.58938E+01	-0.11381E+00
0.10000E+03	0.20000E+02	-0.62831E+01	0.38404E-02
0.10000E+03	0.40000E+02	-0.51376E+01	-0.45150E-01
0.10000E+03	0.60000E+02	-0.38808E+01	-0.24021E-01
0.10000E+03	0.80000E+02	-0.24384E+01	0.24178E-01
IMAGINARY PART			
0.10000E+03	0.10000E+00	-0.41980E+01	0.21029E-01
0.10000E+03	0.20000E+02	-0.49072E+01	0.88891E-01
0.10000E+03	0.40000E+02	-0.46415E+01	0.79488E-01
0.10000E+03	0.60000E+02	-0.35109E+01	0.40617E-01
0.10000E+03	0.80000E+02	-0.22620E+01	0.60965E-01

TOTAL PRESSURE ON DAM FACE, REAL PART 0.16433E+03

TOTAL PRESSURE ON DAM FACE, IMAGINARY PART -0.63302E+03

TOTAL PRESSURE ON DAM FACE, ABSOLUTE VALUE 0.65400E+03

1.3.5 Listing of the BEML.2D Program.

User: jablonski

Node: 16156

Date: Sat Nov 12 23:09:57 1988

File: beaml.2d.ftn

Pages: 44

```

*****
PROGRAM FOR THE SOLUTION OF TWO DIMENSIONAL RESERVOIR
PROBLEM WITH ONE INFINITE BOUNDARY AND DAMPING INCLUDED
BY LINEAR BOUNDARY ELEMENT METHOD
SPECIAL B.C. AT FAR BOUNDARY THROUGH FEM, DAMPING INCLUDED

```

```

DATE: JANUARY 29, 1987,
      MODIFIED JANUARY 30, 1987,
      VERTICAL EXCITATION INCLUDED MARCH 10, 1987

```

```

*****
N = NUMBER OF BOUNDARY ELEMENTS, READ THROUGH INPUT
X = ARRAY CONTAINING X COORDINATES OF THE NODES
Y = ARRAY CONTAINING Y COORDINATES OF THE NODES
G = MATRIX IN THE BOUNDARY ELEMENT EQUATION
H = MATRIX IN THE BOUNDARY ELEMENT EQUATION
FID= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS
      ON THE DAM FACE
FIB= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS
      ON THE BASE
DFI = RIGHT HAND SIDE VECTOR IN BOUNDARY INTEGRAL EQUATION;
      AFTER SOLUTION, IT CONTAINS TP' VALUES AT THE BOUNDARY
      THAT WERE PREVIOUSLY UNKNOWN
LEC = UNIT NO. CONTAINING INPUT
IMP = UNIT NO. ON WHICH OUTPUT IS WRITTEN
NX = 2*N = DIMENSION OF THE SYSTEM OF EQUATIONS
NU = DIMENSION OF H AND G MATRICES IN THE MAIN PROGRAM
ND = DIMENSION OF THE SPECIAL BOUNDARY CONDITION MATRIX AT
      FAR BOUNDARY IN THE MAIN PROGRAM
D = VALUE OF THE DETERMINANT OF COEFFICIENT MATRIX
CONST = OMEGA/C
OMEGA = EXCITING FREQUENCY
C = VELOCITY OF SOUND IN WATER
CONST1=OMEGA
CONST2=FOUNDATION DAMPING COEFFICIENT GAMMA
AFR=REAL PART OF EIGENVECTOR SI
AFI=IMAGINARY PART OF EIGENVECTOR SI
AKR=REAL PART OF KAPPA MATRIX
AKI=IMAGINARY PART OF KAPPA MATRIX
N0=NODE NUMBER AT TOP OF RESERVOIR AT DAM
N1=NODE NUMBER AT THE BASE OF THE DAM
N2=NODE NUMBER AT THE FAR END OF THE BASE
MF=NUMBER OF NODES ON BASE
ME=NUMBER OF NODES ON THE FAR BOUNDARY NOT INCLUDING THE TOP NODE
HY=HEIGHT OF THE RESERVOIR

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
DOUBLE PRECISION X(101),Y(101),G(200,200),H(200,200),
&DFI(200),FID(50),FIB(50),HOLD(200,200)
REAL AFI(20,20),AFR(20,20),AKR(20),AKI(20)
LOGICAL FLAG
LEC=2
IMP=3
ND=20
NU=200

```



```

PI=3.141592653
OPEN (LEC,FORM='FORMATTED',STATUS='OLD',FILE='TEST2DL')
OPEN (IMP,FORM='FORMATTED',STATUS='NEW',FILE='OUT2DL')
OPEN (6,FORM='FORMATTED',STATUS='NEW',FILE='EIGEN2DL')
C
C  READ DATA FROM INPUT FILE
C
CALL INPUT(X,Y,FID,FIB,N0,N1,N2,NOUT,C,DF)
NX=2*N
5 WRITE(IMP,300)
READ(LEC,*,END=15) OMEGA
CONST=OMEGA/C
WRITE(IMP,350) OMEGA,CONST
CONST1=OMEGA
C
C  COMPUTE EIGENVALUES AND EIGENVECTORS FOR THE FAR BOUNDARY
C
IF(ME.GT.0) THEN
IF(CONST2.GT.1.0E-08) THEN
FLAG=.TRUE.
CALL EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI)
ELSE
CALL EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI,IMP,FLAG)
END IF
write(6,2001)
2001 format(' ')
* DO 10 I=1,ME
* WRITE(6,2010) AKR(I),AKI(I)
* 10 CONTINUE
* write(6,2001)
* do 2003 i=1,me
* write(6,2010) (afr(i,j),j=1,me)
* 2003 continue
* write(6,2001)
END IF
C
C  FORM SYSTEM OF EQUATIONS
C
C
CALL FMAT(X,Y,G,H,HOLD,FID,FIB,DFI,N0,N1)
* do 2000 i=1,n
* write(6,2010) (h(i,j),j=1,n)
* 2000 continue
* write(6,2001)
* write(6,2001)
* do 2002 i=1,n
* write(6,2010) (g(i,j),j=1,n)
* 2002 continue
C
C  MODIFY H MATRIX FOR BOUNDARY CONDITIONS AT BASE AND FAR END
C  AND COMPUTE DFI
C
IF(CONST2.GT.1.0E-08) THEN
CALL ADDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,DFI,FID,FIB,DF)
ELSE
CALL ADDUND(H,G,HOLD,AFR,AKR,AKI,N0,N1,N2,DFI,FID,FIB,FLAG,DF)
END IF
* write(6,2001)
* write(6,2001)

```

```

*      do 2005 i=1,nx
*      write(6,2010) (h(i,j),j=1,nx)
* 2005 continue
*      write(6,2001)
*      write(6,2001)
*      write(6,2010) (dfi(i),i=1,nx)

```

C
C
C
C

SOLVE SYSTEM OF EQUATIONS

```

IF (FLAG) THEN
  CALL SLNPD (H,DFI,D,NX,NU,IMP)
ELSE
  CALL SLNPD (H,DFI,D,N,NU,IMP)
END IF
*      DO 999 I=1,N
*      WRITE (IMP,*) DFI (I)
* 999  CONTINUE
*      IF (FLAG) THEN
*      DO 1000 I=1,N
*      WRITE (IMP,*) DFI (I+N)
* 1000 CONTINUE
*      END IF

```

C
C
C
C

OUTPUT RESULTS

```

  CALL OUTPUT (X,Y,FID,FIB,DFI,AFR,AFI,AKR,AKI,
* NO,N1,N2,NOUT,FLAG,DF)
  GO TO 5
2010 format (6e13.5)
300 FORMAT (//,1X,70 (' '),//)
350 FORMAT (1X,'EXCITING FREQUENCY',E14.5,/,1X,
*'WAVE CONSTANT',E14.5,/)
15 STOP
END

```

C
C
C

SUBROUTINE INPUT (X,Y,FID,FIB,N0,N1,N2,NOUT,C,DF)

C
C
C
C

READS INPUT DATA

C
C
C
C

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
DOUBLE PRECISION X(1),Y(1),FID(1),FIB(1),C
CHARACTER*4 TITLE(19)
WRITE (IMP,100)

```

C
C
C

READ TITLE

C

READ (LEC,150) TITLE

C

READ BASIC PARAMETERS

C

```

READ (LEC,*) N,N0,N1,N2,NOUT
READ (LEC,*) HY,C,ALF,DF
CONST2=((1-ALF)/(1+ALF))*(1/C)

```

```

C      WRITE(IMP,300)N,N0,N1,N2,NOUT
      WRITE(IMP,350)HY,C,ALF,CONST2

C
C      READ COORDINATES OF NODAL POINTS OF THE
C      BOUNDARY ELEMENTS IN ARRAY X AND Y
C
      WRITE(IMP,500)
      DO 10 I=1,N
      READ(LEC,*)X(I),Y(I)
10    WRITE(IMP,700)I,X(I),Y(I)

C
C      READ BOUNDARY CONDITION ON THE DAM FACE
C
      WRITE(IMP,760)
      WRITE(IMP,800)
      ILAST=N1-N0+1
      DO 20 I=1,ILAST
      READ(LEC,*)FID(I)
      WRITE(IMP,750)I,FID(I)
20    CONTINUE

C
C      READ BOUNDARY CONDITION ON THE BASE OF THE RESERVOIR
C
      WRITE(IMP,770)
      WRITE(IMP,800)
      ILAST=N2-N1+1
      DO 30 I=1,ILAST
      READ(LEC,*)FIB(I)
      WRITE(IMP,750)I,FIB(I)
30    CONTINUE
      ME=N-N2
      MF=N2-N1+1

C
100  FORMAT(1X,60(' '))//)
150  FORMAT(18A4)
300  FORMAT(1X,'DATA',//,1X,'NUMBER OF BOUNDARY ELEMENTS',
      *I3,//,1X,'NODE NUMBER AT TOP OF RESERVOIR AT DAM',I3,//,
      *1X,'NODE NUMBER AT BASE OF DAM',I3,//,1X,
      *'NODE NUMBER AT FAR END OF BOTTOM',I3,//,1X,
      *'FLAG FOR DETAILED OUTPUT',I3,/)
350  FORMAT(1X,'HEIGHT OF DAM',E14.5,//,1X,
      *'VELOCITY OF SOUND IN WATER',E14.5,//,1X,
      *'WAVE REFLECTION COEFFICIENT',E14.5,//,1X,
      *'FOUNDATION DAMPING COEFFICIENT',E14.5//)
400  FORMAT(2(2X,E14.7,2X))
500  FORMAT(' COORDINATES OF THE NODAL POINTS',
      *//,4X,'POINT',10X,'X',18X,'Y'//)
700  FORMAT(5X,I3,2(5X,E14.7))
800  FORMAT(5X,'NODE',6X,
      *'PRESCRIBED VALUE'//)
750  FORMAT(5X,I3,8X,E14.7)
760  FORMAT(/// ' BOUNDARY CONDITIONS ON THE DAM FACE',//)
770  FORMAT(/// ' BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM',//)

C
C      RETURN

```

END

C
C
C

SUBROUTINE EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI)

C

DOUBLE PRECISION CONST,CONST1,CONST2,HY

C

REAL AFI(ND,ND),AFR(ND,ND),AKR(ND),AKI(ND)

C

DY=SNGL(HY/ME)

T=SNGL(CONST)

T1=SNGL(CONST1)

T2=SNGL(CONST2)

C

CALL FSFEM(AFI,AFR,ME,ND,T,T1,T2,DY,AKR,AKI)

C

RETURN

END

C
C
C

SUBROUTINE FSFEM(AFI,AFR,N,NM,T,T1,T2,DY,AKR,AKI)

C

C

AFR= REAL PART OF THE EIGENVECTOR MATRIX

C

AFI= IMAGINARY PART OF EIGENVECTOR MATRIX

C

AKR=REAL PART OF KAPPA MATRIX DIAGONAL

C

AKI=IMAGINARY PART OF KAPPA MATRIX DIAGONAL

C

N=ME= NO. OF ELEMENTS ON THE FAR BOUNDARY

C

NM=ND= MAX. DIMENSION OF THE EIGENVECTOR MATRIX

C

DY= PASSED LENGTH OF AN ELEMENT (DP IN THE MAIN PROGRAM)

C

T=CONST= OMEGA/C (DP IN THE MAIN PROGRAM)

C

T1=CONST1= OMEGA (DP IN THE MAIN PROGRAM)

C

T2=CONST2=GAMMA

C

REAL A(20, 20),B(20, 20),Z(20, 20),BHOLD(20, 20),
\$ AHOLD(20, 20),AHOLDI(20, 20),BHOLDI(20,20),
\$ ALFR(20),ALFI(20),BETA(20),
\$ EPS1, AI(20,20),BI(20,20),ZI(20,20)

REAL AMAX1

COMPLEX FI(20,20),XM(20,20)

COMPLEX XKAP(20),XK2(20)

REAL AFR(NM,NM),AFI(NM,NM),AKR(NM),AKI(NM)

INTEGER ERROR

DIMENSION INDEX(20)

DATA IWRITE/6/

C

CALL SET(DY,T1,T2,N,NM,A,AI,B,BI)

DO 717 I=1,N

DO 718 J=1,N

AHOLD(I,J)=A(I,J)

BHOLD(I,J)=B(I,J)

AHOLDI(I,J)=AI(I,J)

BHOLDI(I,J)=BI(I,J)

718 CONTINUE

717 CONTINUE

*

WRITE(IWRITE,20)

*

\$ N

*

20 FORMAT(30H1THE FULL MATRIX A OF ORDER,

*

\$ I4,22H IS (PRINTED BY ROWS)/)

```

*      DO 30 I = 1,N
*      30 WRITE(IWRITE,40)
*      $      (A(I,J),AI(I,J),J=1,N)
*      40 FORMAT(5(2F8.4,1HI,3X))
*      WRITE(IWRITE,41)
*      $      N
*      41 FORMAT(30H0THE FULL MATRIX B OF ORDER,
*      $      I4,22H IS (PRINTED BY ROWS)/)
*      DO 42 I = 1,N
*      42 WRITE(IWRITE,43)
*      $      (B(I,J),BI(I,J),J=1,N)
*      43 FORMAT(5(2F8.4,1HI,3X))
C
C
C
C
C      EIGENVALUES AND EIGENVECTORS USING CQZVAL AND CQZVEC
C
      EPS1 = 0.0
      CALL CQZHES(NM,N,A,AI,B,BI,.TRUE.,Z,ZI)
      CALL CQZVAL(NM,N,A,AI,B,BI,EPS1,ALFR,ALFI,BETA,
$      .TRUE.,Z,ZI,ERROR)
      CALL CQZVEC(NM,N,A,AI,B,BI,ALFR,ALFI,BETA,Z,ZI)
      DO 719 I=1,N
      DO 720 J=1,N
      A(I,J)=AHOLD(I,J)
      B(I,J)=BHOLD(I,J)
      AI(I,J)=AHOLDI(I,J)
      BI(I,J)=BHOLDI(I,J)
720 CONTINUE
719 CONTINUE
*      WRITE(IWRITE,292)
*      292 FORMAT(/15X,7HALFR(I),19X,7HALFI(I),19X,7HBETA(I))
*      DO 295 I = 1,N
*      WRITE(IWRITE,293)
*      $      I,ALFR(I),ALFI(I),BETA(I)
*      293 FORMAT(I2,3(1PE23.6,3X))
*      295 CONTINUE
*      WRITE(IWRITE,300)
*      300 FORMAT(/14X,35HCOMPUTED EIGENVALUE AND EIGENVECTOR,20X)
C
      DO 510 K = 1, N
      BETA(K) = AMAX1(BETA(K),1.0E-50)
      ALFR(K) = ALFR(K) / BETA(K)
      ALFI(K) = ALFI(K) / BETA(K)
C
C      ONE EIGENVECTOR.
C
*      340 WRITE(IWRITE,350)
*      $      K,ALFR(K),ALFI(K),(Z(I,K),
*      $      ZI(I,K),I=1,N)
*      350 FORMAT(/I2,1P2E23.6/(5X,2E23.6))
510 CONTINUE
      DO 610 K=1,N
      DO 620 L=1,N
      FI(K,L)=CMPLX(Z(K,L),ZI(K,L))
620 CONTINUE
610 CONTINUE
*      DO 611 I=1,N
*      WRITE(IWRITE,*) (FI(I,J),J=1,N)

```

```

* 611 CONTINUE
    CALL NORMA(FI,B,N,NM,XM)
*    DO 81 I=1,N
*      WRITE(IWRITE,*) (FI(I,J),J=1,N)
* 81 CONTINUE
    DO 613 I=1,N
    DO 612 J=1,N
      AFR(I,J)=REAL(FI(I,J))
      AFI(I,J)=AIMAG(FI(I,J))
612 CONTINUE
613 CONTINUE
    DO 171 I=1,N
      ALFR(I)=ALFR(I)-T*T
      XK2(I)=CMPLX(ALFR(I),ALFI(I))
      XKAP(I)=CSQRT(XK2(I))
*      WRITE(IWRITE,*) XKAP(I)
      AKR(I)=REAL(XKAP(I))
      AKI(I)=AIMAG(XKAP(I))
171 CONTINUE
    RETURN
    END

C
C -----
C
SUBROUTINE EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI
1 ,IMP,FLAG)
C *****
C DOUBLE PRECISION CONST,CONST1,CONST2,HY
C REAL AFR(ND,ND),AKR(ND),AKI(ND)
C REAL ALFR(20),A(20,20),F(20,20)
C LOGICAL FLAG

C DO 10 I=1,ME
C   AKR(I)=0
C   AKI(I)=0
10 CONTINUE
  DY=SNGL(HY/ME)
  T=SNGL(CONST)
  T1=SNGL(CONST1)
  T2=SNGL(CONST2)
  FLAG=.FALSE.
  CALL SETR(DY,ME,ND,A,F)
  CALL JACOBI(A,F,ALFR,AFR,ME,ND)
  DO 30 I=1,ME
    TEMP=ALFR(I)-T*T
    TX=TEMP
    IF(TEMP.LT.0) THEN
      FLAG=.TRUE.
      AKI(I)=SQRT(-TX)
    ELSE IF(TEMP.GT.0) THEN
      AKR(I)=SQRT(TX)
    ELSE
      WRITE(IMP,100)
100 FORMAT(1X,'RESONANCE HAS TAKEN PLACE')
      STOP
    ENL IF
  30 CONTINUE
  RETURN
  END

C

```

 SUBROUTINE CQZHES (NM,N,AR,AI,BR,BI,MATZ,ZR,ZI)

 THIS ALGORITHM QZ FOR THE CLASS OF COMPLEX GENERALIZED MATRIX
 SYSTEMS EXHIBITING THE USE OF QZ TO FIND ALL THE EIGENVALUES
 AND EIGENVECTORS FOR THE EIGENPROBLEM $A^*X = (\text{LAMBDA}) * B^*X$.

THE DIMENSION OF A,AI,B,BI,Z, AND ZI SHOULD BE NM BY NM.
 THE DIMENSION OF ALFR,ALFI,BETA, AND NORM SHOULD BE NM.
 THE DIMENSION OF AHOLD AND BHOLD SHOULD BE NM BY NM.
 THE DIMENSION OF AHOLDI AND BHOLDI SHOULD BE NM BY NM.
 HERE NM = 20.

INTEGER I,J,K,L,N,K1,LB,L1,NM,NK1,NM1
 REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ZR(NM,N),ZI(NM,N)
 REAL R,S,T,TI,U1,U2,XI,XR,YI,YR,RHO,U11
 REAL SQRT,CABS,ABS
 LOGICAL MATZ
 COMPLEX CMLX

THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FIRST STEP OF THE
 QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
 SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.

THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX GENERAL MATRICES AND
 REDUCES ONE OF THEM TO UPPER HESSENBERG FORM WITH REAL (AND NON-
 NEGATIVE) SUBDIAGONAL ELEMENTS AND THE OTHER TO UPPER TRIANGULAR
 FORM USING UNITARY TRANSFORMATIONS. IT IS USUALLY FOLLOWED BY
 CQZVAL AND POSSIBLY CQZVEC.

ON INPUT-

NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
 ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
 DIMENSION STATEMENT,

N IS THE ORDER OF THE MATRICES,

A=(AR,AI) CONTAINS A COMPLEX GENERAL MATRIX,

B=(BR,BI) CONTAINS A COMPLEX GENERAL MATRIX,

MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS
 ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING
 EIGENVECTORS, AND TO .FALSE. OTHERWISE.

ON OUTPUT-

A HAS BEEN REDUCED TO UPPER HESSENBERG FORM. THE ELEMENTS
 BELOW THE FIRST SUBDIAGONAL HAVE BEEN SET TO ZERO, AND THE
 SUBDIAGONAL ELEMENTS HAVE BEEN MADE REAL (AND NON-NEGATIVE),

B HAS BEEN REDUCED TO UPPER TRIANGULAR FORM. THE ELEMENTS
 BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,

Z=(ZR,ZI) CONTAINS THE PRODUCT OF THE RIGHT HAND
 TRANSFORMATIONS IF MATZ HAS BEEN SET TO .TRUE.
 OTHERWISE, Z IS NOT REFERENCED.

```

C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      ***** INITIALIZE Z *****
C      IF (.NOT. MATZ) GO TO 10
C
C      DO 3 I = 1, N
C
C          DO 2 J = 1, N
C              ZR(I,J) = 0.0
C              ZI(I,J) = 0.0
C      2    CONTINUE
C
C          ZR(I,I) = 1.0
C      3    CONTINUE
C      ***** REDUCE B TO UPPER TRIANGULAR FORM WITH
C      TEMPORARILY REAL DIAGONAL ELEMENTS *****
C      10 IF (N .LE. 1) GO TO 170
C          NM1 = N - 1
C
C          DO 100 L = 1, NM1
C              L1 = L + 1
C              S = 0.0
C
C              DO 20 I = L, N
C                  S = S + ABS(BR(I,L)) + ABS(BI(I,L))
C      20    CONTINUE
C
C              IF (S .EQ. 0.0) GO TO 100
C              RHO = 0.0
C
C              DO 25 I = L, N
C                  BR(I,L) = BR(I,L) / S
C                  BI(I,L) = BI(I,L) / S
C                  RHO = RHO + BR(I,L)**2 + BI(I,L)**2
C      25    CONTINUE
C
C              R = SQRT(RHO)
C              XR = CABS(CMPLX(BR(L,L),BI(L,L)))
C              IF (XR .EQ. 0.0) GO TO 27
C              RHO = RHO + XR * R
C              U1 = -BR(L,L) / XR
C              U1I = -BI(L,L) / XR
C              YR = R / XR + 1.0
C              BR(L,L) = YR * BR(L,L)
C              BI(L,L) = YR * BI(L,L)
C              GO TO 28
C
C      27    BR(L,L) = R
C            U1 = -1.0
C            U1I = 0.0
C
C      28    DO 50 J = L1, N
C            T = 0.0
C            TI = 0.0
C
C            DO 30 I = L, N
C                T = T + BR(I,L) * BR(I,J) + BI(I,L) * BI(I,J)

```



```

          TI = TI + BR(I,L) * BI(I,J) - BI(I,L) * BR(I,J)
30      CONTINUE
C
          T = T / RHO
          TI = TI / RHO
C
          DO 40 I = L, N
              BR(I,J) = BR(I,J) - T * BR(I,L) + TI * BI(I,L)
              BI(I,J) = BI(I,J) - T * BI(I,L) - TI * BR(I,L)
40      CONTINUE
C
          XI = U1 * BI(L,J) - U1I * BR(L,J)
          BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
          BI(L,J) = XI
50      CONTINUE
C
          DO 80 J = 1, N
              T = 0.0
              TI = 0.0
C
              DO 60 I = L, N
                  T = T + BR(I,L) * AR(I,J) + BI(I,L) * AI(I,J)
                  TI = TI + BR(I,L) * AI(I,J) - BI(I,L) * AR(I,J)
60          CONTINUE
C
              T = T / RHO
              TI = TI / RHO
C
              DO 70 I = L, N
                  AR(I,J) = AR(I,J) - T * BR(I,L) + TI * BI(I,L)
                  AI(I,J) = AI(I,J) - T * BI(I,L) - TI * BR(I,L)
70          CONTINUE
C
              XI = U1 * AI(L,J) - U1I * AR(L,J)
              AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
              AI(L,J) = XI
80          CONTINUE
C
              BR(L,L) = R * S
              BI(L,L) = 0.0
C
              DO 90 I = L1, N
                  BR(I,L) = 0.0
                  BI(I,L) = 0.0
90          CONTINUE
C
100     CONTINUE
C
          ***** REDUCE A TO UPPER HESSENBERG FORM WITH REAL SUBDIAGONAL
          ***** ELEMENTS, WHILE KEEPING B TRIANGULAR *****
C
          DO 160 K = 1, NM1
              K1 = K + 1
C
              ***** SET BOTTOM ELEMENT IN K-TH COLUMN OF A REAL *****
              IF (AI(N,K) .EQ. 0.0) GO TO 105
              R = CABS(CMPLX(AR(N,K),AI(N,K)))
              U1 = AR(N,K) / R
              U1I = AI(N,K) / R
              AR(N,K) = R
              AI(N,K) = 0.0
C
              DO 103 J = K1, N

```

```

      XI = U1 * AI(N,J) - U1I * AR(N,J)
      AR(N,J) = U1 * AR(N,J) + U1I * AI(N,J)
      AI(N,J) = XI
103  CONTINUE
C
      XI = U1 * BI(N,N) - U1I * BR(N,N)
      BR(N,N) = U1 * BR(N,N) + U1I * BI(N,N)
      BI(N,N) = XI
105  IF (K .EQ. NM1) GO TO 170
      NK1 = NM1 - K
C
      ***** FOR L=N-1 STEP -1 UNTIL K+1 DO -- *****
      DO 150 LB = 1, NK1
        L = N - LB
        L1 = L + 1
C
      ***** ZERO A(L+1,K) *****
      S = ABS(AR(L,K)) + ABS(AI(L,K)) + AR(L1,K)
      IF (S .EQ. 0.0) GO TO 150
      U1 = AR(L,K) / S
      U1I = AI(L,K) / S
      U2 = AR(L1,K) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      AR(L,K) = R * S
      AI(L,K) = 0.0
      AR(L1,K) = 0.0
C
      DO 110 J = K1, N
        XR = AR(L,J)
        XI = AI(L,J)
        YR = AR(L1,J)
        YI = AI(L1,J)
        AR(L,J) = U1 * XR + U1I * XI + U2 * YR
        AI(L,J) = U1 * XI - U1I * XR + U2 * YI
        AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110  CONTINUE
C
      XR = BR(L,L)
      BR(L,L) = U1 * XR
      BI(L,L) = -U1I * XR
      BR(L1,L) = -U2 * XR
C
      DO 120 J = L1, N
        XR = BR(L,J)
        XI = BI(L,J)
        YR = BR(L1,J)
        YI = BI(L1,J)
        BR(L,J) = U1 * XR + U1I * XI + U2 * YR
        BI(L,J) = U1 * XI - U1I * XR + U2 * YI
        BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
120  CONTINUE
C
      ***** ZERO B(L+1,L) *****
      S = ABS(BR(L1,L1)) + ABS(BI(L1,L1)) + ABS(BR(L1,L))
      IF (S .EQ. 0.0) GO TO 150
      U1 = BR(L1,L1) / S
      U1I = BI(L1,L1) / S
      U2 = BR(L1,L) / S

```

```

      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      BR(L1,L1) = R * S
      BI(L1,L1) = 0.0
      BR(L1,L) = 0.0
C
      DO 130 I = 1, L
        XR = BR(I,L1)
        XI = BI(I,L1)
        YR = BR(I,L)
        YI = BI(I,L)
        BR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        BI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        BR(I,L) = U1 * YR - U1I * YI - U2 * XR
        BI(I,L) = U1 * YI + U1I * YR - U2 * XI
130      CONTINUE
C
      DO 140 I = 1, N
        XR = AR(I,L1)
        XI = AI(I,L1)
        YR = AR(I,L)
        YI = AI(I,L)
        AR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        AI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        AR(I,L) = U1 * YR - U1I * YI - U2 * XR
        AI(I,L) = U1 * YI + U1I * YR - U2 * XI
140      CONTINUE
C
      IF (.NOT. MATZ) GO TO 150
C
      DO 145 I = 1, N
        XR = ZR(I,L1)
        XI = ZI(I,L1)
        YR = ZR(I,L)
        YI = ZI(I,L)
        ZR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        ZI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        ZR(I,L) = U1 * YR - U1I * YI - U2 * XR
        ZI(I,L) = U1 * YI + U1I * YR - U2 * XI
145      CONTINUE
C
150      CONTINUE
C
160 CONTINUE
C
170 RETURN
C
      ***** LAST CARD OF CQZHES *****
      END
C
C
C
      SUBROUTINE CQZVAL(NM,N,AR,AI,BR,BI,EPS1,ALFR,ALFI,BETA,
X      MATZ,ZR,ZI,IERR)
C
C      *****
C
      INTEGER I,J,K,L,N,EN,K1,K2,LL,L1,NA,NM,ITS,KM1,LM1,
X      ENM2,IERR,LOR1,ENORN
      REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ALFR(N),ALFI(N),

```

```

X      BETA(N), ZR(NM,N), ZI(NM,N)
REAL R, S, A1, A2, EP, SH, U1, U2, XI, XR, YI, YR, ANI, A1I, A33, A34, A43, A44,
X      BNI, B1I, B33, B44, SHI, U1I, A33I, A34I, A43I, A44I, B33I, B44I,
X      EPSA, EPSB, EPS1, ANORM, BNORM, B3344, B3344I
REAL SQRT, CABS, ABS
INTEGER MAX0
LOGICAL MATZ
COMPLEX Z3
COMPLEX CSQRT, CMPLX
REAL REAL, AIMAG

```

THIS SUBROUTINE IS A COMPLEX ANALOGUE OF STEPS 2 AND 3 OF THE QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS, SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART, AS MODIFIED IN TECHNICAL NOTE NASA TN E-7305(1973) BY WARD.

THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES, ONE OF THEM IN UPPER HESSENBERG FORM AND THE OTHER IN UPPER TRIANGULAR FORM, THE HESSENBERG MATRIX MUST FURTHER HAVE REAL SUBDIAGONAL ELEMENTS. IT REDUCES THE HESSENBERG MATRIX TO TRIANGULAR FORM USING UNITARY TRANSFORMATIONS WHILE MAINTAINING THE TRIANGULAR FORM OF THE OTHER MATRIX AND FURTHER MAKING ITS DIAGONAL ELEMENTS REAL AND NON-NEGATIVE. IT THEN RETURNS QUANTITIES WHOSE RATIOS GIVE THE GENERALIZED EIGENVALUES. IT IS USUALLY PRECEDED BY CQZHEs AND POSSIBLY FOLLOWED BY CQZVEC.

ON INPUT-

NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT,

N IS THE ORDER OF THE MATRICES,

A=(AR,AI) CONTAINS A COMPLEX UPPER HESSENBERG MATRIX WITH REAL SUBDIAGONAL ELEMENTS,

B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,

EPS1 IS A TOLERANCE USED TO DETERMINE NEGLIGIBLE ELEMENTS. EPS1 = 0.0 (OR NEGATIVE) MAY BE INPUT, IN WHICH CASE AN ELEMENT WILL BE NEGLECTED ONLY IF IT IS LESS THAN ROUNDOFF ERROR TIMES THE NORM OF ITS MATRIX. IF THE INPUT EPS1 IS POSITIVE, THEN AN ELEMENT WILL BE CONSIDERED NEGLIGIBLE IF IT IS LESS THAN EPS1 TIMES THE NORM OF ITS MATRIX. A POSITIVE VALUE OF EPS1 MAY RESULT IN FASTER EXECUTION, BUT LESS ACCURATE RESULTS,

MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING EIGENVECTORS, AND TO .FALSE. OTHERWISE,

Z=(ZR,ZI) CONTAINS, IF MATZ HAS BEEN SET TO .TRUE., THE TRANSFORMATION MATRIX PRODUCED IN THE REDUCTION BY CQZHEs, IF PERFORMED, OR ELSE THE IDENTITY MATRIX. IF MATZ HAS BEEN SET TO .FALSE., Z IS NOT REFERENCED.

```

C
C      ON OUTPUT
C
C      A HAS BEEN REDUCED TO UPPER TRIANGULAR FORM.  THE ELEMENTS
C      BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,
C
C      B IS STILL IN UPPER TRIANGULAR FORM, ALTHOUGH ITS ELEMENTS
C      HAVE BEEN ALTERED.  IN PARTICULAR, ITS DIAGONAL HAS BEEN SET
C      REAL AND NON-NEGATIVE.  THE LOCATION BR(N,1) IS USED TO
C      STORE EPS1 TIMES THE NORM OF B FOR LATER USE BY  CQZVEC,
C
C      ALFR AND ALFI CONTAIN THE REAL AND IMAGINARY PARTS OF THE
C      DIAGONAL ELEMENTS OF THE TRIANGULARIZED A MATRIX,
C
C      BETA CONTAINS THE REAL NON-NEGATIVE DIAGONAL ELEMENTS OF THE
C      CORRESPONDING B.  THE GENERALIZED EIGENVALUES ARE THEN
C      THE RATIOS ((ALFR+I*ALFI)/BETA),
C
C      Z CONTAINS THE PRODUCT OF THE RIGHT HAND TRANSFORMATIONS
C      (FOR BOTH STEPS) IF MATZ HAS BEEN SET TO .TRUE.,
C
C      IERR IS SET TO
C      ZERO      FOR NORMAL RETURN,
C      J          IF AR(J,J-1) HAS NOT BECOME
C                ZERO AFTER 50 ITERATIONS.
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      IERR = 0
C      ***** COMPUTE EPSA,EPSB *****
C      ANORM = 0.0
C      BNORM = 0.0
C
C      DO 30 I = 1, N
C          ANI = 0.0
C          IF (I .NE. 1) ANI = ABS(AR(I,I-1))
C          BNI = 0.0
C
C          DO 20 J = I, N
C              ANI = ANI + ABS(AR(I,J)) + ABS(AI(I,J))
C              BNI = BNI + ABS(BR(I,J)) + ABS(BI(I,J))
C      20  CONTINUE
C
C          IF (ANI .GT. ANORM) ANORM = ANI
C          IF (BNI .GT. BNORM) BNORM = BNI
C      30  CONTINUE
C
C      IF (ANORM .EQ. 0.0) ANORM = 1.0
C      IF (BNORM .EQ. 0.0) BNORM = 1.0
C      EP = EPS1
C      IF (EP .GT. 0.0) GO TO 50
C      ***** COMPUTE ROUNDOFF LEVEL IF EPS1 IS ZERO *****
C      EP = 1.0
C      40 EP = EP / 2.0
C          IF (1.0 + EP .GT. 1.0) GO TO 40
C      50 EPSA = EP * ANORM
C          EPSB = EP * BNORM

```

```

C      ***** REDUCE A TO TRIANGULAR FORM, WHILE
C      KEEPING B TRIANGULAR *****
      LOR1 = 1
      ENORN = N
      EN = N
C      ***** BEGIN QZ STEP *****
60  IF (EN .EQ. 0) GO TO 1001
      IF (.NOT. MATZ) ENORN = EN
      ITS = 0
      NA = EN - 1
      ENM2 = NA - 1
C      ***** CHECK FOR CONVERGENCE OR REDUCIBILITY.
C      FOR L=EN STEP -1 UNTIL 1 DO -- *****
70  DO 80 LL = 1, EN
      LM1 = EN - LL
      L = LM1 + 1
      IF (L .EQ. 1) GO TO 95
      IF (ABS(AR(L,LM1)) .LE. EPSA) GO TO 90
80  CONTINUE
C
90  AR(L,LM1) = 0.0
C      ***** SET DIAGONAL ELEMENT AT TOP OF B REAL *****
95  B11 = CABS(CMPLX(BR(L,L),BI(L,L)))
      IF (B11 .EQ. 0.0) GO TO 98
      U1 = BR(L,L) / B11
      U1I = BI(L,L) / B11
C
      DO 97 J = L, ENORN
        XI = U1 * AI(L,J) - U1I * AR(L,J)
        AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
        AI(L,J) = XI
        XI = U1 * BI(L,J) - U1I * BR(L,J)
        BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
        BI(L,J) = XI
97  CONTINUE
C
      BI(L,L) = 0.0
98  IF (L .NE. EN) GO TO 100
C      ***** 1-BY-1 BLOCK ISOLATED *****
      ALFR(EN) = AR(EN,EN)
      ALFI(EN) = AI(EN,EN)
      BETA(EN) = B11
      EN = NA
      GO TO 60
C      ***** CHECK FOR SMALL TOP OF B *****
100 L1 = L + 1
      IF (B11 .GT. EPSB) GO TO 120
      BR(L,L) = 0.0
      S = ABS(AR(L,L)) + ABS(AI(L,L)) + ABS(AR(L1,L))
      U1 = AR(L,L) / S
      U1I = AI(L,L) / S
      U2 = AR(L1,L) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      AR(L,L) = R * S
      AI(L,L) = 0.0
C
      DO 110 J = L1, ENORN

```

```

      XR = AR(L,J)
      XI = AI(L,J)
      YR = AR(L1,J)
      YI = AI(L1,J)
      AR(L,J) = U1 * XR + U1I * XI + U2 * YR
      AI(L,J) = U1 * XI - U1I * XR + U2 * YI
      AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
      AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
      XR = BR(L,J)
      XI = BI(L,J)
      YR = BR(L1,J)
      YI = BI(L1,J)
      BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
      BR(L,J) = U1 * XR + U1I * XI + U2 * YR
      BI(L,J) = U1 * XI - U1I * XR + U2 * YI
      BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110 CONTINUE
C
      LM1 = L
      L = L1
      GO TO 90
C
      ***** ITERATION STRATEGY *****
120 IF (ITS .EQ. 50) GO TO 1000
   IF (ITS .EQ. 10) GO TO 135
C
      ***** DETERMINE SHIFT *****
      B33 = BR(NA,NA)
      B33I = BI(NA,NA)
      IF (CABS(CMPLX(B33,B33I)) .GE. EPSB) GO TO 122
      B33 = EPSB
      B33I = 0.0
122 B44 = BR(EN,EN)
      B44I = BI(EN,EN)
      IF (CABS(CMPLX(B44,B44I)) .GE. EPSB) GO TO 124
      B44 = EPSB
      B44I = 0.0
124 B3344 = B33 * B44 - B33I * B44I
      B3344I = B33 * B44I + B33I * B44
      A33 = AR(NA,NA) * B44 - AI(NA,NA) * B44I
      A33I = AR(NA,NA) * B44I + AI(NA,NA) * B44
      A34 = AR(NA,EN) * B33 - AI(NA,EN) * B33I
      X    = AR(NA,NA) * BR(NA,EN) + AI(NA,NA) * BI(NA,EN)
      A34I = AR(NA,EN) * B33I + AI(NA,EN) * B33
      X    = AR(NA,NA) * BI(NA,EN) - AI(NA,NA) * BR(NA,EN)
      A43 = AR(EN,NA) * B44
      A43I = AR(EN,NA) * B44I
      A44 = AR(EN,EN) * B33 - AI(EN,EN) * B33I - AR(EN,NA) * BR(NA,EN)
      A44I = AR(EN,EN) * B33I + AI(EN,EN) * B33 - AR(EN,NA) * BI(NA,EN)
      SH = A44
      SHI = A44I
      XR = A34 * A43 - A34I * A43I
      XI = A34 * A43I + A34I * A43
      IF (XR .EQ. 0.0 .AND. XI .EQ. 0.0) GO TO 140
      YR = (A33 - SH) / 2.0
      YI = (A33I - SHI) / 2.0
      Z3 = CSQRT(CMPLX(YR**2-YI**2+XR,2.0*YR*YI+XI))
      U1 = REAL(Z3)
      U1I = AIMAG(Z3)
      IF (YR * U1 + YI * U1I .GE. 0.0) GO TO 125
      U1 = -U1
      U1I = -U1I

```

```

125 Z3 = (CMPLX(SH,SHI) - CMPLX(XR,XI) / CMPLX(YR+U1,YI+U1I))
X / CMPLX(B3344,B3344I)
SH = REAL(Z3)
SHI = AIMAG(Z3)
GO TO 140
C ***** AD HOC SHIFT *****
135 SH = AR(EN,NA) + AR(NA,ENM2)
SHI = 0.0
C ***** DETERMINE ZEROth COLUMN OF A *****
140 A1 = AR(L,L) / B11 - SH
A1I = AI(L,L) / B11 - SHI
A2 = AR(L1,L) / B11
ITS = ITS + 1
IF (.NOT. MATZ) LOR1 = L
C ***** MAIN LOOP *****
DO 260 K = L, NA
K1 = K + 1
K2 = K + 2
KM1 = MAX0(K-1,L)
C ***** ZERO A(K+1,K-1) *****
IF (K .EQ. L) GO TO 170
A1 = AR(K,KM1)
A1I = AI(K,KM1)
A2 = AR(K1,KM1)
170 S = ABS(A1) + ABS(A1I) + ABS(A2)
U1 = A1 / S
U1I = A1I / S
U2 = A2 / S
R = SQRT(U1*U1+U1I*U1I+U2*U2)
U1 = U1 / R
U1I = U1I / R
U2 = U2 / R
C
DO 180 J = KM1, ENORN
XR = AR(K,J)
XI = AI(K,J)
YR = AR(K1,J)
YI = AI(K1,J)
AR(K,J) = U1 * XR + U1I * XI + U2 * YR
AI(K,J) = U1 * XI - U1I * XR + U2 * YI
AR(K1,J) = U1 * YR - U1I * YI - U2 * XR
AI(K1,J) = U1 * YI + U1I * YR - U2 * XI
XR = BR(K,J)
XI = BI(K,J)
YR = BR(K1,J)
YI = BI(K1,J)
BR(K,J) = U1 * XR + U1I * XI + U2 * YR
BI(K,J) = U1 * XI - U1I * XR + U2 * YI
BR(K1,J) = U1 * YR - U1I * YI - U2 * XR
BI(K1,J) = U1 * YI + U1I * YR - U2 * XI
180 CONTINUE
C
IF (K .EQ. L) GO TO 240
AI(K,KM1) = 0.0
AR(K1,KM1) = 0.0
AI(K1,KM1) = 0.0
C ***** ZERO B(K+1,K) *****
240 S = ABS(BR(K1,K1)) + ABS(BI(K1,K1)) + ABS(BR(K1,K))
U1 = BR(K1,K1) / S
U1I = BI(K1,K1) / S

```



```

      U2 = BR(K1,K) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      IF (K .EQ. NA) GO TO 245
      XR = AR(K2,K1)
      AR(K2,K1) = U1 * XR
      AI(K2,K1) = -U1I * XR
      AR(K2,K) = -U2 * XR
C
245   DO 250 I = LOR1, K1
      XP = AR(I,K1)
      XI = AI(I,K1)
      YR = AR(I,K)
      YI = AI(I,K)
      AR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      AI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      AR(I,K) = U1 * YR - U1I * YI - U2 * XR
      AI(I,K) = U1 * YI + U1I * YR - U2 * XI
      XR = BR(I,K1)
      XI = BI(I,K1)
      YR = BR(I,K)
      YI = BI(I,K)
      BR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      BI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      BR(I,K) = U1 * YR - U1I * YI - U2 * XR
      BI(I,K) = U1 * YI + U1I * YR - U2 * XI
250   CONTINUE
C
      BI(K1,K1) = 0.0
      BR(K1,K) = 0.0
      BI(K1,K) = 0.0
      IF (.NOT. MATZ) GO TO 260
C
      DO 255 I = 1, N
      XR = ZR(I,K1)
      XI = ZI(I,K1)
      YR = ZR(I,K)
      YI = ZI(I,K)
      ZR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      ZI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      ZR(I,K) = U1 * YR - U1I * YI - U2 * XR
      ZI(I,K) = U1 * YI + U1I * YR - U2 * XI
255   CONTINUE
C
260   CONTINUE
C ***** SET LAST A SUBDIAGONAL REAL AND END QZ STEP *****
      IF (AI(EN,NA) .EQ. 0.0) GO TO 70
      R = CABS(CMPLX(AR(EN,NA),AI(EN,NA)))
      U1 = AR(EN,NA) / R
      U1I = AI(EN,NA) / R
      AR(EN,NA) = R
      AI(EN,NA) = 0.0
C
      DO 270 J = EN, ENORN
      XI = U1 * AI(EN,J) - U1I * AR(EN,J)
      AR(EN,J) = U1 * AR(EN,J) + U1I * AI(EN,J)
      AI(EN,J) = XI
      XI = U1 * BI(EN,J) - U1I * BR(EN,J)

```

```

      BR(EN,J) = U1 * BR(EN,J) + U1I * BI(EN,J)
      BI(EN,J) = XI
270  CONTINUE
C
      GO TO 70
C      ***** SET ERROR -- BOTTOM SUBDIAGONAL ELEMENT HAS NOT
C      BECOME NEGLIGIBLE AFTER 50 ITERATIONS *****
1000 IERR = EN
C      ***** SAVE EPSB FOR USE BY CQZVEC *****
1001 IF (N .GT. 1) BR(N,1) = EPSB
      RETURN
C      ***** LAST CARD OF CQZVAL *****
      END
C
C      -----
C
SUBROUTINE CQZVEC(NM,N,AR,AI,BR,BI,ALFR,ALFI,BETA,ZR,ZI)
C      *****
C
      INTEGER I,J,K,M,N,EN,II,JJ,NA,NM,NN
      REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ALFR(N),ALFI(N),
X      BETA(N),ZR(NM,N),ZI(NM,N)
      REAL R,T,RI,TI,XI,ALMI,ALMR,BETM,EPSB
      REAL CABS
      COMPLEX Z3
      COMPLEX CMLPX
      REAL REAL,AIMAG
C
C
C
C
C      THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FOURTH STEP OF THE
C      QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
C      SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.
C
C      THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES IN UPPER
C      TRIANGULAR FORM, WHERE ONE OF THEM FURTHER MUST HAVE REAL DIAGONAL
C      ELEMENTS. IT COMPUTES THE EIGENVECTORS OF THE TRIANGULAR PROBLEM
C      AND TRANSFORMS THE RESULTS BACK TO THE ORIGINAL COORDINATE SYSTEM.
C      IT IS USUALLY PRECEDED BY CQZHES AND CQZVAL.
C
      ON INPUT-
C
C      NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C      ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C      DIMENSION STATEMENT,
C
C      N IS THE ORDER OF THE MATRICES,
C
C      A=(AR,AI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,
C
C      B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX WITH REAL
C      DIAGONAL ELEMENTS. IN ADDITION, LOCATION BR(N,1) CONTAINS
C      THE TOLERANCE QUANTITY (EPSB) COMPUTED AND SAVED IN CQZVAL,
C
C      ALFR, ALFI, AND BETA ARE VECTORS WITH COMPONENTS WHOSE
C      RATIOS ((ALFR+I*ALFI)/BETA) ARE THE GENERALIZED
C      EIGENVALUES. THEY ARE USUALLY OBTAINED FROM CQZVAL,
C

```

```

C      Z=(ZR,ZI) CONTAINS THE TRANSFORMATION MATRIX PRODUCED IN THE
C      REDUCTIONS BY CQZHE5 AND CQZVAL, IF PERFORMED.
C      IF THE EIGENVECTORS OF THE TRIANGULAR PROBLEM ARE
C      DESIRED, Z MUST CONTAIN THE IDENTITY MATRIX.
C
C      ON OUTPUT-
C
C      A IS UNALTERED,
C
C      B HAS BEEN DESTROYED,
C
C      ALFR, ALFI, AND BETA ARE UNALTERED,
C
C      Z CONTAINS THE EIGENVECTORS. EACH EIGENVECTOR IS NORMALIZED
C      SO THAT THE MODULUS OF ITS LARGEST COMPONENT IS 1.0 .
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      IF (N .LE. 1) GO TO 1001
C      EPSB = BR(N,1)
C      ***** FOR EN=N STEP -1 UNTIL 2 DO -- *****
C      DO 800 NN = 2, N
C          EN = N + 2 - NN
C          NA = EN - 1
C          ALMR = ALFR(EN)
C          ALMI = ALFI(EN)
C          BETM = BETA(EN)
C      ***** FOR I=EN-1 STEP -1 UNTIL 1 DO -- *****
C      DO 700 II = 1, NA
C          I = EN - II
C          R = 0.0
C          RI = 0.0
C          M = I + 1
C
C          DO 610 J = M, EN
C              T = BETM * AR(I,J) - ALMR * BR(I,J) + ALMI * BI(I,J)
C              TI = BETM * AI(I,J) - ALMR * BI(I,J) - ALMI * BR(I,J)
C              IF (J .EQ. EN) GO TO 605
C              XI = T * BI(J,EN) + TI * BR(J,EN)
C              T = T * BR(J,EN) - TI * BI(J,EN)
C              TI = XI
C          605      R = R + T
C                  RI = RI + TI
C          610      CONTINUE
C
C              T = ALMR * BETA(I) - BETM * ALFR(I)
C              TI = ALMI * BETA(I) - BETM * ALFI(I)
C              IF (T .EQ. 0.0 .AND. TI .EQ. 0.0) T = EPSB
C              Z3 = CMPLX(R,RI) / CMPLX(T,TI)
C              BR(I,EN) = REAL(Z3)
C              BI(I,EN) = AIMAG(Z3)
C          700      CONTINUE
C
C      800 CONTINUE
C      ***** END BACK SUBSTITUTION.
C      TRANSFORM TO ORIGINAL COORDINATE SYSTEM.
C      FOR J=N STEP -1 UNTIL 2 DO -- *****

```

```

      DO 880 JJ = 2, N
        J = N + 2 - JJ
        M = J - 1
C
      DO 880 I = 1, N
C
        DO 860 K = 1, M
          ZR(I,J) = ZR(I,J) + ZR(I,K) * BR(K,J) - ZI(I,K) * BI(K,J)
          ZI(I,J) = ZI(I,J) + ZR(I,K) * BI(K,J) + ZI(I,K) * BR(K,J)
860      CONTINUE
C
      880 CONTINUE
C      ***** NORMALIZE SO THAT MODULUS OF LARGEST
C      COMPONENT OF EACH VECTOR IS 1 *****
      DO 950 J = 1, N
        T = 0.0
C
        DO 930 I = 1, N
          R = CABS(CMPLX(ZR(I,J),ZI(I,J)))
          IF (R .GT. T) T = R
930      CONTINUE
C
        DO 940 I = 1, N
          ZR(I,J) = ZR(I,J) / T
          ZI(I,J) = ZI(I,J) / T
940      CONTINUE
C
      950 CONTINUE
C
      1001 RETURN
C      ***** LAST CARD OF CQZVEC *****
      END
C-----
C
C
C      SUBROUTINE NORMA(FI,G,N,NM,XM)
C      *****
C
C      COMPLEX FI(NM,NM),XM(NM,NM),TEMP(20),SUM1,SUM2
C      COMPLEX XI(20,20),X(20,20),X1,X2
C      REAL G(NM,NM)
C      DATA IWRITE/6/
C
C      DO 30 I=1,N
C      DO 40 J=1,N
C      SUM1=(0.0,0.0)
C      DO 50 K=1,N
C      SUM1=SUM1+CMPLX(G(J,K))*FI(K,I)
50 CONTINUE
C      TEMP(J)=SUM1
40 CONTINUE
C      SUM2=(0.0,0.0)
C      DO 41 J=1,N
C      SUM2=SUM2+FI(J,I)*TEMP(J)
41 CONTINUE
C      SUM2=CSQRT(SUM2)
C      DO 51 J=1,N
C      FI(J,I)=FI(J,I)/SUM2
51 CONTINUE

```

```

      30 CONTINUE
C
C FOR CHECK CARRY OUT [FIT]*[G]*[FI]=[I]
C
      DO 100 I=1,N
      DO 110 J=1,N
      X1=(0.0,0.0)
      DO 120 K=1,N
      X1=X1+FI(K,I)*CMPLX(G(K,J))
120 CONTINUE
      X(I,J)=X1
110 CONTINUE
100 CONTINUE
      DO 130 I=1,N
      DO 140 J=1,N
      X2=(0.0,0.0)
      DO 150 K=1,N
      X2=X2+X(I,K)*FI(K,J)
150 CONTINUE
      XI(I,J)=X2
140 CONTINUE
130 CONTINUE
*      DO 161 I=1,N
*      WRITE(IWRITE,*) (XI(I,J),J=1,N)
* 161 CONTINUE
      RETURN
      END

```

```

C
C -----
C
SUBROUTINE SET(DY,T1,T2,N,NM,H,BO,G,BI)
*****
C
C SET CALCULATES MATRIX [A]p=[H]+i[BO]
C AND MATRIX [B]p=[G]
REAL H(NM,NM),BO(NM,NM),G(NM,NM)
REAL BI(NM,NM)
DO 10 I=1,N
  I1=I-1
  I2=I+1
  DO 20 J=1,N
    IF(I.EQ.1.AND.J.EQ.1) THEN
      H(I,J)=1.0
      G(I,J)=1.0
    ELSE IF (I.NE.1.AND.I.EQ.J) THEN
      H(I,J)=2.0
      G(I,J)=2.0
    ELSE IF (J.EQ.I1.OR.J.EQ.I2) THEN
      H(I,J)=-1.0
      G(I,J)=0.5
    ELSE
      H(I,J)=0.0
      G(I,J)=0.0
    END IF
    H(I,J)=H(I,J)/DY
    G(I,J)=G(I,J)*(DY/3)
  20 CONTINUE
10 CONTINUE
DO 30 I=1,N
DO 40 J=1,N

```

```

      IF(I.EQ.1.AND.J.EQ.1) THEN
      BO(I,J)=T2*T1
      ELSE
      BO(I,J)=0.0
      END IF
40  CONTINUE
30  CONTINUE
      DO 80 I=1,N
      DO 90 J=1,N
      BI(I,J)=0.0
90  CONTINUE
80  CONTINUE
      RETURN
      END

```

C
C
C
C
C
C

SUBROUTINE SETR(DY,ME,ND,A,F)

```

      REAL A(ND,ND),F(ND,ND)
      DO 9 I=1,ME
      DO 8 J=1,ME
      A(I,J)=0.
      F(I,J)=0.
8  CONTINUE
9  CONTINUE
      A(1,1)=1/DY
      A(1,2)=-1./DY
      F(1,1)=DY/3
      F(1,2)=.5*DY/3
      DO 10 I=2,ME-1
      A(I,I)=2/DY
      F(I,I)=2*DY/3
      A(I,I-1)=-1/DY
      A(I,I+1)=-1/DY
      F(I,I-1)=.5*DY/3
      F(I,I+1)=.5*DY/3
10 CONTINUE
      A(ME,ME)=2/DY
      A(ME,ME-1)=-1/DY
      F(ME,ME)=2*DY/3
      F(ME,ME-1)=.5*DY/3
      RETURN
      END

```

C
C
C
C
C
C
C
C

SUBROUTINE FMAT(X,Y,G,H,HOLD,FID,FIB,DFI,N0,N1)

FMAT COMPUTES THE G AND H MATRICES AND FORMS
THE SYSTEMS OF EQUATIONS AX=F

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION X(1),Y(1),G(NU,NU),H(NU,NU),
1  FID(1),FIB(1),DFI(1),HOLD(NU,NU)

```

C

```

      NX=2*N
      DO 10 I=1,NX
        DFI(I)=0
      DO 11 J=1,NX
        G(I,J)=0.0
        H(I,J)=0.0
11 CONTINUE
10 CONTINUE
C
      X(N+1)=X(1)
      Y(N+1)=Y(1)
      DO 110 I=1,N
        NF=I+1
        NS=I+N-2
        DO 50 JJ=NF,NS
          IF(JJ.GT.N) THEN
            J=JJ-N
          ELSE
            J=JJ
          END IF
C
          CALL INTE(X(I),Y(I),X(J),Y(J),X(J+1),Y(J+1),A1,A2,B1,B2)
C
          IF(J.LT.N) THEN
            H(I,J+1)=H(I,J+1)+A2
            IF(J.EQ.N1-1) THEN
              DFI(I)=DFI(I)+B2*FID(N1-N0+1)
            ELSE
              G(I,J+1)=G(I,J+1)+B2
            END IF
            ELSE
              H(I,1)=H(I,1)+A2
              G(I,1)=G(I,1)+B2
            END IF
            H(I,J)=H(I,J)+A1
            IF(J.EQ.N0) THEN
              DFI(I)=DFI(I)+B1*FID(1)
            ELSE
              G(I,J)=G(I,J)+B1
            END IF
          50 CONTINUE
C
          NF=I+N-1
          NS=I+N
          DO 95 JJ=NF,NS
            IF(JJ.GT.N) THEN
              J=JJ-N
            ELSE
              J=JJ
            END IF
C
            CALL INLOG(X(J),Y(J),X(J+1),Y(J+1),B1,B2)
            IF (JJ.LE.NF) THEN
              CH=B1
              B1=B2
              B2=CH
            END IF
            IF (J.LT.N) THEN
              IF(J.EQ.N1-1) THEN
                DFI(I)=DFI(I)+B2*FID(N1-N0+1)

```

```

        ELSE
          G(I,J+1)=G(I,J)+B2
        END IF
      ELSE
        G(I,1)=G(I,1)+B2
      END IF
      IF(J.EQ.N0) THEN
        DFI(I)=DFI(I)+B1*FID(1)
      ELSE
        G(I,J)=G(I,J)+B1
      END IF
95 CONTINUE
C
      IF(I.EQ.1) THEN
        X1=X(N)
        Y1=Y(N)
      ELSE
        X1=X(I-1)
        Y1=Y(I-1)
      END IF
      CALL INLOH(X1,Y1,X(I),Y(I),X(I+1),Y(I+1),A)
      H(I,I)=A
110 CONTINUE
C
C ARRANGE THE SYSTEM OF EQUATIONS
C
      DO 150 J=1,N0
        DO 145 I=1,N
          CH=G(I,J)
          G(I,J)=-H(I,J)
          H(I,J)=-CH
145 CONTINUE
150 CONTINUE
C
      DO 160 I=1,N
        DO 170 J=1,N
          HOLD(I,J)=H(I,J)
170 CONTINUE
160 CONTINUE
C
      RETURN
      END
C
C -----
C
      SUBROUTINE INLOG(X1,Y1,X2,Y2,B1,B2)
      *****
C
C COMPUTES THE INTEGRALS ALONG AN ELEMENT
C WHICH INCLUDES THE NODE I
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION LEN,K
      K=CONST
      LEN=DSQRT((X2-X1)**2+(Y2-Y1)**2)
      B1=LEN*(DLOG(LEN*K)-1.5)/(4.*PI)
      B2=LEN*(DLOG(LEN*K)-0.5)/(4.*PI)
      RETURN

```



```

C      END
C      -----
C      SUBROUTINE INLOH(X1,Y1,X2,Y2,X3,Y3,A)
C      *****
C      THIS SUBROUTINE COMPUTES THE DIAGONAL ELEMENT
C      OF H MATRIX
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C      DOUBLE PRECISION LEN
C      X=X3-X2
C      Y=Y3-Y2
C      LEN=DSQRT(X*X+Y*Y)
C      S=Y/LEN
C      C=X/LEN
C      BETA1=ASIN(S)
C      IF(C.LT.0) BETA1=PI-BETA1
C      X=X1-X2
C      Y=Y1-Y2
C      LEN=DSQRT(X*X+Y*Y)
C      S=Y/LEN
C      C=X/LEN
C      BETA2=ASIN(S)
C      IF(C.LT.0) BETA2=PI-BETA2
C      IF(BETA1.LT.BETA2) BETA1=BETA1+2.*PI
C      BETA=BETA1-BETA2
C      A=-1+BETA/(2.*PI)
C      RETURN
C      END
C      -----
C      SUBROUTINE INTE(XP,YP,X1,Y1,X2,Y2,A1,A2,B1,B2)
C      *****
C      INTG COMPUTES INTEGRALS ALONG AN ELEMENT WHICH DOES
C      NOT INCLUDE NODE I USING FOUR POINT GAUSS QUADRATURE
C      C
C      DIST=DISTANCE FROM I TO ELEMENT J
C      RA=DISTANCE FROM I TO THE INTEGRATION POINT
C      ON THE BOUNDARY ELEMENT
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C      DOUBLE PRECISION XCO(4),YCO(4),GI(4),OME(4),LEN,K
C      K=CONST
C      GI(1)=0.86113631
C      GI(2)=-GI(1)
C      GI(3)=0.33998104
C      GI(4)=-GI(3)
C      OME(1)=0.34785485
C      OME(2)=-OME(1)
C      OME(3)=0.65214515
C      OME(4)=-OME(3)
C      AX=(X2-X1)
C      BX=X2+X1
C      AY=Y2-Y1

```

```

BY=Y2+Y1
IF (AX.EQ.0) THEN
    DIST=DABS (XP-X1)
ELSE
    TA=AY/AX
    DIST=DABS ((TA*XP-YP+Y1-TA*X1)/DSQRT (TA*TA+1))
END IF
SIG=(X1-XP)*(Y2-YP)-(X2-XP)*(Y1-YP)
IF (SIG.LT.0) DIST=-DIST
A1=0
A2=0
B1=0
B2=0
LEN=DSQRT (AX*AX+AY*AY)
DO 40 I=1,4
XCO(I)=(AX*GI(I)+BX)/2
YCO(I)=(AY*GI(I)+BY)/2
RA=DSQRT ((XP-XCO(I))**2+(YP-YCO(I))**2)
CON=K*RA
CALL BFUNCT (CON,YB,YBD)
H=YBD*K*LEN*DIST*OME(I)/(RA*16.)
G=YB*LEN*OME(I)/16.
A1=A1+H*(1-GI(I))
A2=A2+H*(1+GI(I))
B1=B1+G*(1-GI(I))
B2=B2+G*(1+GI(I))
40 CONTINUE
RETURN
END

```

C
C
C

SUBROUTINE BFUNCT (XX,Y0X,Y1X)

C
C

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
REAL*8 XULT0,XOSH0,XOSH1

C

INTEGER XF,Z0,Z
BESSEL FUNCTION CALCULATIONS

```

IF (XX-15.9) 1000,1000,2000
1000 A=0.6366197724
    B=0.5772156646
    XULT0=1.0
    SIGMA=0.0
    TY0=0.0
    TY1=0.0
    DO 7 I=1,500
    XF=I
    Z=-1.0
    Z0=XF+1.0
    SINY0=Z**(XF)
    SINY1=Z**(Z0)
    XOSH0=(XX/2.0)**(2*XF)
    XOSH1=(XX/2.0)**(2*XF-1.0)
    XULT0=XULT0*XF
    XULT1=XULT0/XF
    SIGMA=SIGMA+(1.0/XF)
    VALU0=DLOG (XX/2.0)+B-SIGMA

```

```

      VALU1=VALU0+(1.0/(2.0*XF))
      TRMY0=((SINY0*XOSH0)/(XULT0**2))*(VALU0)
      TRMY1=((SINY1*XOSH1)/(XULT0*XULT1))*(VALU1)
      TY0=TY0+TRMY0
      TY1=TY1+TRMY1
      ERY0=DABS(TRMY0/TY0)
      ERY1=DABS(TRMY1/TY1)
      IF(ERY0-ERY1)1,1,2
1    ER=ERY1
      GO TO 10
2    ER=ERY0
10   IF(ER-.001)11,11,7
7    CONTINUE
11   Y0X=A*(DLOG(XX/2.0)+B+TY0)
      Y1X=A*((1.0/XX)-TY1)
      GO TO 100
2000 AH=PI/4.0
      BH=XX-AH
      CH=XX+AH
      S1H=SIN(BH)
      S2H=SIN(CH)
      DHY0=S2H/(8.0*XX)
      DHY1=S1H*3.0/(8.0*XX)
      HH=XX-(3.0*AH)
      S3H=SIN(HH)
      EH=2.0/(PI*XX)
      FH=DSQRT(EH)
      XMY0=S1H-DHY0
      XMY1=S3H+DHY1
      Y0X=XMY0*FH
      Y1X=-XMY1*FH
100  RETURN
      END

```

C
C
C

```

      SUBROUTINE ADDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,
1  DFI,FID,FIB,DF)

```

C
C

```

      *****
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
      DOUBLE PRECISION H(NU,NU),G(NU,NU),T(100,20)
      DOUBLE PRECISION FID(NU),FIB(NU),DFI(NU),HOLD(NU,NU)
      REAL UL(20,20),VB(20,20),VL(20,20),UB(20,20)
      COMPLEX XKAP(20),PSI(20,20),TEMP1(20,20),TEMP2(20,20)
      COMPLEX SUM3,DX(20)
      REAL DXR(20),DXI(20)

```

C
C
C
C
C
C
C

```

      REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
      CONDITIONS INTO ACCOUNT

```

```

      ARRANGE H MATRIX
      REAL PART

```

```

      IF(ME.GT.0)THEN
      DO 20 J=1,ME
      DO 10 I=1,ME

```

```

      UL(I,J)=AFR(I,J)*AKR(J)
      VB(I,J)=AFI(I,J)*AKI(J)
      VL(I,J)=AFI(I,J)*AKR(J)
      UB(I,J)=AFR(I,J)*AKI(J)
10  CONTINUE
20  CONTINUE
C
      DO 40 I=1,ME
      DO 30 J=1,ME
      UL(I,J)=UL(I,J)-VB(I,J)
      VL(I,J)=VL(I,J)+UB(I,J)
30  CONTINUE
40  CONTINUE
C
      DO 70 I=1,N
      DO 60 J=1,ME
      SUM1=0
      SUM2=0
      DO 50 K=1,ME
      KK=K+N2
      SUM1=SUM1+H(I, KK) * DBLE (AFR(K, J))
      SUM2=SUM2+G(I, KK) *DBLE (VL(K, J))
50  CONTINUE
      T(I, J)=SUM1+SUM2
60  CONTINUE
70  CONTINUE
C
      DO 75 I=1,N
      DO 72 J=1,ME
      JJ=J+N2
      H(I, JJ)=T(I, J)
72  CONTINUE
75  CONTINUE
C
C  IMAGINARY PART
C
C
      DO 120 I=1,N
      DO 110 J=1,ME
      SUM1=0
      SUM2=0
      DO 100 K=1,ME
      KK=K+N2
      SUM1=SUM1+G(I, KK) *DBLE (VL(K, J))
      SUM2=SUM2+H(I, KK) *DBLE (AFI(K, J))
100 CONTINUE
      T(I, J)=SUM1+SUM2
110 CONTINUE
120 CONTINUE
C
      DO 140 I=1,N
      DO 130 J=1,ME
      H(N+I, J+N2)=T(I, J)
130 CONTINUE
140 CONTINUE
      END IF
C
      IF (CONST2.GT.1.E-08) THEN
      TEMP=CONST1*CONST2

```

```

      DO 90 I=1,N
      DO 80 J=N1,N2
      H(N+I,J)=G(I,J)*TEMP
80    CONTINUE
90    CONTINUE
      END IF
C
C  FILL IN THE REST OF H MATRIX
C
C
      DO 160 I=1,N
      DO 150 J=1,N
      H(I,J+N)=-H(I+N,J)
      H(I+N,J+N)=H(I,J)
150   CONTINUE
160   CONTINUE
C
C  FORM RIGHT HAND VECTOR
C
      DO 180 I=1,N
      DO 170 J=N0+1,N1-1
      DFI(I)=DFI(I)+G(I,J)*FID(J-N0+1)
170   CONTINUE
180   CONTINUE
C
      DO 200 I=1,N
      DO 190 J=N1,N2
      DFI(I)=DFI(I)+G(I,J)*FIB(J-N1+1)
190   CONTINUE
200   CONTINUE
C
      IF(DF.GT.0.0) THEN
      DO 211 I=1,ME
      XKAP(I)=CMPLX(AKR(I),AKI(I))
      DO 212 J=1,ME
      PSI(I,J)=CMPLX(AFR(I,J),AFI(I,J))
212   CONTINUE
211   CONTINUE
      DO 210 I=1,ME
      DO 220 J=1,ME
      TEMP1(I,J)=PSI(I,J)/(XKAP(J)*XKAP(J))
220   CONTINUE
210   CONTINUE
      DO 230 I=1,ME
      DO 240 J=1,ME
      SUM3=(0.0,0.0)
      DO 250 K=1,ME
      SUM3=SUM3+TEMP1(J,K)*PSI(I,K)
250   CONTINUE
      TEMP2(J,I)=SUM3
240   CONTINUE
230   CONTINUE
      DO 260 I=1,ME
      DX(I)=-TEMP2(I,1)*SNGL(DF)
260   CONTINUE
      DO 270 I=1,ME
      DXR(I)=REAL(DX(I))
      DXI(I)=AIMAG(DX(I))
270   CONTINUE
      *  WRITE(IMP,101)(DXR(I),DXI(I),I=1,ME)

```

```

* 101 FORMAT(2(4X,E14.5))
      DO 280 I=1,N
      DO 290 J=1,ME
      K1=J-ME+N
      DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))
      DFI(I+N)=DFI(I+N)+HOLD(I,K1)*DBLE(DXI(J))
290  CONTINUE
280  CONTINUE
      END IF
      RETURN
      END

C
C -----
C
C SUBROUTINE SLNPD(A,B,D,NX,NU,IMP)
C *****
C
C SOLUTION OF LINEAR SYSTEMS OF EQUATIONS
C BY THE GAUSS ELIMINATION METHOD PROVIDING
C FOR INTERCHANGING ROWS WHEN ENCOUNTERING
C A ZERO DIAGONAL COEFFICIENT
C
C A :SYSTEM MATRIX
C B :ORIGINALLY IT CONTAINS THE INDEPENDENT
C COEFFICIENTS. AFTER SOLUTION IT CONTAINS THE
C VALUES OF THE SYSTEM UNKNOWNNS
C
C NX : ACTUAL NUMBER OF UNKNOWNNS
C
C NU : ROW AND COLUMN DIMENSION OF A
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C DOUBLE PRECISION A(NU,NU),B(NU)
C N1=NX-1
C DO 100 K=1,N1
C K1=K+1
C C=A(K,K)
C IF(DABS(C)-.000001)1,1,3
1 DO 7 J=K1,NX
C
C TRY TO INTERCHANGE ROWS TO GET NON ZERO
C DIAGONAL COEFFICIENT
C
C IF (DABS(A(J,K))-0.000001)7,7,5
5 DO 6 LL=K,NX
C=A(K,LL)
A(K,LL)=A(J,LL)
6 A(J,LL)=C
C=B(K)
B(K)=B(J)
B(J)=C
C=A(K,K)
GO TO 3
7 CONTINUE
8 WRITE(IMP,2)K
D=0.0
GO TO 300
C
C DIVIDE ROW BY DIAGONAL COEFFICIENT

```

```

C
3 C=A(K,K)
DO 4 J=K1,NX
4 A(K,J)=A(K,J)/C
  B(K)=B(K)/C
C
C   ELIMINATE UNKNOWN X(K) FROM ROW I
C
DO 10 I=K1,NX
  C=A(I,K)
DO 9 J=K1,NX
  9 A(I,J)=A(I,J)-C*A(K,J)
  10 B(I)=B(I)-C*B(K)
100 CONTINUE
C
C   COMPUTE LAST UNKNOWN
C
IF (DABS(A(NX,NX))-0.000001)8,8,101
101 B(NX)=B(NX)/A(NX,NX)
C
C   APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING
C   UNKNOWNNS
C
DO 200 LL=1,N1
  K=NX-LL
  K1=K+1
DO 200 J=K1,NX
200 B(K)=B(K)-A(K,J)*B(J)
C
C   COMPUTE VALUE OF DETERMINATE
C
D=1.0
DO 250 I=1,NX
250 D=D*A(I,I)
WRITE(IMP,333)D
333 FORMAT('THE DETERMINANT =',E14.8)
2 FORMAT('***** SINGULARITY IN ROW *****',I5)
300 RETURN
END
C
C -----
C
SUBROUTINE JACOBI(AS,BM,EIGV,X,ME,ND)
C *****
C
REAL EIGV(ND),X(ND,ND), BM(ND,ND)
REAL AS(ND,ND),D(150)
C
C   AS=A : STIFNESS MATRIX
C   BM=F : MASS MATRIX
C   NT=ME : TOTAL DOF
C   X : FINAL EIGENVECTOR
C   EIGV: FINAL EIGENVALUE
C   IFPR=0 :FOR PRINTOUT OF INTERM. RESULTS
C   IFPR=1 :FOR PRINTOUT OF INTERM. RESULTS
C
C   CONSTANSTS
C
NSMAX = 15
RTOL = 10.**(-12)
IFPR = 0

```

```

C
C
C      NT= ME
C
C      INITIALIZE  EIGENVALUE AND EIGENVECTOR MATRICES
C
DO 10 I = 1,NT
IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 9
WRITE (*,2020)
STOP
9      D(I) = AS(I,I)/BM(I,I)
      EIGV(I) = D(I)
10     CONTINUE
      DO 30 I = 1,NT
      DO 20 J = 1,NT
20      X(I,J) = 0.
30      X(I,I) = 1.
      IF (NT.EQ. 1) THEN
C        WRITE (*,*)' EIGENVALUE : ',EIGV(1)
        RETURN
      ELSE
      ENDIF

C      INITIALIZE SWEEP COUNTER AND EIGEN ITERATION
C
      NSWEEP = 0
      NR = NT - 1
40      NSWEEP = NSWEEP + 1
      IF (IFPR .EQ. 1) WRITE (*, 2000) NSWEEP

C
C      CHECK IF PRESENT OFF-DIAGONAL ELEMENTS ARE LARGE ENOUGH TO
C      REQUIRE ZEROING
C
      EPS = (.01**NSWEEP)**2
      DO 210 J = 1,NR
      JJ = J + 1
      DO 210 K = JJ,NT
      EPTOLA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))
      EPTOLB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
      IF((EPTOLA .LT.EPS) .AND. (EPTOLB .LT. EPS)) GOTO 210

C
C      IF ZEROING IS REQUIRED CALCULATE THE ROT MAT ELM CA ,CG
C
      AKK = AS(K,K) * BM(J,K) - BM(K,K) * AS(J,K)
      AJJ = AS(J,J) * BM(J,K) - BM(J,J) * AS(J,K)
      AB = AS(J,J) * BM(K,K) - AS(K,K) * BM(J,J)
      CHECK = (AB*AB + 4.* AKK * AJJ)/4.
      IF (CHECK .LT. 0.) THEN
        WRITE (*,2020)
        STOP
      ELSE
      ENDIF
60      SQCH = SQRT(CHECK)
      D1 = AB/2. + SQCH
      D2 = AB/2. - SQCH
      DEN = D1
      IF (ABS(D2) .GT. ABS(D1)) THEN
      DEN = D2
      ELSE
      ENDIF

```



```

      IF (DEN .EQ. 0.) THEN
70    CA = 0.
      CG = -AS(J,K)/AS(K,K)
      ELSE
80    CA = (AKK)/(DEN)
      CG = - (AJJ)/(DEN)
      ENDIF
C
C    PERFORM GENERALIZED ROT TO ZERO THE PRESENT OFF  DG. ELM
C
90    IF (NT .EQ. 2) GOTO 190
100   JP1 = J + 1
      JM1 = J - 1
      KP1 = K + 1
      KM1 = K - 1
      IF (JM1 .LT. 1) GOTO 130
110   DO 120 I = 1, JM1
      AJ = AS(I,J)
      BJ = BM(I,J)
      AK = AS(I,K)
      BK = BM(I,K)
      AS(I,J) = AJ + CG * AK
      BM(I,J) = BJ + CG * BK
      AS(I,K) = AK + CA * AJ
120   BM(I,K) = BK + CA * BJ
130   IF (KP1 .GT. NT) GOTO 160
140   DO 150 I = KP1, NT
      AJ = AS(J,I)
      BJ = BM(J,I)
      AK = AS(K,I)
      BK = BM(K,I)
      AS(J,I) = AJ + CG * AK
      BM(J,I) = BJ + CG * BK
      AS(K,I) = AK + CA * AJ
150   BM(K,I) = BK + CA * BJ
160   IF (JP1 .GT. KM1) GOTO 190
170   DO 180 I = JP1, KM1
      AJ = AS(J,I)
      BJ = BM(J,I)
      AK = AS(I,K)
      BK = BM(I,K)
      AS(J,I) = AJ + CG * AK
      BM(J,I) = BJ + CG * BK
      AS(I,K) = AK + CA * AJ
180   BM(I,K) = BK + CA * BJ
190   AK = AS(K,K)
      BK = BM(K,K)
      AS(K,K) = AK + 2.* CA * AS(J,K) + CA * CA * AS(J,J)
      BM(K,K) = BK + 2.* CA * BM(J,K) + CA * CA * BM(J,J)
      AS(J,J) = AS(J,J) + 2.* CG * AS(J,K) + CG * CG * AK
      BM(J,J) = BM(J,J) + 2.* CG * BM(J,K) + CG * CG * BK
      AS(J,K) = 0.
      BM(J,K) = 0.
C
C    UPDATE EIGENVALUE MATRIX AFTER EACH ROTATION
C
      DO 200 I = 1, NT
      XJ = X(I,J)
      XK = X(I,K)
      X(I,J) = XJ + CG * XK

```

```

      X(I,K) = XK + CA * XJ
200  CONTINUE
210  CONTINUE
C
C  UPDATE EIGENVALUE AFTER EACH SWEEP
C
      DO 220 I = 1,NT
      IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 220
      WRITE (*,2020)
      STOP
220  EIGV(I) = AS(I,I)/BM(I,I)
      IF (IFPR .EQ. 0) GOTO 230
      WRITE (*,2030)
      WRITE (*,710) (EIGV(I), I = 1,NT)
C
C      CHECK FOR CONVERGENCE
C
230  DO 240 I = 1,NT
      TOL = RTOL * D(I)
      DIF = ABS(EIGV(I) - D(I))
      IF (DIF .GT. TOL) GOTO 280
240  CONTINUE
C
C      CHECK ALL OFF DIAG, TO SEE IF ANOTHER SWEEP IS REQD.
C
      EPS = RTOL ** 2
      DO 250 J = 1,NR
      JJ = J + 1
      DO 253 K = JJ,NT
      EPSA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))
      EPSB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
      IF ((EPSA .LT. EPS) .AND. (EPSB .LT. EPS)) GOTO 253
      GOTO 280
253  CONTINUE
250  CONTINUE
C
C      FILL BOTTOM TRIANG OF RESLT MATPICES AND SCALE EG. VECT.
C
255  DO 260 I = 1,NT
      DO 260 J = 1,NT
      AS(J,I) = AS(I,J)
260  BM(J,I) = BM(I,J)
      DO 270 J = 1,NT
      BB = SQRT(BM(J,J))
      DO 270 K = 1,NT
270  X(K,J) = X(K,J)/BB
C
C      SORT EIGENVALUE AND VECTOR
C
510  KOUNT = 0
      DO 520 I = 1,NT-1
      IF (EIGV(I) .GT. EIGV(I+1)) THEN
      KOUNT = 1
      CHNG = EIGV(I)
      EIGV(I) = EIGV(I+1)
      EIGV(I+1) = CHNG
      DO 530 J = 1,NT
      CHNG = X(J,I)
      X(J,I) = X(J,I+1)
      X(J,I+1) = CHNG

```

```

530  CONTINUE
      ENDIF
520  CONTINUE
      IF (KOUNT .EQ. 1) GOTO 510
      IF (IIPR .EQ. 0) GOTO 276
      WRITE (*,960)
      DO 275 K = 1,NT
      WRITE (*,710) (X(K,J), J = 1,NT)
275  CONTINUE
276  RETURN

C
C      UPDATE D MATRIX AND START NEW SWEEP, IF ALLOWED
C
280  DO 290 I = 1,NT
290  D(I) = EIGV(I)
      IF (NSWEEP .LT. NSMAX) GOTO 40
      GOTO 255
710  FORMAT (10(E12.5,1X))

910  FORMAT (//, ' ** STFNESSS MATRIX **',/)
920  FORMAT (9(G10.4,1X)/
           5X,9(G10.4,1X))
930  FORMAT (//, ' ** MASS MATRIX **',/)
960  FORMAT (//, ' FINAL EIGEN VEC OR OF MODAL SYSTEM',/)
2000 FORMAT (//, ' SWEEP CYCLE IN JACOBI ' ,I4,/)
2020 FORMAT (10X'ERROR SOLUTION STOP', /
           5X,' MATRICES NOT POSITIVE DEFINITE')
2030 FORMAT (' CURRENT EIGENVALUE IN *JACOBI* ARE',/)
300  RETURN
      END

C
C      -----
C
      SUBROUTINE OUTPUT(X,Y,FID,FIB,DFI,AFR,AFI,AKR,AKI,
* NO,N1,N2,NOUT,FLAG,DF)
C      *****
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION X(1),Y(1),FID(1),FIB(1),DFI(1)
      REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
      REAL UL(20,20),UB(20,20),VL(20,20),VB(20,20)
      DOUBLE PRECISION PRA(20),PIA(20),QRA(20),QIA(20)
      COMPLEX XKAP(20),PSI(20,20),TEMP1(20,20),TEMP2(20,20)
      COMPLEX SUM3,DX(20)
      REAL TEMP1R(20,20),TEMP2R(20,20)
      REAL DXR(20),DXI(20),SUM3R
      LOGICAL FLAG

C
C      RES=0.
      WRITE(IMP,500)

C
C      WRITE RESULTS FOR RESERVOIR SURFACE
C
      IF(NOUT.EQ.1)THEN
      WRITE(IMP,600)
      WRITE(IMP,330)
      WRITE(IMP,600)

```

```

WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 10 I=1,N0
WRITE(IMP,200)X(I),Y(I),RES,DFI(I)
10 CONTINUE
IF(FLAG)THEN
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
DO 20 I=1,N0
WRITE(IMP,200)X(I),Y(I),RES,DFI(I+N)
20 CONTINUE
END IF
END IF

C
C WRITE RESULTS FOR DAM FACE
C
WRITE(IMP,600)
WRITE(IMP,340)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
WRITE(IMP,200)X(N0),Y(N0),RES,FID(1)
DO 30 I=N0+1,N1
WRITE(IMP,200)X(I),Y(I),DFI(I),FID(I-N0+1)
30 CONTINUE
IF(FLAG)THEN
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
WRITE(IMP,200)X(N0),Y(N0),RES,RES
DO 40 I=N0+1,N1
WRITE(IMP,200)X(I),Y(I),DFI(N+I),RES
40 CONTINUE
END IF

C
C WRITE RESULTS FOR RESERVOIR BOTTOM
C
IF(NOUT.EQ.1)THEN
WRITE(IMP,600)
WRITE(IMP,350)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 50 I=N1,N2
QR=FIB(I-N1+1)
IF(CONST2.GT.1.0E-08)THEN
QR=QR+CONST1*CONST2*DFI(I+N)
END IF
WRITE(IMP,200)X(I),Y(I),DFI(I),QR
50 CONTINUE
IF(FLAG)THEN
WRITE(IMP,600)
WRITE(IMP,320)

```

```

WRITE(IMP,600)
DO 60 I=N1,N2
  QI=0
  IF(CONST2.GT.1.0E-08) THEN
    QI=QI-CONST1*CONST2*DFI(I)
  END IF
  WRITE(IMP,200)X(I),Y(I),DFI(I+N),QI
60 CONTINUE
  END IF

C
C   WRITE RESULTS FOR THE FAR END
C
  IF(ME.GT.0) THEN
    IF(CONST2.GT.1.0E-08) THEN
      DO 80 J=1,ME
        DO 70 I=1,ME
          UL(I,J)=AFR(I,J)*AKR(J)
          VB(I,J)=AFI(I,J)*AKI(J)
          VL(I,J)=AFI(I,J)*AKR(J)
          UB(I,J)=AFR(I,J)*AKI(J)
70 CONTINUE
80 CONTINUE
      DO 105 I=1,ME
        DO 90 J=1,ME
          UL(I,J)=UL(I,J)-VB(I,J)
          VL(I,J)=VL(I,J)+UB(I,J)
90 CONTINUE
105 CONTINUE
      ELSE IF(FLAG.AND.CONST2.LE.1.0E-08) THEN
        DO 120 J=1,ME
          DO 110 I=1,ME
            UL(I,J)=AFR(I,J)*AKR(J)
            VL(I,J)=AFR(I,J)*AKI(J)
110 CONTINUE
120 CONTINUE
          END IF
          IF(FLAG) THEN
            DO 125 I=1,ME
              SUM1=0
              SUM2=0
              DO 122 J=1,ME
                SUM1=SUM1+UL(I,J)*DFI(N2+J)-VL(I,J)*DFI(N2+J+N)
                SUM2=SUM2+VL(I,J)*DFI(N2+J)+UL(I,J)*DFI(N2+J+N)
122 CONTINUE
              QRA(I)=-SUM1
              QIA(I)=-SUM2
125 CONTINUE
            END IF

C
C
          IF(CONST2.GT.1.0E-08) THEN
            IF(DF.GT.0.0) THEN
              DO 111 I=1,ME
                XKAP(I)=CMPLX(AKR(I),AKI(I))
              DO 112 J=1,ME
                PSI(I,J)=CMPLX(AFR(I,J),AFI(I,J))
112 CONTINUE
111 CONTINUE
              DO 113 I=1,ME
                DO 114 J=1,ME

```

```

TEMP1(I,J)=PSI(I,J)/(XKAP(J)*XKAP(J))
114 CONTINUE
113 CONTINUE
DO 115 I=1,ME
DO 116 J=1,ME
SUM3=(0.0,0.0)
DO 117 K=1,ME
SUM3=SUM3+TEMP1(J,K)*PSI(I,K)
117 CONTINUE
TEMP2(J,I)=SUM3
116 CONTINUE
115 CONTINUE
DO 118 I=1,ME
DX(I)=TEMP2(I,1)*SNGL(DF)
118 CONTINUE
DO 119 I=1,ME
DXR(I)=REAL(DX(I))
DXI(I)=AIMAG(DX(I))
119 CONTINUE
END IF
DO 140 I=1,ME
SUM1=0
SUM2=0
DO 130 J=1,ME
SUM1=SUM1+AFR(I,J)*DFI(N2+J)-AFI(I,J)*DFI(N2+J+N)
SUM2=SUM2+AFI(I,J)*DFI(N2+J)+AFR(I,J)*DFI(N2+J+N)
130 CONTINUE
IF(DF.GT.0.0) THEN
PRA(I)=SUM1+DXR(I)
PIA(I)=SUM2+DXI(I)
ELSE
PRA(I)=SUM1
PIA(I)=SUM2
END IF
140 CONTINUE
ELSE IF(CONST2.LE.1.0E-08.AND.FLAG) THEN
IF(DF.GT.0.0) THEN
DO 141 I=1,ME
DO 142 J=1,ME
IF(AKI(J).GT.0.0) THEN
TEMP1R(I,J)=-AFR(I,J)/(AKI(J)*AKI(J))
ELSE
TEMP1R(I,J)=AFR(I,J)/(AKR(J)*AKR(J))
END IF
142 CONTINUE
141 CONTINUE
DO 145 I=1,ME
DO 146 J=1,ME
SUM3R=0.0
DO 147 K=1,ME
SUM3R=SUM3R+TEMP1R(J,K)*AFR(I,K)
147 CONTINUE
TEMP2R(J,I)=SUM3R
146 CONTINUE
145 CONTINUE
DO 148 I=1,ME
DXR(I)=TEMP2R(I,1)*SNGL(DF)
148 CONTINUE
END IF
DO 160 I=1,ME

```

```

SUM1=0
SUM2=0
DO 150 J=1,ME
SUM1=SUM1+AFR(I,J)*DFI(N2+J)
SUM2=SUM2+AFR(I,J)*DFI(N2+J+N)
150 CONTINUE
IF(DF.GT.0.0) THEN
  IF(FLAG) THEN
    PRA(I)=SUM1+DXR(I)
    PIA(I)=SUM2
  ELSE
    PRA(I)=SUM1+DXR(I)
    PIA(I)=SUM2+DXI(I)
  END IF
ELSE
  PRA(I)=SUM1
  PIA(I)=SUM2
END IF
160 CONTINUE
END IF
C
C
IF(.NOT.FLAG) THEN
IF(DF.GT.0.0) THEN
DO 311 I=1,ME
DO 312 J=1,ME
TEMP1R(I,J)=AFR(I,J)/(AKR(J)*AKR(J))
312 CONTINUE
311 CONTINUE
DO 313 I=1,ME
DO 314 J=1,MME
SUM3R=0.0
DO 315 K=1,ME
SUM3R=SUM3R+TEMP1R(J,K)*AFR(I,J)
315 CONTINUE
TEMP2R(J,I)=SUM3R
314 CONTINUE
313 CONTINUE
DO 316 I=1,ME
DXR(I)=TEMP2R(I,1)*SNGL(DF)
316 CONTINUE
END IF
DO 180 I=1,ME
SUM1=0.
DO 170 J=1,ME
SUM1=SUM1+AFR(I,J)*DFI(N2+J)
170 CONTINUE
IF(DF.GT.0.0) THEN
PRA(I)=SUM1+DXR(I)
ELSE
PRA(I)=SUM1
END IF
180 CONTINUE
C
C
DO 205 J=1,ME
DO 190 I=1,ME
UL(I,J)=AFR(I,J)*AKR(J)
190 CONTINUE
205 CONTINUE

```

```

DO 220 I=1,ME
SUM1=0
DO 210 J=1,ME
SUM1=SUM1+UL(I,J)*DFI(N2+J)
210 CONTINUE
QRA(I)--SUM1
220 CONTINUE
END IF
C
C
WRITE(IMP,600)
WRITE(IMP,360)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 230 I=1,ME
K1=N2+I
WRITE(IMP,200)X(K1),Y(K1),PRA(I),QRA(I)
230 CONTINUE
IF (FLAG) THEN
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
DO 240 I=1,ME
K1=N2+I
WRITE(IMP,200)X(K1),Y(K1),PIA(I),QIA(I)
240 CONTINUE
END IF
END IF
END IF
C
C
C
COMPUTE TOTAL PRESSURE ON DAM
NE=N1-N0
DY=HY/NE
TPR=DFI(N1)
DO 250 I=N0+1,N1-1
TPR=TPR+2*DFI(I)
250 CONTINUE
TPR=DY*TPR/2.
WRITE(IMP,600)
WRITE(IMP,370)TPR
C
C
IF (FLAG) THEN
TPI=DFI(N1+N)
DO 260 I=N0+1,N1-1
TPI=TPI+2*DFI(I+N)
260 CONTINUE
TPI=TPI*DY/2.
WRITE(IMP,600)
WRITE(IMP,380)TPI
TPR=DSQRT(TPR**2+TPI**2)
WRITE(IMP,600)
WRITE(IMP,390)TPR
WRITE(IMP,600)
END IF
C

```



```

C
320 FORMAT(1X,'IMAGINARY PART')
200 FORMAT(4(4X,E14.5))
100 FORMAT(1X,'BOUNDARY  NODES',//
*,10X,'X',17X,'Y',15X,'PRESSURE',5X,'PRESSURE DERIVATIVE')
300 FORMAT(1X,'REAL PART',)
330 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE'
*)
340 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE')
350 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM')
360 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON FAR END')
370 FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, REAL PART',E14.5)
380 FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, IMAGINARY PART',E14.5)
390 FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, ABSOLUTE VALUE',E14.5)
500 FORMAT(1X,70(' '))
600 FORMAT(1X,' ')
    RETURN
    END

C
C -----
C
SUBROUTINE ADDUND(H,G,HOLD,AFR,AKR,AKI,N0,N1,N2,DFI,FID,FIB,
*FLAG,DF)
C *****
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
    COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
    REAL AFR(ND,ND),AKR(ND),AKI(ND)
    DOUBLE PRECISION H(NU,NU),G(NU,NU),T(100,20)
    DOUBLE PRECISION FID(NU),FIB(NU),DFI(NU),HOLD(NU,NU)
    REAL U(20,20),V(20,20)
    REAL TEMP1(20,20),TEMP2(20,20)
    REAL SUM3,DXR(20)
    LOGICAL FLAG

C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C
    IF (ME.GT.0) THEN
        DO 20 J=1,ME
            DO 10 I=1,ME
                U(I,J)=AFR(I,J)*AKR(J)
10          CONTINUE
20          CONTINUE
            IF (FLAG) THEN
                DO 40 J=1,ME
                    DO 30 I=1,ME
                        V(I,J)=AFR(I,J)*AKI(J)
30          CONTINUE
40          CONTINUE
            END IF

C
C
        DO 70 I=1,N
            DO 60 J=1,ME
                SUM1=0

```

```

        SUM2=0
        DO 50 K=1,ME
        KK=K+N2
        SUM1=SUM1+H(I, KK) * DBLE(AFR(K, J))
        SUM2=SUM2+G(I, KK) *DBLE(U(K, J))
50    CONTINUE
        T(I, J)=SUM1+SUM2
60    CONTINUE
70    CONTINUE
C
        DO 75 I=1,N
        DO 72 J=1,ME
        JJ=J+N2
        H(I, JJ)=T(I, J)
72    CONTINUE
75    CONTINUE
C
C IMAGINARY PART
C
        IF (FLAG) THEN
        DO 120 I=1,N
        DO 110 J=1,ME
        SUM1=0
        DO 100 K=1,ME
        KK=K+N2
        SUM1=SUM1+G(I, KK) *DBLE(V(K, J))
100    CONTINUE
        T(I, J)=SUM1
110    CONTINUE
120    CONTINUE
C
        DO 140 I=1,N
        DO 130 J=1,ME
        H(N+I, J+N2)=T(I, J)
130    CONTINUE
140    CONTINUE
C
C FILL IN THE REST OF H MATRIX
C
C
        DO 160 I=1,N
        DO 150 J=1,N
        H(I, J+N)= -H(I+N, J)
        H(I+N, J+N)=H(I, J)
150    CONTINUE
160    CONTINUE
        END IF
        END IF
C
C FORM RIGHT HAND VECTOR
C
        DO 180 I=1,N
        DO 170 J=N0+1, N1-1
        DFI(I)=DFI(I)+G(I, J) *FID(J-N0+1)
170    CONTINUE
180    CONTINUE
C
        DO 200 I=1,N
        DO 190 J=N1, N2
        DFI(I)=DFI(I)+G(I, J) *FIB(J-N1+1)

```

```

190 CONTINUE
200 CONTINUE
C
  IF (DF.GT.0.0) THEN
    DO 210 I=1,ME
      DO 220 J=1,ME
        IF (AKI(J).GT.0.0) THEN
          TEMP1(I,J)=-AFP(I,J)/(AKI(J)*AKI(J))
        ELSE
          TEMP1(I,J)=AFP(I,J)/(AKR(J)*AKR(J))
        END IF
      220 CONTINUE
    210 CONTINUE
      DO 230 I=1,ME
        DO 240 J=1,ME
          SUM3=0.0
          DO 250 K=1,ME
            SUM3=SUM3+TEMP1(J,K)*AFR(I,K)
          250 CONTINUE
          TEMP2(J,I)=SUM3
        240 CONTINUE
      230 CONTINUE
        DO 260 I=1,ME
          DXR(I)=-TEMP2(I,1)*SNGL(DF)
        260 CONTINUE
          DO 290 I=1,N
            DO 300 J=1,ME
              K1=J-ME+N
              DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))
            300 CONTINUE
          290 CONTINUE
            END IF
            RETURN
            END
C
C*****THIS IS THE LAST CARD OF THE LINEAR BEM PROGRAM*****

```

1.4 Constant Element Program for 3D Reservoirs - BEMC.3D

The constant element program for 3D reservoirs is computer implementation of the constant boundary element formulation as presented in Chapters 5 and 6 of Volume 1.

1.4.1 Description of the BEMC.3D Program

The BEMC.3D program can be used to calculate the hydrodynamic pressure on a rigid arch or straight dam storing a three-dimensional finite or infinite water reservoir and subjected to three possible types of harmonic motion: upstream-downstream, cross stream or vertical motion. The program uses the 4-node quadrilateral constant boundary elements. The pressures and the pressure derivatives are assumed to have constant values associated with the mid-nodes of the elements. There are no limitations on the shape of the dam or the finite reservoir. The infinite reservoir has to be divided to two parts: a finite irregular and an infinite uniform channel of constant cross section. The infinite channel is modelled with use of the 4-node quadrilateral linear finite elements. A numerical example follows the short descriptions of input and output data files.

The BEMC.3D program is also based on the formerly presented two-dimensional computer program - BEMC.2D. The program architecture is similar and number of subroutines are the same. The description of them will not be repeated here.

The main program defines the limiting number of constant boundary elements that it can handle. The number is set to 340. The main program calls the following subroutines:

INPUT - to read boundary elements data and print the first part of the output.

EIGEN - to set and calculate the complex eigenvalue problem in case of infinite radiation and damping included.

EIGENR - to set and calculate the real eigenvalue problem in case of infinite radiation only.

FMAT - to form matrices **G** and **H** and rearrange them according to the boundary conditions on the top of the reservoir.

ADDFBC - to form the system of equations for the damped case and apply the boundary conditions on the bottom of the reservoir and at its far end.

5

MicroD

ADDUND - to form the system of equations required for undamped case.

SLNPD - to reorder the vector of computed unknowns and print out the rest of the output file.

The subroutine **EIGEN** calls the subroutine:

FSFEM - to compute the real and imaginary parts of the eigenvector matrix and the real and imaginary parts of the eigenvector.

The subroutine **FSFEM** besides subroutines, which constitute the QZ algorithm, calls the following subroutines:

SETE - to form the complex matrices for the complex eigenvalue problem, which governs the damped vibrations in the infinite channel: $(A + i\omega\gamma B) p = \lambda^2 F p$. This subroutine uses the 2D FEM formulation to calculate the terms of these matrices (5).

NORMA - to normalize the matrix of eigenvectors w.r.t. matrix **F**.

The subroutine **SETE** calls the following subroutines (5):

INPUT2 - to read data for the far boundary finite element mesh.

GAUSSQ - to read positions of Gaussian points and weights associated to each Gaussian point.

SF2 - to evaluate shape functions and their derivatives in the local coordinate system.

JACOB2 - to calculate the coordinates of sampling points and create Jacobian matrix. This subroutine also calculates the determinant and the inverse of the Jacobian matrix and cartesian derivatives of the shape functions.

BMAT - to evaluate matrix B_1 and B_2 , which are required for numerical integration.

EXTRA - to apply boundary conditions on the upper edge of the far end to global matrices required to set the complex eigenvalue problem and calculated inside the subroutine **SETE**.

The subroutine **EIGENR** calls the subroutines:

SETER - to form the real matrices for the real eigenvalue problem, which governs the

undamped vibration in the infinite channel. This subroutine is a modified version of the subroutine SETE.

JACOBI - to solve the generalized real matrix eigenvalue problem (as described earlier in the description of the BEMC.2D program).

The subroutine SETER calls the following subroutines:

INPUT2R - to read data for the far boundary finite element mesh. This subroutine is a modified version of the subroutine INPUT2.

GAUSSQ - the same subroutine as one called by the subroutine SETE.

SF2 - as above.

JACOB2 - as above.

BMAT - as above.

EXTRAR - to apply boundary conditions on the upper edge of the far end to global matrices required to set the real eigenvalue problem.

The subroutine FMAT calls the subroutines:

INTE - to compute the G and H matrix off diagonal elements by means of the numerical integration along the boundary elements. The numerical integration uses 4×4 Gauss quadrature scheme.

INLO - to compute the diagonal elements of matrix G.

The subroutine INLO calls four times the subroutine INTL and sum up the values calculated.

INTL - this subroutine computes the quantities required to calculate the diagonal elements of matrix G. It uses the collapsed rectangular elements to eliminate the singularity. The numerical integration uses 4×4 Gauss quadrature scheme (6).

1.4.2 Input Data.

The following two tables show the order of presentation of data. Two input files must be created for the program BEMC.3D.FTN.

TABLE 1-3

Input Data - 1 for the BEMC.3D Program

Card Group	No. of Cards	Format	FORTTRAN variables
A	1	18A4	TITLE
B1	1	General Format	N = number of boundary elements NC = number of corner nodes N0 = last elem. number at top near the dam N1 = last elem. number at base of the dam N2 = last elem. number at bottom near far end NOUT = 1, flag for detailed output
B2	1	General Format	HY = height of the dam C = velocity of sound in water ALFA = wave reflection coefficient
C	NC	General Format	X, Y = coordinates of corner nodes
D	N	General Format	K = connectivity matrix for each element
E	N1-N0	General Format	FID = boundary conditions on the dam face
F	N2-N1	General Format	FIB = boundary conditions on the foundation base
G	No of cases	General Format	DF = boundary conditions for vertical excitation at the far end
H	No of cases	General Format	OMEGA = excitation frequency

TABLE 1-4

Input Data - 2 for the BEMC.3D Program

Card Group	No. of Cards	Format	FORTTRAN variables
A1	1	General Format	NT=number of top nodes
A2	1	General Format	NELEM = number of finite elements at far boundary NNODE = 4, number of nodes in an element NDIME = 2, number of coordinates required to define each node NGAUS = 2, the order of Gaussian quadrature scheme NGASP = 4, number of Gaussian integration points
B	N	General Format	COORD = cartesian global coordinates of each node in the system associated with far end
C	NELEM	General Format	LNODS = element topology matrix
D	NELEM	General Format	EINDEX = element index matrix describes element type w.r.t. location on the boundary of the infinite channel (req. for damped case only)
E	NT	General	NTOP = matrix containing numbers of top nodes

The element number order can be clearly seen from an example, which is presented after this subsection.

The following table presents the types of the elements on the far boundary, which are required for creation of the vector EINDEX(IELEM) (IELEM = current element number). Values of vector EINDEX depends on the location of a finite element w.r.t. the infinite channel cross section boundaries.

TABLE 1-5

Values of Vector EINDEX

Element Scheme	EINDEX(IELEM) = i.e. Element Type
0
1
2
3
4
5
6
7
8

1.4.3 Output

The program prints out the input data as well as the hydrodynamic pressure and pressure gradient values on the dam face, reservoir's bottom and banks, and the far end. Intermediate results, e.g. for an eigenvalue problem, are possible to be obtained.

1.4.4 Example

For the purpose of illustration, a simple example is presented here (see Fig. 1.3).

Fig. 1.3 The Constant Boundary Element Model of Infinite 3D Reservoir - $N = 29$

A three-dimensional infinite reservoir is solved with use of 29 constant boundary elements. The far end is modelled by four 4-node quadrilateral linear isoparametric finite elements (see Fig. 1.4). The system is subjected to an upstream-downstream harmonic motion with frequency equal to $45.24 \frac{\text{rad}}{\text{s}}$ ($\omega/\omega_1 = 1.2$). The bottom and banks of the reservoir are flexible with reflection coefficient $\alpha_R = 0.5$ (the half of the waves is reflected).

First, the two independent input files are shown and are followed by the output file.

• **Input file - 1 TEST3DC**

```

3D INFINITE RESERVOIR
29,37,7,11,29,1
60.,1440,0.50
  0.0, 60.0,  0.0
  0.0, 60.0, 30.0
  0.0, 60.0, 60.0
  0.0, 30.0,  0.0
  0.0, 30.0, 30.0
  0.0, 30.0, 60.0
  0.0,  0.0,  0.0
  0.0,  0.0, 30.0
  0.0,  0.0, 60.0
30.0, 60.0,  0.0
30.0, 60.0, 30.0
30.0, 60.0, 60.0
30.0, 30.0,  0.0
30.0, 30.0, 60.0
30.0,  0.0,  0.0
30.0,  0.0, 30.0
30.0,  0.0, 60.0
60.0, 60.0,  0.0
60.0, 60.0, 30.0
60.0, 60.0, 60.0
60.0, 30.0,  0.0
60.0, 30.0, 60.0
60.0,  0.0,  0.0
60.0,  0.0, 30.0
60.0,  0.0, 60.0
60.0, 60.0, 15.0
60.0, 60.0, 45.0
60.0, 45.0,  0.0
60.0, 45.0, 15.0
60.0, 45.0, 45.0
60.0, 45.0, 60.0
60.0, 15.0,  0.0
60.0, 15.0, 15.0
60.0, 15.0, 45.0
60.0, 15.0, 60.0
60.0,  0.0, 15.0
60.0,  0.0, 45.0
27,20,31,30
30,31,35,34
34,35,25,37
14,22,20,12
 6,14,12,3
17,25,22,14
 9,17,14,6

```


• Output file OUT3DC

DATA

NUMBER OF BOUNDARY ELEMENTS 29

NUMBER OF CORNER NODES 37

LAST ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DAM 7

LAST ELEMENT NUMBER AT BASE OF DAM 11

LAST ELEMENT NUMBER AT BOTTOM NEAR FAR END 29

FLAG FOR DETAILED OUTPUT 1

HEIGHT OF DAM 0.60000E+02

VELOCITY OF SOUND IN WATER 0.14400E+04

WAVE REFLECTION COEFFICIENT 0.50000E+00

FOUNDATION DAMPING COEFFICIENT 0.23148E-03

COORDINATES OF CORNER NODES

POINT	X	Y	Z	
1	0.0000000E+00	0.6000000E+02	0.0000000E+00	
2	0.0000000E+00	0.6000000E+02	0.3000000E+02	
3	0.0000000E+00	0.6000000E+02	0.6000000E+02	
4	0.0000000E+00	0.3000000E+02	0.0000000E+00	
5	0.0000000E+00	0.3000000E+02	0.3000000E+02	
6	0.0000000E+00	0.3000000E+02	0.6000000E+02	
7	0.0000000E+00	0.0000000E+00	0.0000000E+00	
8	0.0000000E+00	0.0000000E+00	0.3000000E+02	
9	0.0000000E+00	0.0000000E+00	0.6000000E+02	
10	0.3000000E+02	0.6000000E+02	0.0000000E+00	
11	0.3000000E+02	0.6000000E+02	0.3000000E+02	
12	0.3000000E+02	0.6000000E+02	0.6000000E+02	
13	0.3000000E+02	0.3000000E+02	0.0000000E+00	
14	0.3000000E+02	0.3000000E+02	0.6000000E+02	
15	0.3000000E+02	0.0000000E+00	0.0000000E+00	
16	0.3000000E+02	0.0000000E+00	0.3000000E+02	
17	0.3000000E+02	0.0000000E+00	0.6000000E+02	
18	0.6000000E+02	0.6000000E+02	0.0000000E+00	
19	0.6000000E+02	0.6000000E+02	0.3000000E+02	
20	0.6000000E+02	0.6000000E+02	0.6000000E+02	
21	0.6000000E+02	0.3000000E+02	0.0000000E+00	
22	0.6000000E+02	0.3000000E+02	0.6000000E+02	
23	0.6000000E+02	0.0000000E+00	0.0000000E+00	
24	0.6000000E+02	0.0000000E+00	0.3000000E+02	
25	0.6000000E+02	0.0000000E+00	0.6000000E+02	
26	0.6000000E+02	0.6000000E+02	0.1500000E+02	
27	0.6000000E+02	0.6000000E+02	0.4500000E+02	
28	0.6000000E+02	0.4500000E+02	0.0000000E+00	
29	0.6000000E+02	0.4500000E+02	0.1500000E+02	
30	0.6000000E+02	0.4500000E+02	0.4500000E+02	
31	0.6000000E+02	0.4500000E+02	0.6000000E+02	

32	0.6000000E+02	0.1500000E+02	0.0000000E+00
33	0.6000000E+02	0.1500000E+02	0.1500000E+02
34	0.6000000E+02	0.1500000E+02	0.4500000E+02
35	0.6000000E+02	0.1500000E+02	0.6000000E+02
36	0.6000000E+02	0.0000000E+00	0.1500000E+02
37	0.6000000E+02	0.0000000E+00	0.4500000E+02

ELEMENT NO.	CONNECTIVITY			
1	27	20	31	30
2	30	31	35	34
3	34	35	25	37
4	14	22	20	12
5	6	14	12	3
6	17	25	22	14
7	9	17	14	6
8	5	6	3	2
9	4	5	2	1
10	8	9	6	5
11	7	8	5	4
12	2	3	12	11
13	1	2	11	10
14	1	10	13	4
15	4	13	15	7
16	7	15	16	8
17	8	16	17	9
18	11	12	20	19
19	10	11	19	18
20	10	18	21	13
21	13	21	23	15
22	15	23	24	16
23	16	24	25	17
24	32	33	36	23
25	33	34	37	36
26	28	29	33	32
27	29	30	34	33
28	18	26	29	28
29	26	27	30	29

BOUNDARY CONDITIONS ON THE DAM FACE

NODE	PRESCRIBED VALUE
1	0.1019370E+00
2	0.1019370E+00
3	0.1019370E+00
4	0.1019370E+00

BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM

NODE	PRESCRIBED VALUE
1	0.0000000E+00
2	0.0000000E+00
3	0.0000000E+00
4	0.0000000E+00

5	0.0000000E+00
6	0.0000000E+00
7	0.0000000E+00
8	0.0000000E+00
9	0.0000000E+00
10	0.0000000E+00
11	0.0000000E+00
12	0.0000000E+00
13	0.0000000E+00
14	0.0000000E+00
15	0.0000000E+00
16	0.0000000E+00
17	0.0000000E+00
18	0.0000000E+00

EXCITING FREQUENCY 0.45240E+02

WAVE CONSTANT 0.31417E-01

THE DETERMINANT =0.21978558E-01

PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE

BOUNDARY NODES

X	Y	Z	PRESSURE	PRESSURE DERIVATIVE
REAL PART				
0.60000E+02	0.52500E+02	0.52500E+02	0.00000E+00	0.34744E-01
0.60000E+02	0.30000E+02	0.52500E+02	0.00000E+00	0.20320E-01
0.60000E+02	0.75000E+01	0.52500E+02	0.00000E+00	0.34744E-01
0.45000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.17952E-01
0.15000E+02	0.45000E+02	0.60000E+02	0.00000E+00	-0.82560E-01
0.45000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.17952E-01
0.15000E+02	0.15000E+02	0.60000E+02	0.00000E+00	-0.82561E-01

IMAGINARY PART

0.60000E+02	0.52500E+02	0.52500E+02	0.00000E+00	0.56487E-01
0.60000E+02	0.30000E+02	0.52500E+02	0.00000E+00	0.42156E-01
0.60000E+02	0.75000E+01	0.52500E+02	0.00000E+00	0.56487E-01
0.45000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.55072E-01
0.15000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.10067E+00
0.45000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.55072E-01
0.15000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.10067E+00

PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE

BOUNDARY NODES

X	Y	Z	PRESSURE	PRESSURE DERIVATIVE
REAL PART				
0.00000E+00	0.45000E+02	0.45000E+02	0.18831E+01	0.10194E+00
0.00000E+00	0.45000E+02	0.15000E+02	0.19389E+01	0.10194E+00
0.00000E+00	0.15000E+02	0.45000E+02	0.18831E+01	0.10194E+00
0.00000E+00	0.15000E+02	0.15000E+02	0.19389E+01	0.10194E+00
IMAGINARY PART				
0.00000E+00	0.45000E+02	0.45000E+02	-0.12616E+01	0.00000E+00
0.00000E+00	0.45000E+02	0.15000E+02	-0.31098E+01	0.00000E+00
0.00000E+00	0.15000E+02	0.45000E+02	-0.12616E+01	0.00000E+00
0.00000E+00	0.15000E+02	0.15000E+02	-0.31098E+01	0.00000E+00

PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM

BOUNDARY NODES

X	Y	Z	PRESSURE	PRESSURE DERIVATIVE
REAL PART				
0.15000E+02	0.60000E+02	0.45000E+02	0.59655E+00	-0.13329E-01
0.15000E+02	0.60000E+02	0.15000E+02	0.38089E+00	-0.31772E-01
0.15000E+02	0.45000E+02	0.00000E+00	0.30952E+00	-0.33846E-01
0.15000E+02	0.15000E+02	0.00000E+00	0.30952E+00	-0.33846E-01
0.15000E+02	0.00000E+00	0.15000E+02	0.38089E+00	-0.31772E-01
0.15000E+02	0.00000E+00	0.45000E+02	0.59655E+00	-0.13329E-01
0.45000E+02	0.60000E+02	0.45000E+02	-0.40465E+00	-0.86263E-02
0.45000E+02	0.60000E+02	0.15000E+02	-0.13777E+01	-0.22009E-01
0.45000E+02	0.45000E+02	0.00000E+00	-0.15431E+01	-0.23783E-01
0.45000E+02	0.15000E+02	0.00000E+00	-0.15431E+01	-0.23783E-01
0.45000E+02	0.00000E+00	0.15000E+02	-0.13777E+01	-0.22009E-01
0.45000E+02	0.00000E+00	0.45000E+02	-0.40465E+00	-0.86263E-02
0.60000E+02	0.75000E+01	0.75000E+01	-0.17776E+01	-0.21497E-01
0.60000E+02	0.75000E+02	0.30000E+02	-0.10628E+01	-0.15103E-01
0.60000E+02	0.30000E+01	0.75000E+01	-0.17180E+01	-0.22022E-01
0.60000E+02	0.30000E+02	0.30000E+02	-0.99633E+00	-0.15998E-01
0.60000E+02	0.52500E+02	0.75000E+01	-0.17776E+01	-0.21497E-01
0.60000E+02	0.52500E+02	0.30000E+02	-0.10628E+01	-0.15103E-01

IMAGINARY PART

0.15000E+02	0.60000E+02	0.45000E+02	-0.12728E+01	-0.62472E-02
0.15000E+02	0.60000E+02	0.15000E+02	-0.30339E+01	-0.39887E-02
0.15000E+02	0.45000E+02	0.00000E+00	-0.32320E+01	-0.32413E-02
0.15000E+02	0.15000E+02	0.00000E+00	-0.32320E+01	-0.32414E-02
0.15000E+02	0.00000E+00	0.15000E+02	-0.30339E+01	-0.39888E-02
0.15000E+02	0.00000E+00	0.45000E+02	-0.12728E+01	-0.62472E-02
0.45000E+02	0.60000E+02	0.45000E+02	-0.82373E+00	0.42376E-02
0.45000E+02	0.60000E+02	0.15000E+02	-0.21017E+01	0.14428E-01
0.45000E+02	0.45000E+02	0.00000E+00	-0.22711E+01	0.16160E-01
0.45000E+02	0.15000E+02	0.00000E+00	-0.22711E+01	0.16160E-01
0.45000E+02	0.00000E+00	0.15000E+02	-0.21017E+01	0.14428E-01
0.45000E+02	0.00000E+00	0.45000E+02	-0.82373E+00	0.42376E-02
0.60000E+02	0.75000E+01	0.75000E+01	-0.20528E+01	0.18615E-01
0.60000E+02	0.75000E+02	0.30000E+02	-0.14422E+01	0.11130E-01
0.60000E+02	0.30000E+01	0.75000E+01	-0.21029E+01	0.17992E-01
0.60000E+02	0.30000E+02	0.30000E+02	-0.15277E+01	0.10434E-01
0.60000E+02	0.52500E+02	0.75000E+01	-0.20528E+01	0.18616E-01
0.60000E+02	0.52500E+02	0.30000E+02	-0.14422E+01	0.11130E-01

1.4.5 Listing of the BEMC.3D Program.

User: jablonski

Node: 102A2

Date: Sat Nov 19 22:41:52 1988

File: bemc.3d.ftn

Pages: 59

PROGRAM BEMC.3D.NEW.FTN

PROGRAM FOR THE SOLUTION OF THREE DIMENSIONAL RESERVOIR
PROBLEM WITH ONE INFINITE BOUNDARY AND DAMPING INCLUDED
BY CONSTANT BOUNDARY ELEMENT METHOD
SPECIAL B.C. AT FAR BOUNDARY THROUGH FEM, DAMPING INCLUDED

PROGRAMMER: ALEXANDER M. JABLONSKI

DATE: MAY 27, 1988
MODIFIED JULY 6, 1988

CENTER: CASCADE, CARLETON UNIVERSITY

NOTE: THIS PROGRAM IS BASED ON PROGRAM BEMC.3D.I.FTN
PROGRAM BEMC.3D.I.FTN USES REAL PART OF FUNDAMENTAL SOLUTION
PROGRAM BEMC.3D.NEW.FTN USES FULL FUNDAMENTAL SOLUTION
PROGRAM BEMC.3D.NEW1.FTN INCLUDES VERTICAL EXCITATION
PROGRAM BEMC.3D.NEW2.FTN INCLUDES CHANGES TO ACCOMODATE
FINITE RESERVOIR

N = NUMBER OF BOUNDARY ELEMENTS, READ THROUGH INPUT
NC= NUMBER OF CORNER NODES
K = CORNER NODES FOR EACH ELEMENT
X = ARRAY CONTAINING X COORDINATES OF THE NODES
Y = ARRAY CONTAINING Y COORDINATES OF THE NODES
Z = ARRAY CONTAINING Z COORDINATES OF THE NODES
XM = ARRAY CONTAINING X COORDINATES OF THE MIDPOINTS OF THE ELEMENTS
YM = ARRAY CONTAINING Y COORDINATES OF THE MIDPOINTS OF THE ELEMENTS
ZM = ARRAY CONTAINING Z COORDINATES OF THE MIDPOINTS OF THE ELEMENTS
G = MATRIX IN THE BOUNDARY ELEMENT EQUATION
H = MATRIX IN THE BOUNDARY ELEMENT EQUATION
FID= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS
ON THE DAM FACE
FIB= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS
ON THE BASE
DF= VECTOR OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS IN CASE OF VERTICAL
EXCITATION AT THE FAR BOUNDARY
DF1 = RIGHT HAND SIDE VECTOR IN BOUNDARY INTEGRAL EQUATION;
AFTER SOLUTION, IT CONTAINS THE VALUES AT THE BOUNDARY
THAT WERE PREVIOUSLY UNKNOWN
LEC = UNIT NO. CONTAINING INPUT
IMP = UNIT NO. ON WHICH OUTPUT IS WRITTEN
NX = 2*N = DIMENSION OF THE SYSTEM OF EQUATIONS
NU = DIMENSION OF H AND G MATRICES IN THE MAIN PROGRAM
ND = DIMENSION OF THE SPECIAL BOUNDARY CONDITION MATRIX AT
FAR BOUNDARY IN THE MAIN PROGRAM
D = VALUE OF THE DETERMINANT OF COEFFICIENT MATRIX
CONST = OMEGA/C
OMEGA = EXCITING FREQUENCY
C = VELOCITY OF SOUND IN WATER
CONST1=OMEGA
CONST2=FOUNDATION DAMPING COEFFICIENT GAMMA
AFR=REAL PART OF EIGENVECTOR SI
AFI=IMAGINARY PART OF EIGENVECTOR SI
AKR=REAL PART OF KAPPA MATRIX

```

C      AKI=IMAGINARY PART OF KAPPA MATRIX
C      N0=LAST ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DAM
C      N1=LAST ELEMENT NUMBER AT THE BASE OF THE DAM
C      N2=LAST ELEMENT NUMBER AT THE BASE NEAR FAR END
C      MF=NUMBER OF ELEMENTS ON BASE
C      ME=NUMBER OF ELEMENTS ON THE FAR BOUNDARY
C      HY=HEIGHT OF THE RESERVOIR
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND,NB,NC,NY
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C
C      DOUBLE PRECISION X(350),Y(350),Z(350),G(680,680),H(680,680),
&DFI(680),FID(80),FIB(160),HOLD(680,680),XM(340),YM(340),ZM(340),
&DF(80)
C      REAL AFI(80,80),AFR(80,80),AKR(80),AKI(80)
C      DIMENSION K(4,340)
C
C      LOGICAL FLAG
C      LEC=2
C      IMP=3
C      ND=80
C      NB=160
C      NU=340
C      NY=680
C      PI=3.141592653
C      OPEN (LEC,FORM='FORMATTED',STATUS='OLD',FILE='TEST3DC')
C      OPEN (IMP,FORM='FORMATTED',STATUS='NEW',FILE='OUT3DC')
C      OPEN (6,FORM='FORMATTED',STATUS='NEW',FILE='EIGEN3DC')
C
C      READ DATA FROM INPUT FILE
C
C      CALL INPUT(X,Y,Z,K,FID,FIB,N0,N1,N2,NOUT,C,DF)
C      NX=2*N
C      5 WRITE(IMP,300)
C      READ(LEC,*,END=15) OMEGA
C      CONST=OMEGA/C
C      WRITE(IMP,350) OMEGA,CONST
C      CONST1=OMEGA
C
C      COMPUTE EIGENVALUES AND EIGENVECTORS FOR THE FAR BOUNDARY
C
C      IF (ME.GT.0) THEN
C      IF (CONST2.GT.1.0E-08) THEN
C      FLAG=.TRUE.
C      CALL EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,AKR,AKI)
C      ELSE
C      CALL EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,AKR,AKI,IMP,FLAG)
C      END IF
C
C      * write(6,2001)
C      * 2001 format(' ')
C      * DO 10 I=1,ME
C      * WRITE(6,2010) AKR(I),AKI(I)
C      * 10 CONTINUE
C      * write (6,2001)
C      * do 2003 i=1,me
C      * write(6,2010) (afr(i,j),j=1,me)
C      * 2003 continue
C      * write (6,2001)

```

```

*      do 2004 i=1,me
*      write(6,2010) (afi(i,j),j=1,me)
* 2004 continue
*      write(6,2001)
*      END IF
C
C      FORM SYSTEM OF EQUATIONS
C
C      CALL FMAT(X,Y,Z,XM,YM,ZM,K,G,H,HOLD,FID,FIB,DFI,N0,N1)
*      do 2000 i=1,nx
*      write(6,3000)
* 3000 format(//)
*      write(6,2010) (h(i,j),j=1,nx)
* 2000 continue
*      write(6,2001)
*      write(6,2001)
*      do 2002 i=1,nx
*      write(6,3000)
*      write(6,2010) (g(i,j),j=1,nx)
* 2002 continue
C
C      MODIFY H MATRIX FOR BOUNDARY CONDITIONS AT BASE AND FAR END
C      AND COMPUTE DFI
C
      IF(CONST2.GT.1.0E-06) THEN
      CALL ADDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,DFI,FID,FIB,DF)
      ELSE
      CALL ADDUND(H,G,HOLD,AFR,AKR,AKI,N0,N1,N2,DFI,FID,FIB,FLAG,DF)
      END IF
*      write(6,2001)
*      write(6,2001)
*      do 2005 i=1,nx
*      write(6,3000)
*      write(6,2010) (h(i,j),j=1,nx)
* 2005 continue
*      write(6,2001)
*      write(6,2001)
*      write(6,2010) (dfi(i),i=1,nx)
C
C      SOLVE SYSTEM OF EQUATIONS
C
      CALL SLNPD(H,DFI,D,NX,NY,IMP)
C
*      DO 999 I=1,NX
*      WRITE(IMP,*)DFI(I)
* 999 CONTINUE
C
C      OUTPUT RESULTS
C
      CALL OUTPUT(XM,YM,ZM,FID,FIB,DFI,AFR,AFI,AKR,AKI,
*      N0,N1,N2,NOUT,FLAG,DF)
      GO TO 5
2010 format(6e13.5)
300 FORMAT(//,1X,70(' '),//)
350 FORMAT(1X,'EXCITING FREQUENCY',E14.5,/,1X,
*'WAVE CONSTANT',E14.5,/)
15 STOP

```

```

END
C
C -----
C
C SUBROUTINE INPUT(X,Y,Z,K,FID,FIB,N0,N1,N2,NOUT,C,DF)
C *****
C
C READS INPUT DATA
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
C COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C DOUBLE PRECISION X(1),Y(1),Z(1),FID(1),FIB(1),C,DF(1)
C DIMENSION K(4,NU)
C CHARACTER*4 TITLE(18)
C WRITE(IMP,100)
C
C READ TITLE
C
C READ (LEC,150)TITLE
C
C READ BASIC PARAMETERS
C
C READ(LEC,*)N,NC,N0,N1,N2,NOUT
C READ(LEC,*)HY,C,ALF
C CONST2=((1-ALF)/(1+ALF))*(1/C)
C
C WRITE(IMP,300)N,NC,N0,N1,N2,NOUT
C WRITE(IMP,350)HY,C,ALF,CONST2
C
C READ COORDINATES OF NODAL POINTS OF THE
C BOUNDARY ELEMENTS IN ARRAY X, Y AND Z
C
C WRITE(IMP,401)
C DO 10 I=1,NC
C READ(LEC,*)X(I),Y(I),Z(I)
C 10 WRITE(IMP,700)I,X(I),Y(I),Z(I)
C
C READ CONNECTIVITY MATRIX K
C
C WRITE(IMP,601)
C DO 20 I=1,N
C READ(LEC,*)(K(J,I),J=1,4)
C 20 WRITE(IMP,602)I,(K(J,I),J=1,4)
C
C READ BOUNDARY CONDITION ON THE DAM FACE
C
C WRITE(IMP,760)
C WRITE(IMP,800)
C ILAST=N1-N0
C DO 30 I=1,ILAST
C READ(LEC,*)FID(I)
C WRITE(IMP,750)I,FID(I)
C 30 CONTINUE
C
C READ BOUNDARY CONDITION ON THE BASE OF THE RESERVOIR
C

```

```

WRITE(IMP,770)
WRITE(IMP,800)
ILAST=N2-N1
DO 40 I=1,ILAST
  READ(LEC,*)FIB(I)
  WRITE(IMP,750)I,FIB(I)
40 CONTINUE
ME=N-N2
NELEM=ME
MF=N2-N1

C
C   READ BOUNDARY CONDITION FOR VERTICAL EXCITATION AT THE FAR END
C   VECTOR DF(ME)
C
  IF(ME.GT.0) THEN
    WRITE(IMP,780)
    WRITE(IMP,800)
    DO 50 I=1,ME
      READ(LEC,*)DF(I)
      WRITE(IMP,750) I,DF(I)
50 CONTINUE
    END IF

C
C
100 FORMAT(1X,60('*'))//)
150 FORMAT(18A4)
300 FORMAT(1X,'DATA',//,1X,'NUMBER OF BOUNDARY ELEMENTS',
  *I3,/,1X,'NUMBER OF CORNER NODES',I3,/,1X,'LAST ELEMENT NUMBER
  *AT TOP OF RESERVOIR NEAR THE DAM',I3,/,
  *1X,'LAST ELEMENT NUMBER AT BASE OF DAM',I3,/,1X,
  *'LAST ELEMENT NUMBER AT BOTTOM NEAR FAR END',I3,/,1X,
  *'FLAG FOR DETAILED OUTPUT',I3,/)
350 FORMAT(1X,'HEIGHT OF DAM',E14.5,/,1X,
  *'VELOCITY OF SOUND IN WATER',E14.5,/,1X,
  *'WAVE REFLECTION COEFFICIENT',E14.5,/,1X,
  *'FOUNDATION DAMPING COEFFICIENT',E14.5//)
401 FORMAT(1X,'COORDINATES OF CORNER NODES',
  *//,4X,'POINT',10X,'X',10X,'Y',10X,'Z')
501 FORMAT(5X,I4,3(4X,E14.7))
601 FORMAT(//'ELEMENT NO.',5X,'CONNECTIVITY')
602 FORMAT(I4,2X,4(4X,I4))
700 FORMAT(5X,I3,3(5X,E14.7))
750 FORMAT(5X,I3,8X,E14.7)
760 FORMAT(//' BOUNDARY CONDITIONS ON THE DAM FACE',//)
770 FORMAT(//' BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM',//)
780 FORMAT(//' BOUNDARY CONDITION FOR VERTICAL EXCITATION
  *AT THE FAR END - VECTOR DF',//)
800 FORMAT(5X,'NODE',6X,
  *'PRESCRIBED VALUE'//)

C
C
  RETURN
  END

C
C   -----
C
  SUBROUTINE EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,AKR,AKI)
  *****
C
C   PROGRAM EIGEN.3D.FTN
C

```



```

C   FOR SOLVING A GENERAL COMPLEX EIGENVALUE PROBLEM
C   ASSOCIATED WITH A 2D FEM FORMULATION
C   FOR FAR END CONDITION IN 3D-CASE

```

```

C   4-NODE ISOPARAMETRIC LINEAR ELEMENTS USED

```

```

C   DATE: JANUARY 26, 1986
C

```

```

DOUBLE PRECISION CONST,CONST1,CONST2,HY
REAL AFI (ND,ND), AFR (ND,ND), AKR (ND), AKI (ND)
OPEN (4,FORM='FORMATTED',STATUS='OLD',FILE='IN3D')
OPEN (5,FORM='FORMATTED',STATUS='NEW',FILE='HE3D')
OPEN (6,FORM='FORMATTED',STATUS='NEW',FILE='GE3D')
OPEN (7,FORM='FORMATTED',STATUS='NEW',FILE='BEO3D')
OPEN (8,FORM='FORMATTED',STATUS='NEW',FILE='TOT3D')
OPEN (9,FORM='FORMATTED',STATUS='NEW',FILE='I3D')
DATA IWRITE/9/
NCM=100
READ (4,*)NT

```

```

C
C

```

```

T=SNGL(CONST)
T1=SNGL(CONST1)
T2=SNGL(CONST2)

```

```

C

```

```

CALL FSFEM(AFI,AFR,ME,ND,NCM,NT,T,T1,T2,AKR,AKI)

```

```

C

```

```

RETURN
END

```

```

C

```

```

C

```

```

C

```

```

C

```

```

SUBROUTINE FSFEM(AFI,AFR,N,NM,NCM,NT,T,T1,T2,AKR,AKI)

```

```

C

```

```

*****
AFR= REAL PART OF THE EIGENVECTOR MATRIX

```

```

C

```

```

AFI= IMAGINARY PART OF EIGENVECTOR MATRIX

```

```

C

```

```

AKR=REAL PART OF KAPPA MATRIX DIAGONAL

```

```

C

```

```

AKI=IMAGINARY PART OF KAPPA MATRIX DIAGONAL

```

```

C

```

```

N=ME= NO. OF NODES ON THE FAR BOUNDARY

```

```

C

```

```

NM=ND= MAX. DIMENSION OF THE EIGENVECTOR MATRIX

```

```

C

```

```

T=CONST= OMEGA/C ( DP IN THE MAIN PROGRAM )

```

```

C

```

```

T1=CONST1= OMEGA ( DP IN THE MAIN PROGRAM )

```

```

C

```

```

T2=CONST2=GAMMA

```

```

C

```

```

REAL A ( 80, 80),B ( 80, 80),Z ( 80, 80),BHOLD ( 80, 80),
$      AHOLD ( 80, 80),AHOLDI ( 80, 80),BHOLDI ( 80, 80),
$      ALFR ( 80),ALFI ( 80),BETA ( 80),
$      EPS1, AI (80,80),BI (80,80),ZI (80,80)

```

```

REAL AMAX1

```

```

COMPLEX FI (80,80),XM (80,80)

```

```

COMPLEX XKAP (80),XK2 (80)

```

```

REAL AFR (NM,NM),AFI (NM,NM),AKR (NM),AKI (NM)

```

```

INTEGER ERROR

```

```

DIMENSION INDEX (80)

```

```

DATA IWRITE/9/

```

```

C

```

```

CALL SETE (T1,T2,N,NM,NCM,NT,A,AI,B,BI)

```

```

DO 717 I=1,N

```

```

DO 718 J=1,N
  AHOLD(I,J)=A(I,J)
  BHOLD(I,J)=B(I,J)
  AHOLDI(I,J)=AI(I,J)
  BHOLDI(I,J)=BI(I,J)
718 CONTINUE
717 CONTINUE
  WRITE(IWRITE,20)
  $      N
20  FORMAT(30H1THE FULL MATRIX A  OF ORDER,
  $      I4,22H IS (PRINTED BY ROWS)/)
  DO 30 I = 1,N
30  WRITE(IWRITE,40)
  $      (A(I,J),AI(I,J),J=1,N)
40  FORMAT(5(2F8.4,1HI,3X))
  WRITE(IWRITE,41)
  $      N
41  FORMAT(30H0THE FULL MATRIX B  OF ORDER,
  $      I4,22H IS (PRINTED BY ROWS)/)
  DO 42 I = 1,N
42  WRITE(IWRITE,43)
  $      (B(I,J),BI(I,J),J=1,N)
43  FORMAT(5(2F8.4,1HI,3X))

C
C
C
C
C
C
EIGENVALUES AND EIGENVECTORS USING CQZVAL AND CQZVEC

  EPS1 = 0.0
  CALL CQZHES(NM,N,A,AI,B,BI,.TRUE.,Z,ZI)
  CALL CQZVAL(NM,N,A,AI,B,BI,EPS1,ALFR,ALFI,BETA,
  $      .TRUE.,Z,ZI,ERROR)
  CALL CQZVEC(NM,N,A,AI,B,BI,ALFR,ALFI,BETA,Z,ZI)
  DO 719 I=1,N
  DO 720 J=1,N
    A(I,J)=AHOLD(I,J)
    B(I,J)=BHOLD(I,J)
    AI(I,J)=AHOLDI(I,J)
    BI(I,J)=BHOLDI(I,J)
720 CONTINUE
719 CONTINUE
  WRITE(IWRITE,292)
292  FORMAT(/15X,7HALFR(I),19X,7HALFI(I),19X,7HBETA(I))
  DO 295 I = 1,N
    WRITE(IWRITE,293)
  $      I,ALFR(I),ALFI(I),BETA(I)
293  FORMAT(12,3(1PE23.6,3X))
295  CONTINUE
  WRITE(IWRITE,300)
300  FORMAT(/14X,35HCOMPUTED EIGENVALUE AND EIGENVECTOR,20X)

C
  DO 510 K = 1, N
    BETA(K) = AMAX1(BETA(K),1.0E-50)
    ALFR(K) = ALFR(K) / BETA(K)
    ALFI(K) = ALFI(K) / BETA(K)

C
C
C
ONE EIGENVECTOR.
340  WRITE(IWRITE,350)

```

```

$          K,ALFR(K),ALFI(K), (Z(I,K),
$          ZI(I,K),I=1,N)
350  FORMAT(/I2,1P2E23.6/(5X,2E23.6))
510  CONTINUE
      DO 610 K=1,N
      DO 620 L=1,N
      FI(K,L)=CMPLX(Z(K,L),ZI(K,L))
620  CONTINUE
610  CONTINUE
      DO 611 I=1,N
      WRITE(IWRITE,211) (FI(I,J),J=1,N)
611  CONTINUE
      CALL NORMA(FI,B,N,NM,XM)
      DO 81 I=1,N
      WRITE(IWRITE,211) (FI(I,J),J=1,N)
      81 CONTINUE
211  FORMAT(8('(',E12.4,E12.4,')'))
      DO 613 I=1,N
      DO 612 J=1,N
      AFR(I,J)=REAL(FI(I,J))
      AFI(I,J)=AIMAG(FI(I,J))
612  CONTINUE
613  CONTINUE
      DO 171 I=1,N
      ALFR(I)=ALFR(I)-T*T
      XK2(I)=CMPLX(ALFR(I),ALFI(I))
      XKAP(I)=CSQRT(XK2(I))
      WRITE(IWRITE,*)XKAP(I)
      AKR(I)=REAL(XKAP(I))
      AKI(I)=AIMAG(XKAP(I))
171  CONTINUE
      RETURN
      END

```

C
C
C

SUBROUTINE CQZHE\$ (NM,N,AR,AI,BR,BI,MATZ,ZR,ZI)

THIS ALGORITHM QZ FOR THE CLASS OF COMPLEX GENERALIZED MATRIX
SYSTEMS EXHIBITING THE USE OF QZ TO FIND ALL THE EIGENVALUES
AND EIGENVECTORS FOR THE EIGENPROBLEM $A^*X = (\text{LAMBDA}) * B^*X$.

THE DIMENSION OF A,AI,B,BI,Z, AND ZI SHOULD BE NM BY NM.
THE DIMENSION OF ALFR,ALFI,BETA, AND NORM SHOULD BE NM.
THE DIMENSION OF AHOLD AND BHOLD SHOULD BE NM BY NM.
THE DIMENSION OF AHOLDI AND BHOLDI SHOULD BE NM BY NM.
HERE NM = 600.

C

INTEGER I,J,K,L,N,K1,LB,L1,NM,NK1,NM1
REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ZR(NM,N),ZI(NM,N)
REAL R,S,T,TI,U1,U2,XI,XR,YI,YR,RHO,U11
REAL SQRT,CABS,ABS
LOGICAL MATZ
COMPLEX CMPLX

C

THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FIRST STEP OF THE
QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.

C

THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX GENERAL MATRICES AND

C

C REDUCES ONE OF THEM TO UPPER HESSENBERG FORM WITH REAL (AND NON-
 C NEGATIVE) SUBDIAGONAL ELEMENTS AND THE OTHER TO UPPER TRIANGULAR
 C FORM USING UNITARY TRANSFORMATIONS. IT IS USUALLY FOLLOWED BY
 C CQZVAL AND POSSIBLY CQZVEC.

C ON INPUT-

C NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
 C ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
 C DIMENSION STATEMENT,

C N IS THE ORDER OF THE MATRICES,

C A=(AR,AI) CONTAINS A COMPLEX GENERAL MATRIX,

C B=(BR,BI) CONTAINS A COMPLEX GENERAL MATRIX,

C MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS
 C ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING
 C EIGENVECTORS, AND TO .FALSE. OTHERWISE.

C ON OUTPUT-

C A HAS BEEN REDUCED TO UPPER HESSENBERG FORM. THE ELEMENTS
 C BELOW THE FIRST SUBDIAGONAL HAVE BEEN SET TO ZERO, AND THE
 C SUBDIAGONAL ELEMENTS HAVE BEEN MADE REAL (AND NON-NEGATIVE),

C B HAS BEEN REDUCED TO UPPER TRIANGULAR FORM. THE ELEMENTS
 C BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,

C Z=(ZR,ZI) CONTAINS THE PRODUCT OF THE RIGHT HAND
 C TRANSFORMATIONS IF MATZ HAS BEEN SET TO .TRUE.
 C OTHERWISE, Z IS NOT REFERENCED.

C QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
 C APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

 C ***** INITIALIZE Z *****
 C IF (.NOT. MATZ) GO TO 10

C DO 3 I = 1, N

C DO 2 J = 1, N
 C ZR(I,J) = 0.0
 C ZI(I,J) = 0.0

C 2 CONTINUE

C ZR(1,1) = 1.0

C 3 CONTINUE

C ***** REDUCE B TO UPPER TRIANGULAR FORM WITH
 C TEMPORARILY REAL DIAGONAL ELEMENTS *****

C 10 IF (N .LE. 1) GO TO 170
 C NM1 = N - 1

C DO 100 L = 1, NM1
 C L1 = L + 1
 C S = 0.0

```

DO 20 I = L, N
  S = S + ABS(BR(I,L)) + ABS(BI(I,L))
20 CONTINUE
C
  IF (S .EQ. 0.0) GO TO 100
  RHO = 0.0
C
  DO 25 I = L, N
    BR(I,L) = BR(I,L) / S
    BI(I,L) = BI(I,L) / S
    RHO = RHO + BR(I,L)**2 + BI(I,L)**2
25 CONTINUE
C
  R = SQRT(RHO)
  XR = CABS(CMPLX(BR(L,L),BI(L,L)))
  IF (XR .EQ. 0.0) GO TO 27
  RHO = RHO + XR * R
  U1 = -BR(L,L) / XR
  U1I = -BI(L,L) / XR
  YR = R / XR + 1.0
  BR(L,L) = YR * BR(L,L)
  BI(L,L) = YR * BI(L,L)
  GO TO 28
C
27 BR(L,L) = R
  U1 = -1.0
  U1I = 0.0
C
28 DO 50 J = L1, N
  T = 0.0
  TI = 0.0
C
  DO 30 I = L, N
    T = T + BR(I,L) * BR(I,J) + BI(I,L) * BI(I,J)
    TI = TI + BR(I,L) * BI(I,J) - BI(I,L) * BR(I,J)
30 CONTINUE
C
  T = T / RHO
  TI = TI / RHO
C
  DO 40 I = L, N
    BR(I,J) = BR(I,J) - T * BR(I,L) + TI * BI(I,L)
    BI(I,J) = BI(I,J) - T * BI(I,L) - TI * BR(I,L)
40 CONTINUE
C
  XI = U1 * BI(L,J) - U1I * BR(L,J)
  BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
  BI(L,J) = XI
50 CONTINUE
C
  DO 80 J = 1, N
  T = 0.0
  TI = 0.0
C
  DO 60 I = L, N
    T = T + BR(I,L) * AR(I,J) + BI(I,L) * AI(I,J)
    TI = TI + BR(I,L) * AI(I,J) - BI(I,L) * AR(I,J)
60 CONTINUE
C
  T = T / RHO

```

```

C      TI = TI / RHO
C      DO 70 I = L, N
C          AR(I,J) = AR(I,J) - T * BR(I,L) + TI * BI(I,L)
C          AI(I,J) = AI(I,J) - T * BI(I,L) - TI * BR(I,L)
70      CONTINUE
C      XI = U1 * AI(L,J) - U1I * AR(L,J)
C      AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
C      AI(L,J) = XI
80      CONTINUE
C      BR(L,L) = R * S
C      BI(L,L) = 0.0
C      DO 90 I = L1, N
C          BR(I,L) = 0.0
C          BI(I,L) = 0.0
90      CONTINUE
C      100 CONTINUE
C      ***** REDUCE A TO UPPER HESSENBERG FORM WITH REAL SUBDIAGONAL
C      ELEMENTS, WHILE KEEPING B TRIANGULAR *****
C      DO 160 K = 1, NM1
C          K1 = K + 1
C      ***** SET BOTTOM ELEMENT IN K-TH COLUMN OF A REAL *****
C          IF (AI(N,K) .EQ. 0.0) GO TO 105
C          R = CABS(CMPLX(AR(N,K),AI(N,K)))
C          U1 = AR(N,K) / R
C          U1I = AI(N,K) / R
C          AR(N,K) = R
C          AI(N,K) = 0.0
C      DO 103 J = K1, N
C          XI = U1 * AI(N,J) - U1I * AR(N,J)
C          AR(N,J) = U1 * AR(N,J) + U1I * AI(N,J)
C          AI(N,J) = XI
103      CONTINUE
C      XI = U1 * BI(N,N) - U1I * BR(N,N)
C      BR(N,N) = U1 * BR(N,N) + U1I * BI(N,N)
C      BI(N,N) = XI
105      IF (K .EQ. NM1) GO TO 170
C      NK1 = NM1 - K
C      ***** FOR L=N-1 STEP -1 UNTIL K+1 DO -- *****
C      DO 150 LB = 1, NK1
C          L = N - LB
C          L1 = L + 1
C      ***** ZERO A(L+1,K) *****
C          S = ABS(AR(L,K)) + ABS(AI(L,K)) + AR(L1,K)
C          IF (S .EQ. 0.0) GO TO 150
C          U1 = AR(L,K) / S
C          U1I = AI(L,K) / S
C          U2 = AR(L1,K) / S
C          R = SQRT(U1*U1+U1I*U1I+U2*U2)
C          U1 = U1 / R
C          U1I = U1I / R
C          U2 = U2 / R
C          AR(L,K) = R * S
C          AI(L,K) = 0.0

```

```

C      AR(L1,K) = 0.0

C      DO 110 J = K1, N
        XR = AR(L,J)
        XI = AI(L,J)
        YR = AR(L1,J)
        YI = AI(L1,J)
        AR(L,J) = U1 * XR + U1I * XI + U2 * YR
        AI(L,J) = U1 * XI - U1I * XR + U2 * YI
        AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110    CONTINUE
C
        XR = BR(L,L)
        BR(L,L) = U1 * XR
        BI(L,L) = -U1I * XR
        BR(L1,L) = -U2 * XR
C
        DO 120 J = L1, N
        XR = BR(L,J)
        XI = BI(L,J)
        YR = BR(L1,J)
        YI = BI(L1,J)
        BR(L,J) = U1 * XR + U1I * XI + U2 * YR
        BI(L,J) = U1 * XI - U1I * XR + U2 * YI
        BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
120    CONTINUE
C      ***** ZERO B(L+1,L) *****
        S = ABS(BR(L1,L1)) + ABS(BI(L1,L1)) + ABS(BR(L1,L))
        IF (S .EQ. 0.0) GO TO 150
        U1 = BR(L1,L1) / S
        U1I = BI(L1,L1) / S
        U2 = BR(L1,L) / S
        R = SQRT(U1*U1+U1I*U1I+U2*U2)
        U1 = U1 / R
        U1I = U1I / R
        U2 = U2 / R
        BR(L1,L1) = R * S
        BI(L1,L1) = 0.0
        BR(L1,L) = 0.0
C
        DO 130 I = 1, L
        XR = BR(I,L1)
        XI = BI(I,L1)
        YR = BR(I,L)
        YI = BI(I,L)
        BR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        BI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        BR(I,L) = U1 * YR - U1I * YI - U2 * XR
        BI(I,L) = U1 * YI + U1I * YR - U2 * XI
130    CONTINUE
C
        DO 140 I = 1, N
        XR = AR(I,L1)
        XI = AI(I,L1)
        YR = AR(I,L)
        YI = AI(I,L)
        AR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        AI(I,L1) = U1 * XI - U1I * XR + U2 * YI

```


REAL AND NON-NEGATIVE. IT THEN RETURNS QUANTITIES WHOSE RATIOS
GIVE THE GENERALIZED EIGENVALUES. IT IS USUALLY PRECEDED BY
CQZHEs AND POSSIBLY FOLLOWED BY CQZVEC.

ON INPUT-

NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,

N IS THE ORDER OF THE MATRICES,

A=(AR,AI) CONTAINS A COMPLEX UPPER HESSENBERG MATRIX
WITH REAL SUBDIAGONAL ELEMENTS,

B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,

EPS1 IS A TOLERANCE USED TO DETERMINE NEGLIGIBLE ELEMENTS.
EPS1 $\approx$ 0.0 (OR NEGATIVE) MAY BE INPUT, IN WHICH CASE AN
ELEMENT WILL BE NEGLECTED ONLY IF IT IS LESS THAN ROUNDOFF
ERROR TIMES THE NORM OF ITS MATRIX. IF THE INPUT EPS1 IS
POSITIVE, THEN AN ELEMENT WILL BE CONSIDERED NEGLIGIBLE
IF IT IS LESS THAN EPS1 TIMES THE NORM OF ITS MATRIX. A
POSITIVE VALUE OF EPS1 MAY RESULT IN FASTER EXECUTION,
BUT LESS ACCURATE RESULTS,

MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS
ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING
EIGENVECTORS, AND TO .FALSE. OTHERWISE,

Z=(ZR,ZI) CONTAINS, IF MATZ HAS BEEN SET TO .TRUE., THE
TRANSFORMATION MATRIX PRODUCED IN THE REDUCTION
BY CQZHEs, IF PERFORMED, OR ELSE THE IDENTITY MATRIX.
IF MATZ HAS BEEN SET TO .FALSE., Z IS NOT REFERENCED.

ON OUTPUT

A HAS BEEN REDUCED TO UPPER TRIANGULAR FORM. THE ELEMENTS
BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,

B IS STILL IN UPPER TRIANGULAR FORM, ALTHOUGH ITS ELEMENTS
HAVE BEEN ALTERED. IN PARTICULAR, ITS DIAGONAL HAS BEEN SET
REAL AND NON-NEGATIVE. THE LOCATION BR(N,1) IS USED TO
STORE EPS1 TIMES THE NORM OF B FOR LATER USE BY CQZVEC,

ALFR AND ALFI CONTAIN THE REAL AND IMAGINARY PARTS OF THE
DIAGONAL ELEMENTS OF THE TRIANGULARIZED A MATRIX,

BETA CONTAINS THE REAL NON-NEGATIVE DIAGONAL ELEMENTS OF THE
CORRESPONDING B. THE GENERALIZED EIGENVALUES ARE THEN
THE RATIOS ((ALFR+I*ALFI)/BETA),

Z CONTAINS THE PRODUCT OF THE RIGHT HAND TRANSFORMATIONS
(FOR BOTH STEPS) IF MATZ HAS BEEN SET TO .TRUE.,

IERR IS SET TO
ZERO FOR NORMAL RETURN,
J IF AR(J,J-1) HAS NOT BECOME
ZERO AFTER 50 ITERATIONS.

```

C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      IERR = 0
C      ***** COMPUTE EPSA,EPSB *****
C      ANORM = 0.0
C      BNORM = 0.0
C
C      DO 30 I = 1, N
C          ANI = 0.0
C          IF (I .NE. 1) ANI = ABS(AR(I,I-1))
C          BNI = 0.0
C
C          DO 20 J = I, N
C              ANI = ANI + ABS(AR(I,J)) + ABS(AI(I,J))
C              BNI = BNI + ABS(BR(I,J)) + ABS(BI(I,J))
20      CONTINUE
C
C          IF (ANI .GT. ANORM) ANORM = ANI
C          IF (BNI .GT. BNORM) BNORM = BNI
30      CONTINUE
C
C          IF (ANORM .EQ. 0.0) ANORM = 1.0
C          IF (BNORM .EQ. 0.0) BNORM = 1.0
C          EP = EPS1
C          IF (EP .GT. 0.0) GO TO 50
C      ***** COMPUTE ROUNDOFF LEVEL IF EPS1 IS ZERO *****
C          EP = 1.0
40      EP = EP / 2.0
C          IF (1.0 + EP .GT. 1.0) GO TO 40
50      EPSA = EP * ANORM
C          EPSB = EP * BNORM
C      ***** REDUCE A TO TRIANGULAR FORM, WHILE
C          KEEPING B TRIANGULAR *****
C          LOR1 = 1
C          ENORN = N
C          EN = N
C      ***** BEGIN QZ STEP *****
60      IF (EN .EQ. 0) GO TO 1001
C          IF (.NOT. MATZ) ENORN = EN
C          ITS = 0
C          NA = EN - 1
C          ENM2 = NA - 1
C      ***** CHECK FOR CONVERGENCE OR REDUCIBILITY.
C          FOR L=EN STEP -1 UNTIL 1 DO -- *****
70      DO 80 LL = 1, EN
C          LM1 = EN - LL
C          L = LM1 + 1
C          IF (L .EQ. 1) GO TO 95
C          IF (ABS(AR(L,LM1)) .LE. EPSA) GO TO 90
80      CONTINUE
C
C          AR(L,LM1) = 0.0
C      ***** SET DIAGONAL ELEMENT AT TOP OF B REAL *****
95      B11 = CABS(CMPLX(BR(L,L),BI(L,L)))
C          IF (B11 .EQ. 0.0) GO TO 98
C          U1 = BR(L,L) / B11
C          U1I = BI(L,L) / B11

```

```

C      DO 97 J = L, ENORN
        XI = U1 * AI(L,J) - U1I * AR(L,J)
        AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
        AI(L,J) = XI
        XI = U1 * BI(L,J) - U1I * BR(L,J)
        BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
        BI(L,J) = XI
97    CONTINUE
C
        BI(L,L) = 0.0
98    IF (L .NE. EN) GO TO 100
C      ***** 1-BY-1 BLOCK ISOLATED *****
        ALFR(EN) = AR(EN,EN)
        ALFI(EN) = AI(EN,EN)
        BETA(EN) = B11
        EN = NA
        GO TO 60
C      ***** CHECK FOR SMALL TOP OF B *****
100   L1 = L + 1
        IF (B11 .GT. EPSB) GO TO 120
        BR(L,L) = 0.0
        S = ABS(AR(L,L)) + ABS(AI(L,L)) + ABS(AR(L1,L))
        U1 = AR(L,L) / S
        U1I = AI(L,L) / S
        U2 = AR(L1,L) / S
        R = SQRT(U1*U1+U1I*U1I+U2*U2)
        U1 = U1 / R
        U1I = U1I / R
        U2 = U2 / R
        AR(L,L) = R * S
        AI(L,L) = 0.0
C
        DO 110 J = L1, ENORN
            XR = AR(L,J)
            XI = AI(L,J)
            YR = AR(L1,J)
            YI = AI(L1,J)
            AR(L,J) = U1 * XR + U1I * XI + U2 * YR
            AI(L,J) = U1 * XI - U1I * XR + U2 * YI
            AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
            AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
            XR = BR(L,J)
            XI = BI(L,J)
            YR = BR(L1,J)
            YI = BI(L1,J)
            BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
            BR(L,J) = U1 * XR + U1I * XI + U2 * YR
            BI(L,J) = U1 * XI - U1I * XR + U2 * YI
            BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110   CONTINUE
C
        LM1 = L
        L = L1
        GO TO 90
C      ***** ITERATION STRATEGY *****
120   IF (ITS .EQ. 50) GO TO 1000
        IF (ITS .EQ. 10) GO TO 135
C      ***** DETERMINE SHIFT *****
        B33 = BR(NA,NA)

```

```

      B33I = BI(NA,NA)
      IF (CABS(CMPLX(B33,B33I)) .GE. EPSB) GO TO 122
      B33 = EPSB
      B33I = 0.0
122  B44 = BR(EN,EN)
      B44I = BI(EN,EN)
      IF (CABS(CMPLX(B44,B44I)) .GE. EPSB) GO TO 124
      B44 = EPSB
      B44I = 0.0
124  B3344 = B33 * B44 - B33I * B44I
      B3344I = B33 * B44I + B33I * B44
      A33 = AR(NA,NA) * B44 - AI(NA,NA) * B44I
      A33I = AR(NA,NA) * B44I + AI(NA,NA) * B44
      A34 = AR(NA,EN) * B33 - AI(NA,EN) * B33I
      X  - AR(NA,NA) * BR(NA,EN) + AI(NA,NA) * BI(NA,EN)
      A34I = AR(NA,EN) * B33I + AI(NA,EN) * B33
      X  - AR(NA,NA) * BI(NA,EN) - AI(NA,NA) * BR(NA,EN)
      A43 = AR(EN,NA) * B44
      A43I = AR(EN,NA) * B44I
      A44 = AR(EN,EN) * B33 - AI(EN,EN) * B33I - AR(EN,NA) * BR(NA,EN)
      A44I = AR(EN,EN) * B33I + AI(EN,EN) * B33 - AR(EN,NA) * BI(NA,EN)
      SH = A44
      SHI = A44I
      XR = A34 * A43 - A34I * A43I
      XI = A34 * A43I + A34I * A43
      IF (XR .EQ. 0.0 .AND. XI .EQ. 0.0) GO TO 140
      YR = (A33 - SH) / 2.0
      YI = (A33I - SHI) / 2.0
      Z3 = CSQRT(CMPLX(YR**2-YI**2+XR,2.0*YR*YI+XI))
      U1 = REAL(Z3)
      U1I = AIMAG(Z3)
      IF (YR * U1 + YI * U1I .GE. 0.0) GO TO 125
      U1 = -U1
      U1I = -U1I
125  Z3 = (CMPLX(SH,SHI) - CMPLX(XR,XI) / CMPLX(YR+U1,YI+U1I))
      X  / CMPLX(B3344,B3344I)
      SH = REAL(Z3)
      SHI = AIMAG(Z3)
      GO TO 140
C    ***** AD HOC SHIFT *****
135  SH = AR(EN,NA) + AR(NA,ENM2)
      SHI = 0.0
C    ***** DETERMINE ZEROth COLUMN OF A *****
140  A1 = AR(L,L) / B11 - SH
      A1I = AI(L,L) / B11 - SHI
      A2 = AR(L1,L) / B11
      ITS = ITS + 1
      IF (.NOT. MATZ) LOR1 = L
C    ***** MAIN LOOP *****
      DO 260 K = L, NA
        K1 = K + 1
        K2 = K + 2
        KM1 = MAX0(K-1,L)
C    ***** ZERO A(K+1,K-1) *****
        IF (K .EQ. L) GO TO 170
        A1 = AR(K,KM1)
        A1I = AI(K,KM1)
        A2 = AR(K1,KM1)
170  S = ABS(A1) + ABS(A1I) + ABS(A2)
      U1 = A1 / S

```

```

      U1I = A1I / S
      U2 = A2 / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
C
      DO 180 J = KM1, ENORN
        XR = AR(K,J)
        XI = AI(K,J)
        YR = AR(K1,J)
        YI = AI(K1,J)
        AR(K,J) = U1 * XR + U1I * XI + U2 * YR
        AI(K,J) = U1 * XI - U1I * XR + U2 * YI
        AR(K1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(K1,J) = U1 * YI + U1I * YR - U2 * XI
        XR = BR(K,J)
        XI = BI(K,J)
        YR = BR(K1,J)
        YI = BI(K1,J)
        BR(K,J) = U1 * XR + U1I * XI + U2 * YR
        BI(K,J) = U1 * XI - U1I * XR + U2 * YI
        BR(K1,J) = U1 * YR - U1I * YI - U2 * XR
        BI(K1,J) = U1 * YI + U1I * YR - U2 * XI
180    CONTINUE
C
      IF (K .EQ. L) GO TO 240
      AI(K,KM1) = 0.0
      AR(K1,KM1) = 0.0
      AI(K1,KM1) = 0.0
C
      ***** ZERO B(K+1,K) *****
240    S = ABS(BR(K1,K1)) + ABS(BI(K1,K1)) + ABS(BR(K1,K))
      U1 = BR(K1,K1) / S
      U1I = BI(K1,K1) / S
      U2 = BR(K1,K) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      IF (K .EQ. NA) GO TO 245
      XR = AR(K2,K1)
      AR(K2,K1) = U1 * XR
      AI(K2,K1) = -U1I * XR
      AR(K2,K) = -U2 * XR
C
245    DO 250 I = LOR1, K1
      XR = AR(I,K1)
      XI = AI(I,K1)
      YR = AR(I,K)
      YI = AI(I,K)
      AR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      AI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      AR(I,K) = U1 * YR - U1I * YI - U2 * XR
      AI(I,K) = U1 * YI + U1I * YR - U2 * XI
      XR = BR(I,K1)
      XI = BI(I,K1)
      YR = BR(I,K)
      YI = BI(I,K)
      BR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      BI(I,K1) = U1 * XI - U1I * XR + U2 * YI

```

```

                BR(I,K) = U1 * YR - U1I * YI - U2 * XR
                BI(I,K) = U1 * YI + U1I * YR - U2 * XI
250      CONTINUE
C
        BI(K1,K1) = 0.0
        BR(K1,K) = 0.0
        BI(K1,K) = 0.0
        IF (.NOT. MATZ) GO TO 260
C
        DO 255 I = 1, N
            XR = ZR(I,K1)
            XI = ZI(I,K1)
            YR = ZR(I,K)
            YI = ZI(I,K)
            ZR(I,K1) = U1 * XR + U1I * XI + U2 * YR
            ZI(I,K1) = U1 * XI - U1I * XR + U2 * YI
            ZR(I,K) = U1 * YR - U1I * YI - U2 * XR
            ZI(I,K) = U1 * YI + U1I * YR - U2 * XI
255      CONTINUE
C
260      CONTINUE
C      ***** SET LAST A SUBDIAGONAL REAL AND END QZ STEP *****
        IF (AI(EN,NA) .EQ. 0.0) GO TO 70
        R = CABS(CMPLX(AR(EN,NA), AI(EN,NA)))
        U1 = AR(EN,NA) / R
        U1I = AI(EN,NA) / R
        AR(EN,NA) = R
        AI(EN,NA) = 0.0
C
        DO 270 J = EN, ENORN
            XI = U1 * AI(EN,J) - U1I * AR(EN,J)
            AR(EN,J) = U1 * AR(EN,J) + U1I * AI(EN,J)
            AI(EN,J) = XI
            XI = U1 * BI(EN,J) - U1I * BR(EN,J)
            BR(EN,J) = U1 * BR(EN,J) + U1I * BI(EN,J)
            BI(EN,J) = XI
270      CONTINUE
C
        GO TO 70
C      ***** SET ERROR -- BOTTOM SUBDIAGONAL ELEMENT HAS NOT
C      ***** BECOME NEGLIGIBLE AFTER 50 ITERATIONS *****
1000      IERR = EN
C      ***** SAVE EPSB FOR USE BY CQZVEC *****
1001      IF (N .GT. 1) BR(N,1) = EPSB
        RETURN
C      ***** LAST CARD OF CQZVAL *****
        END
C
C      -----
C
        SUBROUTINE CQZVEC(NM,N,AR,AI,BR,BI,ALFR,ALFI,BETA,ZR,ZI)
        *****
C
        INTEGER I,J,K,M,N,EN,II,JJ,NA,NM,NN
        REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ALFR(N),ALFI(N),
X      BETA(N),ZR(NM,N),ZI(NM,N)
        REAL R,T,RI,TI,XI,ALMI,ALMR,BETM,EPSB
        REAL CABS
        COMPLEX Z3
        COMPLEX CMPLX

```

REAL REAL,AIMAG

THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FOURTH STEP OF THE QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS, SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.

THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES IN UPPER TRIANGULAR FORM, WHERE ONE OF THEM FURTHER MUST HAVE REAL DIAGONAL ELEMENTS. IT COMPUTES THE EIGENVECTORS OF THE TRIANGULAR PROBLEM AND TRANSFORMS THE RESULTS BACK TO THE ORIGINAL COORDINATE SYSTEM. IT IS USUALLY PRECEDED BY CQZHES AND CQZVAL.

ON INPUT-

NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT,

N IS THE ORDER OF THE MATRICES,

A=(AR,AI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,

B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX WITH REAL DIAGONAL ELEMENTS. IN ADDITION, LOCATION BR(N,1) CONTAINS THE TOLERANCE QUANTITY (EPSB) COMPUTED AND SAVED IN CQZVAL,

ALFR, ALFI, AND BETA ARE VECTORS WITH COMPONENTS WHOSE RATIOS $((ALFR + I * ALFI) / BETA)$ ARE THE GENERALIZED EIGENVALUES. THEY ARE USUALLY OBTAINED FROM CQZVAL,

Z=(ZR,ZI) CONTAINS THE TRANSFORMATION MATRIX PRODUCED IN THE REDUCTIONS BY CQZHES AND CQZVAL, IF PERFORMED. IF THE EIGENVECTORS OF THE TRIANGULAR PROBLEM ARE DESIRED, Z MUST CONTAIN THE IDENTITY MATRIX.

ON OUTPUT-

A IS UNALTERED,

B HAS BEEN DESTROYED,

ALFR, ALFI, AND BETA ARE UNALTERED,

Z CONTAINS THE EIGENVECTORS. EACH EIGENVECTOR IS NORMALIZED SO THAT THE MODULUS OF ITS LARGEST COMPONENT IS 1.0 .

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW, APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

IF (N .LE. 1) GO TO 1001

EPSB = BR(N,1)

***** FOR EN=N STEP -1 UNTIL 2 DO -- *****

DO 800 NN = 2, N

EN = N + 2 - NN

```

      NA = EN - 1
      ALMR = ALFR(EN)
      ALMI = ALFI(EN)
      BETM = BETA(EN)
C ***** FOR I=EN-1 STEP -1 UNTIL 1 DO -- *****
      DO 700 II = 1, NA
        I = EN - II
        R = 0.0
        RI = 0.0
        M = I + 1
C
        DO 610 J = M, EN
          T = BETM * AR(I,J) - ALMR * BR(I,J) + ALMI * BI(I,J)
          TI = BETM * AI(I,J) - ALMR * BI(I,J) - ALMI * BR(I,J)
          IF (J .EQ. EN) GO TO 605
          XI = T * BI(J,EN) + TI * BR(J,EN)
          T = T * BR(J,EN) - TI * BI(J,EN)
          TI = XI
605      R = R + T
          RI = RI + TI
610      CONTINUE
C
          T = ALMR * BETA(I) - BETM * ALFR(I)
          TI = ALMI * BETA(I) - BETM * ALFI(I)
          IF (T .EQ. 0.0 .AND. TI .EQ. 0.0) T = EPSB
          Z3 = CMPLX(R,RI) / CMPLX(T,TI)
          BR(I,EN) = REAL(Z3)
          BI(I,EN) = AIMAG(Z3)
700      CONTINUE
C
800 CONTINUE
C ***** END BACK SUBSTITUTION.
C TRANSFORM TO ORIGINAL COORDINATE SYSTEM.
C FOR J=N STEP -1 UNTIL 2 DO -- *****
      DO 880 JJ = 2, N
        J = N + 2 - JJ
        M = J - 1
C
        DO 880 I = 1, N
C
          DO 860 K = 1, M
            ZR(I,J) = ZR(I,J) + ZR(I,K) * BR(K,J) - ZI(I,K) * BI(K,J)
            ZI(I,J) = ZI(I,J) + ZR(I,K) * BI(K,J) + ZI(I,K) * BR(K,J)
860          CONTINUE
C
880 CONTINUE
C ***** NORMALIZE SO THAT MODULUS OF LARGEST
C COMPONENT OF EACH VECTOR IS 1 *****
      DO 950 J = 1, N
        T = 0.0
C
        DO 930 I = 1, N
          R = CABS(CMPLX(ZR(I,J),ZI(I,J)))
          IF (R .GT. T) T = R
930      CONTINUE
C
        DO 940 I = 1, N
          ZR(I,J) = ZR(I,J) / T
          ZI(I,J) = ZI(I,J) / T
940      CONTINUE

```



```

C
  950 CONTINUE
C
1001 RETURN
C ***** LAST CARD OF CQZVEC *****
  END
C-----
C
C
  SUBROUTINE NORMA(FI,G,N,NM,XM;
C *****
C
  COMPLEX FI(NM,NM),XM(NM,NM),TEMP(80),SUM1,SUM2
  COMPLEX XI(80,80),X(80,80),X1,X2
  REAL G(NM,NM)
  DATA IWRITE/9/
C
  DO 30 I=1,N
    DO 40 J=1,N
      SUM1=(0.0,0.0)
      DO 50 K=1,N
        SUM1=SUM1+CMPLX(G(J,K))*FI(K,I)
50    CONTINUE
      TEMP(J)=SUM1
40    CONTINUE
      SUM2=(0.0,0.0)
      DO 41 J=1,N
        SUM2=SUM2+FI(J,I)*TEMP(J)
41    CONTINUE
      SUM2=CSQRT(SUM2)
      DO 51 J=1,N
        FI(J,I)=FI(J,I)/SUM2
51    CONTINUE
30    CONTINUE
C
C FOR CHECK CARRY OUT [FIT]*[G]*[FI]=[I]
C
  DO 100 I=1,N
    DO 110 J=1,N
      X1=(0.0,0.0)
      DO 120 K=1,N
        X1=X1+FI(K,I)*CMPLX(G(K,J))
120    CONTINUE
      X(I,J)=X1
110    CONTINUE
100    CONTINUE
      DO 130 I=1,N
        DO 140 J=1,N
          X2=(0.0,0.0)
          DO 150 K=1,N
            X2=X2+X(I,K)*FI(K,J)
150        CONTINUE
          XI(I,J)=X2
140        CONTINUE
130        CONTINUE
          DO 161 I=1,N
            WRITE(IWRITE,111) (XI(I,J),J=1,N)
161        CONTINUE
111    FORMAT(8('(',E12.4,E12.4,')'))

```

RETURN
END

SUBROUTINE SETE(T1,T2,NPOIN,NM,NCM,NT,H,BO,G,BI)

SUBROUTINE SETE
FORMS MATRICES [HE], [GE] AND [BOE]
REQUIRED TO FORMULATE AN EIGENVALUE PROBLEM
AT THE FAR END OF 3-D RESERVOIR
MATRICES [HE], [GE] AND [BOE] ARE FORMED FOR EACH ELEMENT
USING 2D FINITE ELEMENT FORMULATION
TWO-DIMENSIONAL 4-NODE QUADRILATERAL ELEMENTS ARE USED

DATE: AUGUST 4, 1987

NELEM = NUMBER OF ELEMENTS
NM = ND MAX DIMENSION OF THE EIGENVECTOR MATRIX
NPOIN = N NUMBER OF NODAL POINTS (NODES) - NT
NNODE = 4, NUMBER OF NODES IN AN ELEMENT
NDIME = 2, NUMBER OF COORDINATE COMPONENTS REQUIRED TO DEFINE EACH NODE
NGAUS = 2, THE ORDER OF GAUSSIAN NUMERICAL INTEGRATION RULE TO BE EMPLOYED
NGASP = 4, NUMBER OF GAUSSIAN INTEGRATION POINTS

EINDEX(NELEM) = ELEMENT INDEX MATRIX DESCRIBES ELEMENT TYPE W.R.T. LOCATION
ON THE BOUNDARY OF THE CHANNEL (CREATION OF CONTRIBUTIONS
FOR MATRIX [BOE])

LNODS(NELEM,NNODE) = ELEMENT TOPOLOGY - CONNECTIVITY MATRIX
COORD(NPOIN,NDIME) = EACH NODE MUST BE REFERRED TO THE GLOBAL COORDINATE
SYSTEM

POSGP(NGAUS) = POSITION OF GAUSSIAN POINT IN THE LOCAL COORDINATE SYSTEM

WEIGP(NGAUS) = GAUSSIAN WEIGHT ASSOCIATED TO EACH GAUSSIAN POINT

ELCOLD(NDIME,NNODE) = LOCAL ARRAY OF NODAL CARTESIAN COORDINATES
OF AN ELEMENT

SHAPE(NNODE) = SHAPE FUNCTIONS ASSOCIATED WITH EACH NODE OF AN ELEMENT
IN LOCAL COORDINATES SAMPLED IN A POINT WITHIN THE ELEMENT

DERIV(NDIME,NNODE) = SHAPE FUNCTION DERIVATIVE AT SAMPLING POINT WITHIN
THE ELEMENT

GPCOD(NDIME,NGASP) = LOCAL ARRAY OF CARTESIAN COORDINATES OF THE GAUSS
POINTS OF AN ELEMENT

CARTD(NDIME,NNODE) = CARTESIAN SHAPE FUNCTION DERIVATIVES ASSOCIATED WITH
THE NODES OF AN ELEMENT SAMPLED AT ANY POINT WITHIN
THE ELEMENT

HE(NNODE,NNODE) = REAL VALUED MATRIX TO BE FORMED AT THE ELEMENT LEVEL

GE(NNODE,NNODE) = REAL VALUED MATRIX TO BE FORMED AT THE ELEMENT LEVEL

BOE(NNODE,NNODE) = REAL VALUED MATRIX TO BE FORMED AT THE ELEMENT LEVEL
(CONTRIBUTIONS DUE TO BOUNDARY DAMPING)

COMMON/SET/ NELEM,NNODE,NDIME,NGAUS,NGASP

REAL COORD(100,2)
REAL POSGP(2),WEIGP(2),SHAPE(4),DERIV(2,4)
REAL ELCOD(2,4),GPCOD(2,4),CARTD(2,4)

```

REAL HE(4,4),GE(4,4),BOE(4,4),B1(4),B2(4)
REAL LENG1(80),LENG2(80)
REAL HT(80,80),GT(80,80),BOT(80,80),BIT(80,80)
REAL H(NM,NM),G(NM,NM),BO(NM,NM),BI(NM,NM)
INTEGER EINDEX(80)
INTEGER LNODS(80,4)
INTEGER NTOP(80)

C
C
C
READ(4,*)NELEM,NNODE,NDIME,NGAUS,NGASP
CALL INPUT2(NPOIN,LNODS,COORD,EINDEX,NTOP,NCM,NM,NT)

C
CALL GAUSSQ(POSGP,WEIGP)

C
ENTER LOOP OVER EACH ELEMENT
C
DO 100 IELEM=1,NELEM
  INDEX=EINDEX(IELEM)

C
  EVALUATE COORDINATES OF THE ELEMENT
  NODAL POINTS
  C
  DO 10 INODE=1,NNODE
    LNODE=LNODS(IELEM,INODE)
    DO 11 IDIME=1,NDIME
      ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
    11 CONTINUE
  10 CONTINUE

C
  INITIALIZE ELEMENT MATRICES [HE],
  [GE] AND [BOE]
  C
  DO 20 IEVAB=1,NNODE
    DO 21 JEVAB=1,NNODE
      HE(IEVAB,JEVAB)=0.0
      GE(IEVAB,JEVAB)=0.0
      BOE(IEVAB,JEVAB)=0.0
    21 CONTINUE
  20 CONTINUE

C
  KGASP=0

C
  ENTER LOOPS FOR AREA NUMERICAL INTEGRATION
  C
  DO 50 IGAUS=1,NGAUS
    DO 51 JGAUS=1,NGAUS
      KGASP=KGASP+1
      EXISP=POSGP(IGAUS)
      ETASP=POSGP(JGAUS)

C
    EVALUATE THE SHAPE FUNCTIONS AND JACOBIAN
    C
    CALL SF2(EXISP,ETASP,SHAPE,DERIV)
    CALL JACOB2(IELEM,KGASP,GPCOD,CARTD,ELCOD,SHAPE,DERIV,DJACB)

C
    DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)

C
    EVALUATE [B1] AND [B2] MATRICES

```

```

C      CALL BMAT(B1,B2,CARTD)
C
C      CALCULATE MATRIX [HE]
C
      DO 30 IEVAB=1,NNODE
      DO 31 JEVAB=1,NNODE
      HE(IEVAB,JEVAB)=HE(IEVAB,JEVAB)
      &+(B1(IEVAB)*B1(JEVAB)
      &+B2(IEVAB)*B2(JEVAB))*DAREA
31 CONTINUE
30 CONTINUE
C
C      CALCULATE MATRIX [GE]
C
      DO 40 IEVAB=1,NNODE
      DO 41 JEVAB=1,NNODE
      GE(IEVAB,JEVAB)=GE(IEVAB,JEVAB)+SHAPE(IEVAB)*SHAPE(JEVAB)*DAREA
41 CONTINUE
40 CONTINUE
51 CONTINUE
50 CONTINUE
C
      IF(INDEX.EQ.0) GO TO 111
      IF(INDEX.NE.1) GO TO 61
      LENG1(IELEM)=SQRT((ELCOD(1,2)-ELCOD(1,1))**2
      &+(ELCOD(2,2)-ELCOD(2,1))**2)
      GO TO 70
61 IF(INDEX.NE.2) GO TO 62
      LENG1(IELEM)=SQRT((ELCOD(1,2)-ELCOD(1,1))**2
      &+(ELCOD(2,2)-ELCOD(2,1))**2)
      LENG2(IELEM)=SQRT((ELCOD(1,3)-ELCOD(1,2))**2
      &+(ELCOD(2,3)-ELCOD(2,2))**2)
      GO TO 71
62 IF(INDEX.NE.3) GO TO 63
      LENG1(IELEM)=SQRT((ELCOD(1,3)-ELCOD(1,2))**2
      &+(ELCOD(2,3)-ELCOD(2,2))**2)
      GO TO 72
63 IF(INDEX.NE.4) GO TO 64
      LENG1(IELEM)=SQRT((ELCOD(1,3)-ELCOD(1,2))**2
      &+(ELCOD(2,3)-ELCOD(2,2))**2)
      LENG2(IELEM)=SQRT((ELCOD(1,4)-ELCOD(1,3))**2
      &+(ELCOD(2,4)-ELCOD(2,3))**2)
      GO TO 73
64 IF(INDEX.NE.5) GO TO 65
      LENG1(IELEM)=SQRT((ELCOD(1,4)-ELCOD(1,3))**2
      &+(ELCOD(2,4)-ELCOD(2,3))**2)
      GO TO 74
65 IF(INDEX.NE.6) GO TO 66
      LENG1(IELEM)=SQRT((ELCOD(1,4)-ELCOD(1,3))**2
      &+(ELCOD(2,4)-ELCOD(2,3))**2)
      LENG2(IELEM)=SQRT((ELCOD(1,1)-ELCOD(1,4))**2
      &+(ELCOD(2,1)-ELCOD(2,3))**2)
      GO TO 75
66 IF(INDEX.NE.7) GO TO 67
      LENG1(IELEM)=SQRT((ELCOD(1,1)-ELCOD(1,4))**2
      &+(ELCOD(2,1)-ELCOD(2,4))**2)
      GO TO 76
67 LENG1(IELEM)=SQRT((ELCOD(1,1)-ELCOD(1,4))**2
      &+(ELCOD(2,1)-ELCOD(2,4))**2)

```

```

      LENG2 (IELEM) = SQRT ( (ELCOD (1, 2) - ELCOD (1, 1)) ** 2
& + (ELCOD (2, 2) - ELCOD (2, 1)) ** 2)
      GO TO 77
70 BOE (1, 1) = LENG1 (IELEM) / 3
   BOE (1, 2) = BOE (1, 1) / 2
   BOE (2, 1) = BOE (1, 2)
   BOE (2, 2) = BOE (1, 1)
   GO TO 111
71 BOE (1, 1) = LENG1 (IELEM) / 3
   BOE (1, 2) = LENG1 (IELEM) / 6
   BOE (2, 1) = BOE (1, 2)
   BOE (2, 2) = (LENG1 (IELEM) + LENG2 (IELEM)) / 3
   BOE (2, 3) = LENG2 (IELEM) / 6
   BOE (3, 2) = BOE (2, 3)
   BOE (3, 3) = LENG2 (IELEM) / 3
   GO TO 111
72 BOE (2, 2) = LENG1 (IELEM) / 3
   BOE (2, 3) = LENG1 (IELEM) / 6
   BOE (3, 2) = BOE (2, 3)
   BOE (3, 3) = BOE (2, 2)
   GO TO 111
73 BOE (2, 4) = LENG1 (IELEM) / 3
   BOE (2, 3) = LENG1 (IELEM) / 6
   BOE (3, 2) = BOE (2, 3)
   BOE (3, 3) = (LENG1 (IELEM) + LENG2 (IELEM)) / 3
   BOE (3, 4) = LENG2 (IELEM) / 6
   BOE (4, 3) = BOE (3, 4)
   BOE (4, 4) = LENG2 (IELEM) / 3
   GO TO 111
74 BOE (3, 3) = LENG1 (IELEM) / 3
   BOE (3, 4) = LENG1 (IELEM) / 6
   BOE (4, 3) = BOE (3, 4)
   BOE (4, 4) = BOE (3, 3)
   GO TO 111
75 BOE (1, 1) = LENG2 (IELEM) / 3
   BOE (1, 4) = LENG2 (IELEM) / 6
   BOE (4, 1) = BOE (1, 4)
   BOE (3, 3) = LENG1 (IELEM) / 3
   BOE (3, 4) = LENG1 (IELEM) / 6
   BOE (4, 3) = BOE (3, 4)
   BOE (4, 4) = (LENG1 (IELEM) + LENG2 (IELEM)) / 3
   GO TO 111
76 BOE (1, 1) = LENG1 (IELEM) / 3
   BOE (1, 4) = LENG1 (IELEM) / 6
   BOE (4, 1) = BOE (1, 4)
   BOE (4, 4) = BOE (1, 1)
   GO TO 111
77 BOE (1, 1) = (LENG1 (IELEM) + LENG2 (IELEM)) / 3
   BOE (1, 2) = LENG2 (IELEM) / 6
   BOE (2, 1) = BOE (1, 2)
   BOE (2, 2) = LENG2 (IELEM) / 3
   BOE (1, 4) = LENG1 (IELEM) / 6
   BOE (4, 1) = BOE (1, 4)
   BOE (4, 4) = LENG1 (IELEM) / 3
111 DO 110 I=1, NNODE
    WRITE (5, *) (HE (I, J), J=1, NNODE)
    WRITE (6, *) (GE (I, J), J=1, NNODE)
    WRITE (7, *) (BOE (I, J), J=1, NNODE)
110 CONTINUE
100 CONTINUE

```

```

C
C   ASSEMBLYING ELEMENT MATRICES INTO GLOBAL MATRICES
C
C   INITIALIZE GLOBAL MATRICES [H], [G], [BO]
C
DO 120 ISVAB=1,NPOIN+NT
DO 130 JSVAB=1,NPOIN+NT
HT(ISVAB,JSVAB)=0.0
GT(ISVAB,JSVAB)=0.0
BOT(ISVAB,JSVAB)=0.0
BIT(ISVAB,JSVAB)=0.0
130 CONTINUE
120 CONTINUE
REWIND(5)
REWIND(6)
REWIND(7)
DO 140 IELEM=1,NELEM
DO 150 INODE=1,NNODE
READ(5,*)(HE(INODE,JNODE),JNODE=1,NNODE)
READ(6,*)(GE(INODE,JNODE),JNODE=1,NNODE)
READ(7,*)(BOE(INODE,JNODE),JNODE=1,NNODE)
I=LNODS(IELEM,INODE)
DO 160 JNODE=1,NNODE
J=LNODS(IELEM,JNODE)
HT(I,J)=HT(I,J)+HE(INODE,JNODE)
GT(I,J)=GT(I,J)+GE(INODE,JNODE)
BOT(I,J)=BOT(I,J)+BOE(INODE,JNODE)
160 CONTINUE
150 CONTINUE
140 CONTINUE
C
DO 198 I=1,NPOIN+NT
DO 199 J=1,NPOIN+NT
BOT(I,J)=T1*T2*BOT(I,J)
199 CONTINUE
198 CONTINUE
C
DO 170 I=1,NPOIN+NT
WRITE(8,1000)(HT(I,J),J=1,NPOIN+NT)
1000 FORMAT(8(E15.7))
170 CONTINUE
DO 180 I=1,NPOIN+NT
WRITE(8,1000)(GT(I,J),J=1,NPOIN+NT)
180 CONTINUE
DO 190 I=1,NPOIN+NT
WRITE(8,1000)(BOT(I,J),J=1,NPOIN+NT)
190 CONTINUE
C
CALL EXTRA(H,G,BO,BI,HT,GT,BOT,BIT,NTOP,NT,NPOIN,NM)
C
DO 171 I=1,NPOIN
WRITE(8,1001)(H(I,J),J=1,NPOIN)
1001 FORMAT(6(E15.7))
171 CONTINUE
DO 181 I=1,NPOIN
WRITE(8,1000)(G(I,J),J=1,NPOIN)
181 CONTINUE
DO 191 I=1,NPOIN
WRITE(8,1000)(BO(I,J),J=1,NPOIN)
191 CONTINUE

```

```

      RETURN
      END
C
C -----
C
C SUBROUTINE INFUT2
C *****
C
C SUBROUTINE INPUT2 (NPOIN, LNODS, COORD, EINDEK, NTOP, NCM, NM, NT)
C
C COMMON/SET/ NELEM, NNODE, NDIME, NGAUS, NGASP
C
C REAL COORD (NCM, NDIME)
C INTEGER EINDEK (NM)
C INTEGER LNODS (NM, NNODE)
C INTEGER NTOP (NM)
C
C
C N=NPOIN+NT
C
C DO 1 I=1, N
C DO 2 J=1, NDIME
C COORD (I, J)=0.0
2 CONTINUE
1 CONTINUE
C
C READ GLOBAL COORDINATES OF EACH NODE
C
C DO 10 I=1, N
C READ (4, *) (COORD (I, J), J=1, NDIME)
10 CONTINUE
C
C DO 11 I=1, NELEM
C DO 12 J=1, NNODE
C LNODS (I, J)=0.0
12 CONTINUE
11 CONTINUE
C
C READ TOPOLOGY MATRIX
C
C DO 20 I=1, NELEM
C READ (4, *) (LNODS (I, J), J=1, NNODE)
20 CONTINUE
C
C DO 13 I=1, NELEM
C EINDEK (I)=0.0
13 CONTINUE
C
C READ ELEMENT INDEX MATRIX
C
C READ (4, *) (EINDEK (I), I=1, NELEM)
C
C READ (4, *) (NTOP (I), I=1, NT)
C
C RETURN
C END
C -----
C

```

```

C      SUBROUTINE SF2(S,T,SHAPE,DERIV)
C      *****
COMMON/SET/ NELEM,NNODE,NDIME,NGAUS,NGASP
C
C      REAL SHAPE(NNODE),DERIV(NDIME,NNODE)
C
C      INITIALIZE MATRIX SHAPE
C
C      DO 5 I=1,NNODE
C      SHAPE(I)=0.0
5  CONTINUE
C
C      SHAPE FUNCTIONS
C
C      SHAPE(1)=((1-S)*(1-T))/4.0
C      SHAPE(2)=((1+S)*(1-T))/4.0
C      SHAPE(3)=((1+S)*(1+T))/4.0
C      SHAPE(4)=((1-S)*(1+T))/4.0
C
C      SHAPE FUNCTION DERIVATIVES
C
C      INITIALIZE MATRIX DERIV
C
C      DO 10 I=1,NDIME
C      DO 20 J=1,NNODE
C      DERIV(I,J)=0.0
20  CONTINUE
10  CONTINUE
C
C      DERIV(1,1)=(T-1)/4.0
C      DERIV(1,2)=-DERIV(1,1)
C      DERIV(1,3)=(T+1)/4.0
C      DERIV(1,4)=-DERIV(1,3)
C      DERIV(2,1)=(S-1)/4.0
C      DERIV(2,2)=-DERIV(2,1)
C      DERIV(2,3)=-DERIV(2,2)
C      DERIV(2,4)=-DERIV(2,1)
C      RETURN
C      END
C
C      -----
C
C      SUBROUTINE JACOB2(IELEM,KGASP,GPCOD,CARTD,ELCOD,SHAPE,
C      &DERIV,DJACB)
C      *****
COMMON/SET/ NELEM,NNODE,NDIME,NGAUS,NGASP
C
C      REAL XJACM(2,2),XJACI(2,2)
C      REAL GPCOD(NDIME,NGASP),CARTD(NDIME,NNODE)
C      REAL ELCOD(NDIME,NNODE),SHAPE(NNODE)
C      REAL DERIV(NDIME,NNODE)
C
C      COORDINATE OF SAMPLING POINTS
C
C      DO 10 IDIME=1,NDIME
C      GPCOD(IDIME,KGASP)=0.0
C      DO 11 INODE=1,NNODE
C      GPCOD(IDIME,KGASP)=GPCOD(IDIME,KGASP)
C      &+ELCOD(IDIME,INODE)*SHAPE(INODE)
11  CONTINUE

```



```

10 CONTINUE
C
C   CREATE JACOBIAN MATRIX XJACM
C
      DO 20 IDIME=1,NDIME
      DO 21 JDIME=1,NDIME
      XJACM(IDIME,JDIME)=0.0
      DO 22 INODE=1,NNODE
      XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME)
      &+DERIV(IDIME,INODE)
      &*ELCOD(JDIME,INODE)
22 CONTINUE
21 CONTINUE
20 CONTINUE

C
C   CALCULATE DETERMINANT AND INVERSE OF JACOBIAN MATRIX
C
      DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1)
      IF(DJACB.GT.0.0) GO TO 30
      WRITE(6,900) IELEM
      STOP
30 XJACI(1,1)=XJACM(2,2)/DJACB
   XJACI(2,2)=XJACM(1,1)/DJACB
   XJACI(1,2)=-XJACM(1,2)/DJACB
   XJACI(2,1)=-XJACM(2,1)/DJACB

C
C   CALCULATE CARTESIAN DERIVATIVES OF SHAPE FUNCTIONS
C
      DO 40 IDIME=1,NDIME
      DO 41 INODE=1,NNODE
      CARTD(IDIME,INODE)=0.0
      DO 42 JDIME=1,NDIME
      CARTD(IDIME,INODE)=CARTD(IDIME,INODE)+XJACI(IDIME,JDIME)
      &*DERIV(JDIME,INODE)
42 CONTINUE
41 CONTINUE
40 CONTINUE
900 FORMAT(//,24HPROGRAM HALTED IN JACOB2,
&/,11X,22H ZERO OR NEGATIVE AREA,/,
&10X,15H ELEMENT NUMBER,15)
      RETURN
      END

C
C   -----
C
C   SUBROUTINE GAUSSQ(POSGP,WEIGP)
C   *****
C
C   COMMON/SET/ NELEM,NNODE,NDIME,NGAUS,NGASP
C
C   REAL POSGP(NGAUS),WEIGP(NGAUS)
C
C
      POSGP(1)=-0.577350269189626
      WEIGP(1)=1.0
      KGAUS=NGAUS/2
      DO 10 IGASH=1,KGAUS
      JGASH=NGAUS+1-IGASH
      POSGP(JGASH)=-POSGP(IGASH)

```

```

WEIGP (JGASH) =WEIGP (IGASH)
10 CONTINUE
RETURN
END

C
C SUBROUTINE BMAT
C
C SUBROUTINE BMAT (B1, B2, CARTD)
C
C COMMON/SET/ NELEM, NNODE, NDIME, NGAUS, NGASP
C
C REAL B1 (NNODE), B2 (NNODE), CARTD (NDIME, NNODE)
C
C
C INITIALIZE MATRICES B1 & B2
C
C DO 5 INODE=1, NNODE
C   B1 (INODE) =0.0
C   B2 (INODE) =0.0
5 CONTINUE
C
C DO 10 INODE=1, NNODE
C   B1 (INODE) =CARTD (1, INODE)
C   B2 (INODE) =CARTD (2, INODE)
10 CONTINUE
RETURN
END

C
C -----
C
C SUBROUTINE EXTRA (H, G, BO, BI, HT, GT, BOT, BIT, NTOP, NT, NPOIN, NM)
C *****
C
C SUBROUTINE EXTRA APPLIES B.C.'S ON THE UPPER EDGE
C AT THE FAR END TO MATRIX [H] REQUIRED FOR
C THE EIGENVALUE PROBLEM
C
C COMMON/SET/ NELEM, NNODE, NDIME, NGAUS, NGASP
C
C REAL H (NM, NM), HT (NM, NM)
C REAL G (NM, NM), GT (NM, NM)
C REAL BO (NM, NM), BOT (NM, NM)
C REAL BI (NM, NM), BIT (NM, NM)
C DIMENSION NTOP (NM), INEW (80)
C
C DO 10 I=1, NPOIN
C   DO 20 J=I, NPOIN
C     H (I, J) =0.0
C     G (I, J) =0.0
C     BO (I, J) =0.0
C     BI (I, J) =0.0
20 CONTINUE
10 CONTINUE
C
C DO 30 I=1, NPOIN+NT
C   INEW (I) =I
30 CONTINUE
C
C DO 40 K=1, NT
C   L=NTOP (K)

```

```

      M=L+NTOP(1)
      DO 50 I=1,NPOIN+NT
      IF(I.EQ.M.OR.M.GT.NPOIN+NT) GO TO 50
      IF(I.GT.L.AND.I.LT.M) THEN
        INEW(I)=INEW(I)-K
      ELSE IF(I.EQ.L) THEN
        INEW(I)=0
      END IF
50  CONTINUE
40  CONTINUE
C
      DO 60 I=1,NPOIN+NT
      DO 70 J=1,NPOIN+NT
      K1=INEW(I)
      K2=INEW(J)
      IF(K1.EQ.0.OR.K2.EQ.0) GO TO 70
      H(K1,K2)=HT(I,J)
      G(K1,K2)=GT(I,J)
      BO(K1,K2)=BOT(I,J)
      BI(K1,K2)=BIT(I,J)
70  CONTINUE
60  CONTINUE
C
      RETURN
      END
C
C
C
C
      SUBROUTINE EIGENR (AFR,ND,ME,CONST,CONST1,CONST2,AKR,AKI
1  ,IMP,FLAG)
C
      *****
      DOUBLE PRECISION CONST,CONST1,CONST2
      REAL AFR(ND,ND),AKR(ND),AKI(ND)
      REAL ALFR(80),A(80,80),F(80,80)
      LOGICAL FLAG
      OPEN(4,FORM='FORMATTED',STATUS='OLD',FILE='IN1')
      OPEN(5,FORM='FORMATTED',STATUS='NEW',FILE='HE1')
      OPEN(6,FORM='FORMATTED',STATUS='NEW',FILE='GE1')
      OPEN(7,FORM='FORMATTED',STATUS='NEW',FILE='BEO1')
      OPEN(8,FORM='FORMATTED',STATUS='NEW',FILE='TOT')
      OPEN(9,FORM='FORMATTED',STATUS='NEW',FILE='IB3D')
      DATA IWRITE/9/
C
      NCM=100
      READ(4,*)NT
      DO 10 I=1,ME
      AKR(I)=0
      AKI(I)=0
10  CONTINUE
      T=SNGL(CONST)
      T1=SNGL(CONST1)
      T2=SNGL(CONST2)
      FLAG=.FALSE.
      CALL SETER(ME,ND,NCM,NT,A,F)
      CALL JACOBI(A,F,ALFR,AFR,ME,ND)
      DO 30 I=1,ME
      TEMP=ALFR(I)-T*T
      TX=TEMP
      IF(TEMP.LT.0) THEN
      FLAG=.TRUE.

```

```

      AKI(I)=SQRT(-TX)
      ELSE IF(TEMP.GT.0) THEN
      AKR(I)=SQRT(TX)
      ELSE
      WRITE(IMP,100)
100  FORMAT(1X,'RESONANCE HAS TAKEN PLACE')
      STOP
      END IF
30  CONTINUE
      RETURN
      END

```

C
C
C
C

```

SUBROUTINE SETER(NPOIN,NM,NCM,NT,H,G)

```

```

*****

```

C
C
C

```

SUBROUTINE SETER
IS MODIFIED VERSION OF THE SUBROUTINE SETE
IT FORMS MATRICES [HE] AND [GE]
REQUIRED TO FORMULATE AN EIGENVALUE PROBLEM
AT A FAR END OF 3-D RESERVOIR
MATRICES [HE] AND [GE] ARE FORMED FOR EACH ELEMENT
USING 2D FINITE ELEMENT FORMULATION
TWO-DIMENSIONAL 4-NODE QUADRILATERAL ELEMENTS ARE USED

```

C
C
C

```

DATE: DECEMBER 7, 1987
      MODIFIED ON JANUARY 29, 1988

```

C
C
C

```

NELEM = NUMBER OF ELEMENTS
NM = ND MAX DIMENSION OF THE EIGENVECTOR MATRIX
NPOIN = N NUMBER OF NODAL POINTS (NODES) - NUMBER OF TOP NODES
NT = NUMBER OF TOP NODES
NNODE = 4, NUMBER OF NODES IN AN ELEMENT
NDIME = 2, NUMBER OF COORDINATE COMPONENTS REQUIRED TO DEFINE EACH NODE
NGAUS = 2, THE ORDER OF GAUSSIAN NUMERICAL INTEGRATION RULE TO BE EMPLOYED
NGASP = 4, NUMBER OF GAUSSIAN INTEGRATION POINTS

```

C
C
C

```

LNODS(NELEM,NNODE) = ELEMENT TOPOLOGY - CONNECTIVITY MATRIX
COORD(NPOIN,NDIME) = EACH NODE MUST BE REFERRED TO A GLOBAL COORDINATE
SYSTEM

```

C
C
C

```

POSGP(NGAUS) = POSITION OF GAUSSIAN POINT IN THE LOCAL COORDINATE SYSTEM
WEIGP(NGAUS) = GAUSSIAN WEIGHT ASSOCIATED TO EACH GAUSSIAN POINT
ELCOLD(NDIME,NNODE) = LOCAL ARRAY OF NODAL CARTESIAN COORDINATES
OF AN ELEMENT

```

C
C
C

```

SHAPE(NNODE) = SHAPE FUNCTIONS ASSOCIATED WITH EACH NODE OF AN ELEMENT
IN LOCAL COORDINATES SAMPLED IN A POINT WITHIN THE ELEMENT

```

C
C
C

```

DERIV(NDIME,NNODE) = SHAPE FUNCTION DERIVATIVE AT SAMPLING POINT WITHIN
THE ELEMENT

```

C
C
C

```

GPCOD(NDIME,NGASP) = LOCAL ARRAY OF CARTESIAN COORDINATES OF THE GAUSS
POINTS OF AN ELEMENT

```

C
C
C

```

CARTD(NDIME,NNODE) = CARTESIAN SHAPE FUNCTION DERIVATIVES ASSOCIATED WITH
THE NODES OF AN ELEMENT SAMPLED AT ANY POINT
WITHIN THE ELEMENT

```

C
C
C

```

HE(NNODE,NNODE) = REAL VALUED MATRIX TO BE FORMED AT THE ELEMENT LEVEL
GE(NNODE,NNODE) = REAL VALUED MATRIX TO BE FORMED AT THE ELEMENT LEVEL

```

```

C
C
C
COMMON/SET/ NELEM,NNODE,NDIME,NGAUS,NGASP
C
REAL COORD(100,2)
REAL POSGP(2),WEIGP(2),SHAPE(4),DERIV(2,4)
REAL ELCOD(2,4),GPCOD(2,4),CARTD(2,4)
REAL HE(4,4),GE(4,4),B1(4),B2(4)
REAL LENG1(80),LENG2(80)
REAL HT(80,80),GT(80,80)
REAL H(NM,NM),G(NM,NM)
INTEGER LNODS(80,4)
INTEGER NTOP(80)
C
C
C
READ(4,*)NELEM,NNODE,NDIME,NGAUS,NGASP
CALL INPUT2R(NPOIN,LNODS,COORD,NTOP,NCM,NM,NT)
C
CALL GAUSSQ(POSGP,WEIGP)
C
ENTER LOOP OVER EACH ELEMENT
C
DO 100 IELEM=1,NELEM
C
EVALUATE COORDINATES OF THE ELEMENT
C
NODAL POINTS
C
DO 10 INODE=1,NNODE
LNODE=LNODS(IELEM,INODE)
DO 11 IDIME=1,NDIME
ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
11 CONTINUE
10 CONTINUE
C
C
INITIALIZE ELEMENT MATRICES [HE],
[GE] AND [BOE]
C
DO 20 IEVAB=1,NNODE
DO 21 JEVAB=1,NNODE
HE(IEVAB,JEVAB)=0.0
GE(IEVAB,JEVAB)=0.0
21 CONTINUE
20 CONTINUE
C
KGASP=0
C
ENTER LOOPS FOR AREA NUMERICAL INTEGRATION
C
DO 50 IGAUS=1,NGAUS
DO 51 JGAUS=1,NGAUS
KGASP=KGASP+1
EXISP=POSGP(IGAUS)
ETASP=POSGP(JGAUS)
C
C
EVALUATE THE SHAPE FUNCTIONS AND JACOBIAN
C
CALL SF2(EXISP,ETASP,SHAPE,DERIV)
CALL JACOB2(IELEM,KGASP,GPCOD,CARTD,ELCOD,SHAPE,DERIV,DJACB)

```

```

C      DAREA=DJACB*WEIGP (IGAUS) *WEIGP (JGAUS)
C
C      EVALUATE [B1] AND [B2] MATRICES
C
C      CALL BMAT(B1,B2,CARTD)
C
C      CALCULATE MATRIX [HE]
C
      DO 30 IEVAB=1,NNODE
      DO 31 JEVAB=1,NNODE
      HE (IEVAB,JEVAB)=HE (IEVAB,JEVAB)
      &+(B1 (IEVAB)*B1 (JEVAB)
      &+B2 (IEVAB)*B2 (JEVAB)) *DAREA
31 CONTINUE
30 CONTINUE
C
C      CALCULATE MATRIX [GE]
C
      DO 40 IEVAB=1,NNODE
      DO 41 JEVAB=1,NNODE
      GE (IEVAB,JEVAB)=GE (IEVAB,JEVAB)+SHAPE (IEVAB) *SHAPE (JEVAB) *DAREA
41 CONTINUE
40 CONTINUE
51 CONTINUE
50 CONTINUE
C
      DO 110 I=1,NNODE
      WRITE (5,*) (HE (I,J),J=1,NNODE)
      WRITE (6,*) (GE (I,J),J=1,NNODE)
110 CONTINUE
100 CONTINUE
C
C      ASSEMBLYING ELEMENT MATRICES INTO GLOBAL MATRICES
C
C      INITIALIZE GLOBAL MATRICES [H], [G], [BO]
C
      DO 120 ISVAB=1,NPOIN+NT
      DO 130 JSVAB=1,NPOIN+NT
      HT (ISVAB,JSVAB)=0.0
      GT (ISVAB,JSVAB)=0.0
130 CONTINUE
120 CONTINUE
      REWIND (5)
      REWIND (6)
      DO 140 IELEM=1,NELEM
      DO 150 INODE=1,NNODE
      READ (5,*) (HE (INODE,JNODE),JNODE=1,NNODE)
      READ (6,*) (GE (INODE,JNODE),JNODE=1,NNODE)
      I=LNODS (IELEM,INODE)
      DO 160 JNODE=1,NNODE
      J=LNODS (IELEM,JNODE)
      HT (I,J)=HT (I,J)+HE (INODE,JNODE)
      GT (I,J)=GT (I,J)+GE (INODE,JNODE)
160 CONTINUE
150 CONTINUE
140 CONTINUE
C
C      DO 170 I=1,NPOIN+NT

```

```

      WRITE(8,1000) (HT(I,J),J=1,NPOIN+NT)
1000  FORMAT(8(E15.7))
      170  CONTINUE
      DO 180 I=1,NPOIN+NT
      WRITE(8,1000) (GT(I,J),J=1,NPOIN+NT)
180  CONTINUE
C
      CALL EXTRAR(H,G,HT,GT,NTOP,NT,NPOIN,NM)
C
      DO 171 I=1,NPOIN
      WRITE(8,1001) (H(I,J),J=1,NPOIN)
1001  FORMAT(6(E15.7))
      171  CONTINUE
      DO 181 I=1,NPOIN
      WRITE(8,1000) (G(I,J),J=1,NPOIN)
181  CONTINUE
      RETURN
      END
C
C -----
C
C  SUBROUTINE INPUT2R
C  *****
C
C  SUBROUTINE INPUT2R(NPOIN, LNODS, COORD, NTOP, NCM, NM, NT)
C
C  COMMON/SET/ NELEM, NNODE, NDIME, NGAUS, NGASP
C
C  REAL COORD(NCM,NDIME)
C  INTEGER LNODS(NM,NNODE)
C  INTEGER NTOP(NM)
C
C
C  N=NPOIN+NT
C
C  DO 1 I=1,N
C  DO 2 J=1,NDIME
C  COORD(I,J)=0.0
2  CONTINUE
1  CONTINUE
C
C  READ GLOBAL COORDINATES OF EACH NODE
C
C
C  DO 10 I=1,N
C  READ(4,*) (COORD(I,J),J=1,NDIME)
10  CONTINUE
C
C  DO 11 I=1,NELEM
C  DO 12 J=1,NNODE
C  LNODS(I,J)=0.0
12  CONTINUE
11  CONTINUE
C
C  READ TOPOLOGY MATRIX
C
C  DO 20 I=1,NELEM
C  READ(4,*) (LNODS(I,J),J=1,NNODE)
20  CONTINUE
C

```

```

C
C   READ NTOP MATRIX WITH NUMBERS OF TOP NODES
C
C   READ (4, *) (NTOP (I), I=1, NT)
C
C   RETURN
C   END
C
C   -----
C
C   SUBROUTINE EXTRAR (H, G, HT, GT, NTOP, NT, NPOIN, NM)
C   *****
C
C   SUBROUTINE EXTRA APPLIES B.C.'S ON THE UPPER EDGE
C   AT THE FAR END TO MATRIX [H] REQUIRED FOR
C   THE EIGENVALUE PROBLEM
C
C   COMMON/SET/ NELEM, NNODE, NDIME, NGAUS, NGASP
C
C   REAL H (NM, NM), HT (NM, NM)
C   REAL G (NM, NM), GT (NM, NM)
C   DIMENSION NTOP (NM), INEW (80)
C
C   DO 10 I=1, NPOIN
C   DO 20 J=1, NPOIN
C   H(I, J)=0.0
C   G(I, J)=0.0
C   20 CONTINUE
C   10 CONTINUE
C
C   DO 30 I=1, NPOIN+NT
C   INEW(I)=I
C   30 CONTINUE
C
C   DO 40 K=1, NT
C   L=NTOP (K)
C   M=L+NTOP (1)
C   DO 50 I=1, NPOIN+NT
C   IF (I.EQ.M.OR.M.GT.NPOIN+NT) GO TO 50
C   IF (I.GT.L.AND.I.LT.M) THEN
C   INEW(I)=INEW(I)-K
C   ELSE IF (I.EQ.L) THEN
C   INEW(I)=0
C   END IF
C   50 CONTINUE
C   40 CONTINUE
C
C   DO 60 I=1, NPOIN+NT
C   DO 70 J=1, NPOIN+NT
C   K1=INEW(I)
C   K2=INEW(J)
C   IF (K1.EQ.0.OR.K2.EQ.0) GO TO 70
C   H(K1, K2)=HT (I, J)
C   G(K1, K2)=GT (I, J)
C   70 CONTINUE
C   60 CONTINUE
C
C   RETURN
C   END
C

```



```

C -----
C
C SUBROUTINE FMAT(X,Y,Z,XM,YM,ZM,K,G,H,HOLD,FID,FIB,DFI,N0,N1)
C *****
C
C TO COMPUTE G & H MATRICES & FORM THE SYSTEM AX=F
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
C COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C
C DOUBLE PRECISION X(1),Y(1),Z(1),G(NY,1),H(NY,1)
C DOUBLE PRECISION FID(1),FIB(1),DFI(1),HOLD(NY,1)
C DOUBLE PRECISION XM(1),YM(1),ZM(1)
C DIMENSION K(4,1)
C
C NX=2*N
C DO 10 I=1,NX
C   DFI(I)=0.0
C   DO 11 J=1,NX
C     G(I,J)=0.0
11  H(I,J)=0.0
10  CONTINUE
C
C COMPUTE MIDPOINT COORDINATES
C DO 90 I=1,N
C   M1=K(1,I)
C   M2=K(2,I)
C   M3=K(3,I)
C   M4=K(4,I)
C   XM(I)=(X(M1)+X(M2)+X(M3)+X(M4))/4
C   YM(I)=(Y(M1)+Y(M2)+Y(M3)+Y(M4))/4
C   ZM(I)=(Z(M1)+Z(M2)+Z(M3)+Z(M4))/4
C
C COMPUTE G AND H MATRICES
C DO 50 J=1,N
C   M1=K(1,J)
C   M2=K(2,J)
C   M3=K(3,J)
C   M4=K(4,J)
C   IF(I-J)20,30,20
20  CALL INTE(XM(I),YM(I),ZM(I),X(M1),Y(M1),Z(M1),X(M2),Y(M2),Z(M2)
C   *,X(M3),Y(M3),Z(M3),X(M4),Y(M4),Z(M4),H(I,J),H(I+N,J)
C   *,G(I,J),G(I+N,J))
C   H(I,J+N)=-H(I+N,J)
C   H(I+N,J+N)=H(I,J)
C   GO TO 40
30  CALL INLO(XM(I),YM(I),ZM(I),X(M1),Y(M1),Z(M1),X(M2),Y(M2),Z(M2)
C   *,X(M3),Y(M3),Z(M3),X(M4),Y(M4),Z(M4),H(I,J),H(I+N,J)
C   *,G(I,J),G(I+N,J))
C   H(I,J+N)=0.0
C   H(I+N,J+N)=-0.5
40  G(I,J+N)=-G(I+N,J)
C   G(I+N,J+N)=G(I,J)
50  CONTINUE
90  CONTINUE
C
C DO 151 I=1,NX
C   write(imp,102)
C   102 format(//)

```

```

*      WRITE(IMP,101) (H(I,J),J=1,NX)
* 151 CONTINUE
*      DO 152 I=1,NX
*          write(imp,102)
*          WRITE(IMP,101) (G(I,J),J=1,NX)
* 152 CONTINUE
* 101 format(6e13.5)
C      ARRANGE THE SYSTEM OF EQUATIONS
      DO 60 J=1,N0
      DO 70 I=1,N
      CH1=G(I,J)
      G(I,J)=-H(I,J)
      H(I,J)=-CH1
      CH2=G(I+N,J)
      G(I+N,J)=-H(I+N,J)
      H(I+N,J)=-CH2
      G(I,J+N)=-H(I,J+N)
      G(I+N,J+N)=-H(I+N,J+N)
      H(I,J+N)=-H(I+N,J)
      H(I+N,J+N)=H(I,J)
      70 CONTINUE
      60 CONTINUE
C
C
      DO 160 I=1,NX
      DO 170 J=1,NX
      HOLD(I,J)=H(I,J)
      170 CONTINUE
      160 CONTINUE
C
      RETURN
      END
C
C
C      -----
C      SUBROUTINE INTE(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
*XD,YD,ZD,A,B,C,D)
C      *****
C
C      TO COMPUTE G & H MATRICES' OFF DIAGONAL ELEMENTS
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C
C      DOUBLE PRECISION E(4),F(4)
C      DATA E/-.861136D0,-.339981D0,.339981D0,.861136D0/
C      DATA F/.347855D0,.652154D0,.652145D0,.347855D0/
C      A=0
C      B=0
C      C=0
C      D=0
C      PI=3.1415962654
C      DO 100 IN=1,4
C      DO 101 IM=1,4
C      F01=(E(IM)-1)/4
C      F03=(1+E(IM))/4
C      F05=(E(IN)-1)/4
C      F07=(1+E(IN))/4
C      F11=4*F01*F05

```

```

F12=-4*F01*F07
F13=4*F03*F07
F14=-4*F03*F05
X=F11*XA+F12*XB+F13*XC+F14*XD
Y=F11*YA+F12*YB+F13*YC+F14*YD
Z=F11*ZA+F12*ZB+F13*ZC+F14*ZD
X1=F01*(XA-XB)+F03*(XC-XD)
Y1=F01*(YA-YB)+F03*(YC-YD)
Z1=F01*(ZA-ZB)+F03*(ZC-ZD)
X2=F05*(XA-XD)+F07*(XC-XB)
Y2=F05*(YA-YD)+F07*(YC-YB)
Z2=F05*(ZA-ZD)+F07*(ZC-ZB)
G1=Y1*Z2-Z1*Y2
G2=Z1*X2-X1*Z2
G3=X1*Y2-Y1*X2
G=DSQRT(G1*G1+G2*G2+G3*G3)
R=DSQRT((X-XP)**2+(Y-YP)**2+(Z-ZP)**2)
DRN=(G1*(X-XP)+G2*(Y-YP)+G3*(Z-ZP))/(R*G)
RB=CONST*R
RB1=DCOS(RB)
RB2=DSIN(RB)
A=A+DRN*G*F(IN)*F(IM)*(RB1+RB*RB2)/(4*R*R*PI)
B=B+DRN*G*F(IN)*F(IM)*(RB*RB1-RB2)/(4*R*R*PI)
C=C-RB1*G*F(IN)*F(IM)/(4*R*PI)
D=D+RB2*G*F(IN)*F(IM)/(4*R*PI)
101 CONTINUE
100 CONTINUE
RETURN
END

```

```

C-----
C      SUBROUTINE INLO(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
C      *XD,YD,ZD,A,B,C,D)
C      *****
C
C      TO COMPUTE G & H MATRICES' DIAGONAL ELEMENTS
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
C      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C
C      CALL INTL(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
C      *XD,YD,ZD,1.D0,-1.D0,1.D0,1.D0,C1,D1)
C      CALL INTL(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
C      *XD,YD,ZD,1.D0,1.D0,-1.D0,1.D0,C2,D2)
C      CALL INTL(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
C      *XD,YD,ZD,-1.D0,1.D0,-1.D0,-1.D0,C3,D3)
C      CALL INTL(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
C      *XD,YD,ZD,-1.D0,-1.D0,1.D0,-1.D0,C4,D4)
C      A=-.5
C      B=0.
C      C=C1+C2+C3+C4
C      D=D1+D2+D3+D4
C      RETURN
C      END

```

```

C-----
C      SUBROUTINE INTL(XP,YP,ZP,XA,YA,ZA,XB,YB,ZB,XC,YC,ZC,
C      *XD,YD,ZD,ZE11,ZE12,ZE21,ZE22,A,B)
C      *****
C

```

C TO COMPUTE G MATRIX'S DIAGONAL ELEMENTS

C

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2

C

```
DOUBLE PRECISION E(4),F(4)
DATA E/-.861136D0,-.339981D0,.339981D0,.861136D0/
DATA F/.347855D0,.652145D0,.652145D0,.347855D0/
A=0
B=0
PI=3.141592654
DO 100 IN=1,4
DO 110 IM=1,4
C1=1+E(IM)
D1=1+E(IN)
D2=1-E(IN)
X21=(D2*ZE11+D1*ZE12)/4
Y21=(D2*ZE21+D1*ZE22)/4
X22=C1*(ZE12-ZE11)/4
Y22=C1*(ZE22-ZE21)/4
G20=DABS(X21*Y22-X22*Y21)
ZE1=C1*(D2*ZE11+D1*ZE12)/4
ZE2=C1*(D2*ZE21+D1*ZE22)/4
A1=1+ZE1
A2=1-ZE1
B1=1+ZE2
B2=1-ZE2
X=(A1*B2*XA+A1*B1*XB+A2*B1*XC+A2*B2*XD)/4
Y=(A1*B2*YA+A1*B1*YB+A2*B1*YC+A2*B2*YD)/4
Z=(A1*B2*ZA+A1*B1*ZB+A2*B1*ZC+A2*B2*ZD)/4
X1=(B1*(XB-XC)+B2*(XA-XD))/4
Y1=(B1*(YB-YC)+B2*(YA-YD))/4
Z1=(B1*(ZB-ZC)+B2*(ZA-ZD))/4
X2=(A1*(XB-XA)+A2*(XC-XD))/4
Y2=(A1*(YB-YA)+A2*(YC-YD))/4
Z2=(A1*(ZB-ZA)+A2*(ZC-ZD))/4
G1=Y1*Z2-Y2*Z1
G2=Z1*X2-Z2*X1
G3=X1*Y2-X2*Y1
G10=DSQRT(G1*G1+G2*G2+G3*G3)
X0=X-XP
Y0=Y-YP
Z0=Z-ZP
R=DSQRT(X0*X0+Y0*Y0+Z0*Z0)
RB=CONST*R
A=A-DCOS(RB)*G10*G20*F(IN)*F(IM)/(4*R*PI)
110 B=B+DSIN(RB)*G10*G20*F(IN)*F(IM)/(4*R*PI)
100 CONTINUE
RETURN
END
```

C

C

C

```
-----
SUBROUTINE ADDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,
1 DFI,FID,F1B,DF,
```

C

C

```
*****
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
```

```

COMMON/ADD/HY,PI,ME,MF,CONST1,CONST2
REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
DOUBLE PRECISION H(NY,NY),G(NY,NY),T(340,80),DF(ND)
DOUBLE PRECISION FID(ND),FIB(NB),DFI(NY),HOLD(NY,NY)
DOUBLE PRECISION TX(340,80)
REAL UL(80,80),VB(80,80),VL(80,80),UB(80,80)
COMPLEX XKAP(80),PSI(80,80),TEMP1(80,80),TEMP2(80,80)
COMPLEX SUM5,DX(80),SUMT
REAL DXR(80),DXI(80)

```

```

C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C

```

```

      IF (ME.GT.0) THEN
        DO 20 J=1,ME
        DO 10 I=1,ME
          UL(I,J)=AFR(I,J)*AKR(J)
          VB(I,J)=AFI(I,J)*AKI(J)
          VL(I,J)=AFI(I,J)*AKR(J)
          UB(I,J)=AFR(I,J)*AKI(J)
10      CONTINUE
20      CONTINUE

```

```

C
      DO 40 I=1,ME
      DO 30 J=1,ME
        UL(I,J)=UL(I,J)-VB(I,J)
        VL(I,J)=VL(I,J)+UB(I,J)
30      CONTINUE
40      CONTINUE

```

```

C
      DO 41 I=1,N
      DO 31 J=1,ME
        T(I,J)=0.0
31      CONTINUE
41      CONTINUE

```

```

C
      DO 70 I=1,N
      DO 60 J=1,ME
        SUM1=0.
        SUM2=0.
        SUM3=0.
        SUM4=0.
        DO 50 K=1,ME
          KK=K+N2
          SUM1=SUM1+H(I,KK)*DBLE(AFR(K,J))
          SUM2=SUM2-H(I+N,KK)*DBLE(AFI(K,J))
          SUM3=SUM3+G(I,KK)*DBLE(UL(K,J))
          SUM4=SUM4-G(I+N,KK)*DBLE(VL(K,J))
50      CONTINUE
        T(I,J)=SUM1+SUM2+SUM3+SUM4
60      CONTINUE
70      CONTINUE

```

```

C
      DO 75 I=1,N
      DO 72 J=1,ME
        JJ=J+N2
        H(I,JJ)=T(I,J)

```

```

72 CONTINUE
75 CONTINUE
C
C IMAGINARY PART
C
C
      DO 76 I=1,N
      DO 73 J=1,ME
      TX(I,J)=0.0
73 CONTINUE
76 CONTINUE
C
      DO 120 I=1,N
      DO 110 J=1,ME
      SUM1=0.
      SUM2=0.
      SUM3=0.
      SUM4=0.
      DO 100 K=1,ME
      KK=K+N2
      SUM1=SUM1+H(I+N, KK)*DBLE (AFR(K, J) )
      SUM2=SUM2+H(I, KK)*DBLE (AFI(K, J) )
      SUM3=SUM3+G(I, KK)*DBLE (VL(K, J) )
      SUM4=SUM4+G(I+N, KK)*DBLE (UL(K, J) )
100 CONTINUE
      TX(I, J)=SUM1+SUM2+SUM3+SUM4
110 CONTINUE
120 CONTINUE
C
      DO 140 I=1,N
      DO 130 J=1,ME
      H(N+I, J+N2)=TX(I, J)
130 CONTINUE
140 CONTINUE
      END IF
C
      IF (CONST2.GT.1.E-08) THEN
      TEMP=CONST1*CONST2
      DO 90 I=1,N
      DO 80 J=N1+1,N2
      H(N+I, J)=H(N+I, J)+G(I, J)*TEMP
      H(I, J)=H(I, J)-G(N+I, J)*TEMP
80 CONTINUE
90 CONTINUE
      END IF
C
C FILL IN THE REST OF H MATRIX
C
C
      DO 160 I=1,N
      DO 150 J=1,N
      H(I, J+N)= -H(I+N, J)
      H(I+N, J+N)=H(I, J)
150 CONTINUE
160 CONTINUE
C
C FORM RIGHT HAND VECTOR
C
      DO 180 I=1,N
      DO 170 J=N0+1,N1

```

```

      DFI(I)=DFI(I)+G(I,J)*FID(J-N0)
      DFI(I+N)=DFI(I+N)+G(I+N,J)*FID(J-N0)
170  CONTINUE
180  CONTINUE
C
      DO 200 I=1,N
      DO 190 J=N1+1,N2
      DFI(I)=DFI(I)+G(I,J)*FIB(J-N1)
      DFI(I+N)=DFI(I+N)+G(I+N,J)*FIB(J-N1)
190  CONTINUE
200  CONTINUE
C
      IF (DF(1).GT.C.0) THEN
      DO 211 I=1,ME
      XKAP(I)=CMPLX(AKR(I),AKI(I))
      DO 212 J=1,ME
      PSI(I,J)=CMPLX(AFR(I,J),AFI(I,J))
212  CONTINUE
211  CONTINUE
      DO 210 I=1,ME
      DO 220 J=1,ME
      TEMP1(I,J)=PSI(I,J)/(XKAP(J)*XKAP(J))
220  CONTINUE
210  CONTINUE
      DO 230 I=1,ME
      DO 240 J=1,ME
      SUM5=(0.0,0.0)
      DO 250 K=1,ME
      SUM5=SUM5+TEMP1(J,K)*PSI(I,K)
250  CONTINUE
      TEMP2(J,I)=SUM5
240  CONTINUE
230  CONTINUE
      DO 260 I=1,ME
      SUMT=(0.0,0.0)
      DO 261 J=1,ME
      DE=DF(J)
      SUMT=SUMT+TEMP2(I,J)*SNGL(DE)
261  CONTINUE
      DX(I)=-SUMT
260  CONTINUE
      DO 270 I=1,ME
      DXR(I)=REAL(DX(I))
      DXI(I)=AIMAG(DX(I))
270  CONTINUE
      WRITE(IMP,101)(DXR(I),DXI(I),I=1,ME)
101  FORMAT(2(4X,E14.5))
      DO 280 I=1,N
      DO 290 J=1,ME
      K1=J-ME+N
      DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))-HOLD(I+N,K1)*DBLE(DXI(J))
      DFI(I+N)=DFI(I+N)+HOLD(I,K1)*DBLE(DXI(J))+HOLD(I+N,K1)
      &*DBLE(DXR(J))
290  CONTINUE
280  CONTINUE
      END IF
      RETURN
      END
C
C -----

```

```

C
C      SUBROUTINE SLNPD(A,B,D,NX,NY,IMP)
C      *****
C
C      SOLUTION OF LINEAR SYSTEMS OF EQUATIONS
C      BY THE GAUSS ELIMINATION METHOD PROVIDING
C      FOR INTERCHANGING ROWS WHEN ENCOUNTERING
C      A ZERO DIAGONAL COEFFICIENT
C
C      A :SYSTEM MATRIX
C      B :ORIGINALLY IT CONTAINS THE INDEPENDENT
C      COEFFICIENTS. AFTER SOLUTION IT CONTAINS THE
C      VALUES OF THE SYSTEM UNKNOWNNS
C
C      NX : ACTUAL NUMBER OF UNKNOWNNS
C
C      NU : ROW AND COLUMN DIMENSION OF A
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      DOUBLE PRECISION A(NY,NY),B(NY)
C      N1=NX-1
C      DO 100 K=1,N1
C      K1=K+1
C      C=A(K,K)
C      IF(DABS(C)-.000001)1,1,3
1  DO 7 J=K1,NX
C
C      TRY TO INTERCHANGE ROWS TO GET NON ZERO
C      DIAGONAL COEFFICIENT
C
C      IF (DABS(A(J,K))-0.000001)7,7,5
5  DO 6 LL=K,NX
C      C=A(K,LL)
C      A(K,LL)=A(J,LL)
6  A(J,LL)=C
C      C=B(K)
C      B(K)=B(J)
C      B(J)= C
C      C=A(K,K)
C      GO TO 3
7  CONTINUE
8  WRITE(IMP,2)K
C      D=0.0
C      GO TO 300
C
C      DIVIDE ROW BY DIAGONAL COEFFICIENT
C
C      3 C=A(K,K)
C      DO 4 J=K1,NX
C      4 A(K,J)=A(K,J)/C
C      B(K)=B(K)/C
C
C      ELIMINATE UNKNOWN X(K) FROM ROW I
C
C      DO 10 I=K1,NX
C      C=A(I,K)
C      DO 9 J=K1,NX
C      9 A(I,J)=A(I,J)-C*A(K,J)
10 B(I)=B(I)-C*B(K)

```



```

100 CONTINUE
C
C   COMPUTE LAST UNKNOWN
C
      IF (DABS(A(NX,NX))-0.000001)8,8,101
101 B(NX)=B(NX)/A(NX,NX)
C
C   APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING
C   UNKNOWNNS
C
      DO 200 LL=1,N1
      K=NX-LL
      K1=K+1
      DO 200 J=K1,NX
200 B(K)=B(K)-A(K,J)*B(J)
C
C   COMPUTE VALUE OF DETERMINANT
      D=1.0
      DO 250 I=1,NX
250 D=D*A(I,I)
      WRITE(IMP,333)D
333 FORMAT('THE DETERMINANT =',E14.8)
      2 FORMAT('**** SINGULARITY IN ROW ****',I5)
300 RETURN
      END
C
C   -----
C
      SUBROUTINE JACOBI(AS,BM,EIGV,X,ME,ND)
      *****
C
      REAL EIGV(ND),X(ND,ND), BM(ND,ND)
      REAL AS(ND,ND),D(150)
C
      AS=A : STIFNESS MATRIX
      BM=F : MASS MATRIX
      NT=ME : TOTAL DOF
      X : FINAL EIGENVECTOR
      EIGV: FINAL EIGENVALUE
      IFPR=0 :FOR PRINTOUT OF INTERM. RESULTS
      IFPR=1 :FOR PRINTOUT OF INTERM. RESULTS
C
      CONSTANTS
C
      NSMAX = 15
      RTOL = 10.**(-12)
      IFPR = 0
C
C
      NT= ME
C
      INITIALIZE EIGENVALUE AND EIGENVECTOR MATRICES
C
      DO 10 I = 1,NT
      IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 9
      WRITE (*,2020)
      STOP
9      D(I) = AS(I,I)/BM(I,I)
      EIGV(I) = D(I)
10 CONTINUE

```

```

DO 30 I = 1,NT
DO 20 J = 1,NT
20 X(I,J) = 0.
30 X(I,I) = 1.
IF (NT.EQ. 1) THEN
C   WRITE (*,*) ' EIGENVALUE : ',EIGV(1)
   RETURN
   ELSE
ENDIF

C   INITIALIZE SWEEP COUNTER AND EIGEN ITERATION
C
NSWEEP = 0
NR = NT - 1
40 NSWEEP = NSWEEP + 1
IF (IFPR .EQ. 1) WRITE (*, 2000) NSWEEP

C   CHECK IF PRESENT OFF-DIAGONAL ELEMENTS ARE LARGE ENOUGH TO
C   REQUIRE ZEROING
C
EPS = (.01**NSWEEP)**2
DO 210 J = 1,NR
JJ = J + 1
DO 210 K = JJ,NT
EPTOLA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))
EPTOLB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
IF((EPTOLA .LT. EPS) .AND. (EPTOLB .LT. EPS)) GOTO 210

C   IF ZEROING IS REQUIRED CALCULATE THE ROTATION
C   MATRIX ELEMENTS CA ,CG
C
AKK = AS(K,K) * BM(J,K) - BM(K,K) * AS(J,K)
AJJ = AS(J,J) * BM(J,K) - BM(J,J) * AS(J,K)
AB = AS(J,J) * BM(K,K) - AS(K,K) * BM(J,J)
CHECK = (AB*AB + 4.* AKK * AJJ)/4.
IF (CHECK .LT. 0.) THEN
WRITE (*,2020)
STOP
ELSE
ENDIF
60 SQCH = SQRT(CHECK)
D1 = AB/2. + SQCH
D2 = AB/2. - SQCH
DEN = D1
IF (ABS(D2) .GT. ABS(D1)) THEN
DEN = D2
ELSE
ENDIF
IF (DEN .EQ. 0.) THEN
70 CA = 0.
CG = -AS(J,K)/AS(K,K)
ELSE
80 CA = (AKK)/(DEN)
CG = - (AJJ)/(DEN)
ENDIF

C   PERFORM GENERALIZED ROTATION TO ZERO THE PRESENT OFF-DIAGONAL
C   ELEMENTS
C
90 IF (NT .EQ. 2) GOTO 190

```

```

100 JP1 = J + 1
    JM1 = J - 1
    KP1 = K + 1
    KM1 = K - 1
    IF (JM1 .LT. 1) GOTO 130
110 DO 120 I = 1, JM1
    AJ = AS(I, J)
    BJ = BM(I, J)
    AK = AS(I, K)
    BK = BM(I, K)
    AS(I, J) = AJ + CG * AK
    BM(I, J) = BJ + CG * BK
    AS(I, K) = AK + CA * AJ
120 BM(I, K) = BK + CA * BJ
130 IF (KP1 .GT. NT) GOTO 160
140 DO 150 I = KP1, NT
    AJ = AS(J, I)
    BJ = BM(J, I)
    AK = AS(K, I)
    BK = BM(K, I)
    AS(J, I) = AJ + CG * AK
    BM(J, I) = BJ + CG * BK
    AS(K, I) = AK + CA * AJ
150 BM(K, I) = BK + CA * BJ
160 IF (JP1 .GT. KM1) GOTO 190
170 DO 180 I = JP1, KM1
    AJ = AS(J, I)
    BJ = BM(J, I)
    AK = AS(I, K)
    BK = BM(I, K)
    AS(J, I) = AJ + CG * AK
    BM(J, I) = BJ + CG * BK
    AS(I, K) = AK + CA * AJ
180 BM(I, K) = BK + CA * BJ
190 AK = AS(K, K)
    BK = BM(K, K)
    AS(K, K) = AK + 2. * CA * AS(J, K) + CA * CA * AS(J, J)
    BM(K, K) = BK + 2. * CA * BM(J, K) + CA * CA * BM(J, J)
    AS(J, J) = AS(J, J) + 2. * CG * AS(J, K) + CG * CG * AK
    BM(J, J) = BM(J, J) + 2. * CG * BM(J, K) + CG * CG * BK
    AS(J, K) = 0.
    BM(J, K) = 0.

C
C
C    UPDATE EIGENVALUE MATRIX AFTER EACH ROTATION

    DO 200 I = 1, NT
    XJ = X(I, J)
    XK = X(I, K)
    X(I, J) = XJ + CG * XK
    X(I, K) = XK + CA * XJ
200 CONTINUE
210 CONTINUE

C
C
C    UPDATE EIGENVALUE AFTER EACH SWEEP

    DO 220 I = 1, NT
    IF (AS(I, I) .GT. 0. .AND. BM(I, I) .GT. 0.) GOTO 220
    WRITE (*, 2020)
    STOP
220 EIGV(I) = AS(I, I) / BM(I, I)

```

```

      IF (IFPR .EQ. 0) GOTO 230
      WRITE (*,2030)
      WRITE (*,710) (EIGV(I), I = 1,NT)
C
C      CHECK FOR CONVERGENCE
C
230  DO 240 I = 1,NT
      TOL = RTOL * D(I)
      DIF = ABS(EIGV(I) - D(I))
      IF (DIF .GT. TOL) GOTO 280
240  CONTINUE
C
C      CHECK ALL OFF-DIAGONAL ELEMENTS, TO SEE IF ANOTHER SWEEP IS REQUIRED
C
      EPS = RTOL ** 2
      DO 250 J = 1,NR
      JJ = J + 1
      DO 253 K = JJ,NT
      EPSA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))
      EPSB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
      IF((EPSA .LT. EPS) .AND. (EPSB .LT. EPS)) GOTO 253
      GOTO 280
253  CONTINUE
250  CONTINUE
C
C      FILL OUT BOTTOM TRIANGLE OF RESULTANT MATRICES AND SCALE EIGENVECTORS
C
255  DO 260 I = 1,NT
      DO 260 J = 1,NT
      AS(J,I) = AS(I,J)
260  BM(J,I) = BM(I,J)
      DO 270 J = 1,NT
      BB = SQRT(BM(J,J))
      DO 270 K = 1,NT
270  X(K,J) = X(K,J)/BB
C
C      SORT EIGENVALUE AND VECTOR
C
510  KOUNT = 0
      DO 520 I = 1,NT-1
      IF (EIGV(I) .GT. EIGV(I+1)) THEN
      KOUNT = 1
      CHNG = EIGV(I)
      EIGV(I) = EIGV(I+1)
      EIGV(I+1) = CHNG
      DO 530 J = 1,NT
      CHNG = X(J,I)
      X(J,I) = X(J,I+1)
      X(J,I+1) = CHNG
530  CONTINUE
      ENDIF
520  CONTINUE
      IF (KOUNT .EQ. 1) GOTO 510
      IF (IFPR .EQ. 0) GOTO 276
      WRITE (*,960)
      DO 275 K = 1,NT
      WRITE (*,710) (X(K,J), J = 1,NT)
275  CONTINUE
276  RETURN
C

```

```

C      UPDATE D MATRIX AND START NEW SWEEP, IF ALLOWED
C
280  DO 290 I = 1,NT
290  D(I) = EIGV(I)
      IF (NSWEEP .LT. NSMAX) GOTO 40
      GOTO 255
710  FORMAT (10(E12.5,1X))

910  FORMAT (//, ' ** STFNSSS MATRIX **',/)
920  FORMAT (9(G10.4,1X)/
      5X,9(G10.4,1X))
930  FORMAT (//, ' ** MASS MATRIX **',/)
960  FORMAT (//, ' FINAL EIGEN VECTOR OF MODAL SYSTEM',/)
2000 FORMAT (//, ' SWEEP CYCLE IN JACOBI ' ,I4,/)
2020 FORMAT (10X'ERROR SOLUTION STOP', /
      .5X,' MATRICES NOT POSITIVE DEFINITE')
2030 FORMAT (' CURRENT EIGENVALUE IN *JACOBI* ARE',/)
300  RETURN
      END

```

```

C
C      -----
C
      SUBROUTINE OUTPUT(XM,YM,ZM,FID,FIB,DFI,AFR,AFI,AKR,AKI,
* NO,N1,N2,NOUT,FLAG,DF)
*****
C
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
      COMMON/ADD/HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION XM(1),YM(1),ZM(1),FID(1),FIB(1),
      DFI(1),DF(1)
      REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
      REAL UL(80,80),UB(80,80),VL(80,80),VB(80,80)
      DOUBLE PRECISION PRA(80),PIA(80),QRA(80),QIA(80)
      COMPLEX XKAP(80),PSI(80,80),TEMP1(80,80),TEMP2(80,80)
      COMPLEX SUM3,DX(80),SUMT
      REAL TEMP1R(80,80),TEMP2R(80,80)
      REAL DXR(80),DXI(80),SUM3R,SUMT1
      LOGICAL FLAG

C
C      RES=0.
      WRITE(IMP,500)

C
C      WRITE RESULTS FOR RESERVOIR SURFACE
C
      IF(NOUT.EQ.1) THEN
        WRITE(IMP,600)
        WRITE(IMP,330)
        WRITE(IMP,600)
        WRITE(IMP,100)
        WRITE(IMP,600)
        WRITE(IMP,300)
        WRITE(IMP,600)
        DO 10 I=1,N0
          WRITE(IMP,200)XM(I),YM(I),ZM(I),RES,DFI(I)
10      CONTINUE
        WRITE(IMP,600)
        WRITE(IMP,320)
        WRITE(IMP,600)

```

```

DO 20 I=1,N0
WRITE(IMP,200)XM(I),YM(I),ZM(I),RES,DFI(I+N)
20 CONTINUE
END IF

C
C WRITE RESULTS FOR DAM FACE
C
WRITE(IMP,600)
WRITE(IMP,340)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 30 I=N0+1,N1
WRITE(IMP,200)XM(I),YM(I),ZM(I),DFI(I),FID(I-N0)
30 CONTINUE
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
DO 40 I=N0+1,N1
WRITE(IMP,200)XM(I),YM(I),ZM(I),DFI(I+N),RES
40 CONTINUE

C
C WRITE RESULTS FOR RESERVOIR BOTTOM
C
IF(NOUT.EQ.1) THEN
WRITE(IMP,600)
WRITE(IMP,350)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 50 I=N1+1,N2
QR=FIB(I-N1)
IF(CONST2.GT.1.0E-08) THEN
QR=QR+CONST1*CONST2*DFI(I+N)
END IF
WRITE(IMP,200)XM(I),YM(I),ZM(I),DFI(I),QR
50 CONTINUE
WRITE(IMP,600)
WRITE(IMP,320)
WRITE(IMP,600)
DO 60 I=N1+1,N2
QI=0.
IF(CONST2.GT.1.0E-08) THEN
QI=QI-CONST1*CONST2*DFI(I)
END IF
WRITE(IMP,200)XM(I),YM(I),ZM(I),DFI(I+N),QI
60 CONTINUE
END IF

C
C WRITE RESULTS FOR THE FAR END
C
IF(ME.GT.0) THEN
IF(CONST2.GT.1.0E-08) THEN
DO 80 J=1,ME
DO 70 I=1,ME
UL(I,J)=AFR(I,J)*AKR(J)

```

```

      VB(I,J)=AFI(I,J)*AKI(J)
      VL(I,J)=AFI(I,J)*AKR(J)
      UB(I,J)=AFR(I,J)*AKI(J)
70  CONTINUE
80  CONTINUE
      DO 105 I=1,ME
      DO 90 J=1,ME
      UL(I,J)=UL(I,J)-VB(I,J)
      VL(I,J)=VL(I,J)+UB(I,J)
90  CONTINUE
105 CONTINUE
      ELSE IF (FLAG.AND.CONST2.LE.1.0E-08) THEN
      DO 120 J=1,ME
      DO 110 I=1,ME
      UL(I,J)=AFR(I,J)*AKR(J)
      VL(I,J)=AFR(I,J)*AKI(J)
110 CONTINUE
120 CONTINUE
      END IF
      IF (FLAG) THEN
      DO 125 I=1,ME
      SUM1=0.
      SUM2=0.
      DO 122 J=1,ME
      SUM1=SUM1+DBLE(UL(I,J))*DFI(N2+J)-DBLE(VL(I,J))*DFI(N2+J+N)
      SUM2=SUM2+DBLE(VL(I,J))*DFI(N2+J)+DBLE(UL(I,J))*DFI(N2+J+N)
122 CONTINUE
      QRA(I)=-SUM1
      QIA(I)=-SUM2
125 CONTINUE
      END IF
C
C
      IF (CONST2.GT.1.0E-08) THEN
      IF (DF(1).GT.0.0) THEN
      DO 111 I=1,ME
      XKAP(I)=CMPLX(AKR(I),AKI(I))
      DO 112 J=1,ME
      PSI(I,J)=CMPLX(AFR(I,J),AFI(I,J))
112 CONTINUE
111 CONTINUE
      DO 113 I=1,ME
      DO 114 J=1,ME
      TEMP1(I,J)=PSI(I,J)/(XKAP(J)*XKAP(J))
114 CONTINUE
113 CONTINUE
      DO 115 I=1,ME
      DO 116 J=1,ME
      SUM3=(0.0,0.0)
      DO 117 K=1,ME
      SUM3=SUM3+TEMP1(J,K)*PSI(I,K)
117 CONTINUE
      TEMP2(J,I)=SUM3
116 CONTINUE
115 CONTINUE
      DO 118 I=1,ME
      SUMT=(0.0,0.0)
      DO 119 J=1,ME
      DE=DF(J)
      SUMT=SUMT+TEMP2(I,J)*SNGL(DE)

```

[illegible]


```

        IF (.LAG) THEN
          PRA(I) = SUM1 + DBLE(DXR(I))
          PIA(I) = SUM2
        ELSE
          PRA(I) = SUM1 + DBLE(DXR(I))
          PIA(I) = SUM2 + DBLE(DXI(I))
        END IF
      ELSE
        PRA(I) = SUM1
        PIA(I) = SUM2
      END IF
160 CONTINUE
      END IF
C
C
      IF (.NOT.FLAG) THEN
        IF (DF(1).GT.0.0) THEN
          DO 311 I=1,ME
            DO 312 J=1,ME
              TEMP1R(I,J) = AFR(I,J) / (AKR(J) * AKR(J))
312 CONTINUE
311 CONTINUE
          DO 313 I=1,ME
            DO 314 J=1,MME
              SUM3R=0.0
              DO 315 K=1,ME
                SUM3R = SUM3R + TEMP1R(J,K) * AFR(I,J)
315 CONTINUE
              TEMP2R(J,I) = SUM3R
314 CONTINUE
313 CONTINUE
            DO 316 I=1,ME
              SUMT1=0.0
              DO 317 J=1,ME
                DE=DF(J)
                SUMT1 = SUMT1 + TEMP2R(I,J) * SNGL(DE)
317 CONTINUE
              DXR(I) = SUMT1
316 CONTINUE
            END IF
            DO 180 I=1,ME
              SUM1=0.
              DO 170 J=1,ME
                SUM1 = SUM1 + DBLE(AFR(I,J)) * DFI(N2+J)
170 CONTINUE
              IF (DF(1).GT.0.0) THEN
                PRA(I) = SUM1 + DBLE(DXR(I))
              ELSE
                PRA(I) = SUM1
              END IF
180 CONTINUE
C
C
            DO 205 J=1,ME
              DO 190 I=1,ME
                UL(I,J) = AFR(I,J) * AKR(J)
190 CONTINUE
205 CONTINUE
            DO 220 I=1,ME
              SUM1=0.

```

```

      DO 210 J=1,ME
      SUM1=SUM1+DBLE(UL(I,J))*DFI(N2+J)
210  CONTINUE
      QRA(I)=-SUM1
220  CONTINUE
      END IF

```

C
C

```

      WRITE(IMP,600)
      WRITE(IMP,360)
      WRITE(IMP,600)
      WRITE(IMP,100)
      WRITE(IMP,600)
      WRITE(IMP,300)
      WRITE(IMP,600)
      DO 230 I=1,ME
      K1=N2+I
      WRITE(IMP,200)XM(K1),YM(K1),ZM(K1),PRA(I),QRA(I)
230  CONTINUE
      WRITE(IMP,600)
      WRITE(IMP,320)
      WRITE(IMP,600)
      DO 240 I=1,ME
      K1=N2+I
      WRITE(IMP,200)XM(K1),YM(K1),ZM(K1),PIA(I),QIA(I)
240  CONTINUE
      END IF

```

C
C
C

```

320  FORMAT(1X,'IMAGINARY PART')
200  FORMAT(5(4X,E14.5))
100  FORMAT(1X,'BOUNDARY  NODES',//
      *,10X,'X',17X,'Y',17X,'Z',15X,'PRESSURE',5X,'PRESSURE DERIVATIVE')
300  FORMAT(1X,'REAL PART',)
330  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE'
      *)
340  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE')
350  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM')
360  FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON FAR END')
500  FORMAT(1X,70(' '))
600  FORMAT(1X,' ')
      RETURN
      END

```

C
C
C

```

      SUBROUTINE ADDUND(H,G,HOLD,AFR,AKR,AKI,N0,N1,N2,DFI,FID,FIB,
      *FLAG,DF)

```

C
C

```

      *****
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND,NB,NC,NY
      COMMON/ADD/HY,PI,ME,MF,CONST1,CONST2
      REAL AFR(ND,ND),AKR(ND),AKI(ND)
      DOUBLE PRECISION H(NY,NY),G(NY,NY),T(340,80),DF(ND)
      DOUBLE PRECISION FID(ND),FIB(NB),DFI(NY),HOLD(NY,NY)
      DOUBLE PRECISION TX(340,80)
      REAL U(80,80),V(80,80)
      REAL TEMP1(80,80),TEMP2(80,80)

```

```

      REAL SUM4,DXR(80),SUMT
      LOGICAL FLAG
C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C
      IF (ME.GT.0) THEN
        DO 20 J=1,ME
        DO 10 I=1,ME
          U(I,J)=AFR(I,J)*AKR(J)
10      CONTINUE
20      CONTINUE
        IF (FLAG) THEN
          DO 40 J=1,ME
          DO 30 I=1,ME
            V(I,J)=AFR(I,J)*AKI(J)
30      CONTINUE
40      CONTINUE
C
C
        DO 42 I=1,N
        DO 41 J=1,ME
          T(I,J)=0.0
41      CONTINUE
42      CONTINUE
C
        DO 70 I=1,N
        DO 60 J=1,ME
          SUM1=0.
          SUM2=0.
          SUM3=0.
          DO 50 K=1,ME
            KK=K+N2
            SUM1=SUM1+H(I, KK) * DBLE (AFR (K, J) )
            SUM2=SUM2+G(I, KK) *DBLE (U (K, J) )
            SUM3=SUM3-G(I+N, KK) *DBLE (V (K, J) )
50      CONTINUE
          T(I, J)=SUM1+SUM2+SUM3
60      CONTINUE
70      CONTINUE
C
        ELSE
          DO 71 I=1,N
          DO 61 J=1,ME
            SUM1=0.
            SUM2=0.
            DO 51 K=1,ME
              KK=K+N2
              SUM1=SUM1+H(I, KK) *DBLE (AFR (K, J) )
              SUM2=SUM2+G(I, KK) *DBLE (U (K, J) )
51      CONTINUE
            T(I, J)=SUM1+SUM2
61      CONTINUE
71      CONTINUE
          END IF
C
        DO 75 I=1,N

```

```

      DO 72 J=1,ME
      JJ=J+N2
      H(I,JJ)=T(I,J)
72  CONTINUE
75  CONTINUE
C
C  IMAGINARY PART
C
      IF (FLAG) THEN
C
      DO 76 I=1,N
      DO 73 J=1,ME
      TX(I,J)=0.0
73  CONTINUE
76  CONTINUE
C
      DO 120 I=1,N
      DO 110 J=1,ME
      SUM1=0.
      SUM2=0.
      SUM3=0.
      DO 100 K=1,ME
      KK=K+N2
      SUM1=SUM1+H(I+N,KK)*DBLE(AFR(K,J))
      SUM2=SUM2+G(I,KK)*DBLE(V(K,J))
      SUM3=SUM3+G(I+N,KK)*DBLE(U(K,J))
100  CONTINUE
      TX(I,J)=SUM1+SUM2+SUM3
110  CONTINUE
120  CONTINUE
C
      ELSE
      DO 121 I=1,N
      DO 111 J=1,ME
      SUM1=0.
      SUM2=0.
      DO 101 K=1,ME
      KK=K+N2
      SUM2=SUM2+H(I+N,KK)*DBLE(AFR(K,J))
      SUM1=SUM1+G(I+N,KK)*DBLE(U(K,J))
101  CONTINUE
      TX(I,J)=SUM1+SUM2
111  CONTINUE
121  CONTINUE
      END IF
C
      DO 140 I=1,N
      DO 130 J=1,ME
      H(N+I,J+N2)=TX(I,J)
130  CONTINUE
140  CONTINUE
C
C  FILL IN THE REST OF H MATRIX
C
C
      DO 160 I=1,N
      DO 150 J=1,N
      H(I,J+N)=--H(I+N,J)
      H(I+N,J+N)=H(I,J)
150  CONTINUE

```

```

160 CONTINUE
    END IF
C
C FORM RIGHT HAND VECTOR
C
    DO 180 I=1,N
    DO 170 J=N0+1,N1
    DFI(I)=DFI(I)+G(I,J)*FID(J-N0)
    DFI(I+N)=DFI(I+N)+G(I+N,J)*FID(J-N0)
170 CONTINUE
180 CONTINUE
C
    DO 200 I=1,N
    DO 190 J=N1+1,N2
    DFI(I)=DFI(I)+G(I,J)*FIB(J-N1)
    DFI(I+N)=DFI(I+N)+G(I+N,J)*FIB(J-N1)
190 CONTINUE
200 CONTINUE
C
    IF (DF(1).GT.0.0) THEN
    DO 210 I=1,ME
    DO 220 J=1,ME
    IF (AKI(J).GT.0.0) THEN
    TEMP1(I,J)=-AFR(I,J)/(AKI(J)*AKI(J))
    ELSE
    TEMP1(I,J)=AFR(I,J)/(AKR(J)*AKR(J))
    END IF
220 CONTINUE
210 CONTINUE
    DO 230 I=1,ME
    DO 240 J=1,ME
    SUM4=0.0
    DO 250 K=1,ME
    SUM4=SUM4+TEMP1(J,K)*AFR(I,K)
250 CONTINUE
    TEMP2(J,I)=SUM4
240 CONTINUE
230 CONTINUE
    DO 260 I=1,ME
    SUMT=0.0
    DO 270 J=1,ME
    DE=DF(J)
    SUMT=SUMT+TEMP2(I,J)*SNGL(DE)
270 CONTINUE
    DXR(I)=-SUMT
260 CONTINUE
    DO 290 I=1,N
    DO 300 J=1,ME
    K1=J-ME+N
    DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))
    DFI(I+N)=DFI(I+N)+HOLD(I+N,K1)*DBLE(DXR(J))
300 CONTINUE
290 CONTINUE
    END IF
    RETURN
    END
C
C
C*****THIS IS THE END OF THE CONSTANT BEM PROGRAM IN 3D CASE WITH FULL**
C*****COMPLEX FUNDAMENTAL SOLUTION*****

```

CHAPTER 2

USER'S GUIDE TO SUPERPOSITION PROGRAMS

2.1 General Comments

In this chapter two programs based on the boundary element superposition method are presented. At the beginning of the chapter, the computer program SUP.2D based on the boundary element superposition 2D formulation is briefly described. A simple example, including input and output files, is also presented. Next, the computer program SUP.3D based on the boundary element superposition 3D formulation together with an example, is described. The listings of these two programs are also included. Both programs are written in FORTRAN77.

2.2 Superposition Program for 2D Reservoirs - SUP.2D

The superposition program for 2D reservoirs - SUP.2D is based on the formulation presented in detail in Chapters 3 of Volume 1.

2.2.1 Description of the SUP.2D Program

This FORTRAN computer program can be used to calculate the hydrodynamic pressure on a rigid dam storing a two-dimensional finite reservoir and subjected to a horizontal or/and vertical motion. The program uses the constant boundary elements. The pressures and pressure derivatives are assumed to have constant values over the element and equal

to their values at the mid-node of each element.

There are no limitations on the shape of the dam or the finite reservoir. The program can solve only rigid foundation case (damping is not included).

The main program defines the limiting number of boundary elements that it can handle. The number is set to be 65 as shown in program listing. The main program calls the four following subroutines:

INPUT - to read all data and print the first part of the output.

FMAT - to form matrices required to form the system of linear equations and to rearrange them according to types of the boundary conditions.

SLNPD - to solve the system of equations by Gauss elimination.

OUTPUT -to reorder the vector of computed unknowns and print out the rest of the output file.

The subroutine FMAT calls the subroutine:

MBF - to evaluate values of Bessel function $Y_0(kr)$ and its derivative by means of series representation of the Bessel functions (3,4).

2.2.2 Input Data

The following table shows the order of presentation of data in SUP.2D program:

TABLE 2-1

Input Data for the SUP.2D Program

Card Group	No. of Cards	Format	FORTTRAN variables
A	1	18A4	TITLE
B	1	General Format	N = number of boundary elements CONST = $\frac{\omega}{c}$, wave constant D = distance between congruent and real boundaries
C	N+1	General Format	XR, YR = coordinates of nodal points of each element
D	N	General Format	KODE, FI = kode value (0 or 1), boundary condition value of the pressure if KODE = 0 value of the pressure gradient if KODE = 1

The node numbers can start anywhere and numbered in counterclockwise direction looking into reservoir's scheme.

2.2.3 Output Data

The program prints out the input data as well as the hydrodynamic pressure value at each boundary element mid-node. The real and imaginary parts of both, the pressure and pressure derivatives are given.

2.2.4 Example

A two-dimensional finite reservoir impounded by a straight gravity dam is subjected to a horizontal harmonic motion with excitation frequency equal to $27.144 \frac{\text{rad}}{\text{s}}$ ($\omega = 1.2 \omega_1$). The reservoir is modelled using 20 constant boundary elements (see Fig. 2.1). The congruent boundary is placed 60 m. The dam as well as the foundation is rigid. The reservoir dimensions are also shown in Fig. 2.1.

Fig. 2.1 The Constant Boundary Element Superposition Model of Finite 2D Reservoir - $N = 20$

First, the input file is listed and it is followed by the output file.

• Input file - TEST2D

2D FINITE RESERVOIR BY THE SUP

```

20, 0.01885, 60.0
100.0    100.0
80.0     100.0
60.0     100.0
40.0     100.0
20.0     100.0
0.0      100.0
0.0      80.0
0.0      60.0
0.0      40.0
0.0      20.0
0.0      0.0
20.0     0.0
40.0     0.0
60.0     0.0
80.0     0.0
100.0    0.0
100.0    20.0
100.0    40.0
100.0    60.0
100.0    80.0
100.0    100.0
0 0.0
0 0.0
0 0.0
0 0.0
0 0.0
1 0.101937
1 0.101937
1 0.101937
1 0.101937
1 0.101937
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0
1 0.0

```

• Output file - OUT2D

1*****

2D FINITE RESERVOIR BY THE SUP

1DATA

NUMBER OF BOUNDARY ELEMENTS= 20

THE CONSTANT=0.018850 D= 60.00

COORDINATES OF THE EXTREME POINTS OF THE

BOUNDARY ELEMENTS

POINT	X	Y
1	0.1000000E+03	0.1000000E+03
2	0.8000000E+02	0.1000000E+03
3	0.6000000E+02	0.1000000E+03
4	0.4000000E+02	0.1000000E+03
5	0.2000000E+02	0.1000000E+03
6	0.0000000E+00	0.1000000E+03
7	0.0000000E+00	0.8000000E+02
8	0.0000000E+00	0.6000000E+02
9	0.0000000E+00	0.4000000E+02
10	0.0000000E+00	0.2000000E+02
11	0.0000000E+00	0.0000000E+00
12	0.2000000E+02	0.0000000E+00
13	0.4000000E+02	0.0000000E+00
14	0.6000000E+02	0.0000000E+00
15	0.8000000E+02	0.0000000E+00
16	0.1000000E+03	0.0000000E+00
17	0.1000000E+03	0.2000000E+02
18	0.1000000E+03	0.4000000E+02
19	0.1000000E+03	0.6000000E+02
20	0.1000000E+03	0.8000000E+02
21	0.1000000E+03	0.1000000E+03

0 BOUNDARY CONDITIONS

NO:	CODE	PRESCRIBED VALUE
1	0	0.0000000E+00
2	0	0.0000000E+00
3	0	0.0000000E+00
4	0	0.0000000E+00
5	0	0.0000000E+00
6	1	0.1019370E+00
7	1	0.1019370E+00
8	1	0.1019370E+00
9	1	0.1019370E+00
10	1	0.1019370E+00
11	1	0.0000000E+00
12	1	0.0000000E+00
13	1	0.0000000E+00
14	1	0.0000000E+00
15	1	0.0000000E+00
16	1	0.0000000E+00
17	1	0.0000000E+00
18	1	0.0000000E+00
19	1	0.0000000E+00
20	1	0.0000000E+00

0*****

RESULTS

BOUNDARY NODES

REAL PART

X	Y	PRESSURE	PRESSURE	DERIVATIVE
0.90000E+02	0.10000E+03	0.00000E+00	0.21526E+00	
0.70000E+02	0.10000E+03	0.00000E+00	0.20576E+00	
0.50000E+02	0.10000E+03	0.00000E+00	0.18080E+00	
0.30000E+02	0.10000E+03	0.00000E+00	0.14628E+00	
0.10000E+02	0.10000E+03	0.00000E+00	0.39135E-01	
0.00000E+00	0.90000E+02	0.16988E+00	0.10194E+00	
0.00000E+00	0.70000E+02	-.19337E+01	0.10194E+00	
0.00000E+00	0.50000E+02	-.43240E+01	0.10194E+00	
0.00000E+00	0.30000E+02	-.63020E+01	0.10194E+00	
0.00000E+00	0.10000E+02	-.74462E+01	0.10194E+00	
0.10000E+02	0.00000E+00	-.86108E+01	0.00000E+00	
0.30000E+02	0.00000E+00	-.10511E+02	0.00000E+00	
0.50000E+02	0.00000E+00	-.12167E+02	0.00000E+00	
0.70000E+02	0.00000E+00	-.13383E+02	0.00000E+00	
0.90000E+02	0.00000E+00	-.14179E+02	0.00000E+00	
0.10000E+03	0.10000E+02	-.14081E+02	0.00000E+00	
0.10000E+03	0.30000E+02	-.12523E+02	0.00000E+00	
0.10000E+03	0.50000E+02	-.98804E+01	0.00000E+00	
0.10000E+03	0.70000E+02	-.63130E+01	0.00000E+00	
0.10000E+03	0.90000E+02	-.21688E+01	0.00000E+00	

0 *****

2.2.5 Listing of the SUP.2D Program.

User: jablonski

Node: 1587D

Date: Mon Nov 21 15:07:02 1988

File: sup.2d.ftn

Pages: 7

```

C
C
C*****
C
C      PROGRAM SUP.2D.FTN
C
C
C      PROGRAM FOR SOLUTION OF THE 2-D HELMHOLTZ EQUATION
C      BY THE SUPERPOSITION METHOD
C
C      PROGRAMMER: ALEXANDER M. JABLONSKI
C
C      DATE: APRIL 19, 1985
C
C      CENTER: CASCADE, CARLETON UNIVERSITY
C*****
C
C
C

```

```

C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      COMMON/MAIN/ CONST,N,LEC,IMP
C      DOUBLE PRECISION XR(66),YR(66),XMR(65),YMR(65)
C      DOUBLE PRECISION XMF(65),YMF(65),A(66,66),B(66,66)
C      DOUBLE PRECISION FI(65),DFI(65),XL(65),V(65)
C      DIMENSION KODE(65)
C      LEC=2
C      IMP=3
C      OPEN(LEC,FORM='FORMATTED',STATUS='OLD',FILE='TEST2D')
C      OPEN(IMP,FORM='FORMATTED',STATUS='NEW',FILE='CUT2D')
C
C
C

```

```

C
C      LIST OF VARIABLES AND ARRAYS
C
C
C

```

```

C      DATA SET NUMBERS
C      LEC= INPUT DATA SET NUMBER
C      IMP= OUTPUT DATA SET NUMBER
C
C

```

```

C      COORDINATES OF EXTREME POINTS OF BOUNDARY ELEMENTS ON REAL BOUNDARY
C      IN ARRAY XR AND YR
C
C

```

```

C      XR(66)= X-COORDINATES
C      YR(66)= Y-COORDINATES
C
C

```

```

C      N= NUMBER OF BOUNDARY ELEMENTS
C
C

```

```

C      COORDINATES OF MIDPOINTS OF ELEMENTS ON REAL BOUNDARY
C
C

```

```

C      XMR(65)= X-COORDINATES
C      YMR(65)= Y-COORDINATES
C
C

```

```

C      A(N,N)= TRANSFORMATION MATRIX A FOR NEUMANN'S PROBLEM
C      B(N,N)= TRANSFORMATION MATRIX B FOR DIRICHLET'S PROBLEM
C
C

```

```

C      FI(65)= VECTOR OF KNOWN BOUNDARY CONDITIONS
C              IF KODE=0 - FI=KNOWN VALUE OF POTENTIAL(PRESSURE)
C              IF KODE=1 - FI=KNOWN VALUE OF POTENTIAL GRADIENT
C                        (PRESSURE GRADIENT)
C
C

```

```

C
C   DFI(65)= VECTOR OF UNKNOWNNS ( INITIALLY CONTAINS INDEPENDENT
C   COEFFICIENTS)
C
C   CALL INPUT(XR,YR,KODE,FI,D)
C
C
C   CALL FMAT(XR,YR,XMR,YMR,XMF,YMF,A,B,FI,DFI,KODE,D,XL)
C
C   CALL SLNPD(A,DFI,DE)
C
C
C   CALL OUTPUT(XMR,YMR,FI,DFI,A,B,KODE,V)
C
C   STOP
C   END
C   SUBROUTINES ARE ALSO DESCRIBED EACH SEPARATELY

```

```

-----
C
C   SUBROUTINE INPUT(XR,YR,KODE,FI,D)
C   *****
C
C   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C   COMMON/MAIN/ CONST,N,LEC,IMP
C   DOUBLE PRECISION XR(1),YR(1),FI(1),TITLE(18)
C   DIMENSION KODE(1)
C   N=NUMBER OF BOUNDARY ELEMENTS
C
C   WRITE(IMP,100)
C
C   READ NAME OF THE JOB
C
C   READ(LEC,150)TITLE
C   WRITE(IMP,250)TITLE
C
C   READ BASIC PARAMETERS
C
C   READ(LEC,*)N,CONST,D
C   WRITE(IMP,300)N,CONST,D
C
C
C   READ COORDINATES OF EXTREME POINTS OF THE
C   BOUNDARY ELEMENTS IN ARRAY XR AND YR
C
C   WRITE(IMP,500)
C   N1=N+1
C   DO 10 I=1,N1
C   READ(LEC,*)XR(I),YR(I)
10  WRITE(IMP,700)I,XR(I),YR(I)
C
C   READ BOUNDARY CONDITIONS
C   FI(I)=VALUE OF THE POTENTIAL IN NODE I IF KODE
C   =0,VALUE OF THE POTENTIAL DERIVATIVE IF KODE =1
C
C   WRITE(IMP,800)
C   DO 20 I=1,N
C   READ(LEC,*)KODE(I),FI(I)
20  WRITE(IMP,950)I,KODE(I),FI(I)

```

```

100 FORMAT('1',60('*'))
150 FORMAT(18A4)
250 FORMAT(25X,18A4)
* 200 FORMAT(I5,2(F15.10))
300 FORMAT('1','DATA',/,1X,'NUMBER OF BOUNDARY ELEMENTS=',
  *I3,/,1X,'THE CONSTANT=',F8.6,1X,'D=',F6.2)
500 FORMAT('COORDINATES OF THE EXTREME POINTS OF THE
  *BOUNDARY ELEMENTS',
  *//,4X,'POINT',10X,'X',18X,'Y')
700 FORMAT(5X,I3,2(5X,E14.7))
800 FORMAT('0',2X,'BOUNDARY CONDITIONS ',//,5X,'NO:',6X,
  *'CODE',5X,'PRESCRIBED VALUE')
* 900 FORMAT(I5,F11.6)
950 FORMAT(5X,I3,8X,I1,8X,E14.7)
  RETURN
  END

```

C

C

C

C

C

C

C

C

C

C

```

SUBROUTINE FMAT(XR,YR,XMR,YMR,XMF,YMF,A,B,FI,DFI,KODE,D,XL)
*****

```

```

THIS SUBROUTINE COMPUTES A AND B MATRICES AND
FORM THE SYSTEM OF LINEAR ALGEBRAIC EQS OF THE TYPE AX=F

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP
DOUBLE PRECISION XR(1),YR(1),XMR(1),YMR(1),XMF(N),YMF(N)
DOUBLE PRECISION DFI(1),FI(1),A(N,N),B(N,N),XL(N)
DIMENSION KODE(1)
DO 44 I=1,N
  L 44 J=1,N
  A(I,J)=0.0
  B(I,J)=0.0
44 CONTINUE
DO 10 I=1,N
  XMR(I)=(XR(I)+XR(I+1))/2
  YMR(I)=(YR(I)+YR(I+1))/2
  XL(I)=SQRT((XR(I+1)-XR(I))**2+(YR(I+1)-YR(I))**2)
  COSALF=(YR(I+1)-YR(I))/XL(I)
  SINALF=(XR(I)-XR(I+1))/XL(I)
  XMF(I)=XMR(I)+D*COSALF
  YMF(I)=YMR(I)+D*SINALF

```

```

10 CONTINUE
DO 30 J=1,N
DO 30 I=1,N
  RX=SQRT((XMR(I)-XMF(J))**2+(YMR(I)-YMF(J))**2)
  COSTH=((XMR(I)-XMF(J))*(YR(I+1)-YR(I))+(YMR(I)-YMF(J))*
  *(XR(I)-XR(I+1)))/((XL(I))*RX)
  CALL MBF(RX,XJOX,XJ1X,YOX,Y1X)
  TEMP1=YOX
  TEMP2=COSTH*Y1X
  IF(KODE(I).EQ.0) THEN
    A(I,J)=TEMP1
    B(I,J)=TEMP2
  ELSE
    A(I,J)=TEMP2
    B(I,J)=TEMP1
  END IF
30 CONTINUE

```



```

DO 40 I=1,N
DFI(I)=0.0
40 CONTINUE
DO 50 I=1,N
IF(KODE(I).EQ.0) THEN
DFI(I)=DFI(I)+4*FI(I)
ELSE
DFI(I)=DFI(I)+(4/CONST)*FI(I)
END IF
50 CONTINUE
RETURN
END

-----

SUBROUTINE SLNPD(A,B,DE)
*****

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP
SOLUTION OF LINEAR SYSTEM OF EQUATIONS
BY THE GAUSS ELIMINATION METHOD PROVIDING
FOR INTERCHANGING ROWS WHEN ENCOUNTERING
A ZERO DIAGONAL COEFFICIENT

A :SYSTEM MATRIX
B :ORIGINALLY CONTAINS THE INDEPENDENT
COEFFICIENTS. AFTER SOLUTION IT CONTAINS THE
VALUES OF THE SYSTEM UNKNOWNNS

N : ACTUAL NUMBER OF UNKNOWNNS

DOUBLE PRECISION A(N,N),B(N)
NX : ROW AND COLUMN DIMENSION OF A
NX=N

N1=NX-1
DO 100 K=1,N1
K1=K+1
C=A(K,K)
IF(ABS(C)-.000000001)1,1,3
1 DO 7 J=K1,NX

TRY TO INTERCHANGE ROWS TO GET NON ZERO
DIAGONAL COEFFICIENT

IF (ABS(A(J,K))-0.000000001)7,7,5
5 DO 6 L=K,NX
C=A(K,L)
A(K,L)=A(J,L)
6 A(J,L)=C
C=B(K)
B(K)=B(J)
B(J)=C
C=A(K,K)
GO TO 3
7 CONTINUE
8 WRITE(IMP,2)K
D=0.0
GO TO 300

```

```

C
C      DIVIDE ROW BY DIAGONAL COEFFICIENT
C
3  C=A(K,K)
   DO 4 J=K1,NX
4  A(K,J)=A(K,J)/C
   B(K)=B(K)/C
C
C      ELIMINATE UNKNOWN X(K) FROM ROW I
C
   DO 10 I=K1,NX
   C=A(I,K)
   DO 9 J=K1,NX
9  A(I,J)=A(I,J)-C*A(K,J)
10 B(I)=B(I)-C*B(K)
100 CONTINUE
C
C      COMPUTE LAST UNKNOWN
C
   IF (ABS(A(NX,NX))- .00000001) 101,101,102
101 WRITE(IMP,2)NX
   D=0.
   GO TO 300
102 B(NX)=B(NX)/A(NX,NX)
C
C      APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING
C      UNKNOWN
C
   DO 200 L=1,N1
   K=NX-L
   K1=K+1
   DO 200 J=K1,NX
200 B(K)=B(K)-A(K,J)*B(J)
C
C      COMPUTE VALUE OF DETERMINATE
DE=1.0
   DO 250 I=1,NX
250 DE=DE*A(I,I)
2  FORMAT('**** SINGULARITY IN ROW ****',I5)
300 RETURN
END

```

```

C
C-----
C
SUBROUTINE OUTPUT(XMR,YMR,FI,DFI,A,B,KODE,V)
*****

```

```

C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP
DOUBLE PRECISION XMR(1),YMR(1),FI(1),DFI(1)
DOUBLE PRECISION A(N,N),B(N,N),V(N)
DIMENSION KODE(1)
WRITE (IMP,100)
DO 20 I=1,N
V(I)=0.0
DO 10 J=1,N
IF (KODE(I).EQ.0) THEN
V(I)=V(I)+0.25*CONST*B(I,J)*DFI(J)
ELSE
V(I)=V(I)+0.25*B(I,J)*DFI(J)

```

```

      END IF
10  CONTINUE
20  CONTINUE
    DO 30 I=1,N
      IF(KODE(I))16,16,17
16  WRITE(IMP,200)XMR(I),YMR(I),FI(I),V(I)
      GO TO 30
17  WRITE(IMP,200)XMR(I),YMR(I),V(I),FI(I)
200 FORMAT(4(3X,E11.5))
30  CONTINUE
    WRITE(IMP,500)
100 FORMAT('0',60(' '),//,1X,'RESULTS',//,1X,'BOUNDARY  NODES',//
*,1X,'REAL PART',//,8X,'X',12X,'Y',10X,'PRESSURE',4X,'PRESSURE
*DERIVATIVE')
500 FORMAT('0',120(' '))
    RETURN
    END

C
C
C -----
C
SUBROUTINE MBF(RA,XJ0X,XJ1X,Y0X,Y1X)
*****
C
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP
COMMON/ADD/ HY,PI,ME
REAL*8 XULT0,XOSH0,XOSH1
INTEGER XF,Z0,Z
C BESSEL FUNCTION CALCULATIONS
XX=RA*CONST
IF(XX-15.9)1000,1000,2000
1000 A=0.6366197724
    B=0.5772156646
    XULT0=1.0
    SIGMA=0.0
    TY0=0.0
    TY1=0.0
    TJ0=0.0
    TJ1=0.0
    DO 7 I=1,500
      XF=I
      Z=-1
      Z0=XF+1
      SINY0=Z**(XF)
      SINY1=Z**(Z0)
      SINJ0=Z**(Z0)
      SINJ1=Z**(XF)
      XOSH0=(XX/2.0)**(2*XF)
      XOSH1=(XX/2.0)**(2*XF-1)
      XULT0=XULT0*XF
      XULT1=XULT0/XF
      SIGMA=SIGMA+(1.0/XF)
      VALU0=DLOG(XX/2.0)+B-SIGMA
      VALU1=VALU0+(1.0/(2.0*XF))
      TRMY0=((SINY0*XOSH0)/(XULT0**2))*(VALU0)
      TRMY1=((SINY1*XOSH1)/(XULT0*XULT1))*(VALU1)
      TRMJ0=(SINJ0*XOSH0)/(XULT0**2)
      TRMJ1=(SINJ1*XOSH1)/(XULT0*XULT1)
      TY0=TY0+TRMY0
      TY1=TY1+TRMY1

```

```

TJ0=TJ0+TRMJ0
TJ1=TJ1+TRMJ1
ERY0=ABS (TRMY0/TY0)
ERY1=ABS (TRMY1/TY1)
ERJ0=ABS (TRMJ0/TJ0)
ERJ1=ABS (TRMJ1/TJ1)
IF (ERY0-ERY1) 1,1,2
1 ERY=ERY1
  GC TO 5
2 ERY=ERY0
5 IF (ERJ0-ERJ1) 3,3,4
3 ERJ=ERJ1
  GO TO 6
4 ERJ=ERJ0
6 IF (ERY-ERJ) 8,8,9
8 ER=ERJ
  GO TO 10
9 ER=ERY
10 IF (ER-.00001) 11,11,7
7 CONTINUE
11 Y0X=A* (DLOG (XX/2.0)+B+TY0)
  Y1X=A* ((1.0/XX)-TY1)
  XJ0X=1.0-TJ0
  XJ1X=TJ1
  GO TO 100
2000 AH=3.1415926536/4.0
  BH=XX-AH
  CH=XX+AH
  S1H=SIN(BH)
  S2H=SIN(CH)
  DHJ0=S1H/(8.0*XX)
  DHJ1=S2H*3.0/(8.0*XX)
  DHY0=S2H/(8.0*XX)
  DHY1=S1H*3.0/(8.0*XX)
  HH=XX-(3.0*AH)
  S3H=SIN(HH)
  EH=2.0/(3.1415926536*XX)
  FH=SQRT(EH)
  XMJ0=S2H+DHJ0
  XMJ1=S1H+DHJ1
  XMY0=S1H-DHY0
  XMY1=S3H+DHY1
  XJ0X=XMJ0*FH
  XJ1X=-XMJ1*FH
  Y0X=XMY0*FH
  Y1X=-XMY1*FH
100 RETURN
  END

```

C

C***** THIS IS THE END OF THE SUP.2D.FTN PROGRAM*****

2.3 Superposition Program for 3D Reservoirs - SUP.3D

The superposition program for 2D reservoirs - SUP.3D is based on the formulation presented in detail in Chapters 5 and 6 of Volume 1.

2.3.1 Description of the SUP.3D Program

This FORTRAN computer program can be used to calculate the hydrodynamic pressure on a rigid dam storing a three-dimensional finite reservoir and subjected to an upstream-downstream, cross-stream or/and vertical motion. The program uses the constant boundary elements. The pressures and pressure derivatives are assumed to have constant values over the element and equal to their values at the mid-node of each element.

There are no limitations on the shape of the dam or the finite reservoir. The program can solve only rigid foundation case (damping is not included).

The main program defines the limiting number of boundary elements that it can handle. The number is set to be 240 as shown in program listing. The main program calls the four following subroutines:

INPUT - to read all data and print the first part of the output.

FMAT - to form matrices required to form the system of linear equations and to rearrange them according to types of the boundary conditions.

SLNPD - to solve the system of equations by Gauss elimination.

OUTPUT -to reorder the vector of computed unknowns and print out the rest of the output file.

2.3.2 Input Data

The following table shows the order of presentation of data in SUP.3D program:

TABLE 3-1

Input Data for the SUP.3D Program

Card Group	No. of Cards	Format	FORTTRAN variables
A	i	18A4	TITLE
B	1	General Format	N = number of corner nodes NE = number of boundary elements CONST = $\frac{w}{c}$, wave constant D = distance between congruent and real boundaries
C	N	General Format	XR,YR = coordinates of nodal points of each element
D	NE	General Format	KODE, FI = kode value (0 or 1), boundary condition value of the pressure if KODE = 0 value of the pressure gradient if KODE = 1

2.3.3 Output Data

The program prints out the input data as well as the hydrodynamic pressure value at each boundary element mid-node. The real and imaginary parts of both, the pressure and pressure derivatives are given.

2.3.4 Example

Fig. 2.2 The Constant Boundary Element Superposition Model of Finite 3D Reservoir - $N = 24$

A three-dimensional finite reservoir impounded by a straight gravity dam is subjected to a horizontal harmonic motion with excitation frequency equal to $45.24 \frac{\text{rad}}{\text{s}}$ ($\omega = 1.2 \omega_1$). The reservoir is modelled using 24 constant boundary elements (see Fig. 2.2). The congruent boundary is placed on $D = 60 \text{ m}$. The dam as well as the foundation is rigid.

First, the input file is listed and it is followed by the output file.

• **Input file - TEST3D**

```

3D FINITE RESERVOIR
26,24,0.031417,60.0
0.0,60.0,0.0
0.0,60.0,30.0
0.0,60.0,60.0
0.0,30.0,0.0
0.0,30.0,30.0
0.0,30.0,60.0
0.0,0.0,0.0
0.0,0.0,30.0
0.0,0.0,60.0
30.0,60.0,0.0
30.0,60.0,30.0
30.0,60.0,60.0
30.0,30.0,0.0
30.0,30.0,60.0
30.0,0.0,0.0
30.0,0.0,30.0
30.0,0.0,60.0
60.0,60.0,0.0
60.0,60.0,30.0
60.0,60.0,60.0
60.0,30.0,0.0
60.0,30.0,30.0
60.0,30.0,60.0
60.0,0.0,0.0
60.0,0.0,30.0
60.0,0.0,60.0
14,23,20,12
6,14,12,3
17,26,23,14
9,17,14,6
5,6,3,2
4,5,2,1
8,9,6,5
7,8,5,4
2,3,12,11
1,2,11,10
1,10,13,4
4,13,15,7
7,15,16,8
8,16,17,9
11,12,20,19
10,11,19,18
10,18,21,13
13,21,23,15
15,24,25,16
16,25,26,17
21,22,25,24
22,23,26,25
18,19,22,21
19,20,23,22

```


• Output file - OUT3DC

CONSTANT QUADRILATERAL BOUNDARY ELEMENTS

3D FINITE RESERVOIR

NUMBER OF NODES= 26
 NUMBER OF ELEMENTS= 24
 THE CONSTANT= 0.031417000

D= 60.00000000

THE COORDINATES OF NODES

NODE	X	Y	Z
1	0.000000	60.000000	0.000000
2	0.000000	60.000000	30.000000
3	0.000000	60.000000	60.000000
4	0.000000	30.000000	0.000000
5	0.000000	30.000000	30.000000
6	0.000000	30.000000	60.000000
7	0.000000	0.000000	0.000000
8	0.000000	0.000000	30.000000
9	0.000000	0.000000	60.000000
10	30.000000	60.000000	0.000000
11	30.000000	60.000000	30.000000
12	30.000000	60.000000	60.000000
13	30.000000	30.000000	0.000000
14	30.000000	30.000000	60.000000
15	30.000000	0.000000	0.000000
16	30.000000	0.000000	30.000000
17	30.000000	0.000000	60.000000
18	60.000000	60.000000	0.000000
19	60.000000	60.000000	30.000000
20	60.000000	60.000000	60.000000
21	60.000000	30.000000	0.000000
22	60.000000	30.000000	30.000000
23	60.000000	30.000000	60.000000
24	60.000000	0.000000	0.000000
25	60.000000	0.000000	30.000000
26	60.000000	0.000000	60.000000

CONNECTIVITY
ELEMENT NODESBOUNDARY CONDITIONS
CODE VALUE

1	14	23	20	12	0	0.0000000E+00
2	6	14	12	3	0	0.0000000E+00
3	17	26	23	14	0	0.0000000E+00
4	9	17	14	6	0	0.0000000E+00
5	5	6	3	2	1	0.1019370E+00
6	4	5	2	1	1	0.1019370E+00
7	8	9	6	5	1	0.1019370E+00
8	7	8	5	4	1	0.1019370E+00
9	2	3	12	11	1	0.0000000E+00
10	1	2	11	10	1	0.0000000E+00
11	1	10	13	4	1	0.0000000E+00

12	4	13	15	7	1	0.0000000E+00
13	7	15	16	8	1	0.0000000E+00
14	8	16	17	9	1	0.0000000E+00
15	11	12	20	19	1	0.0000000E+00
16	10	11	19	18	1	0.0000000E+00
17	10	18	21	13	1	0.0000000E+00
18	13	21	23	15	1	0.0000000E+00
19	15	24	25	16	1	0.0000000E+00
20	16	25	26	17	1	0.0000000E+00
21	21	22	25	24	1	0.0000000E+00
22	22	23	26	25	1	0.0000000E+00
23	18	19	22	21	1	0.0000000E+00
24	19	20	23	22	1	0.0000000E+00

0*****

RESULTS

BOUNDARY ELEMENTS

REAL PART

	X	Y	Z	PRESSURE	P. DERIVATIVE
1	0.45000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.20869E+00
2	0.15000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.11271E+00
3	0.45000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.20883E+00
4	0.15000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.11312E+00
5	0.00000E+00	0.45000E+02	0.45000E+02	-.65741E+00	0.10194E+00
6	0.00000E+00	0.45000E+02	0.15000E+02	-.39398E+01	0.10194E+00
7	0.00000E+00	0.15000E+02	0.45000E+02	-.66196E+00	0.10194E+00
8	0.00000E+00	0.15000E+02	0.15000E+02	-.39445E+01	0.10194E+00
9	0.15000E+02	0.60000E+02	0.45000E+02	-.19269E+01	0.00000E+00
10	0.15000E+02	0.60000E+02	0.15000E+02	-.54028E+01	0.00000E+00
11	0.15000E+02	0.45000E+02	0.00000E+00	-.59661E+01	0.00000E+00
12	0.15000E+02	0.15000E+02	0.00000E+00	-.59658E+01	0.00000E+00
13	0.15000E+02	0.00000E+00	0.15000E+02	-.54050E+01	0.00000E+00
14	0.15000E+02	0.00000E+00	0.45000E+02	-.19330E+01	0.00000E+00
15	0.45000E+02	0.60000E+02	0.45000E+02	-.30353E+01	0.00000E+00
16	0.45000E+02	0.60000E+02	0.15000E+02	-.73321E+01	0.00000E+00
17	0.45000E+02	0.45000E+02	0.00000E+00	-.79713E+01	0.00000E+00
18	0.45000E+02	0.22500E+02	0.15000E+02	-.73124E+01	0.00000E+00
19	0.45000E+02	0.00000E+00	0.15000E+02	-.73051E+01	0.00000E+00
20	0.45000E+02	0.00000E+00	0.45000E+02	-.30297E+01	0.00000E+00
21	0.60000E+02	0.15000E+02	0.15000E+02	-.75885E+01	0.00000E+00
22	0.60000E+02	0.15000E+02	0.45000E+02	-.31026E+01	0.00000E+00
23	0.60000E+02	0.45000E+02	0.15000E+02	-.76071E+01	0.00000E+00
24	0.60000E+02	0.45000E+02	0.45000E+02	-.31103E+01	0.00000E+00

IMAGINARY PART

1	0.45000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.15520E-01
2	0.15000E+02	0.45000E+02	0.60000E+02	0.00000E+00	0.16823E-01
3	0.45000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.16649E-01
4	0.15000E+02	0.15000E+02	0.60000E+02	0.00000E+00	0.17085E-01
5	0.00000E+00	0.45000E+02	0.45000E+02	-.22859E+00	0.00000E+00
6	0.00000E+00	0.45000E+02	0.15000E+02	-.41344E+00	0.00000E+00
7	0.00000E+00	0.15000E+02	0.45000E+02	-.23329E+00	0.00000E+00
8	0.00000E+00	0.15000E+02	0.15000E+02	-.43356E+00	0.00000E+00
9	0.15000E+02	0.60000E+02	0.45000E+02	-.22273E+00	0.00000E+00
10	0.15000E+02	0.60000E+02	0.15000E+02	-.39619E+00	0.00000E+00
11	0.15000E+02	0.45000E+02	0.00000E+00	-.43301E+00	0.00000E+00
12	0.15000E+02	0.15000E+02	0.00000E+00	-.46158E+00	0.00000E+00
13	0.15000E+02	0.00000E+00	0.15000E+02	-.42725E+00	0.00000E+00

14	0.15000E+02	0.00000E+00	0.45000E+02	-.22694E+00	0.00000E+00
15	0.45000E+02	0.60000E+02	0.45000E+02	-.19173E+00	0.00000E+00
16	0.45000E+02	0.60000E+02	0.15000E+02	-.29978E+00	0.00000E+00
17	0.45000E+02	0.45000E+02	0.00000E+00	-.32890E+00	0.00000E+00
18	0.45000E+02	0.22500E+02	0.15000E+02	-.38361E+00	0.00000E+00
19	0.45000E+02	0.00000E+00	0.15000E+02	-.37500E+00	0.00000E+00
20	0.45000E+02	0.00000E+00	0.45000E+02	-.21579E+00	0.00000E+00
21	0.60000E+02	0.15000E+02	0.15000E+02	-.34894E+00	0.00000E+00
22	0.60000E+02	0.15000E+02	0.45000E+02	-.22224E+00	0.00000E+00
23	0.60000E+02	0.45000E+02	0.15000E+02	-.29017E+00	0.00000E+00
24	0.60000E+02	0.45000E+02	0.45000E+02	-.19740E+00	0.00000E+00

0*****

2.3.5 Listing of the SUP.3D Program.

User: jablonski

Node: 1587D

Date: Mon Nov 21 17:38:50 1988

File: sup.3d.ftn

Pages: 6

```

C
C
C
C*****
C
C      PROGRAM SUP.3D.FTN
C
C
C      PROGRAM FOR SOLUTION OF THE 3-D HELMHOLTZ'S EQUATION
C      BY THE SUPERPOSITION METHOD
C
C      PROGRAMMER: ALEXANDER M. JABLONSKI
C
C      DATE: JUNE 6, 1985
C
C      CENTER: CASCADE, CARLETON UNIVERSITY
C*****
C
C
C

```

```

C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      COMMON/MAIN/ CONST,N,LEC,IMP,NX,NE
C      DOUBLE PRECISION XR(240),YR(240),ZR(240),XMR(240),YMR(240)
C      DOUBLE PRECISION ZMR(240),XMF(210),YMF(210),ZMF(210),A(300,300)
C      DOUBLE PRECISION B(300,300),FI(300),DFI(300),XL(300),V(300)
C      DIMENSION K(4,300)
C      DIMENSION KODE(300)
C      LEC=2
C      IMP=3
C      OPEN(LEC,FORM='FORMATTED',STATUS='OLD',FILE='TEST3D')
C      OPEN(IMP,FORM='FORMATTED',STATUS='NEW',FILE='OUT3D')
C
C
C

```

```

C
C      LIST OF VARIABLES AND ARRAYS
C
C
C

```

```

C      DATA SET NUMBERS
C      LEC= INPUT DATA SET NUMBER
C      IMP= OUTPUT DATA SET NUMBER
C
C
C

```

```

C      COORDINATES OF CORNER NODES OF BOUNDARY ELEMENTS OF REAL BOUNDARY
C      IN ARRAY XR AND YR
C      N=NUMBER OF CORNER NODES
C      NE=NUMBER OF BOUNDARY ELEMENTS
C      NX=2*NE MAXIMUM NUMBER OF EQUATIONS
C
C
C

```

```

C      XR(240)= X-COORDINATES
C      YR(240)= Y-COORDINATES
C
C
C

```

```

C      COORDINATES OF MIDPOINTS OF ELEMENTS ON REAL BOUNDARY
C
C
C

```

```

C      XMR(210)= X-COORDINATES
C      YMR(210)= Y-COORDINATES
C
C
C

```

```

C      A(NX,NX)= TRANSFORMATION MATRIX A FOR NEUMANN'S PROBLEM
C      B(NX,NX)= TRANSFORMATION MATRIX B FOR DIRICHLET'S PROBLEM
C
C
C

```

```

C      FI(210)= VECTOR OF KNOWN BOUNDARY CONDITIONS
C              IF KODE=0 - FI=KNOWN VALUE OF POTENTIAL(PRESSURE)
C              IF KODE=1 - FI=KNOWN VALUE OF POTENTIAL GRADIENT
C
C
C

```

6 of/de 6

MICRO

```

C                                     (PRESSURE GRADIENT)
C
C   DFI(210)= VECTOR OF UNKNOWNNS ( INITIALLY CONTAINS INDEPENDENT
C               COEFFICIENTS)
C
C   CALL INPUT (XR,YR,ZR,K,KODE,FI,D)
C
C   CALL FMAT (XR,YR,ZR,XMR,YMR,ZMR,XMF,YMF,ZMF,A,B,FI,DFI,KODE,D,XL
C   *,K)
C   CALL SLNPD (A,DFI,DE)
C
C
C   CALL OUTPUT (XMR,YMR,ZMR,FI,DFI,A,B,KODE,V)
C
C   STOP
C   END
C   SUBROUTINES ARE ALSO DESCRIBED EACH SEPARATELY
C
C   -----
C
C   SUBROUTINE INPUT (XR,YR,ZR,K,KODE,FI,D)
C   *****
C
C   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C   COMMON/MAIN/ CONST,N,LEC,IMP,NX,NE
C   DOUBLE PRECISION XR(1),YR(1),ZR(1),FI(1)
C   CHARACTER *80 TITLE
C   DIMENSION K(4,1)
C   DIMENSION KODE(1)
C   N=NUMBER OF BOUNDARY ELEMENTS
C
C   WRITE(IMP,100)
C
C   READ NAME OF THE JOB
C
C   READ(LEC,FMT='(A80)') TITLE
C   WRITE(IMP,200) TITLE
C
C   READ BASIC PARAMETERS
C
C   READ(LEC,*) N,NE,CONST,D
C   WRITE(IMP,300) N,NE,CONST,D
C
C
C   READ COORDINATES OF CORNER POINTS OF THE
C   BOUNDARY ELEMENTS IN ARRAY XR YR AND ZR
C
C   NX=2*NE
C   WRITE(IMP,400)
C   DO 10 I=1,N
C   READ(LEC,*) XR(I),YR(I),ZR(I)
10 WRITE(IMP,500) I,XR(I),YR(I),ZR(I)
C
C   READ BOUNDARY CONDITIONS
C   FI(I)=VALUE OF THE POTENTIAL IN NODE I IF KODE
C   =0,VALUE OF THE POTENTIAL DERIVATIVE IF KODE =1
C

```



```

      WRITE(IMP,600)
      DO 20 I=1,NE
20    READ(LEC,*) (K(J,I),J=1,4)
      DO 30 I=1,NE
      READ(LEC,*)KODE(I),FI(I)
      KODE(I+NE)=KODE(I)
      FI(I+NE)=0.0
30    WRITE(IMP,700) I, (K(J,I),J=1,4),KODE(I),FI(I)
100   FORMAT('1',60('*'))
200   FORMAT('//20X,'CONSTANT QUADRILATERAL BOUNDARY ELEMENTS'//20X,A80)
300   FORMAT('//NUMBER OF NODES=',I4,'//NUMBER OF ELEMENTS=',I4,/,
      *'THE CONSTANT=',F12.9/,/'D=',F12.8)
400   FORMAT('//THE COORDINATES OF NODES'//6X,'NODE',14X,'X',18X,'Y',
      *,18X,'Z'//)
500   FORMAT(5X,I4,3(10X,F10.6))
600   FORMAT('//CONNECTIVITY',45X,'BOUNDARY CONDITIONS'//,'ELEMENT',
      *5X,'NODES',40X,'CODE',9X,'VALUE'//)
700   FORMAT(I5,2X,4(4X,I4),19X,I2,5X,E14.7)
      RETURN
      END

```

C
C
C
C

```

      SUBROUTINE FMAT(XR,YR,ZR,XMR,YMR,ZMR,XMF,YMF,ZMF,A,B,FI,DFI,KODE,D
      *,XL,K)

```

C
C
C
C
C

```

      *****

```

```

      THIS SUBROUTINE COMPUTES A AND B MATRICES AND
      FORM THE SYSTEM OF LINEAR ALGEBRAIC EQS OF THE TYPE AX=F

```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NX,NE
      DOUBLE PRECISION XR(1),YR(1),ZR(1),XMR(1),YMR(1),ZMR(1)
      DOUBLE PRECISION XMF(1),YMF(1),ZMF(1),DFI(1),FI(1),A(NX,NX)
      DOUBLE PRECISION B(NX,NX),XL(1)
      DIMENSION K(4,1)
      DIMENSION KODE(1)
      DO 44 I=1,NX
      DO 44 J=1,NX
      A(I,J)=0.0
      B(I,J)=0.0
44    CONTINUE
      DO 10 I=1,NE
      N1=K(1,I)
      N2=K(2,I)
      N3=K(3,I)
      N4=K(4,I)
      XMR(I)=(XR(N1)+XR(N2)+XR(N3)+XR(N4))/4
      YMR(I)=(YR(N1)+YR(N2)+YR(N3)+YR(N4))/4
      ZMR(I)=(ZR(N1)+ZR(N2)+ZR(N3)+ZR(N4))/4
      AX=(YR(N2)-YR(N1))*(ZR(N3)-ZR(N1))-(ZR(N2)-ZR(N1))*(YR(N3)-YR(N1))
      BX=(ZR(N2)-ZR(N1))*(XR(N3)-XR(N1))-(XR(N2)-XR(N1))*(ZR(N3)-ZR(N1))
      CX=(XR(N2)-XR(N1))*(YR(N3)-YR(N1))-(YR(N2)-YR(N1))*(XR(N3)-XR(N1))
      XL(I)=SQRT(AX**2+BX**2+CX**2)
      XN=AX/XL(I)
      YN=BX/XL(I)
      ZN=CN/XL(I)
      XMF(I)=D*XN+XMR(I)
      YMF(I)=D*YN+YMR(I)

```

```

      ZMF(I)=D*ZN+ZMR(I)
10  CONTINUE
      DO 30 J=1,NE
      DO 30 I=1,NE
      RX=SQRT((XMR(I)-XMF(J))**2+(YMR(I)-YMF(J))**2+(ZMR(I)-ZMF(J))**2)
      COSALF=(XMF(I)-XMR(I))/D
      COSBET=(YMF(I)-YMR(I))/D
      COSGAM=(ZMF(I)-ZMR(I))/D
      COSTH=((XMR(I)-XMF(J))*COSALF+(YMR(I)-YMF(J))*COSBET+(ZMR(I)
*-ZMF(J))*COSGAM)/RX
      XX=CONST*RX
      PR=3.141593*RX
      TEMP1=-COS(XX)/PR
      TEMP2=(COSTH/(RX*PR))*(COS(XX)+XX*SIN(XX))
      TEMP3=SIN(XX)/PR
      TEMP4=(COSTH/(RX*PR))*(-SIN(XX)+XX*COS(XX))
      IF(KODE(I).EQ.0) THEN
      A(I,J)=TEMP1
      B(I,J)=TEMP2
      A(I+NE,J)=TEMP3
      B(I+NE,J)=TEMP4
      A(I,J+NE)=-TEMP3
      B(I,J+NE)=-TEMP4
      A(I+NE,J+NE)=TEMP1
      B(I+NE,J+NE)=TEMP2
      ELSE
      A(I,J)=TEMP2
      B(I,J)=TEMP1
      A(I+NE,J)=TEMP4
      B(I+NE,J)=TEMP3
      A(I,J+NE)=-TEMP4
      B(I,J+NE)=-TEMP3
      A(I+NE,J+NE)=TEMP2
      B(I+NE,J+NE)=TEMP1
      END IF
30  CONTINUE
      DO 40 I=1,NX
      DFI(I)=0.0
40  CONTINUE
      DO 50 I=1,NX
      DFI(I)=DFI(I)+4*FI(I)
50  CONTINUE
      RETURN
      END

```

C
C
C

SUBROUTINE SLNPD(A,B,DE)

C
C

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON/MAIN/ CONST,N,LEC,IMP,NX,NE

C

SOLUTION OF LINEAR SYSTEM OF EQUATIONS

C

BY THE GAUSS ELIMINATION METHOD PROVIDING

C

FOR INTERCHANGING ROWS WHEN ENCOUNTERING

C

A ZERO DIAGONAL COEFFICIENT

C

C

A :SYSTEM MATRIX

C

B :ORIGINALLY CONTAINS THE INDEPENDENT

C

COEFFICIENTS. AFTER SOLUTION IT CONTAINS THE

C

```

C      VALUES OF THE SYSTEM UNKNOWNNS
C
C      N : ACTUAL NUMBER OF UNKNOWNNS
C
C      DOUBLE PRECISION A(NX,NX),B(NX)
C      NX : ROW AND COLUMN DIMENSION OF A
C
      N1=NX-1
      DO 100 K=1,N1
      K1=K+1
      C=A(K,K)
      IF (ABS(C)-.00000001)1,1,3
1 DO 7 J=K1,NX
C
C      TRY TO INTERCHANGE ROWS TO GET NON ZERO
C      DIAGONAL COEFFICIENT
C
      IF (ABS(A(J,K))-0.00000001)7,7,5
2 DO 6 L=K,NX
      C=A(K,L)
      A(K,L)=A(J,L)
6 A(J,L)=C
      C=B(K)
      B(K)=B(J)
      B(J)=C
      C=A(K,K)
      GO TO 3
7 CONTINUE
8 WRITE(IMP,2)K
      D=0.0
      GO TO 300
C
C      DIVIDE ROW BY DIAGONAL COEFFICIENT
C
3 C=A(K,K)
      DO 4 J=K1,NX
4 A(K,J)=A(K,J)/C
      B(K)=B(K)/C
C
C      ELIMINATE UNKNOWN X(K) FROM ROW I
C
      DO 10 I=K1,NX
      C=A(I,K)
      DO 9 J=K1,NX
9 A(I,J)=A(I,J)-C*A(K,J)
10 B(I)=B(I)-C*B(K)
100 CONTINUE
C
C      COMPUTE LAST UNKNOWN
C
      IF (ABS(A(NX,NX))-0.00000001)101,101,102
101 WRITE(IMP,2)NX
      D=0.
      GO TO 300
102 B(NX)=B(NX)/A(NX,NX)
C
C      APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING
C      UNKNOWNNS
C
      DO 200 L=1,N1

```

```

      K=NX-L
      K1=K+1
      DO 200 J=K1,NX
200  B(K)=B(K)-A(K,J)*B(J)
C
C      COMPUTE VALUE OF DETERMINATE
      DE=1.0
      DO 250 I=1,NX
250  DE=DE*A(I,I)
      2 FORMAT('**** SINGULARITY IN ROW ****',I5)
300  RETURN
      END
C
C      -----
C
C      SUBROUTINE OUTPUT(XMR,YMR,ZMR,FI,DFI,A,B,KODE,V)
C      *****
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NX,NE
      DOUBLE PRECISION XMR(1),YMR(1),ZMR(1),FI(1),DFI(1)
      DOUBLE PRECISION A(NX,NX),B(NX,NX),V(NX)
      DIMENSION KODE(1)
      WRITE (IMP,100)
      DO 20 I=1,NX
      V(I)=0.0
      DO 10 J=1,NX
      V(I)=V(I)+0.25*B(I,J)*DFI(J)
10  CONTINUE
20  CONTINUE
      DO 30 I=1,NE
      IF (KODE(I)) 16,16,17
16  WRITE (IMP,200) I,XMR(I),YMR(I),ZMR(I),FI(I),V(I)
      GO TO 30
17  WRITE (IMP,200) I,XMR(I),YMR(I),ZMR(I),V(I),FI(I)
30  CONTINUE
      WRITE (IMP,109)
109  FORMAT(1X,'IMAGINARY PART')
      DO 31 I=1,NE
      IF (KODE(I+NE)) 18,18,19
18  WRITE (IMP,200) I,XMR(I),YMR(I),ZMR(I),FI(I+NE),V(I+NE)
      GO TO 31
19  WRITE (IMP,200) I,XMR(I),YMR(I),ZMR(I),V(I+NE),FI(I+NE)
31  CONTINUE
      WRITE (IMP,500)
100  FORMAT('0',120('*'),//,1X,'RESULTS',//,1X,'BOUNDARY ELEMENTS',//
      *,1X,'REAL PART',//,8X,'X',14X,'Y',14X,'Z',10X,
      *'PRESSURE',7X,'P. DERIVATIVE')
200  FORMAT(I3,5(3X,E11.5))
500  FORMAT('0',120('*'))
      RETURN
      END
C
C*****THIS IS THE END OF THE SUP.3D.FTN PROGRAM*****

```

REFERENCES

1. Moler, C.B., Stewart, G.W., "An Algorithm for Generalized Matrix Eigenvalues Problems", *Journal of Numerical Analysis, SIAM, Vol. 10, 1973*, pp. 241-265.
2. Bathe, K.-J., "Finite Element Procedures in Engineering Analysis", *Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982*.
3. MacLachlan, N.W., "Bessel Functions for Engineers", *Oxford at the Clarendon Press, Oxford, England, 1955*.
4. Hanna, Y.G., "Application of Boundary Element Method to Certain Problems in Structural Dynamics", *M.Eng. Thesis Submitted to Carleton University, Ottawa, Canada, 1980*.
5. Owen, D.R.J., Hinton, E., "Finite Element Programming", 1977.
6. Elkhodary, M.A., "Boundary Element Analysis of Hydrodynamic Pressure on Arch Dams", *M.Eng. Thesis Submitted to Carleton University, Ottawa, Canada, 1984*.

END

1|6·0|2·90

FIN