ANALYSIS OF RECTANGULAR CONCRETE TANKS CONSIDERING INTERACTION OF PLATE ELEMENTS

by

Douglas G. Fitzpatrick

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Civil Engineering

APPROVED:

Dr. J. H. Moore, Chairman

Dr. R. M. Barker

Prof. D. A. Garst

December 1982

Blacksburg, Virginia

ACKNOWLEDGEMENT

The author would like to thank his major advisor, Dr. J. Herbert Moore, Professor, Civil Engineering for his guidance and assistance during the course of his studies.

Thanks is also extended to Dr. Richard M. Barker and Prof. Don A. Garst for their support and teaching during the author's studies at Virginia Tech.

The author wishes to thank the Department of Civil Engineering for their funding of this study and for their financial support during his first year of study.

Finally, the author is grateful to his mother and father for their support and encouragement during his collegiate education and special thanks is given to his mother for helping with the typing of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION AND SCOPE

Rectangular tanks have generally been designed as an assemblage of plates with appropriate boundary conditions along the edges. The Portland Cement Association (PCA) published a bulletin[1] in 1969 which contained moment coefficients for plates with triangular and uniform pressure distributions, given boundary conditions and various ratios of length-to-height. The bounday conditions for these plates were either clamped, simply supported or free.

A clamped edge is defined as one that is moment resistant and no rotation or displacement of the joint or edge is possible. A simply supported condition is one that does not permit displacement; however, the edge is non-moment resistant. A free condition permits displacement and is non-moment resistant. A fixed edge is one that is moment resistant but rotation of the joint is possible.

These three conditions do not accurately represent the joints in a rectangular tank as most often built. Most concrete tanks are built with monolithic wall-to-wall and wall-to-footing joints. Assuming monolithic construction, the angle between the tangents to the original surfaces of a wall-to-wall or wall-to-floor joint remain fixed, but the joint is free to rotate. Consequently, the clamped condition is only an accurate boundary condition for the wall-to-wall joints in a square tank under symmetric loading. It is also very difficult to construct a truely unrestrained and non-moment resistant joint that is resistant to leakage. Therefore, the fixed boundary condition as herein defined best represents the true field condition in tanks.

In practice, a moment distribution type of balancing is sometimes used to provide for the continuity and joint rotations possible at an edge. The unbalanced moments at a joint, which develop from unequal lengths of walls and footings or different loading conditions on adjoining plates, are redistributed based on the relative stiffnesses of the adjoining plates.

Although this procedure is easy to carry out, a problem arises in determining the stiffness of a given section of the walls or floor when balancing moments in a strip through the footing and walls. A free condition at the top edge of the wall in a strip would imply that there is no resistance to rotation and this section would have zero stiffness. The strip, however, is removed from the continuity of the plate which provides resistance to rotation. Some designers use the "fixed-end" stiffness of the floor and two-thirds the "fixed-end" stiffness (4EI/L) of the wall to determine the relative stiffnesses at such a joint. A similar situation occurs when balancing moments in a horizontal strip through the four walls. The fact that the joint at the far end of the wall rotates in rectangular tanks and that the cross-section is removed from the continuum of the plate does not permit an accurate assessment of the stiffness of the walls or floor at a joint.

The purpose and scope of this paper is to develop a program that determines the bending moments at a number of locations in the walls and floor, treats these as plates, and takes into account the rotations of the joints. The finite element method of analysis is chosen because of the flexibility and ease with which it can handle arbitrary loadings and boundary conditions. The materials used are assumed to be elastic, homogeneous and isotropic. To enable the practitioner to determine some extreme moment values for design of rectangular concrete tanks, a moment distribution type of process is also developed from the finite element results.

This paper is limited to a study of bending moments in tanks with four walls and a footing, built integrally.

II. LITERATURE REVIEW

The analysis of rectangular concrete tanks with the floor built integrally with the walls has not been fully addressed in any publications. There are no tables complete with moment values for variable sizes of tanks that consider the partial restraint and continuity of the plate intersections, nor has there been an appropriate approximate method developed to determine moment values along the entire edge of interconnected plates.

PCA Bulletin ST-63 [1] contains moment values for plates with edges that are either clamped, simply supported or free (hereafter referred to as conventional boundary conditions). It also contains two tables that account for wall-to-wall interaction in rectangular tanks, but no wall-to-footing moment transfer. The bottom edges of the walls of these tanks are assumed to be simply supported. The author was unable to determine from PCA the basis of or method used to prepare these tables.

The finite element method, which is used in this paper to solve the interaction problem, has been used successfully to solve single plate problems with conventional boundary conditions. Jofriet[2] developed several tables of moment coefficients when he determined the influence of nonuniform wall thickness on vertical bending moments and on horizontal edge moments in walls of length-to-height ratios greater than three. His solutions, however, only included conventional boundary conditions.

Davies and Cheung[3] used the finite element method to determine coefficients for moment values in tanks but assumed that the wall-to-wall joints were clamped, the top edges were either free or simple supported and the bottom edges were simply supported or clamped. In an earlier article,[4] Cheung and Davies analyzed a rectangular tank with a specific ratio of dimensions and assumed (a) the bottom edges of the walls were fully clamped, and (b) the tank was supported on dwarf walls around the perimeter. The wall-to-wall and wall-to-floor joints were monolithic.

Davies did provide for the rotation of the wall-to-wall joint but only for a few very specific cases and generally only at one location, the center of the bottom edge of the wall. In one of his first articles[5] Davies described a moment distribution process for long rectangular tanks. The stiffnesses of the floor and walls in a cross section were equal to the flexural rigidity divided by the length of the element. The joints at the far end of an element were assumed to be clamped, therefore his distribution coefficients did not reflect the ability of the joint to rotate. The majority of his paper was devoted to developing easy methods for determining the fixed-end moments in the floor for a foundation of elastic material,[6] granular soil and cohesive soil. He used simplified limiting reaction pressures for the soils. This procedure was only used at one location in the wall and no collection of moment values for the whole system was given. If the tank was open at the top, Davies determined his bending moments directly from statics, that is, the wall acted like a cantilever, which does not reflect the continuity of the wall.

In another paper,[7] Davies used a classical approach to take into account the rotation of the plate intersections. He assumed the tank was square so that the vertical edges could be clamped and the bottom edge of the walls were elastically restained. He assumed a parabolic distribution of displacement in the plate along the bottom edge and used that to solve the fourth-order ordinary partial differential equation governing plate deflection for the coefficients of displacement in the vertical direction. The coefficients were only determined at the center of the lower edge of the wall. The solutions at the bottom edge of the wall for a clamped condition and simply supported condition were superimposed to obtain an estimation of the rotational stiffness at that point.

The same procedure was carried out for the floor so that the relative stiffnesses between the two members was found for the purpose of distributing the unbalanced moments. This provided a possible solution at the one location but no comprehensive list of moment values was determined for the entire edge along the bottom. A general case of a rectangular tank was not considered.

In a third paper,[8] Davies considered different support conditions. He assumed that part of the floor could lift off the support and he developed a stiffness coefficient at that point based on the approximation that the section acts like a cantilever beam. However, this procedure was carried out at only one location, the center of the wall, and was subjected to a number of limitations.

In a later article,[9] Davies improved upon his previous solution of a tank resting on a flat rigid support when he assumed a polynomial type function to approximate the displacement of the floor. His results correlated well with experimental results but he only determined and compared an analytical moment at one location.

Davies and Long worked together on a paper[10] to determine the behavior of a square tank on an elastic foundation. They solved the Levy and Naviers problems for the stiffness of the floor slab resting on a Winkler foundation and combined this solution with the solution of a previous paper[7] to determine moment values. The limiting case, though, was a square tank and moment values were only compared at the center of the lower edge of the wall.

Brenneman, in his masters thesis[11] at Virginia Polytechnic Institute and State University, developed a finite element program to determine moments in folded plates. It was, however, limited to fold lines being parallel to each other. Beck[12] expanded and developed Brenneman's program, and compared moment values with those in the PCA bulletin. Beck assumed the bottom edge of the walls was simply supported. Due to the limiting requirement that the axes of the folds be required to be parallel, the program was unable to provide for wall-to-floor interaction and moment transfer.

Articles by Wilby,[13] Lightfoot and Ghali,[14] and Moody[15] contained information that was not directly related to this problem.

In summary, a few very specific problems have been solved to determine moment values at a few locations in a rectangular concrete tank. Most of these solutions were long and very theoretical, and would not provide the practicing engineer a quick and easy, yet good, approximate method for determining the moment values throughout a tank.

III. DEVELOPMENT OF ANALYSIS

Finite Element Approach

The finite element method is used in this analysis because of the versatility and ease with which arbitrary loadings and boundary conditions can be handled. The plate continuum is approximated by a finite number of elements, connected at their nodes, that very closely approximate the behavior of the continuum. The finite element procedure that was developed by Brenneman[11] is extended in this paper to permit the analysis of a tank with monolithic walls and floor and also to allow rotations at joints between the plates. The detailed development of the formulation for the finite element was covered in Brenneman's paper and is only summarized here. Although a triangular element is more suitable to matching irregular boundaries, a rectangular element is used to model the structure because Clough and Tocher[16] have found this element to converge faster and provide more accurate answers than the triangular element.

The equation governing the solution of the finite element problem is given as:

$$[K]\ \{q\} = \{Q\} \qquad (1)$$

where

[K] represents the stiffness matrix of the entire system developed from an approximate displacement function,

{q} is a column vector containing the unknown nodal displacements and

{Q} is a column vector containing the loads acting on the system.

The three matrices used in equation (1) must be in the same coordinate system.

The load vector is generally an easy value to obtain but the stiffness matrix of the system is a critical value. A poor approximation of the stiffness of the system could permit the system to behave in a fashion that does not accurately represent its true behavior. Because the elements are connected at their nodes, there are constraints that must be applied to the approximate displacement functions which enable the discretized system to behave more like a continuum. These constraints require that the displacement pattern provide for:

(1) rigid body displacements - so statics is not grossly violated,

(2) constant strain - limiting case for a very fine mesh,

(3) internal element continuity and

(4) continuity at element interfaces - to avoid infinite strains at element boundaries. (This condition can be relaxed and still maintain convergence, although not monotonic convergence.)

Finite Element Theory in General Terms

The boundaries of a finite element are defined by its nodes (see Figure 1). The displacement pattern or shape function, which satisfies the aforementioned criteria, is used to uniquely define the internal displacements in an element given the displacement at the nodes. The displacement function can be written in matrix notation as:

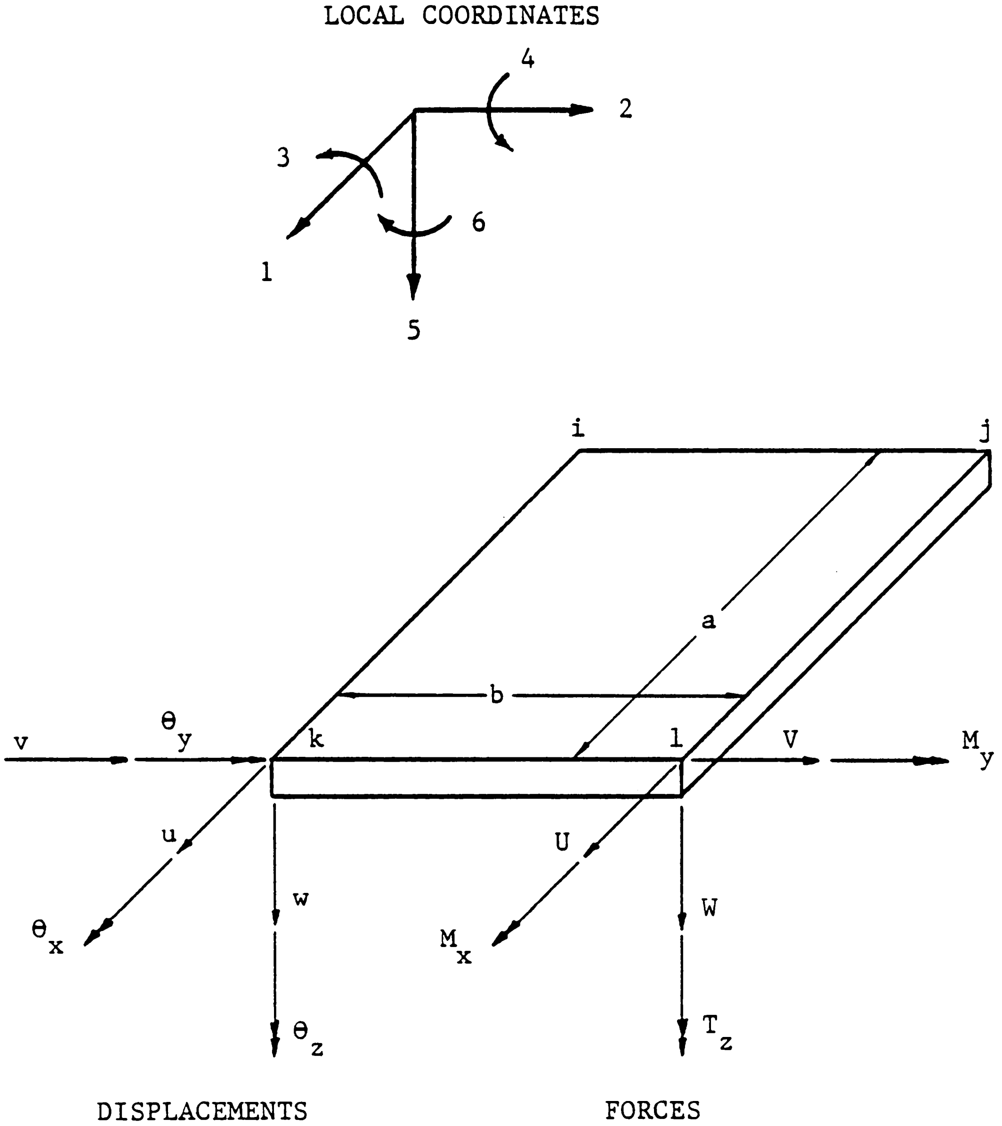


FIGURE 1: Typical element

$$\{u\} = [M]\ \{\alpha\} \tag{2}$$

where

$\{u\}$ = internal displacements at any point in the element,

$[M]$ = coordinates of any point in the element and

$\{\alpha\}$ = generalized coordinates.

The nodal displacements $\{u_n\}$ can be found by:

$$\{u_n\} = [A]\ \{\alpha\} \tag{3}$$

where

$[A]$ is obtained by evaluating $[M]$ at the proper node.

Now the undetermined coefficients in the displacement function can be found by:

$$\{\alpha\} = [A]^{-1}\ \{u_n\} \tag{4}$$

Combining equations (2) and (4)

$$\{u\} = [M]\ [A]^{-1}\ \{u_n\}$$

$$\{u\} = [N]\ \{u_n\} \tag{5}$$

we obtain the internal displacements of an element as a function of the nodal displacements. Strains, which are obtained by differentiation of the displacement, can be written in matrix form as:

$$\{\varepsilon\} = [B]\ \{u_n\} \tag{6}$$

Stresses are related to strains by the constitutive matrix [C] as:

$$\{\sigma\} = [C]\ \{\varepsilon\} \tag{7}$$

Combining equations (6) and (7)

$$\{\sigma\} = [C]\ [B]\ \{u_n\} \tag{8}$$

we obtain the stresses as a function of the nodal displacements. The potential energy of a system can be defined as:

$$\Pi_p = U + W_p \tag{9}$$

where

U is the strain energy of the system and

W_p is the potential energy of any external loads.[17]

The potential energy of the system can be written in matrix form as:

$$\Pi_p = \iiint_{V.} \{\varepsilon\}^T\ \{\sigma\}\ dV - \sum P_i u_i \tag{10}$$

where

P represents any applied loads.

Substituting equations (6) and (8) respectively, the following equation is obtained:

$$\Pi_p = \iiint_{V.} \{u_n\}^T [B]^T [C] [B] \{u_n\} dV - \sum P_i u_i \qquad (11)$$

The system is required to be in equilibrium; thus the minimum potential energy must be found. In order to obtain the minimum potential energy, calculus of variations should be used because of the large numbers of nodal displacements.

Taking the first variation of equation (11) and setting it equal to zero yields:

$$\iiint_{V.} [B]^T [C] [B] \{q\} dV - P_i = 0 \qquad (12)$$

This is in the same form as equation (1) where

$$[k] = \iiint_{V.} [B]^T [C] [B] dV \qquad (13)$$

$$\{Q\} = P_i \qquad (14)$$

Once the strain-displacement matrix [B] is found, the local element stiffness matrix [k] can be determined. The system of local element stiffness matrices are then assembled into a global coordinate stiffness matrix by making appropriate transformations from the local to global coordinate system.

A method of assembling the global stiffness matrix is used so that only the stiffness terms from a degree of freedom at a node are entered into the global stiffness matrix. In other words, if a degree of freedom is zeroed out at a node, its stiffness contribution is not added into the global stiffness matrix. This procedure saves execution time for solving the system of simultaneous equations and does not require any elimination of rows and columns in the stiffness matrix. This does not permit an easy method of applying prescribed boundary conditions. However, the scope of this paper does not require prescribed boundary conditions, so this omission is overlooked.

Once the stiffness matrix is assembled and the load vector determined, equation (1) is solved for the unknown nodal displacements. This process requires that a large number of simultaneous equations be solved. In his master's thesis presented at Virginia Polytechnic Institute,[18] Basham compared the efficiency of several different types of equation solvers. The Linpack equation solver is chosen for this program because it is easy to implement into the program yet still has a shorter execution time than some other schemes.

After the displacements $\{u_n\}$ at the nodes are known, the forces are determined by equation (1).

$$\{f_e\} = \{k\} \{u_e\}$$

where $\{f_e\}$ and $\{u_e\}$ are vectors containing the element nodal forces and element nodal displacements, respectively. This completes the development of the finite element in general terms.

Once an appropriate displacement function is chosen, the stiffness matrix of the element can be determined and the element forces calculated.

Development of Rectangular Element in Combined Extension and Flexure

As mentioned earlier, the details of the development of the element stiffness matrix will not be covered in detail in this paper. The finite element developed is rectangular with four corner nodes and 24 degrees of freedom, six at each node. Associated with each degree of freedom is a force, in matrix form

$$\{q_e\} = \begin{Bmatrix} q_i \\ q_j \\ q_k \\ q_l \end{Bmatrix} \quad \text{and} \quad \{f_e\} = \begin{Bmatrix} f_i \\ f_j \\ f_k \\ f_l \end{Bmatrix} \tag{15}$$

where the subscript e denotes the entire element and the subscripts i, j, k and l denote node numbers as shown on Figure 1 (repeated). A typical node has the following displacements and forces associated with it:

$$\{q_i\} = \begin{Bmatrix} u_i \\ v_i \\ \Theta_{xi} \\ \Theta_{yi} \\ w_i \\ \Theta_{zi} \end{Bmatrix} \quad \text{and} \quad \{f_i\} = \begin{Bmatrix} U_i \\ V_i \\ M_{xi} \\ M_{yi} \\ W_i \\ T_{zi} \end{Bmatrix} \tag{16}$$

These displacements and forces at a node are broken up into three components. The first is the in-plane displacements and forces given by:

$$\{q_i\}^p = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \quad \text{and} \quad \{f_i\}^p = \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} \tag{17}$$

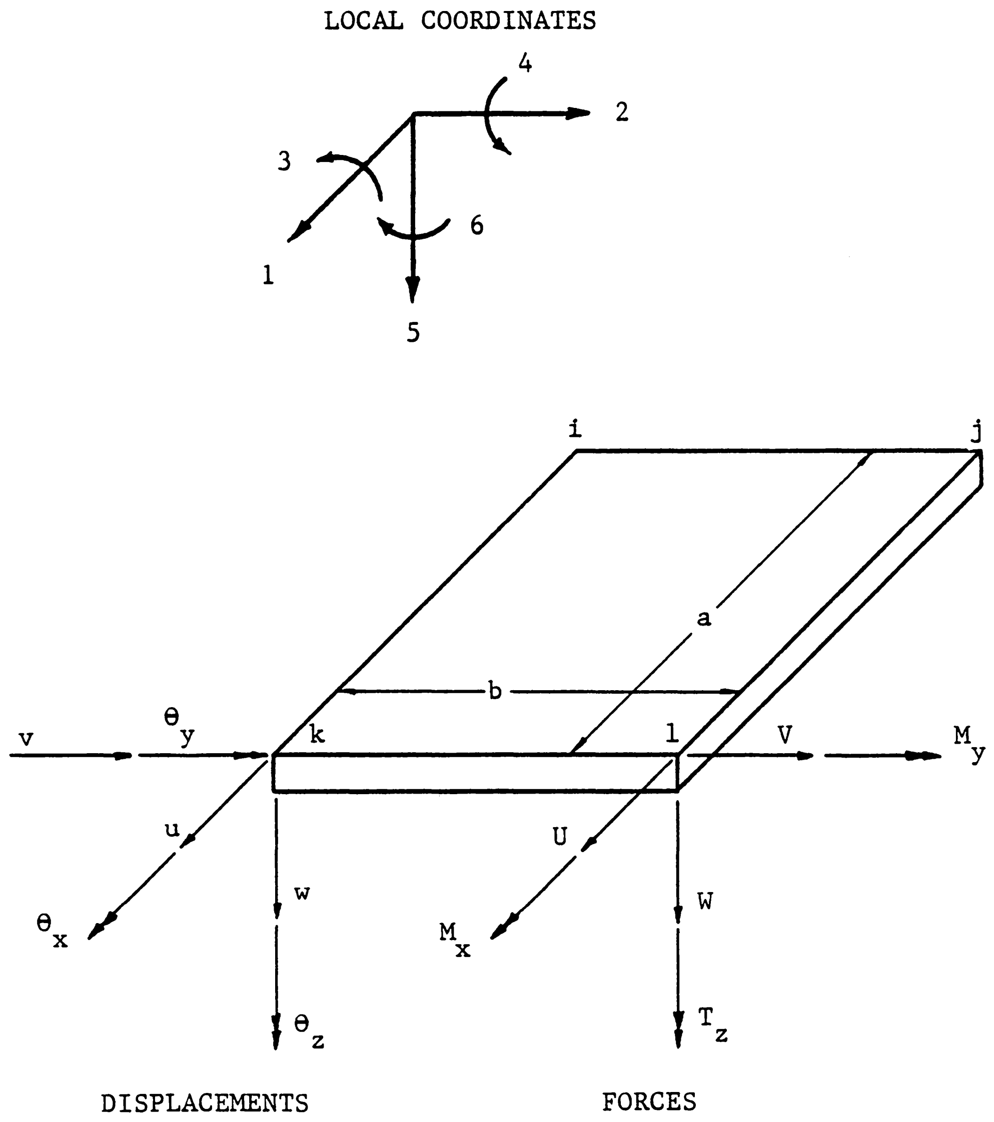


FIGURE 1: Typical element

The second group of terms consists of the displacements and forces associated with plate bending. That is,

$$\{q_i\}^b = \begin{Bmatrix} \Theta_{xi} \\ \Theta_{yi} \\ w_i \end{Bmatrix} \quad \text{and} \quad \{f_i\}^b = \begin{Bmatrix} M_{xi} \\ M_{yi} \\ w_i \end{Bmatrix} \tag{18}$$

The final term is the rotation and corresponding force associated with twisting in the normal (perpendicular) direction of the plate. This single degree of freedom is considered separately in a later section.

The local element coordinate system is also shown in Figure 1 and is important when transformations from local to global coordinates are considered.

The stiffness matrix for an element is a 24 x 24 matrix which can be subdivided into 16 submatrices, each a 6 x 6 matrix containing in-plane, bending and twisting characteristics such that

$$[k_{ij}] = \begin{bmatrix} k_{ij}^p & 0 & 0 \\ 0 & k_{ij}^b & 0 \\ 0 & 0 & k_{ij}^* \end{bmatrix} \qquad i,j = 1,4 \tag{19}$$

where

$[k_{ij}]^p$ is a 2 x 2 matrix that contains the in-plane stiffness of the plate element,

$[k_{ij}]^b$ is a 3 x 3 matrix that contains the bending stiffness terms of the plate element and

$[k_{ij}]^*$ is a 1 x 1 matrix that contains the twisting stiffness term normal to the plane of the plate.

Consider first the determination of the in-plane stiffness matrix terms. This sub-element consists of four nodes with two degrees of freedom at each node, a displacement in the local 1-direction and a displacement in the local 2-direction. Therefore, the displacement function that is chosen must, by necessity, have eight unknown coefficients. Paralleling Brenneman's work, the following displacement function will be adopted as suggested by Zienkiewicz and Cheung[19] and used by Rockey and Evans.[20]

$$u(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy + (\nu/(1-\nu)\alpha_4 - \tfrac{1}{2}\alpha_8)y^2$$
$$v(x,y) = \alpha_5 + \alpha_6 x + \alpha_7 y + \alpha_8 xy + (\nu/(1-\nu)\alpha_8 - \tfrac{1}{2}\alpha_4)x^2 \qquad (20)$$

By performing the formulation as given by the previous section, the stiffness matrix is determined and shown in Table 1 on the following page.

The sub-element required for the development of the plate bending element also has four nodes but has three degrees of freedom at each node, a displacement in the local 3-direction and rotations in the local 4-and 5-directions. Therefore, a displacement function with 12 unknowns must be chosen. The plate bending displacement function adopted for this paper was also suggested by Zienkiewicz and Cheung.[19]

$$w(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 + \alpha_{10} y^3 + \alpha_{11} x^3 y + \alpha_{12} xy^3 \qquad (21)$$

Although this element does not provide compatibility for the normal slopes between elements, Clough and Tocher[16] have shown that this displacement

TABLE 1: In-plane element stiffness matrix

$$
[k_e]^p = \frac{Et}{180(1-\nu^2)}
\begin{bmatrix}
A/p+Bp & D & C/p-Bp & F & -A/p+Bp & -F & -C/p-Bp & -D \\
 & Ap+B/p & -F & -Ap+B/p & F & Cp-B/p & -D & -Cp-B/p \\
 & & A/p+Bp & -D & -C/p-Bp & D & -A/p+Bp & F \\
 & & & Ap+B/p & D & -Cp-B/p & -F & Cp-B/p \\
 & & & & A/p+Bp & -D & C/p-Bp & -F \\
 & & & & & Ap+B/p & F & -Ap+B/p \\
 & & & & & & A/p+Bp & D \\
\text{sym.} & & & & & & & Ap+B/p
\end{bmatrix}
$$

where:

$p = a/b$

$A = 60 + 30\nu^2/(1-\nu)$

$B = 22.5(1-\nu)$

$C = 30 - 30\nu^2/(1-\nu)$

$D = 22.5(1+\nu)$

$F = 22.5(1-3\nu)$

function will provide satisfactory results. The stiffness matrix for the plate bending element is shown in Table 2.

These two independent groups of stiffness terms can now be combined into one stiffness matrix as shown by equation (19). This permits the simultaneous solution of both problems.

Coordinate Transformations

The rectangular element developed in the previous section has only five degrees of freedom at each node. In order to assemble these elements in three dimensions, a sixth degree of freedom must be available so that proper mapping of displacements, forces and stiffness coefficients is possible. Brenneman[11] resolved this problem by incorporating three different coordinate systems.

The five degrees of freedom already developed included three displacements and two in-plane bending rotations. The sixth degree of freedom that needs to be examined is the twisting stiffness normal (perpendicular) to the plane of the plate. If the magnitude of this twisting stiffness is considered, it is intuitive that the resistance to rotation in this direction is considerably larger than the in-plane bending stiffnesses. Therefore, it is assumed for the purposes of this analysis that the twisting stiffness normal to the plate is infinite and can be approximated as a fixed condition.

Although this approximation does not benefit the general folded plate problem, it does, however, lend itself quite well to the case where the plates are joined at 90° angles to each other provided the global coordinate system coincides with the orientation of the plates. The normal

TABLE 2: Plate bending element stiffness matrix

$$
[k_e]^b = \begin{bmatrix}
SA & -SB & -SD & SG & 0 & -SH & SN & 0 & SO & SP & 0 & SQ \\
 & SC & SE & 0 & SI & SJ & 0 & SR & -SS & 0 & ST & -SU \\
 & & SF & SH & SJ & SM & SO & SS & SX & -SQ & SU & SY \\
 & & & SA & SB & SD & SP & 0 & -SQ & SN & 0 & -SO \\
 & & & & SC & SE & 0 & ST & -SU & 0 & SR & -SS \\
 & & & & & SF & SQ & SU & SY & -SO & SS & SX \\
 & & & & & & SA & SB & -SD & SG & 0 & -SH \\
 & & & & & & & SC & -SE & 0 & SI & -SJ \\
 & & & & & & & & SF & SH & -SJ & SM \\
 & & & & & & & & & SA & -SB & SD \\
 & & & & & & & & & & SC & -SE \\
 & \text{sym.} & & & & & & & & & & SF
\end{bmatrix}
$$

where: A, B are half of the element dimensions

$p = a/b$
$Dx = Dy = Et^3/(12(1-\nu^2))$
$D1 = \nu Dx$
$Dxy = 0.5Dx(1-\nu)$
$PDx = Dx/p^2$
$PDy = Dyp^2$
$SA = (20PDy + 8Dxy)B/15A$
$SB = D1$
$SC = (20PDx + 8Dxy)A/15B$
$SD = (30PDy + 15D1 + 6Dxy)/30A$
$SE = (30PDx + 15D1 + 6Dxy)/30B$
$SF = (60PDx + 60PDy + 30D1 + 84Dxy)/60AB$
$SG = (10PDy - 2Dxy)B/15A$
$SH = (-30PDy - 6Dxy)/30A$
$SI = (10PDx - 8Dxy)A/15B$
$SJ = (15PDx - 15D1 - 6Dxy)/30B$
$SM = (30PDx - 60PDy - 30D1 - 84Dxy)/60AB$
$SN = (10PDy - 8Dxy)B/15A$
$SO = (-15PDy + 15D1 + 6Dxy)/30A$
$SP = (5PDy + 2Dxy)B/15A$
$SQ = (15PDy - 6Dxy)/30A$
$SR = (10PDx - 2Dxy)A/15B$
$SS = (30PDx + 6Dxy)/30B$
$ST = (5PDx + 2Dxy)A/15B$
$SU = (15PDx - 6Dxy)/30B$
$SX = (-60PDx + 30PDy - 30D1 - 84Dxy)/60AB$
$SY = (-30PDx - 30PDy + 30D1 + 84Dxy)/60AB$

twisting resistance of the plates can then always be identified and clamped as a boundary condition to eliminate that stiffness term in the system stiffness matrix. This makes it possible for the solution to be independent of the normal stiffness of an element.

The completed local element stiffness matrix at a node would be a 6 x 6 matrix containing three submatrices. The first submatrix, a 2 x 2, would contain the in-plane stiffnesses; the second submatrix, a 3 x 3, would include the bending stiffnesses of the plate; and the third, a 1 x 1, would be a zero provided as a dummy value only to aid in the transformation of coordinate systems.

Rectangular tanks are obviously a good example of plates that meet at 90°. At wall-to-wall joints, a plate in one direction provides an in-plane fixed support to the adjoining plate, preventing vertical rotation in the second plate yet allowing a moment to be developed there. The same support would be provided to the first plate from the second.

In the corners of the tank, the floor plate provides a fixed condition at the bottom node of the wall-to-wall joint, but still allows the joint to rotate throughout its full height. The same fixed condition holds true for the walls and the accompanying wall-to-floor joint.

In summary, throughout the interior of the plate, all the normal rotations to the plate are fixed. At the edges, two rotations are constrained (one normal restraint from each plate) yet allowing the entire joint to rotate. At the corners, three rotations are constrained (one from the normal restraint of each of the three plates).

IV. PROGRAM DEVELOPMENT

Coordinate Systems

At this time it is important to mention the coordinate systems and some terminology that is used throughout the remainder of the paper.

One quarter of the rectangular tank is analyzed to take advantage of symmetry. This minimizes the number of degrees of freedom and the core space required and greatly reduces the execution time of the solve routine. The boundary conditions are automatically applied at the lines of symmetry to decrease user input.

Figure 2 shows a sketch of some of the more pertinent information. It is important to note the orientation of the global axes. The origin of the system is located at the corner of the tank and the axes are coincident with the joints where the plates meet. Plate 1 lies in the global 1-2 plane; plate 2 lies in the global 2-3 plane; and plate 3 lies in the global 1-3 plane. Element dimensions are represented by c, a, and b in the X-, Y- and Z-directions, respectively. The local coordinate system has already been illustrated in Figure 1.

The node numbering scheme proceeds across plates 1 and 2 down to the floor, and then across the floor (with constant X). Assuming eight elements in each of the three directions, a few of the node numbers have been shown on Figure 2.

Three general categories of problems that are analyzed; namely, a single plate problem, a two plate problem and a three plate problem. All three problems have the normal twisting degree of freedom automatically eliminated. The one plate problem corresponds to any single plate analysis

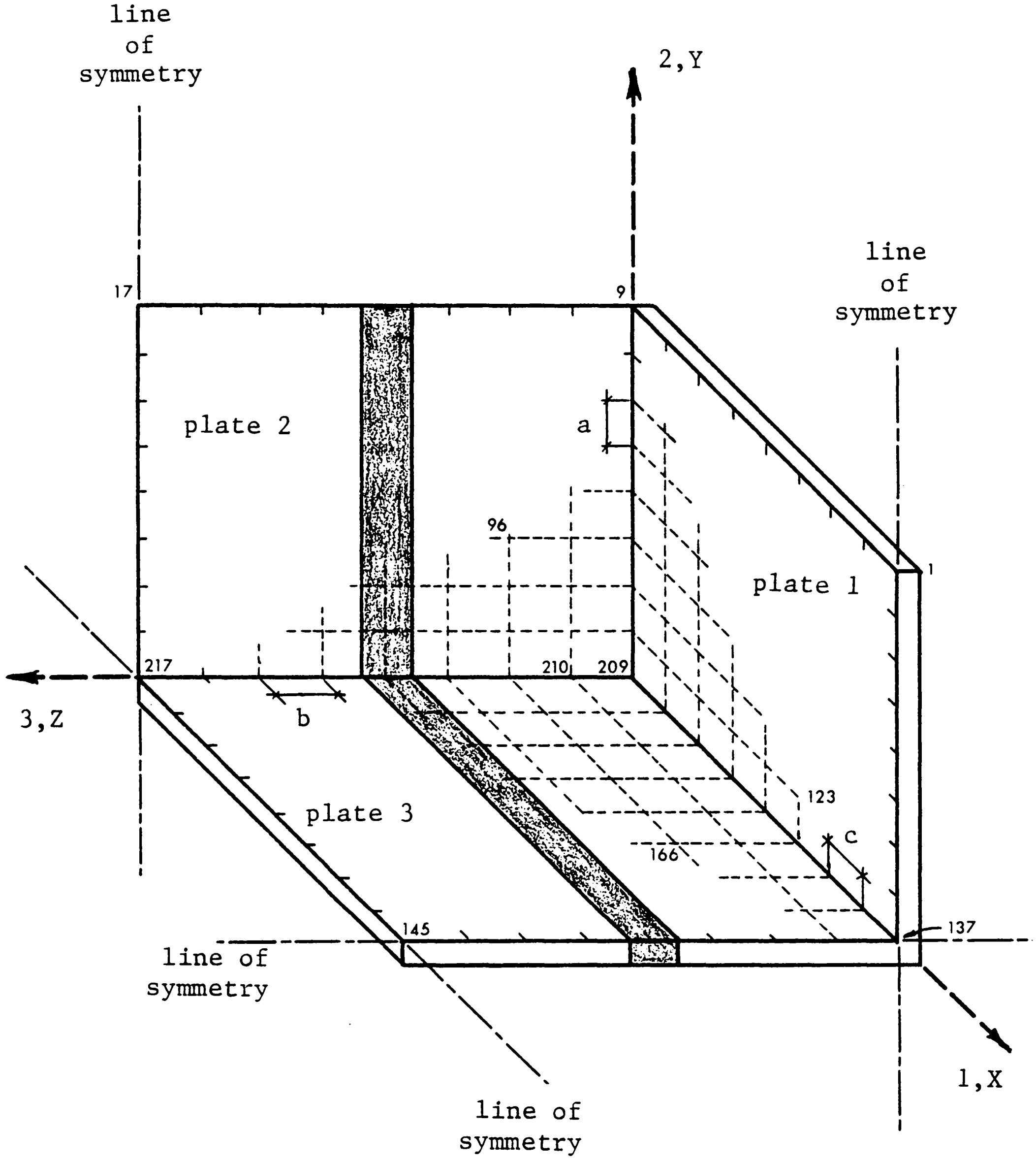


FIGURE 2. One quarter of tank

and will be characterized by a description of the boundary conditions and loading parameters.

The two plate problem refers to the analysis of two plates meeting at 90°. The two plates represent the walls in this paper and represent plates 1 and 2 of Figure 2. The top edge is always considered free and further described by the boundary condition along the bottom edge. Symmetry is utilized and the appropriate boundary conditions are automatically generated along the two cut edges. The joint between the two plates is free to displace and rotate as governed by the loading conditions. This analysis allows wall-to-wall interaction.

The third category, the three plate problem, has appropriate boundary conditions automatically generated to simulate the symmetry of one quarter of a tank (walls and floor). In addition, the floor of the tank is edge-supported. This is discussed in a later section. The top edge of the walls are always considered free. The analysis of this problem is generally characterized by the type of loading acting on the floor slab. By analyzing the three plates together as a unit, it is possible to obtain the interaction of the three plates and permit rotations of the joints that develop from the unbalance in moments.

Loading Considerations

Before a solution to equation (1) can be found, consideration is given to the loads acting on the tank. The loading condition for the walls and floor is handled separately. For the walls, there are generally only two types of loading conditions that normally occur on the walls; namely, a triangular load or a uniform load. The triangular load represents hydrostatic pressure from a fluid or earth pressure from a soil. The uniform load is used to model a surcharge on the tank. The program is designed to handle these loading conditions for a variable height and they can be internal or external loads.

There is an approximation inherent in the development of the load vector for these problems. The loads are idealized as concentrated loads acting at the nodes. The magnitude of the node load is determined by multiplying the tributary area around the node, generally half the element's dimension in each direction, by the average pressure acting over that area. This does not, however, create a significant error provided the mesh chosen is small enough (say 8 x 8).

Two types of loadings are considered for the floor slab. The first type of loading is the inclusion of the stiffness of the soil into the system stiffness matrix, and the second is the consideration of a strip load around the perimeter of the floor slab.

The inclusion of the soil stiffness into the system stiffness matrix is accomplished by approximating the stiffness of the soil in units of force per length and adding this value along the diagonal of the system stiffness matrix at the degrees of freedom in the vertical direction for the nodes of the floor slab.[17]

It is anticipated that a triangular load will normally be applied to the tank's walls, a strip load to the floor slab, and the soil stiffness included as mentioned above. To do this, it is necessary to provide a restraint in the vertical direction so that the system would remain in equilibrium. One solution is to support the floor slab on the edges in the vertical direction. However, this does not accurately represent the action of the system as a whole. It is intuitive that the tank will undergo a settlement if it is filled with a material so such an edge restraint is not appropriate. Another possible solution is to consider the floor slab to be resting on a bed of springs sandwiched between two planes of nodes. It was decided to eliminate the soil stiffness from this study and leave that development to others as it is beyond the initial scope of this paper.

A simpler solution is developed assuming the floor slab to be resting on a homogeneous soil that reacts with a uniform pressure. The settlement of the tank is included in this approximation by assuming that the weight of material inside the tank and the weight of the floor slab cause a uniform settlement of the entire tank. From this settled position, displacement in the vertical direction is constrained. The only remaining unbalanced force then is the weight of the walls.

Paralleling the current AISC steel code, it is assumed that the shear from the walls is transferred through the footing at a slope of 2.5:1. The weight of the walls is then distributed uniformly over a strip around the perimeter of the floor with a width of the thickness of the wall plus 2.5 times the thickness of the footing. This appears to be a better approximation to the distribution of shear rather than distributing the weight of the walls uniformly over the entire floor slab because in a large tank it is difficult to imagine part of the weight of the wall carried by the center portion of the tank.

Now that the stiffness matrix of the finite element has been determined and the loading conditions approximated, equation (1) can be solved for the unknown nodal displacements. With this information, the forces are determined at all the nodal points.

V. DISCUSSION OF RESULTS

Comparison with Known Solutions

Since a program was developed for this paper, it was important to verify its accuracy with well accepted solutions. The analysis of a single plate was considered first because there are many sources of solutions available for this problem with various loadings.

The value of Poisson's ratio used for all of the analyses was 0.2. The modulus of elasticity of the concrete was chosen to be 3000 ksi. The tanks or plates analyzed were generally 10′ in height, but cases where the wall height was not 10′ are mentioned in later sections.

At this time, it is appropriate to introduce some terminology that is used in the remainder of this paper to describe various cross-sections through the tank. A redefining of coordinates is introduced because most practioners who design tanks are familiar with the coordinate system that was adopted by the PCA when it published bulletin ST-63.[1] That coordinate system is shown in Figure 3. The origin of the coordinate system is moved to the center of the tank and the letters a, b and c now represent the full dimensions of the tank in the X-, Y- and Z-directions, respectively. A cross-section cut through the center of plate 1 by an X-Z plane is referred to as a strip at $y = 0$. A strip cut by an X-Z plane through the quarter-point of the wall and floor is located at $y = b/4$, etc. M_x is a vertical moment in the X-direction (or around the Y or the Z axes). M_y and M_z are horizontal moments in the Y- and Z-directions, respectively (or around the X axis).

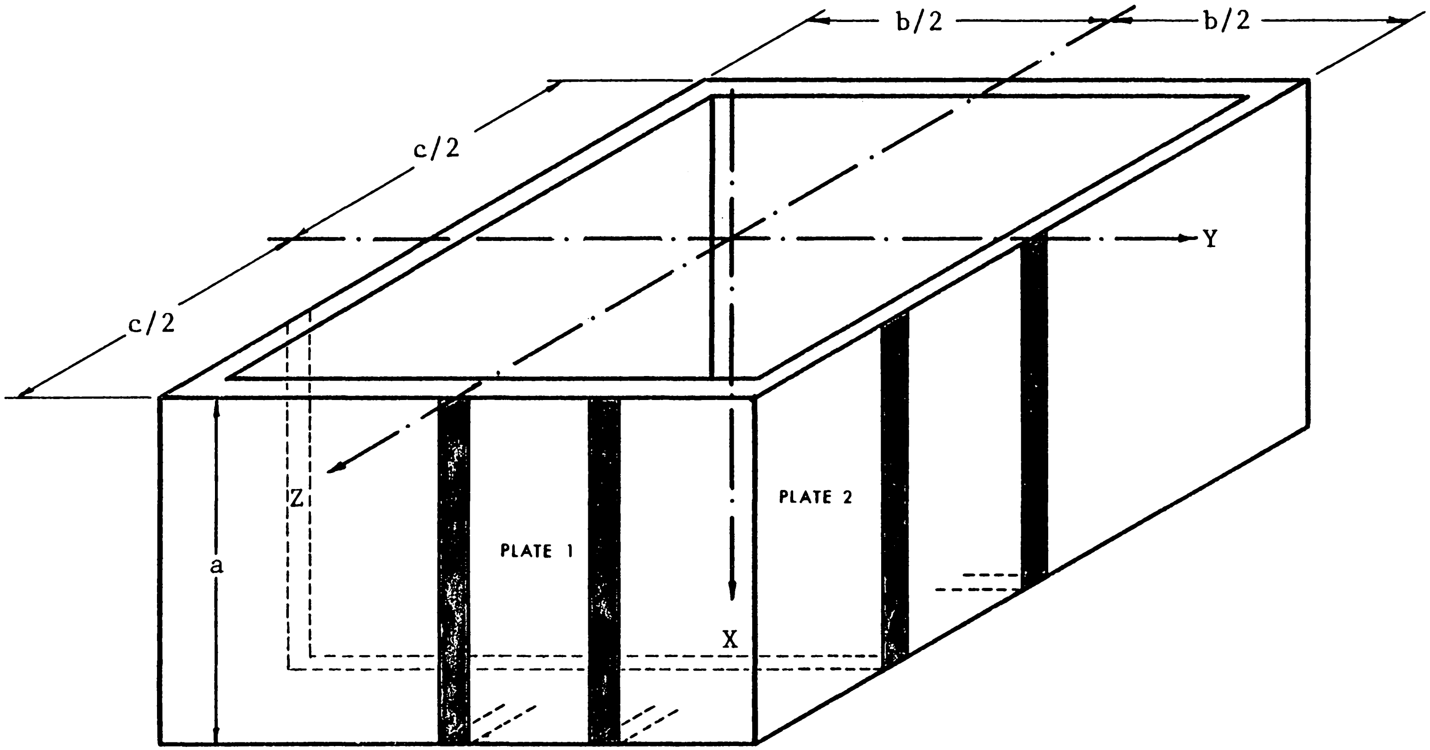


FIGURE 3: PCA coordinate system

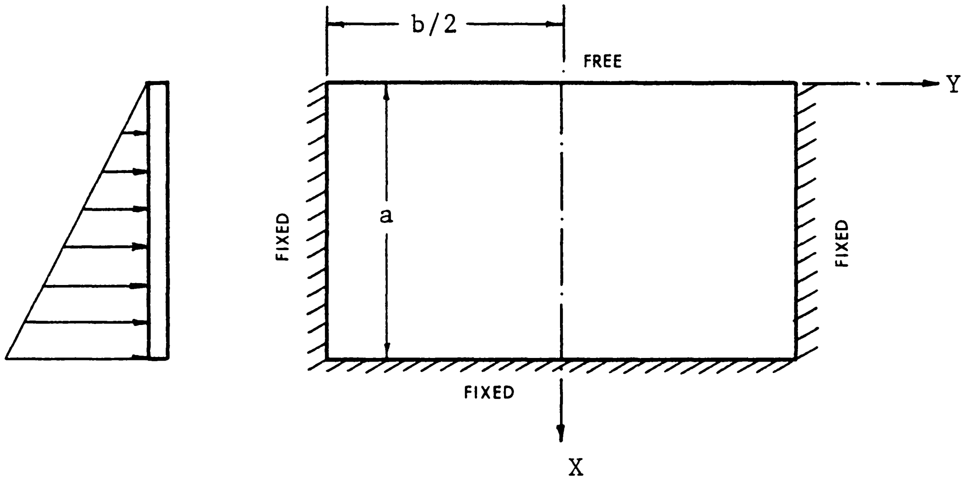
One of the first problems compared with a known solution was a single plate problem having three edges fully clamped, one edge free, and a triangular load as obtained from normal water pressure applied to it. Moment values calculated by the finite element program were compared with those from the PCA bulletin[1] and Jofriet.[2] Shown in Table 3 is a comparison of the horizontal and vertical moments in a cross-section at y = 0. The ratio of width-to-height (b/a) is 2.0. Eight elements are used in each direction and the plate is of uniform thickness.

The maximum vertical and horizontal moments calculated appear to compare fairly well with the PCA values and Jofriet. There are a few places though, where the percentage difference between the answers is fairly significant, caused by the order of magnitude of the numbers. The order of magnitude of the numbers changes by a factor of more than 10. Therefore the relative percent of change appears large for the smaller moment values.

A single plate problem with the two sides clamped, top free, bottom simply supported and a triangular load applied to it was considered. The moment values were compared at y = 0, y = b/4 and y = b/2, and the results are more favorable than the first case. There is greater error at y = b/2, but the comparison with the PCA bulletin at y = 0 is shown in Table 4 for simplicity.

The program developed for this paper is capable of handling tapered wall thicknesses, so it was desirable to compare that solution with a known solution. Jofriet[2] has a few limited tables of moment coefficients for walls with tapered thickness. A wall with three edges clamped and one edge free was compared for b/a = 2.0. The thickness at the bottom of the wall

TABLE 3: Comparison with known solutions



triangular loading

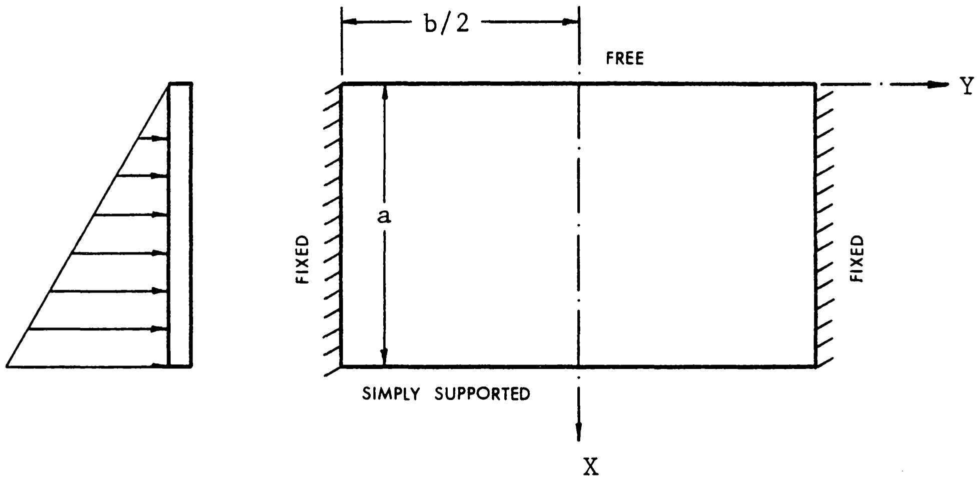
b/a = 2.0

constant thickness

y = 0

		F.E.	PCA	%diff	Jofriet	%diff
x/a = 0	M_y	13.22	12.64	4.59	12.64	4.59
	M_x	0.0	0.0	--	0.0	--
1/4	M_y	11.11	10.76	3.25	11.23	-1.07
	M_x	5.63	6.08	-7.40	5.62	0.18
1/2	M_y	7.71	7.49	2.94	7.96	-3.14
	M_x	7.48	7.02	6.55	7.49	-0.13
3/4	M_y	3.45	1.40	146.00	1.40	146.00
	M_x	1.33	3.74	-64.40	3.28	-59.40
1	M_y	6.42	7.96	-19.30	--	--
	M_x	39.63	40.25	-1.54	39.31	0.81

TABLE 4: Comparison with known solutions



triangular loading

b/a = 2.0

constant thickness

y = 0

		F.E.	PCA	%diff
x/a = 0	M_y	21.83	21.06	3.70
	M_x	0.0	0.0	--
1/4	M_y	19.90	19.66	1.20
	M_x	7.83	7.49	4.50
1/2	M_y	17.20	16.85	2.10
	M_x	15.65	15.44	1.40
3/4	M_y	11.03	11.23	-1.80
	M_x	16.36	16.38	-0.10

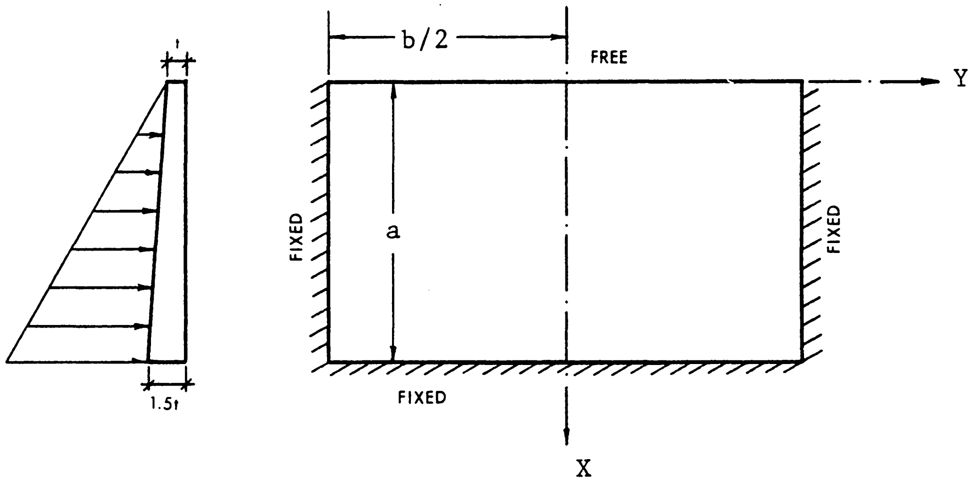
was 1.5 times the thickness at the top. Correlation with Jofriet's solution is quite good at y = 0, y = b/4 and y = b/2. The comparison at y = 0 is shown in Table 5.

The PCA table that is contained in bulletin ST-63, and which accounts for wall-to-wall interaction for the case when the bottom edges of the wall are simply supported by the floor was also used to check results from the program. Adequate coorelation exists for this case also.

The strip loading (vertical load on the footing slab) was also checked against a known solution. For this a single plate was clamped on all four sides and a strip load was applied to it. The need for this loading condition is explained in more detail in a later section. Bauverlag[21] developed an extensive collection of moment coefficients for plates with various loadings and boundary conditions. From this book, a solution for a strip load is obtained by superimposing the solutions of a uniform load with that of an appropriate rectangular load of opposite sign. The maximum moment at the edges for the finite element solution is compared with Bauverlag's values and very good correlation is found.

A plate problem with a triangular load and walls of equal length was examined to check for round-off errors in the solution process that might have occured due to the increased number of degrees of freedom. The answers were symmetric, as expected, because the vertical joint between the walls does not rotate in a square tank. There is, however, a slight difference with the moments that are listed in Table 3. These two problems should have produced similar answers. Although the difference is very small, it did warrant justification. Apparently the vertical joint in the corner of the tank experiences an outward displacement due to the internal

TABLE 5: Comparison with known solutions



triangular loading

b/a = 2.0

tapered thickness

y = 0

		F.E.	PCA	%diff
x/a = 0	M_y	7.78	7.02	10.80
	M_x	0.0	0.0	--
1/4	M_y	7.86	7.96	-1.26
	M_x	3.67	3.74	-1.87
1/2	M_y	6.20	6.08	1.97
	M_x	3.36	3.28	2.44
3/4	M_y	0.75	0.47	60.00
	M_x	9.58	9.83	-2.54
1	M_y	8.02	--	--
	M_x	48.05	48.20	-0.31

hydrostatic loading. This movement is eliminated by the fully clamped condition assumed in the single plate problem. The two problems are therefore not exactly the same, which explains the small discrepency in the moment values. The analyses performed on any two or three plate problems in this paper do not have the vertical or horizontal edges between plates constrained from this type of movement and therefore more accurately represent the true behavior of the tank.

Referring to Moody,[15] Poisson's ratio does appear to effect the value of moments at the interior of a plate and this could be another explanation for some of the discrepencies experienced with the known solutions. It is believed that the PCA tables use 0.15 as the value for Poisson's ratio. Moody pointed out, however, that Poisson's ratio has little effect on the extreme moments of a plate which are most important to design.

Moment Coefficients

The program written for this paper determines the moment values at the nodes in kip-inches. In an attempt to develop a set of tables similar to the PCA tables, the moment values given by the program are divided by half the element length to obtain units of kip-in/in, and then by the specific weight of the fluid and the height cubed. For a constant b/a and c/a ratio, the moment coefficients fluctuate slightly when the thickness of the walls and floor are varied. However, referring to Table 6, for a constant b/a and c/a ratio and the same floor and wall thickness, the moment coefficients are not constant with varying height as they are in the PCA tables for single plates. In other words, the moment coefficients in a tank are a function of the height of the wall. In order to develop moment

TABLE 6: Three plate moment coefficients

b/a = 2.0 c/a = 2.0

Node	height = 10'				height = 8'			
	walls = 8" floor = 10"		walls = 12" floor = 16"		walls = 8" floor = 10"		walls = 12" floor = 16"	
	M_x	M_y	M_x	M_y	M_x	M_y	M_x	M_y
9	-0.082	-0.015	-0.086	-0.017	-0.085	-0.016	-0.091	-0.019
17	0.041	0	0.042	0	0.042	0	0.046	0
73	0.015	0.017	0.016	0.018	0.016	0.018	0.017	0.020
77	-0.077	-0.012	-0.081	-0.012	-0.080	-0.012	-0.087	-0.013
137	-0.004	-0.034	-0.002	-0.025	-0.005	-0.027	-0.001	-0.012
213	-0.002	-0.013	-0.004	-0.005	-0.004	-0.007	-0.005	-0.005
141	-0.031	-0.024	-0.050	-0.038	-0.037	-0.028	-0.070	-0.052
145	-0.027	-0.027	-0.043	-0.043	-0.032	-0.032	-0.060	-0.060
177	-0.025	-0.025	-0.041	-0.041	-0.030	-0.030	-0.054	-0.054

coefficient tables for the three plate problem (i.e., the tank), a group of tables must then be calculated including several values of height for a given set of b/a and c/a ratios, and varying floor and wall thicknesses. To assemble such a collection of tables would be an expensive and lengthy undertaking, and the designer might still lack the table needed to solve his problem. With this in mind, the moment distribution process is looked to as a possible solution.

VI. THE MOMENT DISTRIBUTION PROCESS

General Formulation

The moment distribution method is quite often used to analyze symmetric beam structures that exhibit joint rotations when they are loaded. The rotations develop from unbalanced moments at a joint, whose values are subsequently balanced to provide equilibrium at that joint. The unbalanced moment is redistributed to the adjoining members in proportion to the relative stiffness of each. The main steps in the moment distribution process are to determine the fixed-end moments, calculate the distribution factors, and balance the moments.

In beam structures, the fixed-end moments are determined by locking all joints and calculating the moments at the ends of the beams. A counter-clockwise resisting moment at the end of a beam is considered positive in this paper. It is then necessary to find the stiffness of each member coincident at a joint so that the relative stiffnesses can be found. The stiffness of a member is determined by imposing a unit rotation at one end of the beam and calculating the moment required to cause this unit rotation (as a function of EI/L). This stiffness value reflects the support condition at the far end of the beam. After the member stiffnesses are calculated, the unbalanced moments are redistributed proportional to the relative stiffnesses at a joint. Any external joints are unlocked, balanced and left unlocked. Internal joints are sequentially unlocked and balanced, one at a time. Before the joint is locked, the distributed moment is carried-over to the far end of the beam. For beams of constant cross-section, a carry-over factor of 1/2 is used. The carry-over is performed only if the far end of the beam is clamped at the time the joint is balanced. The balancing of internal joints is carried-out, one at a time, until the carry-over factors are negligible.

In order to apply this process to the tank (an assemblage of three plates), the fixed end moments of the system must be determined. Then the relative stiffness between adjacent members must be calculated so that unbalanced moments can be redistributed. This general process is extended to accomodate a moment distribution method applied to tanks.

Extension to Tank Problem

In an effort to provide the practising designer with a reasonably simple procedure for calculating some of the maximum moments in a rectangular tank, the moment distribution method is modified to redistribute and balance moments at the joints where the walls and floor slab meet. The two main modifications to the moment distribution process as it is applied to beams requires that modified fixed-moments be determined and that the relative stiffnesses between the two plates be calculated. With these two factors developed, the moment distribution process is carried out exactly like the elementary procedure applied to beam structures except that there is no carry-over to the top (or free) edge of the tank.

For this paper, the balancing of the moments is only considered at the joint where the walls and floor slab meet, later referred to as the vertical direction. Since the beam structures can be discretized into individual elements, a similar approximation consisting of two parts is applied to the tank which is a continuum. First, the tank system is broken down into two main sections. The two walls act together as one section and the floor slab acts as the second section. Due to symmetry, each wall, as it is referred to here, is actually only half the length of the wall of the entire tank. The terminology used throughout this section only refers to one quarter of the tank but can obviously be extended to the entire tank.

And second, each section of the tank is divided into strips which provide the beam discretization. These strips permit moment distribution to be carried out at any location along the joint where the plates meet, however, for simplicity, the balancing is only performed at the center ($y = 0$, $z = 0$) and quarter points ($y = b/4$, $z = c/4$) of the entire wall (refer to Figure 3). With this discretization in mind, it is necessary to determine the fixed-end moments on the individual strip elements and calculate the relative stiffnesses of the strips at the joint where balancing is considered.

Determination of Fixed-end Moments

As mentioned earlier, the determination of the fixed-end moments plays an important role in the moment distribution process. It is important to calculate the fixed-end moments in such a way so as to reflect the behavior of the system. Considering the floor slab first, as a very crude approximation, a strip in the floor slab could be idealized as a "beam" removed from the continuum with appropriate loads acting on it. These loadings are a uniform load over the entire length of the "beam" or two sections of uniform load (of greater magnitude) at each end of the "beam" that would represent the strip load. For any location along the floor slab though, the fixed-end moments for this "beam" section would be constant, yet, from plate theory, moments tend to decrease in magnitude toward the corner. Therefore, the "beam" idealization does not satisfactorally represent the behavior of the floor slab.

A second and more suitable arrangement for calculating the fixed-end moments at a location is to analyze the floor slab as a plate and use the moment values of the plate solution at the proper location. This method is adopted because it accurately represents the behavior of the plate. The plate is analyzed with all four edges clamped and is loaded with a uniform load or a strip load around the perimeter of the plate. From this point on, the strip load is used to represent the reaction of the soil pressure on the tank. The nature of this load is explained in a later section.

Some solutions for the moment values at the center and the quarter points of a plate loaded with the strip load are included in Appendix 1. The moment values are in kip-ft/ft/foot of wall height. The magnitude of the loading is determined by dividing the weight of the walls by the area of the strip around the edge of the plate. A fairly comprehensive table of values computed by the finite element method is included in Appendix 1 for several combinations of b/a and c/a. A slightly more extensive listing of moment coefficients for this loading condition can be found in Bauverlag[21] by superimposing uniform and partial load values.

With the fixed-end moments of the floor slab taken care of, it is necessary to determine the fixed-end moments for the wall section. It is anticipated that known solutions would produce satisfactory results for this case, i.e., simply assume the wall-to-wall joint to be clamped and calculate the fixed-end moments at the bottom by assuming that edge to be clamped and the top edge free. However, this does not represent the wall-to-wall interaction that occurs in long tanks. It is necessary to provide a two plate solution that accounts for the horizontal interaction of the walls. The two plates (one quarter of the tank) have a clamped bottom edge and free top edge. The vertical joint between the two plates is unrestrained so that rotation can occur. The fixed-end moments shown in Appendix 2 are calculated by the finite element method at the quarter points and center and are used in conjunction with the corresponding moments from the floor slab in the moment distribution process.

Determination of Stiffness Characteristics

To distribute the unbalanced fixed-end moments, it is necessary to calculate the relative stiffness of the two strips that meet at the joint between the two plates. Consideration is given to a process parallel to that used by Davies,[7] in which the stiffness of the wall was taken to be a function of the clamped moment value and the hinged rotation at a given location. However, to represent the interaction of the plates in the tank, it is necessary to analyze a plate with elastically restrained edges. Although the inclusion of the elastic restraint is a simple matter, the accurate assessment of its value is very difficult to determine for rectangular tanks. But without the relative stiffness of the strips, it is not possible to carry out the moment distribution process.

The moment distribution method can be considered to have three parts, the fixed-end moment values, the relative stiffnesses of the members involved, and the computed answer(balanced moments). Usually the first two parts, as well as the distribution percentages, are known and the answer is found. However, in this case, the fixed-end moments and the answer are known. It is possible then to back-calculate for the relative stiffnesses of the members. If the distribution factors are collected in a compact set of tables, it is possible for the practicing designer to calculate the known solution using a simple moment distribution method. If the stiffness coefficients are only a function of the b/a and c/a ratios, an easy-to-use solution process can be developed to determine the balanced moments provided by joint rotations in a rectangular tank without requiring extensive tables to be developed to cover the moment coefficients for various sizes of tanks.

This approach is adopted for this paper. By trial and error, the relative stiffnesses of the two strips coincident at a joint are calculated such that the subsequent moment distribution with the appropriate fixed-end moments produces the moment at that location as determined by the finite element analysis of the quarter of the tank. The fact that only the relative stiffnesses of the adjoining members need to be found means that the absolute stiffness of each member need not be determined. For simplicity, the stiffness of the wall strip is taken as 4EI/L and the stiffness of the floor strip is (f)4EI/L. I and L are the appropriate properties of a given strip and f is the factor which is found by iteration such that the calculated relative distribution factors produce the desired solution. Since moment values are given in units of kip-ft/ft, a strip is considered to be one foot (12 inches) in width.

The distribution factors are determined by dividing the stiffness of a member by the sum of the stiffnesses at a joint. In this case there are only two strips at a joint. It was hoped that a pattern in the plot of the f factor would develop for various combinations of b/a, c/a, wall thickness and floor thickness, yet remain independent of the height of the tank.

At this point, it is appropriate to provide an example to more clearly show the moment distribution process and the effect of the f factor. Consider a tank with b/a = 2.0 and c/a = 2.0. The walls are 10" thick and the footing is 12" thick. If the height of the walls is assumed to be 10′ (120") high, from Table A1 the fixed-end moment of the floor slab is found to be 0.219 (10) = 2.19 k-ft/ft at z = 0. From Table A2, the fixed-end moment for the wall system is $-0.086\ (0.0624)(10)^3 = -5.37$ k-ft/ft, assuming the tank is filled with water under atmospheric pressure. The stiffness of the wall is given by

$$S_w = \frac{4EI}{L} = \frac{4(3000)(12)(10^3)}{120(12)} = 100{,}000 \text{ k-in}$$

and the stiffness of the floor by

$$S_f = \frac{(f)4EI}{L} = \frac{f(4)(3000)(12)(12^3)}{240(12)} = 86{,}400f \text{ k-in}$$

The relative stiffnesses are then calculated as follows

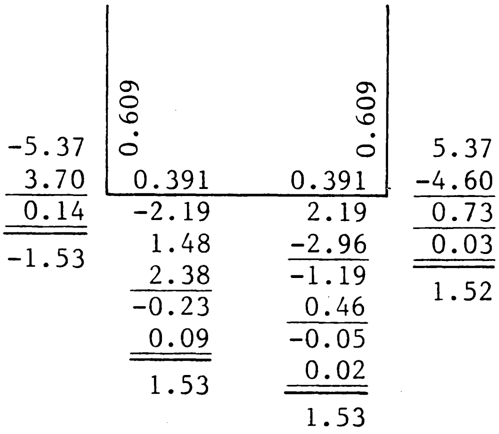
$$r_w = \frac{100{,}000}{100{,}000 + 86{,}400|f|}$$

$$r_f = \frac{86{,}400f}{100{,}000 + 86{,}400|f|}$$

The moment value that is obtained by the finite element program is 1.52 k-ft/ft. If we assume f = 0.744, we obtain

$$r_w = 0.609 \quad \text{and} \quad r_f = 0.391$$

and noting that clockwise rotations on member ends are positive, the moment distribution process is carried out as follows, using a carry-over factor of 1/2:



This result compares quite favorably with the value from the program; therefore, the assumed value of f is good.

It is intuitive that the wall and floor stiffnesses will increase as they approach the edges of the tank. However, it appears as though the wall increases its stiffness at a faster rate than the floor due to the decrease in the f factor. This is probably attributed to the free edge at the top of the walls. It provides little aid to the resistance at a central strip but the support from the edges of the plate is more pronounced at the outer strips.

Now that it is possible to determine the relative stiffness of the strips coincident at a joint, several combinations of wall thickness are considered for b/a = c/a = 2.0. A programmable hand calculator (HP-41CV) was utilized to aid in the calculation of the f factor for the large number of problems solved. Wall thicknesses used include 8", 9", 10" and 12" and floor thicknesses include 10", 11", 12", 13", 14", 15" and 16" for the initial b/a = c/a = 2.0.

The first group of f factors that were calculated at z = 0 for b/a = c/a = 2.0 and wall height equal to 10′ are plotted on a graph having the floor thickness as the independent variable and the f factor as the dependent variable (see Figure 4 on the following page). What developed is a family of curves that form an enclosed area in the vicinity of $f = 0.75$. A similar graph developed at $z = c/4$ also forms an enclosed area close to 0.50 and is shown on Figure 5. For a tank 10′ tall then, it is possible to go to these graphs and determine the required f factor given the thicknesses of the walls and floor, so that the moment distribution process can be carried out.

This provides a simple solution for $b/a = c/a = 2.0$ and a 10′ high wall but the question still persists as to whether or not the f factor is simply a function of the b/a and c/a ratios or whether or not it is also a function of the height. In an effort to resolve this problem, several different sizes of tanks are analyzed, but all have $b/a = c/a = 2.0$. The heights of the different tanks include 7′,8′, 9′ and 15′; the walls are 8" and 12"; and the floors are 10" and 16" thick. This provides a framework for interpolation of values for other combinations of wall and floor thicknesses. The f factors for these problems were calculated and plotted on the same graph as the 10′ wall height to see if a pattern developed.

Figures 6 and 7 show all of these points plotted at $z = 0$ and $z = c/4$, respectively. It is apparent then that the f factors are independent of the wall height and are only a function of the b/a and c/a ratios. A separate graph for each tank of different dimensions is therefore not necessary as was required for the moment coefficients of the three plate problem.

Since the f factor appears to lie in a certain area, it is not necessary to plot as many points as was done for b/a and c/a equal to 2.0.

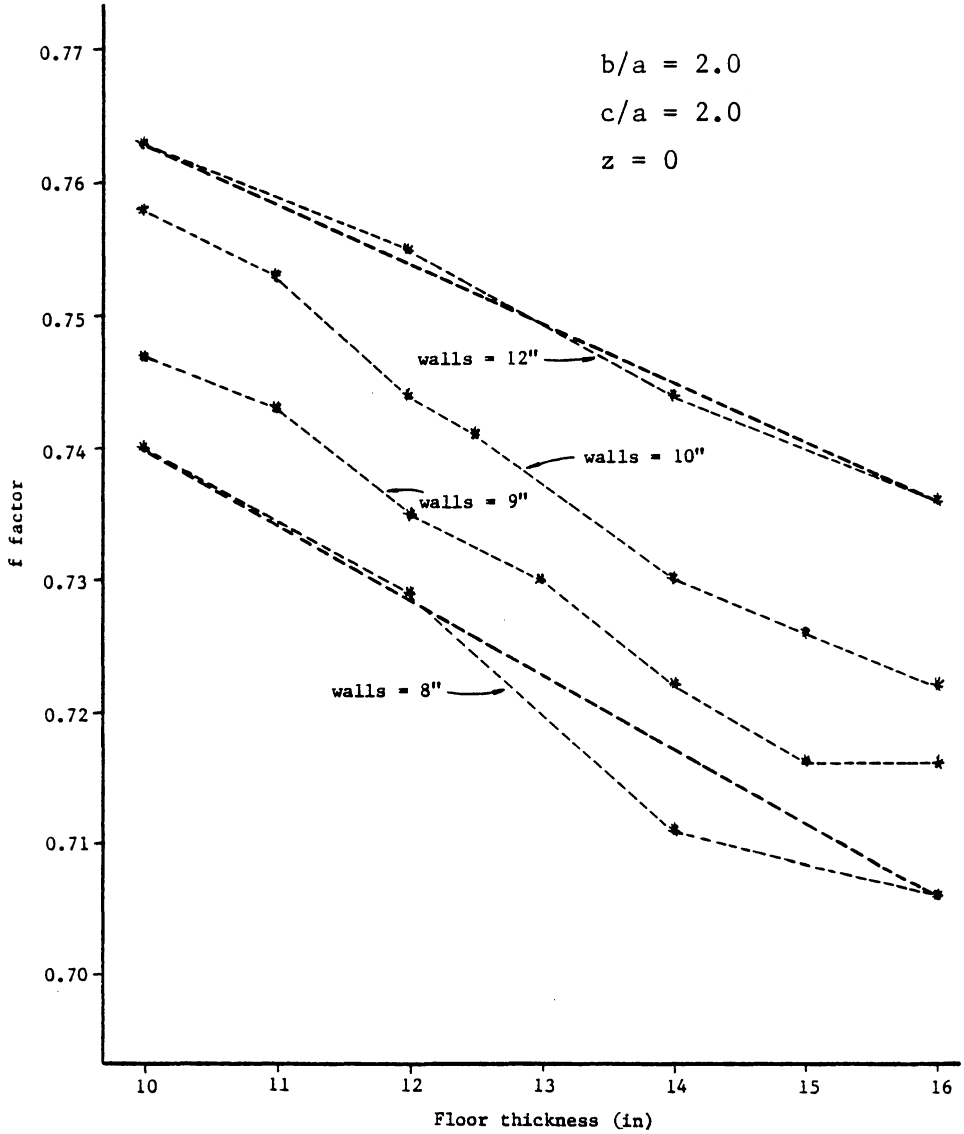


FIGURE 4: Floor stiffness factor, 10' height

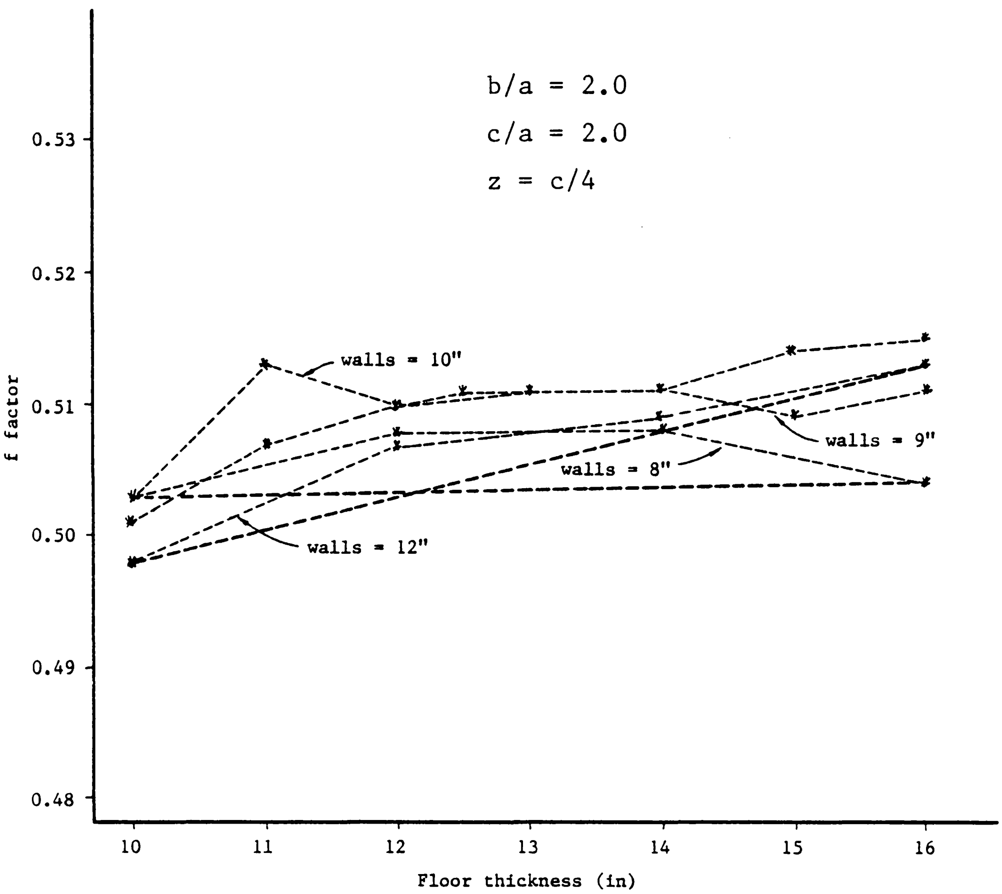


FIGURE 5: Floor stiffness factor, 10' height

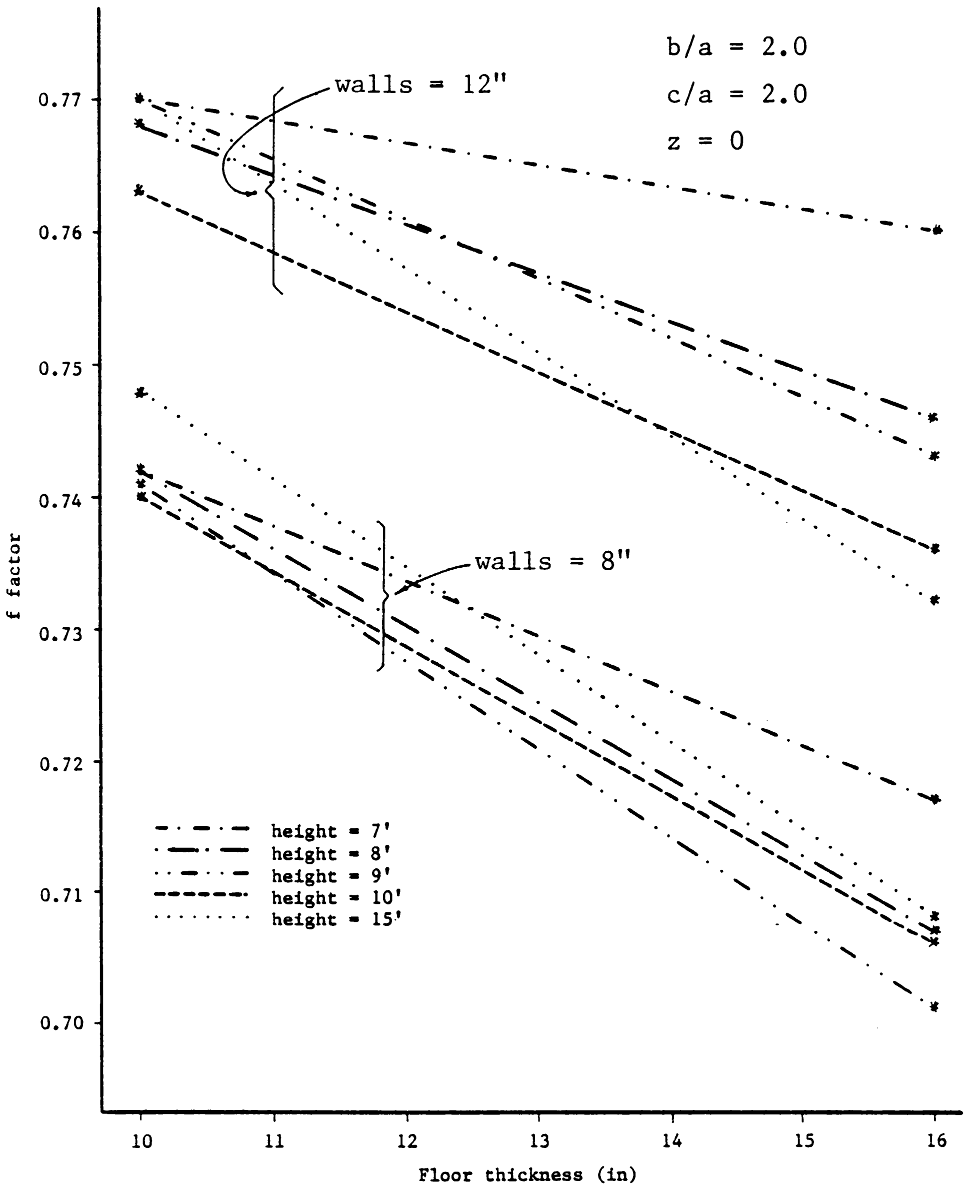


FIGURE 6: Floor stiffness factor, various heights

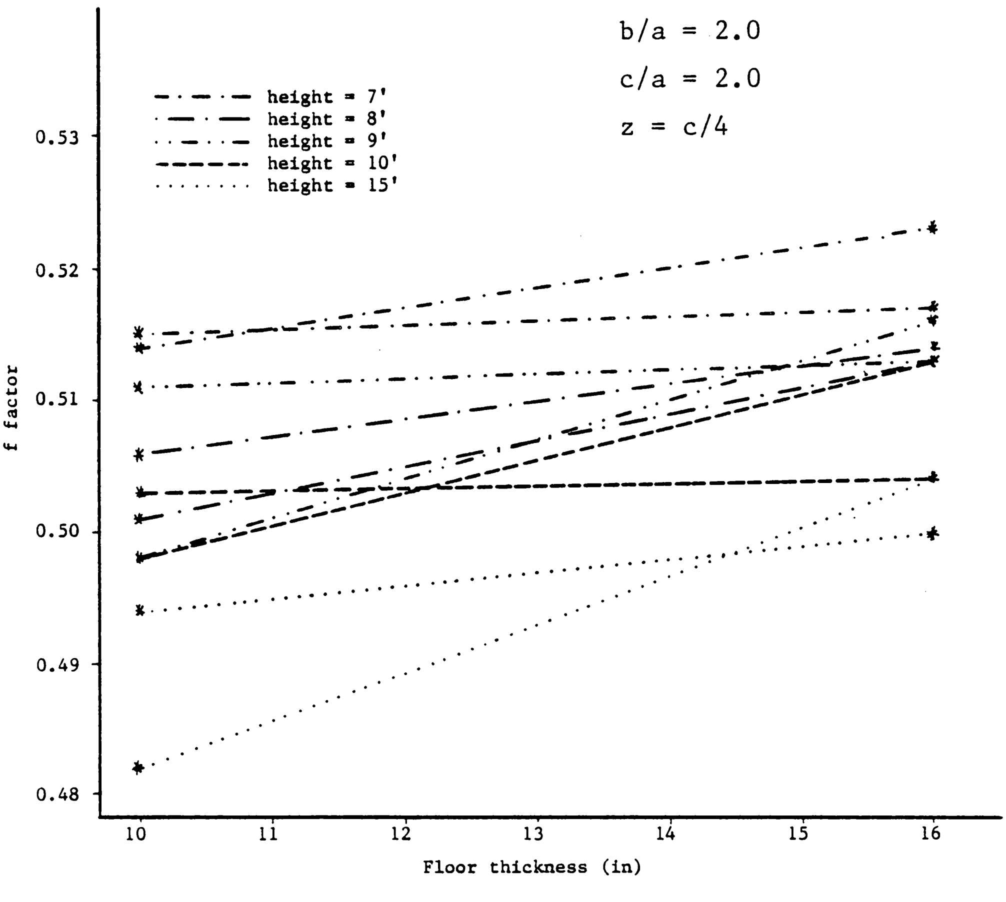


FIGURE 7: Floor stiffness factor, various heights

Therefore, the wall thicknesses for further calculations only include 8", 10" and 12" and the floor thicknesses include 10", 12", 14" and 16". This provides a sufficient number of points so that interpolation can be used for other combinations of wall and floor thicknesses.

Appendix 2 contains tables of the f factor for b/a = 1.0 and c/a equal to 1.0, 2.0 and 3.0. Figures 4 and 5 with b/a and c/a equal to 2.0 are reproduced in Appendix 2 so that all the f factor graphs are located in one place. It should be noted that some of the f factors are negative, especially in the short walls of rectangular tanks. Although this is unconventional, this value will provide the solution given by the finite element program. The floor distribution factor is found by dividing the stiffness of the floor (negative) by the sum of the absolute values of the stiffnesses. The wall distribution factor is the absolute value of the floor added to one (1), so that the total of the two factors is unity. It is believed that the rotation of the vertical corner in a rectangular tank provides an unnatural stiffness to the short wall of a tank with a long side.

It is now possible for a designer, without the finite element program, to perform the moment distribution process and calculate the critical vertical moments at a joint between the walls and floor at the center and the quarter points of the tank while providing for the interaction of the plates as a system. The tables in Appendix 1 provide the required fixed-end moment and the graphs in Appendix 2 provide the means for determining the appropriate f factors.

Example Problem

In order to demonstrate the use of the tables listed in the Appendices and the procedure to calculate the vertical moments in a rectangular tank, consider the following problem:

$b/a = 1.0$ wall thickness = 9"

$c/a = 3.0$ floor thickness = 13"

hydrostatic loading from the interior with $\omega = 80.0$ pcf

height = 12′

Determine the balanced moments at $z = 0$, $z = c/4$, $y = 0$ and $y = b/4$. Looking first at $z = 0$:

$$S_w = \frac{4(3000)(12)(9^3)}{12(12)(12)} = 60{,}750 \text{ k-in}$$

From Appendix 2, Figure A9, $f = 0.997$ so that

$$S_f = \frac{0.997(4)(3000)(12)(13^3)}{144(12)} = 182{,}534 \text{ k-in}$$

and

$$r_w = \frac{60{,}750}{243{,}284} = 0.250$$

$$r_f = \frac{182{,}534}{243{,}284} = 0.750$$

The floor fixed-moment from Table A1 is, at the midspan of the long wall,

$$FEM_f = 12(0.180) = 2.16 \text{ k-ft/ft}$$

and from Table A2, the wall fixed-end moment is

$$FEM_w = \frac{-0.132(80)(12^3)}{1000} = -18.25 \text{ k-ft/ft}$$

The balanced moment is then found by the moment distribution process as follows

(at z = 0)

-18.25	0.250	0.250	18.25
3.19	0.750	0.750	-5.10
0.45	-2.16	2.16	1.20
0.06	7.65	-15.31	0.17
-14.55	9.57	-4.79	14.52
	-1.79	3.59	
	1.34	-0.67	
	-0.25	0.50	
	0.19	-14.52	
	14.55		

Following the same procedure at z = c/4, from Figure A10 in Appendix 2 f = 0.789 so that

$$S_f = \frac{0.789(4)(3000)(12)(13^3)}{144(12)} = 144{,}453 \text{ k-in}$$

and

$$r_w = \frac{60{,}750}{205{,}203} = 0.296$$

$$r_f = \frac{144{,}453}{205{,}203} = 0.704$$

From Table A1, the floor fixed-end moment is

$$FEM_f = 12(0.183) = 2.20 \text{ k-ft/ft}$$

and the wall fixed-end moment from Table A2 is

$$FEM_w = \frac{-0.102(80)(12^3)}{1000} = -14.10 \text{ k-ft/ft}$$

The moment distribution yields

-14.10	0.296	0.296	14.10	(at z = c/4)
3.13	0.704	0.704	-4.82	
0.39	-2.20	2.20	1.10	
0.05	5.74	-11.48	0.14	
-10.53	7.43	-3.72	10.52	
	-1.31	2.62		
	0.92	-0.46		
	-0.16	0.32		
	0.11	-10.52		
	10.53			

Continuing on to the short wall at y = 0, from Table A7 in Appendix 2, f = -0.060 so that

$$S_f = \frac{-0.06(4)(3000)(12)(13^3)}{36(12)(12)} = -3{,}662 \text{ k-in}$$

and

$$r_f = \frac{-3{,}662}{64{,}412} = -0.057$$

$$r_w = 1 + 0.057 = 1.057$$

From Table A1, the floor fixed-end moment at y = 0 is

$$FEM_f = 12(0.146) = 1.75 \text{ k-ft/ft}$$

and from Table A2, the wall fixed-end moment is

$$FEM_w = \frac{-0.020(80)(12^3)}{1000} = -2.76 \text{ k-ft/ft}$$

Subsequent moment distribution yields

(at y = 0)

	1.057	1.057	
-2.76			2.76
4.90	-0.057	-0.057	-4.77
2.14	-1.75	1.75	-0.14
	-0.13	0.26	-2.15
	-0.26	0.13	
	-2.14	0.01	
		2.15	

Finally, calculating the balanced moment at y = b/4 and using Figure A8, f = -0.37, so that

$$S_f = \frac{-0.37(4)(3000)(12)(13^3)}{432(12)} = -22{,}580 \text{ k-in}$$

and

$$r_f = \frac{-22{,}580}{83{,}330} = -0.271$$

$$r_w = 1 + 0.271 = 1.271$$

From Tables A1 and A2 then

$$FEM_f = 12(0.094) = 1.13 \text{ k-ft/ft}$$

$$FEM_w = \frac{-0.012(80)(12^3)}{1000} = -1.66 \text{ k-ft/ft}$$

So that moment distribution yields

	1.271	1.271		
-1.66			1.66	(at y = b/4)
4.03	-0.271	-0.271	-3.55	
0.08	-1.13	1.13	-0.55	
2.45	-0.38	0.76	-2.44	
	-0.86	0.43		
	-0.06	0.12		
	-0.02	2.44		
	-2.45			

As a comparison, this problem was checked against the finite element program. The moment values obtained along the long wall at z = 0 and z = c/4 were found to be 14.48 and 10.54 kip-ft/ft, respectively. These values are very close to the values obtained by the moment distribution procedure. The values at y = 0 and y = b/4 were 2.63 and 2.92 kip-ft/ft, respectively. The moment distribution method does not correlate quite as well in the short wall, although the values are reasonably close. It should be noted that the values on the short wall graphs are significantly more varied in magnitude than the long wall graphs. Consequently, it is more difficult to accurately determine the f factor from the graphs for the short walls. Unfortunately, the final moment value is sensitive to the f factor so an allowance should be considered to accomodate this fact.

A second example was performed following the same procedure except that the tank was loaded from the exterior. The only change was that the wall fixed-end moments had the opposite signs; the same f factors were used. Correlation with finite element program was excellent in the long wall. At z = 0, moment distribution obtained 15.33 k-ft/ft and the program obtained 15.21 k-ft/ft and at z = c/4, 11.49 k-ft/ft compared to 11.58 k-ft/ft.

The values in the short wall did not match up at all. At y = 0, moment distribution obtained 1.66 k-ft/ft and the program obtained -0.051 k-ft/ft and at y = b/4, 0.88 k-ft/ft compared to -1.15 k-ft/ft. A conclusion that should be drawn out of these examples is that the moment values in the long walls can be determined quite accurately but the determination of balanced moments in the short walls should be carried out with some discretion. A possible explanation for the discrepency in the short wall might be that the rotation of the long wall makes the short wall appear overly stiff.

VII. CONCLUSIONS

This paper has developed a finite element program that is capable of analyzing one quarter of a rectangular tank and determining the horizontal and vertical bending moments at a number of locations. The triangular and uniform loadings incorporated into this program can be external or internal and can be the full or partial height of the tank. It is also possible to handle tapered wall sections. By being able to analyze a quarter of the tank as a whole, it is possible to permit joint rotations and allow the natural balancing of moments so that the interaction of the plates can be properly represented.

In addition to the capability of handling three orthogonal plates, any one or two plate system can be analyzed provided the two plates are perpendicular to each other. This aided in the development of the fixed-end moment tables.

The secondary objective of the paper was to calculate moment values at the joints between the plates in a rectangular tank. It was not practical, however, to develop a set of moment coefficients for this problem because the moments were not a constant times the specific weight and the height cubed as was possible with the one and two plate problems. An alternate solution was sought by paralleling the moment distribution method that is used for beam structures. Fortunately, this method eliminated the dependence of the moment values on the height of a tank with given proportions. This allowed a small group of tables to handle a wide variety of tank sizes.

The key assumption that was made in developing this program was that the twisting resistance perpendicular to the plane of the plate is infinite and can be eliminated as a boundary condition. This permitted the development and use of a five degree of freedom element carrying along the sixth degree of freedom as a dummy to properly provide for coordinate transformations.

A shortcoming of this program might be that it does not provide for the slope continuity between element edges. However, the merits of this element have been proven.

No consideration has been given to the horizontal moments in the walls of the tank, which can become large at the top edge of the wall-to-wall joints in rectangular tanks, or to the shear forces. These moments and shears were calculated by the finite element program but were not covered in this paper because they are also dependent upon the height of the tank.

The moment distribution procedure developed in this paper as a design aid provides very satisfactory results for the long walls in a rectangular tank but less accurate answers in the short walls. This might be attributed to an overstiffening effect of the short wall from the long wall.

Future work would include developing a similar procedure for horizontal moments and examining the shearing forces in a rectangular tank.

VIII. BIBLIOGRAPHY

1. "Rectangular Concrete Tanks," Bulletin ST-63, Structural Bureau, Portland Cement Association, 1969.

2. Jofriet, Jan. C., "Design of Rectangular Concrete Tank Walls," Journal, American Concrete Institute, Vol. 72, July, 1975, pp. 329-332.

3. Davies, J. D., Y. K. Cheung, "Bending Moments in Long Walled Tanks," Journal, American Concrete Institute, Vol. 64, October, 1967, pp. 685-690.

4. Cheung, Y. K., and J. D. Davies, "Analysis of Rectangular Tanks--Use of Finite Element Technique," Concrete, Vol. 1, May, 1967, pp. 169-174.

5. Davies, J. D., "Influence of Support Conditions on the Behavior of Long Rectangular Tanks," Journal, American Concrete Institute, Vol. 59, April, 1962, pp. 601-608.

6. Davies, J. D., "Bending Moments in Long Rectangular Tanks on Elastic Foundations," Concrete and Constructional Engineering, Vol. 56, No. 10, October, 1961, pp. 335-338.

7. Davies, J. D., "Bending Moments in Edge Supported Square Concrete Tanks," The Structural Engineer, Vol. 40, May, 1962, pp. 161-166.

8. Davies, J. D., "Analysis of Long Rectangular Tanks Resting on Flat Rigid Supports," Journal, American Concrete Institute, Vol. 60, April, 1963, pp. 487-499.

9. Davies, J. D., "Bending Moments in Square Concrete Tanks Resting on Flat Rigid Supports," The Structural Engineer, Vol. 41, December, 1963, pp. 407-410.

10. Davies, J. D., and Long, J. E., "Behavior of Square Tanks on Elastic Foundations," Journal of the Engineering Mechanics Division, ASCE, Vol. 94, No. EM3, Proc. Paper 5985, June, 1968, pp. 733-772.

11. Brenneman, James, "Analysis of Structures Idealized as Rectangular Elements in Combined Flexure and Extension," Master's Thesis presented at Virginia Polytechnic Institute, Blacksburg, VA, 1969.

12. Beck, R. L., "Analysis of Short-Walled Rectangular Concrete Tanks," Master's Thesis presented at Virginia Polytechnic Institute, Blacksburg, VA, 1972.

13. Wilby, C. A., "Structural Analysis of Reinforced Concrete Tanks," Journal of the Structural Division, ASCE, Vol. 103, May, 1977, pp. 989-1004.

14. Lightfoot, E., and A. Ghali, "The Analysis of Rectangular Concrete Tanks," Proceedings, 50th Anniversary Conference, Institution of Structural Engineers, 1958.

15. Moody, W. T., "Moments and Reactions for Rectangular Plates," Engineering Monograph, No. 27, US Department of the Interior, Bureau of Reclamation, Denver, 1970.

16. Clough, R. W., and J. L. Tocher, "Finite Element Stiffness Matrices for Analysis of Plate Bending," presented at the Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, October, 1965.

17. Desai, C. S., Elementary Finite Element Method, Prentice-Hall Inc., Englewood Cliffs, NJ, 1979, pp. 47-50.

18. Basham, K. D., "A Comparative Investigation of Stiffness Storage and Solution Algorithms Used in Structural Analysis," Master's Thesis presented at Virginia Polytechnic Institute, Blacksburg, VA, 1982.

19. Zienkiewicz, O. C., and Y. K. Cheung, The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Co., London, 1967.

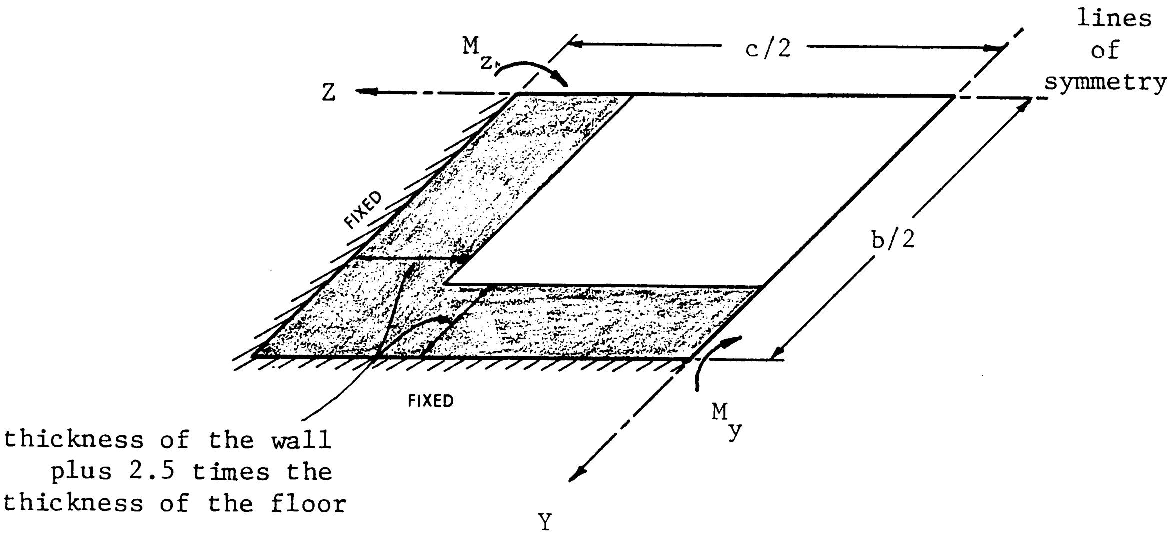
20. Rockey, K. C., and H. R. Evans, "A Finite Element Solution for Folded Plate Structures," presented at the International Conference on Space Structures, held at the University of Surrey, September, 1966.

21. Bauverlag, R. B., Tables for the Analysis of Plates, Slabs and Diaphragms, Library of Congress, CAT # 68-25531, 1969.

IX. APPENDICES

Appendix 1

TABLE A1: Fixed-end Moments - Floor Plate



Moment values are in units of ft-kips/ft per foot of wall height.

Fixed-end moment = (coefficient from table)(height of the wall,ft) ft-kips/ft

b/a = 1.0 Short Wall

		y = 0				y = b/4			
		floor thickness ,in							
c/a	walls	10	12	14	16	10	12	14	16
1.0	8	0.146	0.165	0.180	0.191	0.106	0.114	0.119	0.123
	9	0.169	0.189	0.205	0.217	0.121	0.129	0.135	0.139
	10	0.192	0.214	0.231	0.244	0.138	0.145	0.151	0.156
	12	0.242	0.265	0.284	0.297	0.169	0.177	0.184	0.188
2.0	8	0.124	0.139	0.149	0.157	0.089	0.094	0.097	0.099
	9	0.143	0.159	0.170	0.178	0.101	0.106	0.109	0.111
	10	0.163	0.179	0.191	0.200	0.114	0.119	0.122	0.124
	12	0.205	0.221	0.234	0.243	0.140	0.144	0.147	0.149
3.0	8	0.112	0.124	0.132	0.138	0.078	0.082	0.084	0.085
	9	0.128	0.141	0.150	0.157	0.089	0.093	0.095	0.096
	10	0.146	0.159	0.169	0.175	0.100	0.104	0.106	0.107
	12	0.182	0.196	0.206	0.212	0.123	0.126	0.128	0.128

TABLE A1 (cont.)

b/a = 1.0 Long Wall

		z = 0				z = c/4			
		floor thickness, in							
c/a	walls	10	12	14	16	10	12	14	16
1.0	8	0.146	0.165	0.180	0.191	0.106	0.114	0.119	0.123
	9	0.169	0.189	0.205	0.217	0.121	0.129	0.135	0.139
	10	0.192	0.214	0.231	0.244	0.138	0.145	0.151	0.156
	12	0.242	0.265	0.284	0.297	0.169	0.177	0.184	0.188
2.0	8	0.139	0.160	0.182	0.202	0.133	0.150	0.165	0.178
	9	0.161	0.185	0.209	0.232	0.154	0.172	0.189	0.202
	10	0.184	0.211	0.237	0.263	0.175	0.195	0.213	0.227
	12	0.234	0.263	0.297	0.327	0.219	0.243	0.263	0.279
3.0	8	0.129	0.148	0.165	0.183	0.130	0.150	0.166	0.181
	9	0.149	0.170	0.190	0.209	0.150	0.173	0.191	0.207
	10	0.170	0.193	0.215	0.237	0.172	0.196	0.215	0.234
	12	0.219	0.243	0.269	0.294	0.216	0.245	0.268	0.288

b/a = 2.0

		z = 0				z = c/4			
		floor thickness, in							
c/a	walls	10	12	14	16	10	12	14	16
2.0	8	0.144	0.164	0.191	0.214	0.130	0.144	0.162	0.175
	9	0.167	0.191	0.221	0.246	0.150	0.167	0.186	0.200
	10	0.190	0.219	0.251	0.278	0.170	0.190	0.210	0.225
	12	0.239	0.279	0.315	0.345	0.212	0.238	0.260	0.276

TABLE A2: Vertical Moments for Two Plate Problem
Clamped Bottom Edge

Refer to Figure 3 for the appropriate coordinate system

Moment = (coefficient from table) * (specific weight of fluid) * (height of tank)^3

Negative sign indicates tension on the loaded side.

b/a = 1.0

c/a	x/a	y = 0	y = b/4	z = 0	z = c/4
	0	0	0	0	0
	1/4	+0.001	-0.002	+0.009	+0.008
3.0	1/2	+0.009	+0.004	+0.003	+0.006
	3/4	+0.012	+0.009	-0.038	-0.023
	1	-0.020	-0.012	-0.132	-0.102
	0	0	0	0	0
	1/4	+0.001	-0.002	+0.011	+0.008
2.5	1/2	+0.009	+0.005	+0.009	+0.009
	3/4	+0.011	+0.009	-0.026	-0.015
	1	-0.022	-0.015	-0.116	-0.087
	0	0	0	0	0
	1/4	+0.001	-0.002	+0.012	+0.007
2.0	1/2	+0.009	+0.004	+0.014	+0.011
	3/4	+0.011	+0.008	-0.012	-0.006
	1	-0.024	-0.015	-0.094	-0.068
	0	0	0	0	0
	1/4	+0.002	-0.001	+0.010	+0.005
1.5	1/2	+0.010	+0.005	+0.016	+0.010
	3/4	+0.009	+0.007	+0.001	+0.002
	1	-0.029	-0.019	-0.066	-0.047
	0	0	0	0	0
	1/4	+0.005	+0.002	+0.005	+0.002
1.0	1/2	+0.011	+0.006	+0.011	+0.006
	3/4	+0.009	+0.006	+0.009	+0.006
	1	-0.035	-0.024	-0.035	-0.024

TABLE A2 (cont.)

b/a = 1.5

c/a	x/a	y = 0	y = b/4	z = 0	z = c/4
	0	0	0	0	0
	1/4	+0.009	+0.003	+0.010	+0.008
3.0	1/2	+0.017	+0.011	+0.004	+0.006
	3/4	+0.007	+0.007	-0.036	-0.021
	1	-0.052	-0.035	-0.129	-0.098
	0	0	0	0	0
	1/4	+0.009	+0.003	+0.012	+0.008
2.5	1/2	+0.017	+0.011	+0.010	+0.009
	3/4	+0.006	+0.006	-0.024	-0.013
	1	-0.053	-0.036	-0.112	-0.082
	0	0	0	0	0
	1/4	+0.009	+0.004	+0.012	+0.007
2.0	1/2	+0.016	+0.010	+0.015	+0.011
	3/4	+0.005	+0.005	-0.010	-0.004
	1	-0.055	-0.038	-0.089	-0.063
	0	0	0	0	0
	1/4	+0.009	+0.004	+0.009	+0.004
1.5	1/2	+0.016	+0.010	+0.016	+0.010
	3/4	+0.003	+0.004	+0.003	+0.004
	1	-0.060	-0.041	-0.060	-0.041

b/a = 2.0

c/a	x/a	y = 0	y = b/4	z = 0	z = c/4
	0	0	0	0	0
	1/4	+0.002	+0.006	+0.010	+0.007
3.0	1/2	+0.017	+0.012	+0.004	+0.007
	3/4	-0.006	-0.001	-0.035	-0.019
	1	-0.082	-0.056	-0.127	-0.095
	0	0	0	0	0
	1/4	+0.012	+0.006	+0.012	+0.007
2.5	1/2	+0.016	+0.012	+0.011	+0.010
	3/4	-0.007	-0.001	-0.022	-0.011
	1	-0.083	-0.057	-0.109	-0.079
	0	0	0	0	0
	1/4	+0.012	+0.006	+0.012	+0.006
2.0	1/2	+0.016	+0.011	+0.016	+0.011
	3/4	-0.008	-0.002	-0.008	-0.002
	1	-0.086	-0.059	-0.086	-0.059

Appendix 2

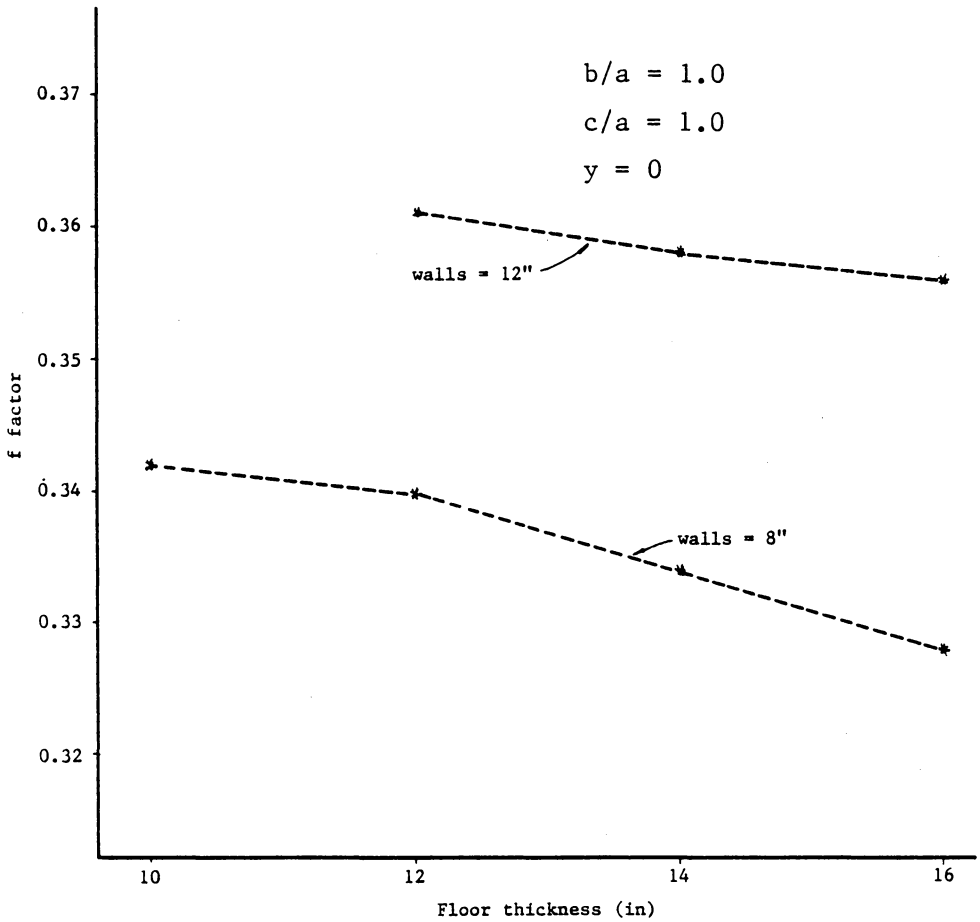


FIGURE A1: Floor stiffness factor

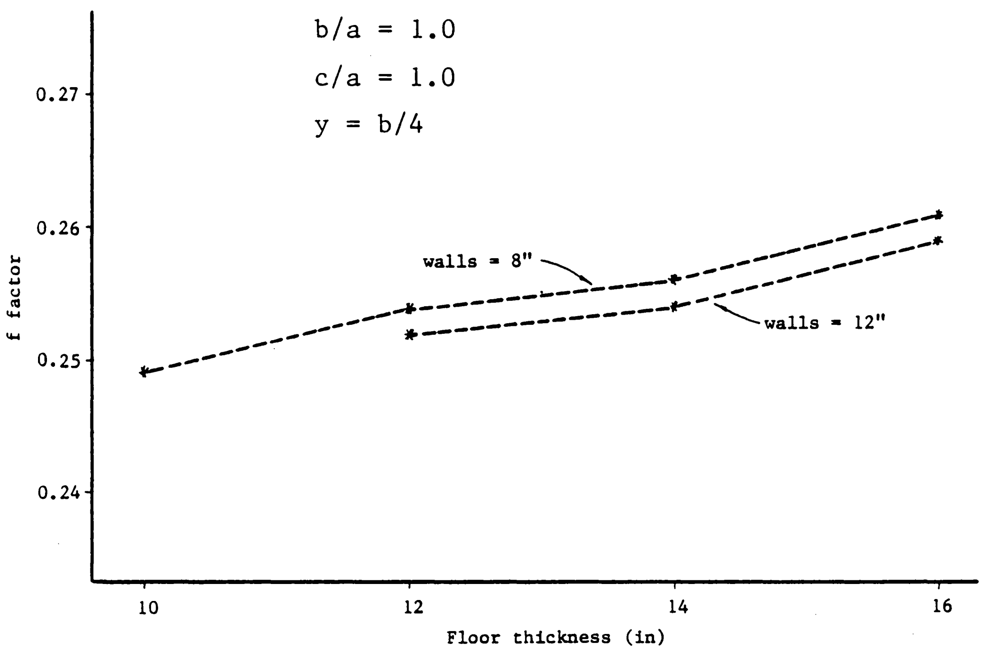


FIGURE A2: Floor stiffness factor

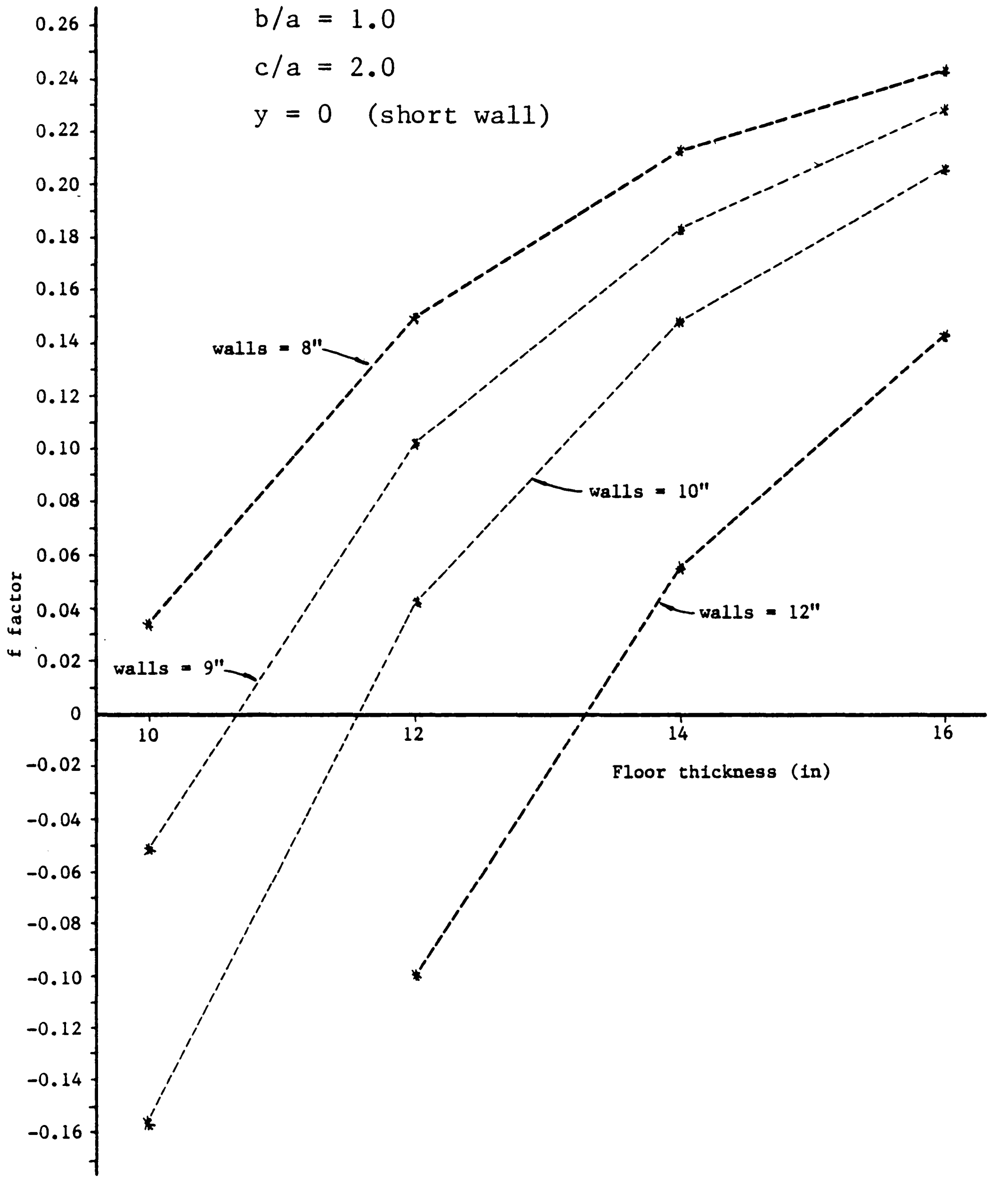


FIGURE A3: Floor stiffness factor

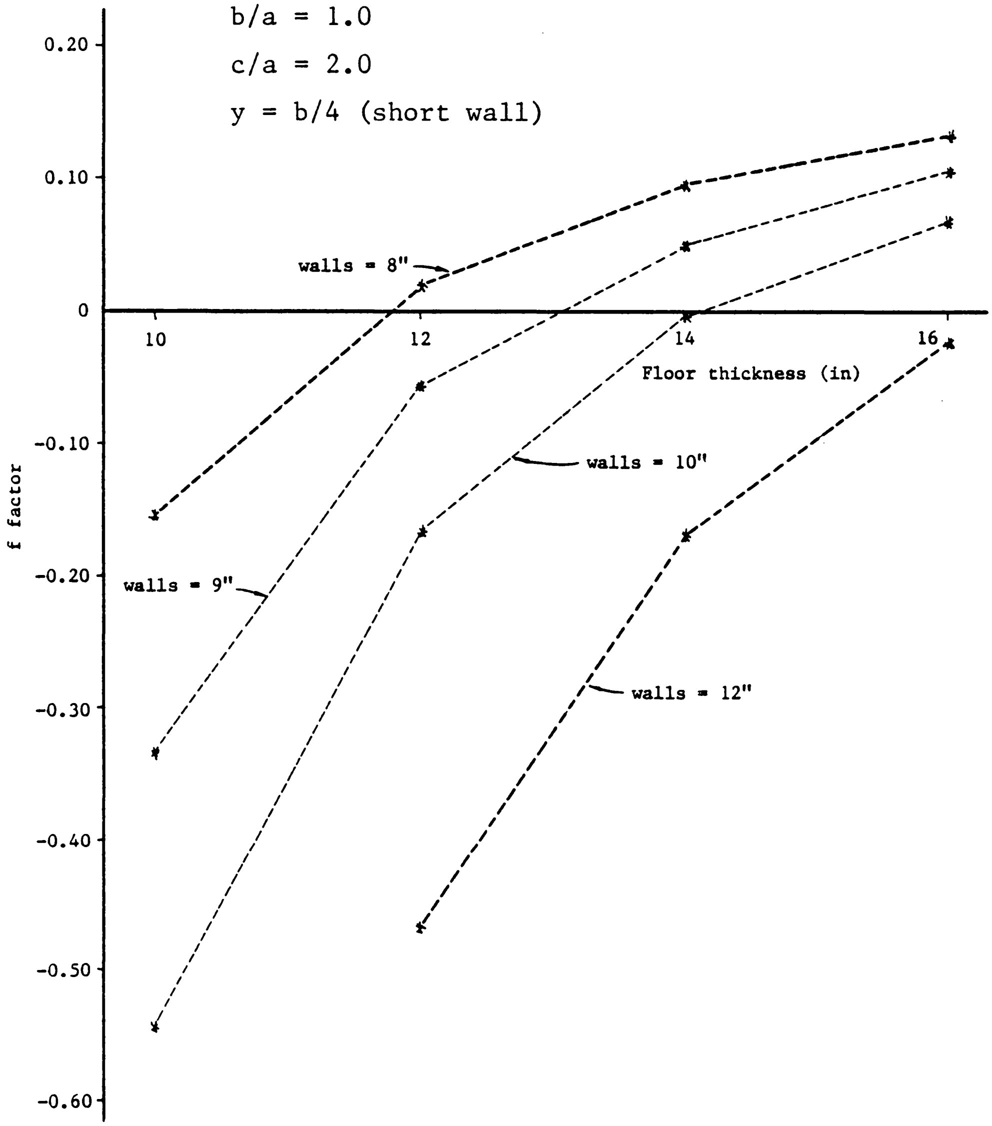


FIGURE A4: Floor stiffness factor

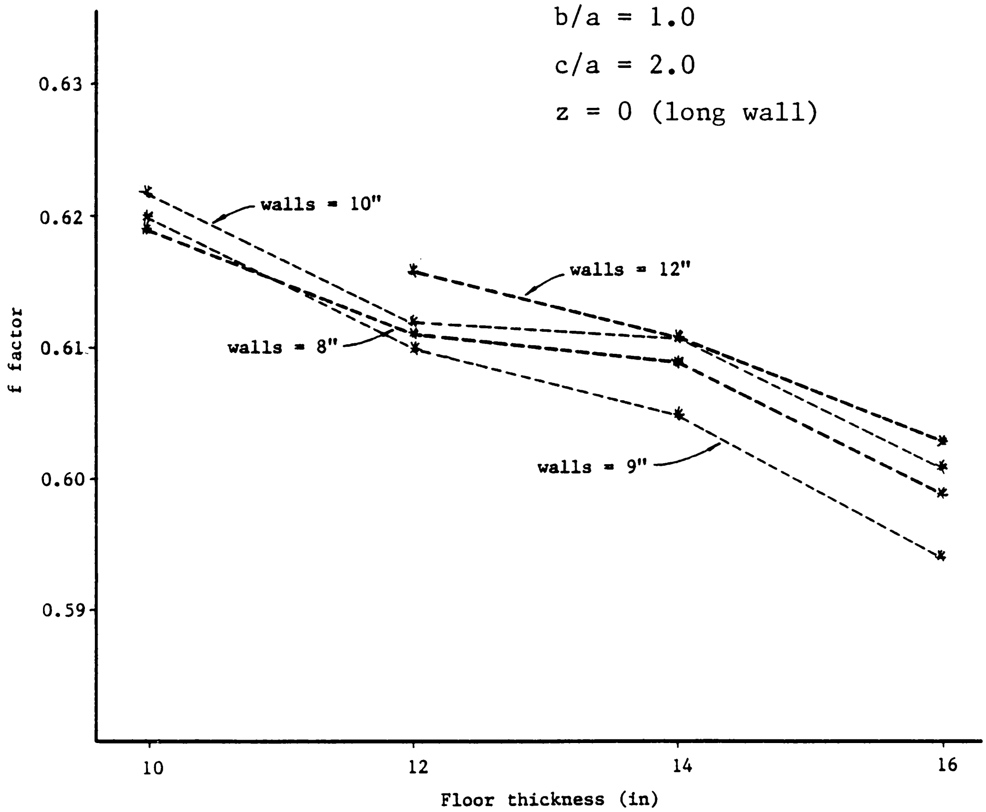


FIGURE A5: Floor stiffness factor

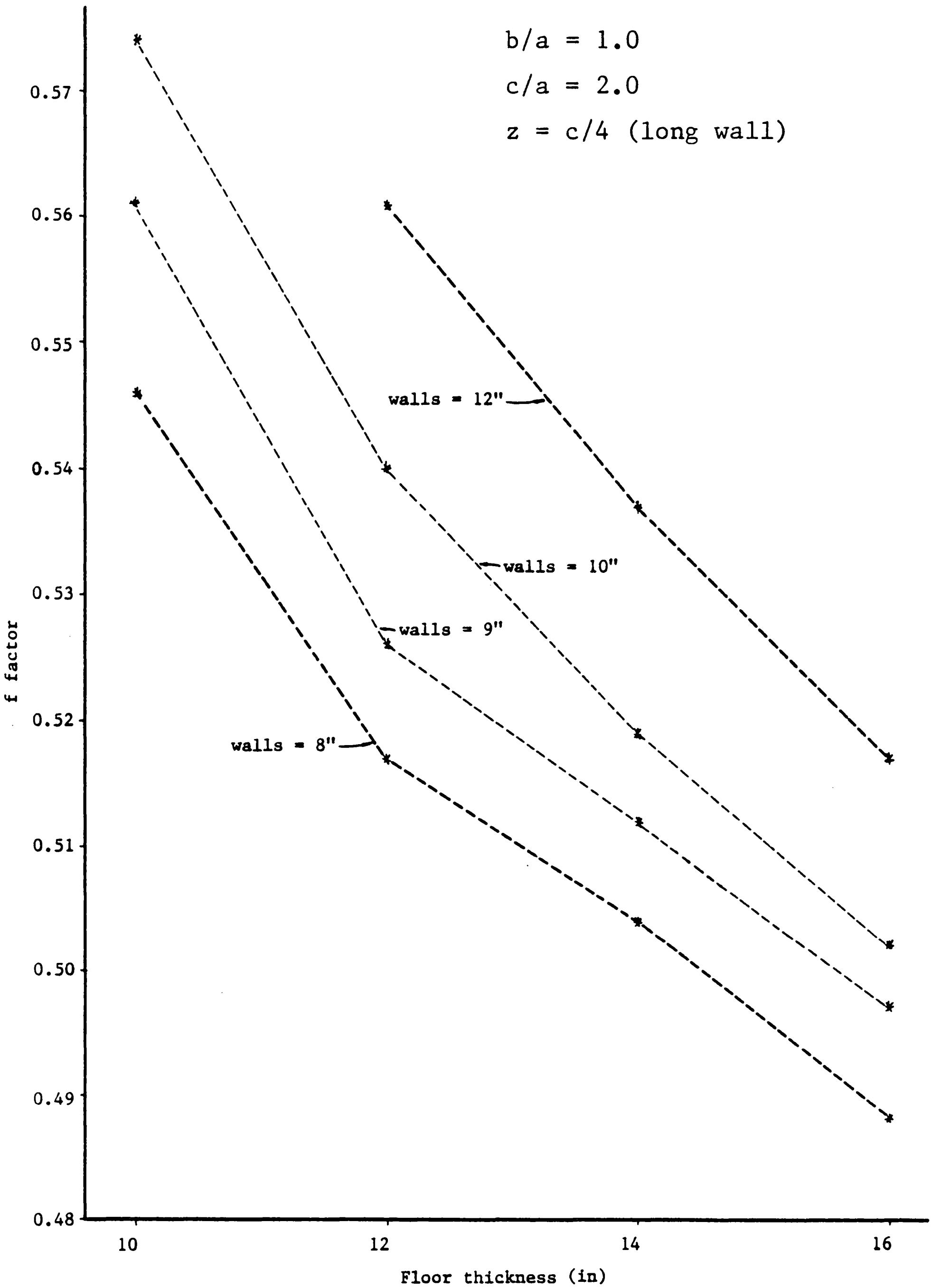


FIGURE A6: Floor stiffness factor

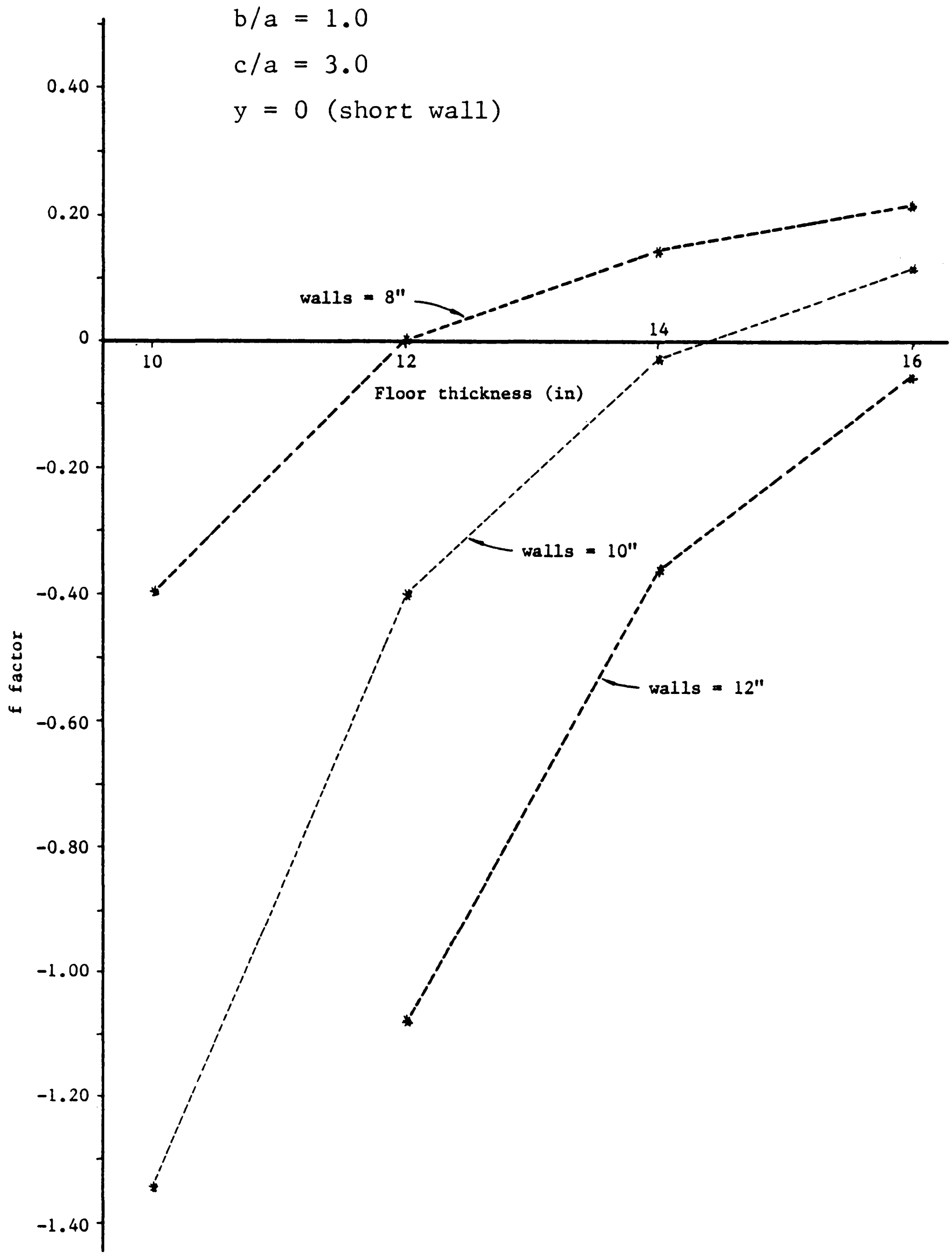


FIGURE A7: Floor stiffness factor

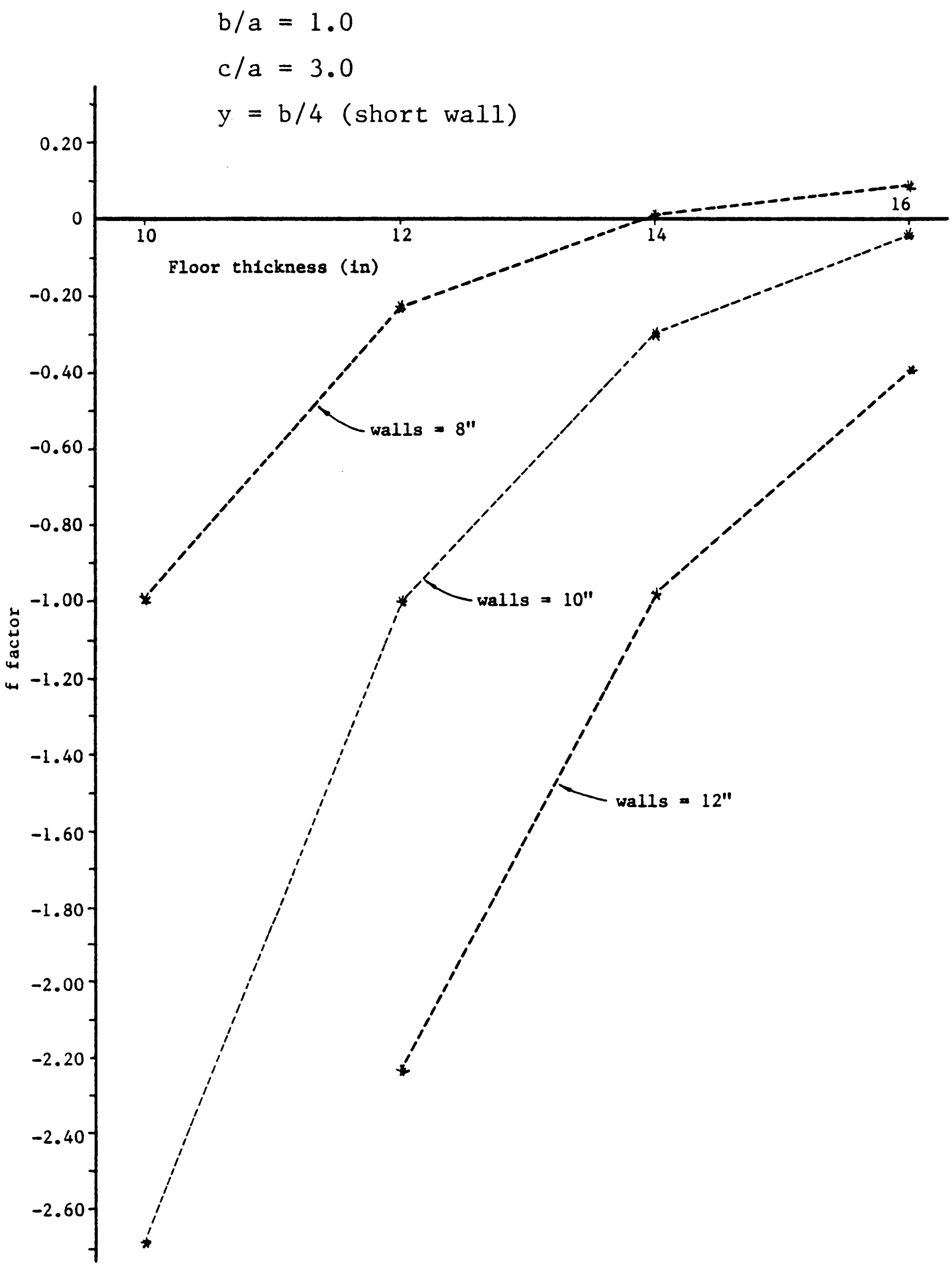


FIGURE A8: Floor stiffness factor

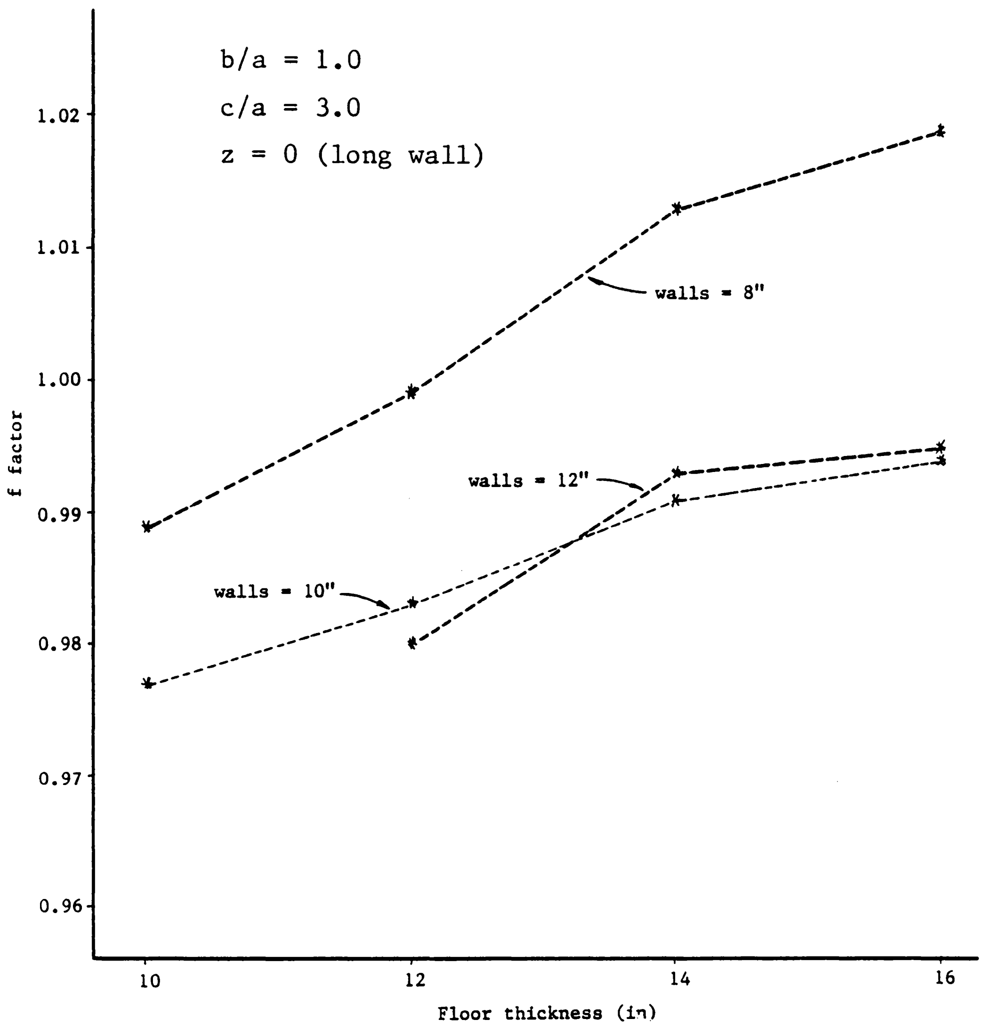


FIGURE A9: Floor stiffness factor

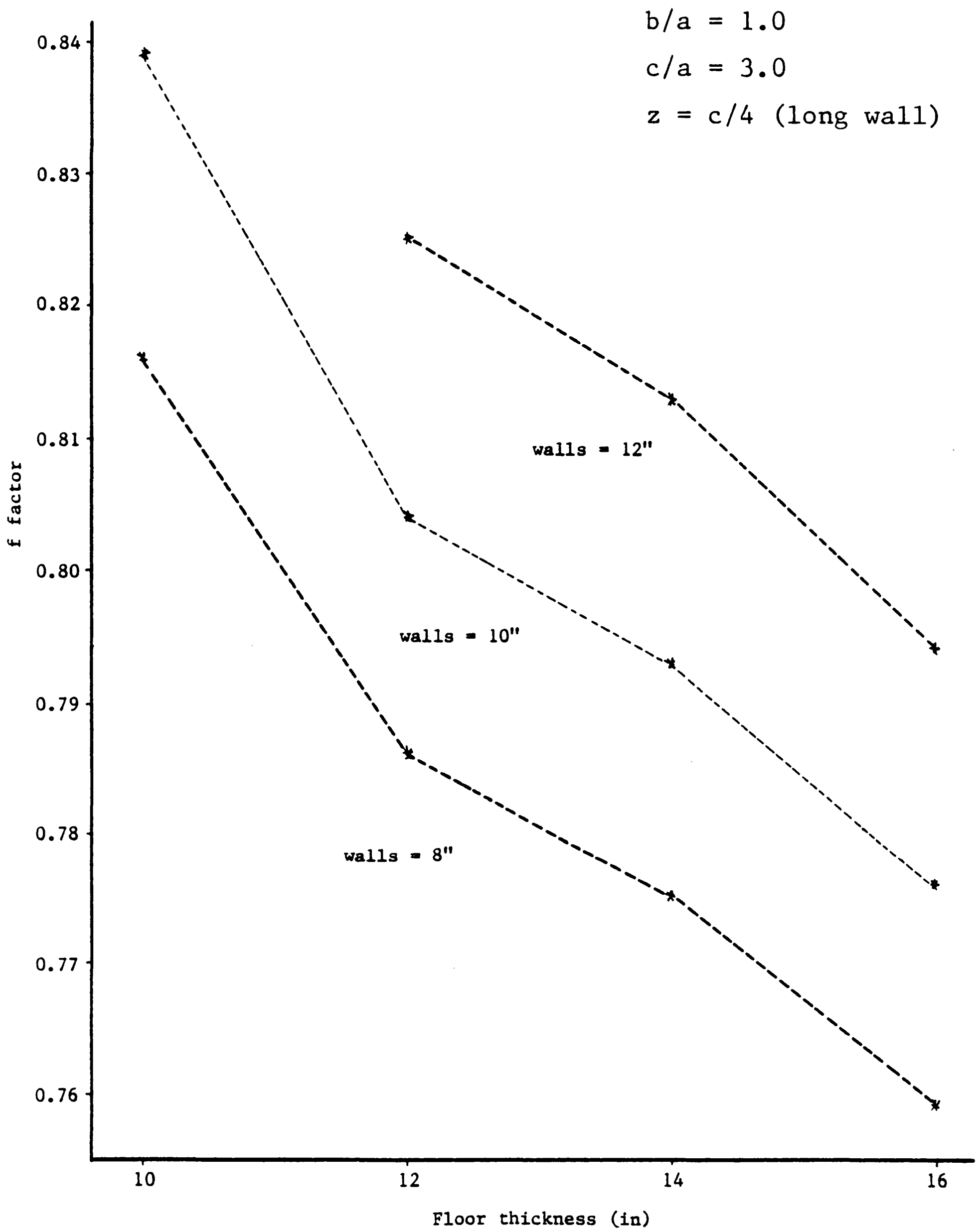


FIGURE A10: Floor stiffness factor

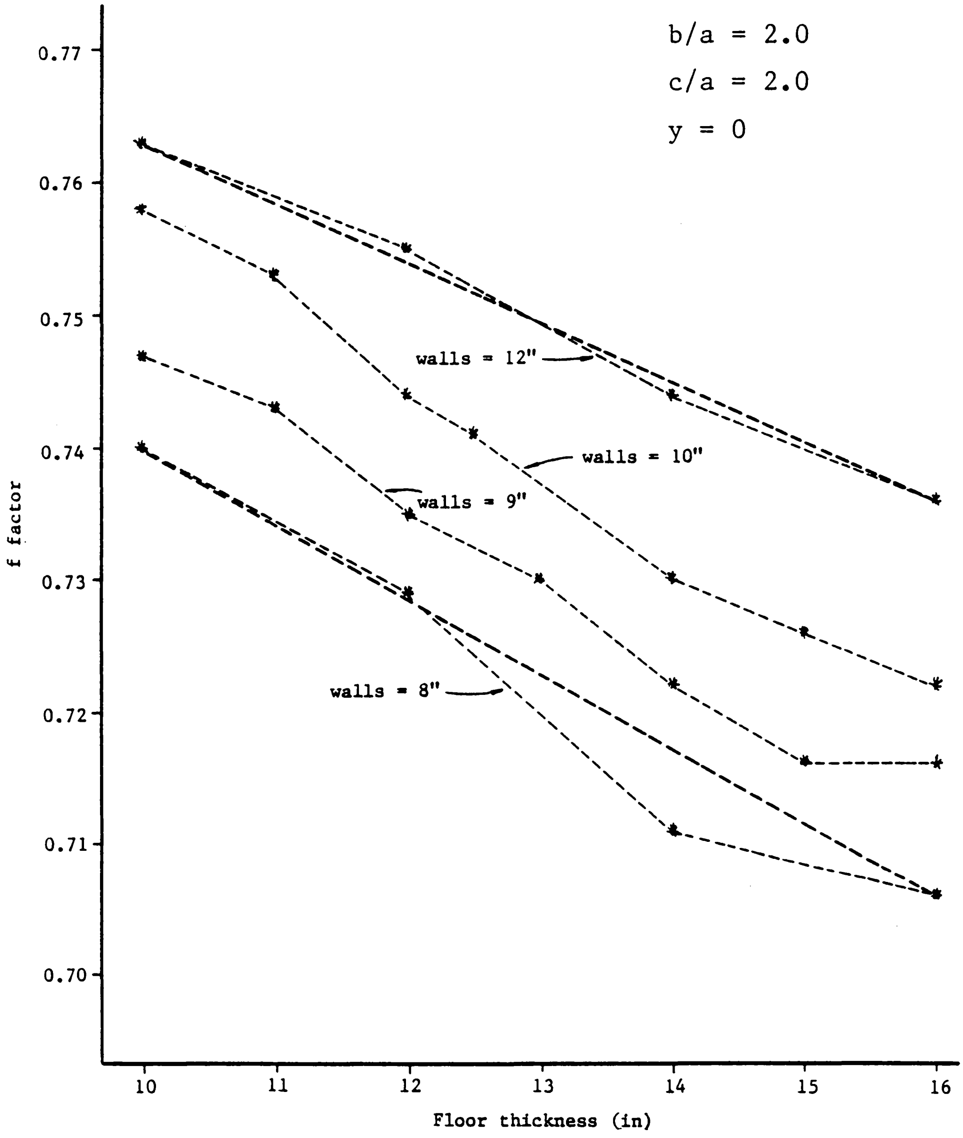


FIGURE A11: Floor stiffness factor

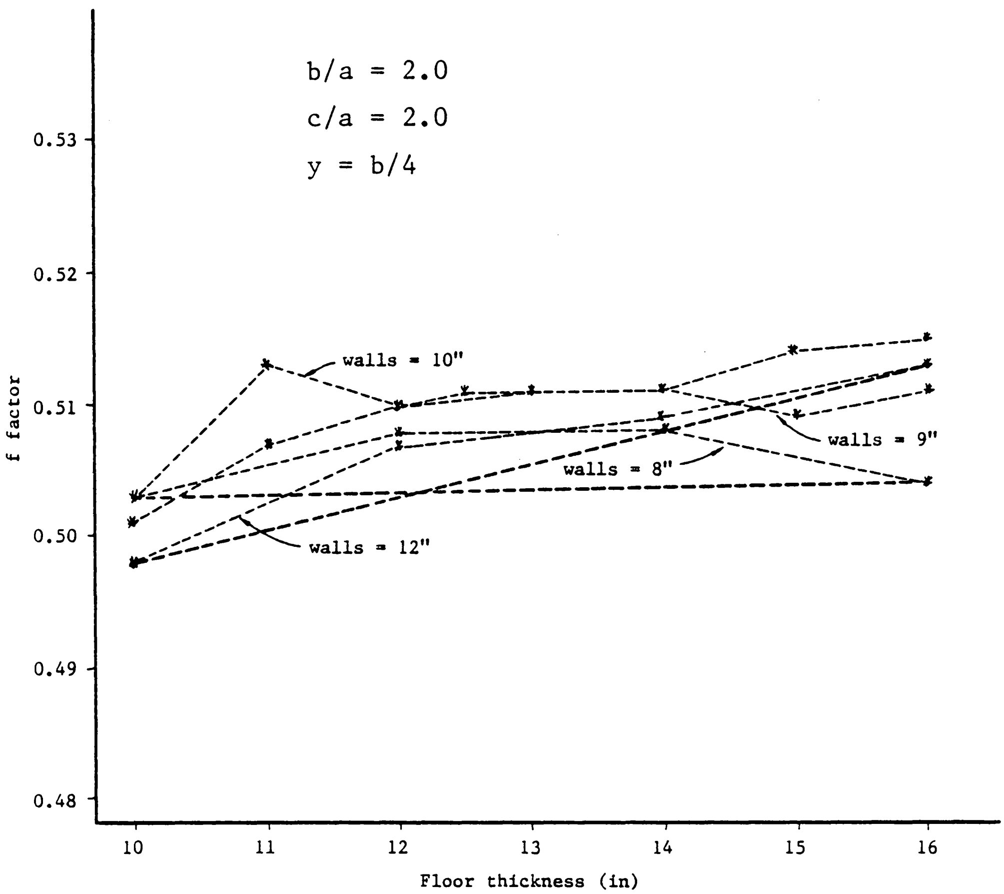


FIGURE A12: Floor stiffness factor

APPENDIX 3

USER'S GUIDE

This appendix is intended to provide a brief description of the subroutines that are included in this program. The required input data is listed at the front of the program in Appendix 4. Output for this program is in kips, inches and radians. Data must be inputted as described by the leading part of the program.

Subroutine DATA:

This subroutine reads in plate dimensions, element meshes, plate thicknesses and material properties. It also calls a subroutine to calculate average thicknesses of the plate elements.

Subroutine THICK:

Subroutine THICK determines the average thicknesses of the plate elements.

Subroutine GEN:

This subroutine generates node and element numbers for the plates.

Subroutine PROCES:

Subroutine PROCES automatically eliminates certain boundary conditions on the plates, the twisting degree of freedom on each plate and all the symmetric boundary conditions on the quarter of a two or three plate problem. It also generates the member codes which contain the degrees of freedom located on each element.

Subroutine LOAD:

This subroutine reads in plate loads and any additional node loads.

Subroutine TRIANG:

This subroutine calculates the node loads for an external or internal hydrostatic load on the walls. The load can be at any height in the tank and must be inputted in units of pounds per cubic feet.

Subroutine UNIF:

This subroutine calculates the node loads for an external or internal uniform load on the walls or floor. The load can be at any height on the walls but must be the full width of the floor.

Subroutine STRIP:

STRIP calculates the node loads on the floor plate for a uniform load around the perimeter of the floor slab with the width equal to the thickness of the wall plus 2.5 times the thickness of the floor slab. The pressure is calculated automatically from the dead weight of the walls.

Subroutine STRIP1:

A modified version of STRIP, this subroutine permits a strip load on a single plate problem. It was designed only for a quarter of a single plate and, therefore, only a symmetric loading can be added to it. The wall and floor thickness must be included and the appropriate weight of the walls for a quarter of the tank must be inputted.

Subroutine DEADWT:

This subroutine calculates the node loads for the walls that include the dead weight of the concrete in the walls. This calculates the total weight of the walls needed in STRIP. This subroutine is not called when one plate is being analyzed.

Subroutine ASSEM:

ASSEM assembles the global system stiffness matrix in a form suitable for solution by the Linpack equation solver.

Subroutine MODIFY:

This subroutine modifies the global stiffness matrix by including the soil stiffness coefficients into it.

Subroutine XLAMDT:

XLAMDT contains the coordinate transformations necessary to transform the local stiffness matrix into the global stiffness matrix.

Subroutine GLOBK:

This subroutine contains the coefficients of the local element stiffness matrix. Only the common terms have been collected in this subroutine. The index matrix is used to identify the remaining terms in the stiffness matrix. This index matrix must be inputted as data in the program.

Subroutine SOLVE:

SOLVE uses the Linpack equation solver to solve the large system of simultaneous equations.

Subroutine FORCE:

This subroutine calculates the nodal displacements for each element in global coordinates and calls a subroutine to calculate the element forces.

Subroutine XKLD:

XKLD transposes global element displacements into local element displacements and claculates the local element forces.

Subroutines SPBFA, SPBSL, SDOT, SAXPY:

These subroutines calculate the nodal displacements given the stiffness matrix stored in a modified banded form and the load vector, all in global coordinates.

```
C**********************************************************************
C                  FINITE ELEMENT PROGRAM DOCUMENTATION                *
C**********************************************************************
C           THIS PROGRAM WAS DESIGNED TO ANALYZE ONE QUARTER OF A
C      RECTANGULAR CONCRETE TANK UTILIZING SYMMETRY TO REDUCE THE NUMBER
C      OF DEGREES OF FREEDOM OF THE SYSTEM.  HOWEVER, ONE OR TWO PLATE
C      PROBLEMS CAN BE ANALYZED.  THE INPUT DATA HAS BEEN MINIMIZED TO
C      PERMIT SOMEONE UNFAMILIAR WITH THE FINITE ELEMENT METHOD TO USE
C      THE PROGRAM.  SOME OF THE PROGRAM FEATURES INCLUDE:
C
C         ==> AUTOMATIC GENERATION OF THE NODE NUMBERS FOR 1,2, OR 3
C                PLATES GIVEN THE NUMBER OF ELEMENTS IN EACH DIRECTION
C         ==> ALLOWANCE FOR TAPERED WALLS
C         ==> AUTOMATIC GENERATION OF ELEMENT NUMBERS
C         ==> AUTOMATIC ELIMINATION OF SOME SYMMETRY BOUNDARY CONDITIONS
C                FOR 1,2, OR 3 PLATES
C         ==> ALLOWANCE FOR ADDITIONAL BOUNDARY CONDITIONS TO BE
C                PRESCRIBED TO ZERO
C         ==> INCLUSION OF BOTH TRIANGULAR AND UNIFORM LOADINGS
C         ==> INCLUSION OF A MODIFIED STRIP LOADING ON THE FLOOR
C                 SLAB TO ACCOUNT FOR DISTRIBUTION OF SHEAR FROM
C                 THE WALLS THROUGH THE FLOOR
C         ==> LOADINGS TO BE INTERNAL OR EXTERNAL
C         ==> LOADINGS TO BE AT ARBITRARY HEIGHT
C         ==> ALLOWANCE FOR ADDITIONAL NODE LOADS
C         ==> INCLUSION OF SOME TRIGGER CARDS TO PREVENT EXECUTION
C                 WITH IMPROPER DATA
C         ==> ALLOWANCE FOR A WINKLER FOUNDATION
C         ==> INCLUSION OF DEAD LOAD FOR WALLS AND FLOOR SLAB
C
C**********************************************************************
C
```

```
C      THE SIMPLIFICATIONS AND ASSUMPTIONS INHERENT IN THIS PROGRAM
C
C         ==> RESISTANCE TO ROTATION IN THE NORMAL DIRECTION OF THE
C                PLATE IS ASSUMED TO BE INFINITE AND IS SUBSEQUENTLY
C                ELIMINATED AS A BOUNDARY CONDITION
C         ==> PLATES MUST BE ORTHONORMAL
C         ==> GLOBAL AXES MUST COINCIDE WITH THE PLATES
C         ==> TRIANGULAR LOADS ARE ONLY PERMITTED ON THE WALLS
C         ==> LOADING CONDITIONS ARE APPROXIMATED AS POINT LOADS AT
C                THE NODES
C         ==> UNIFORM LOAD MUST COVER THE FULL WIDTH OF PLATE 3
C         ==> LOADS MUST BE THE FULL LENGTH OF THE WALL
C         ==> THE THICKNESS OF A TAPERED ELEMENT IS APPROXIMATED BY
C                IT'S AVERAGE THICKNESS
C         ==> THE MODULUS OF ELASTICITY AND POISSON'S RATIO ARE THE
C                SAME FOR ALL THREE PLATES
C
C***********************************************************************
C
C      DESCRIPTION ON SOME OF THE VARIABLE NAMES USED IN THE PROGRAM
C
C       A,B,C ------ ELEMENT DIMENSIONS IN THE GLOBAL 2,3, AND 1-
C                       DIRECTIONS, RESPECTIVELY
C
C       E ---------- MODULUS OF ELASTICITY (KSI)
C
C       IOP -------- A MATRIX OF RANK THREE CONTAINING THE ELEMENT
C                       NUMBERS FOR EACH PLATE
C
C       JCODE ------ CONTAINS THE NUMBERS OF THE DEGREES OF FREEDOM AT
C                       EACH NODE IN GLOBAL COORDINATES
C
```

```
C       MCODE ------ CONTAINS THE DEGREES OF FREEDOM FOR EACH ELEMENT
C
C       NDOF ------- NUMBER OF DEGREES OF FREEDOM
C
C       NELEM ------ NUMBER OF ELEMENTS
C
C       NOP -------- A MATRIX OF RANK THREE CONTAINING THE NODE
C                      NUMBERING SCHEME FOR EACH PLATE
C
C       NNODES ----- NUMBER OF NODES
C
C       NPLTS ------ FLAG INDICATING THE NUMBER OF PLATES BEING
C                      ANALYZED
C
C       NX,NY,NZ --- NUMBER OF ELEMENTS IN THE GLOBAL 1,2, AND 3-
C                       DIRECTIONS, RESPECTIVELY
C
C       Q ---------- REPRESENTS THE LOAD VECTOR BEFORE SUBROUTINE SOLVE
C                      AND THE DISPLACEMENTS AFTER SUBROUTINE SOLVE
C
C       SOIL ------- EQUIVALENT SPRING STIFFNESS OF THE FOUNDATION
C                      AT AN INTERNAL NODE (KIPS/IN)
C
C       SST -------- CONTAINS THE GLOBAL STIFFNESS MATRIX STORED IN
C                      HALF-BANDED FORM THAT CAN BE USED BY THE LINPACK
C                      EQUATION SOLVER
C
C       THK -------- CONTAINS THE STEPPED THICKNESSES FOR EACH PLATE
C
C       THKF ------- THICKNESS OF THE FLOOR
C
C       THKSB,THKST- THICKNESS OF PLATE 1, BOTTOM AND TOP, RESPECTIVELY
```

```
C
C      THKLB,THKLT- THICKNESS OF PLATE 2, BOTTOM AND TOP, RESPECTIVELY
C
C      VNU -------- POISSON'S RATIO
C
C      WC --------- SPECIFIC WEIGHT OF CONCRETE
C
C      X,Y,Z ------ DIMENSIONS OF THE SINGLE PLATE PROBLEM BEING
C                     ANALYZED DETERMINED BY THE BOUNDARY CONDITIONS
C                     BEING USED OR THE DIMENSIONS OF THE SYMMETRIC
C                     PORTION OF THE TWO OR THREE PLATE SYSTEM
C
C***********************************************************************
C                         INPUT OF DATA - UNFORMATTED                  *
C***********************************************************************
C
C  CARD 1 --------- NPLTS                                      (1I)
C
C             ENTER 1,2,3 FOR THE NUMBER OF PLATES TO BE ANALYZED
C
C  CARD 2 --------- X,Y,Z,NX,NY,NZ                             (3R,3I)
C
C             X - DIMENSION IN THE GLOBAL 1-DIRECTION (INCHES)
C             Y - DIMENSION IN THE GLOBAL 2-DIRECTION (INCHES)
C             Z - DIMENSION IN THE GLOBAL 3-DIRECTION (INCHES)
C               *** NOTE: ENTER THE DIMENSIONS THAT CORRESPOND TO THE
C                         BOUNDARY CONDITIONS THAT ARE APPLIED TO THE
C                         PLATE OR PLATES
C                   MUST ENTER Z=0. IF ONLY DOING A SINGLE PLATE PROBLEM
C             NX - NUMBER OF ELEMENTS IN THE GLOBAL 1-DIRECTION
C             NY - NUMBER OF ELEMENTS IN THE GLOBAL 2-DIRECTION
C             NZ - NUMBER OF ELEMENTS IN THE GLOBAL 3-DIRECTION
```

```
C                    (MAXIMUM NUMBER OF ELEMENTS IN ANY DIRECTION
C                     IS EIGHT)
C
C  CARD 3 --------- E,VNU,WC,SOIL                             (4R)
C
C            E - MODULUS OF ELASTICITY OF CONCRETE (KSI)
C            VNU - POISSON'S RATIO OF CONCRETE
C            WC - SPECIFIC WEIGHT OF CONCRETE (PCF)
C            SOIL - ESTIMATED SPRING STIFFNESS OF THE SOIL (K/IN)
C
C  CARD 4 -------- THKST,THKSB,THKLT,THKLB,THKF               (5R)
C
C            THKST - THICKNESS AT THE TOP OF PLATE 1 IN THE GLOBAL
C                      1-2 PLANE (INCHES)
C            THKSB - THICKNESS AT THE BOTTOM OF PLATE 1 IN THE
C                      GLOBAL 1-2 PLANE (INCHES)
C            THKLT - THICKNESS AT THE TOP OF PLATE 2 IN THE GLOBAL
C                      2-3 PLANE (INCHES)
C                         ENTER 0.0 FOR A SINGLE PLATE PROBLEM
C            THKLB - THICKNESS AT THE BOTTOM OF PLATE 2 IN THE
C                      GLOBAL 2-3 PLANE (INCHES)
C                         ENTER 0.0 FOR A SINGLE PLATE PROBLEM
C            THKF - THICKNESS OF THE FLOOR SLAB (INCHES)
C                         ENTER 0.0 FOR 1 OR 2 PLATE PROBLEM
C
C  5TH GROUP OF -- NOD,NDIR                                   (2I)
C    CARDS
C
C            NOD - NODE NUMBER AT WHICH A CONSTRAINT EXISTS
C            NDIR - GLOBAL DIRECTION OF THE CONSTRAINT, BOTH
C                     DISPLACEMENT AND ROTATION CONSTRAINTS
C                     ARE POSSIBLE
```

```
C          *** MUST ENTER  0 0  AS A TRIP CARD TO SIGNIFY THE END
C                    OF THE JOINT CONSTRAINTS ***
C
C   6TH GROUP OF -- LTYPE,NPL,LDIR,W,H                             (3I,2R)
C     CARDS
C
C              LTYPE - INDICATES THE TYPE OF LOADING
C                        ENTER 1 FOR TRIANGULAR LOAD
C                        ENTER 2 FOR UNIFORM LOAD
C                        ENTER 3 FOR APPROXIMATED STRIP LOAD
C              NPL - NUMBER OF THE PLATE TO WHICH THE LOAD IS APPLIED
C                        ENTER 0 FOR LTYPE=3
C                        ENTER 1 FOR PLATE IN THE 1-2 PLANE(PLATE 1)
C                        ENTER 2 FOR PLATE IN THE 2-3 PLANE(PLATE 2)
C                        ENTER 3 FOR THE FLOOR SLAB(PLATE 3)
C              LDIR - INDICATES THE GLOBAL DIRECTION THAT THE LOAD IS
C                        APPLIED.  A NEGATIVE SIGN SHOULD BE ENTERED
C                        WITH THIS VALUE ONLY TO INDICATE THAT THE
C                        LOAD IS APPLIED OPPOSITE TO THE POSITIVE SENSE
C                        OF THE GLOBAL DIRECTIONS.
C                        ENTER 0 FOR LTYPE=3
C              W - MAGNITUDE OF THE LOADING, ALWAYS POSITIVE
C                        IF LTYPE=1, W REPRESENTS THE SPECIFIC WEIGHT OF
C                            THE LOADING (PCF)
C                        IF LTYPE=2, W REPRESENTS THE PRESSURE ON THE
C                            PLATE (PSF)
C                        IF LTYPE=3, ENTER 0.
C                           *** NOTE: IF A SINGLE PLATE PROBLEM IS BEING
C                                 ANALYZED TO DETERMINE THE FIXED END
C                                 MOMENTS ON THE FLOOR SLAB, W REPRESENTS
C                                 THE WEIGHT OF THE WALLS.
C              H - THE HEIGHT OF THE LOADING ON THE WALL.  THIS VALUE
```

```
C                             WILL GENERALLY BE THE FULL HEIGHT OF THE WALL
C                             BUT CAN INCLUDE ANY ARBITRARY HEIGHT.  FOR THE
C                             FLOOR, THE FULL DIMENSION (IN THE X-DIRECTION)
C                             MUST BE ENTERED. (INCHES)
C                             ENTER 0. IF LTYPE=3
C         *** MUST ENTER  0 0 0 0. 0.  AS A TRIP CARD TO INDICATE
C                THE END OF THE PLATE LOADINGS ***
C
C   7TH GROUP OF -- NODE,JDIR,XLOAD                            (2I,R)
C     CARDS
C
C                NODE - NODE NUMBER AT WHICH A CONCENTRATED LOAD IS
C                            APPLIED
C                JDIR - GLOBAL DIRECTION OF THE APPLIED LOAD (POSITIVE)
C                XLOAD - MAGNITUDE OF THE APPLIED LOAD (+ OR -, KIPS
C                             AND INCHES)
C         *** MUST ENTER  0 0 0.  AS A TRIP CARD TO INDICATE THE END
C                 OF ADDITIONAL POINT LOADS
C
```

```
C***********************************************************************
C                               MAIN PROGRAM                           *
C***********************************************************************
C      IMPLICIT REAL*8 (A-H,O-Z)
       COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
      1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
       COMMON/TC/THKST,THKSB,THKLT,THKLB,THKF,WEIGHT
       COMMON/CODE/JCODE(217,6)
       COMMON/SSM/SST(594,993)
       COMMON/FORC/D(24),P21,P32,P11,P22,P31,P33,P12,P13,SA,SB,SC,SD,SE,
      1SF,SG,SH,SI,SJ,SM,SN,SO,SP,SQ,SR,SS,ST,SU,SX,SY,F1,F2,F3,F4,F5,
      2F7,F8,F9,F10,F11,F13,F14,F15,F16,F17,F19,F20,F21,F22,F23,P23,P41
       COMMON/SOLV/MAXID,LDA
       COMMON/COEFF/SOIL
       LDA=594
       READ(5,*) NPLTS
       CALL DATA(NPLTS)
       CALL GEN(NPLTS)
       CALL PROCES(NPLTS)
       CALL LOAD(NPLTS)
       CALL ASSEM(NPLTS)
       CALL SOLVE
       CALL FORCE(NPLTS)
       STOP
       END
```

```
C***********************************************************************
C                              SUBROUTINE DATA                         *
C***********************************************************************
      SUBROUTINE DATA(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/TC/THKST,THKSB,THKLT,THKLB,THKF,WEIGHT
      COMMON/COEFF/SOIL
      READ(5,*) X,Y,Z,NX,NY,NZ
      IF(NX.LE.8.OR.NY.LE.8.OR.NZ.LE.8) GO TO 50
      WRITE(6,100)
      STOP
   50 A=Y/FLOAT(NY)
      IF(NPLTS.NE.1) B=Z/FLOAT(NZ)
      C=X/FLOAT(NX)
      READ(5,*) E,VNU,WC,SOIL
      READ(5,*) THKST,THKSB,THKLT,THKLB,THKF
      DO 10 I=1,NPLTS
      GOTO(11,12,13),I
   11 CALL THICK(THKST,THKSB,THK,NY,I)
      GO TO 10
   12 CALL THICK(THKLT,THKLB,THK,NY,I)
      GO TO 10
   13 CALL THICK(THKF,THKF,THK,NX,I)
   10 CONTINUE
      WRITE(6,101) X,C,Y,A,Z,B
      WRITE(6,201) E,VNU,WC,SOIL
      DO 20 I=1,NPLTS
      WRITE(6,105)
      GOTO(21,21,22),I
   21 J=NY
```

```
      GO TO 25
   22 J=NX
   25 IF(NPLTS.EQ.1) WRITE(6,103) THKSB,THKF
   20 IF(NPLTS.NE.1) WRITE(6,102) I,(THK(K,I),K=1,J)
  100 FORMAT(' YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF ELEMENTS IN THE X
     1-,Y-,OR Z-DIRECTION.  THE MAXIMUM NUMBER OF ELEMENTS AVAILABLE IN
     2THIS PROGRAM IS 8.')
  101 FORMAT(' X=',F10.2,10X,'C=',F10.2/' Y=',F10.2,10X,'A=',F10.2/
     *' Z=',F10.2,10X,'B=',F10.2/)
  102 FORMAT(' STEPPED THICKNESS FOR PLATE',I2,2X,'(INCHES)'/8F8.1)
  103 FORMAT(' THE THICKNESS OF THE WALLS =',F8.1,' INCHES'//
     *' THE THICKNESS OF THE FLOOR PLATE =',F8.1,' INCHES')
  105 FORMAT(/)
  201 FORMAT(' MODULUS OF ELASTICITY=',F10.2,1X,'(KSI)'/
     *' POISSONS RATIO=',F6.2/
     *' SPECIFIC WEIGHT OF CONCRETE=',F10.2,1X,'(PCF)'/
     *' SOIL STIFFNESS=',F8.2,1X,'(KIPS/INCH)'/)
      RETURN
      END
```

```
C**************************************************************************
C                              SUBROUTINE THICK                           *
C**************************************************************************
      SUBROUTINE THICK(T,B,TH,NR,J)
C     IMPLICIT REAL*8 (A-H,O-Z)
      REAL TH(8,3)
      IF(T.EQ.B) GO TO 20
      SLOPE=(B-T)/NR/2
      DO 10 I=1,NR
         M=2*I-1
   10    TH(I,J)=T+M*SLOPE
      RETURN
   20 DO 30 I=1,NR
   30    TH(I,J)=T
      RETURN
      END
```

```
C***********************************************************************
C                             SUBROUTINE GEN                           *
C***********************************************************************
      SUBROUTINE GEN(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      N=0
      NX1=NX+1
      NY1=NY+1
      NZ1=NZ+1
      DO 10 I=1,NY1
         DO 20 J=1,NX1
            N=N+1
   20       NOP(I,J,1)=N
            IF(NPLTS.EQ.1) GO TO 10
      N=N-1
         DO 30 K=1,NZ1
            N=N+1
   30       NOP(I,K,2)=N
   10    CONTINUE
      IF(NPLTS.NE.3) GO TO 90
      N=NOP(NY,NZ1,2)
      DO 60 I=1,NX1
      DO 60 J=1,NZ1
         N=N+1
   60    NOP(I,J,3)=N
      DO 40 I=1,NX1
   40    NOP(NY1,I,1)=NOP(I,1,3)
      DO 50 I=1,NZ1
   50    NOP(NY1,I,2)=NOP(NX1,I,3)
   90 NNODES=N
```

```
      M=0
      DO 210 I=1,NY
      DO 220 J=1,NX
      M=M+1
  220 IOP(I,J,1)=M
      IF(NPLTS.EQ.1) GO TO 210
      DO 230 K=1,NZ
      M=M+1
  230 IOP(I,K,2)=M
  210 CONTINUE
      IF(NPLTS.NE.3) GO TO 150
      DO 240 I=1,NX
      DO 240 J=1,NZ
      M=M+1
  240 IOP(I,J,3)=M
  150 WRITE(6,111)
      WRITE(6,300)
         NNN=1
         WRITE(6,400) NNN
      DO 100 I=1,NY
         WRITE(6,110) (NOP(I,J,1),J=1,NX1)
  100    WRITE(6,310) (IOP(I,J,1),J=1,NX)
      WRITE(6,110) (NOP(NY1,J,1),J=1,NX1)
      IF(NPLTS.EQ.1) RETURN
      WRITE(6,111)
         NNN=2
         WRITE(6,400) NNN
      DO 101 I=1,NY
         WRITE(6,110) (NOP(I,J,2),J=1,NZ1)
  101    WRITE(6,310) (IOP(I,J,2),J=1,NZ)
      WRITE(6,110) (NOP(NY1,J,2),J=1,NZ1)
      IF(NPLTS.EQ.2) RETURN
```

```
      WRITE(6,111)
         NNN=3
         WRITE(6,400) NNN
      DO 102 I=1,NX
         WRITE(6,110) (NOP(I,J,3),J=1,NZ1)
102      WRITE(6,310) (IOP(I,J,3),J=1,NZ)
      WRITE(6,110) (NOP(NX1,J,3),J=1,NZ1)
111   FORMAT(///)
110   FORMAT(9I6)
310   FORMAT(T70,8I6)
300   FORMAT(T5,'NODE NUMBERS',T75,'ELEMENT NUMBERS'//)
400   FORMAT(T61,'PLATE',I2)
      RETURN
      END
```

```
C**********************************************************************
C                               SUBROUTINE PROCES                     *
C**********************************************************************
      SUBROUTINE PROCES(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/CODE/JCODE(217,6)
      COMMON/SOLV/MAXID,LDA
C
C     JOINT CODE CONSTRUCTION
C
      DO 10 I=1,NNODES
      DO 10 J=1,6
   10    JCODE(I,J)=1
C
C     JOINT CONSTRAINTS
C
      NX1=NX+1
      NY1=NY+1
      NZ1=NZ+1
      GOTO(201,202,203),NPLTS
  203 DO 22 I=1,NX1
      DO 22 J=1,NZ1
         M=NOP(I,J,3)
   22    JCODE(M,5)=0
      DO 25 I=1,NX1
         M=NOP(I,NZ1,3)
         JCODE(M,3)=0
   25    JCODE(M,4)=0
      DO 26 J=1,NZ1
         M=NOP(1,J,3)
```

```
      JCODE(M,1)=0
 26   JCODE(M,6)=0
202 DO 21 I=1,NY1
    DO 21 J=1,NZ1
      M=NOP(I,J,2)
 21   JCODE(M,4)=0
    DO 23 I=1,NY1
      M=NOP(I,1,1)
      JCODE(M,1)=0
 23   JCODE(M,5)=0
    DO 24 I=1,NY1
      M=NOP(I,NZ1,2)
      JCODE(M,3)=0
 24   JCODE(M,5)=0
201 DO 20 I=1,NY1
    DO 20 J=1,NX1
      M=NOP(I,J,1)
 20   JCODE(M,6)=0
 30 READ(5,*) NOD,NDIR
    IF(NOD.EQ.0) GO TO 35
    JCODE(NOD,NDIR)=0
    GO TO 30
 35 NDOF=0
    DO 36 I=1,NNODES
    DO 36 J=1,6
      IF(JCODE(I,J).EQ.0) GO TO 36
      NDOF=NDOF+1
      JCODE(I,J)=NDOF
 36   CONTINUE
    DO 40 M=1,NY
    DO 40 N=1,NX
      I=NOP(M,N,1)
```

```
         J=NOP(M,N+1,1)
         K=NOP(M+1,N,1)
         L=NOP(M+1,N+1,1)
         NN=IOP(M,N,1)
         DO 41 NM=1,6
            MCODE(NN,NM)=JCODE(I,NM)
            MCODE(NN,NM+6)=JCODE(J,NM)
            MCODE(NN,NM+12)=JCODE(K,NM)
41          MCODE(NN,NM+18)=JCODE(L,NM)
40       CONTINUE
      IF(NPLTS.EQ.1) GO TO 65
      DO 50 M=1,NY
      DO 50 N=1,NZ
         I=NOP(M,N,2)
         J=NOP(M,N+1,2)
         K=NOP(M+1,N,2)
         L=NOP(M+1,N+1,2)
         NN=IOP(M,N,2)
         DO 51 NM=1,6
            MCODE(NN,NM)=JCODE(I,NM)
            MCODE(NN,NM+6)=JCODE(J,NM)
            MCODE(NN,NM+12)=JCODE(K,NM)
51          MCODE(NN,NM+18)=JCODE(L,NM)
50       CONTINUE
      IF(NPLTS.NE.3) GO TO 65
      DO 60 M=1,NX
      DO 60 N=1,NZ
         I=NOP(M,N,3)
         J=NOP(M,N+1,3)
         K=NOP(M+1,N,3)
         L=NOP(M+1,N+1,3)
         NN=IOP(M,N,3)
```

```
      DO 61 NM=1,6
         MCODE(NN,NM)=JCODE(I,NM)
         MCODE(NN,NM+6)=JCODE(J,NM)
         MCODE(NN,NM+12)=JCODE(K,NM)
61       MCODE(NN,NM+18)=JCODE(L,NM)
60    CONTINUE
65 NELEM=NN
   MAXID=0
   NE=NX*NY+NY*NZ
   DO 70 I=1,NE
      J=0
75    J=J+1
      IS=MCODE(I,J)
      IF(IS.EQ.0) GO TO 75
      J=25
76    J=J-1
      IL=MCODE(I,J)
      IF(IL.EQ.0) GO TO 76
      ID=IL-IS
      IF(ID.GT.MAXID) MAXID=ID
70    CONTINUE
   IF(NPLTS.NE.3) GO TO 81
   NNE=NX*NZ
   DO 80 I=NE,NNE
      MAX=0
      MIN=400
      DO 90 J=1,24
         M=MCODE(I,J)
         IF(M.EQ.0) GO TO 90
         IF(M.LT.MIN) MIN=M
         IF(M.GT.MAX) MAX=M
90    CONTINUE
```

```
          ID=MAX-MIN
          IF(ID.GT.MAXID) MAXID=ID
   80 CONTINUE
   81 IHBW=MAXID+1
C     WRITE(6,105)
C     WRITE(6,99)
C     WRITE(6,100)(I,(JCODE(I,J),J=1,6),I=1,NNODES)
C     WRITE(6,105)
C     WRITE(6,109)
C     WRITE(6,110)(I,(MCODE(I,J),J=1,24),I=1,NELEM)
      WRITE(6,105)
      WRITE(6,200) NNODES,NELEM,NDOF,IHBW
      WRITE(6,105)
C  99 FORMAT(' JOINT',4X,T23,'JOINT CODE'/)
C 100 FORMAT(I4,6X,6I5)
  105 FORMAT(///)
C 109 FORMAT(' ELEMENT',2X,T60,'MEMBER CODE'/)
C 110 FORMAT(I5,5X,24I5)
  200 FORMAT(' NUMBER OF NODES=',I4/' NUMBER OF ELEMENTS=',I4/
     *' NUMBER OF DEGREES OF FREEDOM=',I4/
     *' THE HALF BAND WIDTH=',I4/)
      RETURN
      END
```

```
C***********************************************************************
C                              SUBROUTINE LOAD                         *
C***********************************************************************
      SUBROUTINE LOAD(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/CODE/JCODE(217,6)
      DO 2 I=1,NDOF
    2 Q(I)=0.
      IF(NPLTS.NE.1) CALL DEADWT(NPLTS)
   40 READ(5,*) LTYPE,NPL,LDIR,W,H
      IF(LTYPE.EQ.0) GO TO 50
      GOTO(11,21,31),LTYPE
   99 GOTO(10,20,30),LTYPE
   10 CALL TRIANG(NPL,LDIR,W,H)
      GO TO 40
   20 CALL UNIF(NPL,LDIR,W,H)
      GO TO 40
   30 IF(NPLTS.NE.1) CALL STRIP
      IF(NPLTS.EQ.1) CALL STRIP1(W)
      GO TO 40
   50 READ(5,*) NODE,JDIR,XLOAD
      IF(NODE.EQ.0) RETURN
      M=JCODE(NODE,JDIR)
      IF(M.EQ.0) GO TO 32
      Q(M)=Q(M)+XLOAD
      GO TO 30
   32 WRITE(6,100) M
      STOP
   11 WRITE(6,105)
      WRITE(6,111) NPL,LDIR,W,H
```

```
      GO TO 99
   21 WRITE(6,105)
      WRITE(6,121) NPL,LDIR,W,H
      GO TO 99
   31 WRITE(6,105)
      WRITE(6,131)
      GO TO 99
  100 FORMAT(' THE APPLIED LOAD AT NODE',I5,'CORRESPONDS TO THE LOCATION
     * OF A CONSTRAINT.  CHECK THE LOCATION OF THE APPLIED LOAD.')
  105 FORMAT(//)
  111 FORMAT(' A TRIANGULAR LOAD WAS APPLIED TO PLATE',I2,' IN THE GLOBA
     *L',I4,' DIRECTION.'/T3,' THE INTENSITY OF THE LOAD WAS',F8.2,1X,'P
     *CF AND WAS APPLIED TO A HEIGHT OF',F8.2,1X,'INCHES'/)
  121 FORMAT(' A UNIFORM LOAD WAS APPLIED TO PLATE',I2,' IN THE GLOBAL',
     *I4,' DIRECTION.'/T3,'THE INTENSITY OF THE LOAD WAS',F10.2,' PSF AN
     *D WAS APPLIED TO A HEIGHT OF',F8.2,' INCHES'/)
  131 FORMAT(' A STRIP LOAD APPROXIMATION WAS USED ON THE FLOOR SLAB'/)
      END
```

```
C*********************************************************************
C                          SUBROUTINE TRIANG                         *
C*********************************************************************
      SUBROUTINE TRIANG(NPL,LDIR,W,H)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/CODE/JCODE(217,6)
      NN=0
      P2=0.
      D=Y-H
      NY2=2*NY
      W=W/1728000.
      DO 10 I=1,NY2,2
         DY=FLOAT(I)*A/2.
         IF(DY.LT.D) GO TO 10
         P1=P2
         P2=(DY-D)*W
         NN=NN+1
         DH=A
         IF(NN.EQ.1) DH=DY-D
         P=(P1+P2)/2.
         IF(LDIR.LT.0) P=-P
         GOTO(21,22),NPL
   21    JDIR=3
         K=NX+1
         GO TO 23
   22    JDIR=1
         K=NZ+1
   23    DO 30 L=1,K
         GOTO(31,32),NPL
   31       DL=C
```

```
          GO TO 33
   32     DL=B
   33     IF(L.EQ.1.OR.L.EQ.K) DL=DL/2.
          M=NOP((I+1)/2,L,NPL)
          N=JCODE(M,JDIR)
          IF(N.EQ.0) GO TO 29
          Q(N)=Q(N)+P*DH*DL
C         WRITE(6,101) M,Q(N)
          GO TO 30
   29     WRITE(6,100) M
   30   CONTINUE
   10 CONTINUE
      P1=P2
      P2=H*W
      P=(P1+P2)/2.
      IF(LDIR.LT.0) P=-P
      GOTO(41,42),NPL
   41 JDIR=3
      K=NX+1
      GO TO 43
   42 JDIR=1
      K=NZ+1
   43 DO 50 L=1,K
        GOTO(51,52),NPL
   51   DL=C
        GO TO 53
   52   DL=B
   53   IF(L.EQ.1.OR.L.EQ.K) DL=DL/2.
        M=NOP((I+3)/2,L,NPL)
        N=JCODE(M,JDIR)
        IF(N.EQ.0) GO TO 49
        Q(N)=Q(N)+P*A/2.*DL
```

```
C         WRITE(6,101) M,Q(N)
          GO TO 50
   49     WRITE(6,100) M
   50 CONTINUE
  100 FORMAT(' A CONSTRAINT EXISTS IN THE DIRECTION OF THE APPLIED TRIAN
     1GULAR LOAD AT NODE',I5,/T5,'THE LOAD WAS NOT ENTERED INTO THE LOAD
     2 VECTOR')
C 101 FORMAT(I5,F12.2)
      RETURN
      END
```

```
C*********************************************************************
C                              SUBROUTINE UNIF                       *
C*********************************************************************
      SUBROUTINE UNIF(NPL,LDIR,P,H)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/CODE/JCODE(217,6)
      NN=0
      IF(NPL.EQ.3.AND.H.NE.X) GO TO 90
      GOTO(1,1,3),NPL
    1 NR=2*NY
      F=A
      V=Y
      GO TO 5
    3 NR=2*NX
      F=C
      V=X
    5 P=P/144000.
      D=V-H
      IF(LDIR.LT.0) P=-P
      DO 10 I=1,NR,2
         DY=FLOAT(I)*F/2.
         IF(DY.LT.D) GO TO 10
         NN=NN+1
         DH=F
         IF(NN.EQ.1) DH=DY-D
         GOTO(11,12,13),NPL
   11    JDIR=3
         K=NX+1
         GO TO 15
   12    JDIR=1
```

```
          K=NZ+1
          GO TO 15
   13     JDIR=2
          K=NZ+1
   15     DO 20 L=1,K
             GOTO(21,22,22),NPL
   21        DL=C
             GO TO 25
   22        DL=B
   25        IF(L.EQ.1.OR.L.EQ.K) DL=DL/2.
             M=NOP((I+1)/2,L,NPL)
             N=JCODE(M,JDIR)
             IF(N.EQ.0) GO TO 19
             Q(N)=Q(N)+P*DH*DL
C            WRITE(6,101) M,Q(N)
             GO TO 20
   19        WRITE(6,100) M
   20     CONTINUE
   10  CONTINUE
       DO 30 L=1,K
          GOTO(31,32,32),NPL
   31     DL=C
          GO TO 35
   32     DL=B
   35     IF(L.EQ.1.OR.L.EQ.K) DL=DL/2.
          M=NOP((I+3)/2,L,NPL)
          N=JCODE(M,JDIR)
          IF(N.EQ.0) GO TO 29
          Q(N)=Q(N)+P*F/2.*DL
C         WRITE(6,101) M,Q(N)
          GO TO 30
   29     WRITE(6,100) M
```

```
   30 CONTINUE
      RETURN
   90 WRITE(6,102)
      STOP
  100 FORMAT(' A CONSTRAINT EXISTS IN THE DIRECTION OF THE APPLIED UNIFO
     1RM LOAD AT NODE',I4/T5,'THE LOAD WAS NOT ENTERED INTO THE LOAD VEC
     2TOR')
C 101 FORMAT(I5,F12.2)
  102 FORMAT(' THE UNIFORM LOAD ON THE FLOOR OF THE TANK MUST BE THE FUL
     1L WIDTH OF THE TANK.'/T5,'H MUST EQUAL X AND W MUST BE ADJUSTED SO
     2 THAT THE MULTIPLICATION OF W AND H'/T5,'PROVIDE THE APPROPRIATE P
     3RESSURE ACROSS THE BOTTOM OF THE TANK'/)
      END
```

```
C***********************************************************************
C                              SUBROUTINE STRIP                        *
C***********************************************************************
      SUBROUTINE STRIP
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/TC/THKST,THKSB,THKLT,THKLB,THKF,WEIGHT
      COMMON/CODE/JCODE(217,6)
      WZ=THKSB+2.5*THKF
      WX=THKLB+2.5*THKF
      IF(WZ.GT.Z.OR.WX.GT.X) GO TO 50
      AREA=X*WZ+Z*WX-WX*WZ
      PRESS=WEIGHT/AREA
      WRITE(6,200) WEIGHT,PRESS,WX,WZ
      NX1=NX+1
      NZ1=NZ+1
      NC=2*NZ1
      DX=C/2.
      DZ=B/2.
      TRIP1=WZ+2.*DZ
      DO 10 I=1,NX1
         XHITE=C
         IF(I.EQ.1.OR.I.EQ.NX1) XHITE=XHITE/2.
         DO 20 J=1,NC,2
            ZH=DZ*FLOAT(J)
            IF(ZH.GT.TRIP1) GO TO 10
            WIDTH=B
            IF(J.EQ.1) WIDTH=WIDTH/2.
            IF(ZH.GT.WZ) WIDTH=B-ZH+WZ
            XLOAD=PRESS*WIDTH*XHITE
            L=(J+1)/2
```

```
            M=NOP(I,L,3)
            N=JCODE(M,2)
            IF(N.EQ.0) GO TO 19
            Q(N)=Q(N)+XLOAD
            GO TO 20
 19         WRITE(6,100) M
 20      CONTINUE
 10   CONTINUE
      I=(INT((X-WX)/DX)+1)/2+1
      J=(INT(WZ/DZ)+1)/2+1
      DO 30 K=I,NX1
         XHITE=C
         IF(K.EQ.I) XHITE=(2*I-1)*DX-X+WX
         IF(K.EQ.NX1) XHITE=C/2.
         DO 40 L=J,NZ1
            WIDTH=B
            IF(L.EQ.J) WIDTH=(2*J-1)*DZ-WZ
            IF(L.EQ.NZ1) WIDTH=B/2.
            XLOAD=PRESS*WIDTH*XHITE
            M=NOP(K,L,3)
            N=JCODE(M,2)
            IF(N.EQ.0) GO TO 39
            Q(N)=Q(N)+XLOAD
            GO TO 40
 39         WRITE(6,100) M
 40      CONTINUE
 30   CONTINUE
      RETURN
 50   WRITE(6,101)
      RETURN
100   FORMAT( ' A CONSTRAINT EXISTS IN THE DIRECTION OF THE APPLIED STRI
     *P LOAD AT NODE',I4/T5,'THE LOAD WAS NOT ENTERED INTO THE LOAD VECT
```

```
     *OR')
 101 FORMAT(/' THE STRIP LOAD COVERS THE ENTIRE FLOOR SLAB'/T5,'REENTER
     * AS A UNIFORM LOAD. EXECUTION WAS TERMINATED')
 200 FORMAT(//' THE TOTAL WEIGHT OF THE WALLS=',F10.2,' KIPS'/
     *' THE UNIFORM STRIP PRESSURE=',F10.7,' KSI'/
     *' THE WIDTH OF THE STRIP IN THE X-DIRECTION=',F10.3,' INCHES'/
     *' THE WIDTH OF THE STRIP IN THE Z-DIRECTION=',F10.3,' INCHES'//)
      END
```

```
C***************************************************************************
C                             SUBROUTINE STRIP1                            *
C***************************************************************************
      SUBROUTINE STRIP1(WEIGH1)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/TC/THKYT,THKYB,THKXT,THKXB,THKF,WEIGHT
      COMMON/CODE/JCODE(217,6)
      WX=THKYB+2.5*THKF
      WY=THKXB+2.5*THKF
      IF(WY.GT.Y.OR.WX.GT.X) GO TO 50
      AREA=X*WY+Y*WX-WX*WY
      PRESS=WEIGH1/AREA
      WRITE(6,200) WEIGH1,PRESS,WX,WY
      NX1=NX+1
      NY1=NY+1
      NC=2*NX1
      DX=C/2.
      DY=A/2.
      TRIP1=WX+2.*DX
      DO 10 I=1,NY1
          XHITE=A
          IF(I.EQ.1.OR.I.EQ.NY1) XHITE=XHITE/2.
          DO 20 J=1,NC,2
              XH=DX*FLOAT(J)
              IF(XH.GT.TRIP1) GO TO 10
              WIDTH=C
              IF(J.EQ.1) WIDTH=WIDTH/2.
              IF(XH.GT.WX) WIDTH=C-XH+WX
              XLOAD=PRESS*WIDTH*XHITE
              L=(J+1)/2
```

```
          M=NOP(I,L,1)
          N=JCODE(M,3)
          IF(N.EQ.0) GO TO 19
          Q(N)=Q(N)+XLOAD
          GO TO 20
19        WRITE(6,100) M
20      CONTINUE
10  CONTINUE
    I=((INT((Y-WY)/DY)+1)/2+1
    J=(INT(WX/DX)+1)/2+1
    DO 30 K=I,NY1
        XHITE=A
        IF(K.EQ.I) XHITE=(2*I-1)*DY-Y+WY
        IF(K.EQ.NY1) XHITE=A/2.
        DO 40 L=J,NX1
          WIDTH=C
          IF(L.EQ.J) WIDTH=(2*J-1)*DX-WX
          IF(L.EQ.NX1) WIDTH=C/2.
          XLOAD=PRESS*WIDTH*XHITE
          M=NOP(K,L,1)
          N=JCODE(M,3)
          IF(N.EQ.0) GO TO 39
          Q(N)=Q(N)+XLOAD
          GO TO 40
39        WRITE(6,100) M
40      CONTINUE
30  CONTINUE
    RETURN
50  WRITE(6,101)
    RETURN
100 FORMAT( ' A CONSTRAINT EXISTS IN THE DIRECTION OF THE APPLIED STRI
   *P LOAD AT NODE',I4/T5,'THE LOAD WAS NOT ENTERED INTO THE LOAD VECT
```

```
    *OR')
101 FORMAT(/' THE STRIP LOAD COVERS THE ENTIRE FLOOR SLAB'/T5,'REENTER
    * AS A UNIFORM LOAD. EXECUTION WAS TERMINATED')
200 FORMAT(//' THE TOTAL WEIGHT OF THE WALLS=',F10.2,' KIPS'/
    *' THE UNIFORM STRIP PRESSURE=',F10.7,' KSI'/
    *' THE WIDTH OF THE STRIP IN THE X-DIRECTION=',F10.3,' INCHES'/
    *' THE WIDTH OF THE STRIP IN THE Y-DIRECTION=',F10.3,' INCHES'//)
     END
```

```
C***********************************************************************
C                            SUBROUTINE DEADWT                          *
C***********************************************************************
      SUBROUTINE DEADWT(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/TC/THKST,THKSB,THKLT,THKLB,THKF,WEIGHT
      COMMON/CODE/JCODE(217,6)
      WEIGHT=0.
      NP=NPLTS
      IF(NPLTS.EQ.3) NP=2
      W=WC/1728000.
      DO 10 I=1,NP
         GOTO(11,11,13),I
   11    NR=2*NY
         GO TO 15
   13    NR=2*NX
   15    DO 20 J=1,NR,2
            IF(J.NE.1) GO TO 80
            GOTO(51,52,53),I
   51       T=THKST
            BB=THKSB
            TH2=T
            GO TO 55
   52       T=THKLT
            BB=THKLB
            TH2=T
            GO TO 55
   53       TH=THKF
            GO TO 60
   55       IF(T.EQ.BB) GO TO 61
```

```
      SLOPE=(BB-T)/FLOAT(NR)
   80 IF(T.EQ.BB.OR.I.EQ.3) GO TO 60
      TH1=TH2
      TH2=SLOPE*FLOAT(J)+T
      TH=(TH1+TH2)/2.
      GO TO 60
   61 TH=T
   60 GOTO(21,22),I
   21 K=NX+1
      DH=A
      GO TO 25
   22 K=NZ+1
      DH=A
      GO TO 25
   23 K=NZ+1
      DH=C
   25 IF(J.EQ.1) DH=DH/2.
      DO 30 L=1,K
         GOTO(31,32,32),I
   31    DL=C
         GO TO 35
   32    DL=B
   35    IF(L.EQ.1.OR.L.EQ.K) DL=DL/2.
         M=NOP((J+1)/2,L,I)
         N=JCODE(M,2)
         XLOAD=W*TH*DH*DL
         WEIGHT=WEIGHT+XLOAD
         IF(N.EQ.0) GO TO 29
         Q(N)=Q(N)-XLOAD
C        WRITE(6,101) M,Q(N)
         GO TO 30
   29    WRITE(6,100) M
```

```
   30          CONTINUE
   20       CONTINUE
            IF(T.EQ.BB.OR.I.EQ.3) GO TO 70
            TH1=TH2
            TH2=BB
            TH=(TH1+TH2)/2.
   70       DO 40 L=1,K
               GOTO(41,42,42),I
   41          DL=C
               GO TO 45
   42          DL=B
   45          IF(L.EQ.1.OR.L.EQ.K) DL=DL/2.
               M=NOP((J+3)/2,L,I)
               N=JCODE(M,2)
               XLOAD=W*TH*DH/2.*DL
               WEIGHT=WEIGHT+XLOAD
               IF(N.EQ.0) GO TO 39
               Q(N)=Q(N)-XLOAD
C              WRITE(6,101) M,Q(N)
               GO TO 40
   39          WRITE(6,100) M
   40       CONTINUE
   10 CONTINUE
  100 FORMAT(' A CONSTRAINT EXISTS IN THE DIRECTION OF THE DEAD WEIGHT L
     1OAD AT NODE',I4/T5,'THE LOAD WAS NOT ENTERED INTO THE LOAD VECTOR'
     2)
C 101 FORMAT(I5,F12.2)
      RETURN
      END
```

```
C*************************************************************************
C                          SUBROUTINE ASSEM                              *
C*************************************************************************
      SUBROUTINE ASSEM(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/TC/THKST,THKSB,THKLT,THKLB,THKF,WEIGHT
      COMMON/CODE/JCODE(217,6)
      COMMON/SSM/SST(594,993)
      COMMON/GIND/G(163)
      COMMON/COEFF/SOIL
      DIMENSION INDEX(24,24)
      READ(5,*) ((INDEX(I,J),J=1,24),I=1,24)
      DO 81 I=1,IHBW
      DO 81 J=1,NDOF
   81    SST(I,J)=0.
      TT=0.
      DO 1 I1=1,NPLTS
         CALL XLAMDT(I1,XL)
         GOTO(11,12,13),I1
   11    A1=A
         B1=C
         NR=NY
         NC=NX
         GO TO 15
   12    A1=A
         B1=B
         NR=NY
         NC=NZ
         GO TO 15
   13    A1=C
```

```
      B1=B
      NR=NX
      NC=NZ
15    DO 10 I=1,NR
         T=THK(I,I1)
         IF(T.NE.TT.OR.I.EQ.1) CALL GLOBK(I1,A1,B1,T)
         TT=T
         DO 20 J=1,NC
            NN=IOP(I,J,I1)
            DO 30 JM=1,24
               J1=MCODE(NN,JM)
               IF(J1.EQ.0) GO TO 30
               DO 40 KM=JM,24
                  K=MCODE(NN,KM)
                  IF(K.EQ.0) GO TO 40
                  KB=J1-K+IHBW
                  L=INDEX(JM,KM)
                  IF(L.GT.0) GO TO 41
                  L=-L
                  SST(KB,K)=SST(KB,K)-G(L)
                  GO TO 40
41                SST(KB,K)=SST(KB,K)+G(L)
40             CONTINUE
30          CONTINUE
20       CONTINUE
10    CONTINUE
 1 CONTINUE
   IF(NPLTS.EQ.3.AND.SOIL.GT.0.) CALL MODIFY
   RETURN
   END
```

```
C***********************************************************************
C                          SUBROUTINE MODIFY                           *
C***********************************************************************
      SUBROUTINE MODIFY
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/CODE/JCODE(217,6)
      COMMON/SSM/SST(594,993)
      COMMON/COEFF/SOIL
      NZ1=NZ+1
      NX1=NX+1
      DO 10 I=1,NX1
      DO 10 J=1,NZ1
         K=NOP(I,J,3)
         L=JCODE(K,2)
         IF(L.EQ.0) GO TO 10
         COEF=SOIL
         IF(I.EQ.1.OR.I.EQ.NX1) COEF=COEF/2.
         IF(J.EQ.1.OR.J.EQ.NZ1) COEF=COEF/2.
         SST(IHBW,L)=SST(IHBW,L)+COEF
   10    CONTINUE
      RETURN
      END
```

```
C***********************************************************************
C                             SUBROUTINE XLAMDT                         *
C***********************************************************************
      SUBROUTINE XLAMDT(NPLTS,L)
C     IMPLICIT REAL*8 (A-H,O-Z)
      REAL L(3,6)
      DO 10 I=1,3
      DO 10 J=1,6
   10 L(I,J)=0.
      GOTO(11,12,13),NPLTS
   11 L(1,1)=0.
      L(1,2)=-1.
      L(1,5)=0.
      L(2,1)=-1.
      L(2,2)=0.
      L(2,5)=0.
      L(3,1)=0.
      L(3,2)=0.
      L(3,5)=1.
      RETURN
   12 L(1,1)=0.
      L(1,2)=0.
      L(1,5)=1.
      L(2,1)=-1.
      L(2,2)=0.
      L(2,5)=0.
      L(3,1)=0.
      L(3,2)=1.
      L(3,5)=0.
      RETURN
   13 L(1,1)=-1.
      L(1,2)=0.
```

```
L(1,5)=0.
L(2,1)=0.
L(2,2)=0.
L(2,5)=-1.
L(3,1)=0.
L(3,2)=1.
L(3,5)=0.
RETURN
END
```

```
C***********************************************************************
C                            SUBROUTINE GLOBK                          *
C***********************************************************************
      SUBROUTINE GLOBK(NPL,A1,B1,T)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/GIND/G(163)
      COMMON/FORC/D(24),P21,P32,P11,P22,P31,P33,P12,P13,SA,SB,SC,SD,SE,
     1SF,SG,SH,SI,SJ,SM,SN,SO,SP,SQ,SR,SS,ST,SU,SX,SY,F1,F2,F3,F4,F5,
     2F7,F8,F9,F10,F11,F13,F14,F15,F16,F17,F19,F20,F21,F22,F23,P23,P41
C
C     DETERMINE COMMON TERMS USED IN PLAIN STRAIN MATRIX
C
      P=A1/B1
      PA=60.+30.*VNU**2/(1.-VNU)
      PB=22.5*(1-VNU)
      PC=30.-30.*VNU**2/(1.-VNU)
      ETC=E*T/180./(1.-VNU**2)
      PD=22.5*(1.+VNU)
      PE=22.5*(1.-3.*VNU)
      P11=(PA/P+PB*P)*ETC
      P22=(PA*P+PB/P)*ETC
      P21=PD*ETC
      P31=(PC/P-PB*P)*ETC
      P32=(-PC/P-PB*P)*ETC
      P33=(-PA*P+PB/P)*ETC
      P13=PE*ETC
      P23=(-PA/P+PB*P)*ETC
      P12=(PC*P-PB/P)*ETC
      P41=(-PC*P-PB/P)*ETC
C
```

```
C       DETERMINE COMMON TERMS IN THE PLATE BENDING MATRIX
C
        DX=E*T**3/12./(1.-VNU**2)
        DY=DX
        D1X=VNU*DX
        DXY=0.5*(SQRT(DX*DY)-D1X)
        A2=A1/2.
        B2=B1/2.
        PDX=DX/(P**2)
        PDY=DY*P**2
        SA=(20.*PDY+8.*DXY)*B2/(15.*A2)
        SB=D1X
        SC=(20.*PDX+8.*DXY)*A2/(15.*B2)
        SD=(30.*PDY+15.*D1X+6.*DXY)/(30.*A2)
        SE=(30.*PDX+15.*D1X+6.*DXY)/(30.*B2)
        SF=(60.*PDX+60.*PDY+30.*D1X+84.*DXY)/(60.*A2*B2)
        SG=(10.*PDY-2.*DXY)*B2/(15.*A2)
        SH=(-30.*PDY-6.*DXY)/(30.*A2)
        SI=(10.*PDX-8.*DXY)*A2/(15.*B2)
        SJ=(15.*PDX-15.*D1X-6.*DXY)/(30.*B2)
        SM=(30.*PDX-60.*PDY-30.*D1X-84.*DXY)/(60.*A2*B2)
        SN=(10.*PDY-8.*DXY)*B2/(15.*A2)
        SO=(-15.*PDY+15.*D1X+6.*DXY)/(30.*A2)
        SP=(5.*PDY+2.*DXY)*B2/(15.*A2)
        SQ=(15.*PDY-6.*DXY)/(30.*A2)
        SR=(10.*PDX-2.*DXY)*A2/(15.*B2)
        SS=(30.*PDX+6.*DXY)/(30.*B2)
        ST=(5.*PDX+2.*DXY)*A2/(15.*B2)
        SU=(15.*PDX-6.*DXY)/(30.*B2)
        SX=(-60.*PDX+30.*PDY-30.*D1X-84.*DXY)/(60.*A2*B2)
        SY=(-30.*PDX-30.*PDY+30.*D1X+84.*DXY)/(60.*A2*B2)
        IF(NPL.EQ.5) RETURN
```

```
C
C     DETERMINE COEFFICIENTS OF THE INDEX MATRIX
C
      AA=XL(1,1)
      BB=XL(1,2)
      CC=XL(1,5)
      DD=XL(2,1)
      EE=XL(2,2)
      FF=XL(2,5)
      GG=XL(3,1)
      HH=XL(3,2)
      XI=XL(3,5)
      G(1)=AA*AA*P11+   2.*AA*BB*P21   +BB*BB*P22+CC*CC*SF
      G(2)=DD*AA*P11+(DD*BB+EE*AA)*P21+EE*BB*P22+FF*CC*SF
      G(3)=GG*AA*P11+(GG*BB+HH*AA)*P21+HH*BB*P22+XI*CC*SF
      G(4)=-AA*CC*SD+BB*CC*SE
      G(5)=-DD*CC*SD+EE*CC*SE
      G(6)=-GG*CC*SD+HH*CC*SE
      G(7)=AA*AA*P31+                  BB*BB*P33+CC*CC*SM
      G(8)=DD*AA*P31+(EE*AA-DD*BB)*P13+EE*BB*P33+FF*CC*SM
      G(9)=GG*AA*P31+(HH*AA-GG*BB)*P13+HH*BB*P33+XI*CC*SM
      G(10)=AA*CC*SH+BB*CC*SJ
      G(11)=DD*CC*SH+EE*CC*SJ
      G(12)=GG*CC*SH+HH*CC*SJ
      G(13)=AA*AA*P23+                  BB*BB*P12+CC*CC*SX
      G(14)=DD*AA*P23+(DD*BB-EE*AA)*P13+EE*BB*P12+FF*CC*SX
      G(15)=GG*AA*P23+(GG*BB-HH*AA)*P13+HH*BB*P12+XI*CC*SX
      G(16)=AA*CC*SO+BB*CC*SS
      G(17)=DD*CC*SO+EE*CC*SS
      G(18)=GG*CC*SO+HH*CC*SS
      G(19)=AA*AA*P32-   2.*AA*BB*P21   +BB*BB*P41+CC*CC*SY
      G(20)=DD*AA*P32-(EE*AA+DD*BB)*P21+EE*BB*P41+FF*CC*SY
```

```
G(21)=GG*AA*P32-(HH*AA+GG*BB)*P21+HH*BB*P41+XI*CC*SY
G(22)=-AA*CC*SQ+BB*CC*SU
G(23)=-DD*CC*SQ+EE*CC*SU
G(24)=-GG*CC*SQ+HH*CC*SU
G(25)=DD*DD*P11+    2.*DD*EE*P21   +EE*EE*P22+FF*FF*SF
G(26)=GG*DD*P11+(GG*EE+HH*DD)*P21+HH*EE*P22+XI*FF*SF
G(27)=-AA*FF*SD+BB*FF*SE
G(28)=-DD*FF*SD+EE*FF*SE
G(29)=-GG*FF*SD+HH*FF*SE
G(30)=AA*DD*P31+(BB*DD-AA*EE)*P13+BB*EE*P33+CC*FF*SM
G(31)=DD*DD*P31+                   EE*EE*P33+FF*FF*SM
G(32)=GG*DD*P31+(HH*DD-GG*EE)*P13+HH*EE*P33+XI*FF*SM
G(33)=AA*FF*SH+BB*FF*SJ
G(34)=DD*FF*SH+EE*FF*SJ
G(35)=GG*FF*SH+HH*FF*SJ
G(36)=AA*DD*P23+(AA*EE-BB*DD)*P13+BB*EE*P12+CC*FF*SX
G(37)=DD*DD*P23+                   EE*EE*P12+FF*FF*SX
G(38)=GG*DD*P23+(GG*EE-HH*DD)*P13+HH*EE*P12+XI*FF*SX
G(39)=AA*FF*SO+BB*FF*SS
G(40)=DD*FF*SO+EE*FF*SS
G(41)=GG*FF*SO+HH*FF*SS
G(42)=AA*DD*P32-(AA*EE+BB*DD)*P21+BB*EE*P41+CC*FF*SY
G(43)=DD*DD*P32-    2.*DD*EE*P21   +EE*EE*P41+FF*FF*SY
G(44)=GG*DD*P32-(GG*EE+HH*DD)*P21+HH*EE*P41+XI*FF*SY
G(45)=-AA*FF*SQ+BB*FF*SU
G(46)=-DD*FF*SQ+EE*FF*SU
G(47)=-GG*FF*SQ+HH*FF*SU
G(48)=GG*GG*P11+2.*HH*GG*P21+HH*HH*P22+XI*XI*SF
G(49)=-AA*XI*SD+BB*XI*SE
G(50)=-DD*XI*SD+EE*XI*SE
G(51)=-GG*XI*SD+HH*XI*SE
G(52)=AA*GG*P31+(BB*GG-AA*HH)*P13+BB*HH*P33+CC*XI*SM
```

```
G(53)=DD*GG*P31+(EE*GG-DD*HH)*P13+EE*HH*P33+FF*XI*SM
G(54)=GG*GG*P31+                 HH*HH*P33+XI*XI*SM
G(55)=AA*XI*SH+BB*XI*SJ
G(56)=DD*XI*SH+EE*XI*SJ
G(57)=GG*XI*SH+HH*XI*SJ
G(58)=AA*GG*P23+(AA*HH-BB*GG)*P13+BB*HH*P12+CC*XI*SX
G(59)=DD*GG*P23+(DD*HH-EE*GG)*P13+EE*HH*P12+FF*XI*SX
G(60)=GG*GG*P23+                 HH*HH*P12+XI*XI*SX
G(61)=AA*XI*SO+BB*XI*SS
G(62)=DD*XI*SO+EE*XI*SS
G(63)=GG*XI*SO+HH*XI*SS
G(64)=AA*GG*P32-(AA*HH+BB*GG)*P21+BB*HH*P41+CC*XI*SY
G(65)=DD*GG*P32-(DD*HH+EE*GG)*P21+EE*HH*P41+FF*XI*SY
G(66)=GG*GG*P32-   2.*GG*HH*P21   +HH*HH*P41+XI*XI*SY
G(67)=-AA*XI*SQ+BB*XI*SU
G(68)=-DD*XI*SQ+EE*XI*SU
G(69)=-GG*XI*SQ+HH*XI*SU
G(70)=AA*AA*SA-   2.*AA*BB*SB   +BB*BB*SC
G(71)=DD*AA*SA-(DD*BB+EE*AA)*SB+EE*BB*SC
G(72)=GG*AA*SA-(GG*BB+HH*AA)*SB+HH*BB*SC
G(73)=CC*(-AA*SH+BB*SJ)
G(74)=FF*(-AA*SH+BB*SJ)
G(75)=XI*(-AA*SH+BB*SJ)
G(76)=AA*AA*SG+BB*BB*SI
G(77)=DD*AA*SG+EE*BB*SI
G(78)=GG*AA*SG+HH*BB*SI
G(79)=CC*(AA*SO-BB*SS)
G(80)=FF*(AA*SO-BB*SS)
G(81)=XI*(AA*SO-BB*SS)
G(82)=AA*AA*SN+BB*BB*SR
G(83)=DD*AA*SN+EE*BB*SR
G(84)=GG*AA*SN+HH*BB*SR
```

```
G(85)=AA*AA*SP+BB*BB*ST
G(86)=DD*AA*SP+EE*BB*ST
G(87)=GG*AA*SP+HH*BB*ST
G(88)=DD*DD*SA-    2.*DD*EE*SB   +EE*EE*SC
G(89)=GG*DD*SA-(GG*EE+HH*DD)*SB+HH*EE*SC
G(90)=CC*(-DD*SH+EE*SJ)
G(91)=FF*(-DD*SH+EE*SJ)
G(92)=XI*(-DD*SH+EE*SJ)
G(93)=AA*DD*SG+BB*EE*SI
G(94)=DD*DD*SG+EE*EE*SI
G(95)=GG*DD*SG+HH*EE*SI
G(96)=CC*(DD*SO-EE*SS)
G(97)=FF*(DD*SO-EE*SS)
G(98)=XI*(DD*SO-EE*SS)
G(99)= AA*DD*SN+BB*EE*SR
G(100)=DD*DD*SN+EE*EE*SR
G(101)=GG*DD*SN+HH*EE*SR
G(102)=AA*DD*SP+BB*EE*ST
G(103)=DD*DD*SP+EE*EE*ST
G(104)=GG*DD*SP+HH*EE*ST
G(105)=GG*GG*SA-2.*GG*HH*SB+HH*HH*SC
G(106)=CC*(-GG*SH+HH*SJ)
G(107)=FF*(-GG*SH+HH*SJ)
G(108)=XI*(-GG*SH+HH*SJ)
G(109)=AA*GG*SG+BB*HH*SI
G(110)=DD*GG*SG+EE*HH*SI
G(111)=GG*GG*SG+HH*HH*SI
G(112)=CC*(GG*SO-HH*SS)
G(113)=FF*(GG*SO-HH*SS)
G(114)=XI*(GG*SO-HH*SS)
G(115)=AA*GG*SN+BB*HH*SR
G(116)=DD*GG*SN+EE*HH*SR
```

```
G(117)=GG*GG*SN+HH*HH*SR
G(118)=AA*GG*SP+BB*HH*ST
G(119)=DD*GG*SP+EE*HH*ST
G(120)=GG*GG*SP+HH*HH*ST
G(121)=AA*AA*P11-    2.*AA*BB*P21    +BB*BB*P22+CC*CC*SF
G(122)=DD*AA*P11-(DD*BB+EE*AA)*P21+EE*BB*P22+FF*CC*SF
G(123)=GG*AA*P11-(GG*BB+HH*AA)*P21+HH*BB*P22+XI*CC*SF
G(124)=AA*CC*SD+BB*CC*SE
G(125)=DD*CC*SD+EE*CC*SE
G(126)=GG*CC*SD+HH*CC*SE
G(127)=AA*AA*P32+    2.*AA*BB*P21    +BB*BB*P41+CC*CC*SY
G(128)=DD*AA*P32+(DD*BB+EE*AA)*P21+EE*BB*P41+FF*CC*SY
G(129)=GG*AA*P32+(GG*BB+HH*AA)*P21+HH*BB*P41+XI*CC*SY
G(130)=AA*CC*SQ+BB*CC*SU
G(131)=DD*CC*SQ+EE*CC*SU
G(132)=GG*CC*SQ+HH*CC*SU
G(133)=DD*AA*P23+(EE*AA-DD*BB)*P13+EE*BB*P12+FF*CC*SX
G(134)=GG*AA*P23+(HH*AA-GG*BB)*P13+HH*BB*P12+XI*CC*SX
G(135)=DD*DD*P11-    2.*DD*EE*P21    +EE*EE*P22+FF*FF*SF
G(136)=GG*DD*P11-(GG*EE+HH*DD)*P21+HH*EE*P22+XI*FF*SF
G(137)=AA*FF*SD+BB*FF*SE
G(138)=DD*FF*SD+EE*FF*SE
G(139)=GG*FF*SD+HH*FF*SE
G(140)=AA*DD*P32+(AA*EE+BB*DD)*P21+BB*EE*P41+CC*FF*SY
G(141)=DD*DD*P32+    2.*DD*EE*P21    +EE*EE*P41+FF*FF*SY
G(142)=GG*DD*P32+(GG*EE+HH*DD)*P21+HH*EE*P41+XI*FF*SY
G(143)=AA*FF*SQ+BB*FF*SU
G(144)=DD*FF*SQ+EE*FF*SU
G(145)=GG*FF*SQ+HH*FF*SU
G(146)=AA*DD*P23+(BB*DD-AA*EE)*P13+BB*EE*P12+CC*FF*SX
G(147)=DD*DD*P23+                      EE*EE*P12+FF*FF*SX
G(148)=GG*DD*P23+(HH*DD-GG*EE)*P13+HH*EE*P12+XI*FF*SX
```

```
G(149)=GG*GG*P11-2.*GG*HH*P21+HH*HH*P22+XI*XI*SF
G(150)=AA*XI*SD+BB*XI*SE
G(151)=DD*XI*SD+EE*XI*SE
G(152)=GG*XI*SD+HH*XI*SE
G(153)=DD*GG*P32+(DD*HH+EE*GG)*P21+EE*HH*P41+FF*XI*SY
G(154)=GG*GG*P32+    2.*HH*GG*P21   +HH*HH*P41+XI*XI*SY
G(155)=AA*XI*SQ+BB*XI*SU
G(156)=DD*XI*SQ+EE*XI*SU
G(157)=GG*XI*SQ+HH*XI*SU
G(158)=AA*AA*SA+    2.*AA*BB*SB   +BB*BB*SC
G(159)=DD*AA*SA+(DD*BB+EE*AA)*SB+EE*BB*SC
G(160)=GG*AA*SA+(GG*BB+HH*AA)*SB+HH*BB*SC
G(161)=DD*DD*SA+    2.*DD*EE*SB   +EE*EE*SC
G(162)=GG*DD*SA+(GG*EE+HH*DD)*SB+HH*EE*SC
G(163)=GG*GG*SA+    2.*GG*HH*SB   +HH*HH*SC
RETURN
END
```

```
C***********************************************************************
C                         SUBROUTINE SOLVE                             *
C***********************************************************************
      SUBROUTINE SOLVE
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/SSM/SST(594,993)
      COMMON/SOLV/MAXID,LDA
C
C     REDUCE STIFFNESS MATRIX USING THE LINPACK EQUATION SOLVER
C
      CALL SPBFA(SST,LDA,NDOF,MAXID,INFO)
      IF(INFO.EQ.0) GO TO 90
      WRITE(6,100) INFO
  100 FORMAT(///' *** STOP ***  THE LEADING MINOR OF ORDER',I5,2X,'IS NO
     *T POSITIVE DEFINITE'/)
      STOP
C
C     REDUCE FORCE VECTOR AND BACK SOLVE FOR DISPLACEMENTS
C
   90 CALL SPBSL(SST,LDA,NDOF,MAXID,Q)
C     WRITE(6,11)
C     WRITE(6,10) (I,Q(I),I=1,NDOF)
C  10 FORMAT(6(' Q(',I3,')=',F10.7,3X))
C  11 FORMAT(//' GENERALIZED DISPLACEMENTS'/)
      RETURN
      END
```

```
C***********************************************************************
C                           SUBROUTINE FORCE                          *
C***********************************************************************
      SUBROUTINE FORCE(NPLTS)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/FORC/D(24),P21,P32,P11,P22,P31,P33,P12,P13,SA,SB,SC,SD,SE,
     1SF,SG,SH,SI,SJ,SM,SN,SO,SP,SQ,SR,SS,ST,SU,SX,SY,F1,F2,F3,F4,F5,
     2F7,F8,F9,F10,F11,F13,F14,F15,F16,F17,F19,F20,F21,F22,F23,P23,P41
      WRITE(6,100)
      TT=0.
      DO 1 I1=1,NPLTS
         WRITE(6,101) I1
         GO TO (11,12,13),I1
   11    A1=A
         B1=C
         NR=NY
         NC=NX
         GO TO 15
   12    A1=A
         B1=B
         NR=NY
         NC=NZ
         GO TO 15
   13    A1=C
         B1=B
         NR=NX
         NC=NZ
   15    DO 10 I=1,NR
            T=THK(I,I1)
            IF(T.NE.TT.OR.I.EQ.1) CALL GLOBK(5,A1,B1,T)
```

```
          TT=T
          DO 20 J=1,NC
             NN=IOP(I,J,I1)
             DO 30 K=1,24
                L=MCODE(NN,K)
                IF(L.EQ.0) GO TO 31
                D(K)=Q(L)
                GO TO 30
 31             D(K)=0.
 30          CONTINUE
             CALL XKLD(I1)
             II=NOP(I,J,I1)
             JJ=NOP(I,J+1,I1)
             KK=NOP(I+1,J,I1)
             LL=NOP(I+1,J+1,I1)
             WRITE(6,102) NN,II,F1,F2,F3,F4,F5
             WRITE(6,103) JJ,F7,F8,F9,F10,F11
             WRITE(6,103) KK,F13,F14,F15,F16,F17
             WRITE(6,103) LL,F19,F20,F21,F22,F23
 20       CONTINUE
 10    CONTINUE
  1 CONTINUE
100 FORMAT(///' NX = AXIAL FORCE IN THE LOCAL-1 DIRECTION (KIPS)'/
   *          ' NY = AXIAL FORCE IN THE LOCAL-2 DIRECTION (KIPS)'/
   *          ' MX = MOMENT ABOUT THE LOCAL-1 AXIS (KIP-INCHES)'/
   *          ' MY = MOMENT ABOUT THE LOCAL-2 AXIS (KIP-INCHES)'/
   *          ' V  = SHEAR IN THE LOCAL-3 DIRECTION (KIPS)'////)
101 FORMAT(//' INTERNAL ELEMENT FORCES FOR PLATE',I3///' ELEMENT',3X,
   *'NODE',9X,'NX',15X,'NY',15X,'MX',15X,'MY',15X,'V')
102 FORMAT(/I5,I9,5(2X,F15.4))
103 FORMAT(I14,5(2X,F15.4))
    RETURN
    END
```

```
C**********************************************************************
C                             SUBROUTINE XKLD                         *
C**********************************************************************
      SUBROUTINE XKLD(I1)
C     IMPLICIT REAL*8 (A-H,O-Z)
      COMMON Q(993),THK(8,3),XL(3,6),A,B,C,E,VNU,WC,X,Y,Z,MCO
     1DE(192,24),NOP(9,9,3),IOP(8,8,3),NELEM,NNODES,NDOF,IHBW,NX,NY,NZ
      COMMON/FORC/D(24),P21,P32,P11,P22,P31,P33,P12,P13,SA,SB,SC,SD,SE,
     1SF,SG,SH,SI,SJ,SM,SN,SO,SP,SQ,SR,SS,ST,SU,SX,SY,F1,F2,F3,F4,F5,
     2F7,F8,F9,F10,F11,F13,F14,F15,F16,F17,F19,F20,F21,F22,F23,P23,P41
      CALL XLAMDT(I1,XL)
      AA=XL(1,1)
      BB=XL(1,2)
      CC=XL(1,5)
      DD=XL(2,1)
      EE=XL(2,2)
      FF=XL(2,5)
      GG=XL(3,1)
      HH=XL(3,2)
      XI=XL(3,5)
      D1=AA*D(1)+DD*D(2)+GG*D(3)
      D2=BB*D(1)+EE*D(2)+HH*D(3)
      D3=AA*D(4)+DD*D(5)+GG*D(6)
      D4=BB*D(4)+EE*D(5)+HH*D(6)
      D5=CC*D(1)+FF*D(2)+XI*D(3)
      D6=CC*D(4)+FF*D(5)+XI*D(6)
      D7=AA*D(7)+DD*D(8)+GG*D(9)
      D8=BB*D(7)+EE*D(8)+HH*D(9)
      D9=AA*D(10)+DD*D(11)+GG*D(12)
      D10=BB*D(10)+EE*D(11)+HH*D(12)
      D11=CC*D(7)+FF*D(8)+XI*D(9)
      D12=CC*D(10)+FF*D(11)+XI*D(12)
```

```
      D13=AA*D(13)+DD*D(14)+GG*D(15)
      D14=BB*D(13)+EE*D(14)+HH*D(15)
      D15=AA*D(16)+DD*D(17)+GG*D(18)
      D16=BB*D(16)+EE*D(17)+HH*D(18)
      D17=CC*D(13)+FF*D(14)+XI*D(15)
      D18=CC*D(16)+FF*D(17)+XI*D(18)
      D19=AA*D(19)+DD*D(20)+GG*D(21)
      D20=BB*D(19)+EE*D(20)+HH*D(21)
      D21=AA*D(22)+DD*D(23)+GG*D(24)
      D22=BB*D(22)+EE*D(23)+HH*D(24)
      D23=CC*D(19)+FF*D(20)+XI*D(21)
      D24=CC*D(22)+FF*D(23)+XI*D(24)
      F1 = D1*P11+D2*P21+D7*P31+D8*P13+D13*P23-D14*P13+D19*P32-D20*P21
      F2 = D1*P21+D2*P22-D7*P13+D8*P33+D13*P13+D14*P12-D19*P21+D20*P41
      F3 = D3*SA-D4*SB-D5*SD+D9*SG         -D11*SH+D15*SN         +D17*SO
     *    +D21*SP         +D23*SQ
      F4 =-D3*SB+D4*SC+D5*SE         +D10*SI+D11*SJ         +D16*SR-D17*SS
     *            +D22*ST-D23*SU
      F5 =-D3*SD+D4*SE+D5*SF+D9*SH+D10*SJ+D11*SM+D15*SO+D16*SS+D17*SX
     *    -D21*SQ+D22*SU+D23*SY
      F7 = D1*P31-D2*P13+D7*P11-D8*P21+D13*P32+D14*P21+D19*P23+D20*P13
      F8 = D1*P13+D2*P33-D7*P21+D8*P22+D13*P21+D14*P41-D19*P13+D20*P12
      F9 = D3*SG         +D5*SH+D9*SA+D10*SB+D11*SD+D15*SP         -D17*SQ
     *    +D21*SN         -D23*SO
      F10=         D4*SI+D5*SJ+D9*SB+D10*SC+D11*SE         +D16*ST-D17*SU
     *            +D22*SR-D23*SS
      F11=-D3*SH+D4*SJ+D5*SM+D9*SD+D10*SE+D11*SF+D15*SQ+D16*SU+D17*SY
     *    -D21*SO+D22*SS+D23*SX
      F13= D1*P23+D2*P13+D7*P32+D8*P21+D13*P11-D14*P21+D19*P31-D20*P13
      F14=-D1*P13+D2*P12+D7*P21+D8*P41-D13*P21+D14*P22+D19*P13+D20*P33
      F15= D3*SN         +D5*SO+D9*SP         +D11*SQ+D15*SA+D16*SB-D17*SD
     *    +D21*SG         -D23*SH
```

```
      F16=        D4*SR+D5*SS        +D10*ST+D11*SU+D15*SB+D16*SC-D17*SE
     *            +D22*SI-D23*SJ
      F17= D3*SO-D4*SS+D5*SX-D9*SQ-D10*SU+D11*SY-D15*SD-D16*SE+D17*SF
     *     +D21*SH-D22*SJ+D23*SM
      F19= D1*P32-D2*P21+D7*P23-D8*P13+D13*P31+D14*P13+D19*P11+D20*P21
      F20=-D1*P21+D2*P41+D7*P13+D8*P12-D13*P13+D14*P33+D19*P21+D20*P22
      F21= D3*SP       -D5*SQ+D9*SN        -D11*SO+D15*SG        +D17*SH
     *     +D21*SA-D22*SB+D23*SD
      F22=        D4*ST+D5*SU        +D10*SR+D11*SS        +D16*SI-D17*SJ
     *     -D21*SB+D22*SC-D23*SE
      F23= D3*SQ-D4*SU+D5*SY-D9*SO-D10*SS+D11*SX-D15*SH-D16*SJ+D17*SM
     *     +D21*SD-D22*SE+D23*SF
      RETURN
      END
```

```
C*********************************************************************
C                              SUBROUTINE SPBFA                      *
C*********************************************************************
      SUBROUTINE SPBFA (SST,LDA,NDOF,MAXID,INFO)
      REAL SST(LDA,1)
      DO 30 J=1,NDOF
      INFO=J
      S=0.0
      IK=MAXID +1
      JK=MAXO (J-MAXID,1)
      MU=MAXO (MAXID+2-J,1)
      IF(MAXID.LT.MU) GO TO 20
      DO 10 K=MU,MAXID
      T=SST(K,J)-SDOT(K-MU,SST(IK,JK),1,SST(MU,J),1)
      T=T/SST(MAXID+1,JK)
      SST(K,J)=T
      S=S+T*T
      IK=IK-1
      JK=JK+1
 10   CONTINUE
 20   CONTINUE
      S=SST(MAXID+1,J)-S
      IF(S.LE.0.0) GO TO 40
      SST(MAXID+1,J)=SQRT(S)
 30   CONTINUE
      INFO=0
 40   CONTINUE
      RETURN
      END
```

```
C*****************************************************************************
C                          SUBROUTINE SPBSL                                  *
C*****************************************************************************
      SUBROUTINE SPBSL(SST,LDA,NDOF,MAXID,Q)
      REAL SST(LDA,1),Q(1)
C
C        FORWARD REDUCTION OF CONSTANTS
C
      DO 10 K=1,NDOF
      LM=MIN0 (K-1,MAXID)
      LA=MAXID+1-LM
      LB=K-LM
      T=SDOT(LM,SST(LA,K),1,Q(LB),1)
      Q(K)=(Q(K)-T)/SST(MAXID+1,K)
 10   CONTINUE
C
C     BACKSUBSTITUTION
C
      DO 20 KB=1,NDOF
      K=NDOF +1-KB
      LM=MIN0(K-1,MAXID)
      LA=MAXID+1-LM
      LB=K-LM
      Q(K)=Q(K)/SST(MAXID+1,K)
      T=-Q(K)
      CALL SAXPY(LM,T,SST(LA,K),1,Q(LB),1)
 20   CONTINUE
      RETURN
      END
```

```
C***********************************************************************
C                              FUNCTION SDOT                           *
C***********************************************************************
      FUNCTION SDOT(N,SX,INCX,SY,INCY)
      REAL SX(1),SY(1)
      STEMP=0.0
      SDOT=0.0
      IF(N.LE.0) GO TO 70
      IF(INCX.EQ.1.AND.INCY.EQ.1) GO TO 20
      IX=1
      IY=1
      IF(INCX.LT.0) IX=(-N+1)*INCX+1
      IF(INCY.LT.0) IY=(-N+1)*INCY+1
      DO 10 I=1,N
      STEMP=STEMP+SX(IX)*SY(IY)
      IX=IX+INCX
      IY=IY+INCY
 10   CONTINUE
      SDOT=STEMP
      GO TO 70
 20   M=MOD(N,5)
      IF(M.EQ.0) GO TO 40
      DO 30 I=1,M
      STEMP=STEMP+SX(I)*SY(I)
 30   CONTINUE
      IF(N.LT.5) GO TO 60
 40   MP1=M+1
      DO 50 I=MP1,N,5
      STEMP=STEMP+SX(I)*SY(I)+SX(I+1)*SY(I+1)+SX(I+2)*SY(I+2)+SX(I+3)*SY
     1(I+3)+SX(I+4)*SY(I+4)
 50   CONTINUE
 60   SDOT=STEMP
```

```
70    CONTINUE
      RETURN
      END
```

```
C***********************************************************************
C                           SUBROUTINE SAXPY                           *
C***********************************************************************
      SUBROUTINE SAXPY(N,SA,SX,INCX,SY,INCY)
      REAL SX(1),SY(1),SA
      IF(N.LE.0) RETURN
      IF(SA.EQ.0.0) RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1) GO TO 20
      IX=1
      IY=1
      IF(INCX.LT.0)IX=(-N+1)*INCX+1
      IF(INCY.LT.0)IY=(-N+1)*INCY+1
      DO 10 I=1,N
      SY(IY)=SY(IY)+SA*SX(IX)
      IX=IX+INCX
      IY=IY+INCY
   10 CONTINUE
   20 M=MOD(N,4)
      IF(M.EQ.0) GO TO 40
      DO 30 I=1,M
      SY(I)=SY(I)+SA*SX(I)
   30 CONTINUE
      IF(N.LT.4) RETURN
   40 MP1=M+1
      DO 50 I=MP1,N,4
      SY(I)=SY(I)+SA*SX(I)
      SY(I+1)=SY(I+1) + SA*SX(I+1)
      SY(I+2)=SY(I+2) + SA*SX(I+2)
      SY(I+3)=SY(I+3) + SA*SX(I+3)
   50 CONTINUE
      RETURN
      END
```

```
C*********************************************************************
C                            SAMPLE INPUT DATA                       *
C*********************************************************************
3
120. 120. 120. 8 8 8
3000. 0.2 150. 0.
10. 10. 10. 10. 12.
137 2
146 2
155 2
164 2
173 2
182 2
191 2
200 2
209 2
210 2
211 2
212 2
213 2
214 2
215 2
216 2
217 2
0 0
1 1 -3 62.4 120.
1 2 -1 62.4 120.
3 0 0 0. 0.
0 0 0 0. 0.
0 0 0.
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
22,23,24 ,2,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42
```

43,44,45,46,47 ,3,26,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62
63,64,65,66,67,68,69 ,4,27,49,70,71,72,73,74,75,76,77,78,79,80,81
82,83,84,-22,-45,-67,85,86,87 ,5,28,50,71,88,89,90,91,92,93,94,95
96,97,98,99,100,101,-23,-46,-68,102,103,104 ,6,29,51,72,89,105,106
107,108,109,110,111,112,113,114,115,116,117,-24,-47,-69,118,119
120 ,7,30,52,73,90,106,121,122,123,124,125,126,127,128,129,130
131,132,13,133,134,-79,-96,-112 ,8,31,53,74,91,107,122,135,136
137,138,139,140,141,142,143,144,145,146,147,148,-80,-97,-113
9,32,54,75,92,108,123,136,149,150,151,152,129,153,154,155,156
157,15,38,60,-81,-98,-114 ,10,33,55,76,93,109,124,137,150,158
159,160,-130,-143,-155,85,86,87,-16,-39,-61,82,83,84 ,11,34,56
77,94,110,125,138,151,159,161,162,-131,-144,-156,102,103,104
-17,-40,-62,99,100,101 ,12,35,57,78,95,111,126,139,152,160,162
163,-132,-145,-157,118,119,120,-18,-41,-63,115,116,117 ,13,36
58,79,96,112,127,140,129,-130,-131,-132,121,122,123,-124,-125
-126,7,30,52,-73,-90,-106 ,14,37,59,80,97,113,128,141,153,-143
-144,-145,122,135,136,-137,-138,-139,8,31,53,-74,-91,-107 ,15
38,60,81,98,114,129,142,154,-155,-156,-157,123,136,149,-150
-151,-152,9,32,54,-75,-92,-108 ,16,39,61,82,99,115,130,143,155
85,102,118,-124,-137,-150,158,159,160,-10,-33,-55,76,77,78 ,17
40,62,83,100,116,131,144,156,86,103,119,-125,-138,-151,159,161
162,-11,-34,-56,93,94,95 ,18,41,63,84,101,117,132,145,157,87
104,120,-126,-139,-152,160,162,163,-12,-35,-57,109,110,111 ,19
42,64,-22,-23,-24,13,146,15,-16,-17,-18,7,8,9,-10,-11,-12,1,2
3,-4,-5,-6 ,20,43,65,-45,-46,-47,133,147,38,-39,-40,-41,30,31
32,-33,-34,-35,2,25,26,-27,-28,-29 ,21,44,66,-67,-68,-69,134
148,60,-61,-62,-63,52,53,54,-55,-56,-57,3,26,48,-49,-50,-51 ,22
45,67,85,102,118,-79,-80,-81,82,99,115,-73,-74,-75,76,93,109
-4,-27,-49,70,71,72 ,23,46,68,86,103,119,-96,-97,-98,83,100,116
-90,-91,-92,77,94,110,-5,-28,-50,71,88,89 ,24,47,69,87,104,120
-112,-113,-114,84,101,117,-106,-107,-108,78,95,111,-6,-29,-51
72,89,105

The vita has been removed from
the scanned document

ANALYSIS OF RECTANGULAR CONCRETE TANKS

CONSIDERING

INTERACTION OF PLATE ELEMENTS

Douglas G. Fitzpatrick

Abstract

This study developed a finite element program suitable for analyzing one quarter of a rectangular tank. A rectangular plate element capable of both extension and flexure was used with appropriate coordinate transformations to enable interaction of the floor and wall plates.

Moment values throughout the tank were determined but not collected into tables because of their dependence on the width-to-length ratios and the height of the tank. A moment distribution type of method was developed so that critical vertical moment values could be rapidly determined without the direct use of a complex computer program.