



Interaction of superstructure, foundation, and soil
by John Joseph Hightower

A thesis submitted in partial fulfillment of the requirements for the degree MASTER OF SCIENCE in
CIVIL ENGINEERING

Montana State University

© Copyright by John Joseph Hightower (1981)

Abstract:

A Fortran computer program is developed, to study the interaction of structural plate elements and their supporting subgrades. A linear spring (Winkler) subgrade model is used as well as the option of considering rigid body action of specified areas of the foundation.

Example problems are considered; shear deformation is determined to be a significant contributor to the deflection profile. Contact pressures are found to agree with pressures predicted by other methods. Finally, shear stresses and bending moments are shown to accurately represent values obtained by traditional methods.

STATEMENT OF PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Montana State University, I agree that the Library shall make it freely available for inspection. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by my major professor, or, in his absence, by the director of libraries. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature

John Joseph H. Tower

Date

July 16, 1981

INTERACTION OF SUPERSTRUCTURE, FOUNDATION,
AND SOIL

by

JOHN JOSEPH HIGHTOWER

A thesis submitted in partial fulfillment
of the requirements for the degree

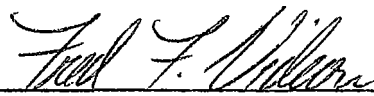
of

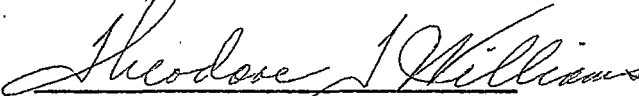
MASTER OF SCIENCE

in

CIVIL ENGINEERING

Approved:


Chairperson, Graduate Committee


Head, Major Department


Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

July 1981

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. Fred F. Videon and Dr. Robert G. Oakberg for their efforts and contributions toward the completion of this investigation.

The author also wishes to thank Bruce A. Suprenant, instructor, who originally inspired this investigation and was responsible for obtaining the required financial support.

TABLE OF CONTENTS

SUBJECT	PAGE
VITA	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	v
LIST OF FIGURES	vi
ABSTRACT	vii
INTRODUCTION	1
ISOPARAMETRIC FINITE ELEMENTS	4
SUBGRADE MODELS	14
Integrated Winkler Subgrade	15
Lumped Winkler Subgrade	19
RIGID BODY ACTION	23
RESULTS	30
Circular Plate With Point Load	30
Notched Rectangular Plate	33
Cantilever Footing Analysis	36
SUMMARY AND CONCLUSIONS	42
RECOMMENDATIONS	44
REFERENCES	48
APPENDICES	
Appendix A: Instructions for Preparing Input Data	50
Appendix B: Program listing	55
Appendix C: Cantilever Footing	84

LIST OF TABLES

TABLE		PAGE
1	MODULUS OF SUBGRADE REACTION	22
2	SOIL CONTACT PRESSURES	35

LIST OF FIGURES

FIGURE		PAGE
1	SIGN CONVENTION FOR PLATE ELEMENT	6
2	LOCAL COORDINATE AXIS AND NODE NUMBERS	10
3	CORNER AND MIDSIDE NODE SHAPE FUNCTIONS	10
4	NATURAL NODAL COORDINATES	17
5	TRIBUTARY AREAS FOR CORNER AND MIDSIDE NODES	20
6	CIRCULAR PLATE: ELEMENT GRID AND NODAL NUMBERING	32
7	CIRCULAR PLATE: DEFLECTION PROFILE	32
8	NOTCHED FOOTING: ELEMENT GRID	35
9	CANTILEVER FOOTING: GEOMETRY AND LOADING	38
10	CANTILEVER FOOTING: ELEMENT GRID	38
11	CANTILEVER FOOTING: SHEAR FORCE AND BENDING MOMENT DIAGRAMS--SECTIONAL	39
12	CANTILEVER FOOTING: SHEAR FORCE AND BENDING MOMENT DIAGRAMS--TOTAL	41

ABSTRACT

A Fortran computer program is developed to study the interaction of structural plate elements and their supporting subgrades. A linear spring (Winkler) subgrade model is used as well as the option of considering rigid body action of specified areas of the foundation.

Example problems are considered; shear deformation is determined to be a significant contributor to the deflection profile. Contact pressures are found to agree with pressures predicted by other methods. Finally, shear stresses and bending moments are shown to accurately represent values obtained by traditional methods.

INTRODUCTION

To a design engineer responsible for the design of a structural footing or series of footings, it is desirable that the design be done quickly, easily, and accurately. The purpose of this paper is to develop a computer program to assist the design engineer faced with this situation, and to investigate the load response of a footing supported by an elastic subgrade.

For a given footing design problem there are several parameters that must be considered by the design engineer. These parameters include loading patterns, size limitations, and allowable contact stresses between the footing and its supporting subgrade. The design engineer must accurately model the footing and the subgrade so that a safe footing design, satisfying these parameters, can be produced.

For small footing design it is commonly assumed that the footing behaves as a rigid body (1,9). Then, by assuming a linear elastic subgrade, it is possible to obtain a linear contact pressure distribution across the bottom of the footing. To obtain shear forces and bending moments, the footing can then be analysed using the equilibrium conditions. Since deformation of the footing is not allowed, the rigid footing model will tend to be conservative in the estimation of contact pressures between

the footing and the subgrade. The conservative estimate of contact stresses may require more materials to be used in the construction of the footing than is necessary. This is due to the greater area of footing required to reduce contact stresses to an allowable value.

This paper will examine a more accurate method of modelling the footing and subgrade than the rigid footing and linear elastic subgrade model. This is accomplished by modelling the subgrade as an elastic material and the footing as a deformable body. Due to computational difficulties and high analysis costs of more representative subgrade models such as the Boussinesq or Pasternak, a linear spring (Winkler) subgrade model is used.

If deformation of the footing is allowed, it becomes necessary to account for changes in the footing stiffness caused by foundation walls or columns. Areas of the footing where this contact occurs will experience little or no deformation. To approximate the increase in stiffness, the capability to model foundation walls or columns as rigid bodies is considered

Foundations supported by a Winkler subgrade will displace uniformly downward when subjected to a uniform loading over the entire surface. This response is not

reflective of true soil behavior. Therefore, the use of this program is limited to the analysis of spread and combined footings.

ISOPARAMETRIC FINITE ELEMENTS

The finite element computer program developed is based on the eight-node quadrilateral isoparametric element. Hinton and Owen, 1977, developed a plate analysis program, BOK1, based on this element (5). This program is used as a basis for the footing analysis program developed in this paper. BOK1 includes the effects of shear deformation, thus allowing the analysis of thick plates.

For the development of BOK1 the following assumptions are made:

- 1) The deflections of the plate are small.
- 2) Normals to the midsurface before deformation remain straight but not necessarily normal to the midsurface after deformation.
- 3) Stresses normal to the midsurface are negligible irrespective of the loading.

The development of the element stiffness matrix is perhaps the most important aspect of the finite element method and is displayed here as taken from (5). For complete documentation of the program and the theory of

isoparametric elements, the reader is referred to (5).

Development of the Element Stiffness Matrix

For the derivation of the stiffness matrix the following notation is used, (The sign convention for this element is illustrated in Fig. 1).

$\bar{\delta}$ = displacement vector

w, θ_x, θ_y = components of $\bar{\delta}$
(including shear deformation)

$\bar{M}$ = vector of bending moments

m_x, m_y, m_{xy} = components of $\bar{M}$

$\bar{Q}$ = vector of shearing forces

q_x, q_y = components of $\bar{Q}$

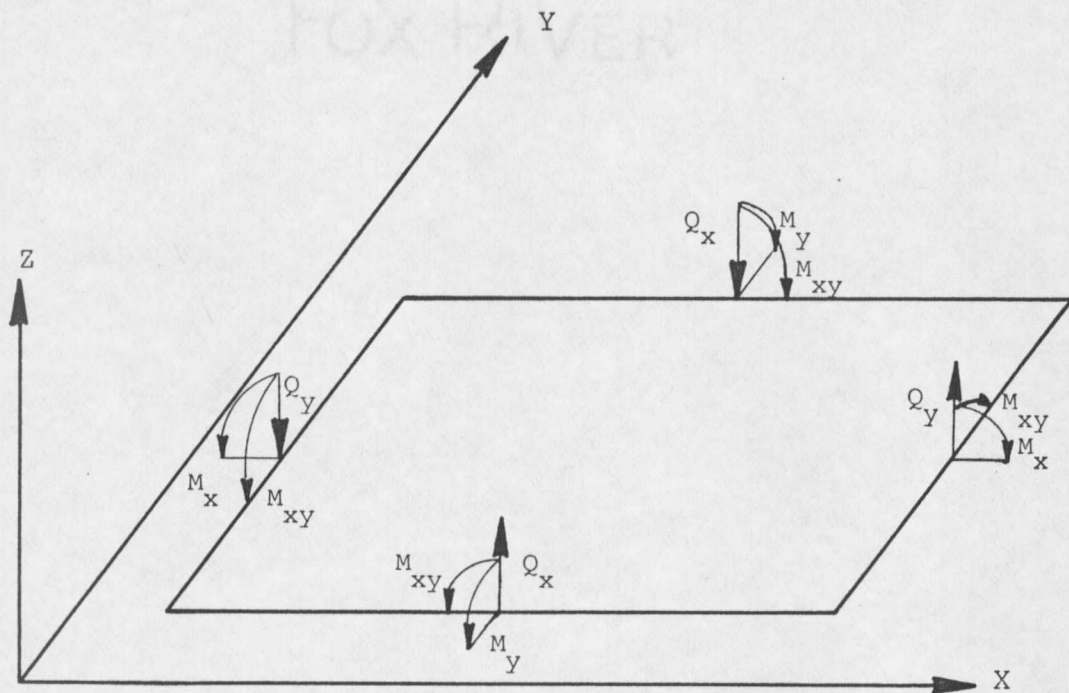
$\bar{X}$ = vector of bending deformation

x_x, x_y, x_{xy} = components of bending deformation

$\bar{\phi}$ = vector of curvatures due to shear

ϕ_x, ϕ_y = components of $\bar{\phi}$

Additionally, the following relationships are established:



SIGN CONVENTION FOR PLATE ELEMENT

FIGURE 1

$$\bar{\delta} = \begin{Bmatrix} w \\ e_x \\ e_y \end{Bmatrix} = \begin{Bmatrix} w \\ \partial w / \partial x + \phi_x \\ \partial w / \partial y + \phi_y \end{Bmatrix}$$

$$\bar{M} = \begin{Bmatrix} m_x \\ m_y \\ m_{xy} \end{Bmatrix} = \begin{Bmatrix} \sigma_{xz} z dz \\ \sigma_{yz} z dz \\ \tau_{xy} z dz \end{Bmatrix}$$

$$\bar{Q} = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = \begin{Bmatrix} \tau_{xz} dz \\ \tau_{yz} dz \end{Bmatrix}$$

$$\bar{X} = \begin{Bmatrix} x_x \\ x_y \\ x_{xy} \end{Bmatrix} = \begin{Bmatrix} -\partial e_x / \partial x \\ -\partial e_y / \partial y \\ -(\partial e_x / \partial y + \partial e_y / \partial x) \end{Bmatrix}$$

The total potential energy of the element is given by

$$\Pi = 0.5 \int_A (m_x x_x + m_y x_y + m_{xy} x_{xy} + q_x \phi_x + q_y \phi_y) dA - \int_A q w dA \quad (1)$$

The stress-strain relationships may be written as

$$M = D_f \bar{X}$$

$$Q = D_s \bar{\phi}$$

where D_f is the matrix of flexural rigidities and D_s is the matrix of shear rigidities and are defined as follows:

$$D_f = Et^3 / (12(1-\nu^2)) \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{vmatrix}$$

$$D_s = Et / (2.4(1+\nu)) \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

Substituting the stress strain relationships into eq. (1) yields

$$= 0.5 \int_A ([X]^t D_f [X] + [\theta]^t D_s [\theta]) dA - \int_A q w dA$$

For isoparametric formulation, element shapes are expressed by the same function as the assumed displacement field and are expressed as

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \sum_{i=1}^8 Ni[I] \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} \quad (2)$$

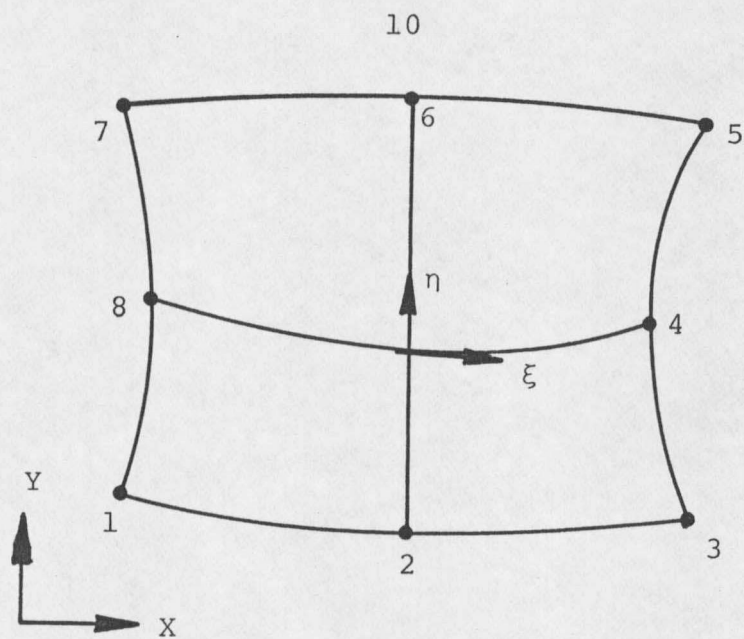
and

$$\delta = \sum_{i=1}^8 N_i[I] \delta_i \quad \text{where } \delta_i = [w_i, e_{xi}, e_{yi}]^t \quad (3)$$

where N_i is the element shape function for the i th node. The element shape functions are

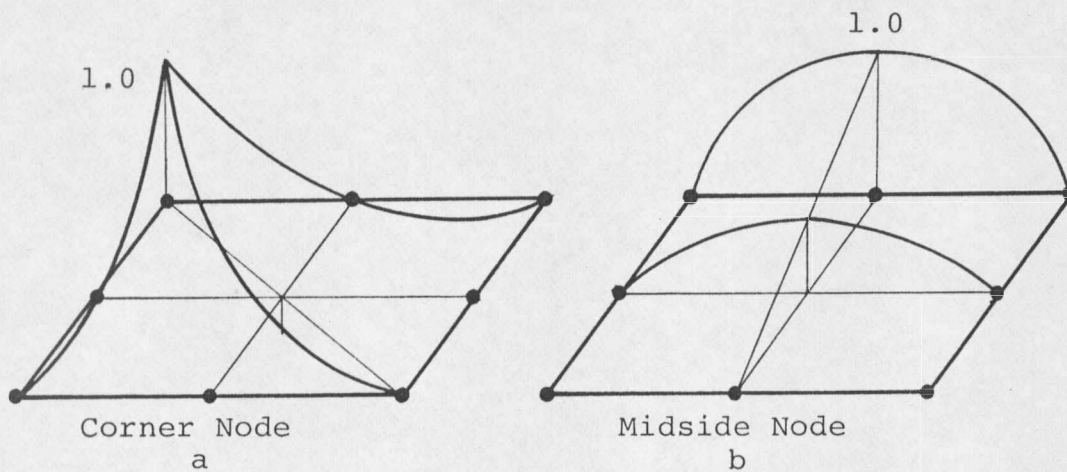
$$\begin{aligned} N1 &= 0.25(1 - \xi\eta - \xi^2 + \xi^2\eta + \xi\eta^2 + \eta^2) \\ N2 &= 0.50(1 - \eta - \xi^2 + \xi^2\eta) \\ N3 &= 0.25(-1 - \xi\eta + \xi^2 - \xi^2\eta + \xi\eta^2 + \eta^2) \\ N4 &= 0.50(1 + \xi - \xi\eta^2 - \eta^2) \\ N5 &= 0.25(-1 + \xi\eta + \xi^2\eta + \xi\eta^2 + \eta^2) \\ N6 &= 0.50(1 + \eta - \xi^2 - \xi^2\eta) \\ N7 &= 0.25(-1 - \xi\eta + \xi^2 + \xi^2\eta - \xi\eta^2 + \eta^2) \\ N8 &= 0.50(1 - \xi + \xi\eta^2 - \eta^2) \end{aligned} \quad (4)$$

ξ and η are local, natural coordinate axis as defined by Fig. 2. The shape functions have the characteristic of equaling 1.0 at their corresponding node, and identically 0.0 at all other nodes. The characteristic shape of corner and midside node shape functions are illustrated in Fig. 3(a,b).



LOCAL COORDINATE AXIS AND NODE NUMBERS

FIGURE 2



CORNER AND MIDSIDE NODE SHAPE FUNCTIONS

FIGURE 3

The strain-displacement relationship for the quadrilateral isoparametric element under consideration is

$$E = \sum_{i=1}^8 B_i \partial_i$$

where

$$[E]^t = [[\bar{X}]^t, [\bar{\theta}]^t],$$

and

$$[B_i] = \begin{Bmatrix} B_{fi} \\ B_{si} \end{Bmatrix} = \begin{vmatrix} 0 & -\partial Ni/\partial x & 0 \\ 0 & 0 & -\partial Ni/\partial y \\ 0 & -\partial Ni/\partial y & -\partial Ni/\partial x \\ \hline \partial Ni/\partial x & -Ni & 0 \\ \partial Ni/\partial y & 0 & -Ni \end{vmatrix}$$

To assemble the strain matrix it is necessary to obtain the derivatives of the shape functions with respect to the cartesian coordinate system. To do this the chain rule of differentiation is used

$$\partial Ni/\partial x = (\partial Ni/\partial \xi)(\partial \xi/\partial x) + (\partial Ni/\partial \eta)(\partial \eta/\partial x) \quad (5)$$

$$\partial Ni / \partial y = (\partial Ni / \partial \xi) (\partial \xi / \partial y) + (\partial Ni / \partial \eta) (\partial \eta / \partial y) \quad (6)$$

or, in a matrix formulation

$$\begin{Bmatrix} \partial Ni / \partial x \\ \partial Ni / \partial y \end{Bmatrix} = \begin{vmatrix} \partial \xi / \partial x & \partial \xi / \partial y \\ \partial \eta / \partial x & \partial \eta / \partial y \end{vmatrix} \begin{Bmatrix} \partial Ni / \partial \xi \\ \partial Ni / \partial \eta \end{Bmatrix} \quad (7)$$

inverting this matrix yields

$$\begin{Bmatrix} \partial Ni / \partial \xi \\ \partial Ni / \partial \eta \end{Bmatrix} = \begin{vmatrix} \partial x / \partial \xi & \partial y / \partial \xi \\ \partial x / \partial \eta & \partial y / \partial \eta \end{vmatrix} \begin{Bmatrix} \partial Ni / \partial x \\ \partial Ni / \partial y \end{Bmatrix} = [J] \begin{Bmatrix} \partial Ni / \partial x \\ \partial Ni / \partial y \end{Bmatrix} \quad (8)$$

where the matrix [J] is known as the Jacobian Matrix.

To evaluate the Jacobian it is necessary to determine the values of $\partial x / \partial \xi$, $\partial y / \partial \xi$, $\partial x / \partial \eta$, and $\partial y / \partial \eta$. Differentiation of eq. 2 yields

$$\partial x / \partial \xi = \sum_{i=1}^8 (\partial Ni / \partial \xi) x_i + Ni (\partial x_i / \partial \eta)$$

since x is a constant, $\partial x / \partial \xi = 0.0$, this yields

$$\partial x / \partial \xi = \sum_{i=1}^8 (\partial N_i / \partial \xi) x_i \quad (9)$$

a similar procedure is used to evaluate $\partial y / \partial \xi$, $\partial x / \partial \eta$, and $\partial y / \partial \eta$.

When $\partial x / \partial \xi$, $\partial y / \partial \xi$, $\partial x / \partial \eta$, and $\partial y / \partial \eta$ are known, the determinant of the Jacobian Matrix is evaluated. If the determinant is less than or equal to zero, there is an error in the coordinate mapping procedure and the element grid selected must be reviewed for errors. Once the Jacobian is known, it is inverted to obtain the shape function derivatives required in the assembly of the strain matrix.

To obtain the element stiffness matrix the following function must be evaluated:

$$[k] = \int_A [B]^T [D] [B] dA = \iint_{\eta, \xi} [B] [D] [B] \det[J] d\xi d\eta \quad (10)$$

To integrate this function a numerical method known as Gauss Quadrature is used. The option of using either a 2 x 2 (4 gauss points) or a 3 x 3 (9 gauss points) rule is available to the user. A review of this integration technique is presented in (4,5).

SUBGRADE MODELS

Since 1867 when E. Winkler modeled an elastic subgrade as closely spaced, independent springs, numerous models for subgrades have been developed. The most common of the more advanced models is based upon the Boussinesq equation for a point load on an elastic half space (3). Other models include: Pasternak, a Winkler model with a plate element bonded to its surface that can deflect only in shear; Hetenyi, a Winkler model with a plate element bonded to its surface that can deflect only in bending; Filonenko-Borodich, a Winkler model with a stretched membrane bonded to its surface; a Generalized model, which consists of a Winkler model with the additional aspect of moment terms proportional to the rotation of a point (6).

Due to computational difficulties encountered, only the Winkler model will be considered. For this model the force exerted is linearly proportional to the deflection of the subgrade.

Two methods of modelling the Winkler subgrade are considered. The first is a more precise method of actually integrating an assumed displacement field over the elemental area. The second is to "lump" the subgrade stiffness at each node, which result in a diagonal subgrade stiffness matrix.

Integrated Winkler Subgrade Model

The proportionality constant for a Winkler subgrade is commonly denoted as B , or "the modulus of subgrade reaction". For the integrated model it is necessary to assume a displacement field. For consistency the displacement field used is the same as used for the plate element. The assumed displacement field for a given point on the element can be described by the function

$$w = a_1 + a_2\xi + a_3\eta + a_4\xi^2 + a_5\xi\eta + a_6\eta^2 + a_7\xi^2\eta + a_8\xi\eta^2 \quad (11)$$

or, for the entire subgrade element

$$w_i = \sum_{i=1}^8 f(\xi_i, \eta_i) a_i \quad (12)$$

where $f(\xi_i, \eta_i) = [1 \ \xi_i \ \eta_i \ \xi_i^2 \ \xi_i\eta_i \ \eta_i^2 \ \xi_i^2\eta_i \ \xi_i\eta_i^2]$

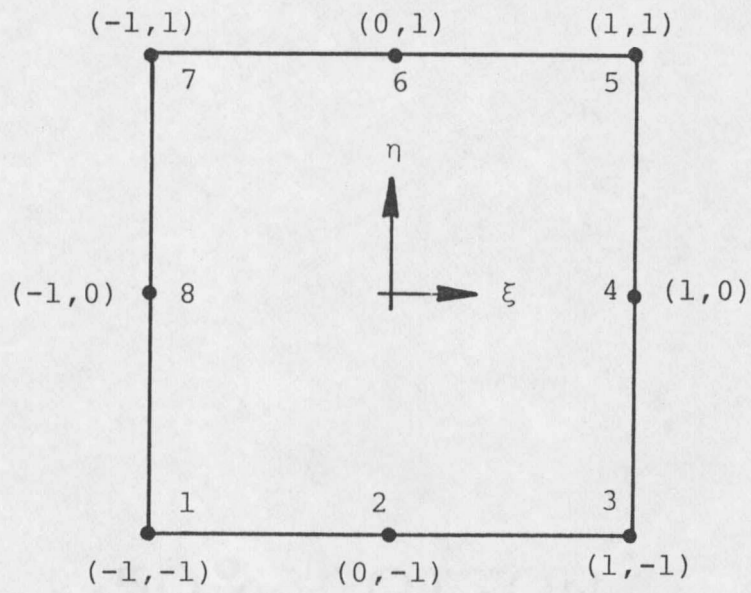
The local nodal numbering scheme and coordinates are illustrated in Fig. 4. Substituting the nodal coordinate values for ξ and η into the matrix $F(\xi, \eta)$ where

$$F(\xi, \eta) = \sum_{i=1}^8 [f(\xi_i, \eta_i)] \quad (13)$$

yields the following matrix:

$$\begin{vmatrix} 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 & -1 & 1 & 1 & -1 \\ 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \end{vmatrix}$$

To solve for the unknown constants it is necessary to invert the above matrix. This yields



NATURAL NODAL COORDINATES

FIGURE 4

$$\begin{vmatrix}
 -0.25 & 0.50 & -0.25 & 0.50 & -0.25 & 0.50 & -0.25 & 0.50 \\
 0.00 & 0.00 & 0.00 & 0.50 & 0.00 & 0.00 & 0.00 & -0.50 \\
 0.00 & -0.50 & 0.00 & 0.00 & 0.00 & 0.50 & 0.00 & 0.00 \\
 0.25 & -0.50 & 0.25 & 0.00 & 0.25 & -0.50 & 0.25 & 0.00 \\
 0.25 & 0.00 & -0.25 & 0.00 & 0.25 & 0.00 & -0.25 & 0.00 \\
 0.25 & 0.00 & 0.25 & -0.25 & 0.25 & 0.00 & 0.25 & -0.50 \\
 -0.25 & 0.50 & -0.25 & 0.00 & 0.25 & -0.50 & 0.25 & 0.00 \\
 -0.25 & 0.00 & 0.25 & -0.50 & 0.25 & 0.00 & -0.25 & 0.50
 \end{vmatrix}$$

Multiplying the above matrix by the displacement vector yields the shape functions recorded earlier (Eq. 4). Using the potential energy of the subgrade

$$U = 0.5 \int_A w dA \quad (14)$$

the following relationship is established:

$$[k] = \int_A [N]^t [N] dA = \iint_{\eta \xi} [N]^t [N] \det[J] d\xi d\eta \quad (15)$$

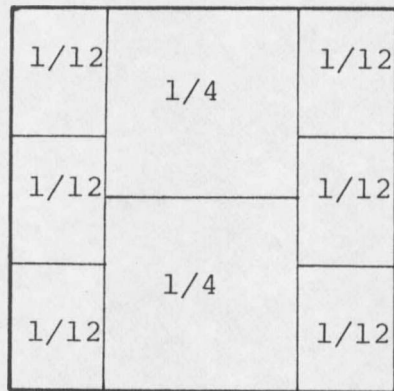
The values for the shape function at a given point (a gauss point) are computed in the subroutine STIFPB. This makes the determination of the subgrade stiffness matrix by the

integrated method an easily implemented procedure.

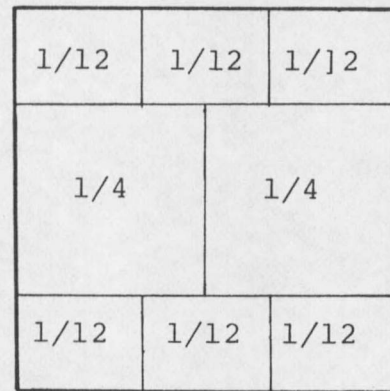
Lumped Winkler Subgrade

A simpler version of the Winkler model is the lumped model. This model consists of a diagonal stiffness matrix whose terms are evaluated by multiplying the subgrade modulus, B , by a tributary area associated with each node. To establish the values of the tributary area the following empirical derivation is presented:

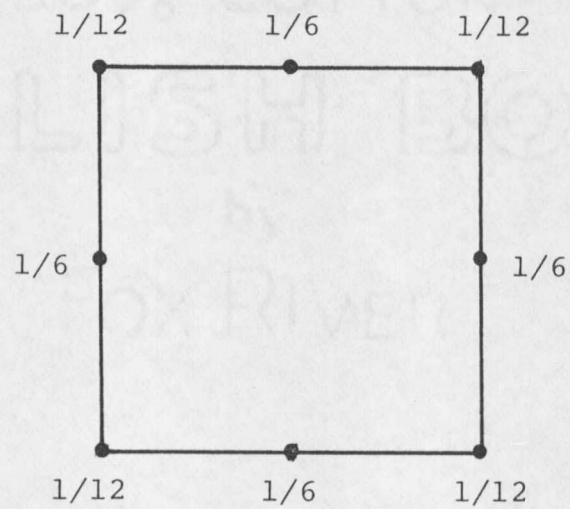
- 1) Divide the plate area into sections as shown in Fig. 5a and compute the tributary areas to be associated with each node.
- 2) Rotate the element 90° and repeat the first step, Fig. 5b.
- 3) Sum the tributary areas for each node given in steps 1, and 2 and divide this sum by 2.0. This gives an approximate tributary area to be associated with each node, Fig. 5c



a



b



c

TRIBUTARY AREAS FOR CORNER AND MIDSIDE NODES
FIGURE 5

This yields tributary stiffnesses of $(1/12)$ times the area for corner nodes, and $(1/6)$ times the area for midside nodes. Since the area is computed in the subroutine STIFPB the addition of this option is also easily accomplished.

The critical factor in using the Winkler model is selecting a value for the subgrade modulus that accurately represents the subgrade. Table 1 gives some guidelines to follow in the selection of the subgrade modulus. It must be realized that this table cannot override the necessity of performing an analysis of the subgrade to be used.

SUBGRADE MODULUS, B, Kips/cubic inch															
100		150				200		250				500			800
General soil rating as subgrade,subbase or base															
Very poor subgrade		Poor subgrade		Fair to good subgrade			Best subgrde		Good subbase			GoodBest basebase			
G-Gravel											GW				
S-Sand											GC				
M-"Mo",very fine sand,silt															
C-Clay											GP				
F-Fines,material less than 0.1mm											GF				
O-Organic															
W-Well graded												SW			
P-Poorly graded												SC			
L-Low to medium compressibility															
H-High compressibility				SP											
		SF													
CH		ML													
OH		CL													
		OL													
MH															

TABLE 1

RIGID BODY ACTION

Areas of the footing in contact with columns or foundation walls will experience little or no in plane deformation. To model this action a plane strain/plane stress element could be used. This modification would require the use of a translated stiffness matrix. The stiffness of the entire wall would have to be condensed or translated to an equivalent stiffness along the surface of the plate. Then, rigid body translations would have to be implemented to translate these terms to the neutral axis of the plate. Axial degrees of freedom would have to be added to the plate in order to properly merge the stiffness matrices of the plate with those of the wall. This would require the expansion of the plate elemental stiffness matrix to a 40 x 40 matrix. Also, the additional stiffnesses added to the system will result in an ill-conditioned stiffness matrix if the wall or column is very stiff compared to the footing. This leads to numerical problems in obtaining the solution.

A simpler method of modelling foundation walls and columns is to treat them as pure rigid bodies. This reduces the degrees of freedom for all nodes in contact with the rigid body to a total of 3. To incorporate the rigid body modes into the program it is necessary to identify all nodes

on a given rigid body. Then, before merging the elemental stiffness matrix into the global matrix, the elemental matrix must be transformed to reduce the degrees of freedom. The number of degrees of freedom for an element will be $(3(8-N) + 3)$ where N is the number of nodes in contact with a rigid body. It is important to understand the development of the transformation matrix.

From basic engineering mechanics it is known that the relative velocities of two points on a rigid body can be expressed as

$$\bar{V}_B = \bar{V}_A + \bar{\Theta} \times \bar{r}_{B/A} \quad (16)$$

This expression can be integrated with respect to time, eliminating a time dependency, to achieve an expression relating relative displacements of two points on a rigid body.

$$\bar{W}_B = \bar{W}_A + \bar{\Theta} \times \bar{r}_{B/A} \quad (17)$$

Expanding this expression by putting it into component form gives

$$\begin{aligned}
w_{\bar{k}B} &= w_{\bar{k}B} + (\theta_{x\bar{i}} + \theta_{y\bar{j}}) \times (r_{x\bar{i}} + r_{y\bar{j}})_{B/A} \\
&= w_{\bar{k}B} + r_y \theta_{x\bar{k}} - r_x \theta_{y\bar{k}} \\
&= w_A + r_y \theta_x - r_x \theta_y
\end{aligned} \tag{18}$$

For plate bending applications the sign convention is not a statical sign convention. To modify eq. 18 the following change in variables must be implemented

$$\begin{aligned}
\theta_x &= \theta_y \\
\theta_y &= -\theta_x
\end{aligned}$$

This change in variables yields

$$w_B = w_A + r_x \theta_x + r_y \theta_y \tag{19}$$

Rotations of any point A will be equivalent to the rotations of a reference point B since it is a rigid body displacement. In matrix form, the relative displacements of two points on a rigid body can be expressed as

$$\begin{Bmatrix} w_B \\ \theta_{xB} \\ \theta_{yB} \end{Bmatrix} = \begin{vmatrix} 1 & r_x & r_y \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{Bmatrix} w_A \\ \theta_{xA} \\ \theta_{yB} \end{Bmatrix} = [T] \begin{Bmatrix} w_A \\ \theta_{xA} \\ \theta_{yB} \end{Bmatrix} \quad (20)$$

To develop the necessary modifications to the program the following relationships will be used:

$$\{\bar{\delta}\} = \begin{Bmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ w_8 \\ \theta_{x8} \\ \theta_{y8} \end{Bmatrix} \quad \text{and} \quad \{\bar{D}\} = \begin{Bmatrix} W_1 \\ \theta_{x1} \\ \theta_{y1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ W_8 \\ \theta_{x8} \\ \theta_{y8} \end{Bmatrix}$$

where $\{\bar{\delta}\}$ is the elemental displacement vector, and $\{\bar{D}\}$ is the system displacement vector with respect to rigid body modes. Characteristics of the transformation matrix are as follows:

$$\{F_e\} = [k_e]\{\bar{\delta}\}$$

but

$$\{\bar{\delta}\} = [T]\{D\}$$

The strain energy of the system becomes

$$u = 0.5\{\bar{\delta}\}^T[k_e]\{\bar{\delta}\} = 0.5\{D\}^T[T]^T[k_e][T]\{D\} \quad (21)$$

This results in a transformed stiffness matrix

$$[k] = [T]^T[k_e][T] \quad (22)$$

The total displacement formulation then becomes

$$\{F\} = [T]\{F_e\} = [T]^T[k_e][T]\{D\} = [k]\{D\} \quad (23)$$

To avoid formulating the entire transformation matrix, requiring $2N^3$ multiplication operations to perform the stiffness transformation, it is convenient to modify only those rows and columns of the elemental stiffness matrix that require modification. This is easily done during the assembly of the global stiffness matrix. To perform this operation it is necessary to identify which, if any, nodes

on an element are in contact with a rigid body. If an element is determined to contain nodes in contact with a rigid body the following sequence of events will occur

- 1) The x and y distances from a reference node to the first node of the element in contact with a rigid body will be calculated.
- 2) The transformation matrix will be formulated; rows of the elemental matrix corresponding to the node under consideration will be pre-multiplied by $[T]^t$; the columns will be post-multiplied by $[T]$.
- 3) The next node on the element in contact with a rigid body will be sought out and the operation will be repeated.
- 4) All 3 x 3 sub-matrices corresponding to 2 nodes in contact with the same rigid body will be overlayed into the same 3 x 3 sub-matrix along the diagonal (their original positions will be set to 0.0).

- 5) If the element is in contact with more than one rigid body the above procedure will be repeated for all rigid bodies in contact with the element.
- 6) The merging of the elemental matrix into the global stiffness matrix will continue as if no modification has occurred.

It is necessary to use a certain amount of intuition when using the rigid body option. Obviously, if the height of a column or wall is similar to the footing thickness, little or no effect of their additional stiffness will be realized by the footing. In this instance the rigid body assumption will give an erroneous solution.

RESULTS

The finite element program yields three quantities that are of primary concern to the user. These include deflections, subgrade contact stresses, and stresses within the footing itself. Three verifiable problems are analysed to document the accuracy of the finite element solution. The three problems include: a circular plate with a point load at the center, a notched rectangular footing with an eccentric column load, and a cantilever footing with two column loads.

Circular Plate With Point Load

The first problem analysed is a circular plate with a point load at the center supported by a Winkler subgrade. This problem was solved exactly, using thin plate theory, by Timoshenko and Woinowsky-Krieger, 1959, (7). A.C. Ugural, 1981, solved a similar problem using a two-parameter Ritz solution (8). The deflections predicted by the exact solution, the Ritz solution, and the finite element solution are compared, and their differences analyzed.

The problem parameters are given as (7)

$$l = (D/B)^{3/4} = 5.0$$

$$P = (0.00102) (8\pi B l^3)$$

$$D = Et^3/(12(1-\nu^2))$$

where

D = plate bending stiffness

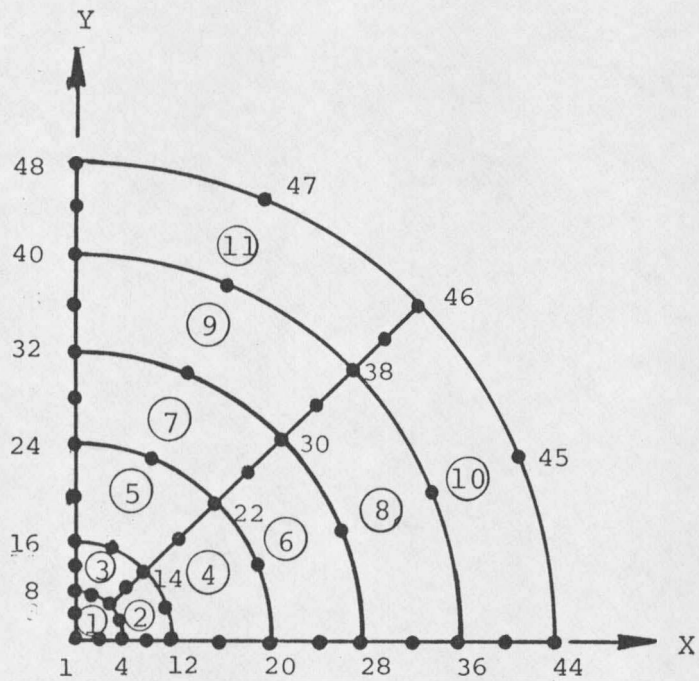
B = modulus of subgrade

E = Youngs modulus of elasticity

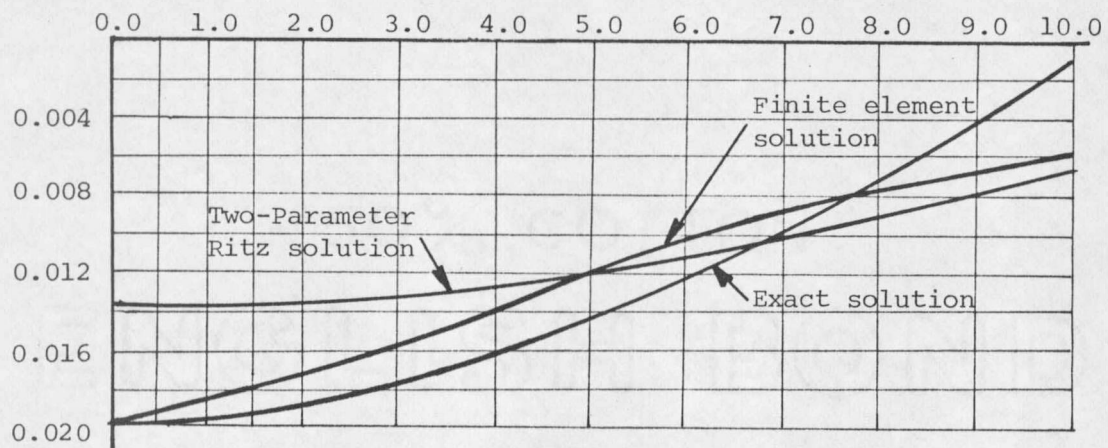
ν = Poissons ratio

P = magnitude of the point load

In order to minimize the cost of analysis, advantage of symmetry is taken and only one quadrant of the plate is analyzed. The boundry conditions, then, are that no bending deformation normal to the X or Y axis is allowed. The element grid and nodal numbering scheme selected is illustrated in Fig. 6 . Fig. 7 illustrates a comparison of the deflection profile along a radial axis predicted by the three methods of analysis. The two-parameter Ritz solution yields a very poor solution when compared to the exact solution. This result is not unexpected due to the poor selection of the assumed deflection function. However, the finite element solution also varies significantly from the exact solution. This difference is attributable to shear



CIRCULAR PLATE - ELEMENT GRID AND NODAL NUMBERING SCHEME
FIGURE 6



CIRCULAR PLATE: DEFLECTION PROFILE (INCHES)

FIGURE 7

deformation. Classical theory suggests that the effects of shear deformation are negligible if the ratio of the plate thickness to the radius of the plate is less than 0.4 (7). This is reflected by the finite element solution of a thin plate with no supporting subgrade. However, shear deformation is predicted to be significant for thin plates supported by a Winkler subgrade.

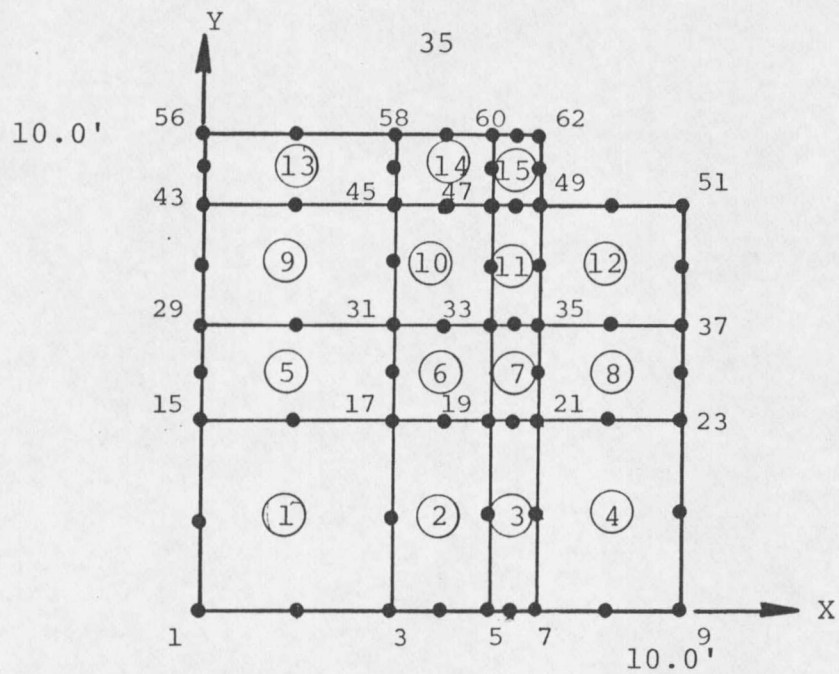
A significant difference is apparent in examining the deflection profiles predicted by the exact solution as compared to the finite element solution. The exact solution predicts positive curvature and, therefore, positive radial bending moments along a radial axis. This behavior is not reflected by the finite element solution. The finite element solution predicts negative, or reverse curvature toward the outside edge of the plate. This profile results in negative moments. Obviously, the difference in signs of the moments predicted is an important one to the design engineer responsible for the safe design of a footing.

Notched Rectangular Plate

The second problem solved is used to verify soil contact pressures predicted by the finite element solution.

A solution for a notched rectangular plate is presented as compared to the solution for a similar plate presented by Bowles, 1977, (1). The footing geometry and the element grid used is illustrated in Fig. 8. A footing depth of 24 inches was selected in order to insure a minimum of bending deformation. The loading is eccentric to the mass center of the footing; therefore, an overturning moment will exist. This insures that the contact stresses at the outside corners of the footing will bound the contact pressures present under the footing. Accordingly, the comparison of predicted pressures will be restricted to the pressures at the corners of the footing. The results predicted by a rigid footing analogy (1), finite element solution (not including shear deformation) (1), and the finite element solution predicted by the program developed in this paper are shown in Table 2.

The results predicted by the finite element solutions compare favorably with each other, certainly within the difference to be expected by considering plates of different thicknesses. Also, a small degree of variance is to be expected due to shear deformation. However, the finite element solutions vary significantly from the solution



NOTCHED FOOTING: ELEMENT GRID

FIGURE 8

SOIL CONTACT PRESSURES: KSF			
NODE	FINITE ELEMENT		RIGID BODY
	PLATE	BOWLES	BOWLES
1	5.49	5.308	4.27
9	5.43	5.287	5.57
49	5.71	5.755	6.51
51	5.62	5.602	6.90
56	5.44	5.292	5.84
62	5.66	5.655	6.75

TABLE 2

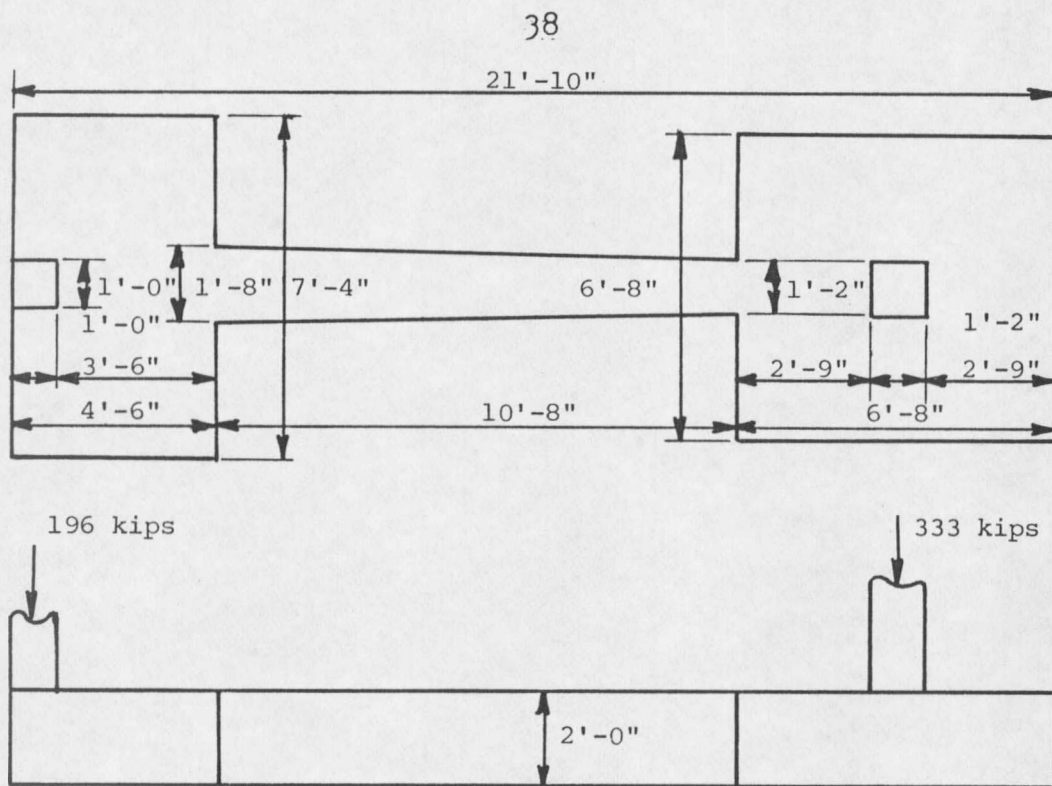
predicted by the rigid footing analysis. As expected, the finite element solution predicts much more uniform contact stresses than the rigid footing analogy. This is due primarily to the relief of contact stresses resulting from bending in the plate and a corresponding increase in contact stresses in low pressure areas. The problem parameters include a maximum allowable contact stress of 6.0 ksi. The rigid footing solution predicts stresses much higher than this, requiring a larger footing to distribute the load. The finite element solution however, predicts a pressure distribution that nowhere exceeds the maximum allowable pressure. This would require neither re-resolution of the problem nor additional materials for the footing.

Cantilever Footing Analysis

Wang and Salmon, 1979, solve the problem of a cantilever footing using a rigid beam assumption (9). A finite element solution of the problem is presented and the solutions compared as a final example. Problem geometry and parameters are illustrated in Fig. 9. Since the footing is symmetric about the X-axis, only one half of the footing need be analysed. This use of symmetry requires the use of

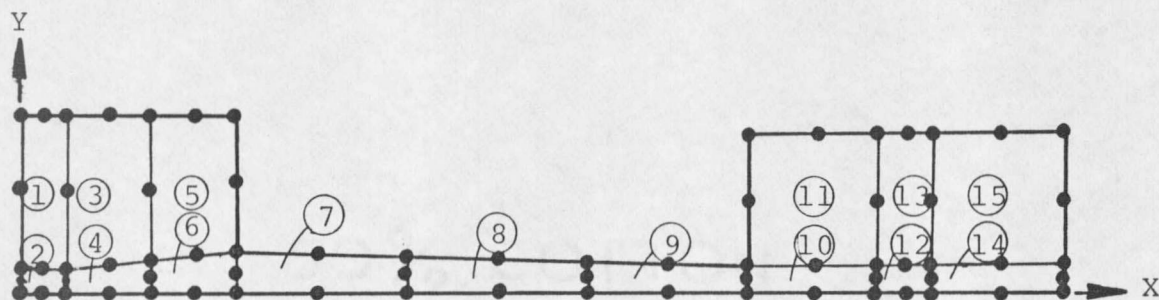
boundry conditions constraining rotation in the Y direction along the X-axis. The finite element grid used is illustrated in Fig. 10. A listing of the computer solution for this problem is given in appendix C.

The finite element solution presents shear forces and bending moments in terms of a unit width. To compare solutions with those predicted by the rigid beam analogy it is necessary to sum the bending moments and shear forces at given points along the X-axis. Fig. 11 illustrates the bending moments and shear forces predicted by the finite element solution at selected points along the X-axis. By computing the approximate areas under each of these curves, moment and shear diagrams for the footing are developed, Fig. 12. These diagrams are compared with the moment and shear diagrams resulting from the rigid beam analogy. The diagrams developed from the finite element solution compare closely with their counterparts. Bending moments under the rigid columns and adjacent areas are close to zero since no bending is allowed to occur in these areas.



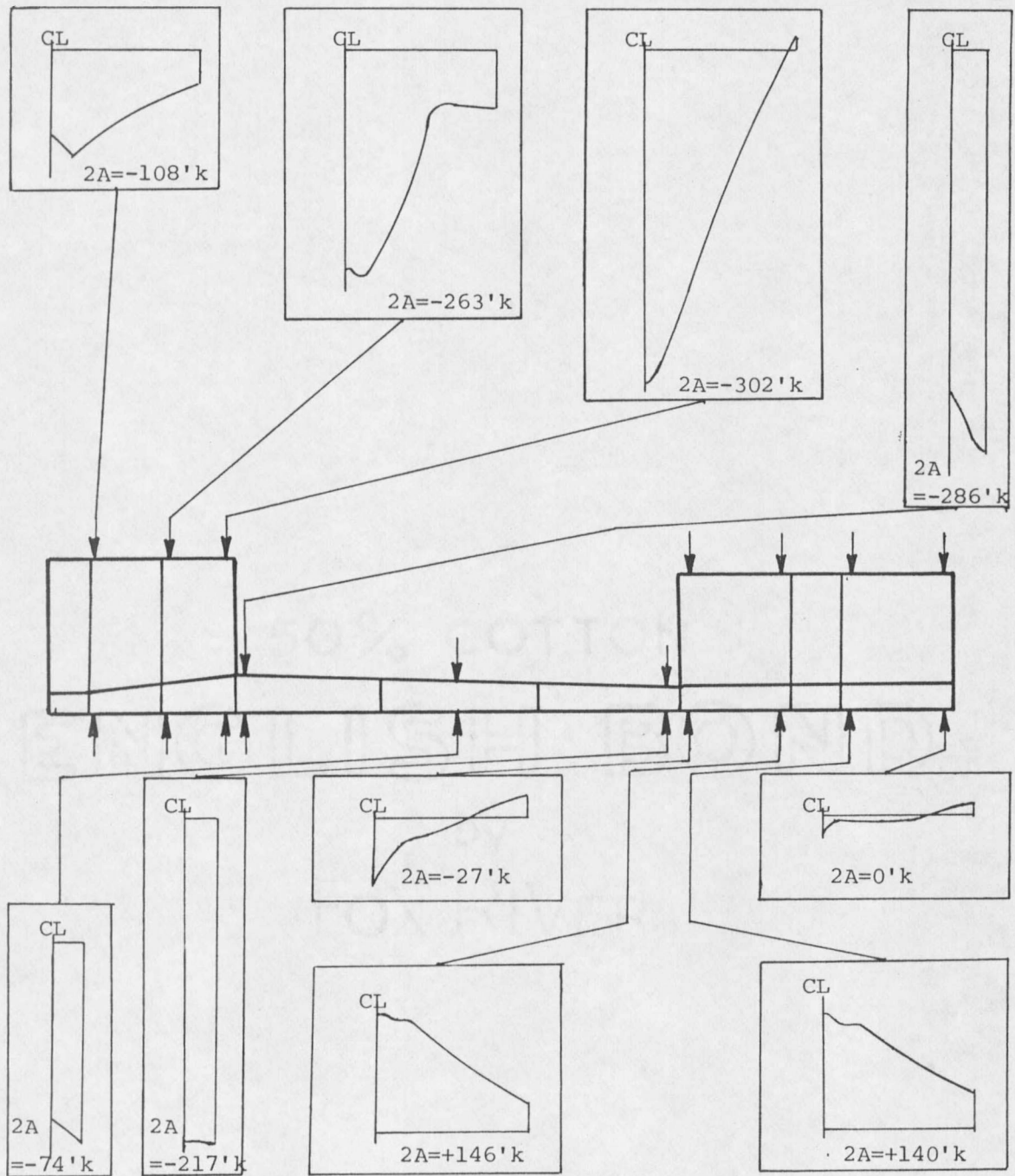
CANTILEVER FOOTING: GEOMETRY AND LOADING

FIGURE 9

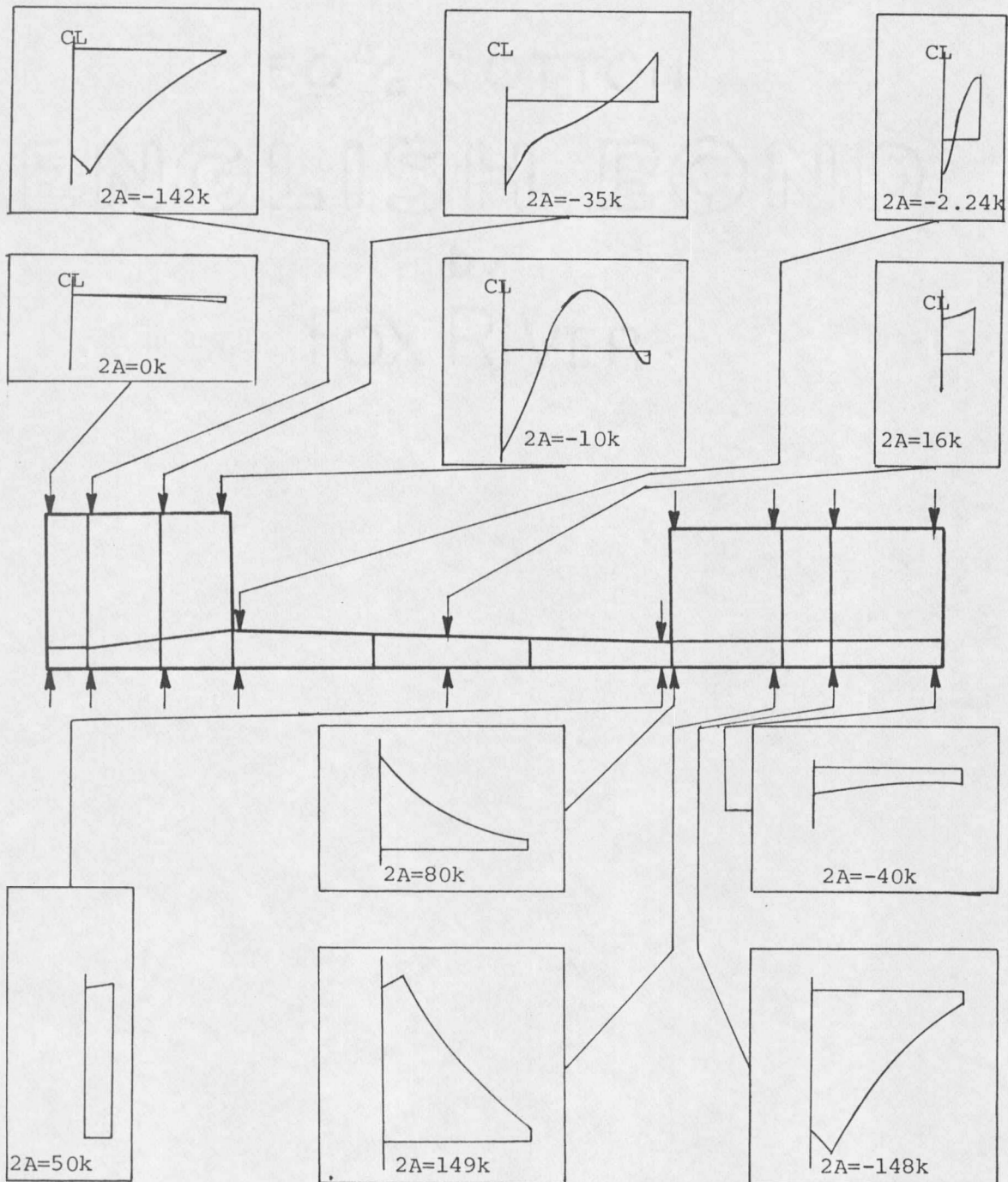


ELEMENT GRID

FIGURE 10



CANTILEVER FOOTING: BENDING MOMENT DIAGRAMS--SECTIONAL
FIGURE 11a



CANTILEVER FOOTING: SHEAR FORCE--SECTIONAL

FIGURE 11b

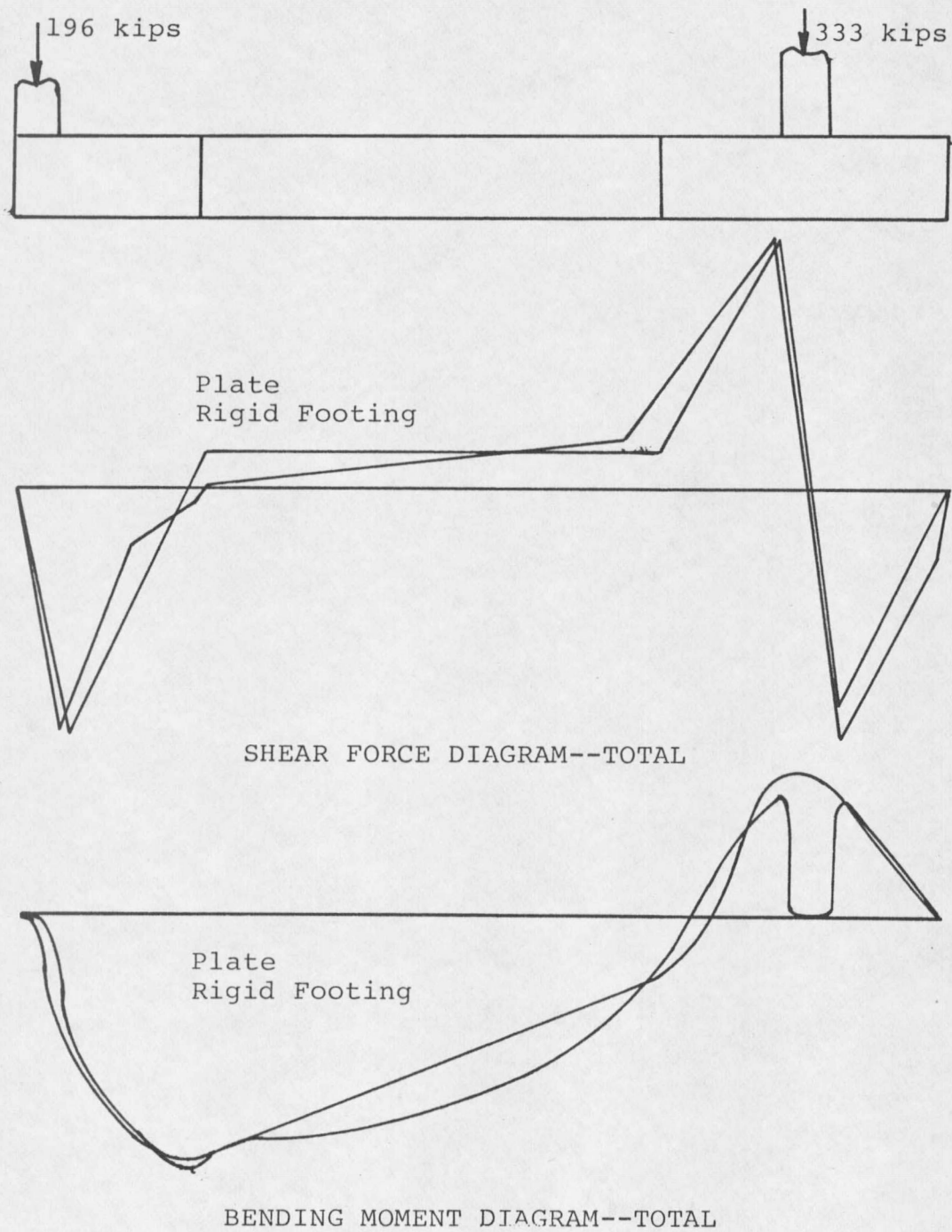


FIGURE 12

SUMMARY AND CONCLUSIONS

This paper presents an overview of the finite element method application to footing analysis. Specifically, the use of the eight-node isoparametric plate element is studied. The modification of a documented finite element plate analysis program, BOK1, to include the effects of a Winkler subgrade and rigid body action of foundation walls and columns is discussed.

Selection of the Winkler model to represent a subgrade is based on limitations of the frontal solution technique used in solving the displacement formula. The frontal technique requires the use of elemental stiffness matrices to maintain its efficiency. For an accurate representation of more advanced subgrade models, elemental matrices are difficult to generate.

It is considered to be desirable to model the stiffening effects of columns and walls in contact with a footing. A rigid body model is selected for efficiency and to avoid numerical instability problems. This requires the reduction of the degrees of freedom of all nodes in contact with the rigid body to the three degrees of freedom of a reference node. After deflections are calculated the translations of the reference node deflections to eliminated nodes is accomplished by rigid body relations.

Shear deformation is determined to have a significant contribution to the total deformation of a plate on a Winkler subgrade. This appears to be true even for plates that would be considered as thin plates in classical theory. This is important to a design engineer because of the possible difference in sign of the bending moments predicted by thin plate analysis as compared to thick plate analysis. The importance of including shear deformation is most prominent under a point load.

Soil contact pressures are shown to vary significantly from those predicted by a rigid footing analogy. The high contact stresses predicted by a rigid footing analogy are "relieved" by allowing bending in the footing. This results in a possible savings of materials needed to construct the footing.

Shear stresses and bending moments predicted by the finite element method agree closely with those predicted by a more approximate method (9). Due to the profiles of the moments or shears at a given cross section it may be possible to vary the spacing of the reinforcement to realize some savings in material.

RECOMENDATIONS

Perhaps the most important modification to the computer program developed is the addition of the ability to use more advanced subgrade models. This alteration would require the modification of the solution routine to include a method of calculating global stiffness addresses for additional degrees of freedom required by the subgrade model selected. Perhaps the simplest method of accomplishing this would be to formulate the entire global stiffness matrix. Unfortunately, this would require large increases in the required memory resources as well as the cost of analysis.

For a much more accurate and theoretically significant program the addition of a viscoelastic subgrade could be employed. This would allow the analysis of various structures, including the analysis of foundations on snow. This would also allow the accurate prediction of the settlement of large structures which is also of interest to the design engineer.

Ultimately, if a large computer system were accessible, the addition of a three-dimensional frame program could be implemented. This, combined with a viscoelastic subgrade, would allow the study of structure behavior over a period of years, thus allowing the study of earthquakes, fatigue problems, and other time dependent loading patterns.

On a more immediate level, there are several modifications that would be of value to a design engineer. One very real problem with a finite element analysis is the large volume of information yielded. This vast bulk of information is, at best, very difficult to interpret. The added capability of graphical contour mapping of stresses, moments, deflections, etc. would vastly simplify the interpretation of the data. Also, the ability to compute the total stresses and moments at a given cross section would enable the design engineer to visualize the total moment and shear diagrams much easier, thus allowing some insight as to the behavior of the footing.

To eliminate shearing deformation, it is necessary to analyze an extremely thin plate. The circular plate problem analyzed is subject to shear deformation for plates with a thickness to radius ratio of 0.02. For plates thinner than this a numerical instability in the program becomes apparent. Since the flexural rigidity of the plate is left constant for the problem, shearing term increases are inversely proportional to the square of the change in thickness. The resulting shear stiffnesses become much greater than the flexural rigidity, resulting in an ill-conditioned matrix. To minimize the effects of this

instability recompiling the program with a double precision option is necessary. Beyond this there is little that can be done. For very thin plates with an extraordinarily high modulus of elasticity, solutions in which the thickness is changed slightly, and the flexural rigidity held constant should be run as a stability check.

A final important modification that should be included is the ability to modify the subgrade stiffness matrix whenever a state of uplift occurs. This condition will occur whenever the resultant of the loading is located outside the kern of the footing. In order to correctly analyze this problem, the subgrade material must not be allowed to bond to the footing as this suggests the ability to pull the footing down. This modification would require the inversion of the subgrade stiffness matrix to obtain the flexibility matrix. Once the flexibility matrix is obtained, the rows and columns of the matrix that correspond to a node in uplift would have to be set to zero. The matrix then must be re-inverted to obtain a modified stiffness matrix (2). The problem then must be resolved to obtain the correct solution. The additional expense of the inversion routine and the extra computer time would have to be justifiable before this solution package should be

implemented.

Further study is needed on the effects of considering rigid body action of columns and walls contacting the footing. Particularly, limiting values of the ratio of the wall or column height to the footing thickness need to be established. This limiting value would be the lowest ratio for which rigid body action could reasonably be expected to yield a good solution.

REFERENCES

- 1 Bowles, Joseph E., 1977, FOUNDATION ANALYSIS AND DESIGN, McGraw-Hill Book Company, New York, 750 pp.
- 2 Cheung, Y. K. and Nag, D. K., 1968, PLATES AND BEAMS ON ELASTIC FOUNDATIONS-LINEAR AND NON-LINEAR BEHAVIOR, Geotechnique, Vol. 18, 250-260.
- 3 Cheung, Y. K. and Zienkiewicz, O. C., 1965, PLATES AND TANKS ON ELASTIC FOUNDATIONS -AN APPLICATION OF THE FINITE ELEMENT METHOD, International Journal of Solids and Structures, Pergamon Press Ltd., Great Britain, Vol. 1, 451-461.
- 4 Cook, Robert D., 1974, CONCEPTS AND APPLICATIONS OF FINITE ELEMENT ANALYSIS, John Wiley and Sons, New York, 402 pp.
- 5 Hinton, E. and Owen, D. R. J., FINITE ELEMENT PROGRAMMING, Academic Press, London, 305 pp.
- 6 Kerr, Arnold D., Sept. 1964, ELASTIC AND VISCOELASTIC FOUNDATION MODELS, Journal of Applied Mechanics, Transactions of the ASME, 491-498.
- 7 Timoshenko, S. and Woinowsky-Krieger, S., 1959, THEORY OF PLATES AND SHELLS, McGraw-Hill Book Company, New York, 580 pp.
- 8 Ugural, A. C., 1981, STRESSES IN PLATES AND SHELLS, McGraw-Hill Book Company, New York, 317 pp.

- 9 Wang, Chu-Kia and Salmon, Charles G.,
1979, REINFORCED CONCRETE DESIGN
3rd. ed., Harper and Row, New York,
918 pp.

Appendix A

Instructions for Preparing Input Data

CARD SET 1--PROBLEM CARD(I5)--one card

Cols. 1-5	NPROB	Total number of problems to be solved in one run.
-----------	-------	--

CARD SET 2--SUBGRADE TYPE(I5)--one card

Cols. 1-5	IFND	Type of subgrade modeled 0=no subgrade 1=lumped Winkler 2=integrated Winkler
-----------	------	---

CARD SET 3--TITLE CARD(18A4)--one card

Cols, 1-72	Title	Title of problem--limited to 72 alphanumeric characters
------------	-------	---

CARD SET 4--CONTROL CARD(12I5)--one card

Cols. 1-5	NPOIN	Total number of nodal points.
6-10	NELEM	Total number of elements.
11-15	NVFIX	Total number of restrained boundary points--where one or more degrees of freedom are restrained.
16-20	NCASE	Total number of load cases to be analysed.
21-25	NTYPE	Blank
26-30	NNODE	Number of nodes per element (= 8)
31-35	NDOFN	Number of degrees of freedom per node (= 3).
36-40	NMATS	Total number of different materials.
41-45	NPROP	Number of independent properties per material (= 4)
46-50	NGAUS	Order of integration formula for numerical integration (2 or 3)
51-55	NDIME	Number of coordinate dimensions (= 2).
56-60	NSTRE	Number of independent

generalized stress
componant (= 5)

CARD SET 5--ELEMENT CARDS-One card for each element.
Total of NELEM cards (see Card Set 4)

Cols. 1-5	NUMEL	Element Number
6-10	MATNO(NUMEL)	Material property number
11-15	LNODS(NUMEL,1)	1st Nodal connection number
46-50	LNODS(NUMEL,8)	8th Nodal connection number

NOTE: The nodal connection numbers must be listed in an anticlockwise sequence, starting from any corner node; the element being viewed from above the plane $z = 0$.

CARD SET 6--NODE CARDS(I5,2F10.5,I5)-One card for each node whose coordinates are to be input.

Cols. 1-5	IPOIN	Nodal point number
6-15	COORD(IPOIN,1)	x-coordinate of node
16-25	COORD(IPOIN,2)	y-coordinate of node
26-30	NDTYP	Type of node
		1 = corner
		2 = midside

NOTES: The coordinates of the highest numbered node must be input, regardless of whether it is a midside node or not

The total number of cards in this set may differ from NPOIN (see Card Set 4) since for element sides which are linear it is only necessary to specify data for corner nodes; intermediate nodal coordinates being automatically interpolated if on a straight line.

If a type one Winkler model is used all nodal coordinates must be specified since it is necessary to identify the node type.

CARD SET 7--RESTRAINED NODE CARDS(1X,I4,2X,3I1,3F10.6)

One card for each restrained node. Total of NVFIX cards (see Card Set 4).

Cols. 2-5	NOFIX(IVFIX)	Restrained Node Number
8	IFPRE(IVFIX,1)	Condition of restraint on displacement, w. 0 No restraint 1 node restrained
9	IFPRE(IVFIX,2)	Condition of restraint on nodal rotation, θ_x . 0 No restraint 1 Node restrained
10	IFPRE(IVFIX,3)	Condition of restraint on nodal rotation, θ_y . 0 No restraint 1 node restrained
11-20	PRESC(IVFIX,1)	The prescribed value of w.
21-30	PRESC(IVFIX,2)	The prescribed value of θ_x .
31-40	PRESC(IVFIX,3)	The prescribed value of θ_y .

NOTE: If a subgrade is used it is not necessary to prescribe any displacements.

CARD SET 8--RIGID BODY CARDS(2I5)-one card

Cols. 1-5	NRGD	Number of rigid bodies
6-10	NNDRGD	Total number of nodes on all rigid bodies

CARD SET 9--REFERENCE NODE CARDS(I5)-One card for each rigid body.

Cols. 1-5	REF(IRGD)	Reference node associated with the rigid body.
-----------	-----------	--

CARD SET 10-MAXIMUM NO. NODES ON RIGID BODY(I5)-one card

Cols. 1-5	MAXRGD	Maximum no. of nodes on any rigid body.
-----------	--------	---

CARD SET 11-RIGID BODY NODE CARDS(I4,8X,20I3)-One card
for each rigid body.

Cols. 1-3 RGDND(IRGD,1) 1st node on rigid body

RGDND(IRGD,MAXRGD) last node on rigid
body

CARD SET 12-MATERIAL CARDS(I5,4F10.5)-One card for each
different material. Total of NMATS cards
(see Card Set 4).

Cols. 1-5	NUMAT	Material identification number.
6-15	PROPS(NUMAT,1)	Elastic Modulus, E.
16-25	PROPS(NUMAT,2)	Poisson's Ratio, ν .
26-35	PROPS(NUMAT,3)	Material Thickness, t.
36-45	PROPS(NUMAT,4)	Distributed load, q.

CARD SET 13-LOAD CASE TITLE CARD(18A4)-one card

Cols. 1-72 TITLE Title of the load case
-limited to 72 characters.

CARD SET 14-LOAD CONTROL CARD(I5)-one card

Cols. 1-5 IPLOD Applied point load
control parameter
0-No applied point loads
1-Applied point loads
to be input.

CARD SET 15-APPLIED LOAD CARDS(I5,3F10.3)-one card
for each loaded nodal point.

Cols. 1-5	LODPT	Node number
6-15	POINT(1)	Load component in the z-direction.
16-25	POINT(2)	Nodal couple in the xz plane.
26-35	POINT(3)	Nodal couple in the yz plane.

NOTES: The last card should be that for the highest
numbered node whether it is loaded or not.

If IPLOD = 0, omit this card set.

CARD SETS 13 TO 15 TO BE REPEATED FOR EACH LOAD CASE
IN ACCORDANCE WITH 'NCASE' IN CARD SET 4.

CARD SETS 3 TO 15 TO BE REPEATED FOR EACH PROBLEM IN
ACCORDANCE WITH 'NPROB' IN CARD SET 1.

Appendix B

Program listing

```

PROGRAM PLATE
C
C*** INPUT,OUTPUT,TAPE105=INPUT,TAPE108=OUTPUT,
C TAPE 1, TAPE 2, TAPE 3, TAPE 4
C
  DIMENSION TITLE(12)
  COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
  * ASDIS(240),ELOAD(24,24),NOFIX(40),
  * IFPRE(40,3),LNODS(25,8),MATNO(25)
  COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
  * CARTD(2,8),DBMAT(5,24),BMATX(5,24),
  * SMATX(5,24,9),POSGP(3),WEIGP(3),
  * GPCOD(2,9),NEROR(24)
  COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
  * NSTRE,NTYPE,NGAUS,NPROP,NMATS,
  * NVFIX,NEVAB,ICASE,NCASE,ITEMP,
  * IPROB,NPROB
  COMMON/WINKLR/
  * IFND,BMODLS,TAREA,NDTYP,ENFLNC,
  * WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
  * BEFMAT(8),ELDISP(24,8),NDFRC(24)
  COMMON/RIGID/ NRGD,MAXRGD,REF(5),RGDND(5,80),LNADS(25,8)
  OPEN (1,NAME='*1',USAGE='CREATE')
  OPEN (2,NAME='*2',USAGE='CREATE')
  OPEN (3,NAME='*3',USAGE='CREATE')
  OPEN (4,NAME='*4',USAGE='CREATE')
  OPEN (5,NAME='*5',USAGE='CREATE')
  OPEN (6,NAME='*6',USAGE='CREATE')
  READ(105,900) NPROB
900 FORMAT(I5)
  WRITE(108,905) NPROB
905 FORMAT(1H0,5X,23HTOTAL NO. OF PROBLEMS =,I5)
  DO 20 IPROB=1,NPROB
  READ (105,920) IFND
920 FORMAT(I5)
  REWIND 1
  REWIND 2
  REWIND 3
  REWIND 4
  REWIND 5
  REWIND 6
  READ(105,910) TITLE
910 FORMAT(18A4)
  WRITE(108,915) IPROB,TITLE
915 FORMAT(1H0,6X,12HPROBLEM NO.,I3,10X,
  *18A4)
  IF(IFND.NE.1) GO TO 930
  WRITE(108,925)
925 FORMAT(1X/1X,'**LUMPED WINKLER FOUNDATION**')
  GO TO 940
930 IF(IFND.NE.2) GO TO 940
  WRITE(108,935)
935 FORMAT(1X/1X,'**INTEGRATED WINKLER FOUNDATION**')
940 CONTINUE
C
C*** CALL THE SUBROUTINE WHICH READS MOST OF THE DATA
C
  CALL INPUT
C
C*** CREATE THE ELEMENT STIFFNESS FILE

```

```

C      CALL STIFPB
C      DO 10 ICASE=1,NCASE
C
C*** COMPUTE LOADS, AFTER READING THE RELEVANT
C      EXTRA DATA
C
C      CALL LOADPB
C
C*** MERGE AND SOLVE THE RESULTING EQUATIONS
C      BY THE FRONTAL SOLVER
C
C      CALL FRONT
C
C*** COMPUTE THE STRESSES IN ALL THE ELEMENTS.
C
C      CALL STREPB
C      CALL FPRES
C      10 CONTINUE
C      20 CONTINUE
C      STOP
C      END
C      SUBROUTINE INPUT
C
C*** READ THE FIRST DATA CARD, AND ECHO IT
C      IMMEDIATELY,
C
C      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
C      *      ASDIS(240),ELOAD(24,24),NOFIX(40),
C      *      IFPRE(40,3),LNODS(25,8),MATNO(25)
C      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
C      *      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
C      *      SMATX(5,24,9),POSGP(3),WEIGP(3),
C      *      GPCOD(2,9),NEROR(24)
C      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
C      *      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
C      *      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
C      *      IPROB,NPROB
C      COMMON/WINKLR/
C      *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
C      *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
C      *      BEFMAT(8),ELDISP(24,8),NDFRC(24)
C      COMMON/RIGID/ NRGD,MAXRGD,REF(5),RGDND(5,80),LNADS(25,8)
C      READ(105,900) NPOIN,NELEM,NVFIX,NCASE,NTYPE,
C      *NNODE,NDOFN,NMATS,NPROP,NGAUS,NDIME,NSTRE
C      800 FORMAT(10I5,F10.5)
C      900 FORMAT(12I5)
C      NEVAB=NDOFN*NNODE
C      WRITE(108,905) NPOIN,NELEM,NVFIX,NCASE,NTYPE,
C      *NNODE,NDOFN,NMATS,NPROP,NGAUS,NDIME,
C      *NSTRE,NEVAB
C      905 FORMAT(/8H NPOIN =,I4,4X,8H NELEM =,I4,
C      * 4X,8H NVFIX =,I4,4X,8H NCASE =,I4,4X,
C      * 8H NTYPE =,I4,4X,8H NNODE =,I4,4X,
C      * 8H NDOFN =,I4,4X,8H NMATS =,I4,4X,
C      * 8H NPROP =,I4,4X,8H NGAUS =,I4,4X,
C      * 8H NDIME =,I4,4X,8H NSTRE =,I4,4X,
C      * 8H NEVAB =,I4)
C      CALL CHECK1
C
C*** ZERO ALL THE NODAL COORDINATES
C      IT IS NECESSARY TO READ IN ALL NODE

```

```

C   COORDINATES FOR A LUMPED FOUNDATION
C
      DO 20 IPOIN=1,NPOIN
      DO 20 IDIME=1,NDIME
20   COORD(IPOIN,IDIME)=0.0
      DO 21 IELEM=1,NELEM
21   WINKMD(IELEM)=0.0
C
C*** READ THE ELEMENT NODAL CONNECTIONS, AND
C   THE PROPERTY NUMBERS,
C
      IF(IFND.EQ.0) GO TO 170
      WRITE(108,810)
810  FORMAT(/8H ELEMENT,3X,8HPROPERTY,18X,12HNODE NUMBERS
      * ,16X,12H SOIL MODULUS)
      DO 110 IELEM=1,NELEM
      READ(105,800) NUMEL,MATNO(NUMEL),
      * (LNODS(NUMEL,INODE),INODE=1,NNODE),WINKMD(NUMEL)
110  WRITE(108,815) NUMEL,MATNO(NUMEL),
      * (LNODS(NUMEL,INODE),INODE=1,NNODE),WINKMD(NUMEL)
815  FORMAT(1X,15,I9,6X,815,5X,F10.5)
      IF(IFND.EQ.2) GO TO 115
C
C*** READ NODAL POINT COORDINATES AND FOUNDATION MODULUS
C
      WRITE(108,820)
820  FORMAT(/25H NODAL POINT COORDINATES,6X,9HNODE TYPE)
      WRITE(108,825)
825  FORMAT(6H NODE,7X,1HX,9X,1HY)
130  READ(105,830) IPOIN,(COORD(IPOIN,IDIME),
      * IDIME=1,NDIME),NDTYPE(IPOIN)
      IF(IPOIN.NE.NPOIN) GO TO 130
830  FORMAT(15,2F10.5,15)
      DO 150 IPOIN=1,NPOIN
150  WRITE(108,835) IPOIN,(COORD(IPOIN,IDIME),
      * IDIME=1,NDIME),NDTYPE(IPOIN)
835  FORMAT(1X,15,2F10.3,5X,15)
      GO TO 190
170  WRITE(108,910)
910  FORMAT(/8H ELEMENT,3X,8HPROPERTY,6X,
      * 12HNODE NUMBERS)
      DO 10 IELEM=1,NELEM
      READ(105,900) NUMEL,MATNO(NUMEL),
      * (LNODS(NUMEL,INODE),INODE=1,NNODE)
10   WRITE(108,915) NUMEL,MATNO(NUMEL),
      * (LNODS(NUMEL,INODE),INODE=1,NNODE)
915  FORMAT(1X,15,I9,6X,815)
C
C*** READ SOME NODAL COORDINATES, FINISHING
C   WITH THE LAST NODE OF ALL.
C
115  CONTINUE
      WRITE(108,920)
920  FORMAT(/25H NODAL POINT COORDINATES)
      WRITE(108,925)
925  FORMAT(6H NODE,7X,1HX,9X,1HY)
30   READ(105,930) IPOIN,(COORD(IPOIN,IDIME),
      * IDIME=1,NDIME)
930  FORMAT(15,5F10.5)
      IF(IPOIN.NE.NPOIN) GO TO 30
C
C*** INTERPOLATE COORDINATES OF MIDSIDE NODES

```



```

C      IF(NDIME.EQ.1) GO TO 40
      CALL NODEXY
40     CONTINUE
      DO 50 IPOIN=1,NPOIN
50     WRITE(108,935) IPOIN,(COORD(IPOIN,IDIME),
      * IDIME=1,NDIME)
935    FORMAT(1X,I5,3F10.3)
190    CONTINUE
      IF(NVFIX.EQ.0) GO TO 90

C
C*** READ THE FIXED VALUES,
C
      WRITE(108,940)
940    FORMAT(//17H RESTRAINED NODES)
      WRITE(108,945)
945    FORMAT(5H NODE,2X,4HCODE,10X,
      *12HFIXED VALUES)
      IF(NDOFN.NE.2)GO TO 70
      DO 60 IVFIX=1,NVFIX
      READ(105,950) NOFIX(IVFIX),(IFPRE(IVFIX,
      * IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,IDOFN),
      * IDOFN=1,NDOFN)
60     WRITE(108,950) NOFIX(IVFIX),(IFPRE(IVFIX,
      * IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,IDOFN),
      * IDOFN=1,NDOFN)
950    FORMAT(1X,I4,3X,2I1,2F10.6)
      GO TO 90
70     DO 80 IVFIX=1,NVFIX
      READ(105,955) NOFIX(IVFIX),(IFPRE(IVFIX,
      * IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,
      * IDOFN),IDOFN=1,NDOFN)
80     WRITE(108,955) NOFIX(IVFIX),(IFPRE(IVFIX,
      * IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,
      * IDOFN),IDOFN=1,NDOFN)
955    FORMAT(1X,I4,2X,3I1,3F10.6)
90     CONTINUE
      READ(105,700) NRGD,NNDRGD
700    FORMAT (2I5)
      WRITE(108,710) NRGD,NNDRGD
710    FORMAT(//26H NUMBER OF RIGID BODIES =,I3,/
      * 35H NUMBER OF NODES ON RIGID BODIES =,I3)
      IF(NRGD.EQ.0) GO TO 240
      WRITE(108,720)
720    FORMAT(//37H RIGID BODY NUMBER    REFERENCE NODE)
      DO 210 IRGD=1,NRGD
      READ(105,730) REF(IRGD)
210    WRITE(108,740) IRGD,REF(IRGD)
730    FORMAT (15)
740    FORMAT(1X,10X,I3,17X,I3)
      READ(105,750) MAXRGD
750    FORMAT(15)
      WRITE(108,760) MAXRGD
760    FORMAT(//43H MAXIMUM NUMBER OF NODES ON A RIGID BODY =,I3)
      DO 220 IRGD=1,NRGD
220    READ(105,770) (RGDND(IRGD,NODE),NODE=1,MAXRGD)
770    FORMAT(24I3)
      WRITE(108,780)
780    FORMAT(//23H RIGID BODY    NODES)
      DO 230 IRGD=1,NRGD
230    WRITE(108,790) IRGD,(RGDND(IRGD,NODE),NODE=1,MAXRGD)
790    FORMAT(14,8X,20I3)

```

```

240 CONTINUE
C
C*** READ THE AVAILABLE SELECTION OF ELEMENT
C   PROPERTIES,
C
      WRITE(108,960)
960 FORMAT(/ / 21H MATERIAL PROPERTIES)
      WRITE(108,965)
965 FORMAT(8H NUMBER,6X,14HYOUNGS MODULUS,2X,
* 15HPOISSON'S RATIO,2X,9HTHICKNESS,2X,
* 12HUNIFORM LOAD)
      DO 100 IMATS=1,NMATS
        READ(105,930) NUMAT,(PROPS(NUMAT,IPROP),
* IPROP=1,NPROP)
100 WRITE(108,970) NUMAT,(PROPS(NUMAT,IPROP),
* IPROP=1,NPROP)
970 FORMAT(1X,15,7X,5E14,6)
C
C*** SET UP GAUSSIAN INTEGRATION CONSTANTS
C
      CALL GAUSSQ
      CALL CHECK2
      RETURN
      END
      SUBROUTINE NODEXY
C
C*** LOOP OVER EACH ELEMENT
C
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
* ASDIS(240),ELOAD(24,24),NOFIX(40),
* IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
* CARTD(2,8),DBMAT(5,24),BMATX(5,24),
* SMATX(5,24,9),POSGP(3),WEIGP(3),
* GPCOD(2,9),NEROR(24)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
* NSTRE,NTYPE,NGAUS,NPROP,NMATS,
* NVFIX,NEVAB,ICASE,NCASE,ITEMP,
* IPROB,NPROB
      COMMON/WINKLR/
* IFND,BMODLS,TAREA,NDTYP,ENFLNC,
* WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
* BEFMAT(8),ELDISP(24,8),NDFRC(24)
      DO 30 IELEM=1,NELEM
C
C*** LOOP OVER EACH ELEMENT EDGE
C
        DO 20 INODE=1,NNODE,2
C
C*** COMPUTE THE NODE NUMBER OF THE FIRST NODE
C
          NODST=LNODS(IELEM,INODE)
          IGASH=INODE+2
          IF(IGASH.GT.NNODE) IGASH=1
C
C*** COMPUTE THE NODE NUMBER OF THE LAST NODE
C
          NODFN=LNODS(IELEM,IGASH)
          MIDPT=INODE+1
C
C*** COMPUTE THE NODE NUMBER OF THE INTERMEDIATE NODE
C

```

```

      NODMD=LNODES(IELEM,MIDPT)
      TOTAL=ABS(COORD(NODMD,1))+
      * ABS(COORD(NODMD,2))
C
C*** IF THE COORDINATES OF THE INTERMEDIATE
C NODE ARE BOTH ZERO, INTERPOLATE BY A
C STRAIGHT LINE.
C
      IF(TOTAL.GT.0.0) GO TO 20
      KOUNT=1
10  COORD(NODMD,KOUNT)=(COORD(NODST,KOUNT)+
      * COORD(NODFN,KOUNT))/2.0
      KOUNT=KOUNT+1
      IF(KOUNT.EQ.2) GO TO 10
20  CONTINUE
30  CONTINUE
      RETURN
      END
      SUBROUTINE GAUSSQ
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
      * ASDIS(240),ELOAD(24,24),NOFIX(40),
      * IFPRE(40,3),LNODES(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      * CARTD(2,8),DBMAT(5,24),BMATX(5,24),
      * SMATX(5,24,9),POSGP(3),WEIGP(3),
      * GPCOD(2,9),NEROR(24)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
      * NSTRE,NTYPE,NGAUS,NPROP,NMATS,
      * NVFIX,NEVAB,ICASE,NCASE,ITEMP,
      * IPROB,NPROB
      COMMON/WINKLR/
      * IFND,BMODLS,TAREA,NDTYP,ENFLNC,
      * WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
      * BEFMAT(8),ELDISP(24,8),NDFRC(24)
      IF(NGAUS.GT.2) GO TO 10
      POSGP(1)=-0.577350269189626
      WEIGP(1)=1.0
      GO TO 20
10  POSGP(1)=-0.774596669241483
      POSGP(2)=0.0
      POSGP(3)=-POSGP(1)
      WEIGP(1)=0.5555555555555556
      WEIGP(2)=0.8888888888888889
      WEIGP(3)=WEIGP(1)
      GO TO 40
20  KGAUS=NGAUS/2
      DO 30 IGASH=1,KGAUS
      JGASH=NGAUS+1-IGASH
      POSGP(JGASH)=-POSGP(IGASH)
      WEIGP(JGASH)=WEIGP(IGASH)
30  CONTINUE
40  CONTINUE
      RETURN
      END
      SUBROUTINE STIFPB
C
C*** CALCULATES ELEMENT STIFFNESS MATRIX
C FOR PLATE BENDING ELEMENT
C
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
      * ASDIS(240),ELOAD(24,24),NOFIX(40),
      * IFPRE(40,3),LNODES(25,8),MATNO(25)

```



```

COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
*   CARTD(2,8),DBMAT(5,24),BMATX(5,24),
*   SMATX(5,24,9),POSGP(3),WEIGP(3),
*   GPCOD(2,9),NEROR(24)
COMMON/CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
*   NSTRE,NTYPE,NGAUS,NPROP,NMATS,
*   NVFIX,NEVAB,ICASE,NCASE,ITEMP,
*   IPROB,NPROB
COMMON/WINKLR/
*   IFND,BMODLS,TAREA,NDTYP,ENFLNC,
*   WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
*   BEFMAT(8),ELDISP(24,8),NDFRC(24)
DIMENSION ESTIF(24,24)

C
C*** LOOP OVER EACH ELEMENT
C
DO 70 IELEM=1,NELEM
  LPROP=MATNO(IELEM)
  IF(IFND.LT.1) GO TO 15
  BMODLS=WINKMD(IELEM)
15 CONTINUE
C
C*** EVALUATE THE COORDINATES OF
C   THE ELEMENT NODAL POINTS
C
DO 10 INODE=1,NNODE
  LNODE=LNODS(IELEM,INODE)
  DO 10 IDIME=1,NDIME
    ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
  10 CONTINUE
C
C*** INITIALIZE THE ELEMENT STIFFNESS MATRIX
C
DO 20 IEVAB=1,NEVAB
  DO 20 JEVAB=1,NEVAB
    ESTIF(IEVAB,JEVAB)=0.0
    IF(IFND.LT.1) GO TO 20
    EFSTIF(IEVAB,JEVAB)=0.0
    CPSTIF(IEVAB,JEVAB)=0.0
  20 CONTINUE
  TAREA=0.0
C
C*** CALCULATE MATRIX OF ELASTIC RIGIDITIES
C
CALL MODPB(LPROP)
KGASP=0
C
C*** ENTER LOOPS FOR NUMERICAL INTEGRATION
C
DO 50 IGAUS=1,NGAUS
  EXISP=POSGP(IGAUS)
  DO 50 JGAUS=1,NGAUS
    ETASP=POSGP(JGAUS)
    KGASP=KGASP+1
C
C*** EVALUATE THE SHAPE FUNCTIONS,
C   ELEMENTAL AREA,ETC.
C
CALL SFR2(EXISP,ETASP)
CALL JACOB2(IELEM,DJACB,KGASP)
DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
TAREA=TAREA+DAREA

```

```

C
C*** EVALUATE THE B AND DB MATRICES
C
      CALL BMATPB
      CALL DBE
C
C*** CALCULATE THE ELEMENT STIFFNESS
C
      DO 30 IEVAB=1,NEVAB
      DO 30 JEVAB=IEVAB,NEVAB
      DO 30 ISTR=1,NSTRE
      ESTIF(IEVAB,JEVAB)=ESTIF(IEVAB,JEVAB)+
      *BMATX(ISTR,IEVAB)*DBMAT(ISTR,JEVAB)
      **DAREA
30 CONTINUE
C
C*** CALCULATE THE INTEGRATED WINKLER
      STIFFNESS MATRIX IF REQUIRED
C
      IF(IFND.NE.2) GO TO 37
      DO 35 INODE=1,NNODE
      I=3*INODE-2
      DO 35 JNODE=INODE,NNODE
      J=3*JNODE-2
35 EFSTIF(I,J)=SHAPE(INODE)*SHAPE(JNODE)*DAREA
      **BMODLS+EFSTIF(I,J)
37 CONTINUE
C
C*** STORE THE COMPONENTS OF THE DB MATRIX
      FOR THE ELEMENT
C
      DO 40 ISTR=1,NSTRE
      DO 40 IEVAB=1,NEVAB
      SMATX(ISTR,IEVAB,KGASP)=
      *DBMAT(ISTR,IEVAB)
40 CONTINUE
50 CONTINUE
C
C*** CONSTRUCT THE LOWER TRIANGLE OF THE
      ELEMENT AND FOUNDATION STIFFNESS MATRICES
C
      DO 60 IEVAB=1,NEVAB
      DO 60 JEVAB=1,NEVAB
      ESTIF(JEVAB,IEVAB)=ESTIF(IEVAB,JEVAB)
      IF(IFND.NE.2) GO TO 60
      EFSTIF(JEVAB,IEVAB)=EFSTIF(IEVAB,JEVAB)
60 CONTINUE
      IF(IFND.LT.1) GO TO 67
      IF(IFND.GT.1) GO TO 65
      DO 63 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      NDTYP=NDTYPE(LNODE)
      IF(NDTYP.EQ.1) ENFLNC=1.0/12.0; GO TO 63
      IF(NDTYP.EQ.2) ENFLNC=1.0/6.0
63 EFSTIF(3*INODE-2,3*INODE-2)=BMODLS*TAREA*ENFLNC
65 DO 66 IEVAB=1,NEVAB
      DO 66 JEVAB=1,NEVAB
66 CPSTIF(IEVAB,JEVAB)=ESTIF(IEVAB,JEVAB)+EFSTIF(IEVAB,JEVAB)
      WRITE(5) EFSTIF
      WRITE(6) CPSTIF
67 CONTINUE
C

```

```

C*** STORE THE STIFFNESS MATRIX,STRESS MATRIX
C AND SAMPLING POINT COORDINATES FOR
C EACH ELEMENT ON DISK FILE
  WRITE(1) ESTIF
  WRITE(3) SMATX,GPCOD
70 CONTINUE
  RETURN
  END
  SUBROUTINE MODPB(LPROP)
C
C*** CALCULATES MATRIX OF ELASTIC RIGIDITIES
C FOR PLATE BENDING ELEMENT
C
  COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
  * ASDIS(240),ELOAD(24,24),NOFIX(40),
  * IFPRE(40,3),LNODS(25,8),MATNO(25)
  COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
  * CARTD(2,8),DBMAT(5,24),BMATX(5,24),
  * SMATX(5,24,9),POSGP(3),WEIGP(3),
  * GPCOD(2,9),NEROR(24)
  COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
  * NSTRE,NTYPE,NGAUS,NPROP,NMATS,
  * NVFIX,NEVAB,ICASE,NCASE,ITEMP,
  * IPROB,NPROB
  COMMON/WINKLR/
  * IFND,BMODLS,TAREA,NDTYP,ENFLNC,
  * WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
  * 9EFMAT(8),ELDISP(24,8),NDFRC(24)
  DO 10 ISTR=1,NSTRE
  DO 10 JSTR=1,NSTRE
  DMATX(ISTR,JSTR)=0.0
10 CONTINUE
  YOUNG=PROPS(LPROP,1)
  POISS=PROPS(LPROP,2)
  THICK=PROPS(LPROP,3)
  DMATX(1,1)=YOUNG*(THICK**3)
  */(12.0*(1.0-POISS**2))
  DMATX(2,2)=DMATX(1,1)
  DMATX(1,2)=POISS*DMATX(1,1)
  DMATX(2,1)=DMATX(1,2)
  DMATX(3,3)=(1.0-POISS)*DMATX(1,1)/2.0
  DMATX(4,4)=YOUNG*THICK/(2.4*(1.0+POISS))
  DMATX(5,5)=DMATX(4,4)
  RETURN
  END
  SUBROUTINE SFR2(S,T)
C
C*** CALCULATES SHAPE FUNCTIONS AND THEIR
C DERIVATIVES FOR 2D ELEMENTS
C
  COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
  * ASDIS(240),ELOAD(24,24),NOFIX(40),
  * IFPRE(40,3),LNODS(25,8),MATNO(25)
  COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
  * CARTD(2,8),DBMAT(5,24),BMATX(5,24),
  * SMATX(5,24,9),POSGP(3),WEIGP(3),
  * GPCOD(2,9),NEROR(24)
  COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
  * NSTRE,NTYPE,NGAUS,NPROP,NMATS,
  * NVFIX,NEVAB,ICASE,NCASE,ITEMP,
  * IPROB,NPROB
  COMMON/WINKLR/

```



```

*      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
*      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
*      BEFMAT(8),ELDISP(24,8),NDFRC(24)
S2=S*2.0
T2=T*2.0
SS=S*S
TT=T*T
ST=S*T
SST=S**2*T
STT=S*T**2
ST2=S*T*2.0

C
C*** SHAPE FUNCTIONS
C
SHAPE(1)=(-1.0+ST+SS+TT-SST-STT)/4.0
SHAPE(2)=(1.0-T-SS+SST)/2.0
SHAPE(3)=(-1.0-ST+SS+TT-SST+STT)/4.0
SHAPE(4)=(1.0+S-TT-STT)/2.0
SHAPE(5)=(-1.0+ST+SS+TT+SST+STT)/4.0
SHAPE(6)=(1.0+T-SS-SST)/2.0
SHAPE(7)=(-1.0-ST+SS+TT+SST-STT)/4.0
SHAPE(8)=(1.0-S-TT+STT)/2.0

C
C*** SHAPE FUNCTION DERIVATIVES
C
DERIV(1,1)=(T+S2-ST2-TT)/4.0
DERIV(1,2)=-S+ST
DERIV(1,3)=(-T+S2-ST2+TT)/4.0
DERIV(1,4)=(1.0-TT)/2.0
DERIV(1,5)=(T+S2+ST2+TT)/4.0
DERIV(1,6)=-S-ST
DERIV(1,7)=(-T+S2+ST2-TT)/4.0
DERIV(1,8)=(-1.0+TT)/2.0
DERIV(2,1)=(S+T2-SS-ST2)/4.0
DERIV(2,2)=(-1.0+SS)/2.0
DERIV(2,3)=(-S+T2-SS+ST2)/4.0
DERIV(2,4)=-T-ST
DERIV(2,5)=(S+T2+SS+ST2)/4.0
DERIV(2,6)=(1.0-SS)/2.0
DERIV(2,7)=(-S+T2+SS-ST2)/4.0
DERIV(2,8)=-T+ST
RETURN
END
SUBROUTINE JACOB2(IELEM,DJACB,KGASP)

C
C*** CALCULATES COORDINATES OF GAUSS POINTS
C      AND THE JACOBIAN MATRIX AND ITS DETERMINANT
C      AND THE INVERSE FOR 2D ELEMENTS
C
      DIMENSION XJACM(2,2),XJACI(2,2)
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
*      ASDIS(240),ELOAD(24,24),NOFIX(40),
*      IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
*      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
*      SMATX(5,24,9),POSGP(3),WEIGP(3),
*      GPCOD(2,9),NEROR(24)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
*      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
*      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
*      IPROB,NPROB
      COMMON/WINKLR/

```

```

      *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
      *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
      *      BEFMAT(8),ELDISP(24,8),NDFRC(24)
C
C*** CALCULATE COORDINATES OF SAMPLING POINT
C
      DO 10 IDIME=1,NDIME
      GPCOD(IDIME,KGASP)=0.0
      DO 10 INODE=1,NNODE
      GPCOD(IDIME,KGASP)=GPCOD(IDIME,KGASP)+
      * ELCOD(IDIME,INODE)*SHAPE(INODE)
10 CONTINUE
C
C*** CREATE JACOBIAN MATRIX, XJACM
C
      DO 20 IDIME=1,NDIME
      DO 20 JDIME=1,NDIME
      XJACM(IDIME,JDIME)=0.0
      DO 20 INODE=1,NNODE
      XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME)+
      * DERIV(IDIME,INODE)*ELCOD(JDIME,INODE)
20 CONTINUE
C
C*** CALCULATE DETERMINANT AND INVERSE OF
C JACOBIAN MATRIX
C
      DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*
      * XJACM(2,1)
      IF(DJACB.GT.0.0) GO TO 30
      WRITE(108,900) IELEM
      STOP
30 XJACI(1,1)=XJACM(2,2)/DJACB
      XJACI(2,2)=XJACM(1,1)/DJACB
      XJACI(1,2)=-XJACM(1,2)/DJACB
      XJACI(2,1)=-XJACM(2,1)/DJACB
C
C*** CALCULATE THE CARTESIAN DERIVATIVES
C
      DO 40 IDIME=1,NDIME
      DO 40 INODE=1,NNODE
      CARTD(IDIME,INODE)=0.0
      DO 40 JDIME=1,NDIME
      CARTD(IDIME,INODE)=CARTD(IDIME,INODE)+
      * XJACI(IDIME,JDIME)*DERIV(JDIME,INODE)
40 CONTINUE
900 FORMAT(//,1X,24HPROGRAM HALTED IN JACOB2,
      * /,11X,22H ZERO OR NEGATIVE AREA,/,
      * 10X,16H ELEMENT NUMBER ,15)
      RETURN
      END
      SUBROUTINE BMATPB
C
C*** CALCULATES STRAIN MATRIX B
C FOR PLATE BENDING ELEMENT
C
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
      * ASDIS(240),ELOAD(24,24),NOFIX(40),
      * IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      * CARTD(2,8),DBMAT(5,24),BMATX(5,24),
      * SMATX(5,24,9),POSGP(3),WEIGP(3),
      * GPCOD(2,9),NEROR(24)

```

```

COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
*          NSTRE,NTYPE,NGAUS,NPROP,NMATS,
*          NVFIX,NEVAB,ICASE,NCASE,ITEMP,
*          IPROB,NPROB
COMMON/WINKLR/
*          IFND,BMODLS,TAREA,NDTYP,ENFLNC,
*          WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
*          BEFMAT(8),ELDISP(24,8),NDFRC(24)
DO 10 ISTR=1,NSTRE
DO 10 IEVAB=1,NEVAB
10 BMATX(ISTR,IEVAB)=0.0
JGASH=0
DO 20 INODE=1,NNODE
IGASH=JGASH+1
BMATX(4,IGASH)=CARTD(1,INODE)
BMATX(5,IGASH)=CARTD(2,INODE)
IGASH=IGASH+1
JGASH=IGASH+1
BMATX(1,IGASH)=-CARTD(1,INODE)
BMATX(3,IGASH)=-CARTD(2,INODE)
BMATX(4,IGASH)=-SHAPE(INODE)
BMATX(2,JGASH)=-CARTD(2,INODE)
BMATX(3,JGASH)=-CARTD(1,INODE)
BMATX(5,JGASH)=-SHAPE(INODE)
20 CONTINUE
RETURN
END
SUBROUTINE DBE
C
C*** CALCULATES D*B
C
COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
*          ASDIS(240),ELOAD(24,24),NOFIX(40),
*          IFPRE(40,3),LNODS(25,8),MATNO(25)
COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
*          CARTD(2,8),DBMAT(5,24),BMATX(5,24),
*          SMATX(5,24,9),POSGP(3),WEIGP(3),
*          GPCOD(2,9),NEROR(24)
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
*          NSTRE,NTYPE,NGAUS,NPROP,NMATS,
*          NVFIX,NEVAB,ICASE,NCASE,ITEMP,
*          IPROB,NPROB
COMMON/WINKLR/
*          IFND,BMODLS,TAREA,NDTYP,ENFLNC,
*          WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
*          BEFMAT(8),ELDISP(24,8),NDFRC(24)
DO 10 ISTR=1,NSTRE
DO 10 IEVAB=1,NEVAB
DBMAT(ISTR,IEVAB)=0.0
DO 10 JSTRE=1,NSTRE
DBMAT(ISTR,IEVAB)=DBMAT(ISTR,IEVAB)+
*DMATX(ISTR,JSTRE)*BMATX(JSTRE,IEVAB)
10 CONTINUE
RETURN
END
SUBROUTINE LOADPB
C
C*** CALCULATE NODAL FORCES FOR PLATE ELEMENT
C
DIMENSION TITLE(12),POINT(3)
COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
*          ASDIS(240),ELOAD(24,24),NOFIX(40),

```



```

*          IFPRE(40,3),LNODS(25,8),MATNO(25)
COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
*          CARTD(2,8),DBMAT(5,24),BMATX(5,24),
*          SMATX(5,24,9),POSGP(3),WEIGP(3),
*          GPCOD(2,9),NEROR(24)
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
*          NSTRE,NTYPE,NGAUS,NPROP,NMATS,
*          NVFIX,NEVAB,ICASE,NCASE,ITEMP,
*          IPROB,NPROB
COMMON/WINKLR/
*          IFND,BMODLS,TAREA,NDTYP,ENFLNC,
*          WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
*          BEFMAT(8),ELDISP(24,8),NDFRC(24)
DO 10 IELEM=1,NELEM
DO 10 IEVAB=1,NEVAB
ELOAD(IELEM,IEVAB)=0.0
10 CONTINUE
READ(105,900) TITLE
900 FORMAT(18A4)
WRITE(108,905) TITLE,ICASE
905 FORMAT(1X/1X,18A4,5X,12H LOAD CASE =,I3)
C
C*** READ DATA CONTROLLING LOADING
C   TYPES TO BE INPUT
C
READ(105,910) IPLOD
910 FORMAT(I5)
WRITE(108,915) IPLOD
915 FORMAT(I5)
GO TO 60
C
C*** READ NODAL POINT LOADS
C
15 CONTINUE
IF(IPLOD.EQ.0) GO TO 115
20 READ(105,920)
*LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
WRITE(108,925)
*LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
920 FORMAT(I5,3F10.5)
925 FORMAT(I5,3F10.5)
C
C*** ASSOCIATE THE NODAL POINT LOADS
C   WITH AN ELEMENT
C
DO 30 IELEM=1,NELEM
DO 30 INODE=1,NNODE
NLOCA=LNODS(IELEM,INODE)
IF(LODPT.EQ.NLOCA) GO TO 40
30 CONTINUE
40 DO 50 IDOFN=1,NDOFN
NGASH=(INODE-1)*NDOFN+IDOFN
ELOAD(IELEM,NGASH)=POINT(IDOFN)+ELOAD(IELEM,NGASH)
50 CONTINUE
IF(LODPT.NE.NPOIN) GO TO 20
GO TO 115
60 CONTINUE
C
C*** LOOP OVER EACH ELEMENT
C
DO 110 IELEM=1,NELEM
LPROP=MATNO(IELEM)

```

```

      UDLOD=PROPS(LPROP,4)
      IF(UDLOD.EQ.0.0) GO TO 110
C
C*** EVALUATE THE COORDINATES OF THE
C ELEMENT NODAL POINTS
C
      DO 70 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      DO 70 IDIME=1,NDIME
      ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
70 CONTINUE
      DO 80 IEVAB=1,NEVAB
      ELOAD(IELEM,IEVAB)=0.0
80 CONTINUE
      KGASP=0
C
C*** ENTER LOOPS FOR NUMERICAL INTEGRATION
C
      DO 100 IGAUS=1,NGAUS
      EXISP=POSGP(IGAUS)
      DO 100 JGAUS=1,NGAUS
      ETASP=POSGP(JGAUS)
      KGASP=KGASP+1
C
C*** EVALUATE THE SHAPE FUNCTIONS AT THE
C SAMPLING POINTS AND ELEMENTAL AREA
C
      CALL SFR2(EXISP,ETASP)
      CALL JACOB2(IELEM,DJACB,KGASP)
      DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
C
C*** CALCULATE LODES AND ASSOCIATE WITH
C ELEMENT NODAL POINTS
C
      DO 90 INODE=1,NNODE
      NPOSN=(INODE-1)*NDOFN+1
      ELOAD(IELEM,NPOSN)=ELOAD(IELEM,NPOSN)+
      *SHAPE(INODE)*UDLOD*DAREA
90 CONTINUE
100 CONTINUE
110 CONTINUE
      GO TO 15
115 WRITE(108,930)
930 FORMAT(1H0,5X,
      *36H TOTAL NODAL FORCES FOR EACH ELEMENT)
      DO 120 IELEM=1,NELEM
      WRITE(108,935) IELEM,
      *(ELOAD(IELEM,IEVAB),IEVAB=1,NEVAB)
120 CONTINUE
935 FORMAT(1X,14,5X,8E12.4/(10X,8E12.4)/
      *(10X,8E12.4))
      RETURN
      END
      SUBROUTINE FRONT
      DIMENSION FIXED(240),EQUAT(60),VECRV(240),
      * GLOAD(60),GSTIF(1830),ESTIF(24,24),ELDIS(3,8),
      * IFFIX(240),NACVA(60),LOCEL(24),NDEST(24)
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
      * ASDIS(240),ELOAD(24,24),NOFIX(40),
      * IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      * CARTD(2,8),DBMAT(5,24),BMATX(5,24),

```



```

*          SMATX(5,24,9),POSGP(3),WEIGP(3),
*          GPCOD(2,9),NEROR(24)
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
*          NSTRE,NTYPE,NGAUS,NPROP,NMATS,
*          NVFIX,NEVAB,ICASE,NCASE,ITEMP,
*          IPROB,NPROB
COMMON/WINKLR/
*          IFND,BMODLS,TAREA,NDTYP,ENFLNC,
*          WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
*          BEFMAT(8),ELDISP(24,8),NDFRC(24)
COMMON/RIGID/ NRGD,MAXRGD,REF(5),RGDND(5,80),LNADS(25,8)
NFUNC(I,J)=(J-J-J)/2+I
MFRON=60
MSTIF=1830
C
C*** INTERPRET FIXITY DATA IN VECTOR FORM
C
10  NTOTV=NPOIN*NDOFN
   DO 100 ITOTV=1,NTOTV
     IFFIX(ITOTV)=0
100  FIXED(ITOTV)=0.0
     IF(NVFIX.EQ.0) GO TO 115
     DO 110 IVFIX=1,NVFIX
       NLOCA=(NOFIX(IVFIX)-1)*NDOFN
     DO 110 IDOFN=1,NDOFN
       NGASH=NLOCA+IDOFN
       IFFIX(NGASH)=IFPRE(IVFIX,IDOFN)
110  FIXED(NGASH)=PRESC(IVFIX,IDOFN)
115  CONTINUE
C
C***CHANGE THE SIGN OF THE LAST APPERANCE
C   OF EACH NODE
C
   IF(NRGD.EQ.0) GO TO 118
   DO 112 IELEM=1,NELEM
     DO 112 INODE=1,NNODE
112  LNADS(IELEM,INODE)=LNODS(IELEM,INODE)
     DO 117 INODE=1,NNODE
       DO 117 IELEM=1,NELEM
         DO 117 IRGD=1,NRGD
           INRGD=REF(IRGD)
         DO 117 IMAX=1,MAXRGD
           IRIGID=RGDND(IRGD,IMAX)
           NODE=LNADS(IELEM,INODE)
           IF(NODE.EQ.IRIGID) LNODS(IELEM,INODE)=INRGD
117  CONTINUE
       WRITE(108,735)
735  FORMAT(1X/1X,' REVISED NODAL CONNECTIONS FOR RIGID BODY')
       DO 119 IELEM=1,NELEM
119  WRITE(108,725) IELEM,(LNODS(IELEM,INODE),INODE=1,NNODE)
118  CONTINUE
725  FORMAT(1X,1X,15,5X,815)
     DO 140 IPOIN=1,NPOIN
       KLAST=0
       DO 130 IELEM=1,NELEM
         DO 120 INODE=1,NNODE
           IF(IABS(LNODS(IELEM,INODE)).NE.IPOIN) GO TO 120
           KLAST=IELEM
           NLAST=INODE
120  CONTINUE
130  CONTINUE
       IF(KLAST.NE.0) LNODS(KLAST,NLAST)=-IPOIN

```

```

140 CONTINUE
C
C*** START BY INITIALIZING IMPORTANT QUANTITIES
C   TO ZERO
C
      DO 150 ISTIF=1,MSTIF
150  GSTIF(ISTIF)=0.0
      DO 160 IFRON=1,MFRON
        GLOAD(IFRON)=0.0
        EQUAT(IFRON)=0.0
        VECRV(IFRON)=0.0
160  NACVA(IFRON)=0
      DO 161 IPOIN=1,NPOIN
161  ASDIS(IPOIN)=0.0
C
C*** AND PREPARE FOR DISC READING AND WRITING
C   OPERATIONS
C
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      IF(IFND.EQ.0) GO TO 162
      REWIND 5
      REWIND 6
162 CONTINUE
C
C*** ENTER MAIN ELEMENT ASSEMBLY-REDUCTION LOOP
C
      NFRON=0
      KELVA=0
      DO 380 IELEM=1,NELEM
        KEVAB=0
        IF(IFND.GT.0) READ (6) ESTIF ; GO TO 165
        READ(1) ESTIF
165  CONTINUE
        DO 170 INODE=1,NNODE
          DO 170 IDOFN=1,NDOFN
            NPOSI=(INODE-1)*NDOFN+IDOFN
            LOCNO=LNODS(IELEM,INODE)
            IF(LOCNO.GT.0) LOCEL(NPOSI)=(LOCNO-1)*
              * NDOFN+IDOFN
            IF(LOCNO.LT.0) LOCEL(NPOSI)=(LOCNO+1)*
              * NDOFN-IDOFN
170  CONTINUE
C
C*** START BY LOOKING FOR EXISTING DESTINATIONS
C
      DO 210 IEVAB=1,NEVAB
        NIKNO=IABS(LOCEL(IEVAB))
        KEXIS=0
        DO 180 IFRON=1,NFRON
          IF(NIKNO.NE.NACVA(IFRON)) GO TO 180
          KEVAB=KEVAB+1
          KEXIS=1
          NDEST(KEVAB)=IFRON
180  CONTINUE
          IF(KEXIS.NE.0) GO TO 210
C
C*** WE NOW SEEK NEW EMPTY PLACES FOR
C   DESTINATION VECTOR
C

```

```

DO 190 IFRON=1,MFRON
  IF(NACVA(IFRON).NE.0) GO TO 190
  NACVA(IFRON)=NIKNO
  KEVAB=KEVAB+1
  NDEST(KEVAB)=IFRON
  GO TO 200
190 CONTINUE
C
C*** THE NEW PLACES MAY DEMAND AN INCREASE
C    IN CURRENT FRONTWIDTH
C
200 IF(NDEST(KEVAB).GT.NFRON) NFRON=NDEST(KEVAB)
210 CONTINUE
  IF(NRGD.EQ.0) GO TO 217
  IALPHA=0
  DO 215 IRGD=1,NRGD
  DO 215 INODE=1,NNODE
215 IF(IABS(LNODS(IELEM,INODE)).EQ.REF(IRGD)) IALPHA=IALPHA+1
  IF(IALPHA.EQ.0) GO TO 217
  CALL TRANSF (ESTIF,IELEM)
C
C *** MODIFY NECESSARY ROWS AND COLUMNS IN ESTIF
C
217 CONTINUE
C
C*** ASSEMBLE ELEMENT LOADS
C
  DO 240 IEVAB=1,NEVAB
  IDEST=NDEST(IEVAB)
  GLOAD(IDEST)=GLOAD(IDEST)+ELOAD(IELEM,IEVAB)
C*** ASSEMBLE THE ELEMENT STIFFNESSES
C    BUT NOT IN RESOLUTION
C
  IF(ICASE.GT.1) GO TO 230
  DO 220 JEVAB=1,IEVAB
  JDEST=NDEST(JEVAB)
  IF(ESTIF(IEVAB,JEVAB).EQ.0.0) GO TO 220
  NGASH=NFUNC(IDEST,JDEST)
  NGISH=NFUNC(JDEST,IDEST)
  IF(JDEST.GE.IDEST) GSTIF(NGASH)=
  * GSTIF(NGASH)+ESTIF(IEVAB,JEVAB)
  IF(JDEST.LT.IDEST) GSTIF(NGISH)=
  * GSTIF(NGISH)+ESTIF(IEVAB,JEVAB)
220 CONTINUE
230 CONTINUE
240 CONTINUE
C
C*** RE-EXAMINE EACH ELEMENT NODE, TO
C    ENQUIRE WHICH CAN BE ELIMINATED
C
  DO 370 IEVAB=1,NEVAB
  NIKNO=-LOC(IEVAB)
  IF(NIKNO.LE.0) GO TO 370
C
C*** FIND POSITIONS OF VARIABLES READY
C    FOR ELIMINATION
C
  DO 350 IFRON=1,NFRON
  IF(NACVA(IFRON).NE.NIKNO) GO TO 350
C
C*** EXTRACT THE COEFFICIENTS OF THE
C    NEW EQUATION FOR ELIMINATION

```



```

C
  IF(ICASE.GT.1) GO TO 260
  DO 250 JFRON=1,MFRON
  IF(IFRON.LT.JFRON) NLOCA=NFUNC(IFRON,JFRON)
  IF(IFRON.GE.JFRON) NLOCA=NFUNC(JFRON,IFRON)
  EQUAT(JFRON)=GSTIF(NLOCA)
250 GSTIF(NLOCA)=0.0
260 CONTINUE
C
C*** AND EXTRACT CORRESPONDING RIGHT
C HAND SIDES
C
  EQRHS=GLOAD(IFRON)
  GLOAD(IFRON)=0.0
  KELVA=KELVA+1
C
C*** WRITE EQUATIONS TO DISK OR TAPE
C
  IF(ICASE.GT.1) GO TO 270
  WRITE(2) EQUAT,EQRHS,IFRON,NIKNO
  GO TO 280
270 WRITE(4) EQRHS
  READ(2) EQUAT,DUMMY,IDUMM,NIKNO
280 CONTINUE
C
C*** DEAL WITH PIVOT
C
  PIVOT=EQUAT(IFRON)
  EQUAT(IFRON)=0.0
C
C*** ENQUIRE WHETHER PRESENT VARIABLE IS
C FREE OR PRESCRIBED
C
  IF(IFFIX(NIKNO).EQ.0) GO TO 300
C
C*** DEAL WITH A PRESCRIBED DEFLECTION
C
  DO 290 JFRON=1,NFRON
290 GLOAD(JFRON)=GLOAD(JFRON)-FIXED(NIKNO)*
  * EQUAT(JFRON)
  GO TO 340
C
C*** ELIMINATE A FREE VARIABLE-DEAL WITH THE
C RIGHT HAND SIDE FIRST
C
300 DO 330 JFRON=1,NFRON
  GLOAD(JFRON)=GLOAD(JFRON)-EQUAT(JFRON)*
  * EQRHS/PIVOT
C
C*** NOW DEAL WITH THE COEFFICIENTS IN CORE
C
  IF(ICASE.GT.1) GO TO 320
  IF(EQUAT(JFRON).EQ.0.0) GO TO 330
  NLOCA=NFUNC(0,JFRON)
  DO 310 LFRON=1,JFRON
  NGASH=LFRON+NLOCA
310 GSTIF(NGASH)=GSTIF(NGASH)-EQUAT(JFRON)*
  * EQUAT(LFRON)/PIVOT
320 CONTINUE
330 CONTINUE
340 EQUAT(IFRON)=PIVOT
C

```

```

C*** RECORD THE NEW VACANT SPACE, AND REDUCE
C FRONTWIDTH IF POSSIBLE
C
      NACVA(IFRON)=0
      GO TO 360
C
C*** COMPLETE THE ELEMENT LOOP IN THE FORWARD
C ELIMINATION
C
      350 CONTINUE
      360 IF(NACVA(NFRON).NE.0) GO TO 370
          NFRON=NFRON-1
          IF(NFRON.GT.0) GO TO 360
      370 CONTINUE
      380 CONTINUE
C
C*** ENTER BACK SUBSTITUTION PHASE, LOOP
C BACKWARDS THROUGH VARIABLES
C
      DO 410 IELVA=1,KELVA
C
C*** READ A NEW EQUATION
C
      BACKSPACE 2
      READ(2) EQUAT,EQRHS,IFRON,NIKNO
      BACKSPACE 2
      IF(ICASE.EQ.1) GO TO 390
      BACKSPACE 4
      READ(4) EQRHS
      BACKSPACE 4
      390 CONTINUE
C
C*** PREPARE TO BACKSUBSTITUTE FROM
C THE CURRENT EQUATION
C
      PIVOT=EQUAT(IFRON)
      IF(IFFIX(NIKNO).EQ.1) VECRV(IFRON)=
      * FIXED(NIKNO)
      IF(IFFIX(NIKNO).EQ.0) EQUAT(IFRON)=0.0
C
C*** BACK-SUBSTITUTE INTO THE CURRENT EQUATION
C
      DO 400 JFRON=1,MFRON
      400 EQRHS=EQRHS-VECRV(JFRON)*EQUAT(JFRON)
C
C*** PUT THE FINAL VALUES WHERE THEY BELONG
C
      IF(IFFIX(NIKNO).EQ.0) VECRV(IFRON)=
      * EQRHS/PIVOT
      IF(IFFIX(NIKNO).EQ.1) FIXED(NIKNO)=-EQRHS
      ASDIS(NIKNO)=VECRV(IFRON)
      410 CONTINUE
      IF(NRGD.EQ.0) GO TO 419
      DO 404 IRGD=1,NRGD
      DO 404 IELEM=1,NELEM
      DO 404 INODE=1,NNODE
      IF(IABS(LNODS(IELEM,INODE)).EQ.REF(IRGD)) GO TO 402
      GO TO 404
      402 NODE=LNADS(IELEM,INODE)
      IF(LNADS(IELEM,INODE).EQ.REF(IRGD)) GO TO 404
      IDUM=3*(NODE-1)+1
      ASDIS(IDUM)=0.0

```

```

      ASDIS(IDUM+1)=0.0
      ASDIS(IDUM+2)=0.0
404  CONTINUE
      WRITE(108,700)
730  FORMAT(1X/1X,' DISPLACEMENTS OF REFERENCE NODES AND FREE
      * NODES ONLY')
      DO 401 IPOIN=1,NPOIN
      NGASH=IPOIN*NDOFN
      NGISH=NGASH-NDOFN+1
401  WRITE(108,920) IPOIN,(ASDIS(IGASH),IGASH=NGISH,NGASH)
      DO 416 IELEM=1,NELEM
      DO 416 INODE=1,NNODE
      DO 415 IRGD=1,NRGD
      DO 415 IMAX=1,MAXRGD
415  IF(IABS(LNODS(IELEM,INODE)).EQ.REF(IRGD)) GO TO 411
      GO TO 416
411  NODE=LNADS(IELEM,INODE)
      REFNO=REF(IRGD)
      X=-(COORD(NODE,1)-COORD(REFNO,1))
      Y=COORD(NODE,2)-COORD(REFNO,2)
      IDUM=3*(NODE-1)+1
      JDUM=3*(REFNO-1)+1
      ASDIS(IDUM)=ASDIS(JDUM)-X*ASDIS(JDUM+1)+Y*ASDIS(JDUM+2)
      ASDIS(IDUM+1)=ASDIS(JDUM+1)
      ASDIS(IDUM+2)=ASDIS(JDUM+2)
      IF(IFFIX(IDUM).EQ.0) GO TO 425
      FIXED(IDUM)=FIXED(JDUM)
425  IF(IFFIX(IDUM+1).EQ.0) GO TO 426
      FIXED(IDUM+1)=FIXED(JDUM)*X+FIXED(JDUM+1)
426  IF(FIXED(IDUM+2).EQ.0) GO TO 416
      FIXED(IDUM+2)=FIXED(JDUM)*(-Y)+FIXED(JDUM+2)
416  CONTINUE
      DO 418 IELEM=1,NELEM
      DO 418 INODE=1,NNODE
      LNODS(IELEM,INODE)=LNADS(IELEM,INODE)
418  CONTINUE
419  CONTINUE
      WRITE(108,900)
900  FORMAT(1H0,5X,13HDISPLACEMENTS)
      IF(NDOFN.NE.2) GO TO 430
      IF(NDIME.NE.1) GO TO 420
      WRITE(108,905)
905  FORMAT(1H0,5X,4HNODE,6X,5HDISP.,7X,
      * 8HROTATION)
      GO TO 440
420  WRITE(108,910)
910  FORMAT(1H0,5X,4HNODE,5X,7HX-DISP.,
      * 7X,7HY-DISP.)
      GO TO 440
430  WRITE(108,915)
915  FORMAT(1H0,5X,4HNODE,6X,5HDISP.,8X,
      * 7HXZ-ROT.,7X,7HYZ-ROT.)
440  CONTINUE
      DO 450 IPOIN=1,NPOIN
      NGASH=IPOIN*NDOFN
      NGISH=NGASH-NDOFN+1
450  WRITE(108,920) IPOIN,(ASDIS(IGASH),IGASH=
      * NGISH,NGASH)
920  FORMAT(1H0,3E14.6)
      WRITE(108,925)
925  FORMAT(1H0,5X,9HREACTIONS)
      IF(NDOFN.NE.2) GO TO 470

```



```

      IF(NDIME.NE.1) GO TO 460
      WRITE(108,930)
930  FORMAT(1H0,5X,4HNODE,6X,5HFORCE,8X,6HMOMENT)
      GO TO 480
460  WRITE(108,935)
935  FORMAT(1H0,5X,4HNODE,5X,7HX-FORCE,7X,
* 7HY-FORCE)
      GO TO 480
470  WRITE(108,940)
940  FORMAT(1H0,5X,4HNODE,6X,5HFORCE,6X,
* 9HXZ-MOMENT,5X,9HYZ-MOMENT)
480  CONTINUE
      DO 510 IPOIN=1,NPOIN
      NLOCA=(IPOIN-1)*NDOFN
      DO 490 IDOFN=1,NDOFN
      NGUSH=NLOCA+IDOFN
      IF(IFFIX(NGUSH).GT.0) GO TO 500
490  CONTINUE
      GO TO 510
500  NGASH=NLOCA+NDOFN
      NGISH=NLOCA+1
      WRITE(108,945) IPOIN,(FIXED(IGASH),IGASH=
* NGISH,NGASH)
510  CONTINUE
945  FORMAT(110,3E14.6)
C
C *** POST FRONT-RESET ALL ELEMENT CONNECTION
C NUMBERS TO POSITIVE VALUES FOR SUBSEQUENT
C USE IN STRESS CALCULATION
C
      DO 520 IELEM=1,NELEM
      DO 520 INODE=1,NNODE
520  LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
      RETURN
      END
      SUBROUTINE TRANSF (ESTIF,IELEM)
      DIMENSION ESTIF(24,24)
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
* ASDIS(240),ELOAD(24,24),NOFIX(40),
* IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
* CARTD(2,8),DBMAT(5,24),BMATX(5,24),
* SMATX(5,24,9),POSGP(3),WEIGP(3),
* GPCOD(2,9),NEROR(24)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
* NSTRE,NTYPE,NGAUS,NPROP,NMATS,
* NVFIX,NEVAB,ICASE,NCASE,ITEMP,
* IPROB,NPROB
      COMMON/WINKLR/
* IFND,BMODLS,TAREA,NDTYP,ENFLNC,
* WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
* BEFMAT(8),ELDISP(24,8),NDFRC(24)
      COMMON/RIGID/ NRGD,MAXRGD,REF(5),RGDND(5,80),LNADS(25,8)
C
C *** MODIFY NECESSARY ROWS AND COLUMNS IN ESTIF
C
      DO 240 IRGD=1,NRGD
      KDUM=0
      LDUM=0
100  CONTINUE
110  DO 220 INODE=1,NNODE
      IF(IABS(LNODS(IELEM,INODE)).EQ.REF(IRGD)) GO TO 130

```



```

GO TO 220
130 NODE=LNADS(IELEM,INODE)
REFNOD=IABS(LNODS(IELEM,INODE))
X=-(COORD(NODE,1)-COORD(REFNOD,1))
Y=COORD(NODE,2)-COORD(REFNOD,2)
IDUM=3*(INODE-1)+1
IF(KDUM.EQ.1) GO TO 140
EL1=ELOAD(IELEM,IDUM)
EL2=ELOAD(IELEM,IDUM+1)
EL3=ELOAD(IELEM,IDUM+2)
ELOAD(IELEM,IDUM+1)=EL2-EL1*X
ELOAD(IELEM,IDUM+2)=(EL1*Y)+EL3
140 CONTINUE
IF(LDUM.GE.1) GO TO 150
LLDUM=3*(INODE-1)+1
150 LDUM=LDUM+1
DO 200 JNODE=1,NNODE
JDUM=3*(JNODE-1)+1
IF(KDUM.EQ.1) IDUM=JDUM;JDUM=3*(INODE-1)+1
E1=ESTIF(JDUM,IDUM)
E2=ESTIF(JDUM,IDUM+1)
E3=ESTIF(JDUM,IDUM+2)
E4=ESTIF(JDUM+1,IDUM)
E5=ESTIF(JDUM+1,IDUM+1)
E6=ESTIF(JDUM+1,IDUM+2)
E7=ESTIF(JDUM+2,IDUM)
E8=ESTIF(JDUM+2,IDUM+1)
E9=ESTIF(JDUM+2,IDUM+2)
IF(KDUM.EQ.1) GO TO 170
C
C*** IF KDUM=0, MODIFY COLUMNS,LOAD VECTOR,AND THE DISPLACEMENT VECTOR
C IF KDUM=1, MODIFY ONLY THE ROWS IN ESTIF
C
ESTIF(JDUM,IDUM)=E1
ESTIF(JDUM,IDUM+1)=(-E1*X)+E2
ESTIF(JDUM,IDUM+2)=E1*Y+E3
ESTIF(JDUM+1,IDUM)=E4
ESTIF(JDUM+1,IDUM+1)=(-E4*X)+E5
ESTIF(JDUM+1,IDUM+2)=E4*Y+E6
ESTIF(JDUM+2,IDUM)=E7
ESTIF(JDUM+2,IDUM+1)=(-E7*X)+E8
ESTIF(JDUM+2,IDUM+2)=E7*Y+E9
GO TO 200
170 CONTINUE
ESTIF(JDUM,IDUM)=E1
ESTIF(JDUM,IDUM+1)=E2
ESTIF(JDUM,IDUM+2)=E3
ESTIF(JDUM+1,IDUM)=(-E1*X)+E4
ESTIF(JDUM+1,IDUM+1)=(-E2*X)+E5
ESTIF(JDUM+1,IDUM+2)=(-E3*X)+E6
ESTIF(JDUM+2,IDUM)=E1*Y+E7
ESTIF(JDUM+2,IDUM+1)=E2*Y+E8
ESTIF(JDUM+2,IDUM+2)=E3*Y+E9
200 CONTINUE
220 CONTINUE
KDUM=KDUM+1
IF(KDUM.EQ.1) GO TO 100
240 CONTINUE
DO 260 IRGD=1,NRGD
LDUM=0
DO 250 INODE=1,NNODE
IF(IABS(LNODS(IELEM,INODE)).EQ.REF(IRGD)) GO TO 221

```

```

GO TO 250
221 IDUM=3*(INODE-1)+1
IF(LDUM.GE.1) GO TO 222
LLDUM=IDUM
222 LDUM=LDUM+1
DO 230 JNODE=1,NNODE
IF(IABS(LNODS(IELEM,JNODE)).NE.REF(IRGD)) GO TO 230
JDUM=3*(JNODE-1)+1
IF(IDUM.EQ.LLDUM.AND.JDUM.EQ.LLDUM) GO TO 230
IF(INODE.EQ.JNODE) GO TO 225
ESTIF(LLDUM,LLDUM)=ESTIF(LLDUM,LLDUM)
* +ESTIF(IDUM,JDUM)+ESTIF(JDUM,IDUM)
ESTIF(LLDUM,LLDUM+1)=ESTIF(LLDUM,LLDUM+1)
* +ESTIF(IDUM,JDUM+1)+ESTIF(JDUM,IDUM+1)
ESTIF(LLDUM,LLDUM+2)=ESTIF(LLDUM,LLDUM+2)
* +ESTIF(IDUM,JDUM+2)+ESTIF(JDUM,IDUM+2)
ESTIF(LLDUM+1,LLDUM)=ESTIF(LLDUM+1,LLDUM)
* +ESTIF(IDUM+1,JDUM)+ESTIF(JDUM+1,IDUM)
ESTIF(LLDUM+1,LLDUM+1)=ESTIF(LLDUM+1,LLDUM+1)
* +ESTIF(IDUM+1,JDUM+1)+ESTIF(JDUM+1,IDUM+1)
ESTIF(LLDUM+1,LLDUM+2)=ESTIF(LLDUM+1,LLDUM+2)
* +ESTIF(IDUM+1,JDUM+2)+ESTIF(JDUM+1,IDUM+2)
ESTIF(LLDUM+2,LLDUM)=ESTIF(LLDUM+2,LLDUM)
* +ESTIF(IDUM+2,JDUM)+ESTIF(JDUM+2,IDUM)
ESTIF(LLDUM+2,LLDUM+1)=ESTIF(LLDUM+2,LLDUM+1)
* +ESTIF(IDUM+2,JDUM+1)+ESTIF(JDUM+2,IDUM+1)
ESTIF(LLDUM+2,LLDUM+2)=ESTIF(LLDUM+2,LLDUM+2)
* +ESTIF(IDUM+2,JDUM+2)+ESTIF(JDUM+2,IDUM+2)
GO TO 228
225 CONTINUE
ESTIF(LLDUM,LLDUM)=ESTIF(LLDUM,LLDUM)
* +ESTIF(IDUM,JDUM)
ESTIF(LLDUM,LLDUM+1)=ESTIF(LLDUM,LLDUM+1)
* +ESTIF(IDUM,JDUM+1)
ESTIF(LLDUM,LLDUM+2)=ESTIF(LLDUM,LLDUM+2)
* +ESTIF(IDUM,JDUM+2)
ESTIF(LLDUM+1,LLDUM)=ESTIF(LLDUM+1,LLDUM)
* +ESTIF(IDUM+1,JDUM)
ESTIF(LLDUM+1,LLDUM+1)=ESTIF(LLDUM+1,LLDUM+1)
* +ESTIF(IDUM+1,JDUM+1)
ESTIF(LLDUM+1,LLDUM+2)=ESTIF(LLDUM+1,LLDUM+2)
* +ESTIF(IDUM+1,JDUM+2)
ESTIF(LLDUM+2,LLDUM)=ESTIF(LLDUM+2,LLDUM)
* +ESTIF(IDUM+2,JDUM)
ESTIF(LLDUM+2,LLDUM+1)=ESTIF(LLDUM+2,LLDUM+1)
* +ESTIF(IDUM+2,JDUM+1)
ESTIF(LLDUM+2,LLDUM+2)=ESTIF(LLDUM+2,LLDUM+2)
* +ESTIF(IDUM+2,JDUM+2)
228 CONTINUE
DO 229 I=IDUM,IDUM+2
DO 229 J=JDUM,JDUM+2
ESTIF(I,J)=0.0
229 ESTIF(J,I)=0.0
230 CONTINUE
250 CONTINUE
260 CONTINUE
RETURN
END
SUBROUTINE STREPB

```

```

C
C*** CALCULATES THE STRESS RESULTANTS AT
C GAUSS POINTS FOR PLATE BENDING ELEMENT

```

```

C
  DIMENSION ELDIS(3,8),STRSG(5)
  COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
  *      ASDIS(240),ELOAD(24,24),NOFIX(40),
  *      IFPRE(40,3),LNODS(25,8),MATNO(25)
  COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
  *      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
  *      SMATX(5,24,9),POSGP(3),WEIGP(3),
  *      GPCOD(2,9),NEROR(24)
  COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
  *      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
  *      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
  *      IPROB,NPROB
  COMMON/WINKLR/
  *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
  *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
  *      BEFMAT(8),ELDISP(24,8),BNDFRC(24)
  WRITE(108,900)
  WRITE(108,905)

C
C*** LOOP OVER EACH ELEMENT
C
  DO 40 IELEM=1,NELEM
C
C*** READ THE STRESS MATRIX, SAMPLING POINT
C  COORDINATES FOR THE ELEMENT
C
  READ(3) SMATX,GPCOD
  WRITE(108,910) IELEM

C
C*** IDENTIFY THE DISPLACEMENTS OF THE
C  ELEMENT NODAL POINTS
C
  DO 10 INODE=1,NNODE
  LNODE=LNODS(IELEM,INODE)
  NPOSN=(LNODE-1)*NDOFN
  DO 10 IDOFN=1,NDOFN
  NPOSN=NPOSN+1
  ELDIS(IDOFN,INODE)=ASDIS(NPOSN)
10 CONTINUE
  KGASP=0

C
C*** ENTER LOOPS OVER EACH SAMPLING POINT
C]
  DO 30 IGAUS=1,NGAUS
  DO 30 JGAUS=1,NGAUS
  KGASP=KGASP+1
  DO 20 ISTRE=1,NSTRE
  STRSG(ISTRE)=0.0
  KGASH=0

C
C*** COMPUTE THE STRESS RESULTANTS
C
  DO 20 INODE=1,NNODE
  DO 20 IDOFN=1,NDOFN
  KGASH=KGASH+1
  STRSG(ISTRE)=STRSG(ISTRE)+
  *SMATX(ISTRE,KGASH,KGASP)*ELDIS(IDOFN,INODE)
20 CONTINUE

C
C*** OUTPUT THE STRESS RESULTANTS
C

```



```

      WRITE(108,915) KGASP,
      *(GPCOD(IDIME,KGASP),IDIME=1,NDIME),
      *(STRSG(ISTRE),ISTRE=1,NSTRE)
30  CONTINUE
40  CONTINUE
900 FORMAT(/,10X,8HSTRESSES,/)
905 FORMAT(1H0,4HG,P.,2X,8HX-COORD.,2X,
      *8HY-COORD.,3X,8HX-MOMENT,4X,8HY-MOMENT,
      *3X,9HXY-MOMENT,2X,10HXZ-S,FORCE,2X,
      *10HYZ-S,FORCE)
910 FORMAT(/,5X,12HELEMENT NO.=,15)
915 FORMAT(15,2F10.4,5E12.5)
      RETURN
      END
      SUBROUTINE FPRES
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
      *      ASDIS(240),ELOAD(24,24),NOFIX(40),
      *      IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      *      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
      *      SMATX(5,24,9),POSGP(3),WEIGP(3),
      *      GPCOD(2,9),NEROR(24)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
      *      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
      *      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
      *      IPROB,NPROB
      COMMON/WINKLR/
      *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
      *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
      *      BEFMAT(8),ELDISP(24,8),BNDFRC(24)
      DIMENSION ELDSP(24)
      WRITE(108,820)
      WRITE(108,830)
      DO 50 IELEM=1,NELEM
      BMODLS=WINKMD(IELEM)
      DO 5 IEVAB=1,NEVAB
5  ELDSP(IEVAB)=0.0
      READ (5) EFSTIF
      WRITE(108,810) IELEM
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      NPOSN=(LNODE-1)*NDOFN
      DO 10 IDOFN=1,NDOFN
      NPOSN=NPOSN+1
      DOFRD=3*(INODE-1)+IDOFN
      ELDSP(DOFRD)=ASDIS(NPOSN)
10  CONTINUE
      DO 30 I=1,NNODE
      K=3*I-2
      BNDFRC(I)=ELDSP(K)*BMODLS
      IF(EFSTIF(K,K).EQ.0.0) BNDFRC(I)=0.0
30  CONTINUE
      DO 40 I=1,NNODE
      LNODE=LNODS(IELEM,I)
      WRITE(108,840) LNODE,(BNDFRC(I))
40  CONTINUE
50  CONTINUE
810 FORMAT(1X/1X,'ELEMENT NUMBER :',2X,13)
820 FORMAT(1X/1X,'WINKLER FOUNDATION NODE POINT FORCES')
830 FORMAT(1X/25X,'NODE',4X,'NORMAL')
840 FORMAT(1X,25X,13,3X,E12.4)
      RETURN

```

```

      END
      SUBROUTINE CHECK1
C
C*** TO CRITISIZE THE DATA CONTROL CARD AND
C PRINT ANY DIAGNOSTICS
C
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
      *      ASDIS(240),ELOAD(24,24),NOFIX(40),
      *      IFPRE(40,3),LNODS(25,8),MATNO(25)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      *      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
      *      SMATX(5,24,9),POSGP(3),WEIGP(3),
      *      GPCOD(2,9),NEROR(24)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
      *      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
      *      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
      *      IPROB,NPROB
      COMMON/WINKLR/
      *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
      *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
      *      BEFMAT(8),ELDISP(24,8),NDFRC(24)
      DO 10 IEROR=1,24
10 NEROR(IEROR)=0
C
C*** CREATE THE DIAGNOSTIC MESSAGES
C
      IF(NPOIN.LE.0) NEROR(1)=1
      IF(NELEM*NNODE.LT.NPOIN) NEROR(2)=1
      IF(NVFIX.LT.1.AND.IFND.EQ.0) NEROR(3)=1
      IF(NVFIX.GT.NPOIN) NEROR(3)=1
      IF(NCASE.LE.0) NEROR(4)=1
      IF(NTYPE.LT.0.OR.NTYPE.GT.2) NEROR(5)=1
      IF(NNODE.LT.3.OR.NNODE.GT.8) NEROR(6)=1
      IF(NDOFN.LT.2.OR.NDOFN.GT.3) NEROR(7)=1
      IF(NMATS.LE.0.OR.NMATS.GT.NELEM) NEROR(8)=1
      IF(NPROP.LT.3.OR.NPROP.GT.5) NEROR(9)=1
      IF(NGAUS.LT.2.OR.NGAUS.GT.3) NEROR(10)=1
      IF(NDIME.LT.1.OR.NDIME.GT.2) NEROR(11)=1
      IF(NSTRE.LT.2.OR.NSTRE.GT.5) NEROR(12)=1
C
C*** EITHER RETURN, OR ELSE PRINT THE ERRORS
C DIAGNOSED
C
      KEROR=0
      DO 20 IEROR=1,12
      IF(NEROR(IEROR).EQ.0) GO TO 20
      KEROR=1
      WRITE(108,900) IEROR
900 FORMAT(//25H *** DIAGNOSIS BY CHECK1,
      * 6H ERROR,I3)
      20 CONTINUE
      IF(KEROR.EQ.0) RETURN
C
C*** OTHERWISE ECHO ALL THE REMAINING DATA
C WITHOUT FURTHER COMMENT
C
      CALL ECHO
      END
      SUBROUTINE ECHO
      DIMENSION NTITL(80)
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRES(40,3),
      *      ASDIS(240),ELOAD(24,24),NOFIX(40),

```

```

      *      IFPRE(40,3),LNODS(25,8),MATNO(25)
      *      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      *      *      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
      *      *      SMATX(5,24,9),POSGP(3),WEIGP(3),
      *      *      GPCOD(2,9),NEROR(24)
      *      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
      *      *      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
      *      *      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
      *      *      IPROB,NPROB
      *      COMMON/WINKLR/
      *      *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
      *      *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
      *      *      BEFMAT(8),ELDISP(24,8),NDFRC(24)
      *      WRITE(108,900)
900  FORMAT(/25H NOW FOLLOWS A LISTING OF,
      * 25H POST-DISASTER DATA CARDS/)
      10 READ(105,905) NTITL
905  FORMAT(80A1)
      WRITE(108,910) NTITL
910  FORMAT(20X,80A1)
      GO TO 10
      END
      SUBROUTINE CHECK2
C
C*** TO CRITICIZE THE DATA FROM SUBROUTINE INPUT
C
      DIMENSION NDFRO(25)
      COMMON/LGDATA/COORD(80,2),PROPS(10,4),PRESC(40,3),
      *      *      ASDIS(240),ELOAD(24,24),NOFIX(40),
      *      *      IFPRE(40,3),LNODS(25,8),MATNO(25)
      *      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(5,5),
      *      *      CARTD(2,8),DBMAT(5,24),BMATX(5,24),
      *      *      SMATX(5,24,9),POSGP(3),WEIGP(3),
      *      *      GPCOD(2,9),NEROR(24)
      *      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,
      *      *      NSTRE,NTYPE,NGAUS,NPROP,NMATS,
      *      *      NVFIX,NEVAB,ICASE,NCASE,ITEMP,
      *      *      IPROB,NPROB
      *      COMMON/WINKLR/
      *      *      IFND,BMODLS,TAREA,NDTYP,ENFLNC,
      *      *      WINKMD(25),NDTYPE(80),EFSTIF(24,24),CPSTIF(24,24),
      *      *      BEFMAT(8),ELDISP(24,8),NDFRC(24)
      *      MFRON=60
C
C*** CHECK AGAINST TO IDENTICAL NONZERO
C  NODAL COORDINATES
C
      DO 10 IELEM=1,NELEM
10  NDFRO(IELEM)=0
      DO 40 IPOIN=2,NPOIN
      KPOIN=IPOIN-1
      DO 30 JPOIN=1,KPOIN
      DO 20 IDIME=1,NDIME
      IF(COORD(IPOIN,IDIME).NE.COORD(JPOIN,
      * IDIME))GO TO 30
      20 CONTINUE
      NEROR(13)=NEROR(13)+1
      30 CONTINUE
      40 CONTINUE
C
C*** CHECK THE LIST OF ELEMENT PROPERTY NUMBERS
C

```



```

      DO 50 IELEM=1,NELEM
50  IF(MATNO(IELEM).LE.0.OR.MATNO(IELEM).GT.
      * NMATS) NEROR(14)=NEROR(14)+1
C
C*** CHECK FOR IMPOSSIBLE NODE NUMBERS
C
      DO 70 IELEM=1,NELEM
      DO 60 INODE=1,NNODE
      IF(LNODS(IELEM,INODE).EQ.0) NEROR(15)=
      * NEROR(15)+1
60  IF(LNODS(IELEM,INODE).LT.0.OR.LNODS(IELEM,
      * INODE).GT.NPOIN) NEROR(16)=NEROR(16)+1
70  CONTINUE
C
C*** CHECK FOR ANY REPETITION OF A NODE
      NUMBER WITHIN AN ELEMENT
C
      DO 140 IPOIN=1,NPOIN
      KSTAR=0
      DO 100 IELEM=1,NELEM
      KZERO=0
      DO 90 INODE=1,NNODE
      IF(LNODS(IELEM,INODE).NE.IPOIN) GO TO 90
      KZERO=KZERO+1
      IF(KZERO.GT.1) NEROR(17)=NEROR(17)+1
C
C*** SEEK FIRST, LAST, AND INTERMEDIATE
      APPEARANCES OF NODE IPOIN
C
      IF(KSTAR.NE.0) GO TO 80
      KSTAR=IELEM
C
C*** CALCULATE INCREASE OR DECREASE IN
      FRONTWIDTH AT EACH ELEMENT STAGE
C
      NDFRO(IELEM)=NDFRO(IELEM)+NDOFN
80  CONTINUE
C
C*** AND CHANGE THE SIGN OF THE LAST
      APPERANCE OF EACH NODE
C
      KLAST=IELEM
      NLAST=INODE
90  CONTINUE
100 CONTINUE
      IF(KSTAR.EQ.0) GO TO 110
      IF(KLAST.LT.NELEM) NDFRO(KLAST+1)=
      * NDFRO(KLAST+1)-NDOFN
      LNODS(KLAST,NLAST)=-IPOIN
      GO TO 140
C
C*** CHECK THAT COORDINATES FOR AN UNUSED
      NODE HAVE NOT BEEN SPECIFIED
C
110 WRITE(108,900) IPOIN
900 FORMAT(/15H CHECK WHY NODE,14,
      * 14H NEVER APPEARS)
      NEROR(18)=NEROR(18)+1
      SIGMA=0.0
      DO 120 IDIME=1,NDIME
120  SIGMA=SIGMA+ABS(COORD(IPOIN,IDIME))
      IF(SIGMA.NE.0.0) NEROR(19)=NEROR(19)+1

```



```

C
C*** CHECK THAT AN UNUSED NODE NUMBER IS NOT
C    A RESTRAINED NODE
C
      IF(IFND.GT.0.AND.NVFIX.EQ.0) GO TO 140
      DO 130 IVFIX=1,NVFIX
130  IF(NOFIX(IVFIX).EQ.IPOIN) NEROR(20)=
      * NEROR(20)+1
140  CONTINUE
C
C*** CALCULATE THE LARGEST FRONTWIDTH
C
      NFRON=0
      KFRON=0
      DO 150 IELEM=1,NELEM
      NFRON=NFRON+NDFRO(IELEM)
150  IF(NFRON.GT.KFRON) KFRON=NFRON
      WRITE(108,905) KFRON
905  FORMAT(//,1X,28HMAX FRONTWIDTH ENCOUNTERED =,15)
      IF(KFRON.GT.MFRON) NEROR(21)=1
C
C*** CONTINUE CHECKING THE DATA FOR THE FIXED VALUES
C
      IF(NVFIX.EQ.0.AND.IFND.GE.1) GO TO 175
      DO 170 IVFIX=1,NVFIX
      IF(NOFIX(IVFIX).LE.0.OR.NOFIX(IVFIX).
      * GT.NPOIN) NEROR(22)=NEROR(22)+1
      KOUNT=0
      DO 160 IDOFN=1,NDOFN
160  IF(IFPRE(IVFIX,IDOFN).GT.0) KOUNT=1
      IF(KOUNT.EQ.0) NEROR(23)=NEROR(23)+1
      KVFIX=IVFIX-1
      DO 170 JVFIX=1,KVFIX
170  IF(IVFIX.NE.1.AND.NOFIX(IVFIX).EQ.
      * NOFIX(JVFIX)) NEROR(24)=NEROR(24)+1
175  CONTINUE
      KEROR=0
      DO 180 IEROR=13,24
      IF(NEROR(IEROR).EQ.0) GO TO 180
      KEROR=1
      WRITE(108,910) IEROR,NEROR(IEROR)
910  FORMAT(//30H*** DIAGNOSIS BY CHECK2, ERROR,
      * 13,6X,18H ASSOCIATED NUMBER,15)
180  CONTINUE
      IF(KEROR.NE.0) GO TO 200
C
C*** RETURN ALL NODAL CONNECTION NUMBERS TO
C    POSITIVE VALUES
C
      DO 190 IELEM=1,NELEM
      DO 190 INODE=1,NNODE
190  LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
      RETURN
200  CALL ECHO
      END

```

0 TOTAL NO. OF PROBLEMS = 1

PROBLEM NO. 1

*** WANG AND SALMON---20.13.2 ***

INTEGRATED WINKLER FOUNDATION

NPOIN = 70 NELEM = 15 NVFIX = 19 NCASE = 1 NTYPE = 0 NNODE = 8 NDOFN = 3
NMATS = 3 NPROP = 4 NGAUS = 3 NDIME = 2 NSTRE = 5 NEVAB = 24

ELEMENT	PROPERTY	NODE NUMBERS								SOIL MODULUS
1	1	3	7	11	12	13	8	5	4	.17500
2	2	1	6	9	10	11	7	3	2	.17500
3	1	11	15	19	20	21	16	13	12	.17500
4	1	9	14	17	18	19	15	11	10	.17500
5	1	19	23	27	28	29	24	21	20	.17500
6	1	17	22	25	26	27	23	19	18	.17500
7	1	25	30	32	33	34	31	27	26	.17500
8	1	32	35	37	38	39	36	34	33	.17500
9	1	37	40	42	43	44	41	39	38	.17500
10	1	42	47	50	51	52	48	44	43	.17500
11	1	44	48	52	53	54	49	46	45	.17500
12	3	50	55	58	59	60	56	52	51	.17500
13	1	52	56	60	61	62	57	54	53	.17500
14	1	58	63	66	67	68	64	60	59	.17500
15	1	60	64	68	69	70	65	62	61	.17500

NODAL POINT COORDINATES		
NODE	X	Y
1	.000	.000
2	.000	3.000
3	.000	6.000

Appendix C: Cantilever Footing

4	.000	25.000
5	.000	44.000
6	6.000	.000
7	6.000	6.000
8	6.000	44.000
9	12.000	.000
10	12.000	3.000
11	12.000	6.000
12	12.000	25.000
13	12.000	44.000
14	22.500	.000
15	22.500	7.000
16	22.500	44.000
17	33.000	.000
18	33.000	4.000
19	33.000	8.000
20	33.000	26.000
21	33.000	44.000
22	43.500	.000
23	43.500	9.000
24	43.500	44.000
25	54.000	.000
26	54.000	5.000
27	54.000	10.000
28	54.000	27.000
29	54.000	44.000
30	75.333	.000
31	75.333	9.500
32	96.667	.000
33	96.667	4.500
34	96.667	9.000
35	118.000	.000
36	118.000	8.500
37	139.333	.000
38	139.333	4.000
39	139.333	8.000
40	160.667	.000
41	160.667	7.500
42	182.000	.000
43	182.000	3.500
44	182.000	7.000

45	182.000	23.500
46	182.000	40.000
47	198.500	.000
48	198.500	7.000
49	198.500	40.000
50	215.000	.000
51	215.000	3.500
52	215.000	7.000
53	215.000	23.500
54	215.000	40.000
55	222.000	.000
56	222.000	7.000
57	222.000	40.000
58	229.000	.000
59	229.000	3.500
60	229.000	7.000
61	229.000	23.500
62	229.000	40.000
63	245.500	.000
64	245.500	7.000
65	245.500	40.000
66	262.000	.000
67	262.000	3.500
68	262.000	7.000
69	262.000	23.500
70	262.000	40.000

RESTRAINED NODES

NODE	CODE	FIXED VALUES		
1	001	.000000	.000000	.000000
6	001	.000000	.000000	.000000
9	001	.000000	.000000	.000000
14	001	.000000	.000000	.000000
17	001	.000000	.000000	.000000
22	001	.000000	.000000	.000000
25	001	.000000	.000000	.000000
30	001	.000000	.000000	.000000
32	001	.000000	.000000	.000000
35	001	.000000	.000000	.000000
37	001	.000000	.000000	.000000

40	001	.000000	.000000	.000000
42	001	.000000	.000000	.000000
47	001	.000000	.000000	.000000
50	001	.000000	.000000	.000000
55	001	.000000	.000000	.000000
58	001	.000000	.000000	.000000
63	001	.000000	.000000	.000000
66	001	.000000	.000000	.000000

NUMBER OF RIGID BODIES = 2
NUMBER OF NODES ON RIGID BODIES = 16

RIGID BODY NUMBER	REFERENCE NODE
1	1
2	50

MAXIMUM NUMBER OF NODES ON A RIGID BODY = 8

RIGID BODY	NODES
1	1 2 3 6 7 9 10 11
2	50 51 52 55 56 58 59 60

MATERIAL PROPERTIES				
NUMBER	YOUNGS MODULUS	POISSON'S RATIO	THICKNESS	UNIFORM LOAD
1	.300000E+04	.250000E+00	.240000E+02	.000000E+00
2	.300000E+04	.250000E+00	.240000E+02	.136000E+01
3	.300000E+04	.250000E+00	.240000E+02	.170000E+01

MAX FRONTWIDTH ENCOUNTERED = 30

*** VERTICAL COLUMN LOADS ONLY ***

TOTAL NODAL FORCES FOR EACH ELEMENT								
0	1	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
		.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
		.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00

.0000E+00 .0000E+00 .0000E+00 .0000E+00 .0000E+00 .0000E+00 .0000E+00 .0000E+00

REVISED NODAL CONNECTIONS FOR RIGID BODY

1	1	1	1	12	13	8	5	4
2	1	1	1	1	1	1	1	1
3	1	15	19	20	21	16	13	12
4	1	14	17	18	19	15	1	1
5	19	23	27	28	29	24	21	20
6	17	22	25	26	27	23	19	18
7	25	30	32	33	34	31	27	26
8	32	35	37	38	39	36	34	33
9	37	40	42	43	44	41	39	38
10	42	47	50	50	50	48	44	43
11	44	48	50	53	54	49	46	45
12	50	50	50	50	50	50	50	50
13	50	50	50	61	62	57	54	53
14	50	63	66	67	68	64	50	50
15	50	64	68	69	70	65	62	61

DISPLACEMENTS OF REFERENCE NODES AND FREE

NODES ONLY

1	.336898E+00	-.355909E-02	.000000E+00
2	.000000E+00	.000000E+00	.000000E+00
3	.000000E+00	.000000E+00	.000000E+00
4	.331997E+00	-.348904E-02	-.172927E-03
5	.326949E+00	-.342256E-02	-.206057E-03
6	.000000E+00	.000000E+00	.000000E+00
7	.000000E+00	.000000E+00	.000000E+00
8	.306334E+00	-.343346E-02	-.222299E-03
9	.000000E+00	.000000E+00	.000000E+00
10	.000000E+00	.000000E+00	.000000E+00
11	.000000E+00	.000000E+00	.000000E+00
12	.290072E+00	-.345413E-02	-.176627E-03
13	.285681E+00	-.343516E-02	-.220808E-03
14	.256309E+00	-.346135E-02	.000000E+00
15	.256089E+00	-.344895E-02	-.625912E-04
16	.249768E+00	-.340708E-02	-.200602E-03
17	.220059E+00	-.328048E-02	.000000E+00
18	.219945E+00	-.329238E-02	-.366664E-04
19	.219687E+00	-.329771E-02	-.704405E-04
20	.217801E+00	-.333616E-02	-.173372E-03
21	.214006E+00	-.337694E-02	-.213347E-03

22	.185937E+00	-.303874E-02	.000000E+00
23	.185633E+00	-.307249E-02	-.120066E-03
24	.179367E+00	-.332231E-02	-.205674E-03
25	.154919E+00	-.266573E-02	.000000E+00
26	.154624E+00	-.272242E-02	-.432663E-04
27	.153975E+00	-.283958E-02	-.861015E-04
28	.148760E+00	-.326520E-02	-.169355E-03
29	.144164E+00	-.331340E-02	-.208120E-03
30	.107907E+00	-.170076E-02	.000000E+00
31	.107399E+00	-.169640E-02	-.111380E-03
32	.836369E-01	-.639302E-03	.000000E+00
33	.834926E-01	-.658911E-03	-.527238E-04
34	.831228E-01	-.670928E-03	-.106313E-03
35	.808835E-01	.294264E-03	.000000E+00
36	.804972E-01	.284345E-03	-.891379E-04
37	.975974E-01	.113426E-02	.000000E+00
38	.975391E-01	.113508E-02	-.359526E-04
39	.973266E-01	.112518E-02	-.717422E-04
40	.130623E+00	.176351E-02	.000000E+00
41	.130424E+00	.175204E-02	-.447138E-04
42	.175044E+00	.214622E-02	.000000E+00
43	.175105E+00	.215516E-02	-.564434E-05
44	.175152E+00	.216650E-02	-.117089E-04
45	.175191E+00	.217601E-02	-.351970E-04
46	.174222E+00	.218555E-02	-.697906E-04
47	.212518E+00	.219456E-02	.000000E+00
48	.212387E+00	.218294E-02	-.233646E-04
49	.210246E+00	.214835E-02	-.551412E-04
50	.249987E+00	.208374E-02	.000000E+00
51	.000000E+00	.000000E+00	.000000E+00
52	.000000E+00	.000000E+00	.000000E+00
53	.247341E+00	.211371E-02	-.788130E-04
54	.245470E+00	.210859E-02	-.718140E-04
55	.000000E+00	.000000E+00	.000000E+00
56	.000000E+00	.000000E+00	.000000E+00
57	.260194E+00	.207843E-02	-.800766E-04
58	.000000E+00	.000000E+00	.000000E+00
59	.000000E+00	.000000E+00	.000000E+00
60	.000000E+00	.000000E+00	.000000E+00
61	.276439E+00	.204069E-02	-.843768E-04
62	.274405E+00	.204909E-02	-.809728E-04

63	.310706E+00	.197250E-02	.000000E+00
64	.310522E+00	.198111E-02	-.314681E-04
65	.307584E+00	.200296E-02	-.747995E-04
66	.342535E+00	.197831E-02	.000000E+00
67	.342544E+00	.197351E-02	-.130420E-04
68	.342437E+00	.197096E-02	-.246402E-04
69	.341576E+00	.198627E-02	-.615966E-04
70	.340007E+00	.196859E-02	-.924109E-04

DISPLACEMENTS

0
0

NODE	DISP.	XZ-ROT.	YZ-ROT.
1	.336898E+00	-.355909E-02	.000000E+00
2	.336898E+00	-.355909E-02	.000000E+00
3	.336898E+00	-.355909E-02	.000000E+00
4	.331997E+00	-.348904E-02	-.172927E-03
5	.326949E+00	-.342256E-02	-.206057E-03
6	.315543E+00	-.355909E-02	.000000E+00
7	.315543E+00	-.355909E-02	.000000E+00
8	.306334E+00	-.343346E-02	-.222299E-03
9	.294189E+00	-.355909E-02	.000000E+00
10	.294189E+00	-.355909E-02	.000000E+00
11	.294189E+00	-.355909E-02	.000000E+00
12	.290072E+00	-.345413E-02	-.176627E-03
13	.285681E+00	-.343516E-02	-.220808E-03
14	.256309E+00	-.346135E-02	.000000E+00
15	.256089E+00	-.344895E-02	-.625912E-04
16	.249768E+00	-.340708E-02	-.200602E-03
17	.220059E+00	-.328048E-02	.000000E+00
18	.219945E+00	-.329238E-02	-.366664E-04
19	.219687E+00	-.329771E-02	-.704405E-04
20	.217801E+00	-.333616E-02	-.173372E-03
21	.214006E+00	-.337694E-02	-.213347E-03
22	.185937E+00	-.303874E-02	.000000E+00
23	.185633E+00	-.307249E-02	-.120066E-03
24	.179367E+00	-.332231E-02	-.205674E-03
25	.154919E+00	-.266573E-02	.000000E+00
26	.154624E+00	-.272242E-02	-.432663E-04
27	.153975E+00	-.283958E-02	-.861015E-04
28	.148760E+00	-.326520E-02	-.169355E-03
29	.144164E+00	-.331340E-02	-.208120E-03
30	.107907E+00	-.170076E-02	.000000E+00
31	.107399E+00	-.169640E-02	-.111380E-03

32	.836369E-01	-.639302E-03	.000000E+00
33	.834926E-01	-.658911E-03	-.527238E-04
34	.831228E-01	-.670928E-03	-.106313E-03
35	.808835E-01	.294264E-03	.000000E+00
36	.804972E-01	.284345E-03	-.891379E-04
37	.975974E-01	.113426E-02	.000000E+00
38	.975391E-01	.113508E-02	-.359526E-04
39	.973266E-01	.112518E-02	-.717422E-04
40	.130623E+00	.176351E-02	.000000E+00
41	.130424E+00	.175204E-02	-.447138E-04
42	.175044E+00	.214622E-02	.000000E+00
43	.175105E+00	.215516E-02	-.564434E-05
44	.175152E+00	.216650E-02	-.117089E-04
45	.175191E+00	.217601E-02	-.351970E-04
46	.174222E+00	.218555E-02	-.697906E-04
47	.212518E+00	.219456E-02	.000000E+00
48	.212387E+00	.218294E-02	-.238646E-04
49	.210246E+00	.214835E-02	-.551412E-04
50	.249987E+00	.208374E-02	.000000E+00
51	.249987E+00	.208374E-02	.000000E+00
52	.249987E+00	.208374E-02	.000000E+00
53	.247341E+00	.211371E-02	-.788130E-04
54	.245470E+00	.210859E-02	-.718140E-04
55	.264574E+00	.208374E-02	.000000E+00
56	.264574E+00	.208374E-02	.000000E+00
57	.260194E+00	.207843E-02	-.800766E-04
58	.279160E+00	.208374E-02	.000000E+00
59	.279160E+00	.208374E-02	.000000E+00
60	.279160E+00	.208374E-02	.000000E+00
61	.276439E+00	.204069E-02	-.843768E-04
62	.274405E+00	.204909E-02	-.809728E-04
63	.310706E+00	.197250E-02	.000000E+00
64	.310522E+00	.198111E-02	-.314681E-04
65	.307584E+00	.200296E-02	-.747995E-04
66	.342535E+00	.197831E-02	.000000E+00
67	.342544E+00	.197351E-02	-.130420E-04
68	.342437E+00	.197096E-02	-.246402E-04
69	.341576E+00	.198627E-02	-.615966E-04
70	.340007E+00	.196859E-02	-.924109E-04

0	REACTIONS			
0	NODE	FORCE	XZ-MOMENT	YZ-MOMENT

1	.000000E+00	.000000E+00	.129152E+04
6	.000000E+00	.000000E+00	.000000E+00
9	.000000E+00	.000000E+00	.000000E+00
14	.000000E+00	.000000E+00	.254225E+03
17	.000000E+00	.000000E+00	.154938E+03
22	.000000E+00	.000000E+00	.294727E+03
25	.000000E+00	.000000E+00	-.253857E+02
30	.000000E+00	.000000E+00	-.117055E+03
32	.000000E+00	.000000E+00	.421934E+02
35	.000000E+00	.000000E+00	-.147383E+02
37	.000000E+00	.000000E+00	.128076E+02
40	.000000E+00	.000000E+00	-.228766E+02
42	.000000E+00	.000000E+00	-.201345E+02
47	.000000E+00	.000000E+00	.294021E+03
50	.000000E+00	.000000E+00	.165896E+04
55	.000000E+00	.000000E+00	.000000E+00
58	.000000E+00	.000000E+00	.000000E+00
63	.000000E+00	.000000E+00	.400564E+03
66	.000000E+00	.000000E+00	.797737E+02

STRESSES

OG,P.	X-COORD.	Y-COORD.	X-MOMENT	Y-MOMENT	XY-MOMENT	XZ-S, FORCE	YZ-S, FORCE
ELEMENT NO.= 1							
1	1.3524	10.2827	.62110E+01	.40347E+02	-.51942E+01	-.33219E-01	-.47801E+01
2	1.3524	25.0000	-.34095E+01	.18362E+02	-.28394E+01	-.12431E+00	-.19732E+01
3	1.3524	39.7173	.21604E+01	.17409E+00	.23393E-01	-.18320E+00	-.11679E+01
4	6.0000	10.2827	.58781E+01	.40866E+02	-.70717E+01	-.17905E+00	-.43554E+01
5	6.0000	25.0000	-.53331E+01	.18884E+02	-.41439E+01	-.47615E+00	-.15149E+01
6	6.0000	39.7173	-.13536E+01	.70017E+00	-.70806E+00	-.28145E+00	-.63476E+00
7	10.6476	10.2827	.52861E+01	.40353E+02	-.91504E+01	-.34080E+00	-.39746E+01
8	10.6476	25.0000	-.75158E+01	.18374E+02	-.56495E+01	-.89531E+00	-.11992E+01
9	10.6476	39.7173	-.51269E+01	.19382E+00	-.16406E+01	-.49841E+00	-.34324E+00
ELEMENT NO.= 2							
1	1.3524	.6762	-.55251E-03	-.15860E-03	-.43004E-03	-.65144E-03	-.10288E-02
2	1.3524	3.0000	-.77161E-03	-.17968E-03	-.16363E-03	-.70605E-03	-.28911E-03
3	1.3524	5.3238	-.50574E-03	-.11274E-03	-.31460E-03	-.54081E-03	-.91616E-03
4	6.0000	.6762	-.62197E-04	-.15210E-04	-.15783E-03	-.20604E-03	-.68452E-03
5	6.0000	3.0000	-.96828E-04	-.19968E-05	.59903E-05	-.15609E-03	-.39048E-04

6	6.0000	5.3238	-.59485E-04	-.41066E-04	-.13921E-03	-.18506E-03	-.69399E-03
7	10.6476	.6762	-.57557E-03	-.14928E-03	-.38542E-03	-.70065E-03	-.75948E-03
8	10.6476	3.0000	-.62716E-03	-.14916E-03	-.76910E-04	-.61621E-03	-.24156E-03
9	10.6476	5.3238	-.56743E-03	-.11291E-03	-.32801E-03	-.56893E-03	-.82106E-03

ELEMENT NO.= 3

1	14.3667	10.4827	-.17615E+02	.31052E+02	-.18035E+01	-.24727E+01	-.29778E+01
2	14.3667	25.1127	-.95887E+01	.15397E+02	-.34253E+01	-.13093E+01	-.72369E+00
3	14.3667	39.7427	-.95195E+01	-.22484E+01	-.28240E+01	-.19791E+00	-.26457E+00
4	22.5000	11.1700	-.30792E+02	.19331E+02	-.18898E+01	-.17032E+01	-.65855E+00
5	22.5000	25.5000	-.17073E+02	.86227E+01	-.20242E+01	-.65752E+00	.23085E+00
6	22.5000	39.8300	-.11663E+02	-.41624E+01	.16889E+00	-.17792E+00	-.28696E+00
7	30.6333	11.8573	-.42968E+02	.12518E+02	-.10603E+01	-.14164E+01	.46563E+00
8	30.6333	25.8873	-.23402E+02	.69475E+01	.28273E+00	-.28948E+00	.45192E+00
9	30.6333	39.9173	-.12510E+02	-.79185E+00	.40615E+01	-.24487E+00	-.58139E+00

ELEMENT NO.= 4

1	14.3667	.7016	-.24111E+02	.47377E+01	-.37402E-01	-.25830E+01	-.64827E-01
2	14.3667	3.1127	-.26947E+02	.37484E+01	.35696E+01	-.27903E+01	.16356E-01
3	14.3667	5.5238	-.30425E+02	.25985E+01	.70701E+01	-.31004E+01	.91348E-01
4	22.5000	.7889	-.39759E+02	.22068E+02	-.10203E+01	-.16758E+01	-.19305E+00
5	22.5000	3.5000	-.38267E+02	.21335E+02	-.62426E+00	-.18503E+01	.14841E-01
6	22.5000	6.2111	-.37270E+02	.20479E+02	-.31033E+00	-.21430E+01	.20088E+00
7	30.6333	.8762	-.60561E+02	.19278E+02	.26139E+01	-.19000E+01	-.23417E+00
8	30.6333	3.8873	-.55162E+02	.18860E+02	.22353E+00	-.18164E+01	-.14732E+00
9	30.6333	6.8984	-.50139E+02	.18348E+02	-.22294E+01	-.18707E+01	-.98339E-01

ELEMENT NO.= 5

1	35.3667	12.2573	-.54787E+02	.71813E+01	.14164E+02	-.12469E+01	.71710E+00
2	35.3667	26.1127	-.15043E+02	.80068E+01	.76250E+01	-.56402E+00	.59979E+00
3	35.3667	39.9681	-.15773E+02	-.12863E+01	.15460E+01	.11452E+01	-.39453E+00
4	43.5000	12.9446	-.59646E+02	.17595E+01	.21412E+02	-.58904E+00	-.24025E+00
5	43.5000	26.5000	-.11455E+02	.55893E+01	.94414E+01	.28858E+00	.25708E+00
6	43.5000	40.0554	-.55200E+01	-.11449E+01	-.20421E+01	.14856E+00	-.19370E-01
7	51.6333	13.6319	-.63483E+02	.20805E+01	.30427E+02	.15539E+00	-.35662E+01
8	51.6333	26.8873	-.65908E+01	.90186E+01	.12696E+02	.14750E+01	-.18860E+01
9	51.6333	40.1427	.61847E+01	.49274E+01	-.45203E+01	-.26359E+00	-.87667E+00

ELEMENT NO.= 6

1	35.3667	.9270	-.64885E+02	.22843E+02	.23299E+01	-.21030E+01	.15403E+00
2	35.3667	4.1127	-.71411E+02	.19510E+02	.56937E+01	-.16727E+01	.23888E+00

3	35.3667	7.2984	-.70924E+02	.17929E+02	.92044E+01	-.13151E+01	.28522E+00
4	43.5000	1.0143	-.94576E+02	.23450E+02	-.13454E+01	-.15822E+01	.54991E+00
5	43.5000	4.5000	-.88416E+02	.24001E+02	.47415E+01	-.16036E+01	.64929E+00
6	43.5000	7.9857	-.76456E+02	.26003E+02	.10954E+02	-.13042E+01	.72390E+00
7	51.6333	1.1016	-.12965E+03	.37058E+01	.59695E+01	-.27398E+01	-.28240E+00
8	51.6333	4.8873	-.11178E+03	.77843E+01	.14086E+02	-.24856E+01	-.59218E+00
9	51.6333	8.6730	-.89136E+02	.13058E+02	.22312E+02	-.15199E+01	-.91285E+00

ELEMENT NO. =		7						
1	58.8086	1.1143	-.15673E+03	-.60152E+01	.49108E+01	-.84186E+00	-.42959E+00	
2	58.8086	4.9436	-.17040E+03	-.95868E+01	.17298E+02	.47296E-01	-.82350E+00	
3	58.8086	8.7730	-.18705E+03	-.13902E+02	.29709E+02	.17167E+01	-.12223E+01	
4	75.3335	1.0707	-.16457E+03	-.74488E+00	-.64867E+01	.59597E+00	.14942E+00	
5	75.3335	4.7500	-.16750E+03	-.13564E+01	-.13790E+00	.42271E+00	.50781E-01	
6	75.3335	8.4293	-.17368E+03	-.27778E+01	.62363E+01	.65268E+00	-.45355E-01	
7	91.8584	1.0270	-.17395E+03	-.21520E+01	.64921E+00	.99028E+00	.24599E-01	
8	91.8584	4.5564	-.16571E+03	.33039E+00	.47127E+00	.94337E+00	.64127E-01	
9	91.8584	8.0857	-.16100E+03	.19308E+01	.32105E+00	.92288E+00	.11362E+00	

ELEMENT NO. =		8						
1	101.4755	1.0016	-.15607E+03	.37447E+00	.51301E+01	.10868E+01	-.12460E+00	
2	101.4755	4.4436	-.15818E+03	.29953E+00	.35237E+01	.13889E+01	-.87198E-01	
3	101.4755	7.8857	-.15940E+03	.44556E+00	.18999E+01	.16190E+01	-.39460E-01	
4	118.0000	.9580	-.14416E+03	-.18789E-01	.11390E+01	.81661E+00	-.21464E-01	
5	118.0000	4.2500	-.14529E+03	-.79526E-01	.12397E+01	.92644E+00	-.25554E-01	
6	118.0000	7.5420	-.14545E+03	.10161E+00	.13216E+01	.10677E+01	-.24928E-01	
7	134.5245	.9143	-.13234E+03	-.81250E+00	-.10995E+01	.19276E+01	.73022E-01	
8	134.5245	4.0564	-.13244E+03	-.87058E+00	.87175E+00	.19616E+01	.33623E-01	
9	134.5245	7.1984	-.13149E+03	-.66392E+00	.28223E+01	.21306E+01	-.64012E-02	

ELEMENT NO. =		9						
1	144.1416	.8889	-.11223E+03	.10035E+01	-.45659E+00	.24655E+01	-.59504E-02	
2	144.1416	3.9436	-.11105E+03	.12324E+01	.15123E+01	.24863E+01	-.42145E-01	
3	144.1416	6.9984	-.11043E+03	.13218E+01	.34916E+01	.26755E+01	-.79672E-01	
4	160.6665	.8453	-.82015E+02	.84826E-02	.72597E+00	.12579E+01	-.10078E+00	
5	160.6665	3.7500	-.82574E+02	-.39403E-01	.12281E+01	.13864E+01	-.10233E+00	
6	160.6665	6.6547	-.83755E+02	-.24284E+00	.17419E+01	.16145E+01	-.10218E+00	
7	177.1913	.8016	-.52542E+02	-.38448E+01	-.19408E+01	.35464E+01	.26674E+00	
8	177.1914	3.5564	-.54933E+02	-.41756E+01	-.30588E+01	.35968E+01	.29173E+00	
9	177.1914	6.3111	-.58016E+02	-.46797E+01	-.41636E+01	.36786E+01	.32213E+00	

8	227.4222	23.5000	.21332E+02	.13859E+02	.11633E+01	.30943E-01	-.13362E+01
9	227.4222	36.2808	.16647E+02	-.15179E+01	-.16869E+01	-.73846E+00	-.42544E+00

ELEMENT NO.= 14

1	232.7191	.7889	.33103E+02	.13553E+02	-.32009E+00	-.34246E+01	-.12429E+00
2	232.7191	3.5000	.31817E+02	.17107E+02	.98225E+00	-.35809E+01	-.95971E-01
3	232.7191	6.2111	.30380E+02	.12673E+02	.22491E+01	-.37914E+01	-.69251E-01
4	245.5000	.7889	.16210E+02	.20711E+02	-.12264E+01	-.12578E+01	-.23508E+00
5	245.5000	3.5000	.16456E+02	.19678E+02	-.11532E+01	-.13322E+01	-.24503E+00
6	245.5000	6.2111	.16552E+02	.19128E+02	-.11161E+01	-.14734E+01	-.26582E+00
7	258.2808	.7889	-.37087E+01	.14622E+02	.77898E+00	-.73664E+00	.14745E+00
8	258.2808	3.5000	-.19295E+01	.14093E+02	-.37738E+00	-.59205E+00	-.11522E+00
9	258.2808	6.2111	-.30106E+00	.13521E+02	-.15699E+01	-.52677E+00	-.39563E+00

ELEMENT NO.= 15

1	232.7191	10.7192	.30008E+02	.27275E+02	.56610E+01	-.30621E+01	-.28641E+01
2	232.7191	23.5000	.16971E+02	.11267E+02	.13924E+00	-.10926E+01	-.10788E+01
3	232.7191	36.2808	.95939E+01	-.33267E+01	-.33304E+01	-.33466E+00	-.42508E+00
4	245.5000	10.7192	.13548E+02	.15544E+02	-.30811E-01	-.11485E+01	-.10272E+01
5	245.5000	23.5000	.72896E+01	.63604E+01	-.18699E+01	-.32647E+00	-.29809E+00
6	245.5000	36.2808	.66909E+01	-.14085E+01	-.16569E+01	-.24528E+00	-.24537E+00
7	258.2808	10.7192	-.18722E+01	.79742E+01	-.36935E+01	-.43569E+00	-.21163E+00
8	258.2808	23.5000	-.13517E+01	.56150E+01	-.18500E+01	-.31093E+00	-.19264E+00
9	258.2808	36.2808	.48283E+01	.46708E+01	.20456E+01	-.45599E+00	-.39455E+00

WINKLER FOUNDATION NODE POINT FORCES

NODE NORMAL

ELEMENT NUMBER : 1

3	.5896E-01
7	.5522E-01
11	.5148E-01
12	.5076E-01
13	.4999E-01
8	.5361E-01
5	.5722E-01
4	.5810E-01

ELEMENT NUMBER : 2

1	.5896E-01
---	-----------

	6	.5522E-01
	9	.5148E-01
	10	.5148E-01
	11	.5148E-01
	7	.5522E-01
	3	.5896E-01
	2	.5896E-01
ELEMENT NUMBER :	3	
	11	.5148E-01
	15	.4482E-01
	19	.3845E-01
	20	.3812E-01
	21	.3745E-01
	16	.4371E-01
	13	.4999E-01
	12	.5076E-01
ELEMENT NUMBER :	4	
	9	.5148E-01
	14	.4485E-01
	17	.3851E-01
	18	.3849E-01
	19	.3845E-01
	15	.4482E-01
	11	.5148E-01
	10	.5148E-01
ELEMENT NUMBER :	5	
	19	.3845E-01
	23	.3249E-01
	27	.2695E-01
	28	.2603E-01
	29	.2523E-01
	24	.3139E-01
	21	.3745E-01
	20	.3812E-01
ELEMENT NUMBER :	6	
	17	.3851E-01
	22	.3254E-01

51 .4375E-01
52 .4375E-01
48 .3717E-01
44 .3065E-01
43 .3064E-01

ELEMENT NUMBER : 11

44 .3065E-01
48 .3717E-01
52 .4375E-01
53 .4328E-01
54 .4296E-01
49 .3679E-01
46 .3049E-01
45 .3066E-01

ELEMENT NUMBER : 12

50 .4375E-01
55 .4630E-01
58 .4885E-01
59 .4885E-01
60 .4885E-01
56 .4630E-01
52 .4375E-01
51 .4375E-01

ELEMENT NUMBER : 13

52 .4375E-01
56 .4630E-01
60 .4885E-01
61 .4838E-01
62 .4802E-01
57 .4553E-01
54 .4296E-01
53 .4328E-01

ELEMENT NUMBER : 14

58 .4885E-01
63 .5437E-01
66 .5994E-01
67 .5995E-01

	25	.2711E-01
	26	.2706E-01
	27	.2695E-01
	23	.3249E-01
	19	.3845E-01
	18	.3849E-01
ELEMENT NUMBER :	7	
	25	.2711E-01
	30	.1888E-01
	32	.1464E-01
	33	.1461E-01
	34	.1455E-01
	31	.1879E-01
	27	.2695E-01
	26	.2706E-01
ELEMENT NUMBER :	8	
	32	.1464E-01
	35	.1415E-01
	37	.1708E-01
	38	.1707E-01
	39	.1703E-01
	36	.1409E-01
	34	.1455E-01
	33	.1461E-01
ELEMENT NUMBER :	9	
	37	.1708E-01
	40	.2286E-01
	42	.3063E-01
	43	.3064E-01
	44	.3065E-01
	41	.2282E-01
	39	.1703E-01
	38	.1707E-01
ELEMENT NUMBER :	10	
	42	.3063E-01
	47	.3719E-01
	50	.4375E-01

ELEMENT NUMBER : 15

68	.5993E-01
64	.5434E-01
60	.4885E-01
59	.4885E-01

60	.4885E-01
64	.5434E-01
68	.5993E-01
69	.5978E-01
70	.5950E-01
65	.5383E-01
62	.4802E-01
61	.4838E-01

N378
H.538
cap 2