AD-A133 142 AN INVESTIGATION OF THE INDIRECT BOUNDARY ELEMENT METHOD IN ONE- AND TWO-..(U) NAVAL CIVIL ENGINEERING LAB PORT HUENEME CA T A SHUGAR ET AL. MAY 83 1/2

UNCLASSIFIED NCEL-TN-1664 F/G 12/1 NL

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

AD-A133 142

TN NO: N-1664

(12)

TITLE: AN INVESTIGATION OF THE INDIRECT BOUNDARY ELEMENT METHOD IN ONE- AND TWO-DIMENSIONAL ELASTOSTATICS

AUTHOR: T. A. Shugar and J. V. Cox

DATE: May 1983

SPONSOR: Naval Material Command

PROGRAM NO: ZR000-01-179

TECHNICAL NOTE

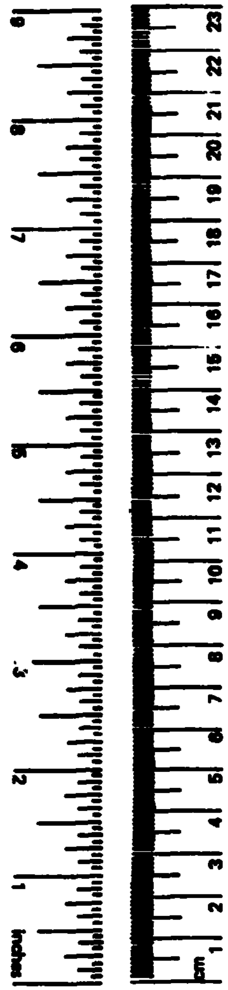
NAVAL CIVIL ENGINEERING LABORATORY
PORT HUENEME, CALIFORNIA 93043

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

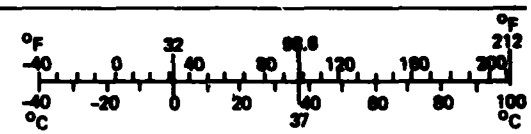
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2,000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1,000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TN-1664	2. GOVT ACCESSION NO. AD-A[illegible] DN187250	3. RECIPIENT'S CATALOG NUMBER 14L
4. TITLE (and Subtitle) AN INVESTIGATION OF THE INDIRECT BOUNDARY ELEMENT METHOD IN ONE- AND TWO-DIMENSIONAL ELASTOSTATICS		5. TYPE OF REPORT & PERIOD COVERED Not final; Oct 1981 – Sep 1982 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) T. A. Shugar and J. V. Cox		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS NAVAL CIVIL ENGINEERING LABORATORY Port Hueneme, California 93043		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N; ZR000-01-179
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Material Command Washington, D.C. 20360		12. REPORT DATE May 1983 13. NUMBER OF PAGES 114
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Boundary element, indirect, elastostatics, microcomputers.

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

This report presents a study of the indirect boundary element method and its potential advantages for solving one- and two-dimensional linear structural/stress analysis problems. The theoretical formulation of the indirect boundary element method is illustrated first by developing the framework of one-dimensional beams resting on elastic foundations, and then extending the same concept to the framework of two-dimensional plane stress or plane strain elastostatics. Computer programs are written both in BASIC and FORTRAN that numerical-

(continued)

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

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Continued

ly implement the theoretical formulations for the one-dimensional application. The program for the two-dimensional application is written in FORTRAN. Accuracy of the indirect boundary element solution is assessed through comparison with theoretical solutions and with solutions from the alternative, direct boundary element method. With regard to edge effects, the indirect boundary element method faired better than the particular direct boundary element implementation that was available for comparison. The results pertaining to the accuracy of the indirect boundary element method to correctly capture stress gradients were very encouraging, even for the constant stress elements employed. The results suggest that the method may be a very economical analysis tool for determining stress concentration factors in elastostatics. Because the necessary input data requirements are less and smaller matrices result, the boundary element methods are more suited to small computers, thus enhancing the use of the methods as effective stress analysis tools.

Library Card

CONTENTS

INTRODUCTION

This report presents a study of the indirect boundary element method and its potential advantages for solving one- and two-dimensional linear structural/stress analysis problems. Currently, the finite element method is very capable of performing these tasks. However, the boundary element method potentially offers an opportunity for increased productivity in these areas.

Background

Because the application of the boundary element method requires only that the boundary of the structure be subdivided, as contrasted with the requirement that the entire domain of the structure be subdivided when applying the finite element method, the boundary element method may increase productivity in linear structural analysis. Sometimes the former method is called a boundary method and the latter a domain method.

A good account of boundary element methods from the perspective of finite element methods can be found in Reference 1. There are two types of boundary element methods: a direct method and an indirect method. The difference between the two methods is not easily explained in a brief manner. The indirect method is rather intuitive, while the direct method is more formal. Both methods are equally effective in general. A comparison of indirect and direct boundary element methods can be found in Reference 2. An early treatment of indirect boundary methods can be found in Reference 3.

Accuracy aside for the moment, the effectiveness of any numerical stress analysis procedure depends primarily on the manual effort required for pre- and post-processing of the required input and output data, and to a lesser extent on the computer usage cost of the associated structural computer program. The boundary element method requires considerably less input and output data preparation, particularly input data, because fewer subdivisions are necessary to describe the structure boundary than the structure itself. The manual effort associated with pre- and post-processing finite element data is often very considerable and frustratingly long, notwithstanding the advantages of current automated techniques. (Obviously, such techniques are also applicable to the boundary element methods as well.) Thus, the boundary element methods offer potential savings by reducing the manual labor of the stress analyst and the engineering technician, particularly at the input data level, for a given structural problem.

The second advantage of the boundary element method involves reduced computational effort in the structural analysis computer program. In linear, static structural analysis by the finite element method it is well known that most of the computational cost lies in the solution of the system of linear algebraic equations that result from the finite element subdivision. In finite element computer programs very effective Gaussian elimination, equation-solving algorithms have evolved that minimize this cost (Ref 4). However, for the same structural analysis problem, the number of linear algebraic equations that must be solved when employing the boundary element method is generally far smaller than in the finite element method. This is because the boundary element approach immediately reduces the structural problem by one dimension due to the necessity of having only to subdivide the boundary of the structure. Thus, three-dimensional problems, as in the stress analysis of solids, are reduced to two-dimensional problems; two-dimensional problems, as in the stress analysis of membranelike plates, are reduced to one-dimensional problems; and one-dimensional problems, as in the analysis of beams, are reduced to what can be termed "point problems." In theory, this amounts to a distinct computational advantage of the boundary element method over the ubiquitous finite element method. There are mitigating considerations, however. In the case of a materially homogeneous problem, for example, the coefficient matrix in the linear algebraic system is full in the boundary element method, whereas the coefficient matrix, though much larger, is both sparse and symmetric in the finite element method.

Objective

The objective of this study is to assess the accuracy and potential of the indirect boundary element method in linear structural analysis through numerical experimentation. The indirect boundary element method is to be explicated and then demonstrated by developing a one-dimensional computer program and a two-dimensional computer program. Another objective is to determine the suitability of the method as a structural/stress analysis tool when implemented on microcomputers.

Scope

The theoretical formulation of the indirect boundary element method is illustrated first by developing the framework of one-dimensional beams resting on elastic foundations, and then extending the same concept to the framework of two-dimensional plane stress or plane strain elastostatics.

Computer programs are written both in BASIC and FORTRAN that numerically implement the theoretical formulations for the one-dimensional application. The program for the two-dimensional application is written in FORTRAN. Accuracy of the indirect boundary element solution is assessed through comparison with theoretical solutions and with solutions from the alternative, direct boundary element method.

THEORETICAL BASIS OF THE INDIRECT BOUNDARY ELEMENT METHOD

The indirect boundary element method is a general numerical solution technique for solving boundary value problems in engineering science. A bibliography is included in this report which shows the breadth of engineering boundary value problems that can be approached with the method.* The necessary theoretical relationships and equations for the method as applied to structural problems come from the theory of elasticity.

*Also see Reference 5 for a wide selection of applications.

This theoretical basis is explicated herein, first with a one-dimensional example of a beam resting on an elastic foundation, and second, with the general problem of two-dimensional elastostatics.

Once the boundary value problem has been completely stated, the numerical solution of that problem by the indirect boundary element method follows three basic steps.

1. Establish the infinite-domain Green's functions appropriate to the boundary value problem.

2. Form and solve the auxiliary boundary value problem in the infinite domain by employing superposition of the established solutions.

3. Invoke the Kirchhoff uniqueness theorem to obtain the solution to the original boundary value problem from the solution of the auxiliary problem.

Since these steps also contain information on the natural limitations of the indirect boundary element method, they are discussed below.

A Green's function is a known solution to the governing differential equation of the given boundary value problem. It is very much like an influence function, a concept familiar to undergraduate civil engineers, which algebraically determines the response, say displacement at some field point, due to a prescribed unit concentrated force at some other point, called a source point. The important concept here is that such a solution to the governing differential equation must be known at the outset for the method to be applicable.* Step 1 implies that the appropriate Green's functions must exist.

The key word in Step 2 is superposition. Thus, the boundary value problem must be linear for the indirect boundary element method to be applicable as a solution technique. It should be noted, however, that despite this limitation, some nonlinear problems are solved with boundary

*It is also true that a Green's function satisfies boundary conditions as well. See References 6 and 7 for good accounts of Green's functions.

element methods (Ref 5 and 8). These approaches must inevitably use a sequence of linearizations. Nonetheless, the principal application of the boundary element method at present is to linear problems. The solution to the auxiliary problem is built up from the superposition of unit solution components provided by the known Green's functions.

Finally, Step 3 implies that the solution to the actual problem can be obtained only if Kirchhoff's uniqueness theorem can be invoked. According to Reference 9, this theorem states:

> If, in addition to the body forces, either the surface forces or the surface displacements are given on the boundary of an elastic body, there exists only one form of equilibrium in the sense that the distribution of stresses and strains in the body is determined uniquely.

The theorem requires that the structural problem be limited to infinitesimal strains and displacements. Exactly why this theorem is invoked will become clearer when applications of the indirect boundary element method are presented in the next section.

Beam Resting on an Elastic Foundation

The necessary equations to be programmed into a computer for the numerical solution of beams on elastic foundations by the indirect boundary element are developed below. This development closely follows the account given in Reference 10. The class of problems addressed is shown in Figure 1. The boundary value problem is stated mathematically as follows. Solve the fourth order differential equation

$$EI \frac{d^4u}{dx^4} = b(x) - k\,u(x) \tag{1}$$

where u = lateral deflection

b = prescribed lateral load

k = elastic foundation stiffness per unit length

EI = beam bendi[illegible] [illegible]gid[illegible]y

The equation is to be solved subject to the following four boundary conditions:

$$\begin{aligned} &\text{at } x = 0, \quad m = 0 \quad \text{and} \quad s = 0 \\ &\text{at } x = L, \quad \theta = 0 \quad \text{and} \quad s = 0 \end{aligned} \tag{2}$$

where m = bending moment

s = shear force

θ = slope

The boundary condition at $x = 0$ implies a free end condition, and at $x = L$ a symmetry condition is implied. Other beam boundary conditions can also be imposed.

Step 1 of the procedure requires the Green's functions to be given for the above problem. These functions can be found in Reference 11 and are applicable to an infinitely long beam and foundation as shown in Figure 2. The appropriate beam response functions at a field point Q for a unit concentrated force at point P are

$$\begin{aligned} \hat{u}(r) &= \frac{\beta}{2k} e^{-\beta r} (\cos \beta r + \sin \beta r) \\ \hat{\theta}(r) &= \frac{\beta^2}{k} e^{-\beta r} \sin \beta r \cdot \operatorname{sgn}(y-x) \\ \hat{m}(r) &= \frac{e^{-\beta r}}{4\beta} (\cos \beta r - \sin \beta r) \\ \hat{s}(r) &= \frac{e^{-\beta r}}{2} \cos \beta r \cdot \operatorname{sgn}(y-x) \end{aligned} \tag{3}$$

$$\text{where } \operatorname{sgn}(y-x) = \begin{cases} 1 & y > x \\ -1 & y < x \\ \text{undefined} & y = x \end{cases}$$

$$\text{and} \qquad \beta = \sqrt[4]{k/4EI}$$

Similarly, the required Green's functions for the beam response at Q due to a unit moment at P are

$$u^*(r) = \frac{2\beta^2}{k}\hat{\theta}(r) = \frac{\beta^2}{k} e^{-\beta r} \sin \beta r \cdot \mathrm{sgn}(y-x)$$

$$\theta^*(r) = -\frac{4\beta^4}{k}\hat{m}(r) = -\frac{\beta^3}{k} e^{-\beta r} (\cos \beta r - \sin \beta r) \tag{4}$$

$$m^*(r) = \hat{s}(r) = \frac{e^{-\beta r}}{2} \cos \beta r \cdot \mathrm{sgn}(y-x)$$

$$s^*(r) = \frac{k}{\beta}\hat{u}(r) = \beta \frac{e^{-\beta r}}{2} (\cos \beta r + \sin \beta r)$$

In Step 2 the auxiliary problem is formed first by imbedding the actual beam and foundation complete with loading into the infinite domain as shown in Figure 3. It is apparent that the required boundary conditions at points 1 and 2 are not imposed in the infinite beam unless something else is done to enforce them. Therefore, the unknown forces ψ_i, and unknown moments ψ_i^*, are applied to these two points to impose the given boundary conditions. The auxiliary problem can now be stated as: find the unknown forces and moments at points 1 and 2 such that the prescribed boundary conditions of the actual problem are satisfied.

The distributed load b(x) can be resolved into N statically equivalent concentrated forces acting on the beam, and, in general, there may also be M concentrated moments acting on the actual beam. Using the Green's functions and the principle of superposition, a set of linear algebraic equations in the two unknown forces ψ_i and the two unknown moments ψ_i^* can be established such that the required boundary conditions are satisfied upon their solution. Construction of this system of equations proceeds as follows:

For $s_1 = 0$ at $x = +\varepsilon$

$$s_1 = \hat{s}(\varepsilon,0)\psi_1 + s^*(\varepsilon,0)\psi_1^* + \hat{s}(\varepsilon,L)\psi_2 + s^*(\varepsilon,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{s}(\varepsilon,x_i)b_i + \sum_{j=1}^{M} s^*(\varepsilon,x_j)b_j^* = 0$$

For $m_1 = 0$ at $x = +\varepsilon$

$$m_1 = \hat{m}(\varepsilon,0)\psi_1 + m^*(\varepsilon,0)\psi_1^* + \hat{m}(\varepsilon,L)\psi_2 + m^*(\varepsilon,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{m}(\varepsilon,x_i)b_i + \sum_{j=1}^{M} m^*(\varepsilon,x_j)b_j^* = 0$$

For $s_2 = 0$ at $x = L-\varepsilon$

$$s_2 = \hat{s}(L-\varepsilon,0)\psi_1 + s^*(L-\varepsilon,0)\psi_1^* + \hat{s}(L-\varepsilon,L)\psi_2 + s^*(L-\varepsilon,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{s}(L-\varepsilon,x_i)b_i + \sum_{j=1}^{M} s^*(L-\varepsilon,x_j)b_j^* = 0$$

For $\theta_2 = 0$ at $x = L-\varepsilon$

$$\theta_2 = \hat{\theta}(L-\varepsilon,0)\psi_1 + \theta^*(L-\varepsilon,0)\psi_1^* + \hat{\theta}(L-\varepsilon,L)\psi_2 + \theta^*(L-\varepsilon,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{\theta}(L-\varepsilon,x_i)b_i + \sum_{j=1}^{M} \theta^*(L-\varepsilon,x_j)b_j^* = 0$$

The term ε represents an arbitrarily small distance to indicate that these functions are evaluated just inside the actual or real beam domain.

In matrix form, these equations are written as

$$\begin{bmatrix} \hat{s}(\varepsilon,0) & s^*(\varepsilon,0) & \hat{s}(\varepsilon,L) & s^*(\varepsilon,L) \\ \hat{m}(\varepsilon,0) & m^*(\varepsilon,0) & \hat{m}(\varepsilon,L) & m^*(\varepsilon,L) \\ \hat{s}(L-\varepsilon,0) & s^*(L-\varepsilon,0) & \hat{s}(L-\varepsilon,L) & s^*(L-\varepsilon,L) \\ \hat{\theta}(L-\varepsilon,0) & \theta^*(L-\varepsilon,0) & \hat{\theta}(L-\varepsilon,L) & \theta^*(L-\varepsilon,L) \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_1^* \\ \psi_2 \\ \psi_2^* \end{Bmatrix} +$$

$$\left[\begin{array}{cccc|cccc} \hat{s}(\varepsilon,x_1) & \hat{s}(\varepsilon,x_2) & \dots & \hat{s}(\varepsilon,x_N) & s^*(\varepsilon,x_1) & s^*(\varepsilon,x_2) & \dots & s^*(\varepsilon,x_M) \\ \hat{m}(\varepsilon,x_1) & \hat{m}(\varepsilon,x_2) & \dots & \hat{m}(\varepsilon,x_N) & m^*(\varepsilon,x_1) & m^*(\varepsilon,x_2) & \dots & m^*(\varepsilon,x_M) \\ \hat{s}(L-\varepsilon,x_1) & \hat{s}(L-\varepsilon,x_2) & \dots & \hat{s}(L-\varepsilon,x_N) & s^*(L-\varepsilon,x_1) & s^*(L-\varepsilon,x_2) & \dots & s^*(L-\varepsilon,x_M) \\ \hat{\theta}(L-\varepsilon,x_1) & \hat{\theta}(L-\varepsilon,x_2) & \dots & \hat{\theta}(L-\varepsilon,x_N) & \theta^*(L-\varepsilon,x_1) & \theta^*(L-\varepsilon,x_2) & \dots & \theta^*(L-\varepsilon,x_M) \end{array}\right] \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \\ b_1^* \\ b_2^* \\ \vdots \\ b_M^* \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \qquad (5)$$

or

$$\underset{\sim}{\underset{\sim}{G}}\,\underset{\sim}{\psi} + \underset{\sim}{\underset{\sim}{H}}\,\underset{\sim}{b} = \underset{\sim}{F} \tag{5a}$$

The solution to the auxiliary problem is then given by

$$\underset{\sim}{\psi} = \underset{\sim}{\underset{\sim}{G}}^{-1}(\underset{\sim}{F} - \underset{\sim}{\underset{\sim}{H}}\,\underset{\sim}{b}) \tag{6}$$

It is interesting to note that the order of the coefficient matrix G is only four, and that this will always be the case independently of the beam and foundation length so long as the they are prescribed homogeneous. Thus, the solution will always require that matrices of only order four be inverted. The one-dimensional boundary value problem has been reduced to a discrete boundary element problem involving only two points in the domain – the boundary points 1 and 2 of the actual beam and foundation problem. This is in contrast to a standard finite element solution approach that would have reduced the continuous problem to a discrete problem at n points (nodes) in the domain along the beam.

For all practical purposes the greater part of the solution effort in the indirect boundary element method is accomplished in Step 2. Since the forces and moments at the end points are now known for the auxiliary problem, and since they, by definition, impose the prescribed boundary conditions in the infinite domain, Step 3 can be addressed. Kirchhoff's uniqueness theorem simply specifies that the solution in the domain [0,L] of the actual problem is the same as the solution for the domain [0,L] of the auxiliary problem. The solution of the actual problem can then be obtained by superposition of the Green's functions applied to the auxiliary problem as follows.

Suppose the beam response is desired at some set of K points within the domain [0,L]. Then for k = 1,2,..., K the displacement, moment, slope, and shear force can be written directly as

$$u(x_k) = \hat{u}(x_k,0)\psi_1 + \hat{u}(x_k,L)\psi_2 + u^*(x_k,0)\psi_1^* + u^*(x_k,L)\psi_2^*$$
$$+ \sum_{i=1}^{N} \hat{u}(x_k,x_i)b_i + \sum_{j=1}^{M} u^*(x_k,x_j)b_j^*$$

$$m(x_k) = \hat{m}(x_k,0)\psi_1 + \hat{m}(x_k,L)\psi_2 + m^*(x_k,0)\psi_1^* + m^*(x_k,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{m}(x_k,x_i)b_i + \sum_{j=1}^{M} m^*(x_k,x_j)b_j^*$$

$$\theta(x_k) = \hat{\theta}(x_k,0)\psi_1 + \hat{\theta}(x_k,L)\psi_2 + \theta^*(x_k,0)\psi_1^* + \theta^*(x_k,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{\theta}(x_k,x_i)b_i + \sum_{j=1}^{M} \theta^*(x_k,x_j)b_j^*$$

$$s(x_k) = \hat{s}(x_k,0)\psi_1 + \hat{s}(x_k,L)\psi_2 + s^*(x_k,0)\psi_1^* + s^*(x_k,L)\psi_2^*$$

$$+ \sum_{i=1}^{N} \hat{s}(x_k,x_i)b_i + \sum_{j=1}^{M} s^*(x_k,x_j)b_j^*$$

In matrix form these equations are

$$\begin{Bmatrix} u(x_k) \\ m(x_k) \\ \theta(x_k) \\ s(x_k) \end{Bmatrix} = \begin{bmatrix} \hat{u}(x_k,0) & \hat{u}(x_k,L) & u^*(x_k,0) & u^*(x_k,L) \\ \hat{m}(x_k,0) & \hat{m}(x_k,L) & m^*(x_k,0) & m^*(x_k,L) \\ \hat{\theta}(x_k,0) & \hat{\theta}(x_k,L) & \theta^*(x_k,0) & \theta^*(x_k,L) \\ \hat{s}(x_k,0) & \hat{s}(x_k,L) & s^*(x_k,0) & s^*(x_k,L) \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \\ \psi_1^* \\ \psi_2^* \end{Bmatrix} +$$

$$\left[\begin{array}{cccc|cccc} \hat{u}(x_k,x_1) & \hat{u}(x_k,x_2) & \ldots & \hat{u}(x_k,x_N) & u^*(x_k,x_1) & u^*(x_k,x_2) & \ldots & u^*(x_k,x_M) \\ \hat{m}(x_k,x_1) & \hat{m}(x_k,x_2) & \ldots & \hat{m}(x_k,x_N) & m^*(x_k,x_1) & m^*(x_k,x_2) & \ldots & m^*(x_k,x_M) \\ \hat{\theta}(x_k,x_1) & \hat{\theta}(x_k,x_2) & \ldots & \hat{\theta}(x_k,x_N) & \theta^*(x_k,x_1) & \theta^*(x_k,x_2) & \ldots & \theta^*(x_k,x_M) \\ \hat{s}(x_k,x_1) & \hat{s}(x_k,x_2) & \ldots & \hat{s}(x_k,x_N) & s^*(x_k,x_1) & s^*(x_k,x_2) & \ldots & s^*(x_k,x_M) \end{array}\right] \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \\ \hline b_1^* \\ b_2^* \\ \vdots \\ b_M^* \end{Bmatrix} \quad (7)$$

Note that no further inversion of matrices is necessary. The above equations are merely algebraic equations giving the desired solution to the actual beam and foundation problem at a set of prescribed points.

The rectangular matrix is the largest matrix to be formed and it is order 4x(N+M), where N is the number of concentrated or statically equivalent lateral forces prescribed on the beam, and M is the number of concentrated moments prescribed along the length of the beam.

With the largest matrix to be inverted (see Step 2) an order four matrix, and the largest matrix to be formed (but not necessarily stored) a 4x(N+M) matrix, the indirect boundary method appears to be ideally suited to microcomputers.

Two-Dimensional Elastostatics

The application of the indirect boundary element method presented here is for the calculation of stresses and displacements in the plane of two-dimensional elastic plates that are subject to stretching due to prescribed edge forces and edge displacements. The form of this presentation follows that given in Reference 12. Figure 4 illustrates the general elastostatics problem under consideration. The normal stress components are σ_x and σ_y, and the shear stress component is τ_{xy}, relative to the x-y coordinate system shown. The corresponding displacement components are u and v. The mechanical properties of the plate are defined by the modulus of elasticity, E, and Poisson's ratio, μ.

The formal boundary value problem can be stated as the requirement for a solution to the two-dimensional stress equilibrium partial differential equations (ignoring body forces) in the domain Ω:

$$\left.\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} &= 0 \\ \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} &= 0 \end{aligned}\right\} \text{ in } \Omega \tag{8}$$

And further, that the solution also satisfy the stress and displacement boundary conditions

$$\sigma_n = \hat{F}_n \quad \text{and} \quad \sigma_t = \hat{F}_t \quad \text{on } \Gamma_F$$
$$u = \hat{u} \quad \text{and} \quad v = \hat{v} \quad \text{on } \Gamma_D \tag{9}$$

where $\hat{F}_n$ and $\hat{F}_t$ are prescribed tractions, in the normal and tangential directions, respectively, along a section of the boundary Γ_F, and $\hat{u}$ and $\hat{v}$ are the prescribed components of displacement in the x and y directions, respectively, along the remaining section of the boundary Γ_D.

The indirect boundary element approach to the solution begins with the establishment of the Green's functions appropriate to the governing partial differential equation. In this case, these functions are solutions for the three stress components and the two displacement components at a field point Q due to a unit concentrated force at a source point P in the plane of an infinite two-dimensional elastic region as shown in Figure 5. The following functions can be found in Reference 12, and pertain to plane strain conditions (i.e., the strain component normal to the flat plate is zero, and the stress component normal to the plate is, in general, nonzero). These functions are generally given in terms of polar coordinates in standard references on the theory of elasticity; see, for example, Reference 13. Nonetheless, the Green's functions in the Cartesian xy system are

$$\sigma_x(Q,P) = \frac{\cos\theta}{2\pi}\left[-\frac{2\lambda + 3G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G}\frac{y^2}{r^2}\right]\frac{x}{r^2} + \frac{\sin\theta}{2\pi}\left[\frac{G}{\lambda + 2G} - \frac{2(\lambda + G)}{\lambda + 2G}\frac{x^2}{r^2}\right]\frac{y}{r^2} \tag{10a}$$

$$\sigma_y(Q,P) = \frac{\sin\theta}{2\pi}\left[-\frac{2\lambda + 3G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G}\frac{x^2}{r^2}\right]\frac{y}{r^2} + \frac{\cos\theta}{2\pi}\left[\frac{G}{\lambda + 2G} - \frac{2(\lambda + G)}{\lambda + 2G}\frac{y^2}{r^2}\right]\frac{x}{r^2} \tag{10b}$$

$$\tau_{xy}(Q,P) = -\frac{\cos\theta}{2\pi}\left[\frac{G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G}\frac{x^2}{r^2}\right]\frac{y}{r^2}$$

$$-\frac{\sin\theta}{2\pi}\left[\frac{G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G}\frac{y^2}{r^2}\right]\frac{x}{r^2} \tag{10c}$$

$$u(Q,P) = \frac{\cos\theta}{2\pi}\left[-\frac{\lambda + 3G}{2G(\lambda + 2G)}\ln r - \frac{\lambda + G}{2G(\lambda + 2G)}\frac{y^2}{r^2}\right]$$

$$+\frac{\sin\theta}{2\pi}\left[\frac{\lambda + G}{2G(\lambda + 2G)}\frac{xy}{r^2}\right] \tag{11a}$$

$$v(Q,P) = \frac{\sin\theta}{2\pi}\left[-\frac{\lambda + 3G}{2G(\lambda + 2G)}\ln r - \frac{\lambda + G}{2G(\lambda + 2G)}\frac{x^2}{r^2}\right]$$

$$+\frac{\cos\theta}{2\pi}\left[\frac{\lambda + G}{2G(\lambda + 2G)}\frac{xy}{r^2}\right] \tag{11b}$$

where $$r = \sqrt{x^2 + y^2} \tag{12a}$$

$$\lambda = \frac{E\,\mu}{(1 + \mu)(1 - 2\mu)} \tag{12b}$$

$$G = \frac{E}{2(1 + \mu)} \,. \tag{12c}$$

Though these equations are for plane strain conditions, they can easily be converted to apply to plane stress conditions with the substitution everywhere of λ' for λ where

$$\lambda' = \frac{2\,\lambda\,G}{\lambda + 2G} \tag{13}$$

In that event the class of problems being considered would be such that the stress component normal to the plane is zero, and the strain component normal to the plane is nonzero.

Step 2 of the indirect boundary element procedure begins with the formation of the auxiliary problem by scribing the actual problem boundary on the infinite sheet of the same material and thickness. An unknown system of normal and tangential tractions, P_n and P_t, are applied to the scribed boundary in the plane. These forces are shown in Figure 6. The specific task is to solve the auxiliary problem for this unknown system such that the given boundary conditions are satisfied in the infinite sheet.

The Green's functions (Equations 10 and 11) and superposition are employed to write equations relating the stresses and displacements at a field point Q on the scribed boundary (but also in the infinite sheet) in terms of the unknown forces at a source point P as P is successively moved around the entire scribed boundary. That is, the influence of the tractions at each source point P on the boundary is superimposed at the field point Q in forming the equations for the total response at field point Q. Since the unknown force system is continuous, the superposition is accomplished by an integral on the boundary, and a boundary integral equation corresponding to the field point Q results.

Since the integration is around the entire boundary ℓ, there will occur a singularity condition when the source point P and the field point Q coincide. To provide for this instance, the boundary integral is divided into two integrals, one along a small segment $\Delta\ell$ that contains the field point Q, and the other along the remaining portion of the boundary $\ell - \Delta\ell$. The contribution of the former integral is evaluated in the limit as the distance r between P and Q vanishes. Determining the singularity value of the integrals is a lengthy mathematical exercise that is omitted here. However, this singularity contribution is important as it will form block diagonal entries (or submatrices) in the coefficient matrix of the system of algebriac equations that will result from the numerical implementation.

The boundary integral equations thusly formed for an arbitrary boundary point Q are given below in Equations 14 and 15, where the first terms on the right-hand side in brackets are contributed by the singularity condition.

$$(\sigma_x)_Q = \left[\frac{\lambda + 2G \sin^2 \theta}{2(\lambda + 2G)} P_n - \sin\theta \cos\theta P_t\right]_P$$

$$+ \int_{\ell-\Delta\ell} (- P_n \sin\theta + P_t \cos\theta)_\ell \left[- c_o c_1 \left(\frac{x_Q - x_P}{r^2}\right) + c_2 \frac{(x_Q - x_P)(y_Q - y_P)^2}{r^4}\right] d\ell$$

$$+ \int_{\ell-\Delta\ell} (P_n \cos\theta + P_t \sin\theta)_\ell \left[c_1\left(\frac{y_Q - y_P}{r^2}\right) - c_2 \frac{(x_Q - x_P)^2(y_Q - y_P)}{r^4}\right] d\ell \qquad (14a)$$

$$(\sigma_y)_Q = \left[\frac{\lambda + 2G \cos^2 \theta}{2(\lambda + 2G)} P_n + \sin\theta \cos\theta P_t\right]_P$$

$$+ \int_{\ell-\Delta\ell} (P_n \cos\theta + P_t \sin\theta)_\ell \left[- c_o c_1 \left(\frac{y_Q - y_P}{r^2}\right) + c_2 \frac{(x_Q - x_P)^2(y_Q - y_P)}{r^4}\right] d\ell$$

$$+ \int_{\ell-\Delta\ell} (-P_n \sin\theta + P_t \cos\theta)_\ell \left[c_1\left(\frac{x_Q - x_P}{r^2}\right) - c_2 \frac{(x_Q - x_P)(y_Q - y_P)^2}{r^4}\right] d\ell \qquad (14b)$$

$$(\tau_{xy})_Q = \left[- \frac{G}{\lambda + 2G} \sin\theta \cos\theta P_n - \frac{1}{2} (\sin^2 \theta - \cos^2 \theta) P_t\right]_P$$

$$- \int_{\ell-\Delta\ell} (-P_n \sin\theta + P_t \cos\theta)_\ell \left[c_1\left(\frac{y_Q - y_P}{r^2}\right) + c_2 \frac{(x_Q - x_P)^2(y_Q - y_P)}{r^4}\right] d\ell$$

$$- \int_{\ell-\Delta\ell} (P_n \cos\theta + P_t \sin\theta)_\ell \left[c_1\left(\frac{x_Q - x_P}{r^2}\right) + c_2 \frac{(x_Q - x_P)(y_Q - y_P)^2}{r^4}\right] d\ell \qquad (14c)$$

$$(u)_Q = \left[c_3 \ln \frac{\Delta\ell}{2} - 1 + c_4\right] \Delta\ell \sin\theta P_n - c_3\left(\ln \frac{\Delta\ell}{2} - 1\right) \Delta\ell \cos\theta P_t$$

$$+ \int_{\ell-\Delta\ell} (-P_n \sin\theta + P_t \cos\theta)_\ell \left[- c_3 \ln r - c_4 \frac{(y_Q - y_P)^2}{r^2}\right] d\ell$$

$$+ \int_{\ell-\Delta\ell} (P_n \cos\theta + P_t \sin\theta)_\ell \left[c_4 \frac{(x_Q - x_P)(y_Q - y_P)}{r^2}\right] d\ell \qquad (15a)$$

$$(v)_Q = \left[- c_3\left(\ln \frac{\Delta\ell}{2} - 1\right) - c_4 \right] \Delta\ell \cos\theta\, P_n - c_3\left(\ln \frac{\Delta\ell}{2} - 1\right) \Delta\ell \sin\theta\, P_t$$

$$+ \int_{\ell-\Delta\ell} (P_n \cos\theta + P_t \sin\theta)_\ell \cdot \left[c_3 \ln r - c_4 \frac{(x_Q - x_P)^2}{r^2} \right] d\ell$$

$$+ \int_{\ell-\Delta\ell} (-P_n \sin\theta + P_t \cos\theta)_\ell \left[c_4 \frac{(x_Q - x_P)(y_Q - y_P)}{r^2} \right] d\ell \qquad (15b)$$

where $c_o = 1 + \frac{2(\lambda + G)}{G}$

$$c_1 = \frac{G}{2\pi(\lambda + 2G)}$$

$$c_2 = \frac{\lambda + G}{\pi(\lambda + 2G)}$$

$$c_3 = \frac{\lambda + 3G}{4\pi G(\lambda + 2G)}$$

$$c_4 = \frac{\lambda + G}{4\pi G(\lambda + 2G)}$$

At an arbitrary point Q on the boundary, the prescribed boundary condition either for stress or for displacement is imposed. The unknown force distributions P_n and P_t appearing on the right-hand sides of Equations 14 or 15 will correspond to these conditions. The equilibrium of a material point Q on Γ_F is considered with the aid of Figure 7. Equating the horizontal and vertical forces to zero on the stress block shown yields the two following equations:

$$(\sigma_x)_Q \Delta y - (\tau_{xy})_Q \Delta x + \hat{F}_t \cos\theta \Delta s - \hat{F}_n \sin\theta \Delta s = 0$$

$$- (\sigma_y)_Q \Delta x + (\tau_{xy})_Q \Delta y + \hat{F}_t \sin\theta \Delta s + \hat{F}_n \cos\theta \Delta s = 0$$

All variables refer to the point Q. These equations can easily be simplified to the following form:

$$(\sigma_x)_Q \sin^2\theta + (\sigma_y)_Q \cos^2\theta - 2(\tau_{xy})_Q \cos\theta \sin\theta = \hat{F}_n$$

$$\left[(\sigma_y)_Q - (\sigma_x)_Q\right] \cos\theta \sin\theta - (\tau_{xy})_Q (\sin^2\theta - \cos^2\theta) = \hat{F}_t \qquad (16)$$

Though Equations 16 appear to contain three unknowns, $(\sigma_x)_Q$, $(\sigma_y)_Q$, and $(\tau_{xy})_Q$, there are only two actual unknowns, since by substituting Equations 14 for these variables, the left-hand sides then contain only P_n and P_t as unknowns. In this way two equations in two unknowns are written for each point Q on the boundary Γ_F.

Likewise at each point Q on Γ_D, two equations in the unknowns P_n and P_t are obtained since

$$u_Q = \hat{u}$$

$$v_Q = \hat{v} \qquad (17)$$

and substitution of Equations 15 for u_Q and v_Q leaves the left-hand sides expressed in terms of P_n and P_t.

The numerical implementation of the boundary integral equations proceeds as follows. The entire boundary is subdivided into N straight line segments at the center of which the pair of equations in either Equations 16 or 17 is applied. The result is a system of 2N boundary integral equations containing the unknown tractions P_n and P_t acting over each segment. P_n and P_t are interpolated at the same N points, and 2N equations in 2N unknown discrete values of P_n and P_t are obtained. Interpolation of the unknown traction distribution at the N center points is consistent with assuming the tractions to be constant over the segment. Other numerical schemes can be constructed also. For example, the unknown tractions can be interpolated assuming a linear or parabolic distribution over the segment. In this study the constant interpolation scheme was implemented, although the equations were also developed for linear interpolation.

Once the interpolation scheme has been applied, the discrete variables P_n and P_t can be factored outside the integrands. The remaining integral expressions, when evaluated over each segment, form the coefficients of these discrete variables in the system of algebraic equations. The integrations were carried out analytically in this study, as was the evaluation of the singularity contribution given in Equations 14 and 15. The integrals could have been evaluated numerically instead. However, such an approach essentially would have resulted in a numerical integration over each segment occurring within a straight line approximation of the boundary (i.e., an approximation within an approximation). Again, the details of the analytical integration are omitted. The system of 2N linear algebraic equations which results is

$$\underset{\approx}{K}\,\underset{\sim}{P} = \hat{\underset{\sim}{B}} \tag{18}$$

The square coefficient matrix $\underset{\approx}{K}$ contains the results of the analytical integration of the singularities in 2x2 blocks on the diagonal and the analytical integrations of the terms other than the singularities in the off-diagonal blocks. The vector $\underset{\sim}{P}$ contains unknown variables P_n and P_t at each of the N points. The vector $\hat{\underset{\sim}{B}}$ contains the prescribed boundary values of either the tractions ($\hat{F}_n$ and $\hat{F}_t$) or the displacements ($\hat{u}$ and $\hat{v}$) at each of the N points. Some details of the numerical implementation and these matrices may be found in Appendix A.

The solution of this system yields 2N values for the unknown tractions on the scribed boundary such that the prescribed boundary conditions are imposed in the infinite sheet. The solution is written as

$$\underset{\sim}{P} = \underset{\approx}{K}^{-1}\,\hat{\underset{\sim}{B}} \tag{19}$$

In this study the solution was carried out by a Gaussian elimination algorithm with partial pivoting. It should be mentioned that for homogeneous problems the coefficient matrix $\underset{\approx}{K}$ is both full and nonsymmetric. It has no special structure that could otherwise have been exploited as in finite element solutions of the same class of problems. The solution, Equation 19, concludes Step 2 of the indirect boundary element procedure.

Step 3 is described as follows. The prescribed displacement and stress conditions of the actual problem have been imposed on the scribed boundary in the infinite sheet of the auxiliary problem. The Kirchoff uniqueness theorem is invoked, which specifies that the solution to the actual problem is therefore identical to the solution of the auxiliary problem. As a result, the stresses and displacements on and within the scribed boundary of the infinite sheet are identical to the stresses and displacements on and within the finite elastic sheet of the actual problem.

Given the 2N values for P_n and P_t, the stress or displacement response at any prescribed field point Q within the scribed boundary can be found using the appropriate Green's functions, Equations 10 or 11. Referring to Figure 8, the stress and displacement responses at a prescribed field point Q are given by the following equations. The stresses at Q are

$$(\sigma_x)_Q = \sum_{i=1}^{N} \frac{(-\hat{P}_{ni} \sin\theta_i + \hat{P}_{ti} \cos\theta_i)}{2\pi} \left[-\frac{2\lambda + 3G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G} \left(\frac{y^2}{r^2}\right)_i \right] \left(\frac{x}{r^2}\right)_i$$

$$+ \sum_{i=1}^{N} \frac{(\hat{P}_{ni} \cos\theta_i + \hat{P}_{ti} \sin\theta_i)}{2\pi} \left[\frac{G}{\lambda + 2G} - \frac{2(\lambda + G)}{\lambda + 2G} \left(\frac{x^2}{r^2}\right)_i \right] \left(\frac{y}{r^2}\right)_i \qquad (20a)$$

$$(\sigma_y)_Q = \sum_{i=1}^{N} \frac{(\hat{P}_{ni} \cos\theta_i + \hat{P}_{ti} \sin\theta_i)}{2\pi} \left[-\frac{2\lambda + 3G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G} \left(\frac{x^2}{r^2}\right)_i \right] \left(\frac{y}{r^2}\right)_i$$

$$+ \sum_{i=1}^{N} \frac{(-\hat{P}_{ni} \sin\theta_i + \hat{P}_{ti} \cos\theta_i)}{2\pi} \left[\frac{G}{\lambda + 2G} - \frac{2(\lambda + G)}{\lambda + 2G} \left(\frac{y^2}{r^2}\right)_i \right] \left(\frac{x}{r^2}\right)_i \qquad (20b)$$

$$(\tau_{xy})_Q = -\sum_{i=1}^{N} \frac{(-\hat{P}_{ni} \sin\theta_i + \hat{P}_{ti} \cos\theta_i)}{2\pi} \left[\frac{G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G} \left(\frac{x^2}{r^2}\right)_i \right] \left(\frac{y}{r^2}\right)_i$$

$$- \sum_{i=1}^{N} \frac{(\hat{P}_{ni} \cos\theta_i + \hat{P}_{ti} \sin\theta_i)}{2\pi} \left[\frac{G}{\lambda + 2G} + \frac{2(\lambda + G)}{\lambda + 2G} \left(\frac{y^2}{r^2}\right)_i \right] \left(\frac{x}{r^2}\right)_i \qquad (20c)$$

The displacements at Q are

$$(u)_Q = \sum_{i=1}^{N} \frac{(-\hat{P}_{ni} \sin \theta_i + \hat{P}_{ti} \cos \theta_i)}{2\pi} \left[- \frac{\lambda + 3G}{2G(\lambda + 2G)} \ln r_i - \frac{\lambda + G}{2G(\lambda + 2G)} \left(\frac{y^2}{r^2} \right)_i \right]$$

$$+ \sum_{i=1}^{N} \frac{(\hat{P}_{ni} \cos \theta_i + \hat{P}_{ti} \sin \theta_i)}{2\pi} \left[\frac{\lambda + G}{2G(\lambda + 2G)} \left(\frac{xy}{r^2} \right)_i \right] \tag{21a}$$

$$(v)_Q = \sum_{i=1}^{N} \frac{(\hat{P}_{ni} \cos \theta_i + \hat{P}_{ti} \sin \theta_i)}{2\pi} - \left[\frac{\lambda + 3G}{2G(\lambda + 2G)} \ln r_i - \frac{\lambda + G}{2G(\lambda + 2G)} \left(\frac{x^2}{r^2} \right)_i \right]$$

$$+ \sum_{i=1}^{N} \frac{(-\hat{P}_{ni} \sin \theta_i + \hat{P}_{ti} \cos \theta_i)}{2\pi} \left[\frac{\lambda + G}{2G(\lambda + 2G)} \left(\frac{xy}{r^2} \right)_i \right] \tag{21b}$$

COMPUTATIONAL PERFORMANCE OF THE INDIRECT BOUNDARY ELEMENT METHOD

Beam Resting on an Elastic Foundation

A numerical study was conducted by comparing computed results from two indirect BEM programs with theoretical solutions for beam-on-an-elastic-foundation problems. The first program was written in FORTRAN and implemented on a Cyber 175 mainframe computer. The second program, developed independent of the first, was written in Applesoft BASIC on an Apple II Plus microcomputer. Other boundary element computer programs have been written for microcomputers (Ref 14 and 15), but the authors are unaware of any based upon the indirect method for stress analysis.

There are some differences in these two programs. The FORTRAN version contains closed-form solutions for full-span uniform and bilinear varying loading, requires batch input, and assumes free-end boundary conditions. The BASIC version approximates both uniform and linear varying loads (partial or full span) with equally spaced concentrated loads. The input is interactive, and each set of boundary conditions can be specified in terms of displacement, rotation, moment, or shear. The larger real word size of the Cyber 175 computer (60 bits versus 40 bits for the Apple) and the closed-form solution for linear varying loads contribute to the better accuracy of FORTRAN versions. The BASIC version has more flexibility with interactive input, more general loading, and the ability to model more complex boundary conditions.

The problems shown in Figures 9 through 14 are the subjects of the numerical study for one-dimensional applications. These problems include concentrated loads, a partial uniform load, and exploit symmetry conditions. Appendix B includes additional problems for the BASIC version, showing linear varying loads and asymmetry modeling. The listing of the BASIC program is included in Appendix C.

As shown in Figure 9, the first problem consist of a 50-kip load on the center of a beam. The moment and deflection at the center of the beam are compared with theory. The results show both programs agreeing exactly (to the indicated accuracy) with theory for deflection at the midspan. Both programs give accurate results for the moment value, the FORTRAN version being the most accurate. Figure 10 demonstrates the use of symmetry in the solution of the same problem.

The mainframe version of the program executed 100 times faster than the microcomputer version. Using a compiled version of the BASIC program nearly doubled its speed. Though the programs were constructed independently, this does give a rough estimate of the speed difference between the mainframe and this generation of microcomputer.

Figure 11 illustrates the second example involving two symmetric concentrated loads. The moment and deflection at the center of the beam are compared with theory. Figure 12 shows the model of the problem incorporating symmetry boundary conditions. Both programs give equally good results. The symmetric modeling gives the same values as the full beam model.

Figures 13 and 14 illustrate the third example involving a partial uniform load symmetric about the centerline. The moment and deflection at the center of the beam and the slope at the ends of the beam are compared with theory. The BASIC program gives better results for the end slopes, while the FORTRAN program gives better results for the centerline moment. The accuracy of the bending moment calculation from the BASIC version using symmetry is equal to that from the FORTRAN version. The full model uses 20 concentrated loads to approximate the partial uniform load. The symmetry model uses both 10 and 20 concentrated loads to approximate the partial uniform load. All models indicate about the same accuracy, indicating that 10 concentrated loads are adequate to represent the partial uniform load in the symmetry model. The interactive mode, allowable with the BASIC version, makes the microcomputer version more valuable as an engineering tool.

Other examples presented in Appendix C include a partial span, linear varying load, and antisymmetric loading. The inversion test (see Appendix B for an explanation) indicated that the solution might have been inaccurate. A possible problem with the boundary integral method is that the principal coefficient matrix ([G] in this case) contains terms that can vary by orders of magnitude. The printout of the [G] matrix indicates zero terms (to three decimal places) for the displacement and rotation rows. The matrix might require preconditioning to ensure consistently good numerical accuracy.

Of the 11 problems studied, all but one indicate the BEM to be very accurate for the beam-on-an-elastic-foundation problems. The only approximations made in the algorithms were the reduction of continuous loads to a series of concentrated loads in the BASIC program.

Two-Dimensional Elastostatics

A brief numerical study was conducted on three simple example problems in which the solutions from an indirect BEM program, a direct BEM program, and theory were compared. Both programs incorporated constant boundary elements. The indirect BEM program (BIM2D) was developed as part of this study, while the direct BEM program (PGM18) was developed by C. A. Brebbia at Southampton and is described in Reference 16. Appendix D contains a listing of the BIM2D program.

In the boundary element method, most of the computer time is spent integrating the boundary integrals over the elements to form the system matrix $\underset{\sim}{K}$ and calculating internal responses depending on how much internal information is sought. In the finite element method the solution of the system of algebraic equations is usually the longest calculation.

In addition to accuracy considerations, BIM2D was developed using analytical integrations over the elements to minimize the cost of boundary integrations. This offered better accuracy for a given cost. Integrations within the direct BEM program are carried out numerically except for the singularity coefficients - diagonal blocks of the system matrix. Several observations drawn from the numerical study are discussed below.

The ripple effect (as shown in Figures 15 and 16) occurs near a boundary where a relatively few number of elements are used. The greatest deviations occurred near abrupt geometry discontinuities (corners) and loading discontinuities.

The first problem studied was a 1-inch-thick square plate in hydrostatic tension. As shown in Figure 17 the model consisted of 12 elements of unit length on a side, all with a prescribed normal traction of 1,000 psi. Internal stresses were calculated in the upper quarter of the plate. The stresses at four lines of 30 response points each are shown plotted in Figure 15.

Except for the edge region, within one element length of the boundary, the direct method algorithm is more accurate. Excluding the edge region, the indirect algorithm's error, 5.6% and less, is on the order of 10 times that of the direct algorithm's error. In the edge region the indirect algorithm is more accurate, and this is the critical region in most applications. The direct method behaves pathologically near the boundary, while the indirect algorithm is relatively stable. Both methods exhibit a ripple effect or artificial stress oscillation, but it is much more pronounced in the direct algorithm. It is interesting that the normal stress computed from the direct method changes direction of instability near the corner. That is, σ_y goes one way and σ_x goes the other.

The indirect method always underestimates the response where the stresses are largest. For most response lines examined the response begins with a nearly constant accuracy, decreases in accuracy, and then recovers near the boundary. The presence of the boundary influences a larger region in the indirect algorithm, but this effect is less severe.

The second problem studied was a 1-inch-thick rectangular plate subjected to 1,000 psi hydrostatic tension. As shown in Figure 18, the model consisted of two 20-element sides and two 4-element sides. Each element had a length of unity. The internal responses were calculated in the upper quarter of the plate. The stresses along four lines of 50 response points each were calculated and are shown plotted in Figure 16. This problem was also the subject of a numerical study conducted by John Bode (Ref 12).

In this example the computed stress response was sampled within one-tenth of an element length of the plate edge. The resulting ripple effect is much more pronounced than it was in the square plate results. The indirect method exhibits a ripple effect of about ±1.0%. This is overshadowed by the strong divergence from the solution exhibited by the direct method algorithm. As with the first example the direct method is more accurate excluding the edge region. However, a high boundary/domain ratio problem (in two and three dimensions) requires an algorithm that is more accurate in the edge region, since this region comprises a greater portion of the domain. The indirect method continues to exhibit a loss and then recovery in accuracy near the boundary except in the corner region.

The corner region is the area of strongest divergence for both methods. Since the term "constant element" alludes to the artificial boundary loads, the BEM will not give an exact solution for a uniform stress field (unlike the finite element method with a constant strain element, for example). In the first two problems considered the prescribed boundary conditions were constant (hydrostatic tension). The indirect BEM solves the problem in the infinite domain, and the gradient of the artificial boundary stresses is noted to be high near the corners. It is believed that the artificial boundary stresses tended to increase near the corners to maintain the square shape of the plate.

A concentration of elements near the corners yields a more accurate solution by better modeling of the artificial tractions. However, for a model with a fixed number of elements, overconcentration near the corners increases the ripple effect near the large elements. One method for refining a boundary mesh might be based on maintaining a constant value for the total artificial traction per element. Bode's study showed that beveling the corners was not effective with constant elements.

In Bode's study, 96 well-placed elements gave average values almost as good as 288 equally spaced elements. The ripple effect, however, was more prominent. Central processing unit (CPU) time is the major factor in the number of elements. The following table shows the relationship between the number of elements and CPUs in Bode's study. The third problem in our study, a stress concentration problem, did not show cost as strongly dependent on the number of elements (see Figure 19).

Number of Elements	Normalized CPUs
96	1.0
192	3.5
288	8.65

Forty-eight elements of unit length were used in both the square and rectangular plate problems. At the response lines half an element length away from the edge, the results from the square plate problem were more accurate. At the center of the plate, stress errors of 2% to 3% were observed for the square plate and 5% to 7% were observed for the rectangular plate. Thus, the proximity of edges has an adverse effect on stress response accuracy.

The higher inaccuracies in the rectangular plate problem are probably due to each response point being relatively closer to more edges and a greater number of elements. In this example the BEM slightly decreased in accuracy as the narrowness of the domain increased. For a given accuracy the difference in analysis cost for the two plate problems would not vary much with the BEM. However, the cost would be expected to vary if the finite element method (FEM) were to be used.

The third problem studied was a 1-inch-thick square plate with a circular cut out that involves high stress gradients or stress concentrations. Uniform compression stress of 1 psi was applied to two opposite edges. Various numbers of elements were used to model the external and internal boundaries, but in all cases the elements for a given boundary were of uniform size. The various models are indicated in Figure 19. A vertical and a horizontal line, of 37 response points each, extended from the hole to the outer boundary. Figure 20 shows the stress response for the different models. The theoretical solution shown assumes the elastic plate to be of infinite extent.

Since the response lines did not closely parallel an edge, no ripple effect was observed. As with the other two problems the direct algorithm diverges strongly at about half an element length from the boundary. Because of the strong divergence, the direct algorithm was only used for comparison with the coarse model, model 1. For the indirect algorithm the responses are relatively accurate, diverging slightly in the edge regions. The models with the finer element subdivision on a given boundary tend to give the most accurate results near that boundary. The level of subdivision on distant boundaries has only a secondary effect. Figure 20b illustrates the variation of σ_x along the vertical response line. For this response the localized effect of the boundaries is evident. Near the circular edge where the stress gradient is highest, models 3 and 4 show the best results and contain approximately 4% error. Models 1 and 2 have an error of between 5% and 6%. The number of elements used on the square has only a secondary effect.

Models 2 and 4 show the best results along the square edge and contain approximately 8% error. Models 1 and 3 have an error of approximately 13%. Again, the number of elements used to model the distant boundary, in this case the circular edge, has only a secondary effect.

Figure 20c illustrates the variation of σ_y along the vertical response line. In theory σ_y should be zero at both boundaries. As mentioned earlier the theoretical curve corresponds to an infinite plate, and this is the main source of discrepancy. For models 1 and 3 (both having a 28-element square boundary) a slight decrease in accuracy with a recovery near the square boundary is exhibited again.

Figure 20d illustrates the variation of σ_x along the horizontal response line. In theory this stress should be 0 at the circle and -1 psi (the applied traction) at the outer boundary. The theoretical solution is approaching -1 psi slower than the BEM solution.

Figure 20e illustrates the variation of σ_y along the horizontal response line. The accuracy trends previously mentioned are slightly deviated from near the circle because in this example model 2 gives slightly better results than model 3 while having fewer elements on the circular boundary. The maximum error for σ_y on the circular boundary was 20% for this response line. But it must be noted that σ_y is of lesser importance in the stress concentration example.

In general, the constant-element BEM exhibited a good ability to model stress gradients. Unlike the uniform stress field problems, the stress gradient problem showed the BEM to overestimate stress values (i.e., to be conservative). The boundary effects are observed to be local and relative to the element size. Distant boundaries are seen to have only a secondary effect.

SUMMARY AND CONCLUSIONS

The indirect boundary element method has been investigated for application to one- and two-dimensional elastostatics problems in structural analysis. The theoretical basis of the method has been described by beginning with the simple problem of a beam resting on an elastic foundation and then extending the theory to a more useful class of problems pertaining to the plane stress and plane strain analysis in elastostatics.

The numerical implementation of the theoretical formulation was illustrated with the development of computer programs for both small and large computers. Stress analysis capability using these computer programs was assessed by comparison of results to theoretical solutions and to results from another computer program that is based on the alternative direct boundary element formulation.

Three basic concepts constitute the theoretical formulation of the indirect boundary element method: Green's functions, superposition, and the Kirchhoff uniqueness theorem.

Green's functions are akin to the more familiar influence functions, and they give the stress and displacement response at an arbitrary field point due to a unit force at an arbitrary source point in the domain. These functions are classical results from the theory of elasticity, and those which were used herein apply to infinite domains only. They suggest the reformulation of actual problems in the finite domain in terms of problems in the infinite domain. This gives rise to the auxiliary problem, which is solved in place of the actual problem, and thus the origin of the word "indirect" in the indirect boundary element method.

Through superposition of the Green's functions, a set of boundary integral equations is constructed whose solution imposes the prescribed boundary conditions on a contour in the infinite domain that is identical to the boundary contour of the actual finite problem.

Kirchhoff's uniqueness theorem then requires that the solution of the auxiliary problem be identical to the solution of the actual problem. The solution of the auxiliary problem is accomplished numerically. The boundary is discretized into N straight line segments over which unknown artificial normal and tangential stress tractions are interpolated. In this study a constant value for the unknowns was assumed over each segment. The integration of the boundary integrals over a segment was carried out analytically, and the results are the coefficients of the 2N unknown tractions. A linear system of 2N algebraic equations in the 2N unknowns occurs and is solved by Gaussian elimination. The coefficient matrix has little exploitable structure for the homogeneous examples considered. It is both fully populated and nonsymmetric. Once the artificial tractions along the boundary are known, the stress and displacement response may be determined by superposition.

The results from the rather simple two-dimensional numerical experiments carried out in this study suggest that the boundary element methods are susceptible to accuracy deterioration within one element length of the boundary or edge of the domain. However, within the domain the accuracy is very satisfactory, usually always within 5% of exact solutions.

Very near the edges, however, a ripple effect occurs in the computed stress values. The ripple effect is an oscillation in the stress magnitude about the exact value. The oscillation along a line paralleling an edge increases as a corner of the domain is approached. That is, as another edge is approached, the ripple effect becomes worse.

Since the critical region of the ripple effect is within one element length of an edge, element size is a factor in achieving accuracy. Local element size gradation can be employed to achieve improved accuracy near an edge, while distant element size has little influence on the accuracy in the vicinity of this edge.

Inasmuch as significant stresses often occur on boundaries, fine subdivisions of those boundaries may be necessary to achieve desired accuracy.

With regard to edge effects, the indirect boundary element method faired better than the particular direct boundary element implementation that was available for comparison.

The results pertaining to the accuracy of the indirect boundary element method to correctly capture stress gradients were very encouraging, even for the constant stress elements employed. The results suggest that the method may be a very economical analysis tool for determining stress concentration factors in elastostatics.

Compared to finite element methods, the necessary input data requirements are less, and smaller matrices result. The boundary element methods are therefore more suited to small computers. This would then allow the personal interactive advantages of microcomputers to further enhance the boundary element methods as effective stress analysis tools.

RECOMMENDATION

From the results of this study, it is believed that a combined finite element and boundary element computer program may prove successful in contributing to the reduction in the high costs now associated with nonlinear finite element programs. A two-dimensional program could be developed and used to evaluate potential advantages in the solution of nonlinear problems, particularly soil-structure interaction problems where, for example, a buried structure along with some surrounding soil may be behaving nonlinearly and the remaining half-space soil is behaving linearly. However, the proper theoretical and numerical treatment of the interface equations must be thoroughly evaluated and then implemented. This step is a necessary prerequisite to an effective implementation. The recommendation is to undertake the theoretical and numerical treatment of coupling the finite element and indirect boundary element methods.

ACKNOWLEDGMENT

The authors would like to thank Mr. Robert N. Murtha of the Structures Division at NCEL for authoring the one-dimensional FORTRAN program developed in this study. Further, the authors would like to thank Mr. Nathan Shoemaker, Mathematician at the Computer Center at NCEL, for developing several integration subroutines for two-dimensional boundary integrals.

REFERENCES

1. O. C. Zinekiewicz. The finite element method, Chapter 23. London, England, McGraw-Hill Book Company, Ltd., 1977.

2. P. K. Banerjee and R. Butterfield. Boundary element methods in engineering science. London, England, McGraw-Hill Book Company, 1981.

3. C. E. Massonnet. "Numerical use of integral procedures," Chapter 10 in Stress Analysis, O. C. Zienkiewicz and G. S. Hollister, Editors. London, England, John Wiley and Sons Ltd., 1965.

4. K. Bathe and E. L. Wilson. Numerical methods in finite element analysis. Englewood Cliffs, N.J., Prentice-Hall, Inc., 1976.

5. P. K. Banerjee and R. Butterfield, Editors. Development in boundary element methods - 1. London, England, Applied Sciences Publishers Ltd., 1979.

6. G. Arfken. Mathematical methods for physicists, Chapter 16. New York, N.Y., Academic Press, 1968.

7. P. M. Morse and H. Feshbach. Methods of theoretical physics, part I, chapter 7. New York, N.Y., McGraw-Hill Book Company, Inc., 1953.

8. C. A. Brebbia, Editor. "Boundary element methods," in Proceedings of the Third International Seminar, Irvine, Calif. Berlin, Springer-Verlag, 1981.

9. C. Wang. Applied elasticity. New York, N.Y., McGraw-Hill Book Company, 1953.

10. R. Butterfield. "New concepts illustrated by old problems," Chapter 1 in Developments in Boundary Element Methods - 1, P. K. Banerjee and R. Butterfield, Editors. London, England, Applied Sciences Publishers Ltd., 1979, pp 1-20.

11. M. Hentenyi. Beams on elastic foundation. University of Michigan Press, Ann Arbor, Michigan, 1946.

12. J. H. Bode. The solution of mixed boundary value problems in the theory of elasticity by a boundary integral equation technique, Ph. D. thesis, Michigan Technological University. Houghton, Michigan, 1976.

13. S. Timoshenko and J. N. Goodier. Theory of elasticity. New York, N.Y., McGraw-Hill Book Company, Inc., 1951.

14. S. Walker. "Approximate fundamental solutions and alternative formulations," in Boundary Element Methods, C. A. Brebbia, Editor, in Proceedings of the Third International Seminar, Irvine, Calif. Berlin, Springer-Verlag, 1981.

15. J. J. Waters and J. M. Nelson. "A program for solving potential problems on a desktop computer," Advanced Engineering Software, vol 2, no. 2, 1980.

16. C. A. Brebbia. The boundary element method for engineers. London, England, Pentech Press, 1978.

BIBLIOGRAPHY - Boundary Element Method

Banaugh, R. P., and W. Goldsmith (1963). "Diffraction of steady elastic waves by surfaces of arbitrary shape," Journal Applied Mechanics, vol 30, 1963, pp 589-595.

Banerjee, P. K., and R. Butterfield, Editors (1979). Developments in boundary element methods - 1. London, England, Applied Science Publishers, Ltd, 1979, p. 292.

Brebbia, C. A. (1978). The boundary element method for engineers. London, England, Pentech Press Limited, 1978, p. 189.

Brebbia, C. A., and R. Nakaguma (1979). "Boundary elements in stress analysis," Journal of Engineering Mechanical Division, ASCE, vol. 105, no. EM1, Feb 1979, pp 65-69.

Cruse, T. A., and F. J. Rizzo (1968). "A direct formulation and numerical solution of the general transient elastodynamic problem, I," Journal Mathematical Analysis and Application. vol 22, 1968, pp 244-259.

Cruse, T. A. (1969). "Numerical solutions in three dimensional elastostatics," International Journal of Solids and Structures, vol 5, Dec 1969.

Cruse, T. A. (1974). "An improved boundary integral equation method for three-dimensional stress analysis," Computer and Structures, vol 4, 1974, pp 741-754.

Cruse, T. A., and F. J. Rizzo, Editors (1975). "Boundary integral equation method: Computational applications in applied mechanics," Proceedings, ASME Special Publication, no. 11, 1975.

Geers, T. L., W. A. Loden, and H. C. Yee (1977). "Boundary-element analysis of fluid-solid impact," T. Belytschko and T. L. Geers, editors, Computational Methods for Fluid Structure Interaction Problems, ASME, AMD-Vol 26, 1977.

Jeng, G. and A. Wexler (1977). "Isoparametric finite element variational solution of integral equations for three-dimensional fields," International Journal Numerical Methods Engineering, vol 11, 1977, pp 1455-1474.

Lachat, J. C., and J. O. Watson (1976). "Effective numerical treatment of boundary integral equations: A formulation for three-dimensional elastostatics," International Journal Numerical Methods Engineering, vol 10, 1976, pp 99-1006.

Shaw, R. P. (1968). Comments on "Numerical solution for transient scattering from a hard surface of arbitrary shape retarded potential technique," Journal of Acoustical Society of America, vol 43, 1968, pp 638-639.

Shaw, R. P. (1973). "Integral equation formulation of dynamic acoustic fluid-elastic solid interaction problems, J.A.S.A., vol 53, 1973, pp 514-520.

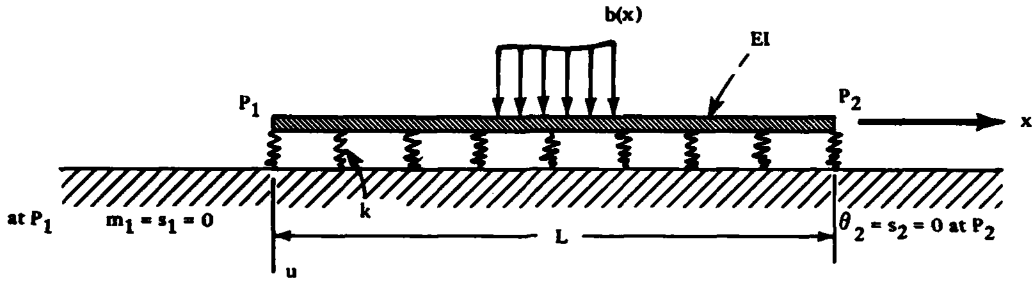


Figure 1. Beam resting on an elastic foundation — the actual problem.

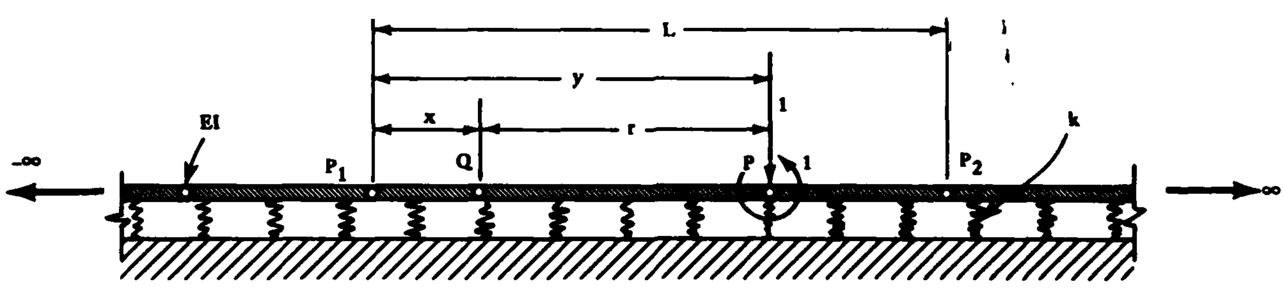


Figure 2. Infinite beam and elastic foundation.

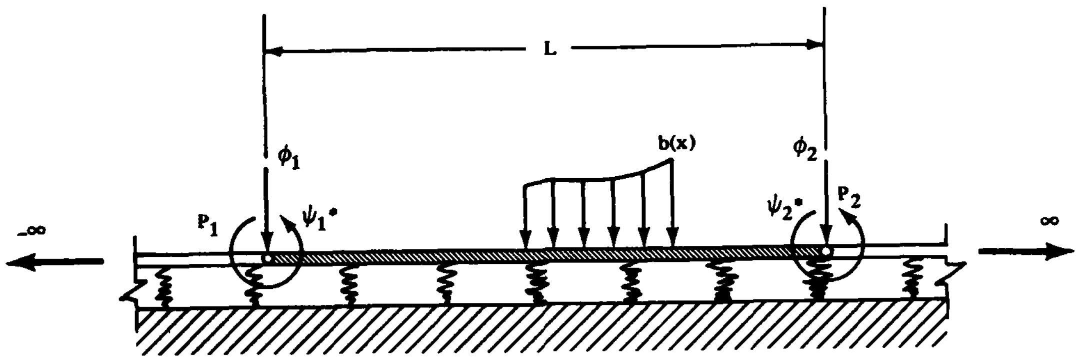


Figure 3. The auxiliary problem.

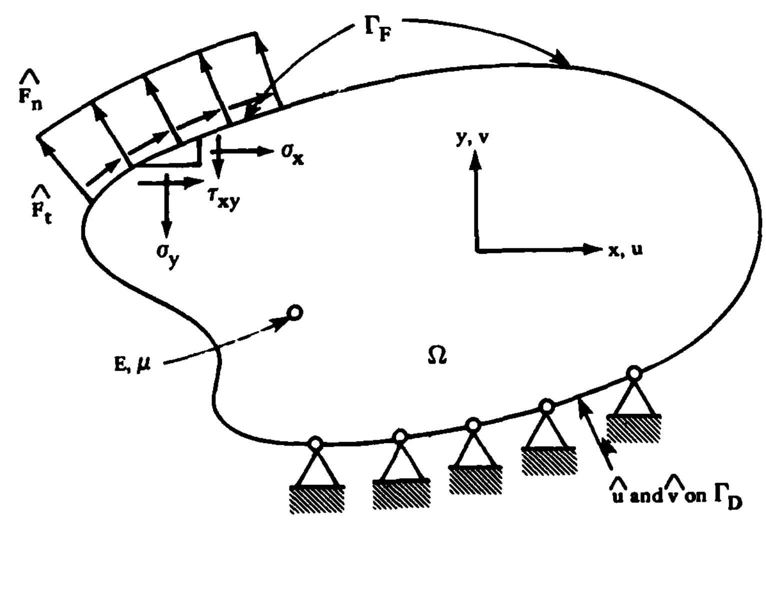


Figure 4. Two-dimensional elastostatic plate – the actual problem.

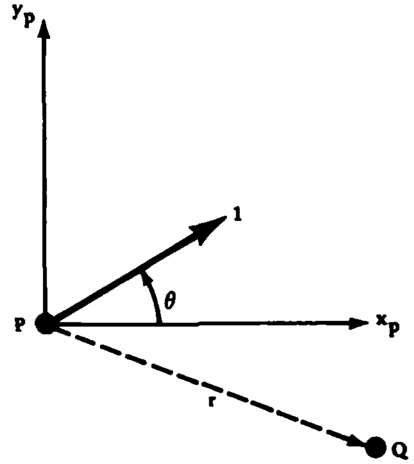


Figure 5. Source point P and field point Q in an infinite elastostatics region.

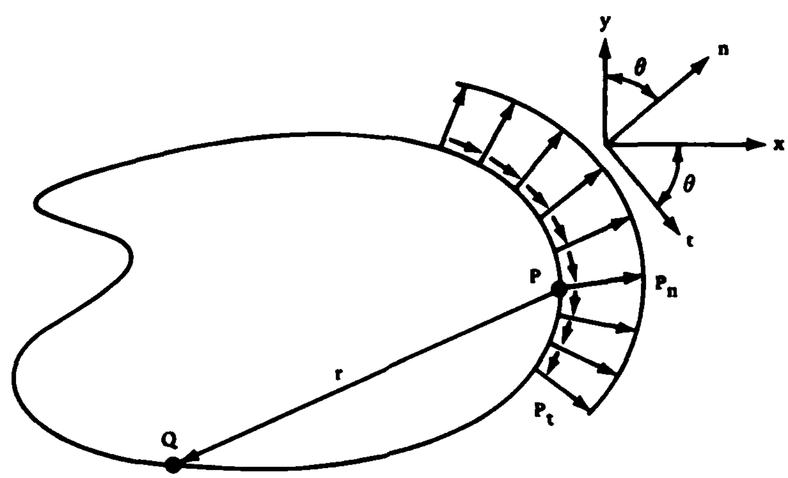


Figure 6. Actual two-dimensional region embedded in infinite two-dimensional region – the auxiliary problem.

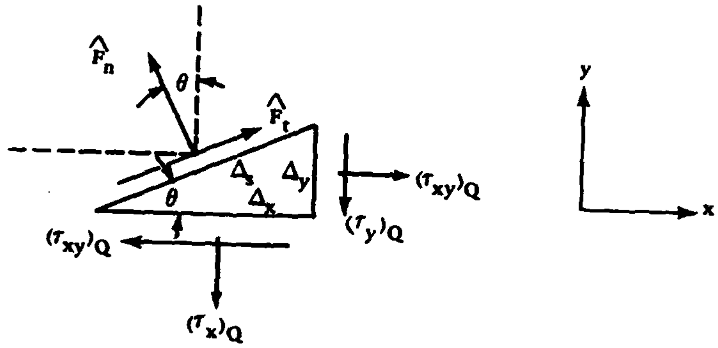


Figure 7. Equilibrium of point Q on the boundary.

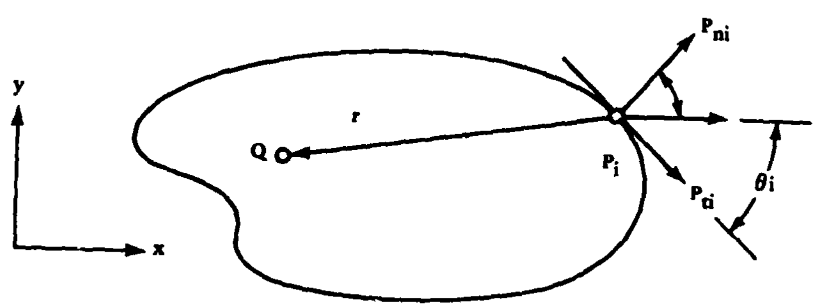
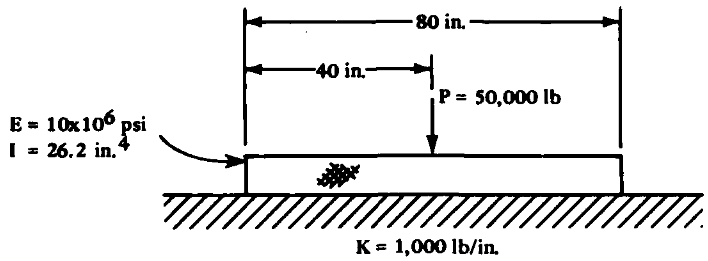


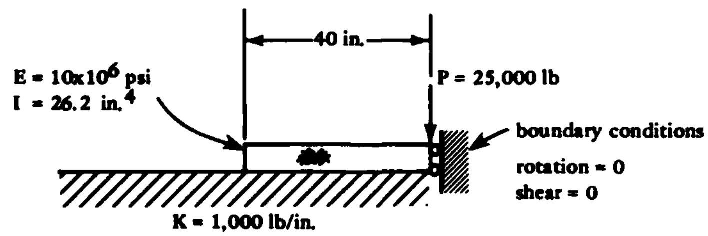
Figure 8. Response at a prescribed internal point Q.



Results at Centerline

	Theory	Boundary Element Micro-BASIC	Boundary Element Mainframe-FORTRAN
Deflection (in.)	0.861	0.861	0.861
Moment (in.-lb)	417,135	416,898	417,130
Execution time (sec)		38-interpreted 21-compiled	0.35

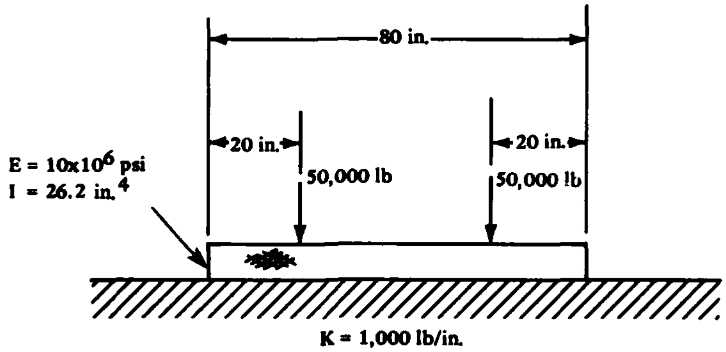
Figure 9. Full model of a concentrated load on centerline.



Results at Center Line

	Theory	Boundary Element Micro-BASIC
Deflection (in.)	0.861	0.861
Moment (in.-lb)	417,135	416,948

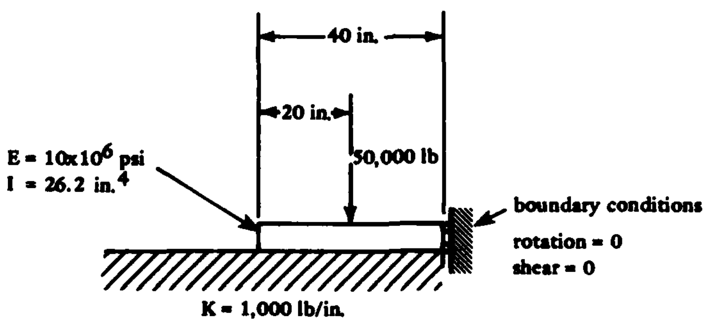
Figure 10. Symmetric model of a concentrated load on centerline.



Results at Ends of Beam

	Theory	Boundary Element Micro-BASIC	Boundary Element Mainframe-FORTRAN
Deflection (in.)	1.104	1.103	1.103
Moment (in.-lb)	0	0	0

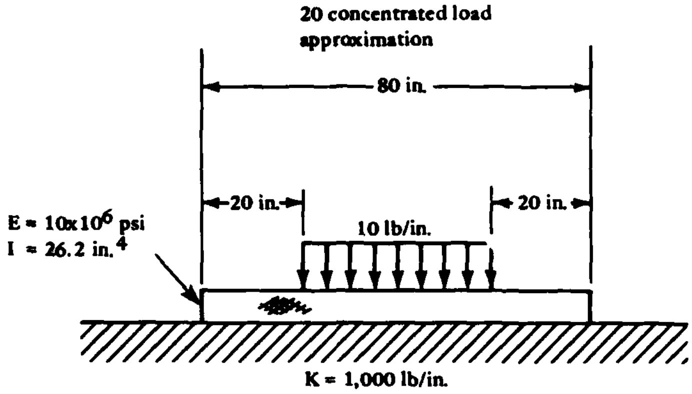
Figure 11. Full model of symmetric concentrated load.



Results at End of Beam

	Theory	Boundary Element Micro-BASIC
Deflection (in.)	1.104	1.103
Moment (in.-lb)	0	0

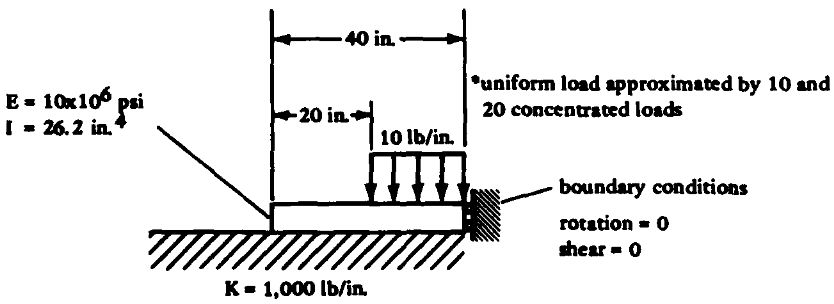
Figure 12. Symmetric model of symmetric concentrated loads.



<u>Results</u>

	Theory	Boundary Element Micro-BASIC	Boundary Element Mainframe-FORTRAN
Deflection at center (in.)	6.25×10^{-3}	6.26×10^{-3}	6.26×10^{-3}
Moment at center (in.-lb)	1535	1531	1532
Slope at ends (rad)	1.1539×10^{-4}	1.1539×10^{-4}	1.1533×10^{-4}

Figure 13. Full model of partial uniform load.



<u>Results</u>

	Theory	Boundary Element Micro-BASIC (10)*	(20)*
Deflection at center (in.)	6.25×10^{-3}	6.26×10^{-3}	6.26×10^{-3}
Moment at center (in.-lb)	1535	1532	1532
Slope at left end (rad)	1.1539×10^{-4}	1.1541×10^{-4}	1.1534×10^{-4}

Figure 14. Symmetric model of partial uniform load.

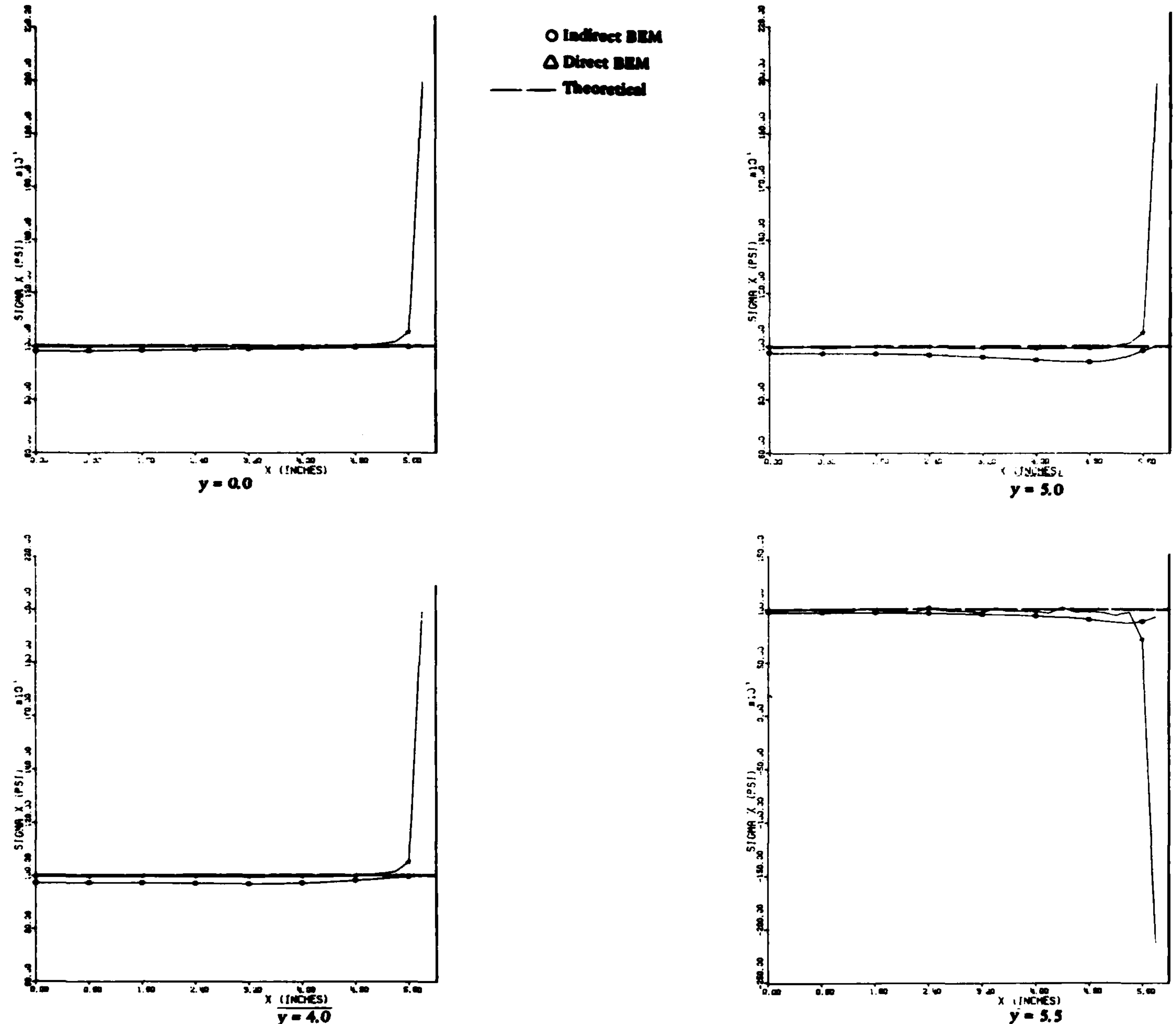


Figure 15a. Square plate model, normal stress, σ_X, at four response lines.

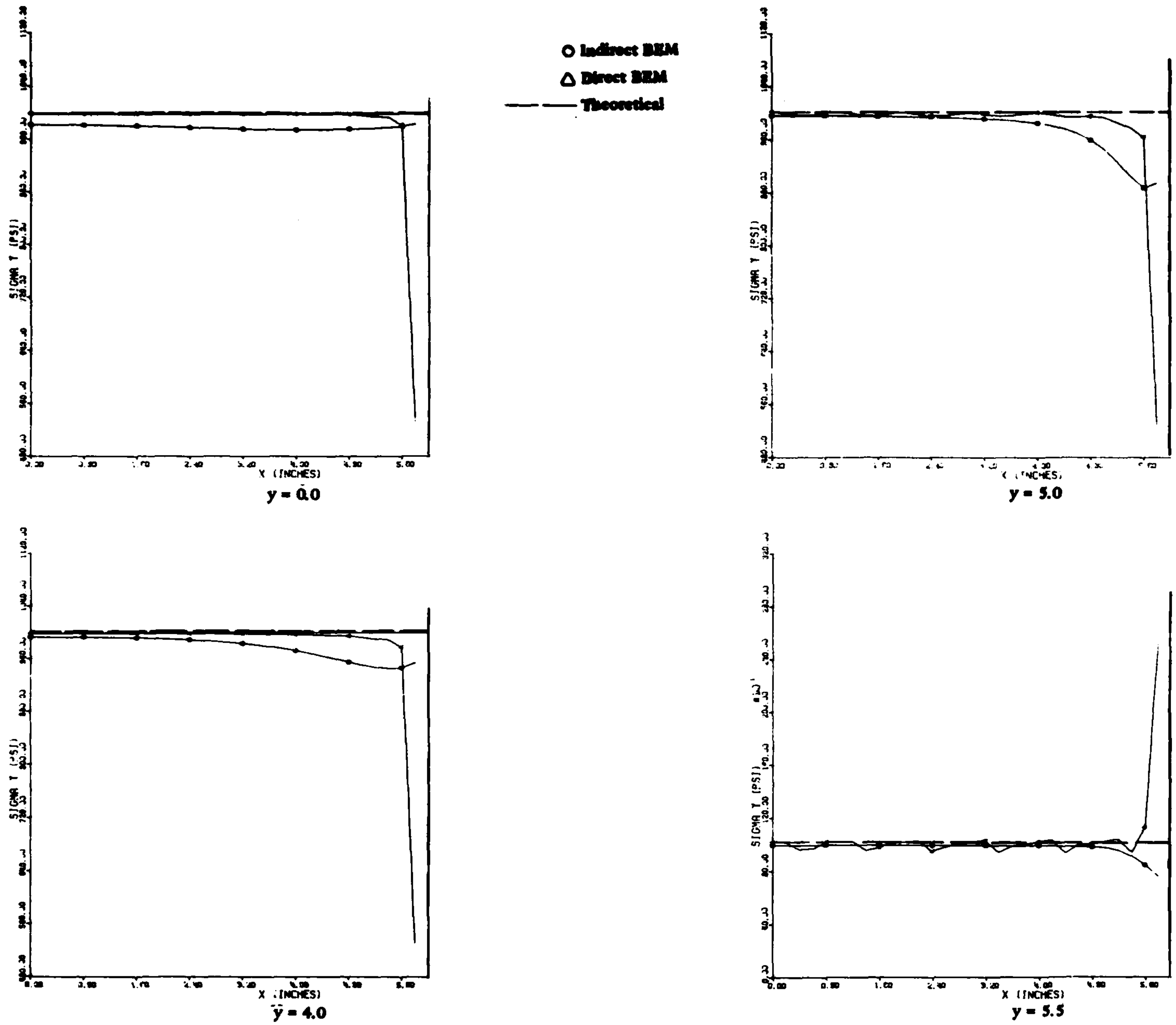


Figure 15b. Normal stress, σ_y, at four response lines.

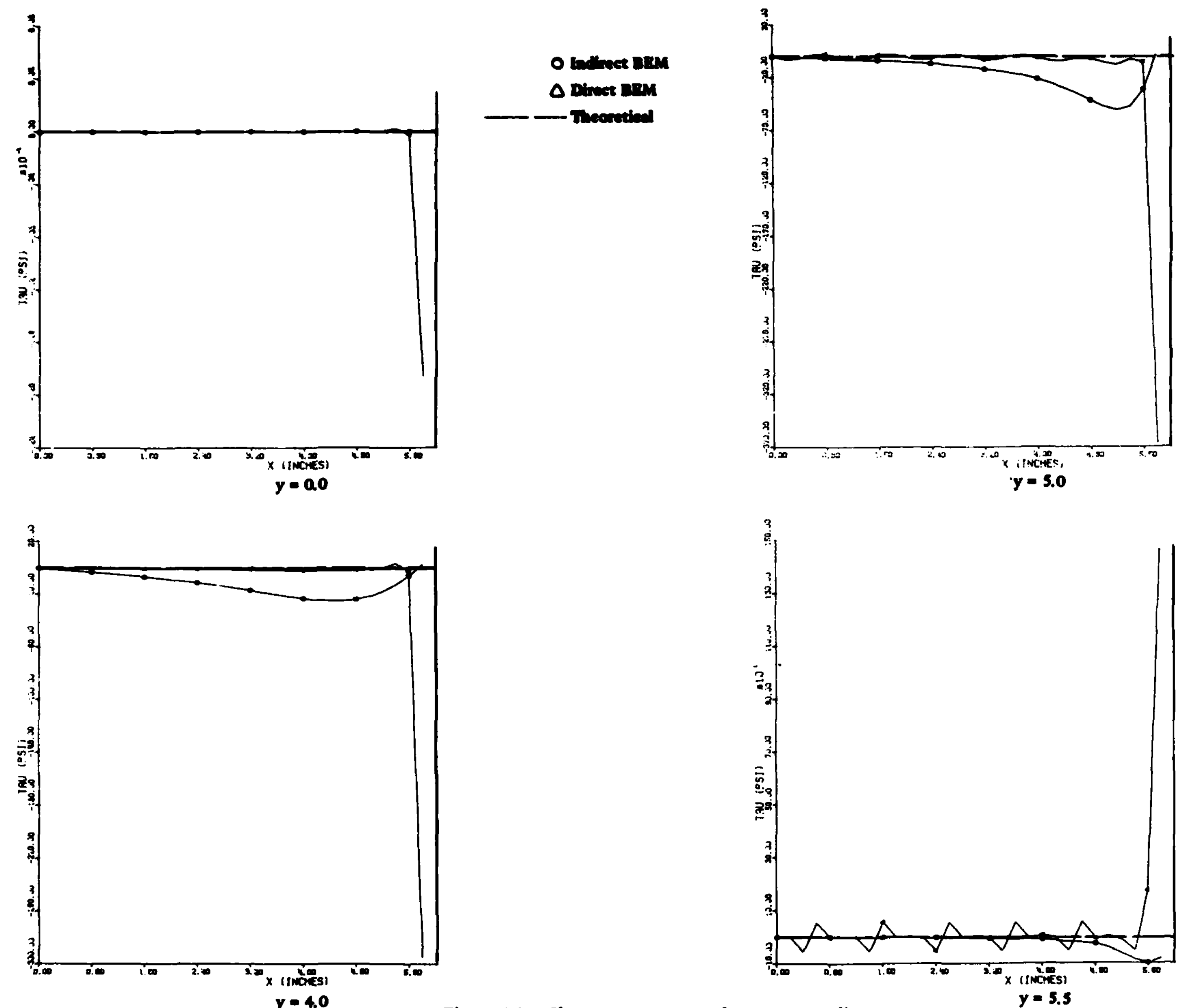


Figure 15c. Shear stress, τ_{xy}, at four response lines.

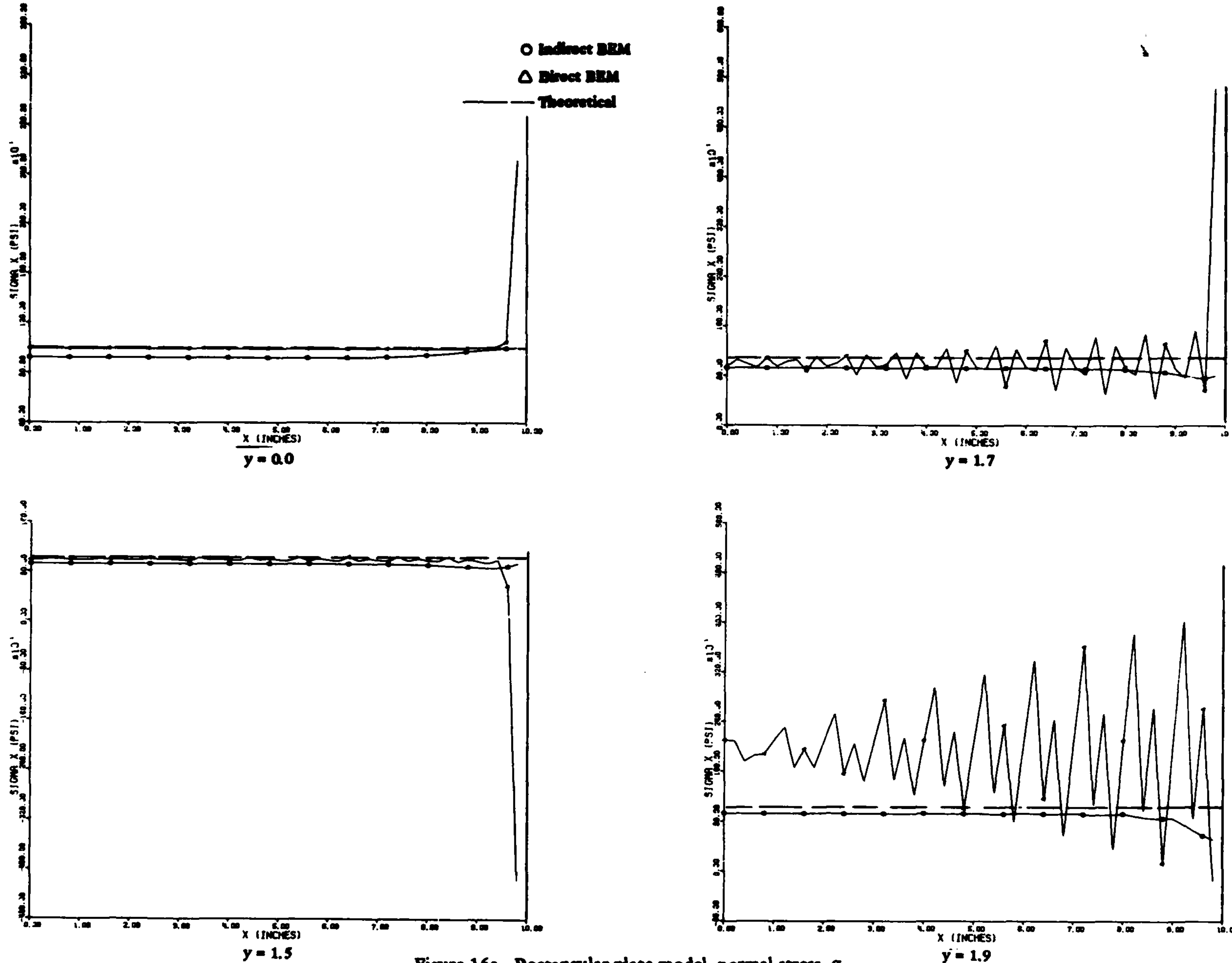


Figure 16a. Rectangular plate model, normal stress, σ_x, at four response lines.

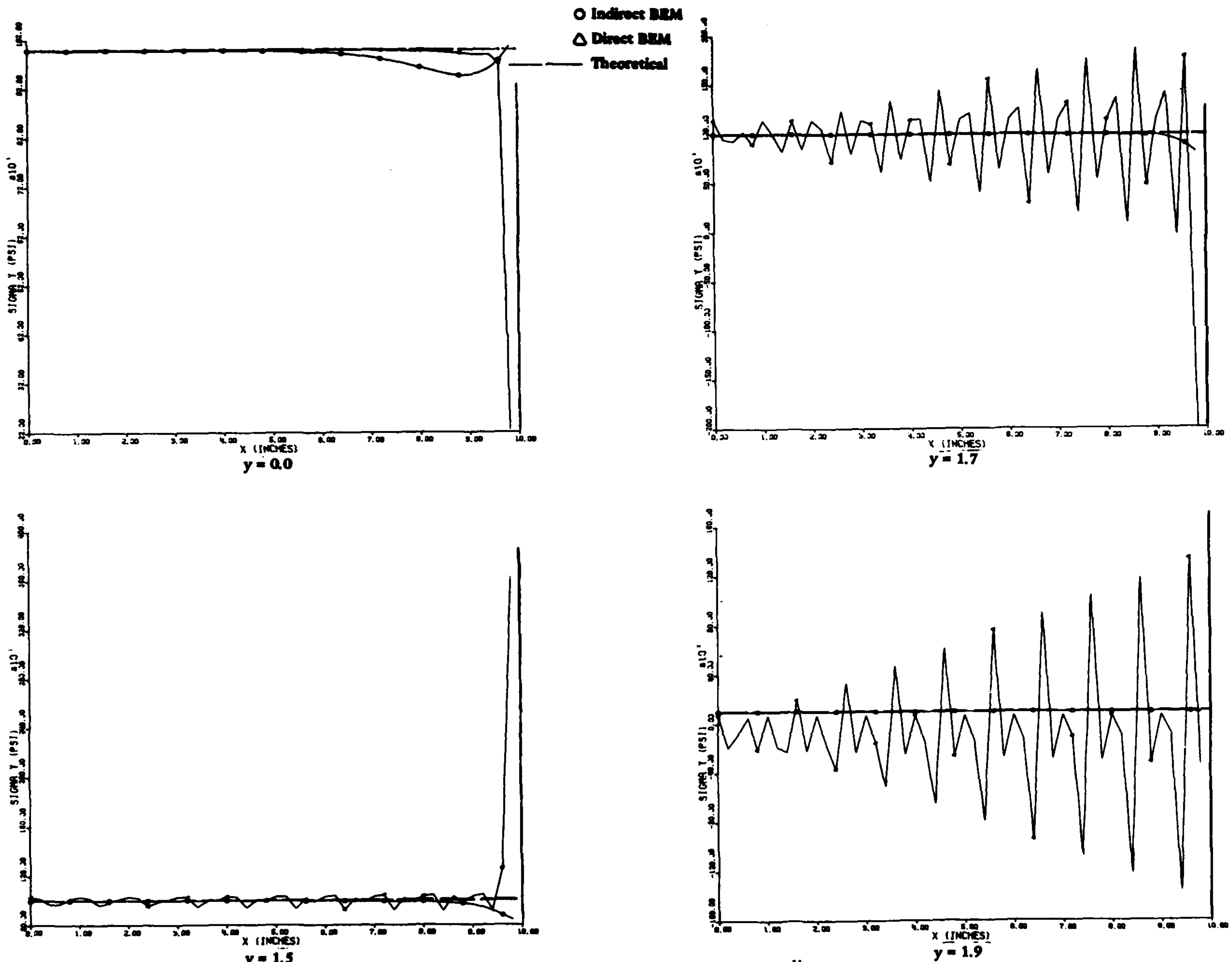


Figure 16b. Normal stress, σ_y, at four response lines.

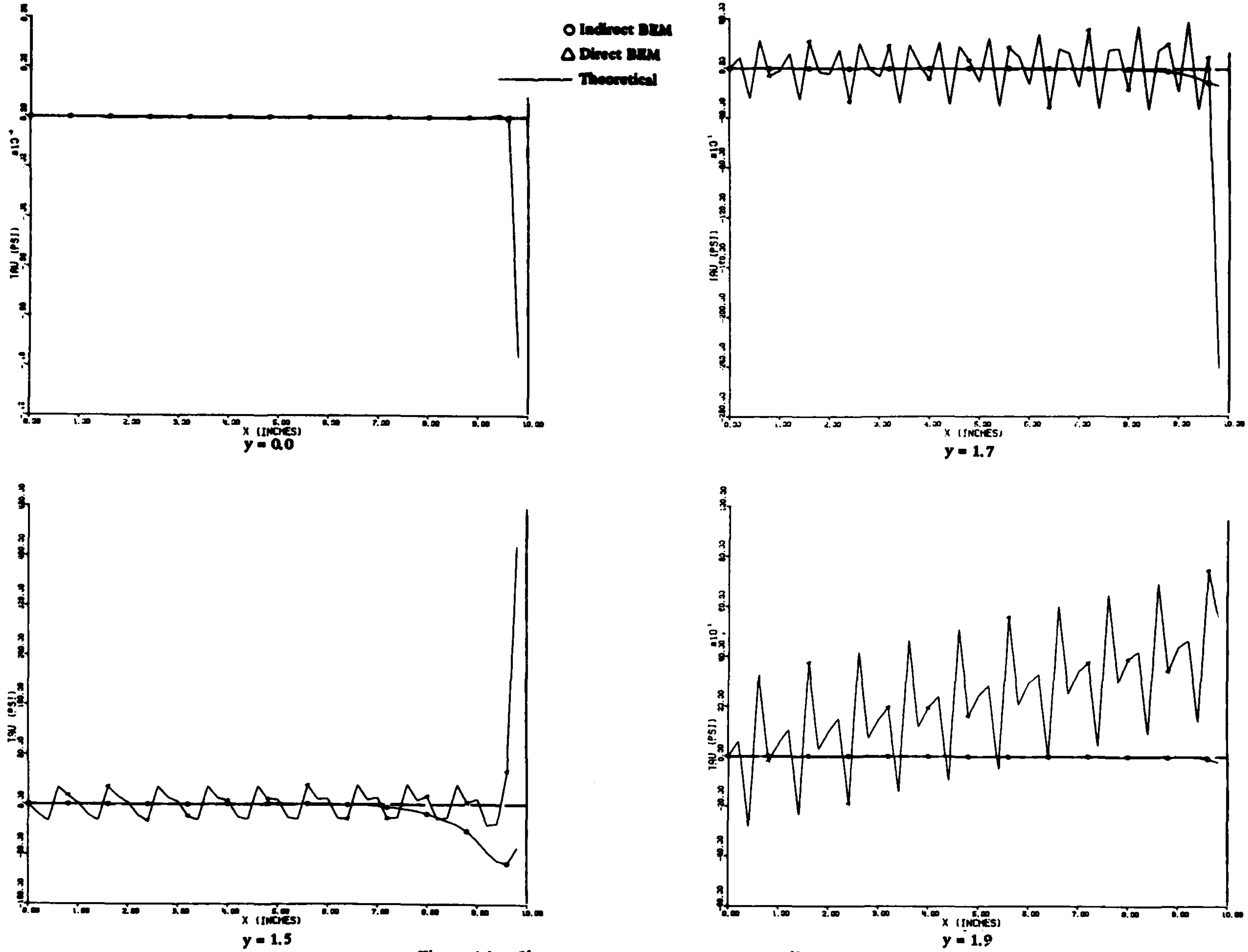


Figure 16c. Shear stress, τ_{xy}, at tour response lines.

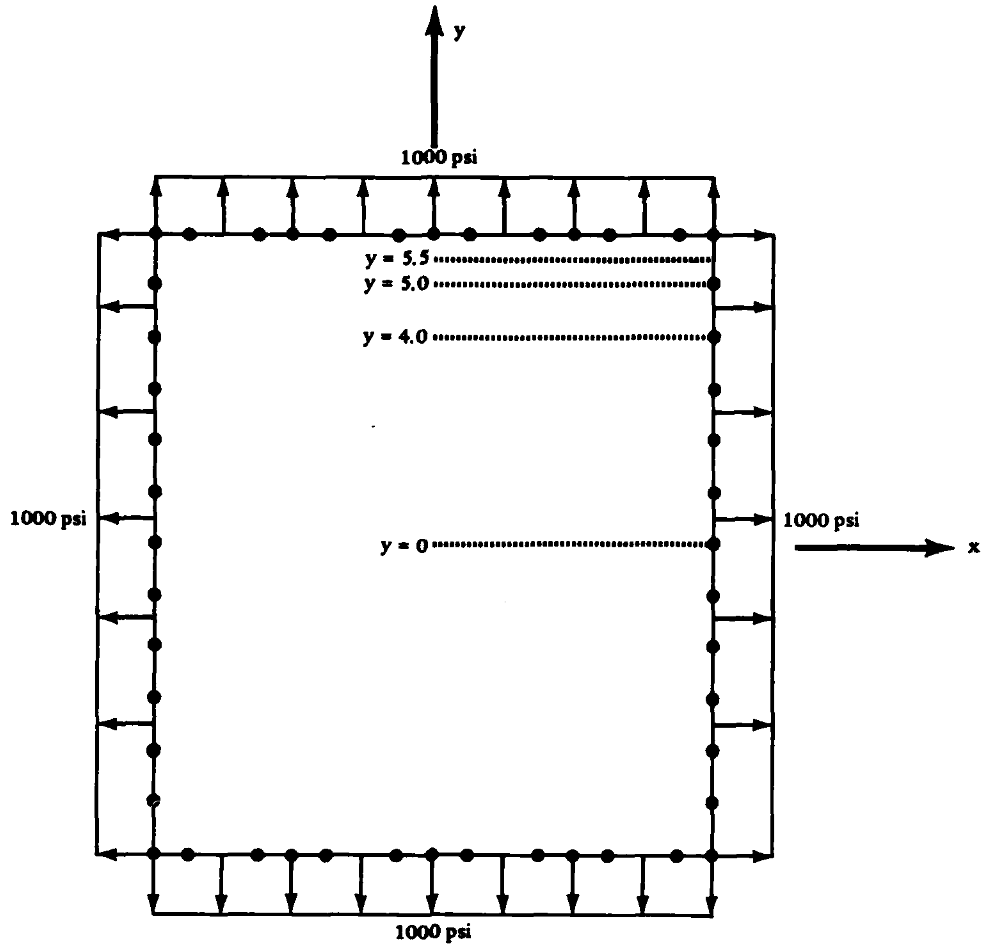


Figure 17. Square plate model.

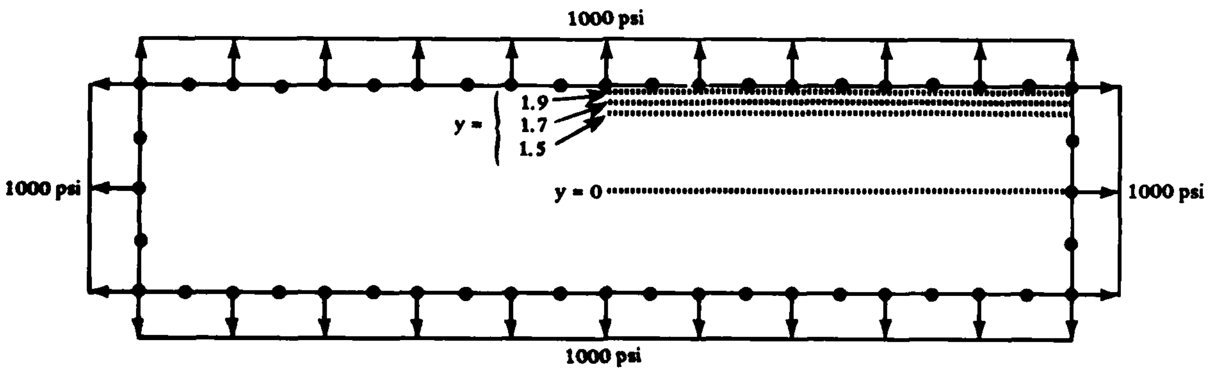
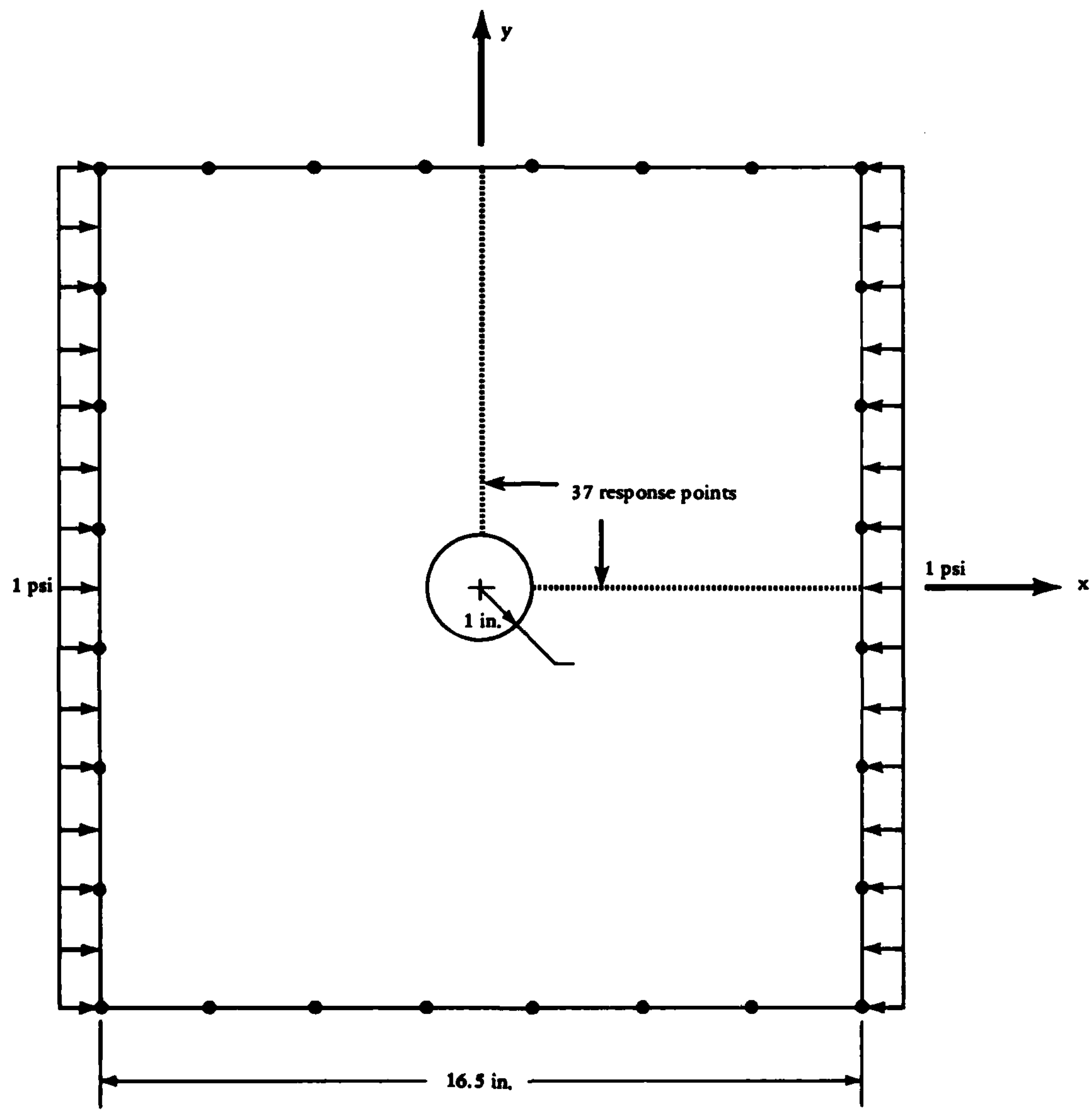


Figure 18. Rectangular plate model.



Model Number	Number of Elements			Normalized Computer Cost
	Square	Circle	Total	
1	28	24	52	1
2	56	24	80	1.6
3	28	48	76	1.4
4	56	48	104	2.4

Figure 19. Square plate stress concentration models.

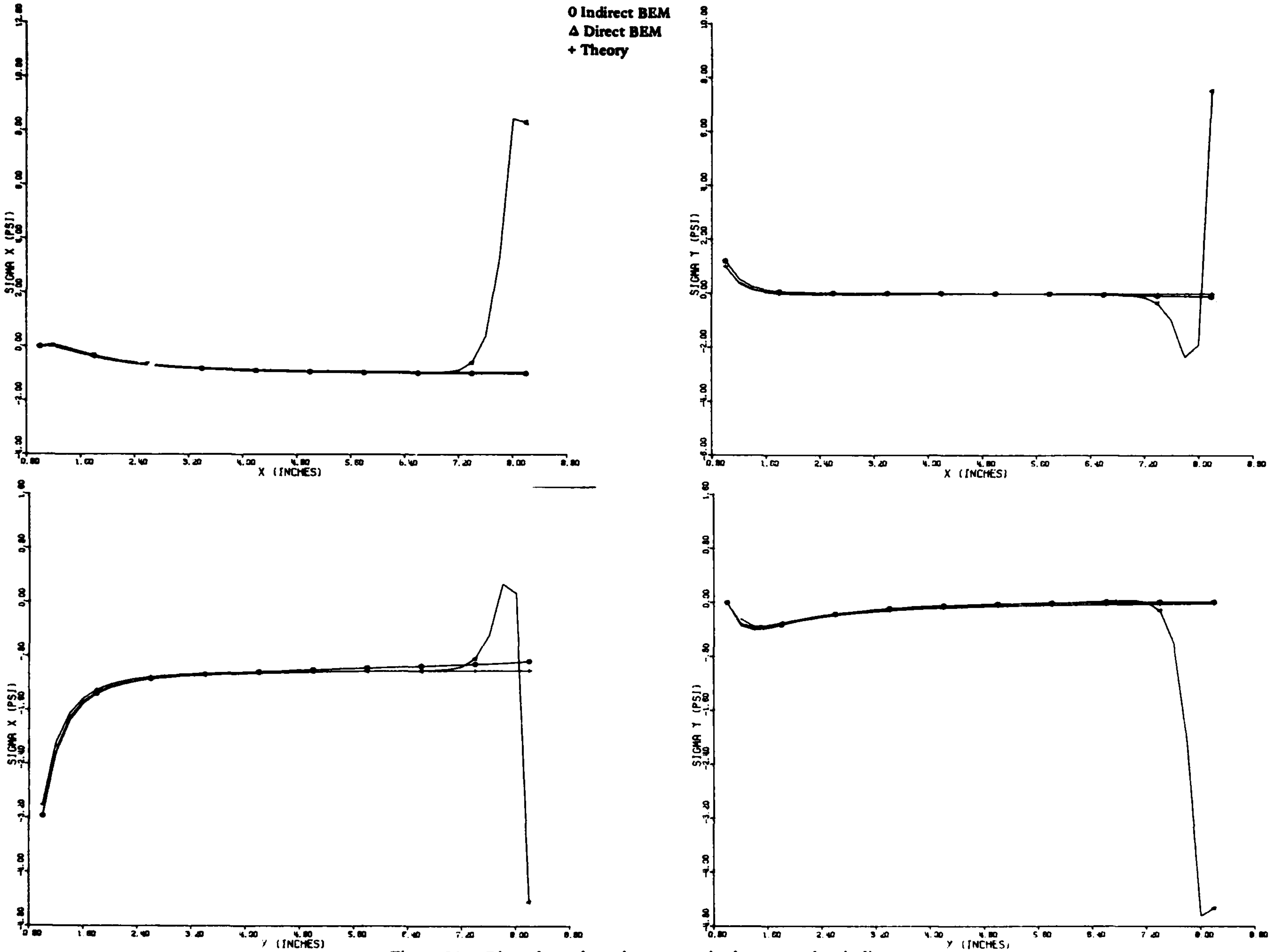


Figure 20a. Direct boundary element method compared to indirect boundary element method for model 1.

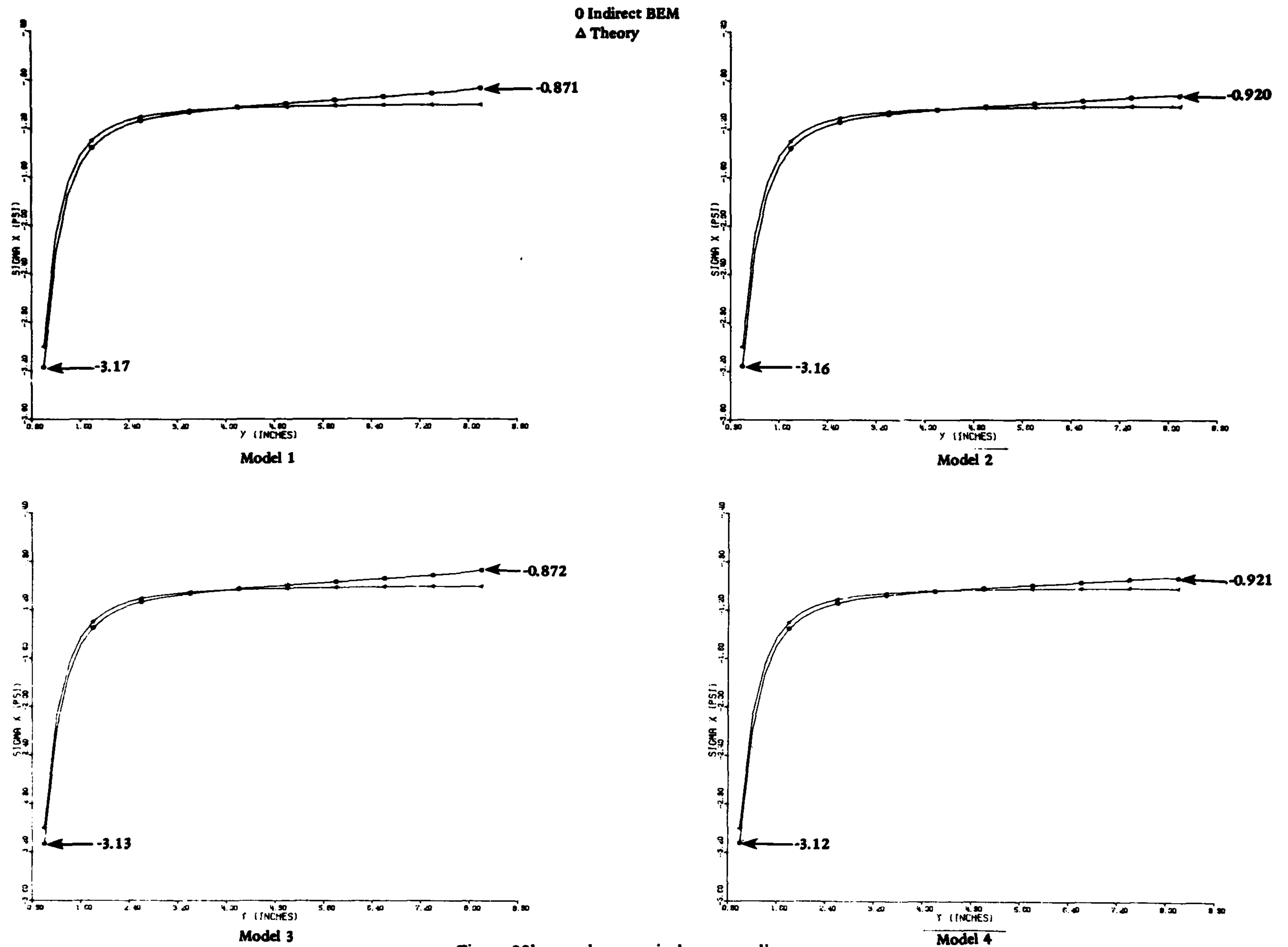


Figure 20b. σ_x along vertical response line.

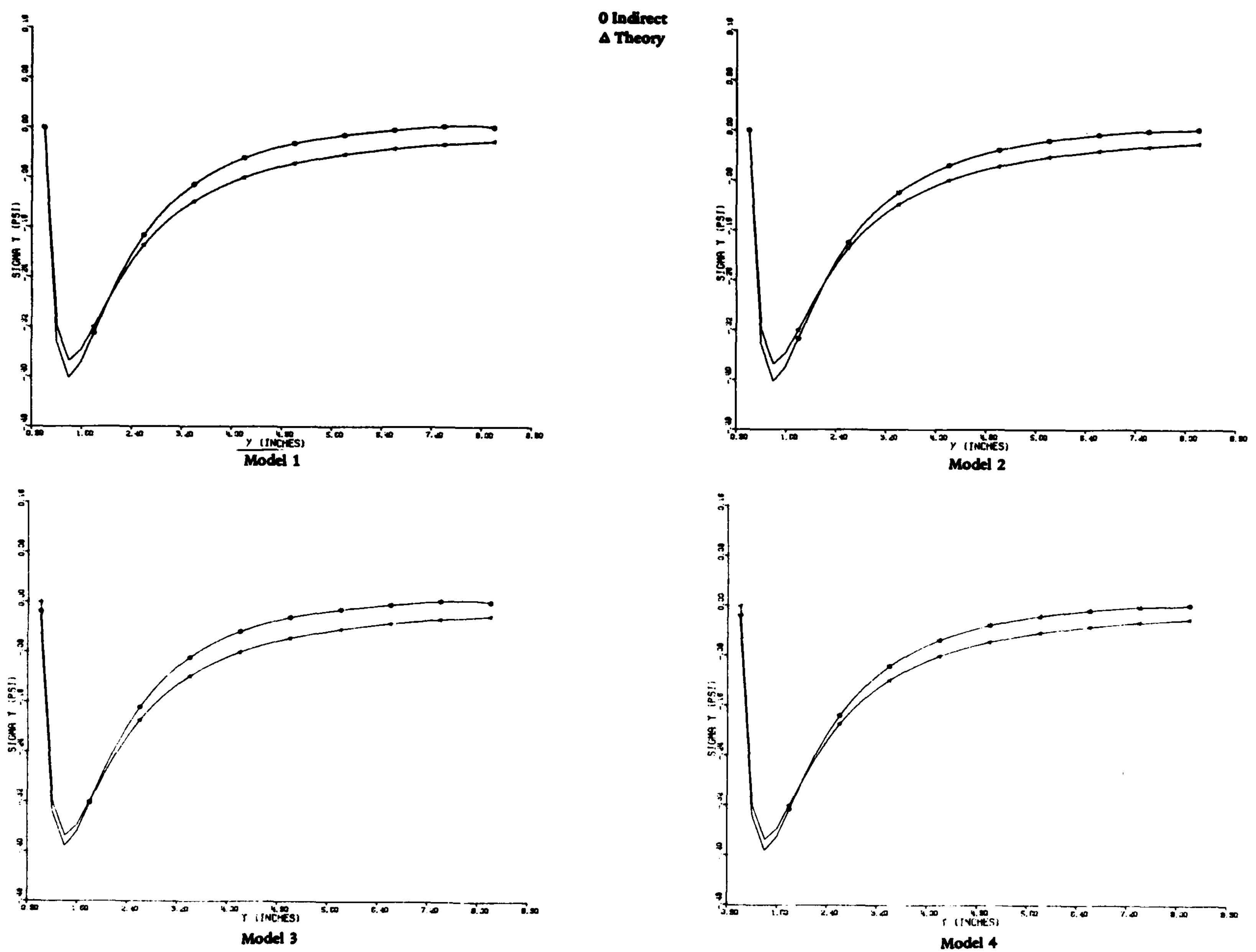


Figure 20c. σ_y along vertical response line.

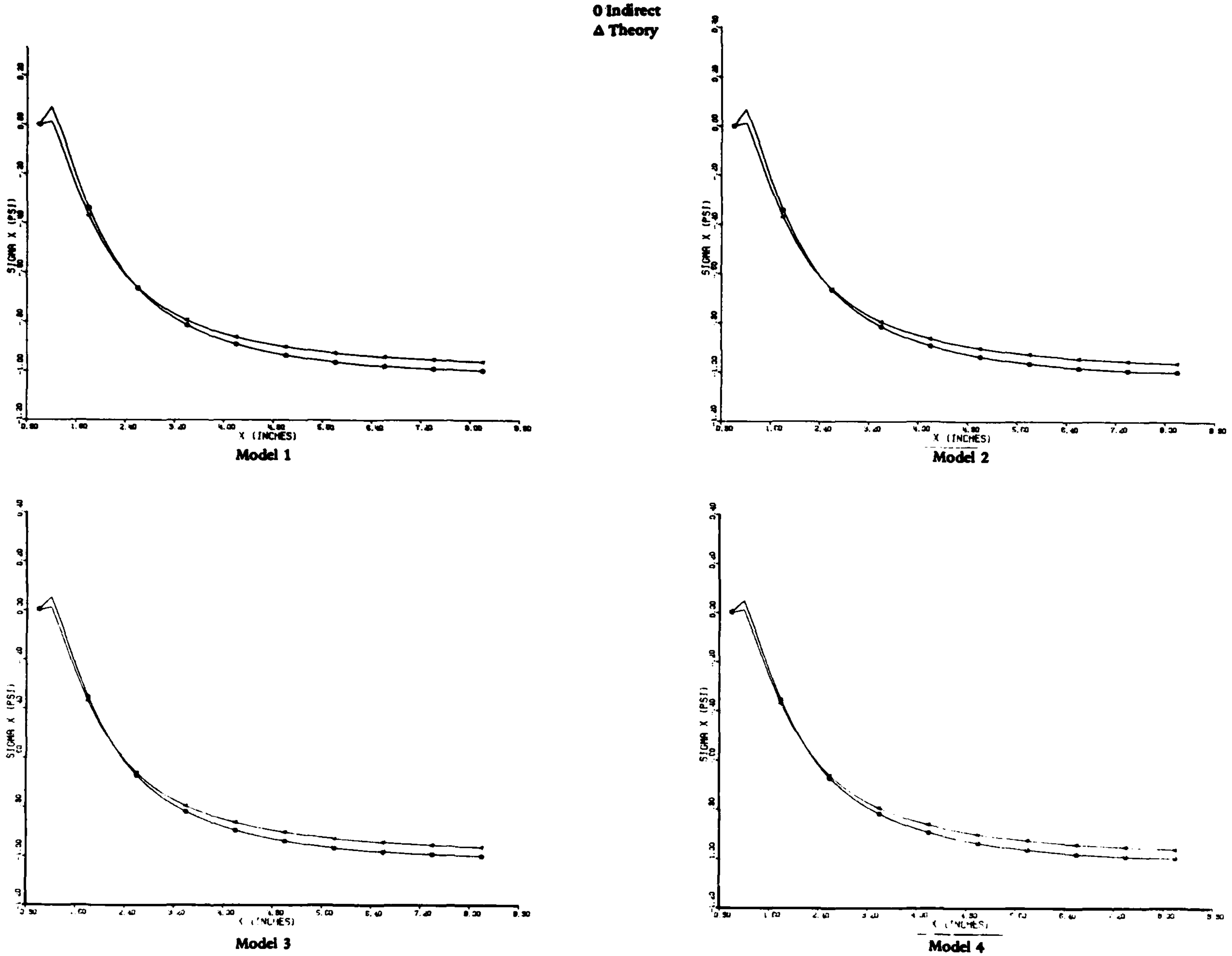


Figure 20d. σ_x along horizontal response line.

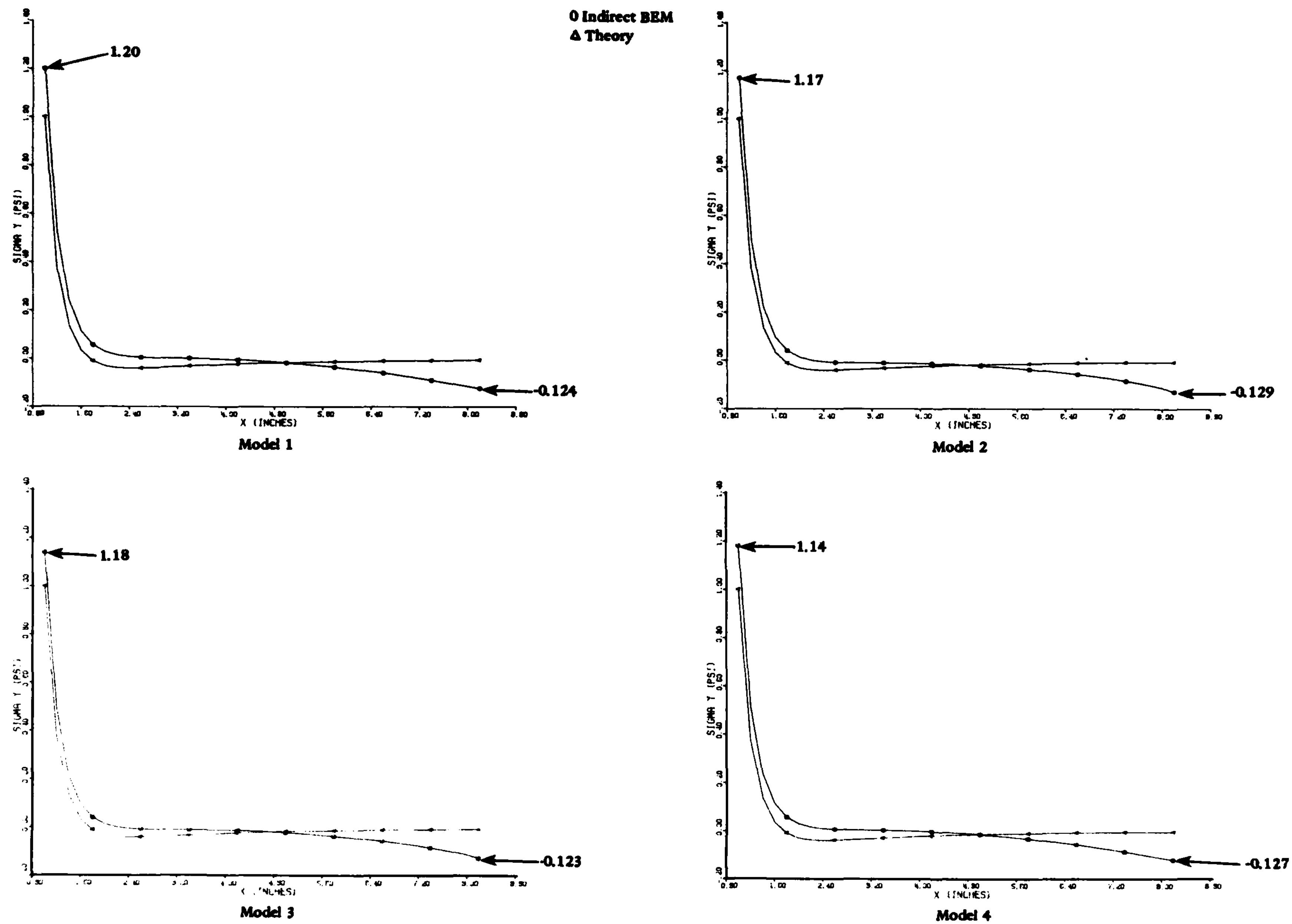


Figure 20e. σ_y along horizontal response line.

Appendix A

ASPECTS OF THE NUMERICAL IMPLEMENTATION

This appendix includes further detail on the numerical implementation of the indirect boundary element method (BEM). Each section explains one aspect that was omitted for clarity in the text of the report on two-dimensional elastostatics. Though some of the detail is unique to the BIM2D program, the overall methodology is general in nature.

SIMPLIFIED BOUNDARY INTEGRAL EQUATIONS

Equation 14a is rewritten as follows. The contribution of the singular element is written as a product of influence coefficients and discrete artificial tractions. The integration along the remainder of the boundary is written as the summation of the integrations along the nonsingular elements. In the present study, these integrations were carried out analytically, but these details are omitted and only the results are given. Using the following substitutions,

$$SXN_{Q,Q} = \frac{\lambda + 2G \sin^2 \theta}{2(\lambda + 2G)} \quad \text{(A-1a)}$$

$$SXT_{Q,Q} = - \sin \theta \cos \theta \quad \text{(A-1b)}$$

$$\gamma 1_{Q,P} = \int_P \left[- c_o \; c_1 \left(\frac{x_Q - x_P}{r^2} \right) + c_2 \; \frac{(x_Q - x_P)(y_Q - y_P)^2}{r^4} \right] d\ell \quad \text{(A-2a)}$$

$$\gamma 2_{Q,P} = \int_P \left[c_1 \frac{(y_Q - y_P)}{r^2} - c_2 \frac{(x_Q - x_P)^2 (y_Q - y_P)}{r^4} \right] d\ell \qquad \text{(A-2b)}$$

Equation 14a can be rewritten in the following form:

$$(\sigma_x)_Q = SXN_{Q,Q} P_{n_Q} + SXT_{Q,Q} P_{t_Q} + \sum_{\substack{P=1 \\ P \neq Q}}^{N} \left[(- P_{n_P} \sin\theta + P_{t_P} \cos\theta) \gamma 1_{Q,P} + (P_{n_P} \cos\theta + P_{t_P} \sin\theta) \gamma 2_{Q,P} \right] \qquad \text{(A-3)}$$

Combining terms with respect to the boundary tractions gives

$$(\sigma_x)_Q = SXN_{Q,Q} P_{n_Q} + SXT_{Q,Q} P_{t_Q} + \sum_{\substack{P=1 \\ P \neq Q}}^{N} \left[P_{n_P}(-\sin\theta\, \gamma 1_{Q,P} + \cos\theta\, \gamma 2_{Q,P} + P_{t_P}(\cos \gamma 1_{Q,P} + \sin\theta\, \gamma 2_{Q,P}) \right] \qquad \text{(A-4)}$$

The integration along the remainder of boundary can now be written in coefficient form. The nonsingular influence coefficients are given by:

$$SXN_{Q,P} = - \sin\theta\, \gamma 1_{Q,P} + \cos \gamma 2_{Q,P} \qquad \text{(A-5a)}$$

$$SXT_{Q,P} = \cos \gamma 1_{Q,P} + \sin \gamma 2_{Q,P} \qquad \text{(A-5b)}$$

Thus, Equation A-4 is further simplified to:

$$(\sigma_x)_Q = \sum_{P=1}^{N} P_{n_P} SXN_{Q,P} + \sum_{P=1}^{N} P_{t_P} SXT_{Q,P} \qquad \text{(A-6a)}$$

where the singular coefficients, subscripted Q,Q, are given by Equations A-1, and the nonsingular coefficients, subscripted Q,P, are given by Equations A-5.

In a similar manner $(\sigma_y)_Q$, $(\tau_{xy})_Q$, $(u)_Q$, and $(v)_Q$ can be rewritten as:

$$(\sigma_y)_Q = \sum_{P=1}^{N} P_{n_P}\ SYN_{Q,P} + \sum_{P=1}^{N} P_{t_P}\ SYT_{Q,P} \qquad \text{(A-6b)}$$

$$(\tau_{xy})_Q = \sum_{P=1}^{N} P_{n_P}\ TN + \sum_{P=1}^{N} P_{t_P}\ TT \qquad \text{(A-6c)}$$

$$(u)_Q = \sum_{P=1}^{N} P_{n_P}\ UN + \sum_{P=1}^{N} P_{t_P}\ UT \qquad \text{(A-6d)}$$

$$(v)_Q = \sum_{P=1}^{N} P_{n_P}\ VN + \sum_{P=1}^{N} P_{t_P}\ VT \qquad \text{(A-6e)}$$

The coefficients $SYN_{Q,P}$ and $SYT_{Q,P}$ are defined in a manner similar to that shown in Equations A-1 and A-5 using Equations 14b; the coefficients TN and TT use Equation 14, the coefficients UN and UT use Equation 15a, and the coefficients VN and VT use Equation 15b.

PRESCRIBED BOUNDARY TRACTIONS IN TERMS OF THE ARTIFICIAL BOUNDARY TRACTIONS

Equation 16 gives the prescribed boundary tractions in terms of the unknown stresses $(\sigma_x)_Q$, $(\sigma_y)_Q$, and $(\tau_{xy})_Q$. Writing the unknown stresses in terms of the artificial boundary tractions, using Equations A-6a, b, and c, gives:

$$\sum_{P=1}^{N}\Big\{P_{n_P}[\sin^2\theta(SXN_{Q,P}) + \cos^2\theta(SYN_{Q,P}) - 2\sin\theta\cos\theta\,(TN_{Q,P})]$$
$$+ P_{t_P}[\sin^2\theta(SXT_{Q,P}) + \cos^2\theta(SYT_{Q,P}) - 2\sin\theta\cos\theta\,(TN_{Q,P})]\Big\} = \hat{F}_{N_Q} \qquad \text{(A-7a)}$$

$$\sum_{P=1}^{N}\Big\{P_{n_P}[-\cos\theta\sin\theta\,(SYN_{Q,P}) + \cos\theta\sin\theta\,(SYN_{Q,P}) + (\cos^2\theta - \sin^2\theta)(TN_{Q,P})]$$
$$+ P_{t_P}[-\cos\theta\sin\theta\,(SXT_{Q,P}) + \cos\theta\sin\theta\,(SYT_{Q,P}) + (\cos^2\theta - \sin^2\theta)(TT_{Q,P})]\Big\} = \hat{F}_{t_Q} \qquad \text{(A-7b)}$$

The quantities within the square brackets are the coefficients of the matrix $\underset{\sim}{K}$. Thus, Equations A-7 can be rewritten as:

$$\sum_{P=1}^{N} \left\{ P_{n_P} K_{ij} + P_{t_P} K_{i,j+1} \right\} = \hat{F}_{n_Q} \tag{A-8a}$$

$$\sum_{P=1}^{N} \left\{ P_{n_P} K_{i+1,j} + P_{t_P} K_{i+1,j+1} \right\} = \hat{F}_{t_Q} \tag{A-8b}$$

where $i = 2xQ - 1$
$j = 2xP - 1$

In this form it is apparent that the influence of the artificial boundary tractions at each element P with the prescribed boundary tractions at element Q results in a block of four terms (a 2x2 submatrix) in the coefficient matrix.

PRESCRIBED BOUNDARY DISPLACEMENTS IN TERMS OF THE ARTIFICIAL BOUNDARY TRACTIONS

Equation 17 gives the prescribed boundary displacements $(\hat{u}, \hat{v})$ in terms of the unknown boundary displacements (u_Q, v_Q). Writing these displacements in terms of the artificial boundary tractions, using Equations A-6d and e, gives:

$$\sum_{P=1}^{N} \left\{ P_{n_P} [UN_{Q,P}] + P_{t_P} [UT_{Q,P}] \right\} = \hat{u}_Q \tag{A-9a}$$

$$\sum_{P=1}^{N} \left\{ P_{n_P} [VN_{Q,P}] + P_{t_P} [VT_{Q,P}] \right\} = \hat{v}_Q \tag{A-9b}$$

The quantities within the square brackets are the coefficients of the matrix $\underset{\sim}{K}$. Thus, Equations A-9 can be rewritten in the same form as Equations A-8:

$$\sum_{P=1}^{N} \left\{ P_{n_P} K_{i,j} + P_{t_P} K_{i,j+1} \right\} = \hat{u}_Q \qquad \text{(A-10a)}$$

$$\sum_{P=1}^{N} \left\{ P_{n_P} K_{i+1,j} + P_{t_P} K_{i+1,j+1} \right\} = \hat{v}_Q \qquad \text{(A-10b)}$$

where $i = 2xQ - 1$
$j = 2xP - 1$

Again, the interaction of an ordered pair of elements results in a block of four terms in the coefficient matrix.

SYSTEM OF EQUATIONS

Equations A-8 and Equations A-10 each yield a set of two equations in 2N unknowns. Either set of equations can be written in matrix notation as:

$$\begin{bmatrix} K_{i,1} & K_{i,2} & K_{i,3} & K_{i,4} & \cdots & K_{i,2N-1} & K_{i,2N} \\ K_{i+1,1} & K_{i+1,2} & K_{i+1,3} & K_{i+1,4} & \cdots & K_{i+1,2N-1} & K_{i+1,2N} \end{bmatrix} \begin{Bmatrix} P_{n_1} \\ P_{t_1} \\ P_{n_2} \\ P_{t_2} \\ \vdots \\ P_{n_N} \\ P_{t_N} \end{Bmatrix} = \begin{Bmatrix} B_i \\ B_{i+1} \end{Bmatrix} \qquad \text{.... (A-11)}$$

where $i = 2xQ - 1$

B_i = prescribed x displacement or normal traction at element Q

B_{i+1} = prescribed y displacement or tangential traction at element Q

For each straight-line element around the model boundary, either the boundary displacements or tractions are prescribed. Thus, each element produces two equations. The total system of equations (Equation 18) is formed by combining N pairs of equations, such as Equations A-10. Thus, 2N unknown artificial boundary tractions are expressed in terms of 2N prescribed boundary tractions or displacements.

Appendix B

BIM1D DOCUMENTATION AND LISTING, BASIC VERSION

PURPOSE

This appendix illustrates how the nature of the boundary integral method lends itself to the solution of structural analysis problems on a microcomputer.

SCOPE

The BIM1D program solves the problem of a beam on an elastic foundation. The loading is limited to concentrated loads; however, it is set up to generate loads to approximate linear varying continuous loads. The boundary condition at each end of the finite beam can be specified by entering the value of two of the following quantities:

- shear
- moment
- displacement
- rotation

This allows modeling half of the beam for symmetrical and asymmetrical loading cases (see Appendix C).

METHODOLOGY

The theory for the program is explained in detail in the body of the report. The following discussion briefly explains the major subroutines.

<u>Input Routine</u> (Beginning Statement Number 1050)

The input routine is interactive only to the extent that it prompts the user for input. It does not check the range of the user's input (with the exception of allowing a maximum of 20 uses of defined concentrated loads) and does not allow editing of input.

<u>Generator Routine</u> (Beginning Statement Number 920)

The generator routine generates concentrated loads to approximate the continuous loads. These loads are added to the concentrated load arrays (both positions and values).

Matrices Development Routine (Beginning Statement Number 600)

The matrices development routine initially adjusts the position of all concentrated loads such that no load is within a given distance, epsilon, from either boundary. The routine then develops both the [G] and [Hp] matrices.

Matrix Inversion Routine (Beginning Statement Number 750)

The matrix inversion routine inverts the 4x4 [G] matrix and performs an accuracy test. In the accuracy test the flexibility matrix [G] is multiplied by its inverse to give the identity matrix. The sum of all terms in the identity matrix is printed on the screen and should approach 4 (the order of the matrix). If [G] is a singular matrix the program ends.

Solve Routine (Beginning Statement Number 730)

The solve routine solves for the boundary forces and moments such that when they are applied to the infinite beam, the points within the boundaries respond as if they were on the finite beam.

Response Input Routine (Beginning Statement Number 1800)

The response input routine, as the initial input routine, is interactive only to the extent of prompting the user for input. It allows a maximum of 50 response points and does not allow a response point outside of the span.

Response Routine (Beginning Statement Number 250)

The response routine initially adds the boundary moments to the applied moment array (it is set up in this manner so that applied moments are a simpler addition); similarly, the boundary forces are added to the applied force array. Each response point is then compared to each concentrated load position. If they coincide the response point is moved to the right by the amount epsilon (response points at L are moved to the left). With singularities avoided, the responses can be calculated. For each response point, [kB] and [kM] are developed then multiplied by (B) and (M), respectively. This routine prints on the screen the response point that is being calculated.

Ouput Routine (Beginning Statement Number 3000)

The output routine formats and prints the calculated results and user input. A "DEBUG PRINTOUT" is optional as explained in the User Instructions. Very large numbers can cause an illegal quantity error.

USER INSTRUCTIONS

I. PROGRAM INPUT

A. Model Input.

1. Enter the foundation spring constant.
2. Enter the modulus of elasticity of the beam.
3. Enter the moment of inertia of the beam.
4. Enter the span of the beam.
5. Enter the number of continuous loads.

<u>For Each Continuous Load</u>

5a. Enter the position and value of the left end of the continuous load in the format position, value.
5b. Repeat Step 6 for the right end of the continuous load.
5c. Enter the number of concentrated loads to approximate the continuous load.
6. Enter the number of concentrated loads.

<u>For Each Concentrated Load</u>

Enter the position and value of the load in the format position, value.

<u>For Each Boundary Condition</u>

7. Enter the numerical code of the known boundary value.
8. Enter the value of the known boundary condition.

Following the problem input, the program prints on the screen the title of the major subroutines it enters. Within the matrix inversion subroutine an accuracy test is performed. (For an explanation of the individual subroutines see the METHODOLOGY section of this appendix.)

I. PROGRAM INPUT (Continued)

B. Response Input

1. Enter the number of response points where shear, moment, displacement, and rotation are to be calculated.

<u>For Each Response Point</u>

2. Enter the position of the response point

The response subroutine prints the number of the response point where the solution is being calculated. Following the last response point the printing of the output begins.

II. PROGRAM OUTPUT

The user input and response output are always printed. The user has the option to obtain a "DEBUG PRINTOUT." The "DEBUG PRINTOUT" consists of a list of the concentrated loads that approximate the continuous loads and the boundary loads (boundary moments are not included). The flexibility matrix and its inverse are also printed. Appendix C contains a few printouts. The last problem includes an annotated copy of the user interaction.

Appendix C

BIM1D EXAMPLE PROBLEMS, BASIC VERSION

```
10  CLEAR : GOTO 1540: REM   START OF PGM
15  REM
20  REM

50  REM   SUBROUTINE TO CALCULATE APPROPRIATE GREEN'S FUNCTION
60  REM   ENTER WITH RR,RC,BETA,X,Y, AND SYN
70  REM   EXIT WITH INPUT AND DM
80  REM
90 BR = BETA *  ABS (Y - X)
100  ON RC GOTO 120,150,120,150
110  REM
120  REM   GREEN'S FUNCTIONS FOR A UNIT FORCE
130  ON RR GOTO 170,180,190,200
140  REM
150  REM   GREEN'S FUNCTIONS FOR A UNIT MOMENT
160  ON RR GOTO 210,220,230,240
170 DM = BETA / 2 / SK *  EXP ( - BR) * ( COS (BR) +  SIN (BR)): RETURN
180 DM = BETA ↑ 2 / SK *  EXP ( - BR) *  SIN (BR) * SYN: RETURN
190 DM =  EXP ( - BR) / 4 / BETA * ( COS (BR) -  SIN (BR)): RETURN
200 DM =  EXP ( - BR) / 2 *  COS (BR) * SYN: RETURN
210 DM = BETA ↑ 2 / SK *  EXP ( - BR) *  SIN (BR) * SYN: RETURN
220 DM =  - BETA ↑ 3 / SK *  EXP ( - BR) * ( COS (BR) -  SIN (BR)): RETURN
230 DM =  EXP ( - BR) / 2 *  COS (BR) * SYN: RETURN
240 DM = BETA *  EXP ( - BR) / 2 * ( COS (BR) +  SIN (BR)): RETURN

245  REM
250  REM   BEAM RESPONSE SUBROUTINE
260  REM
270  PRINT "RESPONSE"
280  REM   PUT BOUNDARY MOMENTS INTO APPLIED MOMENT ARRAY
290 XM(0) = 0:M(0) = PHI(1):XM(1) = L:M(1) = PHI(3)
295  REM
300  REM   APPEND BOUNDARY FORCES INTO APPLIED FORCE ARRAY
310 XB(NT + 1) = 0:B(NT + 1) = PHI(0):XB(NT + 2) = L:B(NT + 2) = PHI(2)
311  REM
312  REM    COMPARE TO ALL POINT LOADS
```

```
314  FOR I = 0 TO NK - 1
316  FOR J = 1 TO NT + 2
318  IF XA(I) < > XB(J) THEN  GOTO 322
320 XA(I) = XA(I) +  SGN (L - EP - XA(I)) * EPSIL: GOTO 318: REM    INC AND CHK AGAIN
322  NEXT J,I
324  REM
330  REM   (R)= (KB)X(B) + (KM)X(M)
340  FOR K = 0 TO NK - 1: REM   EACH RESPONSE POINT
345 X = XA(K):RC = 1: PRINT "RESPONSE PT "K + 1
347  REM
350  REM   DEVELOP THE KB MATRIX FOR POINT K
360  FOR J = 0 TO NT + 1: REM   FOR EACH COLUMN OF KB
370 Y = XB(J + 1):SYN =  SGN (Y - X)
380  FOR I = 0 TO 3: REM    FOR EACH ROW OF KB
390 RR = I + 1: GOSUB 50:KB(I,J) = DM: NEXT I,J
395  REM
400  REM   DEVELOP THE KM MATRIX FOR POINT K
410 RC = 2
420  FOR J = 0 TO 1: REM   FOR EACH COLUMN OF KM
430 Y = XM(J):SYN =  SGN (Y - X)
440  FOR I = 0 TO 3
450 RR = I + 1: GOSUB 50:KM(I,J) = DM: NEXT I,J
460  REM   OBTAIN R MATRIX
470  FOR I = 0 TO 3;R(I,K) = 0
480  FOR J = 0 TO NT + 1:R(I,K) = R(I,K) + B(J + 1) * KB(I,J): NEXT J
490  FOR J = 0 TO 1:R(I,K) = R(I,K) + M(J) * KM(I,J): NEXT J,I,K
500  RETURN


599  REM
600  REM   MATRIX DEVELOPER SUBROUTINE
610  REM   INITIAL VERSION W/O HM AND BM MATRICES
615  REM
620  PRINT "MATRIX DEVELOPMENT"
623  FOR I = 1 TO NT
626  IF XB(I) > EP GOTO 633
630 XB(I) = XB(I) + EP: GOTO 626
633  IF XB(I) < (L - EP) GOTO 637
636 XB(I) = XB(I) - EP: GOTO 633
```

```
637  NEXT : PRINT "G MATRIX"
638  REM
640  REM   COLUMN AT A TIME
650  FOR J = 0 TO 3:RC = J + 1:Y = (J > = 2) * L:SYN =  SGN (J - 1.9)
660  FOR I = 0 TO 3:RR = BC(I):X = EP + (L - 2 * EP) * (I > = 2): GOSUB 50:G(I,J) = DM: NEXT I,J
670  REM
680  PRINT "HP MATRIX"
690  REM   ROW AT A TIME
700 RC = 1
710  FOR I = 0 TO 3:SYN =  SGN (1.9 - I):RR = BC(I):X = EP + (L - 2 * EP) * (I > = 2)
720  FOR J = 0 TO NT - 1:Y = XB(J + 1): GOSUB 50:HP(I,J) = DM: NEXT J,I
725  RETURN


728  REM
730  REM   SOLVE FOR ACTUAL BEAM UNKNOWNS
733  REM   SUBROUTINE
735  REM   PHI=GI X (F - HP X B)
737  REM
739  PRINT "SOLVE FOR ACTUAL BEAM UNKNOWNS"
740  REM
741  REM   USE (A) TO STORE (F - HP X B)
742  FOR I = 0 TO 3:A(I,0) = F(I)
743  FOR J = 0 TO NT - 1:A(I,0) = A(I,0) - HP(I,J) * B(J + 1): NEXT J,I
744  REM   CALCULATE PHI MATRIX
745  FOR I = 0 TO 3:PHI(I) = 0
746  FOR J = 0 TO 3:PHI(I) = PHI(I) + GI(I,J) * A(J,0): NEXT J,I
747  RETURN


750  PRINT "MATRIX INVERSION"
760  FOR I = 0 TO 3
765  FOR J = 0 TO 3:A(I,J) = G(I,J):GI(I,J) = 0: NEXT J
770 GI(I,I) = 1: NEXT I
780  FOR J = 0 TO 3
790  FOR I = J TO 3
```

```
800  IF A(I,J) < > 0 THEN 830
810  NEXT I
820  PRINT "SINGULAR MATRIX": END
830  FOR K = 0 TO 3:Y = A(J,K):A(J,K) = A(I,K):A(I,K) = Y:Y = GI(J,K):GI(J,K) = GI(I,K):GI(I,K) = Y: NEXT K
840 X = 1 / A(J,J)
850  FOR K = 0 TO 3:A(J,K) = X * A(J,K):GI(J,K) = X * GI(J,K): NEXT K
860  FOR N = 0 TO 3: IF N = J THEN 890
870 X =  - A(N,J)
880  FOR K = 0 TO 3:A(N,K) = A(N,K) + X * A(J,K):GI(N,K) = GI(N,K) + X * GI(J,K): NEXT K
890  NEXT N,J
895  REM
900  REM    INVERSION TESTER
901 X = 0
902  FOR K = 0 TO 3
903  FOR I = 0 TO 3:A(K,I) = 0
904  FOR J = 0 TO 3
906 A(K,I) = G(K,J) * GI(J,I) + A(K,I)
908  NEXT J:X = X + A(K,I)
909  NEXT I,K
912  PRINT "INVERSION TEST = "X
915  RETURN


917  REM
920  REM   GENERATE CONCENTRATED LOADS AND STORE IN THE B(I) AND XB(I)
930  REM   ARRAYS COMMON TO THE CONCENTRATED LOAD INPUT
950  REM
955  IF NL = 0 THEN  RETURN
960  PRINT "GENERATING CONCENTRATED LOADS": LET N = NC
965  REM
970  REM   OUTER LOOP FOR EACH CONTINUOUS LOAD
980  FOR I = 1 TO NL:NCA(0) = (WL(I,1) - WL(I,0)) / (XL(I,1) - XL(I,0)):XB(0) = (XL(I,1) - XL(I,0)) / NCA(I): REM  SLOPE AND INCREMENT OF X
985  REM
990  REM   INNER LOOP FOR EACH CONC. LOAD TO APPROXIMATE
1000  FOR J = 1 TO NCA(I):N = N + 1:XB(N) = XL(I,0) + XB(0) * (J - .5):B(N) = (NCA(0) * (XB(N) - XL(I,0)) + WL(I,0)) * XB(0): NEXT J,I: RETURN
```

```
1040  REM
1050  REM   PROBLEM INPUT------NO FRILLS ADDED
1060  INPUT "SPRING CONSTANT = ";SK
1070  INPUT "E = ";E
1080  INPUT "I = ";IZ
1090  INPUT "L = ";L
1100 BETA = (SK / E / IZ / 4) ↑ .25:EPSIL = L / 10000
1105  REM
1110  REM   LOADING CONDITION INPUT
1120  REM   CONTINUOUS LOAD INPUT
1130  INPUT "NUMBER OF CONTINUOUS LOADS = ";NL
1140  IF NL = 0 THEN  GOTO 1220
1150  DIM XL(NL,1),WL(NL,1),NCA(NL)
1160  PRINT
1170  FOR I = 1 TO NL: PRINT "CONTINUOUS LOAD NUMBER "I
1180  INPUT "POSITION AND VALUE OF LOAD ON            LEFT END = ";XL(I,0),WL(I,0)
1190  INPUT "POSITION AND VALUE OF LOAD ON            RIGHT END = ";XL(I,1),WL(I,1)
1200  INPUT "NUMBER OF CONCENTRATED LOADS TO APPROX-  IMATE THE CONTINUOUS LOAD = ";NCA(I)
1210 NT = NT + NCA(I): NEXT : PRINT
1215  REM
1220  REM   CONCENTRATED LOAD INPUT
1230  INPUT "NUMBER OF CONCENTRATED LOADS = ";NC
1233  IF NC < 0 OR NC > 20 GOTO 1230
1236 NT = NT + NC: REM   TOTAL NO. OF CONC. LOADS
1240  DIM B(NT + 2),XB(NT + 2),HP(3,NT - 1),KB(3,NT + 1)
1250  IF NC = 0 THEN  GOTO 1270
1260  FOR I = 1 TO NC: PRINT "CONCENTRATED LOAD NUMBER "I: INPUT "POSTION AND VALUE OF LOAD = ";XB(I),B(I): NEXT
1265  REM
1270  REM   BOUNDARY CONDITION INPUT
1280  REM
1290  PRINT : PRINT "BOUNDARY CONDITION INPUT"
1300  PRINT "ENTER APPROPRIATE NUMERIC CODE FOR KNOWN BOUNDARY CONDITION":N = 5
1310  PRINT  TAB( N)"1. DISPLACEMENT"
1320  PRINT  TAB( N)"2. ROTATION"
1330  PRINT  TAB( N)"3. MOMENT"
1340  PRINT  TAB( N)"4. SHEAR"
1350  PRINT
1360  FOR N = 1 TO 4: ON N GOSUB 1410,1400,1420,1400: GOSUB 1430: NEXT : RETURN
1400  PRINT : RETURN
```

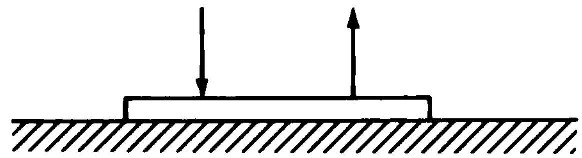
```
1410  PRINT : PRINT "LEFT BOUNDARY": PRINT : RETURN
1420  PRINT : PRINT "RIGHT BOUNDARY": PRINT : RETURN
1430  PRINT "BOUNDARY CONDITION "N", CODE =";: INPUT " ";BC(N - 1)
1440  ON BC(N - 1) GOSUB 1460,1470,1480,1490
1450  PRINT BC$" VALUE =";: INPUT " ";F(N - 1): RETURN
1460 BC$ = "DISPLACEMENT": RETURN
1470 BC$ = "ROTATION": RETURN
1480 BC$ = "MOMENT": RETURN
1490 BC$ = "SHEAR": RETURN
1540  REM   BEGINNING OF PROGRAM
1670  DIM G(3,3),PHI(3),F(3),BC(3),GI(3,3),A(3,3),XM(1),M(1),KM(3,1)
1680  HOME : PRINT " BOUNDARY INTEGRAL METHOD"
1690  GOSUB 1050: REM   INPUT
1695  GOSUB 920: REM   GENERATOR
1700  GOSUB 600: REM   MATRICES DEVELOPMENT
1710  GOSUB 750: REM   MATRIX INVERSION
1720  GOSUB 730: REM   SOLVE
1725  GOSUB 1800: REM   RESPONSE INPUT
1730  GOSUB 250: REM   RESPONSE
1740  GOSUB 3000: REM  OUTPUT
1750  END


1799  REM
1800  REM   RESPONSE INPUT SUBROUTINE-----NO FRILLS
1810  REM
1820  PRINT "RESPONSE INPUT": PRINT
1830  INPUT "NUMBER OF RESPONSE POINTS = ";NK
1835  IF NK < 1 OR NK > 50 THEN  GOTO 1830
1840  DIM XA(NK - 1),R(3,NK - 1)
1850  FOR N = 0 TO NK - 1
1860  PRINT "RESPONSE POINT "N + 1" POSITION =": INPUT " ";XA(N)
1870  IF XA(N) < 0 OR XA(N) > L GOTO 1860
1880  NEXT
1890  RETURN
```

```
2999 REM
3000 REM  OUTPUT ROUTINE
3001 PR# 1:I$ =  CHR$ (9)
3002 PRINT I$;"80N"
3003 LM = 10
3004 PRINT  SPC( LM)"BEAM ON AN ELASTIC FOUNDATION"
3006 PRINT  SPC( LM)"INDIRECT BOUNDARY INTEGRAL METHOD"
3007 PRINT  SPC( LM)"THEORY: DR. TED SHUGAR"
3010 PRINT  SPC( LM)"PROGRAM: JAMES V. COX"
3013 J = 79 - LM:L$ = ""
3015 FOR I = 1 TO J:L$ = L$ + "-": NEXT
3020 PRINT  SPC( LM)L$
3025 PRINT  SPC( 35 - LM / 2)"USER INPUT"
3030 PRINT  SPC( LM)L$
3060 PRINT  SPC( LM)"SPRING" SPC( 8)"MODULUS OF" SPC( 6)"MOMENT OF" SPC( 5)""
3070 PRINT  SPC( LM)"CONSTANT" SPC( 6)"ELASTICITY" SPC( 6)"INERTIA" SPC( 7)"SPAN"
3080 A = SK:W% = 9:D% = 2: GOSUB 60000: PRINT  SPC( LM + 9 -  LEN (A$));A$;
3090 A = E:W% = 11:D% = 2: GOSUB 60000: PRINT  SPC( 16 -  LEN (A$));A$;
3100 A = IZ:W% = 9:D% = 2: GOSUB 60000: PRINT  SPC( 14 -  LEN (A$));A$;
3110 A = L:W% = 7:D% = 2: GOSUB 60000: PRINT  SPC( 12 -  LEN (A$));A$
3120 IF NL = 0 GOTO 3325
3200 PRINT
3210 PRINT  SPC( LM)"CONTINUOUS" SPC( 3)"LEFT END" SPC( 1)"" SPC( 11)"RIGHT END" SPC( 1)"" SPC( 11)"NO. OF LOADS"
3220 PRINT  SPC( LM)"LOAD NO." SPC( 5)"POSITION" SPC( 3)"VALUE" SPC( 4)"POSITION" SPC( 4)"VALUE" SPC( 4)"TO APPROXIMATE"
3230 PRINT  SPC( LM)"----------" SPC( 3)"--------" SPC( 3)"-----" SPC( 4)"--------" SPC( 4)"-----" SPC( 4)"--------------"
3240 FOR I = 1 TO NL
3250 A = I:W% = 10:D% = 0: GOSUB 60000: PRINT  SPC( LM + 10 -  LEN (A$));A$;
3260 A = XL(I,0):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 11 -  LEN (A$));A$;
3270 A = WL(I,0):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 9 -  LEN (A$));A$;
3280 A = XL(I,1):W% = 9:D% = 2: GOSUB 60000: PRINT  SPC( 11 -  LEN (A$));A$;
3290 A = WL(I,1):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 9 -  LEN (A$));A$;
3300 A = NCA(I):W% = 13:D% = 0: GOSUB 60000: PRINT  SPC( 16 -  LEN (A$));A$
3320 NEXT I
3325 DEBUG$ = "N":J = 1:K = NC
3330 IF NC = 0 GOTO 3470
3350 PRINT
3360 PRINT  SPC( LM)"CONCENTRATED"
3370 PRINT  SPC( LM)"LOAD NO." SPC( 8)"POSITION" SPC( 10)"VALUE"
3380 PRINT  SPC( LM)"--------" SPC( 8)"--------" SPC( 10)"-----"
```

```
3390  FOR I = J TO K
3400 A = I:W% = 6:D% = 0: GOSUB 60000: PRINT  SPC( LM + 6 -  LEN (A$));A$;
3410 A = XB(I):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 18 -  LEN (A$));A$;
3420 A = B(I):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 16 -  LEN (A$));A$
3440  NEXT I
3470  PRINT
3475  IF DEBUG$ = "Y" GOTO 3860
3480  PRINT  SPC( LM + 10)"LEFT END" SPC( 22)"RIGHT END"
3490  PRINT  SPC( LM + 10)"--------" SPC( 22)"---------"
3500  FOR I = 0 TO 1
3505  ON BC(I) GOSUB 1460,1470,1480,1490
3506 BC$ = BC$ + " ="
3510 A$ = BC$: PRINT  SPC( LM + 14 -  LEN (A$))A$;
3520 A = F(I):W% = 10:D% = 2: GOSUB 60000: PRINT  SPC( 12 -  LEN (A$));A$;
3525  ON BC(I + 2) GOSUB 1460,1470,1480,1490
3526 BC$ = BC$ + " ="
3530 A$ = BC$: PRINT  SPC( 19 -  LEN (A$))A$;
3540 A = F(I + 2):W% = 10:D% = 2: GOSUB 60000: PRINT  SPC( 12 -  LEN (A$));A$
3560  NEXT I
3570  PRINT : PRINT  SPC( LM)L$: PRINT  SPC( 37 - LM / 2)"OUTPUT": PRINT  SPC( LM)L$
3600  PRINT
3610  PRINT  SPC( LM)"RESPONSE"
3620  PRINT  SPC( LM)"POINT NO." SPC( 3)"POSITION" SPC( 3)"SHEAR" SPC( 6)"MOMENT" SPC( 5)"DISPLACEMENT" SPC( 3)"ROTATION"
3630  PRINT  SPC( LM)"---------" SPC( 3)"--------" SPC( 3)"-----" SPC( 6)"------" SPC( 5)"------------" SPC( 3)"--------"
3640  FOR I = 1 TO NK
3650 A = I:W% = 9:D% = 0: GOSUB 60000: PRINT  SPC( LM + 9 -  LEN (A$));A$;
3660 A = XA(I - 1):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 11 -  LEN (A$));A$;
3670 A = R(3,I - 1):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 11 -  LEN (A$));A$;
3680 A = R(2,I - 1):W% = 8:D% = 2: GOSUB 60000: PRINT  SPC( 11 -  LEN (A$));A$;
3690 A = R(0,I - 1):W% = 6:D% = 3: GOSUB 60000: PRINT  SPC( 9 -  LEN (A$));A$;
3700 A = R(1,I - 1):W% = 8:D% = 3: GOSUB 60000: PRINT  SPC( 17 -  LEN (A$));A$
3720  NEXT I
3800  REM  OPTIONAL DEBUG PRINTOUT
3805  PR# 0
3810  INPUT "DO YOU WANT THE DEBUG PRINTOUT (Y/N)?";DEBUG$
3820  IF DEBUG$ < > "Y" AND DEBUG$ < > "N" GOTO 3810
3830  IF DEBUG$ = "N" THEN  RETURN
3831 J = NC + 1:K = NT + 2
3833  PR# 1
3836  PRINT I$;"80N"
```

```
3840  PRINT  CHR$ (12)
3850  GOTO 3350
3860  REM  MOMENT LIST
4000  REM  G MATRIX OUTPUT
4010  PRINT  SPC( LM)"G MATRIX"
4020  PRINT  SPC( LM + 9)"0" SPC( 12)"1" SPC( 12)"2" SPC( 12)"3"
4030  FOR I = 0 TO 3
4040 A = I:W% = 3:D% = 0: GOSUB 60000: PRINT  SPC( LM + 3 -  LEN (A$));A$;
4050  FOR J = 0 TO 3
4060 A = G(I,J):W% = 10:D% = 3: GOSUB 60000: PRINT  SPC( 13 -  LEN (A$));A$;
4070  NEXT J
4080  PRINT
4100  NEXT I
4110  PRINT
4150  REM  GI MATRIX OUTPUT
4160  PRINT  SPC( LM)"GI MATRIX"
4170  PRINT  SPC( LM + 9)"0" SPC( 12)"1" SPC( 12)"2" SPC( 12)"3"
4180  FOR I = 0 TO 3
4190 A = I:W% = 3:D% = 0: GOSUB 60000: PRINT  SPC( LM + 3 -  LEN (A$));A$;
4200  FOR J = 0 TO 3
4210 A = GI(I,J):W% = 10:D% = 3: GOSUB 60000: PRINT  SPC( 13 -  LEN (A$));A$;
4220  NEXT J
4230  PRINT
4250  NEXT I
59000  RETURN


59999  REM FORMAT ROUTINE
60000 A =  INT (A * 10 ↑ D% + .5) /  INT (.1 + 10 ↑ D%):A$ =  STR$ (A):A =  LEN (A$)
60010  IF D% = 0 THEN  RETURN
60013  IF A $  = D% GOTO 60020
60016  IF  MID$ (A$,A - D%,1) =  CHR$ (46) THEN  RETURN
60020  FOR ZQ = 1 TO A: IF  MID$ (A$,ZQ,1) =  CHR$ (46) THEN  GOTO 60040
60025  IF  MID$ (A$,ZQ,1) = "E" THEN  RETURN
60030  NEXT ZQ:A$ = A$ +  CHR$ (46):A = D%: GOTO 60050
60040 A = ZQ + D% - A
60050  FOR ZQ = 1 TO A:A$ = A$ +  CHR$ (48): NEXT ZQ: RETURN
```

BEAM ON AN ELASTIC FOUNDATION
INDIRECT BOUNDARY INTEGRAL METHOD
THEORY: DR. TED SHUGAR
PROGRAM: JAMES V. COX



USER INPUT

SPRING CONSTANT	MODULUS OF ELASTICITY	MOMENT OF INERTIA	SPAN
1000.00	10000000.00	25.00	80.00

CONCENTRATED LOAD NO.	POSITION	VALUE
1	20.00	100000.00
2	60.00	-100000.00

	LEFT END		RIGHT END
MOMENT =	0.00	MOMENT =	0.00
SHEAR =	0.00	SHEAR =	0.00

OUTPUT

RESPONSE POINT NO.	POSITION	SHEAR	MOMENT	DISPLACEMENT	ROTATION
1	.01	0.00	0.00	3.506	-.070
2	10.00	31536.83	163395.97	2.803	-.072
3	20.01	-44185.37	606177.57	2.024	-.087
4	30.00	-28616.21	250467.73	1.064	-.103
5	40.00	-23254.13	0.00	0.000	-.108
6	50.00	-28616.21	-250467.73	-1.064	-.103
7	60.01	55782.23	-606084.80	-2.026	-.087
8	70.00	31536.83	-163395.97	-2.803	-.072
9	79.99	0.00	0.00	-3.506	-.070

CONCENTRATED LOAD NO.	POSITION	VALUE
3	0.00	145799.80
4	80.00	-145799.80

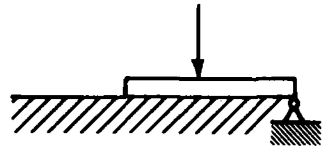
G MATRIX

	0	1	2	3
0	7.902	-.500	-.878	-.033
1	-.500	.016	-.033	0.000
2	-.878	.033	7.902	.500
3	.033	0.000	.500	.016

GI MATRIX

	0	1	2	3
0	-.128	-4.086	9E-03	-.639
1	-4.037	-64.853	.110	-13.050
2	9E-03	.639	-.128	4.086
3	-.110	-13.050	4.037	-64.853

BEAM ON AN ELASTIC FOUNDATION
INDIRECT BOUNDARY INTEGRAL METHOD
THEORY: DR. TED SHUGAR
PROGRAM: JAMES V. COX



USER INPUT

SPRING CONSTANT	MODULUS OF ELASTICITY	MOMENT OF INERTIA	SPAN
1000.00	10000000.00	25.00	40.00

CONCENTRATED LOAD NO.	POSITION	VALUE
1	20.00	100000.00

LEFT END		RIGHT END	
MOMENT =	0.00	DISPLACEMENT =	0.00
SHEAR =	0.00	MOMENT =	0.00

OUTPUT

RESPONSE POINT NO.	POSITION	SHEAR	MOMENT	DISPLACEMENT	ROTATION
1	0.00	0.00	0.00	3.505	-.070
2	10.00	31541.23	163486.32	2.802	-.072
3	20.00	-44196.82	606448.52	2.024	-.087
4	30.00	-28625.78	250496.21	1.064	-.103
5	40.00	-23268.66	0.00	0.000	-.108

CONCENTRATED

LOAD NO.	POSITION	VALUE
2	0.00	145741.48
3	40.00	-77289.86

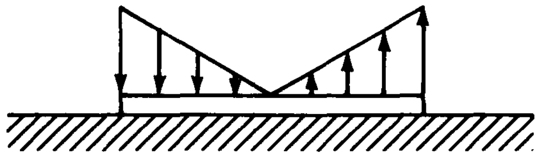
G MATRIX

	0	1	2	3
0	7.904	-.500	-1.456	.043
1	-.500	.016	.043	6E-03
2	0.000	0.000	0.000	0.000
3	-1.456	-.043	7.904	.500

GI MATRIX

	0	1	2	3
0	-.138	-4.724	-32267.163	.065
1	-4.146	-77.903	-784819.603	1.226
2	-.022	.347	61323.234	-2E-03
3	-.408	-25.876	-1130194.410	2.325

BEAM ON AN ELASTIC FOUNDATION
INDIRECT BOUNDARY INTEGRAL METHOD
THEORY: DR. TED SHUGAR
PROGRAM: JAMES V. COX



USER INPUT

SPRING CONSTANT	MODULUS OF ELASTICITY	MOMENT OF INERTIA	SPAN
1000.00	10000000.00	25.00	80.00

CONTINUOUS LOAD NO.	LEFT END POSITION	VALUE	RIGHT END POSITION	VALUE	NO. OF LOADS TO APPROXIMATE
1	0.00	20.00	80.00	-20.00	20

LEFT END		RIGHT END	
MOMENT =	0.00	MOMENT =	0.00
SHEAR =	0.00	SHEAR =	0.00

OUTPUT

RESPONSE POINT NO.	POSITION	SHEAR	MOMENT	DISPLACEMENT	ROTATION
1	.01	0.00	0.00	.020	0.000
2	10.01	-29.41	32.86	.015	0.000
3	20.00	-.74	3.34	.010	0.000
4	30.01	-9.81	11.93	5E-03	0.000
5	40.00	-.88	0.00	0.000	0.000
6	50.01	10.11	-11.93	-5E-03	0.000
7	60.00	-.74	-3.34	-.010	0.000
8	70.01	30.36	-32.85	-.015	0.000
9	79.99	0.00	0.00	-.020	0.000

CONCENTRATED LOAD NO.	POSITION	VALUE
1	2.00	76.00
2	6.00	68.00
3	10.00	60.00
4	14.00	52.00
5	18.00	44.00
6	22.00	36.00
7	26.00	28.00
8	30.00	20.00
9	34.00	12.00
10	38.00	4.00
11	42.00	-4.00
12	46.00	-12.00
13	50.00	-20.00
14	54.00	-28.00
15	58.00	-36.00
16	62.00	-44.00
17	66.00	-52.00
18	70.00	-60.00
19	74.00	-68.00
20	78.00	-76.00
21	0.00	880.14
22	80.00	-880.14

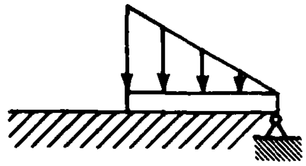
G MATRIX

	0	1	2	3
0	7.902	-.500	-.878	-.033
1	-.500	.016	-.033	0.000
2	-.878	.033	7.902	.500
3	.033	0.000	.500	.016

GI MATRIX

	0	1	2	3
0	-.128	-4.086	9E-03	-.639
1	-4.037	-64.853	.110	-13.050
2	9E-03	.639	-.128	4.086
3	-.110	-13.050	4.037	-64.853

BEAM ON AN ELASTIC FOUNDATION
INDIRECT BOUNDARY INTEGRAL METHOD
THEORY: DR. TED SHUGAR
PROGRAM: JAMES V. COX



USER INPUT

SPRING CONSTANT	MODULUS OF ELASTICITY	MOMENT OF INERTIA	SPAN
1000.00	10000000.00	25.00	40.00

CONTINUOUS LOAD NO.	LEFT END POSITION	VALUE	RIGHT END POSITION	VALUE	NO. OF LOADS TO APPROXIMATE
1	0.00	20.00	40.00	0.00	10

LEFT END		RIGHT END	
MOMENT =	0.00	DISPLACEMENT =	0.00
SHEAR =	0.00	MOMENT =	0.00

OUTPUT

RESPONSE POINT NO.	POSITION	SHEAR	MOMENT	DISPLACEMENT	ROTATION
1	0.00	0.00	0.00	.020	0.000
2	10.00	-29.43	33.52	.015	0.000
3	20.00	-.74	4.01	.010	0.000
4	30.00	-9.86	12.43	5E-03	0.000
5	40.00	-.94	0.00	0.000	0.000

CONCENTRATED LOAD NO.	POSITION	VALUE
1	2.00	76.00
2	6.00	68.00
3	10.00	60.00
4	14.00	52.00
5	18.00	44.00
6	22.00	36.00
7	26.00	28.00
8	30.00	20.00
9	34.00	12.00
10	38.00	4.00
11	0.00	879.82
12	40.00	-250.66

G MATRIX

	0	1	2	3
0	7.904	-.500	-1.456	.043
1	-.500	.016	.043	6E-03
2	0.000	0.000	0.000	0.000
3	-1.456	-.043	7.904	.500

GI MATRIX

	0	1	2	3
0	-.138	-4.724	-32267.163	.065
1	-4.146	-77.903	-784819.603	1.226
2	-.022	.347	61323.234	-2E-03
3	-.408	-25.876	-1130194.410	2.325

1

Annotated User Interaction

]RUN
BOUNDARY INTEGRAL METHOD
SPRING CONSTANT = 1000
E = 10E6
I = 25
L = 100
NUMBER OF CONTINUOUS LOADS = 1

CONTINUOUS LOAD NUMBER 1
POSITION AND VALUE OF LOAD ON
LEFT END = 20,2000
POSITION AND VALUE OF LOAD ON
RIGHT END = 60,500
NUMBER OF CONCENTRATED LOADS TO APPROX-
IMATE THE CONTINUOUS LOAD

NUMBER OF CONCENTRATED LOADS = 1
CONCENTRATED LOAD NUMBER 1
POSTION AND VALUE OF LOAD = 80,10E3

BOUNDARY CONDITION INPUT
ENTER APPROPRIATE NUMERIC CODE FOR KNOWN
BOUNDARY CONDITION

1. DISPLACEMENT
2. ROTATION
3. MOMENT
4. SHEAR

LEFT BOUNDARY

BOUNDARY CONDITION 1, CODE = 1
DISPLACEMENT VALUE = 0

BOUNDARY CONDITION 2, CODE = 3
MOMENT VALUE = 0

RIGHT BOUNDARY

BOUNDARY CONDITION 3, CODE = 2
ROTATION VALUE = 0

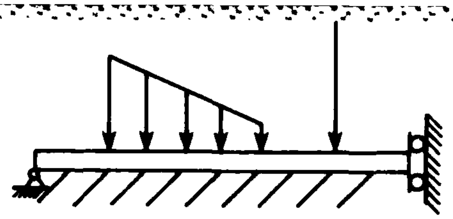
BOUNDARY CONDITION 4, CODE = 4
SHEAR VALUE = 0
•GENERATING CONCENTRATED LOADS
•MATRIX DEVELOPEMENT
G MATRIX
HP MATRIX
•MATRIX INVERSION
INVERSION TEST = 4.17263406
•SOLVE FOR ACTUAL BEAM UNKNOWNS

•RESPONSE INPUT

NUMBER OF RESPONSE POINTS = 11
RESPONSE POINT 1 POSITION =
0
RESPONSE POINT 2 POSITION =
10
RESPONSE POINT 3 POSITION =
20
RESPONSE POINT 4 POSITION =
30
RESPONSE POINT 5 POSITION =
40
RESPONSE POINT 6 POSITION =
50
RESPONSE POINT 7 POSITION =
60
RESPONSE POINT 8 POSITION =
70
RESPONSE POINT 9 POSITION =
80
RESPONSE POINT 10 POSITION =
90
RESPONSE POINT 11 POSITION =
100
•RESPONSE
RESPONSE PT 1
RESPONSE PT 2
RESPONSE PT 3
RESPONSE PT 4
RESPONSE PT 5
RESPONSE PT 6
RESPONSE PT 7
RESPONSE PT 8
RESPONSE PT 9
RESPONSE PT 10
RESPONSE PT 11

Underlined quantities represent user input.
• - indicates the printout of a subroutine name

BEAM ON AN ELASTIC FOUNDATION
INDIRECT BOUNDARY INTEGRAL METHOD
THEORY: DR. TED SHUGAR
PROGRAM: JAMES V. COX



USER INPUT

SPRING CONSTANT	MODULUS OF ELASTICITY	MOMENT OF INERTIA	SPAN
1000.00	10000000.00	25.00	100.00

CONTINUOUS LOAD NO.	LEFT END POSITION	VALUE	RIGHT END POSITION	VALUE	NO. OF LOADS TO APPROXIMATE
1	20.00	2000.00	60.00	500.00	20

CONCENTRATED LOAD NO.	POSITION	VALUE
1	80.00	10000.00

LEFT END		RIGHT END	
DISPLACEMENT =	0.00	ROTATION =	0.00
MOMENT =	0.00	SHEAR =	0.00

OUTPUT

RESPONSE POINT NO.	POSITION	SHEAR	MOMENT	DISPLACEMENT	ROTATION
1	.01	7596.64	0.00	0.000	.028
2	10.00	8965.02	80464.25	.271	.026
3	20.00	12914.83	187869.61	.510	.021
4	30.00	790.72	252020.07	.675	.012
5	40.00	-6419.81	220317.12	.742	2E-03
6	50.00	-9662.24	137073.14	.722	-5E-03
7	60.00	-9651.91	38122.64	.649	-9E-03
8	70.00	-3616.04	-27461.57	.558	-9E-03
9	80.01	-8444.32	-37195.25	.477	-7E-03
10	90.00	-4005.30	-98869.39	.416	-4E-03
11	99.99	0.00	-118683.19	.393	0.000

DO YOU WANT THE DEBUG PRINTOUT (Y/N)?Y

CONCENTRATED LOAD NO.	POSITION	VALUE
2	21.00	3925.00
3	23.00	3775.00
4	25.00	3625.00
5	27.00	3475.00
6	29.00	3325.00
7	31.00	3175.00
8	33.00	3025.00
9	35.00	2875.00
10	37.00	2725.00
11	39.00	2575.00
12	41.00	2425.00
13	43.00	2275.00
14	45.00	2125.00
15	47.00	1975.00
16	49.00	1825.00
17	51.00	1675.00
18	53.00	1525.00
19	55.00	1375.00
20	57.00	1225.00
21	59.00	1075.00
22	0.00	-21444.46
23	100.00	12439.27

G MATRIX

	0	1	2	3
0	0.000	0.000	0.000	0.000
1	7.901	-.500	-.328	-.021
2	0.000	0.000	0.000	0.000
3	.021	-1E-03	.500	.016

GI MATRIX

	0	1	2	3
0	63152.345	0.000	41453.797	.086
1	998138.796	-1.998	1337181.850	.053
2	-2674.363	0.000	998913.625	1.996
3	43117.033	-.083	-31576172.600	.025

Appendix D

BIM2D LISTINGS

```
      PROGRAM BIM2D(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7)
C
C     MAIN PROGRAM
C
      DIMENSION X(152),Y(152),BC(304),KBC(152),P(304),LNODE(5)
      REAL MU,K(304,304),LAMDA
      COMMON /CEES/ C0,C1,C2,C3,C4
      MK = 304
C  INPUT AND INPUT PRINT ROUTINES
      CALL INFO(X,Y,KBC,BC,NOE,E,MU,KPROB,LNODE,MK)
      CALL INPRNT(X,Y,KBC,BC,NOE,E,MU,KPROB,LNODE)
      PI = 2.*ACOS(0.)
C
C  CALCULATIONS WHICH ARE FUNCTIONS OF THE MATERIAL ONLY
C
      G = E/2./(1.+MU)
C  PLANE STRAIN
      LAMDA = E*MU/(1.+MU)/(1.-2.*MU)
C  PLANE STRESS
      IF(KPROB.EQ.0) LAMDA = 2.*LAMDA*G/(LAMDA+2.*G)
C
      C0 = 1. + 2.*(LAMDA+G)/G
      C4 = PI*(LAMDA + 2.*G)
      C1 = G/2./C4
      C2 = (LAMDA+G)/C4
      C3 = (LAMDA+3.*G)/4./C4/G
      C4 = (LAMDA+G)/4./G/C4
C
C  INFLUENCE MATRIX DEVELOPMENT ROUTINE
      CALL KMAKER(NOE,LAMDA,G,PI,X,Y,K,KBC,MK,LNODE)
C
C
C  SOLVE FOR ARTIFICIAL BOUNDARY LOADS
      N = 2*NOE
      CALL AXEQB(K,P,BC,N,MK)
C
C
C  CALCULATE RESPONSE AT SPECIFIED POINTS
      CALL RESPON(NOE,LAMDA,G,PI,X,Y,P,LNODE)
      STOP
      END
```

```
      SUBROUTINE MANUAL
C
C
C
C     VARIABLE LIST
C     =============
C
C     A:  A VALUE FOR NONSINGULAR INTEGRATIONS.
C     B:       " "
C     BC(I),BC(I1): BOUNDARY TRACTIONS OR DISPLACEMENTS FOR ELEMENT (I+1)/2
C     CTHETM:  COS(THETAM).
C     CTHETN:  COS(THETAN).
C     CO:  A CONSTANT FOR A GIVEN MATERIAL.
C     C1:        " "
C     C2:        " "
C     C3:        " "
C     C4:        " "
C     CSX:  .EQ.0: CALCULATE SX RESPONSE AT A GIVEN POINT.
C     CSY:       " "          SY
C     CT:        " "          T
C     CU:        " "          U
C     CV:        " "          V
C     DL:  ELEMENT LENGTH.
C     DX:  FOR A GIVEN ELEMENT = X(N) - X(N+1).
C     DY:  FOR A GIVEN ELEMENT = Y(N) - Y(N+1).
C     E:   MODULUS OF ELASTICITY.
C     EC:  SQUARE OF DISTANCE FROM CENTER OF M TO NODE 1 OF N.
C     EE:  X DISTANCE FROM CENTER OF M TO NODE 1 OF N.
C     EG:  Y DISTANCE FROM CENTER OF M TO NODE 1 OF N.
C     EPSIL:  A TOLERANCE USED TO DECIDE BETWEEN INTEGRAL EQUATIONS.
C     G:  SHEAR MODULUS OF ELASTICITY.
C     GAM1:  INTEGRATIONS OVER THE LENGTH OF ELEMENT N.
C     .            " "
C     .            " "
C     .            " "
C     GAM10:       " "
C     I:      (K) FIRST SUBSCRIPT.
C     I1:     I+1
C     J:      (K) SECOND SUBSCRIPT.
C     J1:     J+1
C     JB:     FLAG TO INDICATE A BOUNDARY RESPONSE POINT.
C     K(I,J): INFLUENCE MATRIX TO OBTAIN ARTIFICIAL BOUNDARY LOADS.
C     KBC(M): KNOWN BOUNDARY QUANTITIES .EQ.0: STRESSES AND .NE.0: DISPLACEMENTS.
C     KPROB:  .EQ.0: PLANE STRESS AND .NE.0: PLANE STRAIN.
C     LAMDA:  LAME' CONSTANT.
C     LNODE(NBOUND): LAST NODE OF BOUNDARY NBOUND.
C     M:      OUTER LOOP ELEMENT NUMBER TO DEVELOP 2 (K) ROWS.
C     MBOUND: A BOUNDARY NUMBER COUNTER.
C     M1:     M+1
C     MK:     MAXIMUM ORDER OF (K), NOE * 2
C     MU:     POISSON'S RATIO.
```

```
C     N:       INNER LOOP ELEMENT NUMBER TO DEVELOP 2 (K) COLUMNS.
C     NBOUND: A BOUNDARY NUMBER COUNTER.
C     N1:      N+1
C     NOE:     NUMBER OF BOUNDARY ELEMENTS.
C     P(I),P(I1): ARTIFICIAL BOUNDARY LOADS AT ELEMENT (I+1)/2
C     PI:      3.14159...
C     THETAM: ANGLE OF ELEMENT M.
C     THETAN: ANGLE OF ELEMENT N.
C     STHETM: SIN(THETAM).
C     STHETN: SIN(THETAN).
C     SXN:     SX INFLUENCE COEF DUE TO NORMAL BOUNDARY STRESS.
C     SXT:     SX INFLUENCE COEF DUE TO TANGENTIAL BOUNDARY STRESS.
C     SXR:     SX STRESS AT A GIVEN RESPONSE POINT.
C     SYN:     SY INFLUENCE COEF DUE TO NORMAL BOUNDARY STRESS.
C     SYT:     SY INFLUENCE COEF DUE TO TANGENTIAL BOUNDARY STRESS.
C     SYR:     SY STRESS AT A GIVEN RESPONSE POINT.
C     TN:      T INFLUENCE COEF DUE TO NORMAL BOUNDARY STRESS.
C     TT:      T INFLUENCE COEF DUE TO TANGENTIAL BOUNDARY STRESS.
C     TR:      T STRESS AT A GIVEN RESPONSE POINT.
C     UN:      U INFLUENCE COEF DUE TO NORMAL BOUNDARY STRESS.
C     UP:      UPPER LIMIT ON ELEMENT INTEGRATION.
C     UT:      U INFLUENCE COEF DUE TO TANGENTIAL BOUNDARY STRESS.
C     UR:      U DISPLACEMENT AT A GIVEN RESPONSE POINT.
C     VN:      V INFLUENCE COEF DUE TO NORMAL BOUNDARY STRESS.
C     VT:      V INFLUENCE COEF DUE TO TANGENTIAL BOUNDARY STRESS.
C     VR:      V DISPLACEMENT AT A GIVEN RESPONSE POINT.
C     X(N):    X COORDINATE OF NODE N, THE FIRST NODE OF ELEMENT N.
C     XQM:     X COORDINATE OF THE CENTER OF ELEMENT M.
C     XR:      X COORDINATE OF THE RESPONSE POINT.
C     Y(N):    Y COORDINATE OF NODE N, THE FIRST NODE OF ELEMENT N.
C     YQM:     Y COORDINATE OF THE CENTER OF ELEMENT M.
C     YR:      Y COORDINATE OF THE RESPONSE POINT.
C
C
C
C     INPUT FILE
C     ==========
C     THE FOLLOWING DESCRIPTION OF THE INPUT FILE IS WRITTEN AS THOUGH
C     A DECK OF CARDS IS BEING USED AS THE INPUT MEDIUM.  ONE MUST
C     REFER TO THE VARIABLE LIST TO USE THIS BREIF DESCRIPTION.
C
C
C   CARD/SET   DATA DESCRIPTION                             FORMAT
C   --------   ----------------                             ------
C   1          KPROB  (PROBLEM TYPE: PL STRESS/PL STRAIN)   (I5)
C
C
C   2          E,MU   (MATERIAL PROPERTIES)                 (E10,F10)
C
C
C   3          LNODE(1),...,LNODE(5) (LAST NODE NUMBERS)    (5I5)
C
C
```

```
C   SET 4      BOUNDARY DESCRIPTION CARDS
C
C              X(1),Y(1),KBC(1),BC(1),BC(2)             (2F10,I5,2F10)
C              .    .    .      .     .
C              .    .    .      .     .
C              .    .    .      .     .
C              X(NOE),Y(NOE),KBC(NOE),BC(2*NOE-1),BC(2*NOE)
C
C
C   SET 5      RESPONSE SPECIFICATION CARDS
C              IF THE USER INDICATES THAT THE RESPONSE IS BEING
C              CALCULATED ON A BOUNDARY ELEMENT (JB.NE.0) THEN
C              THE ELEMENT NUMBER (I) MUST BE ENTERED, AND BOTH
C              XR AND YR ARE SET TO THE MID-ELEMENT COORDINATES.
C              THE PROGRAM DOES NOT LIMIT THE NUMBER OF RESPONSE
C              POINTS.
C
C              JB,I,,XR,YR,CSX,CSY,CT,CU,CV             (2I5,2F10,5I5)
C              .  .  .  .  .   .   .  .  .
C              .  .  .  .  .   .   .  .  .
C              .  .  .  .  .   .   .  .  .
C              JB,I,,XR,YR,CSX,CSY,CT,CU,CV
C
C
C
C     SUBROUTINE LIST
C     ===============
C     THE FOLLOWING SUBROUTINE LIST INDICATES THE LEVELS OF HIERARCHY
C     WITHIN THE PROGRAM, BUT DOES NOT COMPLETELY DEFINE THE FLOW
C     OF EXECUTION.  SOME OF THE SECOND ORDER, AND LOWER LEVEL,
C     ROUTINES ARE NOT ALWAYS EXECUTED.  FUNCTIONS ARE NOT LISTED.
C
C
C     BIM2D:  MAIN PROGRAM
C
C        INFO:  INPUTS DATA, EXCEPT FOR RESPONSE DATA.
C
C        INPRNT:  PRINTS THE INPUT DATA.
C
C        KMAKER:  GENERATES THE INFLUENCE COEFFICIENT MATRIX (K).
C           PREK:  CALCULATES THETAM, AND MATRIX ROW NUMBERS I AND I1.
C           PREK:  CALCULATES THETAN, AND MATRIX COL NUMBERS J AND J1.
C           PREGAM:  CALCULATES VALUES IN PREPARATION FOR NONSINGULAR
C                    INTEGRATIONS, UP, EE, EG, A, B, AND EC.
C           GAMMAF:  CALCULATES NONSINGULAR INTEGRATIONS WHEN ELEMENT
C                    M BOUNDARY STRESSES ARE KNOWN.
C           SXC:  CONTROL ROUTINE FOR SXN AND SXT CALCULATIONS SENDS
                  CONTROL TO SINGULAR OR NONSINGULAR ROUTINE.
C              SXNS:  CALCULATES NONSINGULAR SXN AND SXT.
C              SXS:   CALCULATES SINGULAR SXN AND SXT.
C           SYC:  SY CONTROL ROUTINE.
C              SYNS:  NONSINGULAR CALCULATIONS.
C              SYS:   SINGULAR CALCULATIONS
C           TC:   T CONTROL ROUTINE.
```

```
C             TNS:   NONSINGULAR CALCULATIONS.
C             TS:    SINGULAR CALCULATIONS.
C
C          PREK:  CALCULATES THETAN, AND MATRIX COL NUMBERS J AND J1.
C          PREGAM:  CALCULATES VALUES IN PREPARATION FOR NONSINGULAR
C                   INTEGRATIONS, UP, EE, EG, A, B, AND EC.
C          GAMMAU:  CALCULATES NONSINGULAR INTEGRATIONS WHEN ELEMENT
C                   M BOUNDARY DISPLACEMENTS ARE KNOWN.
C          UC:   U CONTROL ROUTINE.
C             UNS:  NONSINGULAR CALCULATIONS.
C             US:   SINGULAR CALCULATIONS.
C          VC:   V CONTROL ROUTINE.
C             VNS:  NONSINGULAR CALCULATIONS.
C             VS:   SINGULAR CALCULATIONS.
C
C       AXEQB:  MATRIX SOLUTION ROUTINE TO SOLVE FOR ARTIFICIAL
C               BOUNDARY LOADS.
C          FACTOR
C          SUBST
C
C       RESPON:  READS INPUT DATA FOR RESPONSE CALCULATIONS,
C                CALCULATES THE USER SPECIFIED RESPONSES,
C                AND PRINTS THE CALCULATED RESPONSES.
C          THE ROUTINES CALLED FROM RESPON ARE NOT LISTED.  THEY ARE
C          MUCH THE SAME AS THOSE CALLED FROM KMAKER.
C
      RETURN
      END
```

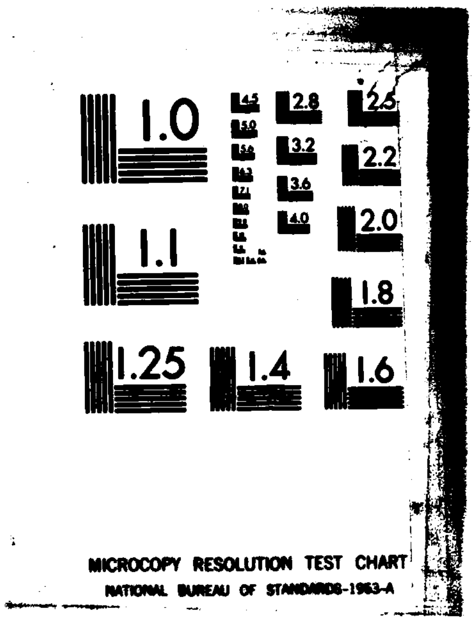
```
      SUBROUTINE INFO(X,Y,KBC,BC,NOE,E,MU,KPROB,LNODE,MK)
C
C  THIS ROUTINE READS ALL THE INPUT DATA EXCEPT FOR THE RESPONSE
C  POINT DATA.  A TRAILER CARD IS USED TO INDICATE THE END OF THE
C  ELEMENT INPUT, AND THE NUMBER OF ELEMENTS IS COUNTED TO
C  PREVENT TOO MANY ELEMENTS.
C
      DIMENSION X(1),Y(1),KBC(1),BC(1),LNODE(1)
      REAL MU
C
C  ENTER PROBLEM TYPE, PLANE STRAIN OR PLANE STRESS.
C
      READ(5,1000) KPROB
1000  FORMAT(I5)
C
C  ENTER MATERIAL PROPERTIES
C
      READ(5,1010) E,MU
1010  FORMAT(E10.3,F10.3)
C
C  ENTER LAST NODE NUMBER FOR EACH BOUNDARY
C
      READ(5,1050)(LNODE(I),I=1,5)
1050  FORMAT(5I5)
C
C  LOOP FOR ELEMENT INPUT
C
C     INITIALIZE
      NOE = 0
      N = 0
C
10    N = N + 1
      I = 2*N - 1
      I1 = I + 1
C  CHECK FOR EXCESSIVE ELEMENT INPUT
      IF(N.GT.MK/2) GO TO 20

C
C  ELEMENT INPUT
      READ(5,1020)X(N),Y(N),KBC(N),BC(I),BC(I1)
1020  FORMAT(2F10.3,I10,2F10.3)
C
C  CHECK FOR TRAILER
      IF(X(N).GT.7.777E+6) RETURN
      NOE = N
      GO TO 10
C
C
C  N .GT. MK/2
C
20    READ(5,1030)TR
```

AD-A133 142 AN INVESTIGATION OF THE INDIRECT BOUNDARY ELEMENT METHOD IN ONE- AND TWO-..(U) NAVAL CIVIL ENGINEERING LAB PORT HUENEME CA T A SHUGAR ET AL. MAY 83 2/2

UNCLASSIFIED NCEL-TN-1664 F/G 12/1 NL

END

FILMED


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
1030  FORMAT(F10.3)
      IF(TR.GT.7.777E+6) RETURN
      N=MK/2
      WRITE(6,1040) N
1040  FORMAT(1H0,38HNUMBER OF ELEMENTS EXCEEDS MAXIMUM OF ,I2,1H.)
      STOP
      END


      SUBROUTINE INPRNT(X,Y,KBC,BC,NOE,E,MU,KPROB,LNODE)
C
C  THIS ROUTINE PRINTS THE INPUT DATA, NOT INCLUDING RESPONSE INPUT,
C  IN TABULAR FORM.
C
      DIMENSION X(1),Y(1),KBC(1),BC(1),LNODE(1)
      REAL MU
C
C  PRINT PROGRAM TITLE
      WRITE(6,1000)
1000  FORMAT(1H1)
1010  FORMAT(1X,T22,36H************************************)
1020  FORMAT(1X,T22,1H*)
1025  FORMAT(1X,T22,1H*,T57,1H*)
1030  FORMAT(1H+,T57,1H*)
      WRITE(6,1010)
      WRITE(6,1025)
      WRITE(6,1025)
      WRITE(6,1020)
      WRITE(6,1040)
1040  FORMAT(1H+,T37,5HBIM2D)
      WRITE(6,1030)
      WRITE(6,1025)
      WRITE(6,1020)
      WRITE(6,1050)
1050  FORMAT(1H+,T27,27HA BOUNDARY INTEGRAL PROGRAM)
      WRITE(6,1030)
      WRITE(6,1020)
      WRITE(6,1060)
1060  FORMAT(1H+,T27,29HFOR 2D ELASTOSTATICS PROBLEMS)
      WRITE(6,1030)
      WRITE(6,1025)
      WRITE(6,1020)
      WRITE(6,1070)
1070  FORMAT(1H+,T32,17HDEVELOPED AT NCEL)
      WRITE(6,1030)
      WRITE(6,1025)
      WRITE(6,1025)
      WRITE(6,1010)
      WRITE(6,1080)
1080  FORMAT(//////)
```

```
C
      WRITE(6,1110)
      IF(KPROB.EQ.0) WRITE(6,1090)
      IF(KPROB.NE.0) WRITE(6,1100)
1110  FORMAT(1X,24HPROBLEM TYPE:  PLANE STR)
1090  FORMAT(1H+,T26,3HESS)
1100  FORMAT(1H+,T26,3HAIN)
C
C  MATERIAL PROPERTIES
      WRITE(6,1120)
1120  FORMAT(1H0,20HMATERIAL PROPERTIES,T30,1HE,T45,2HMU)
      WRITE(6,1130) E,MU
1130  FORMAT(1X,T22,E10.3,T39,F10.3)
C
C  MULTIPLE BOUNDARY LAST NODES
      WRITE(6,1200) (LNODE(I),I=1,5)
1200  FORMAT(1H0,19HBOUNDARY LAST NODES/5I5)
C
C  ELEMENT DATA
      WRITE(6,1140)
1140  FORMAT(1H0,12HELEMENT DATA)
      WRITE(6,1150)
1150  FORMAT(1X,T10,3HNO.,T25,1HX,T35,1HY,T41,8HKNOWN BC,T55,3HBC1,
     &T65,3HBC2)
C
      DO 10 N=1,NOE
      WRITE(6,1160)N,X(N),Y(N)
1160  FORMAT(1X,I12,F10.3,F10.3)
      IF(KBC(N).EQ.0) WRITE(6,1170)
1170  FORMAT(1H+,T45,1HF)
      IF(KBC(N).NE.0) WRITE(6,1180)
1180  FORMAT(1H+,T45,1HU)
      I = 2*N - 1
      I1 = I + 1
      WRITE(6,1190)BC(I),BC(I1)
1190  FORMAT(1H+,T50,2F10.3)
10    CONTINUE
      RETURN
      END


       SUBROUTINE KMAKER(NOE,LAMDA,G,PI,X,Y,K,KBC,MK,LNODE)
C
C     MATRIX GENERATION ROUTINE
C     THIS ROUTINE DEVELOPS (K) FOR THE EQUATION (K)(P)=(BC).
C     WHERE (P) IS THE THE COLUMN MATRIX OF ARTIFICIAL BOUNDARY
C     STRESSES AND (BC) IS THE COLUMN MATRIX OF KNOWN
C     BOUNDARY CONDITIONS.
```

```
C
      DIMENSION X(1),Y(1),KBC(1),LNODE(1)
      REAL K(MK,1), LAMDA
      COMMON /SHOE/ FOR,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,F02,F12,
     &F22,F32,F42,FLG
      COMMON /CEES/ C0,C1,C2,C3,C4
C
C     SET BOUNDARY NUMBER
      MBOUND = 1
      DO 10 M=1,NOE
      IF (M.GT.LNODE(MBOUND)) MBOUND = MBOUND + 1
      CALL PREK(M,M1,I,I1,STHETM,CTHETM,DL,X,Y,LNODE,MBOUND)
      XQM = (X(M)+X(M1))/2.
      YQM = (Y(M)+Y(M1))/2.
      NBOUND = 1
      IF(KBC(M).NE.0) GO TO 20
C     BOUNDARY STRESSES KNOWN
      DO 30 N=1,NOE
      IF (N.GT.LNODE(NBOUND)) NBOUND = NBOUND + 1
      CALL PREK(N,N1,J,J1,STHETN,CTHETN,DL,X,Y,LNODE,NBOUND)
      IF(I.EQ.J) GO TO 40
C     FOR NONSINGULARITY
      CALL PREGAM(UP,DL,A,B,EC,EE,EG,XQM,YQM,X,Y,CTHETN,
     &STHETN,N)
      CALL GAMMAF(A,B,EC,UP,GAM1,GAM2,GAM3,GAM4,GAM5,GAM6,EE,CTHETN,
     &EG,STHETN)
40    CALL SXC(SXN,SXT,STHETN,CTHETN,GAM1,GAM2,LAMDA,G,I,J)
      CALL SYC(SYN,SYT,STHETN,CTHETN,GAM3,GAM4,LAMDA,G,I,J)
      CALL TC(TN,TT,STHETN,CTHETN,GAM5,GAM6,LAMDA,G,I,J)
      K(I,J) = STHETM*STHETM*SXN + CTHETM*CTHETM*SYN -
     &2.*STHETM*CTHETM*TN
      K(I,J1) = STHETM*STHETM*SXT + CTHETM*CTHETM*SYT -
     &2.*STHETM*CTHETM*TT
      K(I1,J) = -CTHETM*STHETM*SXN + CTHETM*STHETM*SYN +
     &(CTHETM*CTHETM - STHETM*STHETM)*TN
      K(I1,J1) = -CTHETM*STHETM*SXT + CTHETM*STHETM*SYT +
     &(CTHETM*CTHETM - STHETM*STHETM)*TT
30    CONTINUE
      GO TO 10
C
C
C     BOUNDARY DISPLACEMENTS KNOWN
20    DO 50 N=1,NOE
      IF (N.GT.LNODE(NBOUND)) NBOUND = NBOUND + 1
      CALL PREK(N,N1,J,J1,STHETN,CTHETN,DL,X,Y,LNODE,NBOUND)
      IF(I.EQ.J) GO TO 60
C     FOR NONSINGULARITY
      CALL PREGAM(UP,DL,A,B,EC,EE,EG,XQM,YQM,X,Y,CTHETN,
     &STHETN,N)
      CALL GAMMAU(A,B,EC,UP,GAM7,GAM8,GAM9,GAM10,EE,CTHETN,EG,STHETN)
60    CALL UC(UN,UT,STHETN,CTHETN,GAM7,GAM8,DL,C3,C4,I,J)
      CALL VC(VN,VT,STHETN,CTHETN,GAM9,GAM10,DL,C3,C4,I,J)
      K(I,J) = UN
```

```
      K(I,J1) = UT
      K(I1,J) = VN
      K(I1,J1) = VT
50    CONTINUE
10    CONTINUE
      RETURN
      END


      SUBROUTINE PREK(N,N1,J,J1,STHETN,CTHETN,DL,X,Y,LNODE,NBOUND)
C
C     DETERMINES ELEMENT ANGLE AND SEVERAL ARRAY INDICES
C
      DIMENSION X(1),Y(1),LNODE(1)
      N1 = N+1
      IF (N.EQ.LNODE(NBOUND).AND.NBOUND.EQ.1) N1=1
      IF (N.EQ.LNODE(NBOUND).AND.NBOUND.NE.1) N1=LNODE(NBOUND-1)+1
      DX = X(N1) - X(N)
      DY = Y(N1) - Y(N)
      DL = SQRT(DX*DX + DY*DY)
      STHETN = DY/DL
      CTHETN = DX/DL
      J = 2*N - 1
      J1 = J + 1
      RETURN
      END


      SUBROUTINE PREGAM(UP,DL,A,B,EC,EE,EG,XQM,YQM,X,Y,CTHETN,
     &STHETN,N)
C
C     DETERMINES SEVERAL VALUES IN PREPARATION FOR NONSINGULAR
C     INTEGRATIONS
C
      DIMENSION X(1),Y(1)
      UP = DL
      EE = XQM - X(N)
      EG = YQM - Y(N)
      A = 1.
      B = -2.*(CTHETN*EE + STHETN*EG)
      EC = EE*EE + EG*EG
      RETURN
      END
```

```
      SUBROUTINE GAMMAF(A,B,C,UP,GAM1,GAM2,GAM3,GAM4,GAM5,GAM6,
     &E,F,G,H)
C
C     CALCULATES NONSINGULAR INTEGRATIONS WHEN BOUNDARY STRESSES
C     ARE KNOWN.
C
      COMMON  /SHOE/ F0R,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      COMMON /CEES/ C0,C1,C2,C3,C4
      F01=S01(A,B,C,UP)
      F11=S11(A,B,C,UP)
      F02=S02(A,B,C,UP)
      F12=S12(A,B,C,UP)
      F22=S22(A,B,C,UP)
      F32=S32(A,B,C,UP)
      Q1=E*F01-F*F11
      Q2=G*F01-H*F11
      Q3=E*G*G*F02-(2.*E*G*H+G*G*F)*F12+(E*H*H+2.*F*G*H)*F22-F*H*H*F32
      Q4=E*E*G*F02-(2.*E*F*G+E*E*H)*F12+(F*F*G+2.*E*F*H)*F22-F*F*H*F32
      GAM1=-C0*C1*Q1+C2*Q3
      GAM2=C1*Q2-C2*Q4
      GAM3=-C0*C1*Q2+C2*Q4
      GAM4=C1*Q1-C2*Q3
      GAM5=C1*Q2+C2*Q4
      GAM6=C1*Q1+C2*Q3
      RETURN
      END


      SUBROUTINE GAMMAU(A,B,C,UP,GAM7,GAM8,GAM9,GAM10,
     &E,F,G,H)
C
C     CALCULATES NONSINGULAR INTEGRATIONS WHEN BOUNDARY DISPLACEMENTES
C     ARE KNOWN.
C
      COMMON  /SHOE/ F0R,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      COMMON /CEES/ C0,C1,C2,C3,C4
      F01=S01(A,B,C,UP)
      F11=S11(A,B,C,UP)
      F21=S21(A,B,C,UP)
      FLG=SLG(A,B,C,UP)
      Q5=G*G*F01-2.*G*H*F11+H*H*F21
      Q6=E*E*F01-2.*E*F*F11+F*F*F21
      Q7=E*G*F01-(E*H+F*G)*F11+F*H*F21
      Q8=FLG
      GAM7=-C3*Q8 -C4*Q5
      GAM8=C4*Q7
      GAM9=-C3*Q8-C4*Q6
      GAM10=C4*Q7
      RETURN
      END
```

```
      FUNCTION S01(A,B,C,UP)
      COMMON  /SHOE/ FOR,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN1(X)=2./R*ATAN((2.*A*X+B)/R)
      FUN2(X)=2./(2.*A*X+B)
      FUN3(X)=1./R*ALOG((2.*A*X+B-R)/(2.*A*X+B+R))
      BAC=B*B - 4.*A*C
      R=SQRT(ABS(BAC))
      EPSIL = 1.0E-10
      IF(BAC.GE.-EPSIL.AND.BAC.LE.EPSIL) GO TO 2
      IF(BAC) 1,2,3
    1 S01=FUN1(UP)-FUN1(0.)
      RETURN
    2 S01=FUN2(UP)-FUN2(0.)
      RETURN
    3 S01=FUN3(UP)-FUN3(0.)
      RETURN
      END


      FUNCTION S11(A,B,C,UP)
      COMMON  /SHOE/ FOR,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN(X)=1./(2.*A)*ALOG(A*X*X+B*X+C)
      S11=FUN(UP)-FUN(0.)-B/(2.*A)*F01
      RETURN
      END


      FUNCTION S21(A,B,C,UP)
      COMMON  /SHOE/ FOR,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN(X)=X/A-B/(2.*A)*ALOG(A*X*X+B*X+C)
      S21=FUN(UP)-FUN(0.)+(B*B-2.*A*C)/(2.*A*A)*F01
      RETURN
      END


      FUNCTION S02(A,B,C,UP)
      COMMON  /SHOE/ FOR,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN1(X)=(2.*A*X+B)/((-BAC)*(A*X*X+B*X+C))
      FUN2(X)=-1./A/A/3./(B/2./A+X)**3.
      BAC = B*B - 4.*A*C
      EPSIL = 1.0E-10
      IF(BAC.GE.-EPSIL.AND.BAC.LE.EPSIL) GO TO 2
      IF(BAC) 1,2,1
    1 S02=FUN1(UP)-FUN1(0.)+2.*A/(-BAC)*F01
      RETURN
    2 S02=FUN2(UP)-FUN2(0.)
      RETURN
      END
```

```
      FUNCTION S12(A,B,C,UP)
      COMMON  /SHOE/ F0R,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN1(X)=-(2.*C+B*X)/((-BAC)*(A*X*X+B*X+C))
      FUN2(X)=-(B/2.+3.*A*X)/(3.*A*X+1.5*B)
      BAC = B*B - 4.*A*C
      EPSIL = 1.0E-10
      IF(BAC.GE.-EPSIL.AND.BAC.LE.EPSIL) GO TO 2
      IF(BAC) 1,2,1
    1 S12=FUN1(UP)-FUN1(0.)-B/(-BAC)*F01
      RETURN
    2 S12=FUN2(UP)-FUN2(0.)
      RETURN
      END


      FUNCTION S22(A,B,C,UP)
      COMMON  /SHOE/ F0R,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN(X)=-X/(A*(A*X*X+B*X+C))
      S22=FUN(UP)-FUN(0.)+C/A*F02
      RETURN
      END


      FUNCTION S32(A,B,C,UP)
      COMMON  /SHOE/ F0R,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN(X)=-X*X/(A*(A*X*X+B*X+C))
      S32=FUN(UP)-FUN(0.)+F11/A+C*F12/A
      RETURN
      END


      FUNCTION SLG(A,B,C,UP)
      COMMON  /SHOE/ F0R,F1R,F2R,F3R,F4R,F01,F11,F21,F31,F41,
     1   F02,F12,F22,F32,F42,FLG
      FUN(X)=X*ALOG(SQRT(A*X*X+B*X+C))
      SLG=FUN(UP)-A*F21-B*.5*F11
      RETURN
      END
```

```
      SUBROUTINE SXC(SXN,SXT,STHETN,CTHETN,GAM1,GAM2,LAMDA,G,I,J)
C
C     SX CONTROL ROUTINE TO DECIDE BETWEEN SINGULAR AND NONSINGULAR
C     INFLUENCE COEFFICIENTS.
C
      REAL LAMDA
      IF (I.EQ.J) GO TO 10
C  NONSINGULAR
      CALL SXNS(SXN,SXT,STHETN,CTHETN,GAM1,GAM2)
      RETURN
C  SINGULAR
10    CALL SXS(SXN,SXT,STHETN,CTHETN,LAMDA,G)
      RETURN
      END


      SUBROUTINE SXNS(SXN,SXT,STHETN,CTHETN,GAM1,GAM2)
C
C     SIGMA X INFLUENCE COEFS FOR NONSINGULARITY CONDITIONS
C
      SXN = -STHETN*GAM1 + CTHETN*GAM2
      SXT = CTHETN*GAM1 + STHETN*GAM2
      RETURN
      END


      SUBROUTINE SXS(SXN,SXT,STHETN,CTHETN,LAMDA,G)
C
C     SIGMA X INFLUENCE COEFS FOR SINGULARITY CONDITIONS
C
      REAL LAMDA
      SXN = (LAMDA + 2.*G*STHETN*STHETN) /2. /(LAMDA + 2.*G)
      SXT = -STHETN*CTHETN
      RETURN
      END


      SUBROUTINE SYC(SYN,SYT,STHETN,CTHETN,GAM3,GAM4,LAMDA,G,I,J)
C
C     SY CONTROL ROUTINE
C
      REAL LAMDA
      IF (I.EQ.J) GO TO 10
C  NONSINGULAR
      CALL SYNS(SYN,SYT,STHETN,CTHETN,GAM3,GAM4)
      RETURN
C  SINGULAR
10    CALL SYS(SYN,SYT,STHETN,CTHETN,LAMDA,G)
      RETURN
      END
```

```
      SUBROUTINE SYNS(SYN,SYT,STHETN,CTHETN,GAM3,GAM4)
C
C     SIGMA Y INFLUENCE COEFS FOR NONSINGULARITY CONDITIONS
C
      SYN = CTHETN*GAM3 - STHETN*GAM4
      SYT = STHETN*GAM3 + CTHETN*GAM4
      RETURN
      END


      SUBROUTINE SYS(SYN,SYT,STHETN,CTHETN,LAMDA,G)
C
C     SIGMA Y INFLUENCE COEFS FOR SINGULARITY CONDITIONS
C
      REAL LAMDA
      SYN = (LAMDA + 2.*G*CTHETN*CTHETN) /2. /(LAMDA + 2.*G)
      SYT = STHETN*CTHETN
      RETURN
      END


      SUBROUTINE TC(TN,TT,STHETN,CTHETN,GAM5,GAM6,LAMDA,G,I,J)
C
C     TAU CONTROL ROUTINE
C
      REAL LAMDA
      IF (I.EQ.J) GO TO 10
C  NONSINGULAR
      CALL TNS(TN,TT,STHETN,CTHETN,GAM5,GAM6)
      RETURN
C  SINGULAR
10    CALL TS(TN,TT,STHETN,CTHETN,LAMDA,G)
      RETURN
      END


      SUBROUTINE TNS(TN,TT,STHETN,CTHETN,GAM5,GAM6)
C
C     TAU INFLUENCE COEFS FOR NONSINGULARITY CONDITIONS
C
      TN = STHETN*GAM5 - CTHETN*GAM6
      TT = -CTHETN*GAM5 - STHETN*GAM6
      RETURN
      END
```

```
      SUBROUTINE TS(TN,TT,STHETN,CTHETN,LAMDA,G)
C
C     TAU INFLUENCE COEFS FOR SINGULARITY CONDITIONS
C
      REAL LAMDA
      TN = -G / (LAMDA + 2.*G)*STHETN*CTHETN
      TT = -.5 * (STHETN*STHETN - CTHETN*CTHETN)
      RETURN
      END


      SUBROUTINE UC(UN,UT,STHETN,CTHETN,GAM7,GAM8,DL,C3,C4,I,J)
C
C     U CONTROL ROUTINE
C
      IF (I.EQ.J) GO TO 10
C  NONSINGULAR
      CALL UNS(UN,UT,STHETN,CTHETN,GAM7,GAM8)
      RETURN
C  SINGULAR
10    CALL US(UN,UT,STHETN,CTHETN,DL,C3,C4)
      RETURN
      END


      SUBROUTINE UNS(UN,UT,STHETN,CTHETN,GAM7,GAM8)
C
C     U INFLUENCE COEFS FOR NONSINGULARITY CONDITIONS
C
      UN = -STHETN*GAM7 + CTHETN*GAM8
      UT = CTHETN*GAM7 + STHETN*GAM8
      RETURN
      END


      SUBROUTINE US(UN,UT,STHETN,CTHETN,DL,C3,C4)
C
C     U INFLUENCE COEFS FOR SINGULARITY CONDITIONS
C
      UN = DL * STHETN * (C3*(ALOG(DL/2.) -1.) + C4)
      UT = -C3 * (ALOG(DL/2.) -1.) * DL * CTHETN
      RETURN
      END
```

```
      SUBROUTINE VC(VN,VT,STHETN,CTHETN,GAM9,GAM10,DL,C3,C4,I,J)
C
C     V CONTROL ROUTINE
C
      IF (I.EQ.J) GO TO 10
C     NONSINGULAR
      CALL VNS(VN,VT,STHETN,CTHETN,GAM9,GAM10)
      RETURN
C     SINGULAR
10    CALL VS(VN,VT,STHETN,CTHETN,DL,C3,C4)
      RETURN
      END



      SUBROUTINE VNS(VN,VT,STHETN,CTHETN,GAM9,GAM10)
C
C     V INFLUENCE COEFS FOR NONSINGULARITY CONDITIONS
C
      VN = CTHETN*GAM9 - STHETN*GAM10
      VT = STHETN*GAM9 + CTHETN*GAM10
      RETURN
      END


      SUBROUTINE VS(VN,VT,STHETN,CTHETN,DL,C3,C4)
C
C     V INFLUENCE COEFS FOR SINGULARITY CONDITIONS
C
      VN = -DL * CTHETN * (C3*(ALOG(DL/2.) -1.) + C4)
      VT = -C3 * (ALOG(DL/2.) -1.) * DL * STHETN
      RETURN
      END
```

```
      SUBROUTINE AXEQB(A,X,B,N,M)
      DIMENSION A(M,1), X(1), B(1), IPIVOT(304), D(304), W(304,304)
C
C THIS SUBROUTINE SOLVES THE LINEAR SYSTEM AX = B
C IN THIS APPLICATION THE ARTIFICIAL BOUNDARY STRESSES ARE SOLVED FOR
C
      CALL FACTOR (A,A,IPIVOT,D,N,IFLAG,M)
      IF (IFLAG .EQ. 1) GO TO 10
      WRITE (6,1000)
      STOP
   10 CONTINUE
      DO 100 I = 1,N
  100 X(I) = 0.0
      CALL SUBST (A,B,X,IPIVOT,N,M)
      RETURN
 1000 FORMAT(19H1MATRIX IS SINGULAR)
      END


      SUBROUTINE FACTOR(A,W,IPIVOT,D,N,IFLAG,M)
      DIMENSION A(M,1),IPIVOT(1),D(1)
      DIMENSION W(M,1)
      IFLAG = 1
C     INITIALIZE W, IPIVOT, D
      DO 10 I=1,N
      IPIVOT(I) = I
      ROWMAX = 0.
      DO 9 J = 1,N
      W(I,J) = A(I,J)
    9 ROWMAX = AMAX1(ROWMAX,ABS(W(I,J)))
      IF (ROWMAX .EQ. 0.)  GO TO 999
   10 D(I) = ROWMAX
C     GAUSS ELIMINATION WITH SCALED PARTIAL PIVOTING.
      NM1 = N-1
      IF (NM1 .EQ. 0) RETURN
      DO 20 K = 1,NM1
      J = K
      KP1 = K + 1
      IP = IPIVOT(K)
      COLMAX = ABS(W(IP,K))/D(IP)
      DO 11 I = KP1,N
      IP = IPIVOT(I)
      AWIKOV = ABS(W(IP,K))/D(IP)
      IF (AWIKOV .LE. COLMAX) GO TO 11
      COLMAX = AWIKOV
      J = I
   11 CONTINUE
      IF (COLMAX .EQ. 0.) GO TO 999
C
      IPK = IPIVOT(J)
      IPIVOT(J) = IPIVOT(K)
      IPIVOT(K) = IPK
```

```
      DO 20 I = KP1,N
      IP = IPIVOT(I)
      W(IP,K) = W(IP,K)/W(IPK,K)
      RATIO = -W(IP,K)
      DO 20 J = KP1,N
   20 W(IP,J) = RATIO*W(IPK,J) + W(IP,J)
      IF (W(IP,N) .EQ. 0.) GO TO 999
      RETURN
  999 IFLAG = 2
      RETURN
      END



      SUBROUTINE SUBST(W,B,X,IPIVOT,N,M)
      DIMENSION W(M,1),B(1),X(1),IPIVOT(1)
      IF (N.GT.1) GO TO 10
      X(1) = B(1)/W(1,1)
      RETURN
   10 IP = IPIVOT(1)
      X(1) = B(IP)
      DO 15 K = 2,N
      IP = IPIVOT(K)
      KM1 = K-1
      SUM = 0.
      DO 14 J= 1,KM1
   14 SUM = W(IP,J)*X(J) + SUM
   15 X(K) = B(IP) - SUM
C
      X(N) = X(N)/W(IP,N)
      K = N
      DO 20 NP1MK = 2,N
      KP1 = K
      K = K- 1
      IP = IPIVOT(K)
      SUM = 0.
      DO 19 J = KP1,N
   19 SUM = W(IP,J)*X(J) + SUM
   20 X(K) = (X(K) - SUM)/W(IP,K)
      RETURN
      END
```

```
      SUBROUTINE RESPON (NOE,LAMDA,G,PI,X,Y,P,LNODE)
C
C  THIS ROUTINE CALCULATES THE RESPONSE AT EACH USER DEFINED
C  POINT OF INTEREST.  THE RESULTS ARE SENT TO DEVICE 6, WHICH
C  IS INITIALLY OUTPUT.
C  THE USER HAS THE OPTION OF DEFINING THE RESPONSE POINT
C  AS WITHIN OR ON THE BOUNDARY.  FOR THAT GIVEN POINT HE ALSO
C  HAS THE OPTION OF WHICH RESPONSES TO CALCULATE.
C
      COMMON/CEES/C0,C1,C2,C3,C4
      DIMENSION X(1),Y(1),P(1),LNODE(1)
      INTEGER CSX,CSY,CT,CU,CV
      REAL LAMDA
C
C  PRINT OUTPUT HEADER
      WRITE(6,3000)
3000  FORMAT(1H1,T4,8HRESPONSE/1X,T4,8H--------)
      WRITE(6,3010)
3010  FORMAT(1H0,T4,7HELEMENT,T16,1HX,T26,1HY,T36,2HSX,T46,2HSY,T56,1HT
     &,T66,1HU,T76,1HV/)
C
C  RESPONSE POINT CALCULATION LOOP
C
10    READ(5,3015)JB,I,XR,YR,CSX,CSY,CT,CU,CV
3015  FORMAT(2I5,2F10.0,5I5)
      IF(JB.EQ.77777) RETURN
      IF(JB.EQ.0) GO TO 20
C  RESPONSE AT A BOUNDARY ELEMENT
      WRITE(6,3020)I
3020  FORMAT(1X,T5,I5)
      I1 = I + 1
C
C  DETERMINE LAST NODE NUMBER
      DO 15 N=1,5
      IF(LNODE(N).EQ.0) GO TO 16
      IF(I.EQ.LNODE(N).AND.N.EQ.1) I1=1
15    IF(I.EQ.LNODE(N).AND.N.NE.1) I1=LNODE(N-1)+1
C
16    XR = (X(I) + X(I1))/2.
      YR = (Y(I) + Y(I1))/2.
      GO TO 30
C  RESPONSE NOT AT A BOUNDARY ELEMENT
20    WRITE(6,3030)XR,YR
3030  FORMAT(1X,T11,2F10.3)
      I = 0

C  INITIALIZE THE RESPONSES
30    SXR = 0.
      SYR = 0.
      TR = 0.
      UR = 0.
      VR = 0.
```

```
C  INITIALIZE BOUNDARY NUMBER
      NBOUND=1
C  FOR EACH LOAD (ASSUMES BOUNDARY LOADS ONLY)
C
      DO 40 N=1,NOE
      IF(N.GT.LNODE(NBOUND)) NBOUND=NBOUND+1
      CALL PREK(N,N1,J,J1,STHETN,CTHETN,DL,X,Y,LNODE,NBOUND)
      CALL PREGAM(UP,DL,A,B,EC,EE,EG,XR,YR,X,Y,CTHETN,STHETN,N)
      IF(CSX.NE.0.AND.CSY.NE.0.AND.CT.NE.0) GO TO 50
C  THERE ARE STRESS CALCULATIONS
      CALL GAMMAF(A,B,EC,UP,GAM1,GAM2,GAM3,GAM4,GAM5,GAM6,EE,CTHETN,
     &EG,STHETN)
      IF(CSX.NE.0) GO TO 60
      CALL SXC(SXN,SXT,STHETN,CTHETN,GAM1,GAM2,LAMDA,G,I,N)
      SXR = SXR + SXN*P(J) + SXT*P(J1)
60    IF(CSY.NE.0) GO TO 70
      CALL SYC(SYN,SYT,STHETN,CTHETN,GAM3,GAM4,LAMDA,G,I,N)
      SYR = SYR + SYN*P(J) + SYT*P(J1)
70    IF(CT.NE.0) GO TO 50
      CALL TC(TN,TT,STHETN,CTHETN,GAM5,GAM6,LAMDA,G,I,N)
      TR = TR + TN*P(J) + TT*P(J1)
50    IF(CU.NE.0.AND.CV.NE.0) GO TO 40
C  THERE ARE DISPLACEMENT CALCULATIONS
      CALL GAMMAU(A,B,EC,UP,GAM7,GAM8,GAM9,GAM10,EE,CTHETN,EG,STHETN)
      IF(CU.NE.0) GO TO 80
      CALL UC(UN,UT,STHETN,CTHETN,GAM7,GAM8,LAMDA,G,I,N)
      UR = UR + UN*P(J) + UT*P(J1)
80    IF(CV.NE.0) GO TO 40
      CALL VC(VN,VT,STHETN,CTHETN,GAM9,GAM10,LAMDA,G,I,N)
      VR = VR + VN*P(J) + VT*P(J1)
40    CONTINUE
C  INDIVIDUAL RESPOPNSE POINT PRINT OUT
      IF(CSX.NE.0) GO TO 90
      WRITE(6,3400) SXR
3400  FORMAT(1H+,T31,E10.3)
90    IF(CSY.NE.0) GO TO 100
      WRITE(6,3500) SYR
3500  FORMAT(1H+,T41,E10.3)
100   IF(CT.NE.0) GO TO 110
      WRITE(6,3600) TR
3600  FORMAT(1H+,T51,E10.3)
110   IF(CU.NE.0) GO TO 120
      WRITE(6,3700) UR
3700  FORMAT(1H+,T61,E10.3)
120   IF(CV.NE.0) GO TO 130
      WRITE(6,3800) VR
3800  FORMAT(1H+,T71,E10.3)
C  MAKE OUTPUT FILE (TAPE7) FOR PLOTTING.
130   IF(JB.NE.0) GO TO 10
      WRITE(7,4000) XR,YR,SXR,SYR,TR,UR,VR
4000  FORMAT(2F10.4,5E15.6)
      GO TO 10
      END
```

DISTRIBUTION LIST

ARMY Fal Engr. Letterkenny Army Depot. Chambersburg. PA
AF HQ LEEH (J Stanton) Washington. DC
AFB CESCH. Wright-Patterson; HQ MAC/DEEE. Scott. Il; SAMSO/MNND. Norton AFB CA; Stinfo Library. Offutt NE
ARCTICSUBLAB Code 54. San Diego. CA
ARMY ARRADCOM. Dover. NJ; BMDSC-RE (H. McClellan) Huntsville AL; DAEN-MPE-D Washington DC; ERADCOM Tech Supp Dir. (DELSD-L) Ft. Monmouth. NJ; Tech. Ref. Div.. Fort Huachuca. AZ
ARMY - CERL Library, Champaign IL
ARMY COASTAL ENGR RSCH CEN Fort Belvoir VA; R. Jachowski. Fort Belvoir VA
ARMY COE Philadelphia Dist. (LIBRARY) Philadelphia. PA
ARMY CORPS OF ENGINEERS MRD-Eng. Div.. Omaha NE; Seattle Dist. Library. Seattle WA
ARMY CRREL A. Kovacs. Hanover NH; Library. Hanover NH
ARMY DARCOM Code DRCMM-CS Alexandria VA
ARMY ENG DIV ED-CS (S.Bolin) Huntsville. AL; HNDED-CS. Huntsville AL; HNDED-FD. Huntsville. AL
ARMY ENG WATERWAYS EXP STA Library. Vicksburg MS
ARMY ENVIRON. HYGIENE AGCY HSE-EW Water Qual Eng Div Aberdeen Prov Grnd MD
ARMY MATERIALS & MECHANICS RESEARCH CENTER Dr. Lenoe. Watertown MA
ARMY MISSILE R&D CMD SCI Info Cen (DOC) Redstone Arsenal, AL
ARMY-DEPOT SYS COMMAND DRSDS-AI Chambersburg. PA
ASST SECRETARY OF THE NAVY Spec. Assist Submarines. Washington DC
BUREAU OF RECLAMATION Code 1512 (C. Selander) Denver CO
CNM MAT-0718. Washington. DC; NMAT - 044. Washington DC
CNO Code NOP-964. Washington DC; Code OPNAV 09B24 (H); Code OPNAV 22. Wash DC; Code OPNAV 23. Wash DC; OP-23 (Capt J.H. Howland) Washinton. DC; OP987J. Washington. DC
COMFLEACT. OKINAWA PWD - Engr Div, Sasebo. Japan; PWO. Kadena. Okinawa; PWO. Sasebo. Japan
COMOCEANSYSLANT PW-FAC MGMNT Off Norfolk. VA
COMRNCF Nicholson. Tampa. FL; Nicholson. Tampa. FL
DEFENSE CIVIL PREPAREDNESS AGENCY Washington. DC
DEFENSE INTELLIGENCE AGENCY DB-4C1 Washington DC
DEFFUELSUPPCEN DFSC-OWE (Term Engrng) Alexandria. VA; DFSC-OWE. Alexandria VA
DOD Explosives Safety Board (Library), Washington DC
DOE Div Ocean Energy Sys Cons/Solar Energy Wash DC
DTIC Defense Technical Info Ctr/Alexandria. VA
DTNSRDC Anna Lab. Code 4121 (R A Rivers) Annapolis. MD
DTNSRDC Code 1706. Bethesda MD; Code 172 (M. Krenzke). Bethesda MD
DTNSRDC Code 284 (A. Rufolo). Annapolis MD
GSA Assist Comm Des & Cnst (FAIA) D R Dibner Washington. DC
LIBRARY OF CONGRESS Washington. DC (Sciences & Tech Div)
MARINE CORPS BASE M & R Division. Camp Lejeune NC; PWD - Maint. Control Div. Camp Butler. Kawasaki. Japan; PWO Camp Lejeune NC; PWO. Camp Pendleton CA; PWO. Camp S. D. Butler. Kawasaki Japan
MARINE CORPS HQS Code LFF-2. Washington DC
MCAS Facil. Engr. Div. Cherry Point NC; CO. Kaneohe Bay HI; Code S4. Quantico VA; Facs Maint Dept - Operations Div. Cherry Point; PWD. Dir. Maint. Control Div.. Iwakuni Japan; PWO. Yuma AZ; SCE. Futema Japan
MCLB B520. Barstow CA
MCRD SCE. San Diego CA
NAF PWD - Engr Div. Atsugi. Japan; PWO. Atsugi Japan
NARF Code 640. Pensacola FL; Equipment Engineering Division (Code 61000). Pensacola. FL
NAS CO. Guantanamo Bay Cuba; Code 114. Alameda CA; Code 183 (Fac. Plan BR MGR); Code 18700. Brunswick ME; Dir of Engrng. PWD. Corpus Christi. TX; Dir. Util. Div.. Bermuda; PW (J. Maguire). Corpus Christi TX; PWD - Engr Div Dir. Millington. TN; PWD - Engr Div. Oak Harbor. WA; PWD Maint. Div.. New Orleans. Belle Chasse LA; PWD. Code 1821H (Pfankuch) Miramar. SD CA; PWO Belle Chasse. LA; PWO Chase Field Beeville. TX; PWO Key West FL; PWO Lakehurst. NJ; PWO Sigonella Sicily; PWO Whiting Fld. Milton FL; PWO. Dall.s TX; PWO. Glenview IL; PWO. Kingsville TX; PWO. Miramar. San Diego CA; SCE Norfolk. VA; SCE. Barbers Point HI; Security Offr. Alameda CA
NASDC-WDC T. Fry. Manassas VA
NATL RESEARCH COUNCIL Naval Studies Board. Washington DC
NAVACT PWO. London UK
NAVAEROSPREGMEDCEN SCE. Pensacola FL
NAVAIRDEVCEN Code 813. Warminster PA; PWD. Engr Div Mgr. Warminster. PA
NAVCHAPGRU Engineering Officer. Code 60 Williamsburg. VA

NAVCOASTSYSCEN CO, Panama City FL; Code 715 (J Quirk) Panama City, FL; Code 715 (J. Mittleman) Panama City, FL; Library Panama City, FL; PWO Panama City, FL
NAVCOMMAREAMSTRSTA SCE Unit 1 Naples Italy; SCE, Wahiawa HI
NAVCOMMSTA Code 401 Nea Makri, Greece; PWO, Exmouth, Australia
NAVCONSTRACEN Curriculum/Instr. Stds Offr, Gulfport MS
NAVEDTRAPRODEVCEN Technical Library, Pensacola, FL
NAVEDUTRACEN Engr Dept (Code 42) Newport, RI
NAVELEXSYSCOM Code PME 124-61, Washington, DC; PME 124-612, Wash DC
NAVEODTECHCEN Code 605, Indian Head MD
NAVFAC PWO, Brawdy Wales UK; PWO, Centerville Bch, Ferndale CA; PWO, Point Sur, Big Sur CA
NAVFACENGCOM Code 03 Alexandria, VA; Code 03T (Essoglou) Alexandria, VA; Code 043 Alexandria, VA; Code 044 Alexandria, VA; Code 0453 (D. Potter) Alexandria, VA; Code 0453C, Alexandria, VA; Code 0454B Alexandria, Va; Code 04A1 Alexandria, VA; Code 1113, Alexandria, VA; **Code 09M54, Tech Lib, Alexandria, VA**
NAVFACENGCOM - CHES DIV. Code 101 Wash, DC; Code 407 (D Scheesele) Washington, DC; Code FPO-1C Washington DC; FPO-1 Washington, DC; FPO-1EA5 Washington DC; Library, Washington, D.C.
NAVFACENGCOM - LANT DIV. Eur. BR Deputy Dir, Naples Italy; Library, Norfolk, VA; RDT&ELO 102A, Norfolk, VA
NAVFACENGCOM - NORTH DIV. (Boretsky) Philadelphia, PA; CO; Code 04 Philadelphia, PA; Code 1028, RDT&ELO, Philadelphia PA; Library, Philadelphia, PA; ROICC, Contracts, Crane IN
NAVFACENGCOM - PAC DIV. (Kyi) Code 101, Pearl Harbor, HI; CODE 09P PEARL HARBOR HI; Code 402, RDT&E, Pearl Harbor HI; Commander, Pearl Harbor, HI; Library, Pearl Harbor, HI
NAVFACENGCOM - SOUTH DIV. Code 90, RDT&ELO, Charleston SC; Library, Charleston, SC
NAVFACENGCOM - WEST DIV. Code 04B San Bruno, CA; Library, San Bruno, CA; O9P/20 San Bruno, CA; RDT&ELO Code 2011 San Bruno, CA
NAVFACENGCOM CONTRACTS AROICC, Quantico, VA; Colts Neck, NJ; Contracts, AROICC, Lemoore CA; Eng Div dir, Southwest Pac, Manila, PI; OICC, Southwest Pac, Manila, PI; OICC-ROICC, NAS Oceana, Virginia Beach, VA; OICC/ROICC, Balboa Panama Canal; OICC/ROICC, Norfolk, VA; ROICC AF Guam; ROICC Code 495 Portsmouth VA; ROICC Key West FL; ROICC, Keflavik, Iceland; ROICC, NAS, Corpus Christi, TX; ROICC, Pacific, San Bruno CA; ROICC, Point Mugu, CA; ROICC, Yap
NAVMAG SCE, Subic Bay, R.P.
NAVOCEANO Code 3432 (J. DePalma), Bay St. Louis MS; Library Bay St. Louis, MS
NAVOCEANSYSCEN Code 09 (Talkington), San Diego, CA; Code 4473 Bayside Library, San Diego, CA; Code 4473B (Tech Lib) San Diego, CA; Code 5204 (J. Stachiw), San Diego, GA; Code 5214 (H. Wheeler), San Diego CA; Code 5221 (R.Jones) San Diego Ca; Hawaii Lab (R Yumori) Kailua, HI; Hi Lab Tech Lib Kailua HI
NAVORDMISTESTFAC PWD - Engr Dir, White Sands, NM
NAVORDSTA PWO, Louisville KY
NAVPGSCOL C. Morers Monterey CA; E. Thornton, Monterey CA
NAVPHIBASE CO, ACB 2 Norfolk, VA; Code S3T, Norfolk VA; Harbor Clearance Unit Two, Little Creek, VA; SCE Coronado, SD,CA
NAVREGMEDCEN Code 3041, Memphis, Millington TN; PWD - Engr Div, Camp Lejeune, NC; PWO, Camp Lejeune, NC; SCE; SCE San Diego, CA; SCE, Camp Pendleton CA; SCE, Guam
NAVSCOLCECOFF C35 Port Hueneme, CA; CO, Code C44A Port Hueneme, CA
NAVSCSOL PWO, Athens GA
NAVSEASYSCOM Code SEA OOC Washington, DC
NAVSECGRUACT PWO, Adak AK; PWO, Torri Sta, Okinawa
NAVSHIPREPFAC Library, Guam; SCE Subic Bay
NAVSHIPYD Bremerton, WA (Carr Inlet Acoustic Range); Code 202.4, Long Beach CA; Code 202.5 (Library) Puget Sound, Bremerton WA; Code 400, Puget Sound; Code 440 Portsmouth NH; Code 440, Norfolk; Code 440, Puget Sound, Bremerton WA; L.D. Vivian; Library, Portsmouth NH; PWD (Code 420) Dir Portsmouth, VA; PWD (Code 460) Portsmouth, VA; PWO, Bremerton, WA; PWO, Mare Is.; Tech Library, Vallejo, CA
NAVSTA CO Roosevelt Roads P.R. Puerto Rico; Dir Engr Div, PWD, Mayport FL; Engr. Dir., Rota Spain; Long Beach, CA; Maint. Div. Dir/Code 531, Rodman Panama Canal; PWD - Engr Dept, Adak, AK; PWD - Engr Div, Midway Is.; PWO, Keflavik Iceland; SCE, Guam; SCE, San Diego CA
NAVSUPPACT Engr. Div. (F. Mollica), Naples Italy; PWO Naples Italy
NAVTECHTRACEN SCE, Pensacola FL
NAVWPNCEN Code 2636 China Lake; Code 3803 China Lake, CA; PWO (Code 266) China Lake, CA; ROICC (Code 702), China Lake CA
NAVWPNSTA Code 092, Concord CA
NAVWPNSTA PW Office Yorktown, VA
NAVWPNSTA PWD - Maint. Control Div., Concord, CA; PWD - Supr Gen Engr, Seal Beach, CA; PWO, Charleston, SC; PWO, Seal Beach CA
NAVWPNSUPPCEN Code 09 Crane IN

NCBC Code 10 Davisville, RI; Code 15, Port Hueneme CA; Code 155, Port Hueneme CA; Code 156, Port Hueneme, CA; Code 430 (PW Engrng) Gulfport, MS; PWO (Code 80) Port Hueneme, CA; PWO, Gulfport, MS
NCR 20, Commander
NMCB FIVE, Operations Dept; Forty, CO; THREE, Operations Off.
NOAA (Dr. T. Mc Guinness) Rockville, MD; Library Rockville, MD
NORDA Code 410 Bay St. Louis, MS; Code 440 (Ocean Rsch Off) Bay St. Louis MS
NRL Code 5800 Washington, DC; Code 5843 (F. Rosenthal) Washington, DC; Code 8441 (R.A. Skop), Washington DC
NROTC J.W. Stephenson, UC, Berkeley, CA
NSC Code 54.1 Norfolk, VA
NSD SCE, Subic Bay, R.P.
NUCLEAR REGULATORY COMMISSION T.C. Johnson, Washington, DC
NUSC Code 131 New London, CT; Code 332, B-80 (J. Wilcox) New London, CT; Code EA123 (R.S. Munn), New London CT; Code TA131 (G. De la Cruz), New London CT
ONR Central Regional Office, Boston, MA; Code 481, Bay St. Louis, MS; Code 485 (Silva) Arlington, VA; Code 700F Arlington VA
PACMISRANFAC HI Area Bkg Sands, PWO Kekaha, Kauai, HI
PHIBCB 1 P&E, San Diego, CA
PMTC Code 4253-3, Point Mugu, CA; EOD Mobile Unit, Point Mugu, CA
PWC ACE Office Norfolk, VA; CO Norfolk, VA; CO, (Code 10), Oakland, CA; CO, Great Lakes IL; CO, Pearl Harbor HI; Code 10, Great Lakes, IL; Code 105 Oakland, CA; Code 120, Oakland CA; Library, Code 120C, San Diego, CA; Code 128, Guam; Code 154 (Library), Great Lakes, IL; Code 200, Great Lakes IL; Code 400, Great Lakes, IL; Code 400, Oakland, CA; Code 400, Pearl Harbor, HI; Code 400, San Diego, CA; Code 420, Great Lakes, IL; Code 420, Oakland, CA; Code 424, Norfolk, VA; Library, Guam; Library, Norfolk, VA; Library, Oakland, CA; Library, Pearl Harbor, HI; Library, Pensacola, FL; Library, Subic Bay, R.P.; Library, Yokosuka JA
TVA Solar Group, Arnold, Knoxville, TN
UCT ONE OIC, Norfolk, VA
UCT TWO OIC, Port Hueneme CA
US DEPT OF INTERIOR Bur of Land Mgmnt Code 583, Washington DC
US GEOLOGICAL SURVEY Off. Marine Geology, Piteleki, Reston VA
USCG (G-MP-3/USP/82) Washington Dc; (Smith), Washington, DC; G-EOE-4 (T Dowd), Washington, DC
USCG R&D CENTER CO Groton, CT; D. Motherway, Groton CT
USDA Forest Service Reg 3 (R. Brown) Albuquerque, NM
USNA Ch. Mech. Engr. Dept Annapolis MD; ENGRNG Div, PWD, Annapolis MD; PWO Annapolis MD; USNA/Sys Eng Dept, Annapolis, MD
USS FULTON WPNS Rep. Offr (W-3) New York, NY
NUSC Library, Newport, RI