DEVELOPMENT OF A COUPLED FINITE ELEMENT - BOUNDARY ELEMENT PROGRAM FOR A MICROCOMPUTER

by

Steven Andrew Brown

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Mechanics

APPROVED:

M. P. Kamat, Chairman

L. G. Kraige

D. T. Mook

June, 1985

Blacksburg, Virginia

DEVELOPMENT OF A COUPLED FINITE ELEMENT - BOUNDARY ELEMENT PROGRAM FOR A MICROCOMPUTER

by

Steven Andrew Brown

M. P. Kamat, Chairman

Engineering Mechanics

(ABSTRACT)

This thesis describes the development of a coupled finite element - boundary element program for a microcomputer. The steps outlined in the thesis include the adaptation of a mainframe-based boundary element code for use on a microcomputer, the verification of this program with sample problems, the development of an algorithm for coupling the Finite Element Method to the Boundary Element Method, the implementation of the coupling algorithm with finite element and boundary element codes, including the development of a Constant Strain Triangular finite element, and the verification of the coupled program with sample problems. Conclusions are drawn from the results presented, and suggestions are made for future research in this area.

ACKNOWLEDGEMENTS

This work would not have been possible without the never ending patience of Dr. Kamat, who took the time to answer an infinite number of questions posed by the author. He also took the time to critique the manuscript for this work, for which the author is indebted to him. He also provided financial support for the author from the U. S. Air Force research grant F33615-831K-3214. For receiving this support, the author wishes to express his appreciation to the U. S. Air Force in general and to the technical monitor Dr. V. B. Venkayya in particular.

The author would like to thank the members of his examining committee Dr. Kraige and Dr. Mook for their valuable suggestions and to Dr. Frederick for assistance and guidance throughout the author's graduate program of study.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

The solution of complex engineering problems has been expedited by the use of numerical techniques for many years. Simple analysis techniques with calculations made by hand or with a slide rule were the standard for a long time. The advent of the digital computer allowed more complicated types of analysis to be performed, with high speed computations. Finite element analysis was one of the analysis techniques whose popularity rose with that of the digital computer. As the Finite Element Method (FEM) became highly developed, other paths of analysis were examined. One of the techniques which came to the forefront was the Boundary Element Method (BEM), developed on the basis of boundary integral equations. In the late 1970's, the personal computer (PC) burst on to the scene. Making high speed computing available at a low cost, the personal computer advanced very rapidly into the 1980's. With the availability of the microcomputer becoming greater, a new breed of engineering techniques was developed for the PC. It made sense to adapt some existing techniques to the PC, such as the Finite Element Method (FEM) and the Boundary Element Method (BEM). However, the problems which engineers faced also became more and more complex. There are many situations where a combination of FEM and BEM would be advantageous. These situations dictate the development of a mixed finite element - boundary element program.

The Finite Element Method is a technique for solving boundary, eigen, and initial value problems by discretizing the domain of the problem into subregions called finite elements. The finite elements have known interpolation functions which allow a function to be represented over an entire element as a linear combination of the values of the function at certain points in the element called nodes. Assembling all of the elements together using an appropriate error minimization criterion gives a set of N equations in terms of N unknown function values at the nodes. After the solution is found for the nodal function values, one can obtain the approximate solution of the original unknown function for the entire domain.

FEM can trace its heritage to the late 1950's, when it was developed by Turner, Clough, Martin, and Topp [1] for use in the solution of structural mechanics problems. An earlier work by Courant used a piecewise approximation scheme in dealing with torsion problems in the early 1940's, but it was the work of the four aforementioned researchers and the advent of the modern high speed digital computer which spawned its growth. Since then FEM has grown and diversified into such areas as fluid mechanics, heat transfer, soil mechanics, vibration (modal), and transient response types of analysis. This diversification has led to a very widespread use of finite element analysis and has resulted in the development of large, general purpose computer programs such as NASTRAN, ANSYS, and STRUDL. These programs allow the engineer to solve a wide variety of problems.

While the FEM utilizes domain discretization, BEM uses a discretization of the boundary of the domain of the problem. Both FEM and BEM can be derived on the basis of the method of weighted residuals. BEM, with its roots in the direct method of the boundary integral formulation, eliminates domain integrals from the equations, so that only boundary terms are left. The boundary is divided into subsections called boundary elements, having interpolation functions similar in nature to those in finite elements. BEM utilizes the fundamental solution for the governing differential equation and generates a system of equations interrelating all known and unknown boundary values. Once all the boundary values are known, the function can be evaluated at other points in the domain of the problem.

The history of BEM can be traced to boundary integral equations. This early work centered around fluid mechanics problems and used what is referred to as an "indirect" method of analysis. It was the 1960's work of Cruse and Rizzo [2], dealing with elastostatics problems using the so-called "direct" approach, that led to the present BEM. However, this period of time was the heyday of finite element research, so very little research work was done in the area of boundary elements. This began to change in the 1970's, when researchers began to look past finite elements for a better technique, especially for problems of fracture mechanics. Brebbia [3] made pioneering contributions in the area of BEM, especially relating the technique to other solution techniques such as Galerkin and point collocation. Later work has included time dependent and nonlinear problems, and the research continues. BEM seems to show a promising future for certain types of problems, but there are currently very few computer codes in widespread use.

There is not much literature available in the area of coupling finite elements with boundary elements. The book by Brebbia [4] outlines a few ways to combine the two schemes. All of these methods essentially generate a stiffness matrix for the boundary element model, which is then assembled with the element stiffness matrices of the finite element portion of the model. An interesting method proposed by Dunbar [5] utilizes an iterative approach in which the solution to the tractions on the finite element - boundary element interface is assumed, and the resulting displacements at the interface are calculated. These resulting displacements are used with the boundary element formulation to solve for the tractions at the interface. These tractions are then substituted into the finite element model to get the new displacements at the interface. This iterative procedure is repeated until the solutions converge. Another method, suggested by Kamat and the author [6], may have more possibilities. This method uses the minimization of a scalar function, with the scalar function being a combination of the potential energy for the finite element model and scalar form of the method of fundamental solutions [7] for the boundary element model. The composite scalar function is a weighted combination of the scalar function for each of the two models. Additionally, a preconditioned conjugate gradient technique is used to minimize the function. This eliminates the need for having to generate large stiffness matrices in core and thereby makes the method suitable for the solution of extremely large scale FEM-BEM coupled problems. The results from this technique are encouraging, and it appears that the method could be easily extended to nonlinear problems as well.

In tackling the problem of implementing a coupled FEM-BEM program on a personal computer two distinct areas of work were necessary. First, a boundary element code was implemented on the personal computer. In the author's opinion, any program developed for the PC should be as user-friendly as possible, because the person who uses a PC-based program may not have computer experience and would expect the program to be as helpful as possible. The author had previously developed finite element programs [8],[9] for the PC, and they incorporated many user-friendly features. The experience gained from developing these programs was incorporated in the PC-based BEM program. The second area of research centered on the actual coupling of the two methods. It was desirable to make the coupling as user transparent as possible; i.e., the user should be able to use the coupled program with a minimal knowledge of BEM. In this vein of thinking, the finite element program previously developed by the author was used as a basis for the coupling, with suitable modifications of the BEM program.

PROBLEM FORMULATION

The following derivations of the Finite Element Method (FEM) and the Boundary Element Method (BEM) are provided to give a basis for the techniques involved in coupling the two schemes together. The derivations are intended to give a brief outline of the formulation of the two methods, and are not meant to be rigorous. More comprehensive details of the derivation of FEM may be found in the books by Zienkiewicz [11] and Reddy [12], and of the derivation of BEM in the book by Brebbia et al [4].

Both FEM and BEM can be derived by the method of weighted residuals. The weighted residual statement for problems of elastostatics is

$$\int_\Omega (\sigma_{jk,j} + b_k)\, u_k^* \, d\Omega = 0 \qquad (1)$$

where σ_{jk} is the stress tensor, b_k is the body force vector, u_k is the displacement vector, and Ω represents the domain of the problem. Integrate Equation 1 by parts (or equivalently apply Green's theorem) once to obtain

$$-\int_\Omega \sigma_{jk}\, u_{k,j}^* \, d\Omega + \int_\Gamma n_j\, \sigma_{jk}\, u_k^* \, d\Gamma + \int_\Omega b_k\, u_k^* \, d\Omega = 0 \qquad (2)$$

with Γ being the boundary of the problem. Note that from the symmetry of the stress tensor, σ_{jk},

$$\int_\Omega \sigma_{jk}\, u^*_{k,j}\, d\Omega = \int_\Omega \sigma_{jk}\, \varepsilon^*_{jk}\, d\Omega = \int_\Omega D_{jklm}\, \varepsilon_{lm}\, \varepsilon^*_{jk}\, d\Omega = \int_\Omega \varepsilon_{lm}\, \sigma^*_{lm}\, d\Omega$$

where the strain tensor is written as ε_{jk}. Hence, Equation 2 may be rewritten as

$$-\int_\Omega D_{jklm}\, \varepsilon_{lm}\, \varepsilon^*_{jk}\, d\Omega + \int_\Gamma n_j\, \sigma_{jk}\, u^*_k\, d\Gamma + \int_\Omega b_k\, u^*_k\, d\Omega = 0 \qquad (3)$$

or as

$$-\int_\Omega \sigma^*_{jk}\, \varepsilon_{jk}\, d\Omega + \int_\Gamma n_j\, \sigma_{jk}\, u^*_k\, d\Gamma + \int_\Omega b_k\, u^*_k\, d\Omega = 0 \qquad (4)$$

THE FINITE ELEMENT METHOD (FEM)

Starting with Equation 3, choose the starred quantities (which may be considered as virtual quantities) to be represented by the same interpolation functions as the unstarred quantities. This results in Equation 3 being the basis for the Finite Element Method using the Galerkin technique. The first term will become symmetric in the derivatives of the interpolation functions, which leads to a symmetric stiffness matrix. Also, for those displacement components which are prescribed, the second term of the equation is zero. Thus, for the Finite Element formulation, set

$$u^*_k = \delta u_k$$

$$\varepsilon^*_{jk} = \delta\varepsilon_{jk} = \delta \left[\frac{1}{2} (u_{j,k} + u_{k,j}) \right] = \frac{1}{2} (\delta u_{j,k} + \delta u_{k,j})$$

The displacement field $\{u^e\}$ for an element is interpolated in terms of nodal displacements

$$\{u^e\} = [N] \{U^e\} \quad , \quad \{\delta u^e\} = [N] \{\delta U^e\} \tag{5}$$

with $\{U^e\}$ being the vector of nodal displacements for the element and [N] the interpolation matrix. Combining these relations with the strain-displacement equations, the element strains can be written

$$\{\varepsilon^e\} = [B] \{U^e\} \quad , \quad \{\delta\varepsilon^e\} = [B] \{\delta U^e\} \tag{6}$$

Substitution of Equation 5 and Equation 6 into Equation 3 yields

$$-\{\delta U^e\}^T \left(\int_\Omega [B]^T [D] [B] \, d\Omega \right) \{U^e\} + \{\delta U^e\}^T \{F^e\} + \{\delta U^e\}^T \{F^e_b\} = 0 \tag{7}$$

The second term of this equation arises from the portion of the boundary on which tractions are prescribed. Since the vector $\{\delta U^e\}$ is arbitrary, the only way Equation 7 can be satisfied is for

$$[K^e] \{U^e\} = \{F^e\} + \{F^e_b\} \tag{8}$$

where

$$[K^e] = \int_{\Omega} [B]^T [D] [B] \, d\Omega \tag{9}$$

The next step is to obtain the global system of equations for the assembled finite element model. Let the matrix [A] be defined such that if {U} is the vector of global nodal displacements and {F} is the global force vector, then

$$\{U^e\} = [A] \{U\} \quad , \quad \{F\} = [A]^T (\{F^e\} + \{F_b^e\})$$

Assembly of Equation 8 for the n elements in terms of the global unknowns gives

$$\left(\sum_{i=1}^{n} [A]^T [K^e]_i [A] \right) \{U\} = \sum_{i=1}^{n} [A]^T (\{F^e\} + \{F_b^e\})$$

which reduces to the familiar form

$$[K] \{U\} = \{F\} \tag{10}$$

The element used for the purposes of this study was a three node constant strain triangular element. This element uses linear interpolation functions for the displacements, and thereby yields constant strains within the element. After the appropriate interpolation functions are selected and substituted into the strain-displacement equations, the [B] matrix of Equation 6 becomes

$$[B] = \frac{1}{2Ae}\begin{bmatrix} \beta(1) & 0 & \beta(2) & 0 & \beta(3) & 0 \\ 0 & \gamma(1) & 0 & \gamma(2) & 0 & \gamma(3) \\ \gamma(1) & \beta(1) & \gamma(2) & \beta(2) & \gamma(3) & \beta(3) \end{bmatrix}$$

where $\beta(i) = y(j) - y(k)$ and $\gamma(i) = x(k) - x(j)$, x and y being the coordinates of the nodes, with i, j, and k permutating in their natural order. A derivation of this matrix is given in Appendix A. This element has the capability of solving plane stress or plane strain problems, using [D] matrices with the appropriate forms.

THE BOUNDARY ELEMENT METHOD (BEM)

The BEM formulation begins with the form of the weighted residual statement shown in Equation 4. This equation is integrated by parts once again to obtain

$$-\int_{\Gamma} n_j \, \sigma^{*}_{jk} \, u_k \, d\Gamma + \int_{\Gamma} n_j \, \sigma_{jk} \, u^{*}_k \, d\Gamma + \int_{\Omega} \sigma^{*}_{jk,j} \, u_k \, d\Omega$$

$$\int_{\Omega} b_k \, u^{*}_k \, d\Omega = 0 \qquad (11)$$

Let $p_k = n_j \, \sigma_{jk}$ and substitute the boundary conditions

$$p_k = n_j \, \sigma_{jk} = p_k' \quad \text{on} \quad \Gamma_\sigma$$
$$u_k = u_k' \quad \text{on} \quad \Gamma_u$$

where $\Gamma = \Gamma_\sigma + \Gamma_u$. This yields

$$-\int_{\Gamma u} u_k' \, p_k^* \, d\Gamma - \int_{\Gamma \sigma} u_k \, p_k^* \, d\Gamma + \int_{\Gamma u} p_k \, u_k^* \, d\Gamma + \int_{\Gamma \sigma} p_k' \, u_k^* \, d\Gamma + \int_{\Omega} \sigma_{jk,j}^* \, u_k \, d\Omega + \int_{\Omega} b_k \, u_k^* \, d\Omega = 0 \qquad (12)$$

Next, the quantities u_k^*, ε_{jk}^*, and σ_{jk}^* are chosen to be the solutions of

$$\sigma_{jk,j}^* + \delta_l^i = 0$$

where δ_l^i is the Dirac delta function representing a unit load at an interior point i in the l direction. Accordingly,

$$\int_{\Omega} \sigma_{jk,j}^* \, u_k \, d\Omega = -\int_{\Omega} \delta_l^i \, u_k \, d\Omega = -u_l^i \quad , \; l=1,\ 2,\ 3$$

Equation 12 then becomes

$$u_l^i = -\int_{\Gamma} u_k \, p_{lk}^* \, d\Gamma + \int_{\Gamma} p_k \, u_{lk}^* \, d\Gamma + \int_{\Omega} b_k \, u_{lk}^* \, d\Omega \qquad (13)$$

Equation 13 enables the evaluation of the displacement at any interior point once the boundary quantities are known. The solutions for p_{lk}^* and u_{lk}^*, resulting from the fundamental solution, are detailed in Appendix B. Since these are known quantities, the entire expression may be evaluated. The next step is to formulate a system of equations interrelating the boundary values. To accomplish this, the Dirac delta function (or unit load) is applied at a point i on the boundary, instead of at an interior point. To take account of the singularity at point i, the point is enclosed in a half sphere, and a limit is taken as the volume of the half sphere approaches zero. From this analysis which entails using the expressions for p^*_{lk} and u^*_{lk}, it can be shown [4] that Equation 13 becomes

$$c_i u_l^i = -\int_\Gamma u_k p^*_{lk} \, d\Gamma + \int_\Gamma p_k u^*_{lk} \, d\Gamma + \int_\Omega b_k u^*_{lk} \, d\Omega \qquad (14)$$

for any point on the boundary. If the boundary is smooth, $c_i = 1/2$. Equation 14 is the basis for the Boundary Element Method.

The next step is to discretize the boundary into a series of subregions, or elements. Interpolation functions ϕ_u and ϕ_p, representing displacement and traction fields in terms of the nodal values of the fields, must be chosen in the form

$$\{u\} = [\phi_u] \{U\} \quad , \quad \{p\} = [\phi_p] \{P\} \qquad (15)$$

Note that the interpolation functions can be chosen to be different for the displacements and tractions. The only guiding consideration in this choice is the guarantee for the existence of all the integrals in Equation 14. Substitution of the relations from Equation 15 into Equation 14 results in

$$[c]\{U\} + \sum_{j=1}^{m} \left(\int_{\Gamma j} \{P^*\} [\phi_u] \, d\Gamma \right) \{U\} = \sum_{j=1}^{m} \left(\int_{\Gamma j} \{U^*\} [\phi_p] \, d\Gamma \right) \{P\} \quad . \qquad (16)$$

where [c] is a diagonal matrix with 1/2 along its diagonal for a smooth boundary. Γj is the surface of the jth element, and m is the total number of such boundary elements. If [H'] and [G] are defined to be

$$[H'] = \sum_{j=1}^{m} \left(\int_{\Gamma j} \{P^*\} [\phi_u] \, d\Gamma \right)$$

$$[G] = \sum_{j=1}^{m} \left(\int_{\Gamma j} \{U^*\} [\phi_p] \, d\Gamma \right)$$

Equation 16 becomes

$$[c]\{U\} + [H']\{U\} = [G]\{P\} \qquad (17)$$

Furthermore, the diagonal matrix [c] and [H'] can be combined to form the [H] matrix, so Equation 17 reduces to

$$[H]\{U\} = [G]\{P\} \qquad (18)$$

which is the familiar form of the BEM equation. Since either displacements or tractions must be prescribed at all points on the boundary, Equation 18 reduces to a set of M equations with M unknowns, the unknowns being a combination of displacements and tractions. However, unlike the Finite Element Method, the system of equations involves a matrix that is neither sparse nor symmetric. After all the tractions and displacements on the boundary have been found, displacements at interior points can be found from Equation 12. By differentiating Equation 12 and using the stress-displacement relations, stresses at interior points can be easily determined [4]. See Appendix B for more information on these equations.

COUPLING DETAILS

The technique by which BEM is coupled to FEM is by creating a "stiffness matrix" for the boundary element which may be assembled with the global finite element stiffness matrix. The boundary element is treated effectively as a "super" finite element, with a variable number of nodes. The finite element program does not "know" that this element is a boundary element, as it is treated like any other finite element. Starting with the boundary element equation in the form of Equation 18, premultiply both sides by $[G]^{-1}$ to get

$$[G]^{-1}\,[H]\,\{U\} = \{P\} \tag{19}$$

Notice that this equation is in a form similar to that for element stiffness matrix equation for the Finite Element Method, as seen in Equation 8. The difference is that the Boundary Element Method monitors tractions while the Finite Element Method monitors forces. It is necessary to introduce a [T] matrix to relate forces to tractions by

$$\{F\} = [T]\ \{P\}$$

Hence, we can write Equation 19 as

$$[T]\ [G]^{-1}\ [H]\ \{U\} = [T]\ \{P\} = \{F^e\}$$

Finally, if $[K^{be}]$ is defined as

$$[K^{be}] = [T]\ [G]^{-1}\ [H] \tag{20}$$

Equation 19 becomes

$$[K^{be}]\ \{U\} = \{F^e\}$$

which is identical to the form found in Equation 8. The boundary element stiffness matrix is then assembled with the finite element stiffness matrices to form the global stiffness matrix, with the nodes in the boundary element corresponding to global nodes in the problem. This coupling technique is transparent to FEM, while using the boundary element to represent part of the domain allows much better accuracy in representing the displacement and stress field across that region than would be possible with a finite element mesh using the using the displacement formulation.

While in theory the [T] matrix effectively relates tractions to forces at the boundary, the implementation of this becomes difficult for boundary elements whose geometric nodes coincide with the interpolation nodes. There is a possibility that the normals can be discontinuous at certain points along the boundary, such as at corners. The Boundary Element Method uses a double nodes at such points thereby permitting the specification of consistent boundary conditions. However, the resulting "stiffness matrix" has to be condensed by imposing the conditions of continuity of displacements at such nodes.

Another consideration which comes into play is the fact that the stiffness matrix generated by BEM is generally a non-symmetric matrix. However, as the number of boundary elements increases, the matrix becomes more symmetric. It has been shown [4] that the matrix may be symmetrized without great error, and that this symmetrization has less and less effect as the boundary element mesh is refined. This consideration is important since most finite element programs store only the upper or lower triangular portions of both the element and global stiffness matrices. This fact dictates that the BEM stiffness matrix be symmetric and also be stored the same way.

IMPLEMENTATION

PC-BASED BEM PROGRAM SNAP/BE

Development work for the PC-based BEM program SNAP/BE centered on two areas: adapting a mainframe-based FORTRAN source code for use on the microcomputer and creating a preprocessor which would allow a user-friendly interface between the user and the program. The source code adaptation sounds trivial, but there are some important factors to consider when modifying a code to function on a microcomputer. The preprocessor is a very important link in the program; it contains the routines for data input and graphics output options, as well as being the basic controller of the program.

SNAP/BE FEATURES

The key in developing the overall structure of the program was the split BASIC-FORTRAN operation. A DOS batch file controls the overall operation of the program, accessing the BASIC preprocessor and then the FORTRAN-based solution processor. A set of files which are used exclusively by the program allows the BASIC preprocessor to transfer information to the solution processor. The reason BASIC was chosen as the base for the preprocessor is that BASIC is generally standard with all the computers, it has decent graphics capabilities which allow a pictorial representation of the problem, and it has superior character string handling which is essential for developing a program which is "user-friendly". Using FORTRAN for the solution processor was done not only because of the availability of a FORTRAN source code, but also due to the "number crunching" speed of the language. Also, the compiled code could be made compatible with machines with or without 8087 math coprocessors.

The other feature that makes SNAP/BE noteworthy is its user-friendly operation. A user-friendly program should prompt the user for all of the necessary information, without having the user consult the manual at every step in the program. The key to achieving this result is its menu driven structure. The menus employed in this program are simple to use; all that the user does is move the cursor to the line of the desired activity and press <RETURN>. The idea behind this is similar to the use of the mouse with Apple's Macintosh. See Figure 1 for an example of a menu used in SNAP/BE. Another function the user-friendly program should provide is that of error trapping, or checking to see that the user does not make an improper entry. In the event that a mistake is made, it should not be a fatal one, causing a loss of data which was input previously. SNAP/BE utilizes this for all of the data input routines, making it difficult for the user to "crash" the program.

A feature which is based in FEM type programs is data generation capability. This allows the user to enter the equivalent of many lines of data in only one line. SNAP/BE utilizes nodal coordinate generation, as well as generation of boundary conditions.

```
WELCOME TO THE 2-d ELASTOSTATIC BOUNDARY ELEMENT PROGRAM

Move cursor to desired line and press <RETURN>

     ACTIVITY:

     Enter new data
     Recall data from file
     Exit program




SNAP/BE Copyright 1984
```

Figure 1. Screen 1 - BEM Input Menu

One last item which is very important for any program which uses data entry to represent some sort of model is the presence of graphics capabilities to give a pictorial image of the data entered. SNAP/BE has a routine which plots the boundary of the problem, locates the internal calculation points, and indicates which type of boundary conditions are prescribed on each element. There is also the option to zoom or compress the picture, in order to give a better view of the model.

PROGRAM ARCHITECTURE

SNAP/BE is split into two basic sections: a BASIC preprocessor; and a FORTRAN-based solution processor. Figure 2 shows a flowchart of the program. Invoking the batch file, the BEM Input Menu appears on the screen (see Figure 1). The user is given three options, either to recall data from existing files, enter a new set of data, or to exit the program. This menu controls where the data originates from. If the new data entry option is selected, the program proceeds to the general data input screen (Figure 3). This prompts the user for the general data for the problem, including number of boundary elements, number of internal calculation points, number of non-intersecting surfaces, and the last element in each of these surfaces. The internal calculation points are points inside the domain of the problem at which displacements and stresses are calculated. This boundary element program has the capability of supporting non-intersecting surfaces within the model, which allows discontinuities to be modeled within a problem. The next screen (Figure 4) prompts for the shear modulus and Poisson's ratio of the material under consideration.

The program then prompts for the coordinates of the internal calculation points one point per screen (see Figure 5). The nodal coordinate input screen (Figure 6) follows, and utilizes a line input format. This type of input differs from the regular (single value) input in that the computer will recognize from the number of entries in the line if nodal generation is present, and prompts the user if an error occurred in the line format. The increment which the program asks for is the increment between node numbers. The last screen (Figure 7) for boundary condition input is also of the line format type. The program requires boundary conditions in the X and Y directions, and these conditions can be generated for a specified number of sequential elements.

The program then goes to the BEM Action Menu (Figure 8), which gives the user the choice of plotting the model, saving the data and perfoming the solution, saving the data without performing the solution, and exiting the program without saving the data. If the plot option is specified, the program stays in BASIC and accesses the plotting routine. After exiting the plotting routine, the program returns to the BEM Output Menu. If any of the other options are selected, BASIC is exited and the batch file controls the logic of the program. Before exiting to DOS, the program creates an execution check file which tells the batch file what execution path to take, via the commands available for batch files.

The final step in the program, if selected, is performing the solution. This FORTRAN-based code reads the data from the program data file which has been generated by the preprocessor, and after performing the

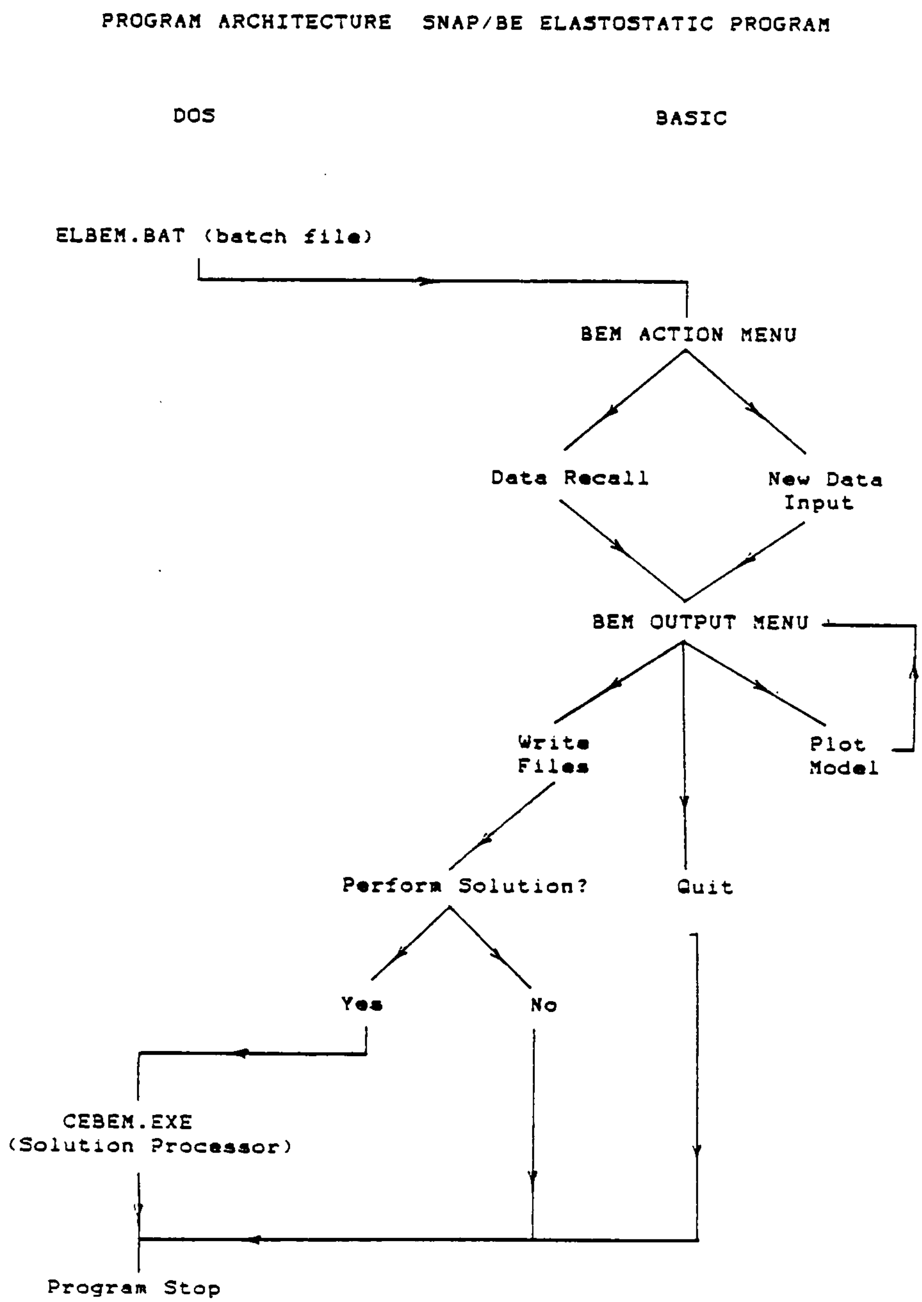

Figure 2. SNAP/BE Program Architecture

```
GENERAL INPUT

Enter a title for this problem (<72 characters)
sample beam prob. 2 - 20 elements

Enter number of boundary elements 20

Enter number of internal calculation points 5

Enter number of different surfaces - non-intersecting boundaries 1

Enter last node in surface number  1  20
```

Figure 3. Screen 2 - General Data Input Screen

```
MATERIAL PROPERTY INPUT

Enter the shear modulus for the material 11e6

Enter Poisson's ratio for the material .3
```

Figure 4. Screen 3 - Material Property Input

INTERNAL CALCULATION POINT DATA

Enter x coordinate of internal calculation point number 5 .5

Enter y coordinate of internal calculation point number 5 .5

Figure 5. Screen 4 - Internal Point Coordinate Input

```
                 COORDINATE INPUT OF FIRST ENDPOINT OF EACH ELEMENT
1  0,0,1,0,6
7  1,.2,1,1,11
12  .8,1,0,1,16
17  0,.8,0,.2,20




Element # - X , Y (, X2 , Y2 , 2nd element #)
```

Figure 6. Screen 5 - Nodal Coordinate Input

```
                         BOUNDARY CONDITION INPUT

For CODE X or CODE Y enter a D for prescribed displacement or a T for traction
For VAL X or VAL Y enter the corresponding value of displacement or traction


 1   t,0,t,0,5
 6   t,1000,t,0,5
 11  t,0,t,0,5
 16  d,0,t,0,2
 18  d,0,d,0
 19  d,0,t,0,2




Element # - CODE X , VAL X , CODE Y , VAL Y (, # elements)
```

Figure 7. Screen 6 - Boundary Condition Input

B.E.M. ACTION MENU

Plot boundary element mesh
Save data - continue execution
Save data - halt execution
Exit program - no data saved

Move cursor to desired line and press <RETURN>

Figure 8. Screen 7 - BEM Action Menu

solution, writes the results in the program results file. The results include the complete set of displacements and tractions at all of the boundary nodes, and the displacements and stresses at the internal points.

A few notes on the modification of the source code should be made. The FORTRAN compiler used for this work was Microsoft's FORTRAN version 3.1. This compiler included the ability to access the 8087 math coprocessor chip, if installed. This was a priority since its installation can increase the speed of a program, like this one, with mostly math computations, by a factor of more than ten [12]. The way that MS-FORTRAN and most other languages for the PC access files is through OPEN statements. These must be included within the source and they take the place of FILEDEF statements issued to the mainframe through an EXEC file. Probably the most important consideration is that when manipulating large arrays in MS-FORTRAN, they should be placed in named COMMON blocks. This has a limitation, however, as MS-FORTRAN 3.1 limits the size of the largest COMMON block to 64K bytes. While this implies significant information, it translates into a square matrix of 128 X 128, limiting the number of elements for an elasticity program to 64. Version 3.2 of MS-FORTRAN allows longer arrays, but they must be single dimensioned. Also, using memory 64K at a time rapidly consumes the limited memory available, but unless the machine has more than 256K RAM available, any benefits gained by modifying the program to gain more capacity would be limited.

FEM-BEM COUPLED PROGRAM

The initial work dealing with the coupling of FEM and BEM combined the BEM program described above, with constant values of displacement and tractions for each element, and a PC-based finite element code developed by the author and Kamat [8]-[9], utilizing eight node isoparametric elements with parabolic variations in displacements along the boundaries. The boundary element code was added to the finite element code, with the boundary element model being regarded as a super finite element whose "stiffness matrix" is given by Equation 20. However, problems arose in this formulation, not only with the code, but with modeling. The difficulty in modeling comes from the fact that the interpolation nodes of the boundary element are not the same as the geometric nodes. It was therefore difficult to connect nodes along the interface of the two models. The scheme which was attempted used dummy nodes for the boundary element, such that the interpolation nodes along the interface for the two models coincided. However, this method caused difficulties at the corners of the BEM model, where the model's boundary had to be distorted slightly to incorporate the geometry of the corner. These problems were disconcerting, and would require extraneous information from the user concerning the BEM discretization. Hence, it was decided to convert the program to one using constant strain triangular elements for the finite element model, and linear elements for the boundary element model.

Using both finite elements and boundary elements with linear variations of displacements seemed to be a logical decision because it would eliminate any questions about displacement incompatibility between the elements, which is important for the displacement-based finite element formulation used here. A code for linear boundary elements was obtained, and adapted into the general structure of the PC-based program. A routine was written for the constant strain triangular element (CST); the stiffness matrix for which can be formed explicitly without the need for any numerical integration required of higher order elements. See Appendix A for the derivation of the stiffness matrix for the CST element.

Another change in the second phase of the development was the use of the mainframe for the initial validation of the program code. One drawback of the PC is that compilation of long programs such as this one is very time consuming. For example, compilation of the initial code alone took anywhere from fifteen to twenty minutes, not to mention the time for the linking step. Hence, it was more expedient to develop the code initially on the mainframe, while following all the rules necessary to allow the code to be compiled on the PC with minimal modification. The preprocessor of a prototype version of SNAP/FE was modified to allow for the input of the boundary element, including graphic display of the final model. This was not as refined as the preprocessor for SNAP/BE, but still aided in the data entry of the problem while containing some of the user-friendly features. The input data files for the program were designed to be uploaded to the mainframe, and used with the program.

The development of the coupled program with linear elements proceeded in much the same manner as in the initial development work, in that the

BEM code was used to generate a stiffness matrix which would then be assembled with that of the finite element model. The theory for this is outlined in the Problem Formulation section. One question which arises in using linear boundary elements is an understanding of how to form the [T] matrix. On the basis of work equivalence, the assumption of linear displacement variation implies that the elements of the [T] matrix should be one half the lengths of the elements adjacent to each respective node. However, this creates a problem (alluded to in the Problem Formulation section) concerning a proper specification of boundary conditions at nodes where a unique normal is not defined such as at corner nodes. The Boundary Element Method uses double nodes at corners and at other places where the surface normals of adjacent elements do not lie in the same direction. Initially, only the "smearing" of elements was considered, but after some curious results the double node idea was examined. An overriding concern of this development work was to keep the data input as simple as possible, so it was desirable that the user should not be the one to decide on the locations of double nodes. This meant that some sort of logic would have to be worked into the program to account for the double nodes. The first case tried was to put double nodes at all the global boundary element nodes, even for straight boundaries. This apparently caused singularities at the nodes with straight sides. A more sophisticated scheme was then implemented, whereby if the normals of adjacent elements varied by more than fifteen degrees, then an extra node would be placed at the common node between the two elements. This would cause the size of the BEM matrices to increase depending on the shape of the boundary, but these nodes would have to be constrained to have the same displacements. This task turned out to be easier than expected, as they were identified by the same degree of freedom when assembled. A comparison of the "smearing" method and the double node method is made in the next section.

DISCUSSION OF RESULTS

RESULTS - BEM PROGRAM FOR PC

The first step in analyzing the PC-based SNAP/BE boundary element program was to check to see if it performed properly. Two verification problems were chosen, one a pressure vessel problem documented in the book by Brebbia, the other a simple plate under tension problem, the numerical results for which can be compared to theoretical values.

The pressure vessel discretization is shown in Figure 9. The results obtained from the solution by SNAP/BE compare favorably with those reported in the book, especially considering the fact that SNAP/BE performs all of its calculations in single precision arithmetic. Table 1 shows values of displacements, stresses, and tractions generated by the program compared to the documented values.

The second verification problem, a plate under tension, was discretized in two different ways to see the effect on the solution by the different models. The first problem was modeled with one quarter symmetry using sixteen boundary elements, as shown in Figure 10. The second discretization employed a full model of a plate, as shown in Figure 11. This model had twenty boundary elements. Table 2 shows the values for displacements, stresses, and tractions for both elements at selected points, and compares them to the theoretical values. As can be

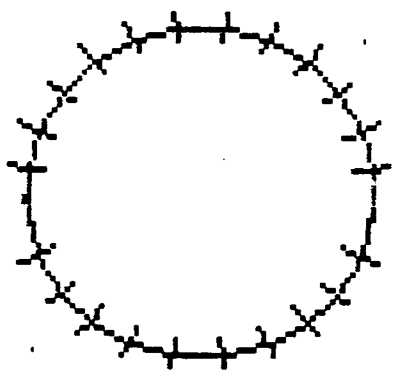

Figure 9. Boundary Element Model of Pressure Vessel Problem

Table 1. Comparison of Pressure Vessel Results

QUANTITY	REFERENCE VALUE	SNAP/BE RESULTS
Y DISPLACEMENT AT CENTROID OF ELEMENT 3	-1.15972E-02	-1.15972E-02
Y TRACTION ON ELEMENT 1	-9.65900E+01	-9.65900E+01
X DISPLACEMENT AT CENTROID OF ELEMENT 6	-1.640153E-03	-1.640153E-03
X TRACTION ON ELEMENT 4	-8.66000E+01	-8.66000E+01
Y DISPLACEMENT OF INTERNAL POINT 2	8.519233E-04	8.5193E-04
XY STRESS AT INTERNAL POINT 2	-5.717716E+01	-5.7178E+01
X DISPLACEMENT OF INTERNAL POINT 5	8.030049E-04	8.0300E-04
X STRESS AT INTERNAL POINT 5	-2.529407E+01	-2.5295E+01

Table 2 (Part 1 of 2). Comparison of Results for Plate Under Tension

QUANTITY	THEORETICAL VALUE	SNAP/BE RESULTS PROBLEM 1
X DISPLACEMENT AT CENTROID OF ELEMENT 7	1.0000E-04	0.947045E-04
X DISPLACEMENT AT CENTROID OF ELEMENT 3	0.62500E-04	0.588679E-04
X TRACTION AT ELEMENT 7	1.0000E+03	1.0000E+03
X TRACTION AT ELEMENT 14	-1.00000E+03	-0.99016E+03
X DISPLACEMENT AT INTERIOR POINT X=.9 Y=.9	0.90000E-04	0.84221E-04
X STRESS AT INTERIOR POINT X=.9 Y=.9	1.0000E+03	0.83361E+03

QUANTITY	THEORETICAL VALUE	SNAP/BE RESULTS PROBLEM 2
X DISPLACEMENT AT CENTROID OF ELEMENT 3	0.50000E-04	0.471147E-04
X DISPLACEMENT AT CENTROID OF ELEMENT 8	1.00000E-04	0.945341E-04

Table 2 (Part 2 of 2). Comparison of Results for Plate Under Tension

QUANTITY	THEORETICAL VALUE	SNAP/BE RESULTS PROBLEM 2
X TRACTION ON ELEMENT 8	1.0000E+03	1.0000E+03
X TRACTION ON ELEMENT 18	-1.0000E+03	-0.996164E+03
X DISPLACEMENT AT INTERIOR POINT X=.9 Y=.9	0.90000E-04	0.84390E-04
X STRESS AT INTERIOR POINT X=.9 Y=.9	1.00000E+03	0.92497E+03

seen, the solutions generated by SNAP/BE compare favorably with the exact values, especially when considering the coarseness of the meshes used. The stresses for the full model seemed to be more accurate than the quarter symmetry model, possibly due to the more accurate representation of the model.

After the initial verification problems were modeled, two larger problems were modeled in order to show the versatility of the program. The problems chosen were a classical plate with a central hole, and a gear tooth problem. The plate with a hole is easily verified with classical solutions and is generally a well understood problem; the gear tooth is a problem where the use of boundary elements can be more advantageous than the use of finite elements, due to the large stress gradients near the root of the tooth.

There were three discretizations of the plate with a hole considered, in order to examine the effect of boundary conditions on the solution. The models are shown in Figure 12, Figure 13, and Figure 14. A finite element discretization of the same problem with quarter symmetry is shown in Figure 15 and Figure 16. It was found that the solution for the BEM discretization with one quarter symmetry did not give results for the stresses near the hole as good as the full model or the model with one half symmetry. This is probably due to the fact that the largest stress concentrations for the quarter symmetry problem were near the corner, and constant boundary elements are known not to be able to handle sharp corners well. Table 3 gives a summary of selected displacements, tractions,

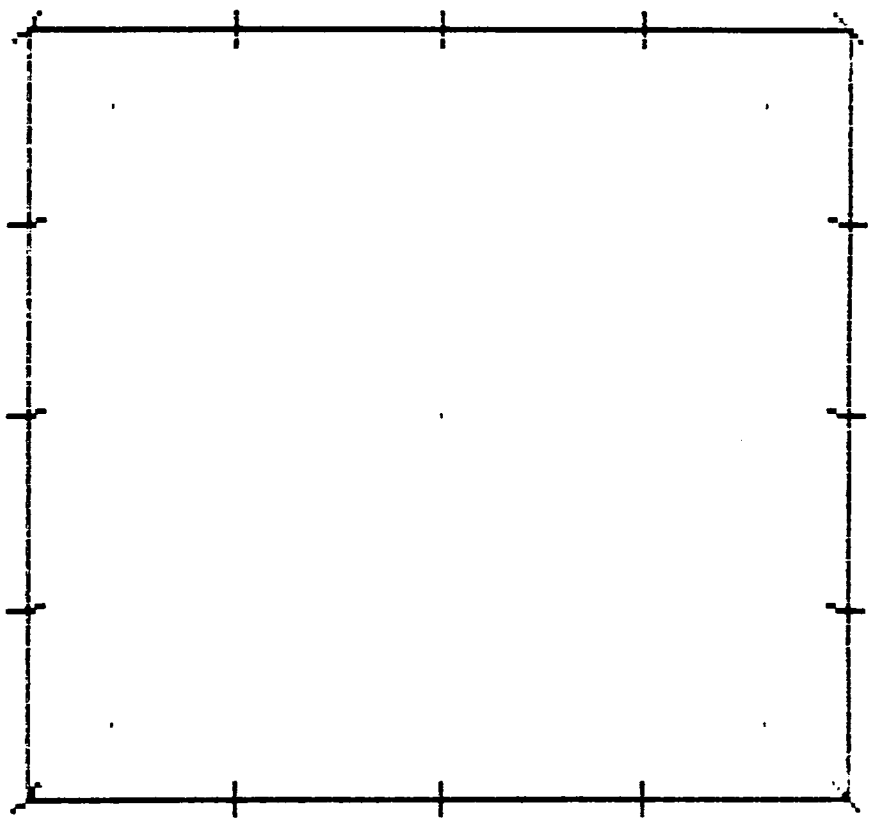

Figure 10. BEM Plate Under Tension - Quarter Symmetry

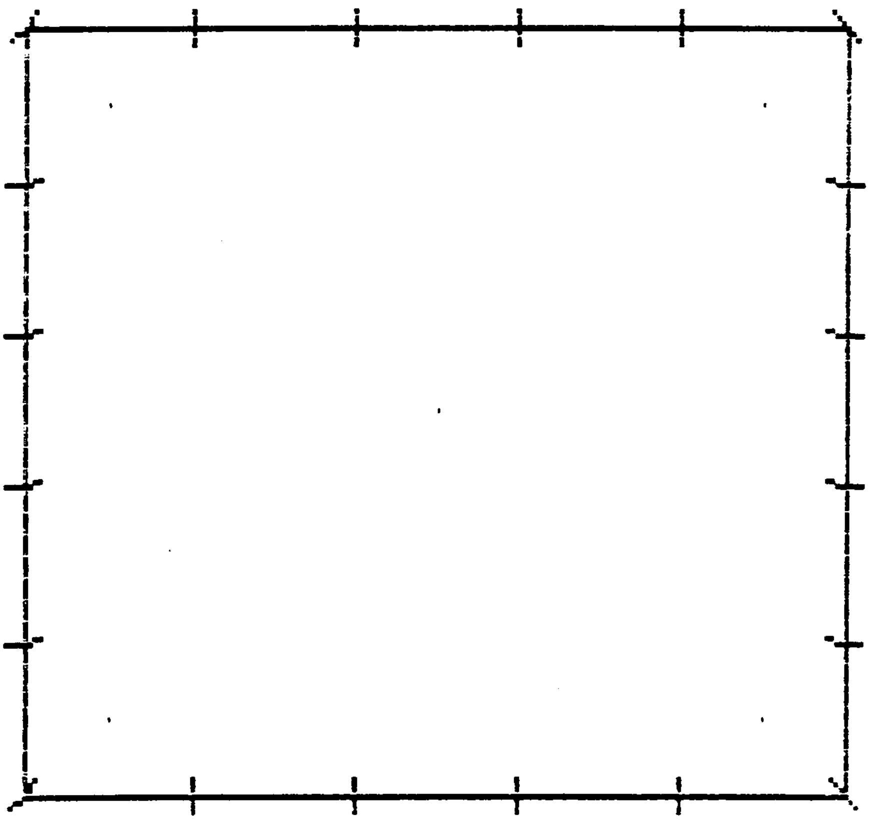

Figure 11. BEM Plate Under Tension - Full Model

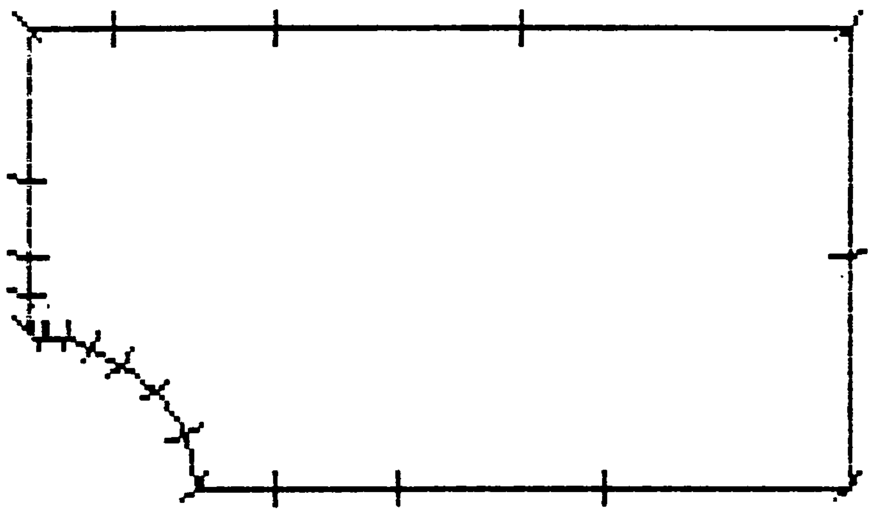

Figure 12. BEM Plate With Hole - Quarter Symmetry

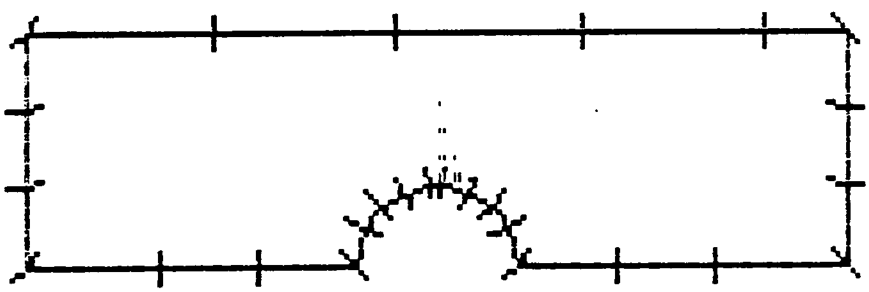

Figure 13. BEM Plate With Hole - Half Symmetry

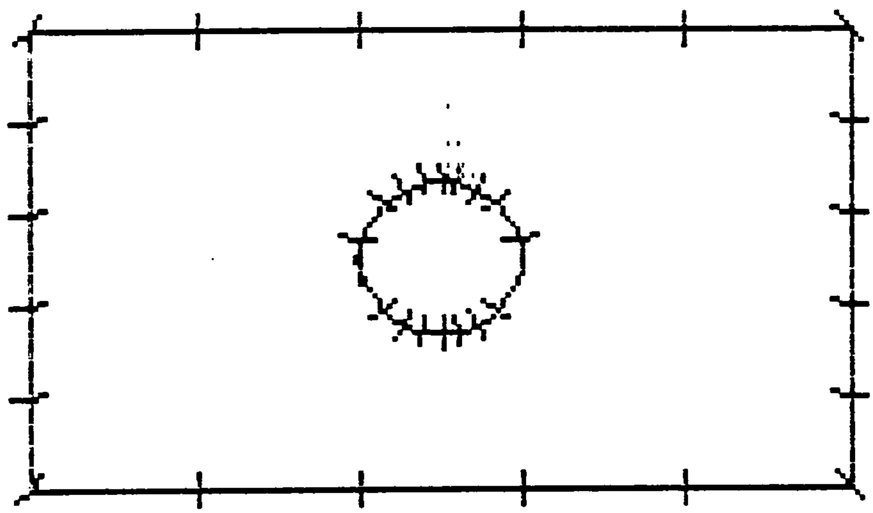

Figure 14. BEM Plate With Hole - Full Model

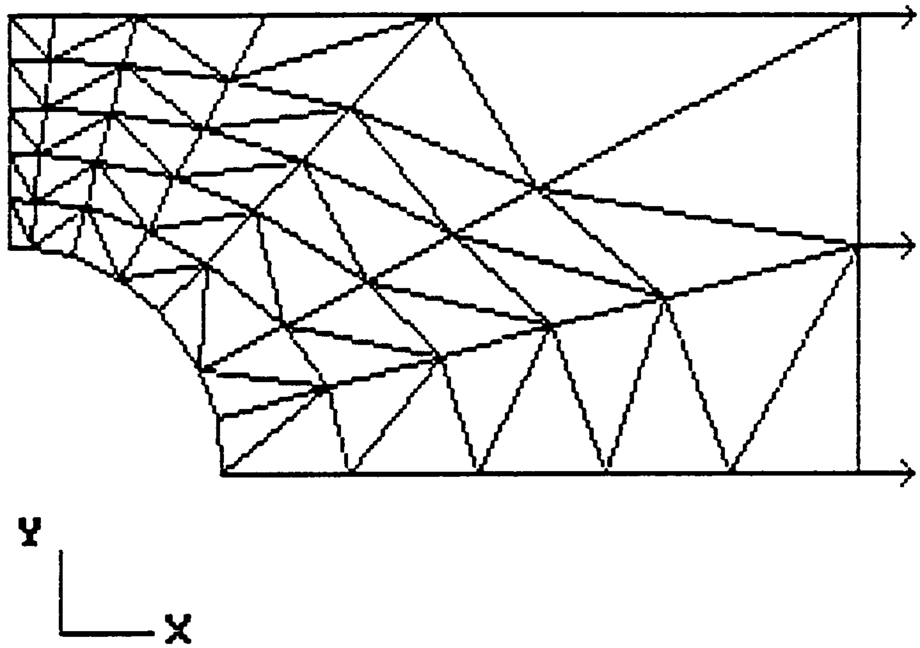

Figure 15. FEM Plate With Hole - Coarse Discretization

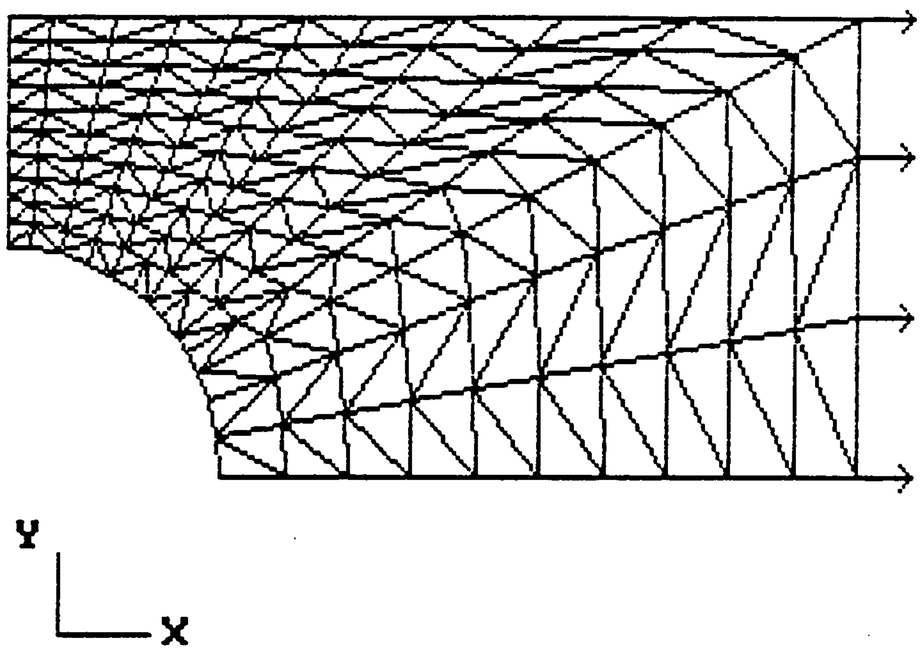

Figure 16. FEM Plate With Hole - Fine Discretization

Table 3. Comparison of Results for Plate With Hole Discretizations

QUANTITY	QUARTER SYMMETRY	HALF SYMMETRY	FULL MODEL
X DISPLACEMENT OF TOP RIGHT CORNER X.001	0.178875	0.178162	0.179701
X DISPLACEMENT AT INSIDE OF HOLE ON X AXIS X.001	0.101953	0.110994	0.118458
X STRESS AT A POINT NEAR HOLE X=0.1 Y=1.1 X1000	3.0077	2.6055	2.9539

and stresses for each of the three models.

The gear tooth problem was modeled to show the capabilities of the Boundary Element Method. Figure 17 shows a typical gear tooth discretization with 53 boundary elements, while Figure 18 shows a comparable finite element discretization of the same model, using 221 nodes and 384 CST finite elements to describe the problem. While the computation time required for the two models was similar, the BEM problem required much less input data than the finite element model, which resulted in less overall time being spent solving the problem. The Boundary Element Method also has the advantage of being able to choose the points inside the domain of the problem at which stresses are calculated without the use of any numerical integration with quadratures, whereas the FEM does not give the user any such freedom of choice, and would require numerical integration by quadratures for models with higher order elements. Furthermore, the stress and displacement fields calculated for the boundary element model are continuous throughout the domain, while the stress fields are only piecewise continuous for the finite element model.

RESULTS - COUPLED FEM-BEM PROGRAM

The first step in analyzing the coupled FEM-BEM program was determining whether the boundary element stiffness matrix was being formed correctly, and if the calculation of stresses and displacements at internal points was being performed correctly. This step was verified by again solving simple plate under tension problem, this time with a

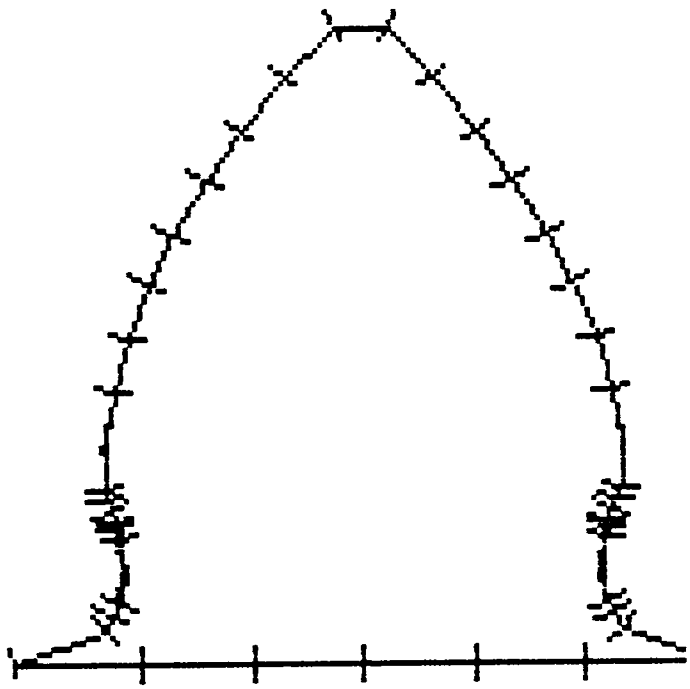

Figure 17. Gear Tooth Model - BEM Discretization

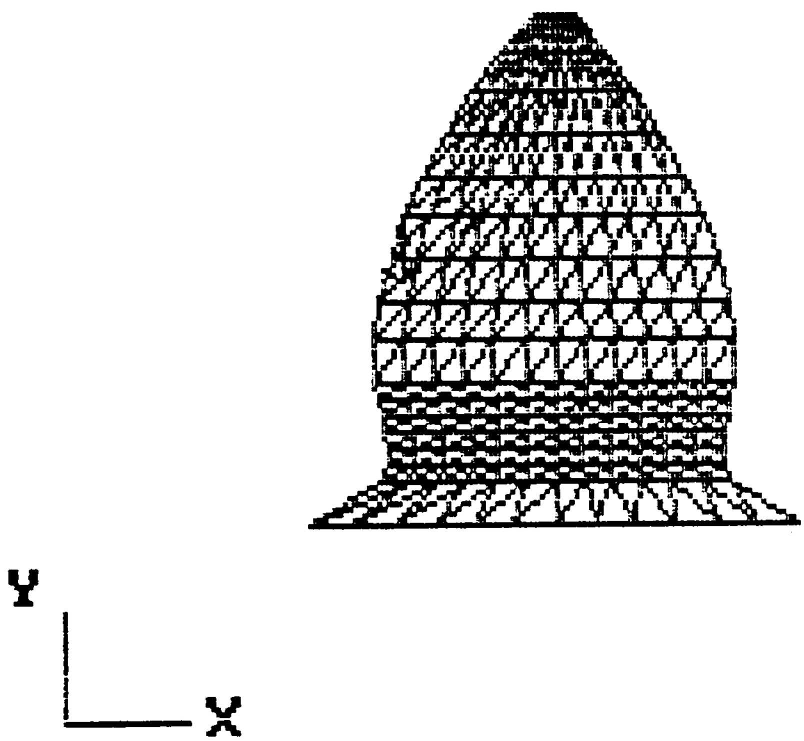

Figure 18. Gear Tooth Model - FEM Discretization

domain which was comprised entirely of a boundary element formed from the coupling technique. This would help give a feel for how the coupled element might perform. After the element was checked out, the "super" element could be coupled with the CST finite elements to solve the plate with a hole problem, as examined previously using the pure BEM program.

The verification of the coupled boundary element for the problem of a plate under tension was examined using four different discretizations of the problem illustrated in Figure 19. The first model utilized eight identical boundary elements, two on each side. The second and third discretizations both had sixteen nodes, with four elements on each side, but they differed in that the second model had elements of equal length while the third model had elements of unequal length, with the smaller elements being at the corners and the larger elements being at the centers of the sides. The fourth discretization of the problem had 32 nodes, with eight equal length elements on each side. The discretizations for the problem are shown in Figure 20 and Figure 21.

The results of these four "plate under tension" problems were quite interesting and are tabulated in Table 4. The coarse model, with two nodes per side, gave answers which were not quite those rendered by a pure BEM discretization of the same problem. The coupled model gave displacements which were too low at the center of the right edge, and stresses on the corners of the boundary which were nowhere close to the expected values. It was surmised that this condition may have been caused by the error introduced as a result of the artificial symmetrization of

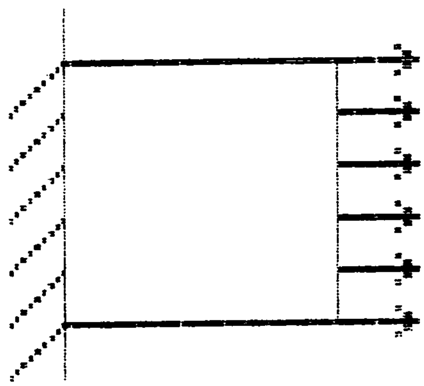

Figure 19. Pictorial Description of Plate Under Tension Problem

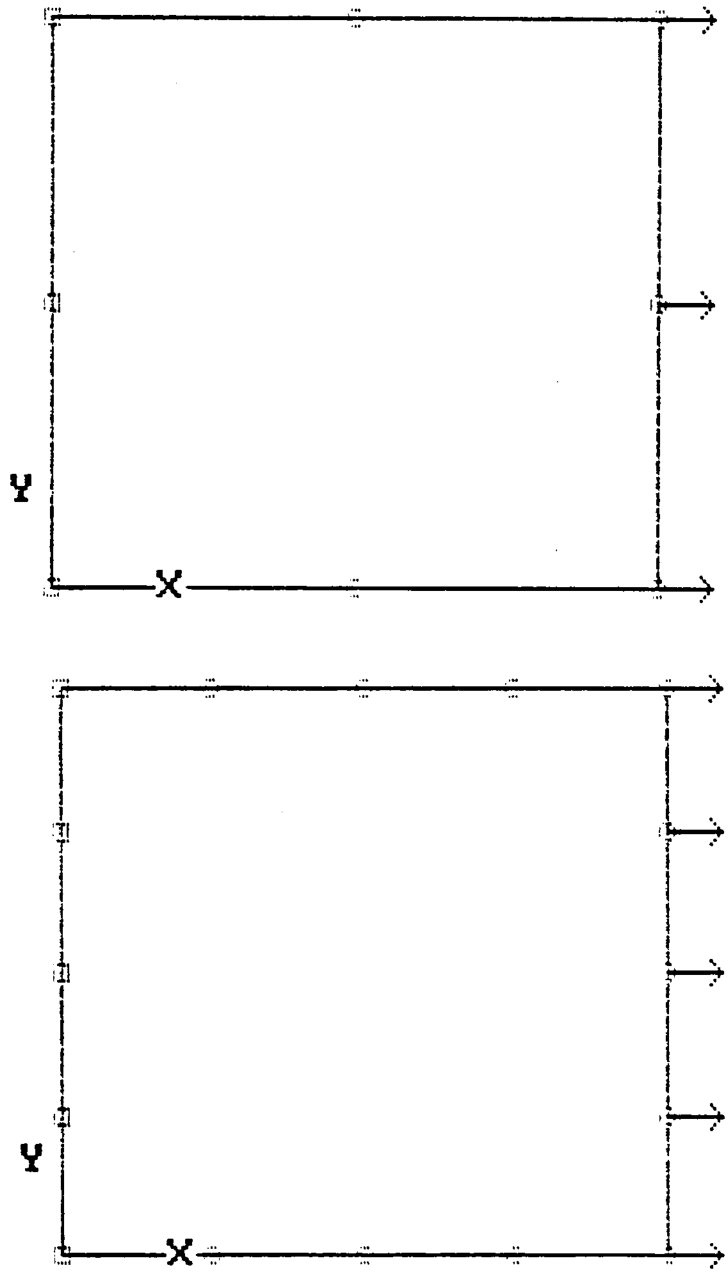

Figure 20. First and Second Discretizations of Plate Under Tension

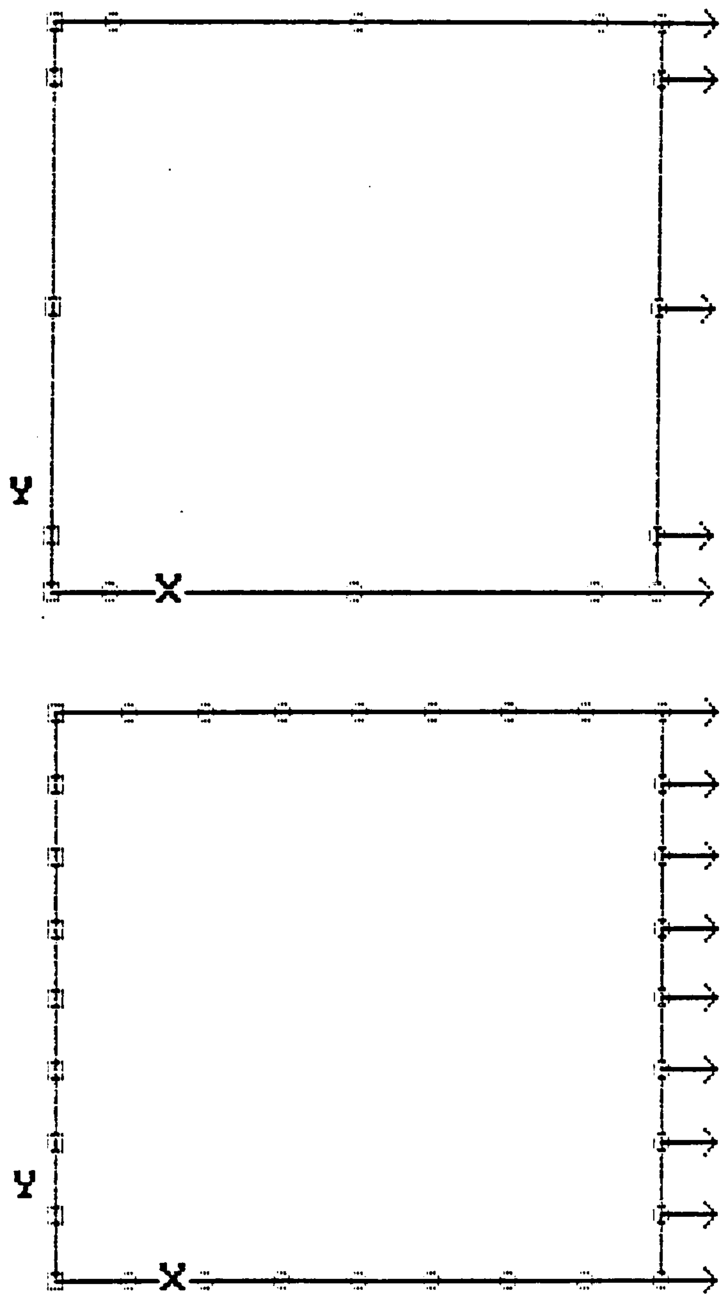

Figure 21. Third and Fourth Discretizations of Plate Under Tension

Table 4 (Part 1 of 2). Comparison of Results

QUANTITY	PROBLEM 1 8 ELEMENTS	PROBLEM 2 16 ELEMENTS	PROBLEM 3 16 ELEMENTS	PROBLEM 4 32 ELEMENTS
X DISPL. AT X=1 Y=.5 X.0001	0.941881	0.983213	1.05689	0.995647
X DISPL. AT X=.5 Y=1 X.0001	0.492282	0.497420	0.534956	0.499556
X DISPL. AT X=1 Y=1 X.0001	1.01748	1.03425	1.20180	1.02688
X DISPL. AT X=.9 Y=.9 X.0001	0.8783	0.9012	0.9598	0.9004
X STRESS AT X=1 Y=.5 X1000	1.079	0.9919	1.111	1.001
X STRESS AT X=.5 Y=1 X1000	1.026	1.075	1.036	1.010
X STRESS AT X=1 Y=1 X1000	0.7560	0.7626	1.742	0.7813
X STRESS AT X=.9 Y=.9 X1000	0.6155	0.9327	1.069	0.9686

Table 4 (Part 2 of 2). Comparison of Results

QUANTITY	PROBLEM 1 8 ELEMENTS	PROBLEM 2 16 ELEMENTS	PROBLEM 3 16 ELEMENTS	PROBLEM 4 32 ELEMENTS
STRESS AT X=.5 Y=.5	0.9177	0.9740	1.058	0.9930

the stiffness matrix and/or the coarseness of the mesh. It should be noted that the error of the displacement at the center of the right edge was only of the order of six percent, but the disturbing fact was that a similar model solved with BEM yields very nearly the exact solution. The second model, with a total of sixteen boundary elements, did much to improve the solution. The maximum error in the displacement was less than two percent, and the stresses showed some improvement, but they still were much too low near the corners. It appeared that this coupled element showed some of the same peculiarities at the corners as the constant elements did in the pure BEM program. In an effort to improve the stress calculation, the sixteen element model was revised, with the elements at the corners being made smaller. While this would seemingly improve the results over the other sixteen element model, it actually made them worse, especially the stresses. The displacements at the right edge varied by ten percent, and the stresses seemed to oscillate throughout the domain and on the edges, with the stresses being too large at the corners and too low at the center of the edges, with all variations in between. This effect could be caused by difficulties in modeling corners, and indicates that the model may be sensitive to the relative size of its elements as well as fineness of the discretization. The final model analyzed contained 32 elements, with eight equal length elements per side. The results of this model were quite good, with the displacements being very close to the exact value, and the stresses being very accurate also. Even the very corner nodes showed sizable improvements in the stresses, although they were still about 25 percent low. However, stresses calculated inside the domain near the corner were very close to the exact values.

The next stage in the testing of the coupled code involved solving truly coupled problems, namely the problem of a plate with a hole. There were two discretizations involved in this analysis, one with a 17 element boundary element model coupled with a 9 CST finite element model, and another with a 16 element boundary element model coupled with a 36 CST finite element model. These models, shown in Figure 22, both utilize one quarter symmetry. The first model, a rather coarse finite element model coupled with a boundary element model with large variations in the sizes of elements, gave reasonable displacement results, but was unsatisfactory as far as the stress results were concerned. It is believed that these stress deviations could be traced to the coarseness of the mesh, and possibly to the large variation in element length, an effect seen previously in the third problem for the plate under tension. The refinement of this mesh attempted to even the element length for all the boundary elements, and refine the finite element mesh. The results for this model, by comparison with those of the previous model, as shown in Table 5, were just as good for the displacement, but much superior for the stresses. These results also compare favorably with those obtained using an all finite element model, also shown in Table 5. The finite element model used for comparison had 154 nodes and 260 elements yielding a total of 286 degrees of freedom, while the coupled model had only 68 total degrees of freedom. This reduction in the number of degrees of freedom corresponded to a much lower solution time for the coupled model, with the coupled problem requiring 0.02 seconds of mainframe (IBM 3081) CPU time and the pure finite element solution requiring 0.2 seconds of CPU time.

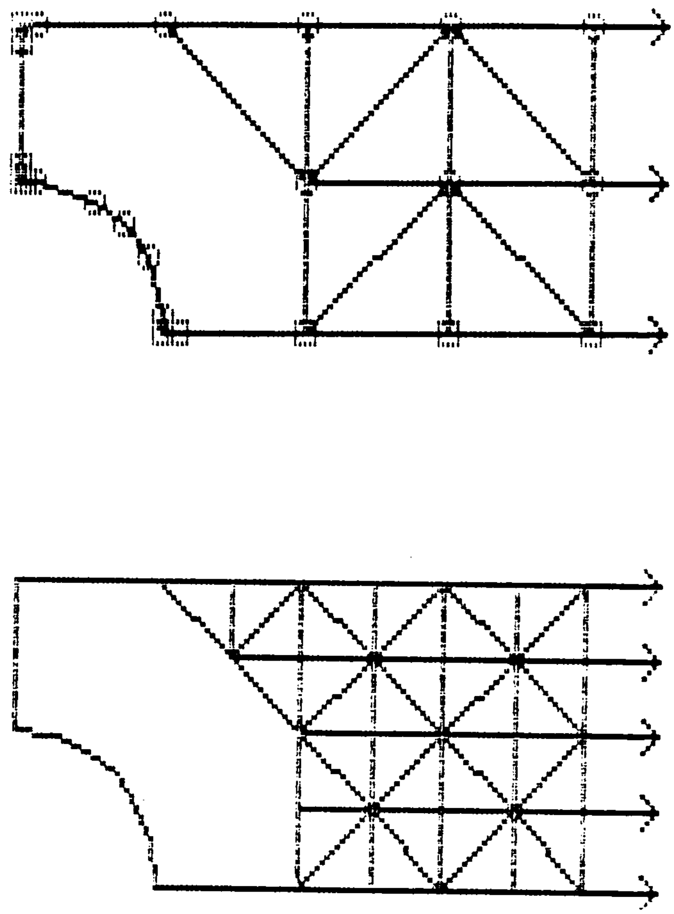

Figure 22. Coupled Discretizations for Plate With Hole Problem

Table 5. Comparison of Results for Plate With Hole

QUANTITY	COUPLED MODEL 1 RESULTS	COUPLED MODEL 2 RESULTS	PURE FEM SOLUTION
X DISPLACEMENT AT X=20 , Y=10 X.001	0.259362	0.264971	0.266245
X DISPLACEMENT AT X=20 , Y=0 X.001	0.285755	0.288247	0.286829
X DISPLACEMENT AT X=5 , Y=0 X.001	0.199264	0.198853	0.192883
X STRESS AT UPPER RIGHT CORNER X1000	1.06953	1.00749	1.03611
STRESS CONCENTRATION AT HOLE	3.253	3.133	3.25268

The coupled model also required much less storage space for the stiffness matrix, with the total number of array cells used being 846 for the coupled problem, and 6531 for the finite element model. This fact is a very important consideration when developing a program for use on a microcomputer.

Another phase of the coupled program which was examined was the use of double nodes at discontinuous boundary sections. It was found that using double nodes in the assembly of the boundary element stiffness matrix did not significantly affect the solution for displacements for any of the problems considered; however, it did modify the stresses somewhat, although not necessarily for the better. The disadvantage of using double nodes is that it did increase the CPU time for assembly by a large margin.

RECOMMENDATIONS FOR FUTURE RESEARCH

The use of the coupled FEM-BEM program could be quite advantageous for modeling complex engineering problems, such as the wing with a hole shown in Figure 23. The use of higher order boundary elements (quadratic in displacements and linear in stresses) to give a good representation of large stress and displacement gradients close to the hole, coupled with a coarse finite element mesh away from the hole, would improve not only the solution time for the problem but also the displacement and stress predictions. Hence, it is recommended that: (1) a higher order boundary element code be developed and coupled with a compatible finite element code; (2) implement of double precision arithmetic on the PC if necessary or tractable; and (3), examine the possibility of using other linkage algorithms such as those proposed by Dunbar [5] and Kamat and the author [6].

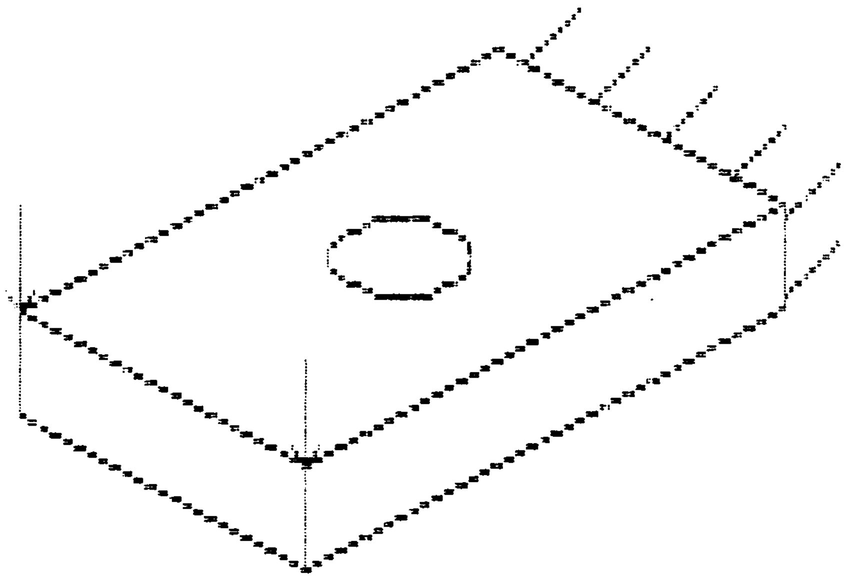

Figure 23. Complex Engineering Problem - Wing With Hole

CONCLUSION

The steps for the development of a coupled Finite Element Method - Boundary Element Method program for a personal computer have been shown. These steps included adapting a boundary element program for use on a PC, including implementation of the user-friendly features necessary for a PC-based program, the steps taken in coupling the Boundary Element Method to the Finite Element Method, and actual implementation of such a program complete with example problems for verification and analysis of the code. The work completed here has resulted in a very powerful computational tool for the PC, namely the full feature BEM program SNAP/BE, and a coupled FEM-BEM code capable of solving complex problems. The future of coupled FEM-BEM work appears promising, although more research is definitely warranted especially in the area of developing new linkage algorithms. This type of program is ideal for use on a microcomputer, since the primary difficulty in developing PC-based programs is the lack of memory. This coupling allows problems with complex geometries to be solved with reasonable accuracy and a minimum of memory and solution time requirements.

APPENDIX A. DERIVATION OF [B] MATRIX FOR CST FINITE ELEMENT

The first step is to represent the displacements over the domain of the element. A three term polynomial is used because the element has three nodes. The displacement fields for the element can be written as

$$\begin{aligned} u(x,y) &= c_1 + c_2x + c_3y \\ v(x,y) &= d_1 + d_2x + d_3y \end{aligned} \qquad (21)$$

We can write displacements u(x,y) at the element nodes as

$$\begin{aligned} u_1 = u(x_1,y_1) &= c_1 + c_2\, x_1 + c_3\, y_1 \\ u_2 = u(x_2,y_2) &= c_1 + c_2\, x_2 + c_3\, y_2 \\ u_3 = u(x_3,y_3) &= c_1 + c_2\, x_3 + c_3\, y_3 \end{aligned}$$

where (x_i,y_i), i=1,2,3 are the nodal coordinates with respect to a fixed system of global axes. Solving these equations for c_i in terms of u_i, x, and y gives

$$\begin{aligned} c_1 = [\ u_1\ (\ x_2y_3 - x_3y_2\ ) &+ u_2\ (\ x_3y_1 - x_1y_3\ ) \\ & u_3\ (\ x_1y_2 - x_2y_1\ )\ ]\ /\ 2A_e \end{aligned}$$

$$c_2 = [\ u_1\ (\ y_2 - y_3\ ) + u_2\ (\ y_3 - y_1\ ) + u_3\ (\ y_1 - y_2\ )\ ]\ /\ 2A_e$$

$$c_3 = [\ u_1\ (x_3 - x_2)\ +\ u_2\ (x_1 - x_3)\ +\ u_3\ (x_2 - x_1)\ ]\ /\ 2A_e$$

where the area of the element A_e is

$$2A_e = (x_2 y_3 - x_3 y_2) + (x_3 y_1 - x_1 y_3) + (x_1 y_2 - x_2 y_1)$$

Substituting the expressions for c_i into Equation 21 gives

$$u(x,y) = u_1 \Psi_1(x,y) + u_2 \Psi_2(x,y) + u_3 \Psi_3(x,y) = \sum_{i=1}^{3} u_i\ \Psi_i^e \tag{22}$$

where the interpolation functions Ψ_i^e are defined as

$$\Psi_i^e = (\ \alpha_i + \beta_i\ x + \gamma_i\ y\ )\ /\ 2A_e$$

with

$$\alpha_i = x_j y_k - x_k y_j\ ,\quad \beta_i = y_j - y_k\ ,\ \text{and}\quad \gamma_i = x_k - x_j$$

A similar procedure can be followed to get the expression for the displacement field v(x,y), which results in

$$v(x,y) = v_1 \Psi_1(x,y) + v_2 \Psi_2(x,y) + v_3 \Psi_3(x,y) = \sum_{i=1}^{3} v_i\ \Psi_i^e \tag{23}$$

with the Ψ_is being the same as those defined previously. Now, introduce the strain-displacement equations

$$\varepsilon_x = \frac{du}{dx} \quad , \quad \varepsilon_y = \frac{dv}{dy} \quad , \quad \varepsilon_{xy} = \frac{dv}{dx} + \frac{du}{dy} \tag{24}$$

Substitution of the expressions for u and v from Equation 22 and Equation 23 into Equation 24 gives

$$\varepsilon_x = (u_1\beta_1 + u_2\beta_2 + u_3\beta_3) / 2A_e$$

$$\varepsilon_y = (v_1\gamma_1 + v_2\gamma_2 + v_3\gamma_3) / 2A_e$$

$$\varepsilon_{xy} = (u_1\gamma_1 + v_1\beta_1 + u_2\gamma_2 + v_2\beta_2 + u_3\gamma_3 + v_3\beta_3) / 2A_e$$

These equations can be written in the familiar form

$$\{\varepsilon\} = [B] \{U\}$$

with

$$\{\varepsilon\} = [\varepsilon_x \ \ \varepsilon_y \ \ \varepsilon_{xy}]^T , \ \{U\} = [u_1 \ \ v_1 \ \ u_2 \ \ v_2 \ \ u_3 \ \ v_3]^T$$

and

$$[B] = (1 / 2A_e) \begin{bmatrix} \beta1 & 0 & \beta2 & 0 & \beta3 & 0 \\ 0 & \gamma1 & 0 & \gamma2 & 0 & \gamma3 \\ \gamma1 & \beta1 & \gamma2 & \beta2 & \gamma3 & \beta3 \end{bmatrix}$$

This [B] matrix can then be substituted into Equation 9 from the Problem Formulation Section to obtain the element stiffness matrix.

APPENDIX B. BEM EQUATIONS FOR DISPLACEMENTS AND STRESSES

The expressions for the fundamental displacement and traction tensors u^*_{ij} and p^*_{ij} for two dimensional plane strain problems are

$$u^*_{ij} = \frac{-1}{8\pi(1-\nu)G} \{ (3 - 4\nu) \ln(r) \delta_{ij} - r,_i r,_j \}$$

and

$$p^*_{ij} = \frac{-1}{4\pi(1-\nu)r} \{ [(1-2\nu) \delta_{ij} + r,_i r,_j] \frac{\partial r}{\partial n} - (1-2\nu) (r,_i n_j - r,_j n_i) \}$$

where r is the distance from the point of application of the load to the point under consideration and n_i, i=1,2,3 are the direction cosines of the outward unit normal to the surface of the body. Substitution of the above equations into Equation 13 yields the values of displacements at any interior point or any boundary point. Integration of Equation 13 is performed by Gaussian quadrature. Corresponding equations for the plane stress case are obtained by replacing Poisson's ratio ν by $\nu/1+\nu$

The equation for evaluating stresses at internal points can be obtained by using the stress-displacement relations which involve differentiating the displacements at internal points, as given by Equation 13, with respect to the coordinates of the internal points. These expressions are very lengthy and hence omitted from this appendix. The reader is referred to reference [4] for details of these expressions.

BIBLIOGRAPHY

1. Turner, M. J., Clough, R. W., Martin, H. C., and Topp, L. C., "Stiffness and Deflection Analysis of Complex Structures", Journal Aeronautical Sciences, Vol. 23, No. 4, 1956, pp.805-824.

2. Cruse, T. A., and Rizzo, F. J., "A Direct Formulation and Numerical Solution of the General Transient Elasto-Dynamic Problem", Journal Math Analysis Appl., Vol. 1, No.22, 1968, pp.244-259.

3. Brebbia, C. A.,The Boundary Element Method For Engineers, Halstead Press, New York, 1978.

4. Brebbia, C. A., Telles, J. C. F., and Wrobel, L. C., Boundary Element Techniques, Springer-Verlag, New York, 1984.

5. Dunbar, W. S., "Finite Element - Boundary Element Linkage Algorithm", Proceedings of the 5th ASCE-EMD Specialty Conference, Vol. 1, 1984, pp. 128-131.

6. Brown, S. A., and Kamat, M. P., "Coupling of Finite and Boundary Element Solutions Using a Direct Minimization Approach", Proceedings of the 7th Intl. Conf. on BEM, Villa Olms, Italy, Sept., 1985.

7. Fairweather, G., and Johnston, R. L., "The Method of Fundamental Solutions for Problems in Potential Flow", Appl. Math. Modelling, Vol. 8, 1984, pp.265-270.

8. Brown, S. A., and Kamat, M. P., "Structural Analysis of Large Scale Problems Using a Microcomputer", Proceedings of the Second National Conference on Microcomputers in Civil Engineering, 1984, pp.302-306.

9. Brown, S. A., and Kamat, M. P., "SNAP : Structural Numerical Analysis Programs for Microcomputers", VPI&SU Report, VPI-E-84-31, 1984.

10. Zienkiewicz, O. C., The Finite Element Method, Third Edition, McGraw-Hill Book Company, 1977.

11. Reddy, J. N.,An Introduction to the Finite Element Method , McGraw-Hill Book Co., New York, 1984.

12. Cooke, Walter, "Getting Started With an 8087", PC World ,December, 1983, pp.256-265.

The vita has been removed from the scanned document