A Multilevel Superelement Substructuring for Boxlike Caisson Analysis

By

Saeed Aroni Hesari

A thesis
submitted in partial fulfillment of the
requirements for the degree of
Master of Applied Science in Civil Engineering

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF OTTAWA
OTTAWA, ONTARIO, APRIL 1994
Ottawa-Carleton Civil Engineering Institute

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-00470-8

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

A multilevel superelement substructuring approach for linear-static analysis of rectangular boxlike caissons is developed. The structure is considered as an assemblage of superelements in three substructural levels. These are panel, layer, and caisson superelements. Each superelement substructure is derived by applying the principle of static condensation.

Rectangular 9-node flat shell elements, which are composed of membrane stress and Mindlin plate bending elements, are assembled together to make the stiffness and load matrices of each panel substructure. Every panel substructure is condensed to a panel superelement. Then all of them in a layer are assembled together and condensed to layer superelement. Finally, the layer superelement(s) are assembled and condensed to yield to caisson superelement.

After assembling process, equilibrium equations are solved for the caisson superelement. Then a multilevel recovery process, with an inverse sequence of assembling one, gives the internal degrees of freedom for caisson, layer, and panel superelements, respectively. Finding all displacements, one can finally compute internal forces for every individual flat shell element.

This special purpose superelement substructure method provides considerable efficiency in practical analysis and design of boxlike caissons. Based on this approach, a computer program is developed which takes the most advantage from the repetition of similar individual elements and panels. Several numerical examples are solved indicating good agreement with other analytical or numerical methods and proper convergency for design purposes.

Acknowledgements

The author is grateful to Dr. M. S. Cheung for his supervision and guidance throughout this study. The author is also thankful to Dr. N. J. Gardner and Dr. M. Bruneau for providing access to SAP90 program.

Moreover, the financial assistance provided by the government of Iran to pursue this study is highly appreciated. I also would like to express my sincere appreciation to my family for their continued support and encouragement.

Contents

List of Figures and Diagrams

Figures

Diagrams

List of Tables

Notations

a	size of square plate
A	the area of element
A_f	foundation area
B	foundation width
$\underline{B}$	strain-displacement relation matrix
$\underline{B}_b$	bending strain-displacement relation matrix
$\underline{B}_s$	shear " " " "
$\underline{B}_m$	membrane " " " "
dA	element of area
$\underline{d}$	global displacement vector
$\underline{d}'$	local " "
$\underline{d}_m$	membrane (in-plane) displacement vector
$\underline{d}_p$	plate bending (normal) " "
D	flexural rigidity of plate
$\underline{D}$	constitutive matrix
$\underline{D}_b$	bending constitutive matrix
$\underline{D}_s$	shear " "
$\underline{D}_m$	membrane " "
E	Young's modulus
$\underline{F}$	global force vector
$\underline{F}'$	local " "
$\underline{F}_f$	force vector of flat shell element
G	shear modulus
h	height
I	second moment of area

$\underline{\mathbf{I}}$ unit matrix

$\underline{\mathbf{J}}$ Jacobian matrix

$|\mathbf{J}|$ determinant of Jacobian matrix

$\mathbf{k}$ shear correction factor

$\mathbf{k}$ subgrade modulus

$\mathbf{k_x}$ " " in x direction

$\mathbf{k_y}$ " " " y "

$\mathbf{k_z}$ " " " z "

$\underline{\mathbf{k}}_{es}$ stiffness matrix of a foundation element

$\underline{\mathbf{K}}_{es}$ " " " the entire foundation model

$\mathbf{K_x}$ lumped elastic stiffness in x direction

$\mathbf{K_y}$ " " " " y "

$\mathbf{K_z}$ " " " " z "

$\underline{\mathbf{K}}$ stiffness matrix

$\underline{\mathbf{K}}_b$ bending contribution of stiffness matrix

$\underline{\mathbf{K}}_s$ shear " " " "

$\underline{\mathbf{K}}_m$ membrane " " " "

$\underline{\mathbf{K}}_p$ stiffness matrix of the Mindlin plate bending element

$\underline{\mathbf{K}}_f$ " " " " " flat shell "

$\mathbf{L}$ length (of beam, plate or caisson)

$\mathbf{m_{ob}}$ bending moment per unit width at center

$\mathbf{m_{eub}}$ " " " " " " " of the edge

$\mathbf{m_{ct}}$ torsional " " " " " corner

$\mathbf{M_{xx}}$ moment per unit witdh of plate about y in yz plane

$\mathbf{M_{yy}}$ " " " " " " " x " xz "

$\mathbf{M_{xy}}$ " " " " " " " x " yz "

$\underline{\mathbf{M}}$ vector of moments per unit width

$\mathbf{n}$ numerical integration order

N	shape function
p	pressure
P_A	active lateral force (per unit width)
$\underline{P}$	load vector (of a substructure)
$\underline{P}^*$	condensed $\underline{P}$ " " "
q	vertical surcharge
q	distributed lateral load
r_o	equivalent radius of a rectangular foundation
S_x	shear rigidity associated with x direction
S_y	" " " " y "
$\underline{S}$	stiffness matrix of a substructure
$\underline{S}^*$	condensed $\underline{S}$
t	thickness (of plate or beam)
$\underline{T}$	transformation matrix
u	in-plane displacement in local x direction
v	" " " " y "
v_{eo}	shear force per unit width at center of the edge
V	volume of an element
V_{xz}	shear force per unit width in yz plane in z direction
V_{yz}	" " " " " " xz " " z "
$\underline{V}$	vector of shear forces per unit width
w	normal displacement in local z direction
w_o	normal displacement at center of plate
w_t	tip deflection (of cantilever beam)
W	width of beam, plate or caisson
Z	vertical distance
α	wall inclination from horizontal
β	slope of backfill surface

γ specific gravity

γ_{ds} " " of dry soil

γ_{ws} " " " wet "

γ_w " " " water

δ wall friction angle with soil

$\underline{\delta}$ displacement vector of a substructure

ϵ strain

$\underline{\epsilon}_b$ bending strain vector

$\underline{\epsilon}_s$ shear " "

$\underline{\epsilon}_m$ membrane " "

η natural coordinate associated with local y

θ_x normal rotation about x

θ_y normal rotation about y

θ_z in-plane rotation about z (drilling freedom)

κ curvature

$\underline{\kappa}_b$ bending curvature vector

$\underline{\kappa}_s$ shear " "

$\underline{\lambda}$ directional cosines matrix

$\underline{\lambda}_{bp}$ $\underline{\lambda}$ for base panels

$\underline{\lambda}_{lp}$ $\underline{\lambda}$ for longitudinal vertical panels

$\underline{\lambda}_{tp}$ $\underline{\lambda}$ for transverse " "

ν Poisson's ratio

ξ natural coordinate associated with local x

σ stress

$\underline{\sigma}_b$ bending stress vector

$\underline{\sigma}_s$ shear " "

$\underline{\sigma}_m$ membrane " "

ϕ soil internal friction angle

Chapter 1

INTRODUCTION

1.1 General

The name "caisson" is derived from a French word which means a chest or box. In civil engineering, it refers to a boxlike structure, round or rectangular, which is sunk from the surface of the land or in water to a desired depth [1]. Based on construction method and utilization, three main types of caissons are; open; pneumatic; and floating caissons.

The first two types are mainly used for deep foundations. They are enclosing shells which finally become an integral part of the foundation. The third type is usually used in relatively shallow water and is suitable for sites where the bottom can be levelled and has adequate bearing capacity.

This thesis is concerned with the analysis of rectangular floating caissons, which are mainly used as marine structures like jetties, quay walls, and breakwaters. These structures can be analyzed by an efficient finite element method. Generally, an efficient computer program utilizes a good computing strategy and effective numerical techniques.

Based on the stated requirements, a superelement substructure has been developed to analyze boxlike caisson structures. A computer program is described which models the caisson structure using superelements made up of rectangular Mindlin flat shell elements. The computer program uses a direct stiffness method and is currently limited to isotropic materials in a linear static analysis.

1.2 General Philosophy of Substructuring

In substructuring, a structure is divided into two or more parts. Each part, which is called a substructure, is regarded as a "superelement", that is a single element with many nodes on its boundary and many interior degrees of freedom. The name "macroelement" is appropriate too.

"Multilevel substructuring" is also possible in which a main substructure itself is divided into secondary substructures. The substructuring method is a finite element process which its elements are substructures with many internal degrees of freedom. In this method, in contrast to the conventional finite element, a single stiffness matrix operating on all degrees of freedom of structure is not formed. Instead, there are several stiffness matrices, one for each substructure, which are finally assembled to make the stiffness matrix of entire structure.

Boxlike caisson structures can be regarded as an assemblage of layer superelement substructures which are connected at their exterior common boundaries. Fig. 1.1 shows layer superelement substructures I, II and III for a six compartment caisson structure. The first layer substructure includes 6 base panel and 17 vertical panel (9 longitudinal and 8 transverse) superelements. Whereas the second and third layer substructures only include 17 vertical panel superelements.

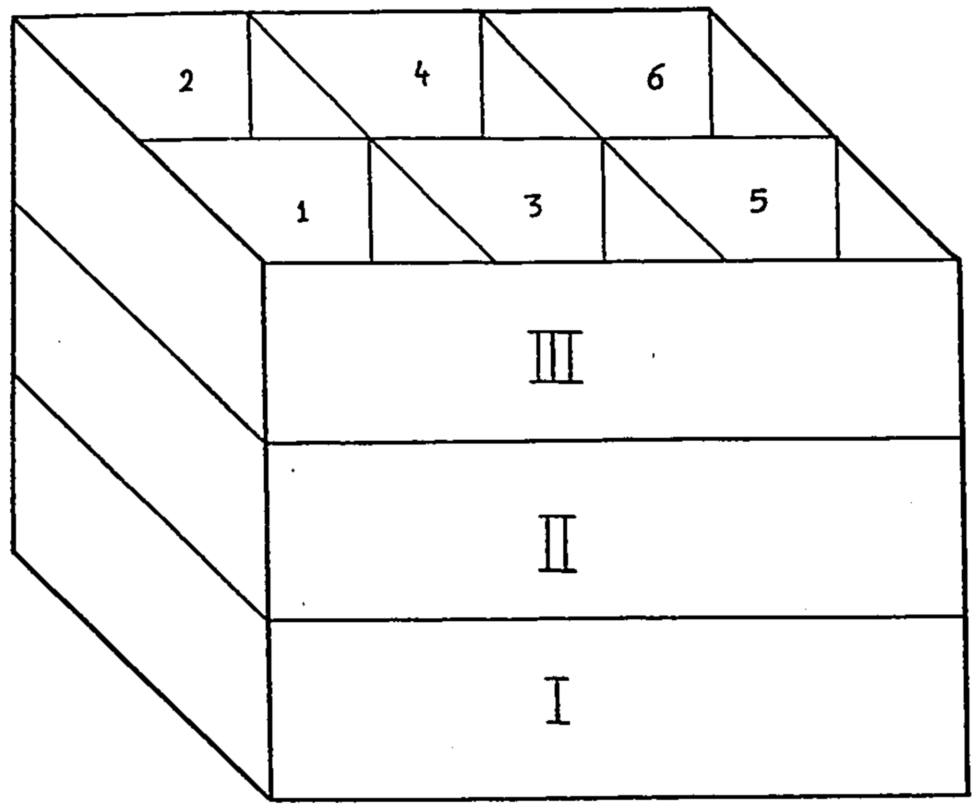


Fig. 1.1 A general view of a three-layer six-compartment caisson

Each panel substructure itself can be regarded as an assemblage of flat shell elements. After assembling, all degrees of freedom related to its internal nodes are condensed by applying the standard process of static condensation. This leads to the panel superelement to be formed. Since all elements of a panel substructure are similar, the stiffness matrix of individual elements is calculated once. Also, the stiffness matrix calculation is not repeated for a panel substructure which is similar to the previously assembled panel.

Again, after assembling the panel superelements into a layer substructure, the process of static condensation yields to the layer superelement. Then the layer superelements are assembled together to make entire caisson structure which is finally condensed to the caisson superelement.

Unlike the conventional finite element, the multilevel superelement substructuring method divides the process of solving equilibrium equations into several sequential stages. For the present study, these stages include; the caisson superelement; the layer superelement; and the panel superelement substructural levels. Therefore, it takes the most advantage from the repetition of similar individual elements and panel superelements, and also reduces the required storage capacity of the computer program.

1.3 Review of Previous Studies

The current thesis is concerned with the structural analysis of caissons by using the superelement substructures which are made up of Mindlin flat shell elements. Therefore, the review of Mindlin plate bending elements and superelement substructuring as well as the development of caisson structure itself are briefly presented.

1.3.1 Mindlin Plate Bending Review

Mindlin [2] developed a plate bending theory which accounted for both bending and shear deformations for the first time in 1951. Based on this theory, a variety of shell and plate elements, assuming independent rotation and displacement interpolation, have been proposed since the early days of the finite element method when the difficulties of ensuring full C-1 continuous interpolation were discovered.

Whilst such elements are capable of dealing satisfactorily with thick shell and plate components, their performance degenerates rapidly as the shell and plate becomes thin [3]. Many researchers tried to overcome this difficulty known as "shear lock" phenomena. In the following paragraphs, a brief review of quadrilateral displacement-based finite elements is presented.

a) Reduced Integration

In 1971, Zienkiewictz and co-workers [4] showed that reducing the order of numerical integration improves the performance of Mindlin elements for a thin plate or shell. At the same time, Pawsey and Clough [5] achieved similar results. Many authors investigated the use of reduced integration in developing different Mindlin elements.

In 1978, Pugh, Hinton, and Zienkiewicz [3] performed a series of numerical experiments on five different plate elements. Demonstrating the use of reduced integration, they indicated that the main reason for its success is the relaxation of a constraint on the shear strain. They also showed that the performance of a nine-node Lagrangian element is near optimal.

b) Selective Integration

Many researchers [3,5,6] realized that the reduced integration technique may lead to undesirable zero or low energy modes for a highly distorted element. Whereas a selective integration eliminates (for Serendipity elements) or decreases (for Lagrangian elements) the number of zero energy modes and improves the results. This technique uses reduced integration for transverse shear and full integration for the bending contribution to the stiffness matrix.

Even the selective integration does not remove the entire difficulty in Lagrangian family of elements. In 1978, Hughes and Cohen [7] developed the "heterosis element" which neither locks nor possesses unwanted zero energy modes in plate bending. The heterosis element is formulated using nine node Lagrangian shape functions for rotations and eight node Serendipity shape functions for lateral displacements.

c) **Kirchhoff Mode Criteria**

In 1981, Hughes and Tezduyar [8] introduced displacement elements based on Kirchhoff mode criteria. The interpolation function of these elements are so chosen that the shear angles obey the Kirchhoff constraints throughout the element domain or at certain key positions [9]. Therefore, different interpolation patterns have to be used for the displacement and rotations.

These elements are not convenient for practical application, and special procedures have to be adopted to retain the simplicity of the resulting elements. Based on these considerations, some highly effective quadrilateral elements have been developed [8,10].

d) **Substitute Shear Strain Fields**

In 1985, Bathe and Dvorkin [11], used substitute shear strain fields in developing the bilinear (4-node) Mindlin shell element. Huang and Hinton [12] developed the quadratic (8 and 9 node) Mindlin plate elements based on the same concept. In the development of these elements, it was observed that the polynomial terms for the rotations and the derivatives of the lateral displacement do not match. Whils for hin pla e si ua ions, hey mus ma ch o be cancelled wi h each o her. Thus in he new elemen s, appropria e polynomial erms in he shear s rain represen a ion were adop ed o overcome he difficul y [9].

A review of he developmen and applica ion of his me hod is presen ed by Hin on and Huang in heir paper en i led "A Family of Quadrila eral Mindlin Plate Elements with Substitute Shear Strain Fields" [9]. They indicated that selectively integrated Mindlin plates may be viewed as fully integrated elements with substitute shear strain fields.

e) Addition of Non-Confirming Displacement Modes

Another approach to improve the basic accuracy of the isoparametric element by eliminating the excessive shear strain is the addition of non-confirming displacement modes. It was firstly adopted by Wilson et al [13] in 1971. Then it was further modified by Cook [14] and Taylor et al [15]. A summary of the development and application of this method is given by Choi and Kim [16]. These two authors have also developed a coupled use of both reduced integration and adding non-confirming modes in quadratic Mindlin plate element [16,17].

1.3.2 Superelement Substructuring Review

In the early 70's, Wilson referred to the static condensation as a similar method to the substructure technique, frontal solution method, and matrix partitioning [18]. He mentioned that all of these methods appear to an application of the basic Gaussian elimination procedure. He also extended the static condensation method to condense the stress-displacement transformation matrix. Therefore, it became unnecessary to calculate the eliminated degrees of freedom to evaluate the element stresses.

GangaRao et al [19] used a macro approach to analyze ribbed and grid plate systems in 1975. Then Gutkowsky and Wang introduced finite panel method for the analysis of continuous isotropic plate systems [20]. Gutkowsky developed the method to analyze orthotropic plate systems [21]. Finally in the same decade, Wah [22] partitioned a quadrilateral plate into two triangles. He applied compatibility equations and used a combination of analytical procedures based on the least square method.

In 1983, Jacobsen [23] introduced a finite element program which combined the multilevel superelement concept with database concepts. He compared his program with other popular finite element programs having superelement and substructure capabilities.

Parsons, GangaRao, and Peterson [24] developed a macro flexibility approach for the analysis of plates in 1985. The deflected shape of their rectangular macroelements were obtained by shape functions which satisfied all four boundary conditions and the bi-harmonic equation. They also verified the convergence of the macro approach.

In late 80's, Liu and Chen [25] presented a two-dimensional superelement mesh generator using variable order triangular or quadrilateral elements. Depending on the geometrical complexity and material variation, their superelements are manually determined to be refined into different linear or quadratic elements.

Li [26] utilized a macroelement substructure to analyze straight box-girder bridges. She used rectangular four-node (bilinear) flat shell elements to model the similar macroelement substructures of the straight box-girder bridge. In 1992, a three-dimensional mesh generator was presented by Martins and Barata Marques [27]. Their program generates the isoparametric superelements consisting of 20-node hexahedral elements specified by the user. It automatically subdivides the analyzed structure into a mesh of 8-node hexahedral elements.

1.3.3 Review of the Caisson Structure Development

White presented a historical review of caissons in Chapter 10 of "Foundation Engineering" [1]. He mentioned the early eighteenth century caisson construction using four-sided wooden boxes. These early open boxes were sunk due to their own weight or by the aid of weights placed on top of them. But it was difficult to attain a great depth because of the wet ground flowing up into the bottom of the box. Brunel (1806-1859), who was a British pioneer in railroad engineering, adapted the pneumatic process to caisson work. He used cast- iron cylinders with bottom cutting edge and heavy weights and air-lock doors at the top. Hence compressed air was introduced to keep the water out.

Being named open and pneumatic caisson, the two above mentioned types are also called drop caissons [1]. They may be defined as enclosing shells for getting a foundation to its final position [28]. Many authors [1,28,29,30,31,32] have explained the development and applications of these caissons, which are mainly used as deep foundations.

Floating caissons, also called box [28,29,31,32] or closed-box [1] caissons, are usually used in shallow water. They are shaped as boxes with sides and bottom. Floating caissons are built on dry land and towed to the site, then they are filled with sand, gravel or concrete to be lowered to the prepared and levelled bed. They have been used in river and harbour development in many parts of the world [28]. The development of floating caissons, which are mainly used as jetties, quay walls, and breakwaters [33], is presented by many investigators [1,28,29,30,31,32,33]. Preliminary design of a quay wall caisson is explained through a numerical example in "Shore Protection Manual" [33]. The stability of a caisson under extreme wave action during construction is evaluated in this example.

In recent years, researchers are mostly concerned with the evaluation of dynamic forces like waves, seismic loads, and ice impacts on marine caissons. In 1988, a software system for the analysis and preliminary design of marine caissons [34], called CAISSON, was developed by Acres International Limited for Public Works, Canada. Dynamic forces are treated like quasi-static loads in this program. It uses a rectangular nine-node Lagrangian Mindlin flat shell element for structural modelling. Also, it has a substructure option which can condense each four adjacent elements into a 25-node superelement.

1.4 Objective and Scope

The objective of this thesis is to develop the superelement substructures for the modelling and analysis of boxlike caisson structures. This method of superelement substructuring has been derived from the classic displacement-based finite element and provides a simpler and more effective procedure in the analysis and design of boxlike caissons.

A special purpose computer program has been developed in this thesis which takes the most advantage from the repetition and other special characteristics of the boxlike caisson structures. Reading a very small input data file, it automatically generates all finite element mesh data including the nodal coordinates and element node numbering. It also generates the equivalent nodal loads of elements pertaining to dead weight, hydrostatic and earth pressure, and concentrated loads.

The computer program is planned to be as general as possible and capable of treating any kinds of rectangular boxlike structures as well as two dimensional plate bending and membrane problems. All displacements, support reactions, and internal forces including; bending moments; shear forces; and membrane stresses can be obtained at required locations.

1.5 Outline of the Thesis

The first chapter introduces a general feature of the present study. Since this thesis is specially concerned with the analysis of boxlike caissons, an introduction to caisson structure is given in Chapter two. Chapter three deals with Mindlin plate bending theory and derives the Mindlin flat shell element.

Multilevel caisson substructuring, which is the main contribution of his hesis, is presented in Chapter four. Chapter five includes application of the program and verification of the results. Several numerical examples are solved and the results are compared with those of other investigators. The last chapter is devoted to the conclusions of present thesis and includes some recommendations for further study.

Appendix A represents a brief explanation and the flow chart of each principal subroutine of the main program. Appendix B shows the sample input files of a 3-D example given in chapter five. Finally, in Appendix C a list of the computer program in FORTRAN is presented.

Chapter 2

CAISSON STRUCTURE

2.1 Introduction

Floating caissons are boxlike structures which are built on land or in a dry dock, launched and towed to the desired site, and then sunk into their final position. They are usually used as marine structures like breakwaters, quay walls, and jetties in shallow waters provided that a fairly strong bed is available. The bed should be prepared and levelled before the sinking process. For sinking the caissons, internal compartments are usually filled with sand or gravel. After installing the caissons, a backfill and/or a concrete deck may be constructed on top of them.

2.2 Geometry and Physical Characteristics

The boxlike caisson considered in this study consists of vertical external walls in a rectangular arrangement on a rectangular base slab. The interior of the caisson may be divided into compartments with vertical internal walls in two perpendicular directions parallel to the external walls.

A caisson may be divided into various layers in height. Material properties and thicknesses may differ from a one layer to another. Each part of a wall in a layer or the base slab, which is separated from its adjacent parts by perpendicular walls, is called a panel. All vertical panels parallel to the length or width of the caisson are called longitudinal or transverse panels, respectively.

A global system of Cartesian coordinates is utilized to define the caisson geometry. Three right-handed axes x, y, and z are parallel to the length, width, and height of the caisson (Fig. 2.1). The origin of the coordinates is at the intersection of the midsurfaces of the two external walls and base slab.

Caissons are usually constructed with reinforced concrete. However, steel and composite steel and concrete structures are also practical alternatives. A steel caisson may be constructed with stiffened flat shells. The combination of steel frames and reinforced concrete slabs could be an example of a composite caisson. In this study, we are concerned with the reinforced concrete caissons made up of simple flat shells.

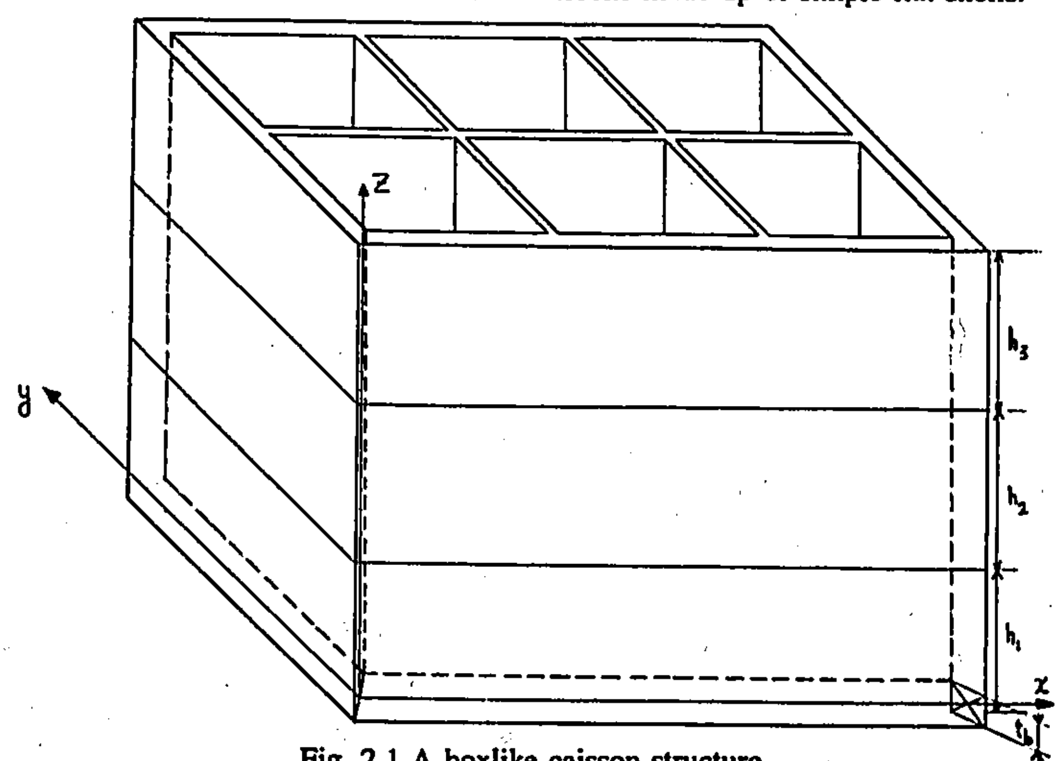


Fig. 2.1 A boxlike caisson structure

2.3 Foundation

The caisson structure is supported by the soil bed (subgrade) which could be considered as an elastic foundation. Using elastic half-space equations [34], the foundation modulus of subgrade reaction is computed as follows.

$$K_x = K_y = \frac{8}{2-\nu} G r_o$$

$$K_z = \frac{4}{1-\nu} G r_o$$

where;

K_x , K_y = lumped elastic stiffness in horizontal directions

K_z = " " " " vertical direction

G = shear modulus of the foundation material

ν = Poisson's ratio " " " "

$r_o = \sqrt{\frac{BL}{\pi}}$ = *equivalent raduis*

B = foundation width

L = " length

The elastic modulus of subgrade reaction, k is equal to the lumped stiffness divided by the foundation area (e.g. $k_x = K_x / A_f$). It should be noted that for the small values of poisson's ratio, vertical and horizontal modulus of subgrade reaction are almost equal.

$$K \approx 4 G r_o \qquad\qquad k = \frac{K}{A_f} \approx \frac{4 G r_o}{A_f}$$

The computer program of this thesis utilizes a single value, specified by the user, for subgrade modulus in all three directions.

2.4 Loading

Both static and dynamic loads should be considered in a complete structural analysis of the caissons. Static loads are; dead weight of the ballast including water and/or soil and the structure itself; hydrostatic and soil pressure; and external concentrated forces. Dynamic loads include ice, wave, and seismic forces. Wind force is usually negligible due to the small projected area of the caisson above its waterline. Dynamic loads are beyond the scope of this study. However, they can be easily considered as quasi-static loads.

a) Dead weight

Knowing the geometry and material specific gravity of layers, one can easily calculate the dead weight of a caisson structure. The ballast weight is also determined provided that the levels of water or soil and their specific gravities are given at all compartments for each load case. Dead weights are applied in downward vertical (-z) direction.

b) Hydrostatic pressure

Hydrostatic pressure increases linearly with the increase of water depth ($p = \gamma h$).

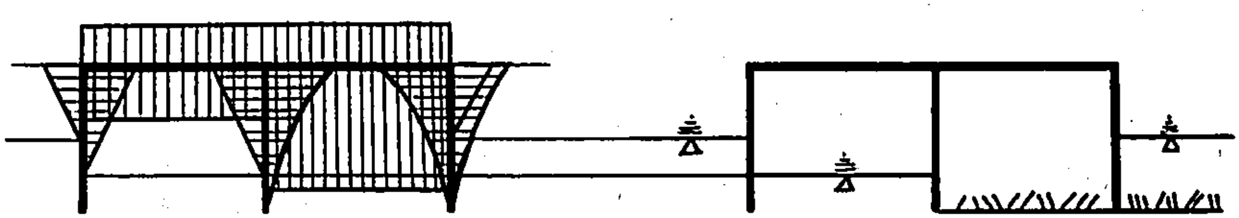


Fig. 2.2 Hydrostatic and soil pressure on a boxlike caisson

It exerts a lateral linearly varying and an uplift force on vertical and base panels, respectively (Fig. 2.2). As it was mentioned before, hydrostatic or soil pressure on base panels is considered as a ballast dead weight regarding internal compartments.

c) Soil pressure

The lateral soil pressure force is computed assuming a granular cohesionless backfill using Coulomb's theory of active earth pressure. Generally, it is assumed that the caisson moves away from the soil mass [34], hence active soil pressure is developed. Terzaghi [35] gives the active lateral soil force predicted by Coulomb's theory as:

$$P_A = \frac{1}{2}\left(\gamma + \frac{2Nq}{Z}\right) Z^2 \frac{K_A}{Sin\ \alpha\ Cos\ \delta}$$

$$N = \frac{Sin\ \alpha}{Sin\ (\alpha+\beta)}$$

$$K_A = \frac{Sin^2(\alpha+\phi)\ Cos\ \delta}{Sin\ \alpha\ Sin\ (\alpha-\delta)\left[1 + \left[\frac{Sin\ (\phi+\delta)\ Sin\ (\phi-\beta)}{Sin\ (\alpha-\delta)\ Sin\ (\alpha+\beta)}\right]^{0.5}\right]^2}$$

where;

γ = soil specific gravity (dry or submerged for soil above or below water table, respectively),

q = vertical surcharge on top of backfill,

Z = distance from the top of backfill,

ϕ = soil internal friction angle,

α = wall inclination from horizontal (always 90 degree for vertical wall caisson),

β = slope of backfill surface from horizontal,

and δ = wall friction angle (with soil).

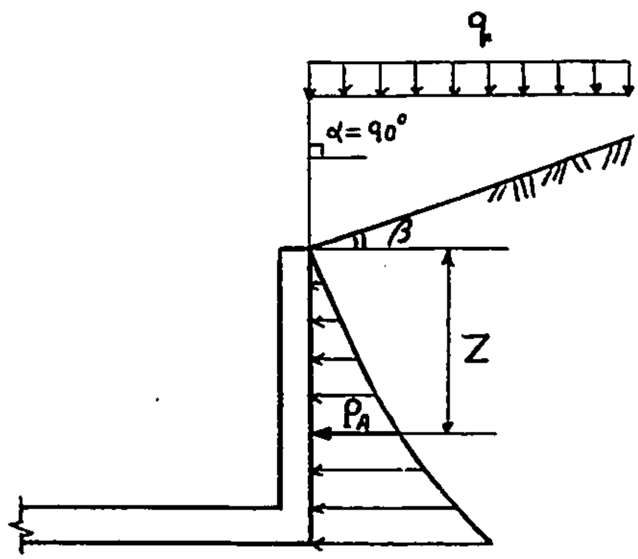


Fig. 2.3 Lateral soil pressure on a vertical wall

d) Concentrated forces

The external concentrated forces and their point of application are also considered in this study. Each set of forces may include six (3 forces and 3 moments) components, and its point of application is defined in global coordinates. These forces represent crane loads, bollards, and so on.

Chapter 3

MINDLIN FLAT SHELL ELEMENT

3.1 Introduction

A marine caisson is a three dimensional structure made up of some fairly thick panels which sustain both lateral and in-plane loads. Flat shell elements, with both plate bending and membrane behaviour, are used for structural modelling.

Mindlin plate elements, which account for transverse shear deformation as well as bending deformation, can model the fairly thick panels of a caisson better than classical thin plate elements. Therefore, a Mindlin flat shell element, which is the combination of a Mindlin plate bending element and a membrane element, is developed to model the caisson structure.

Because in this case the bending and membrane actions are not coupled, the bending element and membrane element are developed separately and then combined together into a Mindlin flat shell element. Because of the rectangular geometry of caisson, rectangular element is a good choice, and a nine-node Lagrangian element with quadratic interpolation functions in two directions is adopted.

All superelements used in this caisson analysis are made up of nine-node rectangular Mindlin flat shell elements.

3.2 Plate Bending Element

3.2.1 General

The plate bending element adopted in this thesis is based on Mindlin plate theory. Mindlin plate theory assumes: 1) points on the middle surface of a plate move only in the lateral direction, and 2) any line that is normal to the mid-surface remains a straight line during deformation, but this line is not necessarily normal to the deformed mid-surface. Thus transverse shear deformation is allowed, and the displacement of a point not on the mid-surface depends on rotations θ_x and θ_y. These are the rotations of the line that was normal to the mid-surface of the undeformed plate, and are independent of the lateral displacement of the mid-surface. Therefore, any point on the mid-surface of a Mindlin plate has three independent displacements: lateral displacement w, and normal rotations θ_x and θ_y.

Since a Mindlin plate has three independent degrees of freedom at any point, it is a C-0 type element. This type of element only needs the continuity of interpolated values themselves, not their derivatives. Whereas in a classic (Kirchhoff) thin plate element, the rotations θ_x and θ_y are the derivatives of w in x and y directions. Thus a classic thin plate is a C-1 element which needs the continuity of interpolated quantities as well as their first derivatives. Therefore, the formulation of a Mindlin plate element is easier than that of a classic thin plate element [36].

3.2.2 Shape Function

The shape functions of a nine-node Lagrangian (quadrilateral isoparametric) element are given by many authors [36,37]. They can be derived by using quadratic interpolation functions in two directions as follows:

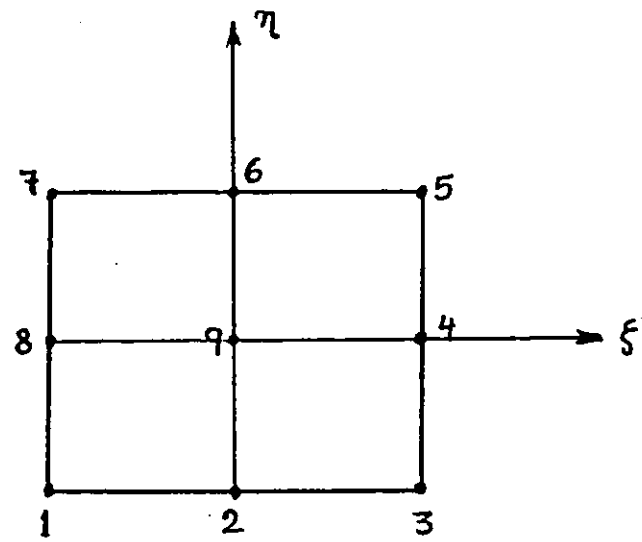


Fig 3.1 A rectangular nine node Lagrangian element

$$N_1 = \xi\eta(1-\xi)(1-\eta)/4 \qquad N_2 = \eta(1-\xi^2)(\eta-1)/2$$

$$N_3 = \xi\eta(1+\xi)(1-\eta)/4 \qquad N_4 = \xi(1-\eta^2)(\xi+1)/2$$

$$N_5 = \xi\eta(1+\xi)(1+\eta)/4 \qquad N_6 = \eta(1-\xi^2)(\eta+1)/2$$

$$N_7 = \xi\eta(1-\xi)(1+\eta)/4 \qquad N_8 = \xi(1-\eta^2)(\xi-1)/2$$

$$N_9 = (1-\xi^2)(1-\eta^2) \qquad (3.1)$$

where ξ and η are natural coordinates (between -1 and +1). Any shape function N_i is equal to 1 at node i and equal to 0 at other nodes.

These shape functions can be used to find all three independent degrees of freedom at any point of (ξ,η):

$$w = \sum_{i=1}^{i=9} N_i w_i = N_1 w_1 + N_2 w_2 + \ldots + N_9 w_9$$

$$\theta_x = \sum_{i=1}^{i=9} N_i \theta_{xi} = N_1 \theta_{x1} + N_2 \theta_{x2} + \ldots + N_9 \theta_{x9} \qquad (3.2)$$

$$\theta_y = \sum_{i=1}^{i=9} N_i \theta_{yi} = N_1 \theta_{y1} + N_2 \theta_{y2} + \ldots + N_9 \theta_{y9}$$

where w_i , θ_{xi} and θ_{yi} are the nodal displacements and N_i is the shape function value of point (ξ,η) at node i.

The cartesian coordinates of any point of an element are also interpolated with these shape functions:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \sum_{i=1}^{i=9} \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} \qquad (3.3)$$

Jacobian matrix of element is defined as:

$$\underline{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \sum_{i=1}^{i=9} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} \{x_i \quad y_i\} \qquad (3.4)$$

The inverse of $\underline{J}$ may be expressed as:

$$\underline{J^{-1}} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \qquad (3.5)$$

where $|J|$ is determinant of the Jacobian matrix.

To evaluate the cartesian shape function derivatives, the chain rule of differentiation is used:

$$\frac{\partial N_i}{\partial x} = \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial x}$$
$$\frac{\partial N_i}{\partial y} = \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial y} \qquad (3.6)$$

or in matrix form:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = \underline{J}^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} \qquad (3.7)$$

The strain-displacement matrix, B, can then be explained in terms of shape functions and their Cartesian derivatives.

3.2.3 Strain-Displacement Relation

For the element shown in fig. 3.2, the bending strains ϵ_{xx} , ϵ_{yy} , and γ_{xy} vary linearly through the plate thickness.

$$\underline{\epsilon_b} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = z \begin{Bmatrix} \frac{\partial \theta_y}{\partial x} \\ -\frac{\partial \theta_x}{\partial y} \\ -\frac{\partial \theta_x}{\partial x} + \frac{\partial \theta_y}{\partial y} \end{Bmatrix} = z \begin{Bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{Bmatrix} = z\ \underline{\kappa_b} \qquad (3.8)$$

where κ_{xx} , κ_{yy} , and κ_{xy} are curvatures.

The transverse shear strains are assumed to be constant through the element thickness.

$$\underline{\epsilon_s} = \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial w}{\partial x} + \theta_y \\ \dfrac{\partial w}{\partial y} - \theta_x \end{Bmatrix} = \begin{Bmatrix} \kappa_{xz} \\ \kappa_{yz} \end{Bmatrix} = \underline{\kappa_s} \tag{3.9}$$

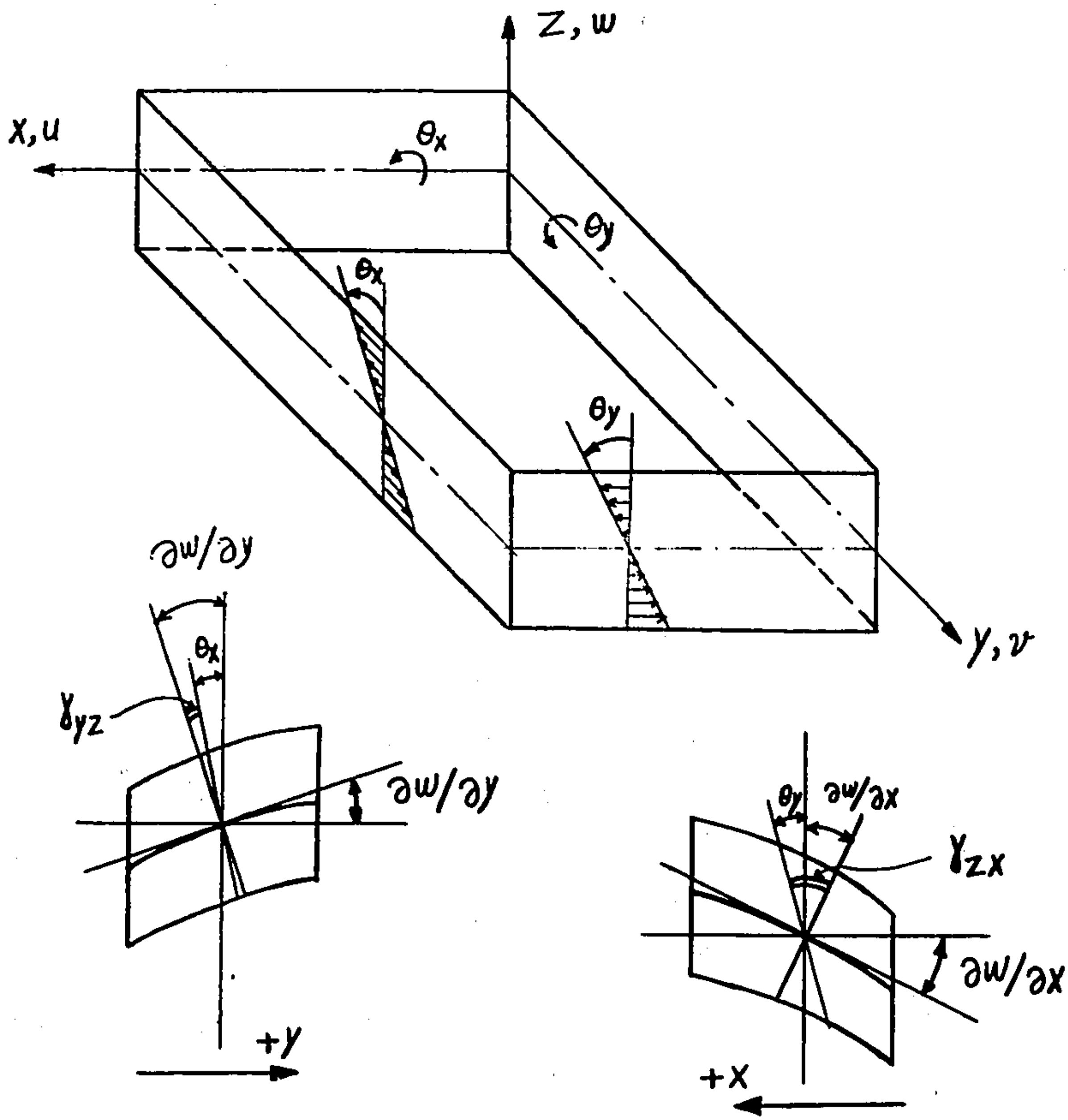


Fig. 3.2 A Mindlin plate bending element

3.2.4 Stress-strain Relation

For plane stress condition, i.e. $\sigma_{zz} = 0$, and for an isotropic material, plate bending and shear stresses are as follows:

a) bending stresses:

$$\underline{\sigma_b} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \tag{3.10}$$

b) shear stresses:

$$\underline{\sigma_s} = \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = G \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \tag{3.11}$$

where G = $\frac{E}{2(1+\nu)}$ = shear modulus,

E = Elastic modulus,

ν = Poisson's ratio.

Since the bending strains and consequently the bending stresses vary linearly with z, it is desirable to express the stress field in terms of bending moments. The basic equations for bending moments per unit width of the plate are:

$$M_{xx} = \int_{-t/2}^{t/2} \sigma_{xx}\, z\, dz \qquad M_{yy} = \int_{-t/2}^{t/2} \sigma_{yy}\, z\, dz \qquad M_{xy} = \int_{-t/2}^{t/2} \tau_{xy}\, z\, dz$$

Substituting $\underline{\sigma_b}$ from eq. (3.10) and $\underline{\epsilon_b}$ from eq. (3.8) in the above equations gives:

$$\underline{M} = \begin{Bmatrix} M_{xx} \\ M_{yy} \\ M_{xy} \end{Bmatrix} = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{Bmatrix} = \underline{D}_b \cdot \underline{\kappa}_b \tag{3.12}$$

where $D = \dfrac{E}{1-\nu^2} \displaystyle\int_{-\frac{t}{2}}^{\frac{t}{2}} z^2\, dz = \dfrac{Et^3}{12\,(1-\nu^2)}$ is the flexural rigidity of the plate.

Similarly, for the shear forces:

$$V_{xz} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{xz}\, dz = k\tau_{xz}\, t \qquad\qquad V_{yz} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{yz}\, dz = k\tau_{yz}\, t$$

where k is the shear correction factor, which permits the parabolic distribution of τ_{zx} and τ_{yz} to be replaced by uniform distributions. Assuming k=5/6 for a rectangular cross section [36], we can write the above equations as:

$$\underline{V} = \begin{Bmatrix} V_{xz} \\ V_{yz} \end{Bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \underline{D}_s \cdot \underline{\kappa}_s \tag{3.13}$$

where $S_x = S_y = \dfrac{5\,Gt}{6} = \dfrac{5\,Et}{12\,(1+\nu)}$.

Differentiating the basic relations for displacements (equations 3.1) with respect to x gives:

$$\frac{\partial w}{\partial x} = \sum_{i=1}^{i=9} \frac{\partial N_i}{\partial x} w_i \qquad \frac{\partial \theta_x}{\partial x} = \sum_{i=1}^{i=9} \frac{\partial N_i}{\partial x} \theta_{xi} \qquad \frac{\partial \theta_y}{\partial x} = \sum_{i=1}^{i=9} \frac{\partial N_i}{\partial y} \theta_{yi}$$

And similar relationships could be derived for y direction. Now we can write the curvature vectors $\underline{\kappa}_b$ and $\underline{\kappa}_s$ in terms of the nodal displacements:

$$\underline{\kappa}_b = \begin{Bmatrix} \frac{\partial \theta_y}{\partial x} \\ -\frac{\partial \theta_x}{\partial y} \\ \frac{\partial \theta_y}{\partial y} - \frac{\partial \theta_x}{\partial x} \end{Bmatrix} = \sum_{i=1}^{i=9} \begin{bmatrix} 0 & 0 & \frac{\partial N_i}{\partial x} \\ 0 & -\frac{\partial N_i}{\partial y} & 0 \\ 0 & -\frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \end{bmatrix} \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} = \sum_{i=1}^{i=9} \underline{B}_{b_i} \, \underline{d}_{p_i} \tag{3.14}$$

$$\underline{\kappa}_s = \begin{Bmatrix} \frac{\partial w}{\partial x} + \theta_y \\ \frac{\partial w}{\partial y} - \theta_x \end{Bmatrix} = \sum_{i=1}^{i=9} \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & N_i \\ \frac{\partial N_i}{\partial y} & -N_i & 0 \end{bmatrix} \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} = \sum_{i=1}^{i=9} \underline{B}_{s_i} \, \underline{d}_{p_i} \tag{3.15}$$

where;

$\underline{d}_{p_i}$ = vector of normal displacements of node i,

$\underline{B}_{b_i}$ = bending strain-displacement submatrix of node i,

$\underline{B}_{s_i}$ = shear strain-displacement submatrix of node i.

3.2.5 Stiffness Matrix

From virtual work consideration, it can be shown that:

$$K = \int \underline{B}^T \underline{D} \, \underline{B} \, dA \tag{3.16}$$

where;

$\underline{K}$ = stiffness matrix of element,

$\underline{B}$ = strain-displacement matrix of element,

$\underline{D}$ = constitutive matrix,

dA = element area.

Now we can write the bending and shear parts of the stiffness matrix of the Mindlin plate element.

For the bending part :

$$\underline{K_b} = \int \underline{B_b^T}\, \underline{D_b}\, \underline{B_b}\, dA \qquad (3.17)$$

where $\underline{B_b} = [\, \underline{B_{b_1}}\ ,\ \underline{B_{b_2}}\ ,\ \ldots\ ,\ \underline{B_{b_9}}] = [\ \underline{B_{b_i}}]\ ,\ i=1,2,\ldots,9\ .$

Substituting $\underline{B_b}$ in eq. (3.17) leads to the following equation:

$$\underline{K_b}_{(27\times27)} = \begin{bmatrix} \underline{K_b}_{11} & \underline{K_b}_{12} & \cdots & \underline{K_b}_{19} \\ \underline{K_b}_{21} & \underline{K_b}_{22} & \cdots & \underline{K_b}_{29} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \underline{K_b}_{91} & \underline{K_b}_{92} & \cdots & \underline{K_b}_{99} \end{bmatrix} = [\underline{K_b}_{ij\,(3\times3)}] \qquad (3.18)$$

where $\underline{K_b}_{ij} = \int \underline{B_b}_i^{\,T}\, \underline{D_b}\, \underline{B_b}_j\, dA$, and $\begin{cases} i = 1,2,\ldots,9 \\ j = 1,2,\ldots,9 \end{cases}$.

And similarly, the shear part of stiffness matrix is:

$$\underline{K_s}_{(27\times27)} = [\ \underline{K_s}_{ij\,(3\times3)}\ ]\ ,\ \begin{cases} i=1,2,\ldots,9 \\ j=1,2,\ldots,9 \end{cases} \qquad (3.19)$$

where $\underline{K_s}_{ij} = \int \underline{B_s}_i\, \underline{D_s}\, \underline{B_s}_j\, dA$ . .

The stiffness matrix of the Mindlin plate element is sum of the bending and shear contributions:

$$\underline{K_p}_{(27\times27)} = \underline{K_b} + \underline{K_s} = [\underline{K_p}_{ij\,(3\times3)}]\ ,\ \begin{cases} i=1,2,\ldots,9 \\ j=1,2,\ldots,9 \end{cases} \qquad (3.20)$$

where $\underline{K_p}_{ij} = \underline{K_b}_{ij} + \underline{K_s}_{ij} = \int (\underline{B_b}_i^T \underline{D_b} \underline{B_b}_j + \underline{B_s}_i^T \underline{D_s} \underline{B_s}_j) \; dA$.

The above integrals for $\underline{K_b}_{ij}$ and $\underline{K_s}_{ij}$ are calculated numerically by Gaussian quadrature method for quadrilateral area A:

$$\begin{aligned} \underline{K_b}_{ij} &= \int_{-1}^{1}\int_{-1}^{1} \underline{B_b}_i^T \underline{D_b} \underline{B_b}_j \; |J| \; d\xi \; d\eta \\ \underline{K_s}_{ij} &= \int_{-1}^{1}\int_{-1}^{1} \underline{B_s}_i^T \underline{D_s} \underline{B_s}_j \; |J| \; d\xi \; d\eta \end{aligned} \qquad (3.21)$$

The stiffness matrix of a rectangular element with midside nodes is integrated exactly by full integration, i.e. quadrature order 3. Since each integration point imposes two constraints for shear strains, the full integration may lead to shear lock phenomena for very thin plates. Choosing a reduced or selective integration and redefining the transverse shear interpolation are two techniques to overcome this difficulty [36].

In this thesis, the bending stiffness is determined using quadrature order 3, and quadrature order 2 is used to integrate the shear stiffness. This selective integration rule provides the correct contribution for shear component in rectangular and parallelogram shaped elements [5]. Also, it prevents shear lock for the case of a very thin element.

3.3 Membrane Element

3.3.1 General

A nine-node rectangular membrane element with two degrees of freedom at each node is adopted. These two degrees of freedom are in plane displacements u and v in the x and y directions respectively.

3.3.2 Shape Function

Again the same shape functions for a nine-node Lagrangian element are used to interpolate the nodal degrees of freedom.

$$u = \sum_{i=1}^{i=9} N_i \; u_i = N_1 \; u_1 + N_2 \; u_2 + \ldots + N_9 \; u_9$$

$$v = \sum_{i=1}^{i=9} N_i \; v_i = N_1 \; v_1 + N_2 \; v_2 + \ldots + N_9 \; v_9 \tag{3.22}$$

3.3.3 Strain-Displacement Relation

Strain-displacement relation for a membrane element [36] is presented in the following equation:

$$\underline{\epsilon_m} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} \tag{3.23}$$

Differentiating eq. (3.22) with respect to x and y and substituting in eq. (3.23) gives:

$$\underline{\epsilon_m} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \sum_{i=1}^{i=9} \begin{bmatrix} \dfrac{\partial N_i}{\partial x} & 0 \\ 0 & \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial y} & \dfrac{\partial N_i}{\partial x} \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} = \sum_{i=1}^{i=9} \underline{B_m}_i \; \underline{d_m}_i \tag{3.24}$$

where;

$\underline{d}_{m_i}$ = vector of in-plane displacements of node i,

$\underline{B}_{m_i}$ = membrane strain-displacement submatrix of node i.

3.3.4 Stress-Strain Relation

The membrane stresses of an isotropic material are:

$$\underline{\sigma}_m = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \tag{3.25}$$

or in simple form: $\underline{\sigma}_m = \underline{D}_m \, \underline{\epsilon}_m$

where $\underline{D}_m$ is membrane constitutive matrix.

3.3.5 Stiffness Matrix

The stiffness matrix of a membrane element can be obtained as follows:

$$\underline{K}_m = \int \underline{B}_m^T \, \underline{D}_m \, \underline{B}_m \, dA \tag{3.26}$$

where $\underline{B}_m = [\ \underline{B}_{m_1}\ ,\ \underline{B}_{m_2}\ ,\ \ldots\ ,\ \underline{B}_{m_9}] = [\ \underline{B}_{m_i}]\ ,\ i=1,2,\ldots,9$.

$$
\underline{K_m}_{(18\times18)} = \begin{bmatrix} \underline{K_m}_{11} & \underline{K_m}_{12} & \cdots & \underline{K_m}_{19} \\ \underline{K_m}_{21} & \underline{K_m}_{22} & \cdots & \underline{K_m}_{29} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \underline{K_m}_{91} & \underline{K_m}_{92} & \cdots & \underline{K_m}_{99} \end{bmatrix} = [\ \underline{K_m}_{ij\,(2\times2)}\ ] \qquad (3.27)
$$

where ;

$$
\underline{K_m}_{ij} = \int \underline{B_m}_i^T \, \underline{D_m} \, \underline{B_m}_j \; dA = \int_{-1}^{1}\int_{-1}^{1} \underline{B_m}_i^T \, \underline{D_m} \, \underline{B_m}_j \; |J| \; d\xi \; d\eta \quad ,
$$

and $\begin{cases} i = 1, 2, \ldots, 9 \\ j = 1, 2, \ldots, 9 \end{cases}$.

To calculate the above integral, a Gaussian quadrature of order 2 is used.

3.4 Mindlin Flat Shell Element

3.4.1 General

Mindlin flat shell element is the combination of a Mindlin plate bending element and a membrane element. It is subjected to both lateral and in-plane loads. This element has five degrees of freedom per node. They include two in-plane displacements (u and v) and three normal displacements (w, θ_x , and θ_y).

3.4.2 Stiffness Matrix

The stiffness matrix of a Mindlin flat shell element is easily obtained by combining the corresponding stiffness matrices of plate and membrane elements [36]. There is no interaction between the two submatrices because the normal displacements associated with plate behaviour do not affect the in-plane displacements associated with membrane action and vice versa.

Since the flat shell element is to be used for modelling the three dimensional caisson structure, the sixth degree of freedom of in-plane rotation, θz, is also considered. Now the stiffness matrix of the Mindlin flat shell element can be written as:

$$\underline{K_f}_{(54\times54)} = \begin{bmatrix} \underline{K_f}_{11} & \underline{K_f}_{12} & \cdots & \underline{K_f}_{19} \\ \underline{K_f}_{21} & \underline{K_f}_{22} & \cdots & \underline{K_f}_{29} \\ \vdots & \vdots & \cdots & \vdots \\ \underline{K_f}_{91} & \underline{K_f}_{92} & \cdots & \underline{K_f}_{99} \end{bmatrix} = [\,\underline{K_f}_{ij\,(6\times6)}\,] \tag{3.28}$$

where each flat shell stiffness submatrix $\underline{K_f}_{ij}$ is:

$$\underline{K_f}_{ij} = \begin{bmatrix} \underline{K_m}_{ij} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & \underline{K_p}_{ij} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 \end{matrix} & 0 \end{bmatrix} \begin{matrix} \ldots u \\ \ldots v \\ \ldots w \\ \ldots \theta_x \\ \ldots \theta_y \\ \ldots \theta_z \end{matrix} \;, and \begin{cases} i=1,2,\ldots,9 \\ j=1,2,\ldots,9 \end{cases} \tag{3.29}$$

and $\underline{K_p}_{ij}$ and $\underline{K_m}_{ij}$ are the corresponding stiffness submatrices for plate bending element (3×3), and membrane element (2×2), respectively.

We can rearrange the stiffness matrix of the flat shell element as it appears in the following equilibrium equation:

$$
\underline{F_\ell}=\begin{Bmatrix} \underline{F_x} \\ \underline{F_y} \\ \hline \underline{F_z} \\ \underline{M_x} \\ \underline{M_y} \\ \hline \underline{M_z} \end{Bmatrix}=\left[\begin{array}{c|c|c} [K_m]_{18\times18} & [0]_{18\times27} & [0]_{18\times9} \\ \hline [0]_{27\times18} & [K_p]_{27\times27} & [0]_{27\times9} \\ \hline [0]_{9\times18} & [0]_{9\times27} & [0]_{9\times9} \end{array}\right]\begin{Bmatrix} \underline{u} \\ \underline{v} \\ \hline \underline{w} \\ \underline{\theta_x} \\ \underline{\theta_y} \\ \hline \underline{\theta_z} \end{Bmatrix} \tag{3.30}
$$

where each of the force vectors ($\underline{F_x}$, $\underline{F_y}$, ...) and displacement vectors ($\underline{u}$, $\underline{v}$,...) consists of nodal forces and displacements, respectively.

As shown in eq. 3.30, there are zero submatrices associated with $[\theta_z]$ in the stiffness matrix. The reason is that θ_z was not considered as a degree of freedom (drilling freedom) in developing the flat shell stiffness matrix. To overcome this problem, which leads to the singularity of stiffness matrix, we can replace the $[0]_{9\times9}$ with the matrix shown in the following equation:

$$
\begin{Bmatrix} M_{z1} \\ M_{z2} \\ M_{z3} \\ \cdot \\ \cdot \\ \cdot \\ M_{z9} \end{Bmatrix}=\alpha E V\begin{bmatrix} 1 & -1/8 & -1/8 & \dots & -1/8 \\ -1/8 & 1 & -1/8 & \dots & -1/8 \\ -1/8 & -1/8 & 1 & \dots & -1/8 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ -1/8 & -1/8 & -1/8 & \dots & 1 \end{bmatrix}\begin{Bmatrix} \theta_{z1} \\ \theta_{z2} \\ \theta_{z3} \\ \cdot \\ \cdot \\ \cdot \\ \theta_{z9} \end{Bmatrix} \tag{3.31}
$$

where E is elastic modulus, V is element volume, and α is a number such as 0.3 or less. The added matrix provides each θ_z degree of freedom with a fictitious stiffness but offers no resistance to the mode $\theta_{z1} = \theta_{z2} = \dots = \theta_{z9}$ or to any other rigid body motion [36].

3.4.3 Equivalent Load Vector

Equivalent load vector for each element is made up of equivalent nodal load vectors:

$$\underline{F} = \{\underline{F}_1, \underline{F}_2, \ldots, \underline{F}_9\}^T = \{\underline{F}_i\}^T, \quad i=1,2,\ldots,9 \tag{3.32}$$

where $\underline{F}_i = \{F_{xi}, F_{yi}, F_{zi}, M_{xi}, M_{yi}, M_{zi}\}^T$.

Two load types are considered. They are concentrated load, and distributed surface load.

a) The equivalent nodal load vector due to each concentrated load P acting at the point(ξ_1 , η_1) of the natural coordinate system can be derived from the following equation:

$$\begin{Bmatrix} F_{xi} \\ F_{yi} \\ F_{zi} \\ M_{xi} \\ M_{yi} \\ M_{zi} \end{Bmatrix} = N_i(\xi_1, \eta_1) \begin{Bmatrix} P_x \\ P_y \\ P_z \\ P_{xx} \\ P_{yy} \\ P_{zz} \end{Bmatrix} \tag{3.33}$$

or $\underline{F}_i = N_i(\xi_1, \eta_1)\,\underline{P}$ where $N_i(\xi_1, \eta_1)$ is the shape function value of point(ξ_1 , η_1) at node i, and $\underline{P}$ consists of six possible components of load P (three forces and three moments in x, y, and z directions respectively).

b) Distributed lateral load q acting in local z direction gives the following equivalent nodal load vector:

$$\underline{F}_i = \{0, 0, F_{zi}, 0, 0, 0\} \tag{3.34}$$

where;

$$F_{zi} = \int q\ N_i\ dA = \int_{-1}^{1}\int_{-1}^{1} q\ N_i\ |J|\ d\xi\ d\eta \quad .$$

The above integral is determined numerically using the Gaussian quadrature method for quadrilateral area A.

$$F_{zi} = \sum_{j=1}^{n} \sum_{k=1}^{n} W_j\ W_k\ N_i(\xi_{jk}, \eta_{jk})\ q_{jk} \qquad (3.35)$$

where;

W_j , W_k = Gaussian weights

ξ_{jk} , η_{jk} = natural coordinates of each Gauss point (j , k)

q_{jk} = value of the lateral load q at each Gauss point (j , k)

n = order of integration.

Generally, a more rapidly changing function q(x,y) needs a higher order of integration. For smooth distributions, a third order (n=3) integration provides enough accuracy.

For special case of a uniformly distributed lateral load ($q=q_0$) we have:

$$F_{zi} = q_0 \int_{-1}^{1}\int_{-1}^{1} N_i\ |J|\ d\xi\ d\eta = q_0\ \alpha_i \qquad (3.36)$$

where:

$$\underline{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_9\}^T = \frac{A}{36}\{1, 4, 1, 4, 1, 4, 1, 4, 16\}^T \quad .$$

Therefore, the whole equivalent element load vector, which only includes z components is:

$$E_z = \frac{q_0 A}{36} \{1, 4, 1, 4, 1, 4, 1, 4, 16\}^T \tag{3.37}$$

The dead weight of each element (gravity load), which acts in the global Z direction, has a similar equivalent load vector with $q_0 = t\gamma$ being the weight per unit area of element.

3.5 Transformation of Coordinates

3.5.1 General

To assemble individual elements into a three dimensional structure, it is necessary to establish a global coordinate system. Then each element could be transformed from its local coordinates to the global one.

3.5.2 Transformation Matrix

A Mindlin flat shell element in local coordinates has already been discussed in section 3.4. A transformation matrix, $\underline{T}$, can transform the displacement components from global coordinats xyz to local coordinates x'y'z' as follows:

$$\underline{d}' = \underline{T}\,\underline{d} \tag{3.38}$$

where $\underline{d}'$ and $\underline{d}$ are element displacement vectors in local and global coordinates, respectively. Since the same amount of work is done in the two coordinate systems, we have:

$$\underline{F}'\underline{d}' = \underline{F}\underline{d} \tag{3.39}$$

Substiuting eq. (3.38) in the above equation and simplifying gives:

$$\underline{F} = \underline{T}^T \underline{F}' \tag{3.40}$$

Then the transformed stiffness matrix, $\underline{\mathbf{K}}$, is derived as:

$$\underline{K} = \underline{T}^T \underline{K}_f \underline{T} \tag{3.41}$$

where $\underline{\mathbf{K}}_f$ is the stiffness matrix in the local coordinates (eq. 3.28), and $\underline{\mathbf{T}}$ is shown in the following equation:

$$\underline{T} = \begin{bmatrix} \underline{T}_{11} & \underline{T}_{12} & \cdot & \cdot & \cdot & \underline{T}_{19} \\ \underline{T}_{21} & \underline{T}_{22} & \cdot & \cdot & \cdot & \underline{T}_{29} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \underline{T}_{91} & \underline{T}_{92} & \cdot & \cdot & \cdot & \underline{T}_{99} \end{bmatrix} = [\underline{T}_{ij}] \tag{3.42}$$

where;

$$\begin{cases} \underline{T}_{ij} = \underline{L} & for \quad i=j \\ \underline{T}_{ij} = \underline{0} & for \quad i \neq j \end{cases} \quad and \quad \begin{cases} i=1, 2, \ldots, 9 \\ j=1, 2, \ldots, 9 \end{cases}$$

$$\underline{L} = \begin{bmatrix} \underline{\lambda} & \underline{0} \\ \underline{0} & \underline{\lambda} \end{bmatrix}$$

$$\underline{\lambda} = \begin{bmatrix} \lambda_{x'x} & \lambda_{x'y} & \lambda_{x'z} \\ \lambda_{y'x} & \lambda_{y'y} & \lambda_{y'z} \\ \lambda_{z'x} & \lambda_{z'y} & \lambda_{z'z} \end{bmatrix}$$

and $\lambda_{x'x}$ = cosine of angle between x' and x, and so on.

3.5.3 Caisson Transformation Matrices

A caisson structure has three sets of panels including base, longitudinal, and transverse panels. The first one lies in horizental xy plane, and the next two lie in vertical xz and yz planes, respectively (fig.3.3). Three different transformation matrices are considered for these three sets of panels.

a) For base panels, the local and global coordinates are the same:

$$\lambda_{x'x} = \lambda_{y'y} = \lambda_{z'z} = 1$$
$$\lambda_{x'y} = \lambda_{x'z} = 0$$
$$\lambda_{y'x} = \lambda_{y'z} = 0$$
$$\lambda_{z'x} = \lambda_{z'y} = 0$$

or in matrix form:

$$\lambda_{bp} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{3\times 3}$$

$$L_{bp} = I_{6\times 6}$$
$$T_{bp} = I_{54\times 54}$$

In other words, no transformation is required for base panels.

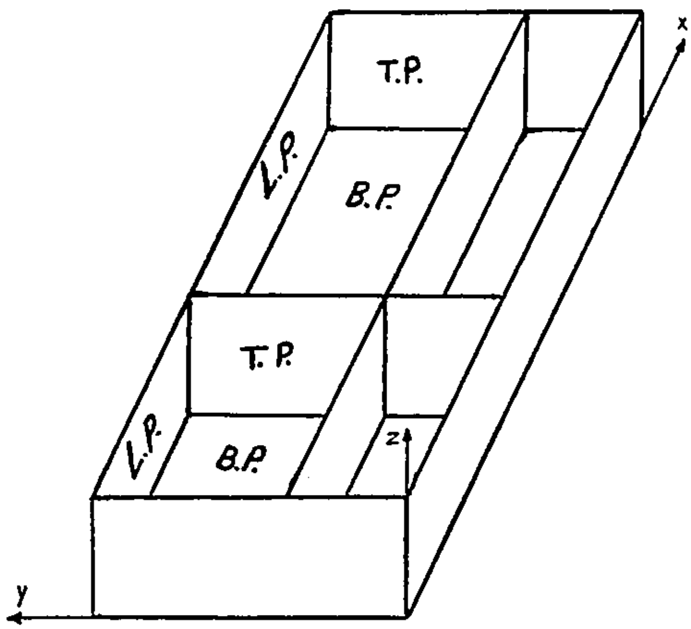


Fig. 3.3 Three orthogonal planes of caisson panels

b) For longitudinal panels:

$$\lambda_{x'x} = \lambda_{y'z} = -\lambda_{z'y} = 1$$
$$\lambda_{x'y} = \lambda_{x'z} = 0$$
$$\lambda_{y'x} = \lambda_{y'y} = 0$$
$$\lambda_{z'x} = \lambda_{z'z} = 0$$

or in matrix form:

$$\lambda_{lp} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

c) For transverse panels:

$$-\lambda_{x'y} = \lambda_{y'z} = -\lambda_{z'x} = 1$$
$$\lambda_{x'x} = \lambda_{x'z} = 0$$
$$\lambda_{y'x} = \lambda_{y'y} = 0$$
$$\lambda_{z'y} = \lambda_{z'z} = 0$$

or in matrix form:

$$\lambda_{tp} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

Chapter 4

MULTILEVEL CAISSON SUBSTRUCTURING

4.1 General

Static condensation method is applied to obtain superelement substructures from the assemblage of flat shell elements. This is an exact method and does not include any approximations. It leads to the same accurate results as those given by the corresponding finite elements used in the substructure assemblage. The procedure of static condensation and its application to multilevel caisson substructuring are discussed in this chapter. Also, an outline of the computer program is presented.

4.2 The Procedure of Static Condensation

In this approach, any substructure will be modeled as a superelement which is condensed to its boundary nodes. The nodes of each substructure are divided into internal and boundary nodes. The internal nodes, also referred to as condensed or slave nodes, lie within the internal domain of a substructure. Whereas the boundary nodes, also referred to as remained or master nodes, lie on the boundary of the substructure and may be common with another adjacent substructure.

By forcing displacement compatibility of common nodes, it is possible to determine displacements and internal forces at each node of a substructure in terms of the boundary nodal displacements. This allows to eliminate the internal "degrees of freedom" (dof's) and form the substructure. Thus, the structural equilibrium equations are only associated with common boundary dof's between substructures. By solving these equilibrium equations, common boundary nodal displacements are obtained first. Then, the displacements and internal forces of internal nodes can be easily obtained by applying boundary constraints.

The method outlined above acts analogous to a Gauss elimination prior to the stiffness assemblage of a substructure. Thus, it reduces the size of the stiffness matrix and enhances the storage capacity of the computer program. The procedure of static condensation is formulated below.

The equilibrium equation is of the form:

$$\underset{\sim}{S}\,\underset{\sim}{\delta} = \underset{\sim}{P} \tag{4.1}$$

where $\underline{S}$, $\underline{\delta}$, and $\underline{P}$ are stiffness matrix, displacement vector, and load vector of a substructure, respectively. They can be partitioned as shown in the following equation:

$$\begin{bmatrix} \underline{S}_{bb} & \underline{S}_{bi} \\ \underline{S}_{ib} & \underline{S}_{ii} \end{bmatrix} \begin{Bmatrix} \underline{\delta}_b \\ \underline{\delta}_i \end{Bmatrix} = \begin{Bmatrix} \underline{P}_b \\ \underline{P}_i \end{Bmatrix} \tag{4.2}$$

where subscripts i and b represent internal and boundary values, respectively.

Since the equilibrium equations for the condensed nodes of a substructure only contain the dof's of the substructure itself, the internal nodal displacements can be expressed in terms of the boundary ones. This is achieved by solving the lower partition of the eq. 4.2 for $\underline{\delta}_i$ as follows:

$$\underline{\delta}_i = \underline{S}_{ii}^{-1} \left(\underline{P}_i - \underline{S}_{ib}\underline{\delta}_b\right) \tag{4.3}$$

Substituting $\underline{\delta}_i$ from eq. 4.3 into the upper partition of eq. 4.2 gives:

$$\underline{S}^* \underline{\delta}_b = \underline{P}^* \tag{4.4}$$

in which:

$$\underline{S}^* = \underline{S}_{bb} - \underline{S}_{bi} \underline{S}_{ii}^{-1} \underline{S}_{ib}$$
$$\underline{P}^* = \underline{P}_b - \underline{S}_{bi} \underline{S}_{ii}^{-1} \underline{P}_i$$

$\underline{S}^*$ and $\underline{P}^*$ are the boundary stiffness matrix and load vector of the substructure after static condensation. The condensed substructure can now be treated like any other element, and $\underline{S}^*$ and $\underline{P}^*$ are assembled into the whole structure in the usual way. After solving for $\underline{\delta}_b$ from the structural equilibrium equations, $\underline{\delta}_i$ can be obtained by a recovery process (eq. 4.3). $\underline{\delta}_i$ is needed for the calculation of internal forces.

4.3 Caisson Substructures

4.3.1 General

A caisson structure consists of one or more layer substructures in height. The first layer includes both vertical panels and horizontal base panel(s) at its bottom. Whereas the next possible layers only have vertical panels in longitudinal and transverse directions (Fig. 4.1).

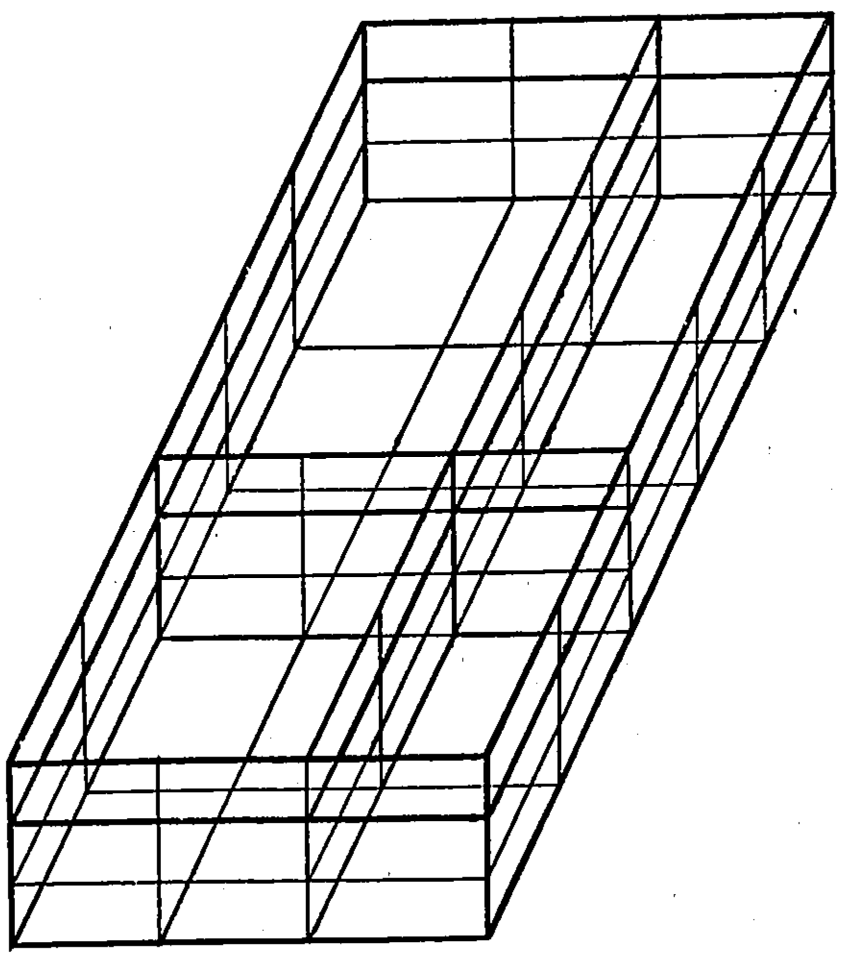


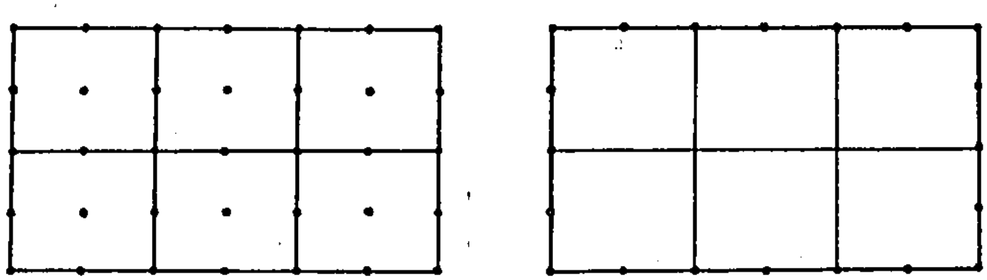
Fig. 4.1 A typical caisson with two layer substructures

4.3.2 Panel Superelement

Each panel is modeled as a regular mesh. Material properties and thickness could be different in different panels. Also, panel span and number of elements in horizontal direction could be different in every row of longitudinal or transverse panels. But within an individual panel, material properties and element sizes are constant.

A panel is a substructure which is made up of one or more flat shell elements. Assemblage of each panel has two steps. First, all of its individual elements are assembled together. Then, the panel is condensed to its boundary nodes. This condensed panel is called "panel superelement". The assemblage process is performed simultaneously and similarly for both of the stiffness matrix and load vector of the panel.

After the boundary nodal displacements of the panel are computed, the internal ones are determined by a recovery process as explained in section 4.2. Then the internal forces of individual elements, including bending forces and membrane stresses, can be found for each panel.



Before condensation

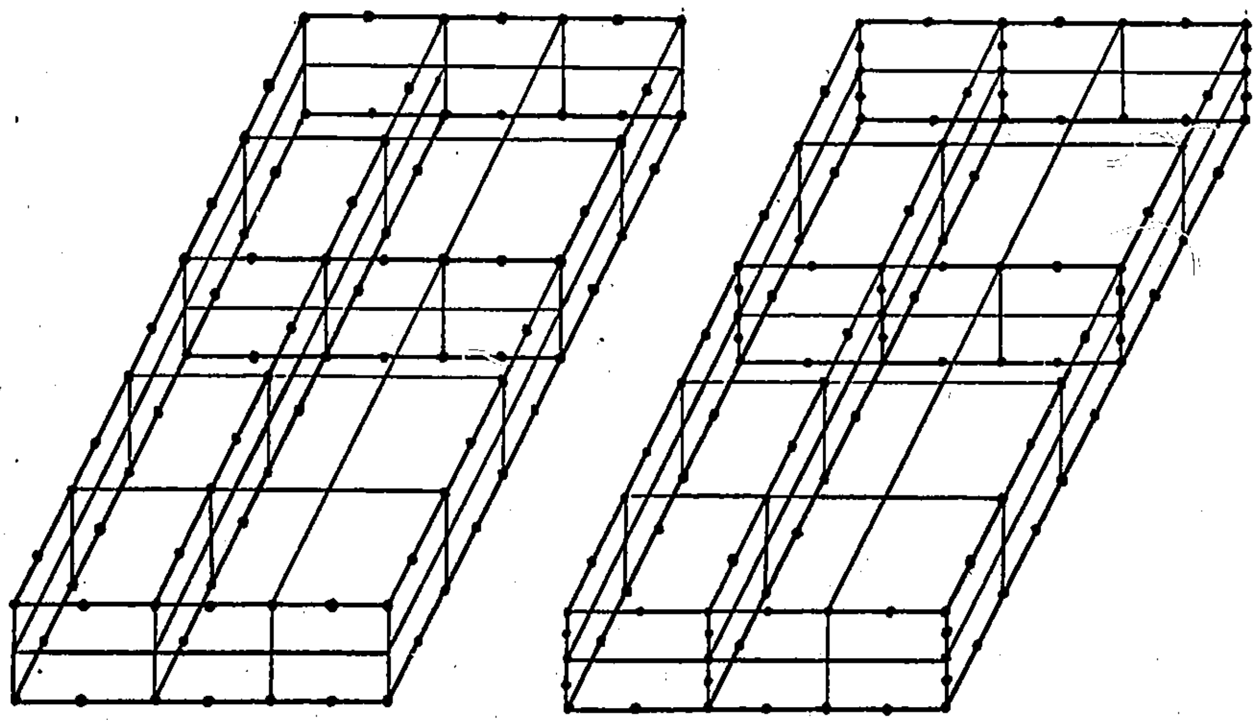
After condensation
(panel superelement)

Fig. 4.2 A panel substructure before and after condensation

4.3.3 Layer Superelement

Material properties, thickness of panels, height and number of elements in height could be different in different layers. But the number of longitudinal and transverse panels, and number of their elements in horizontal direction are the same for all layers. The first layer substructure consists of at least four vertical (two longitudinal and two transverse) panels and one base panel.

The assemblage steps of a layer substructure are quite similar to those of a panel one. First, all panel superelements are assembled together. Then, the layer is condensed to its boundary nodes at its top and bottom levels. This condensed layer is called **"layer superelement"**. The same assemblage process is performed for both of the stiffness matrix and load vector of the layer substructure.



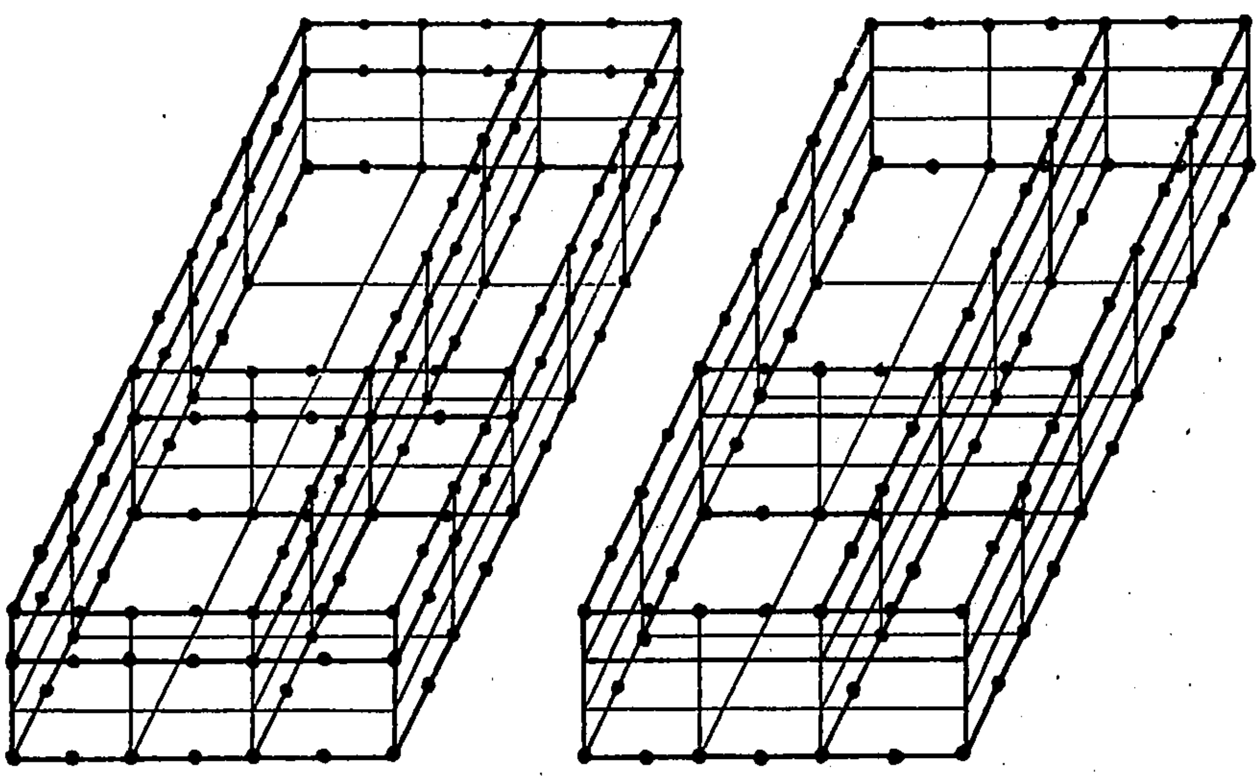
Before condensation **After condensation (layer superelement)**

Fig. 4.3 A layer substructure before and after condensation

After the boundary nodal displacements of the layer are found, its internal ones are determined by the recovery process. These boundary and internal displacements altogether make the boundary nodal displacements of all the panel superelements of the layer substructure.

4.3.4 Caisson Superelement

At the final stage of caisson assemblage, all the layer superelements are assembled together and condensed to make "**caisson superelement**". This caisson superelement has dof's only at its base and deck levels. Both the stiffness matrix and load vector of the entire caisson structure are assembled in a similarl way.



Before condensation after condensation

(caisson superelement)

Fig. 4.4 A caisson structure before and after condensation

Solving equilibrium equation 4.4 for the caisson superelement gives the boundary nodal displacements of the caisson structure. Then, the internal ones are found by the recovery process. These boundary and internal displacements altogether make the boundary nodal displacements of all the layer superelements of the caisson structure.

4.3.5 Foundation Modelling

The caisson structure is supported by soil which can be modeled as an elastic support. Elastic support can be represented by a foundation stiffness matrix, $\underline{k}_{es}$ for a foundation element and $\underline{K}_{es}=\Sigma\, \underline{k}_{es}$ for the entire foundation model [36]. These foundation elements have the same shape and size as those of the base panel(s) of the caisson structure.

A Winkler idealization [36] for the foundation modelling gives a $\underline{k}_{es}$ as follows:

$$k_{es} = \int k\, \underline{N}^T\, \underline{N}\, dA \qquad (4.5)$$

where:

k = subgrade modulus

$\underline{N}$ = shape function of the element

A = area of the element

For our special case of a nine node rectangular Lagrangian element, we have:

$$k_{es}=\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{19} \\ k_{21} & k_{22} & \cdots & k_{29} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ k_{91} & k_{92} & \cdots & k_{99} \end{bmatrix} = [K_{ij}] \quad ,\begin{cases} i=1,2,\ldots,9 \\ j=1,2,\ldots,9 \end{cases} \qquad (4.6)$$

where;

$\underline{k}_{ij} = \underline{0}$ (for $i \neq j$),

$$\underline{k}_{ij} = \alpha_{ij} \begin{bmatrix} k_x & 0 & 0 & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 & 0 & 0 \\ 0 & 0 & k_z & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{zz} \end{bmatrix} \quad (\textit{for} \quad i=j) \ ,$$

and

$$\begin{cases} \alpha_{ij} = \dfrac{A}{36} & \textit{for} \quad i=j=1,3,5,7 \quad (\textit{corner} \quad \textit{nodes}) \\ \alpha_{ij} = \dfrac{A}{9} & \textit{for} \quad i=j=2,4,6,8 \quad (\textit{midside} \quad \textit{nodes}) \quad (\textit{section} \quad 3.4.3) \\ \alpha_{ij} = \dfrac{4A}{9} & \textit{for} \quad i=j=9 \quad (\textit{center} \quad \textit{node}) \end{cases}$$

$$\begin{cases} k_x = k_y = k_z = k = \textit{subgrade} \quad \textit{modulus} \quad (\textit{section} \quad 2.3) \\ k_{xx} = k_{yy} = k_{zz} = 0 \end{cases}$$

All the base panel elements are assigned a foundation stiffness $\underline{k}_{es}$ when assembling these elements.

4.4 Computer Program Outline

The computer program performs the following main steps:

1. Input data including geometry and material properties.
2. Process the input data to generate the desired mesh.
3. Input loading data including water and soil levels of all compartments and point load(s) at each load case.

4. Process the input loading to generate the load matrices: $\underline{P}_e$ of elements; and $\underline{P}_c$ of the entire caisson structure.
5. Develop and condense load and stiffness matrices in panel and layer substructural levels.
6. Sum the load and stiffness matrices of the layer superelement, $\underline{P}_l^*$ and $\underline{S}_l^*$, into the entire caisson load and stiffness matrices $\underline{P}_c$ and $\underline{S}_c$.
7. Condense the internal dof's of both $\underline{P}_c$ and $\underline{S}_c$ to yield to the caisson superelement load and stiffness matrices $\underline{P}_c^*$ and $\underline{S}_c^*$.
8. Solve equilibrium equations for the caisson superelement to find the boundary nodal displacements of the entire caisson structure $\underline{\delta}_{cb}$.
9. Recover internal nodal displacements for; the entire caisson structure $\underline{\delta}_{ci}$; and then for the layer and panel substructures $\underline{\delta}_{li}$ and $\underline{\delta}_{pi}$.
10. Write nodal displacements and base reactions.
11. Compute the internal forces of individual elements including bending forces and membrane stresses.

A flow chart of the main program is given in the next page. Contribution of each subroutine of the main program from the above steps is shown below.

Subroutine	**Steps**
GEOMET	1 and 2
LOADS	3 and 4
GSTFNS	5 and 6
SOLVEQ	7, 8 and 9
OUTDIS	10
OUTSTR	11

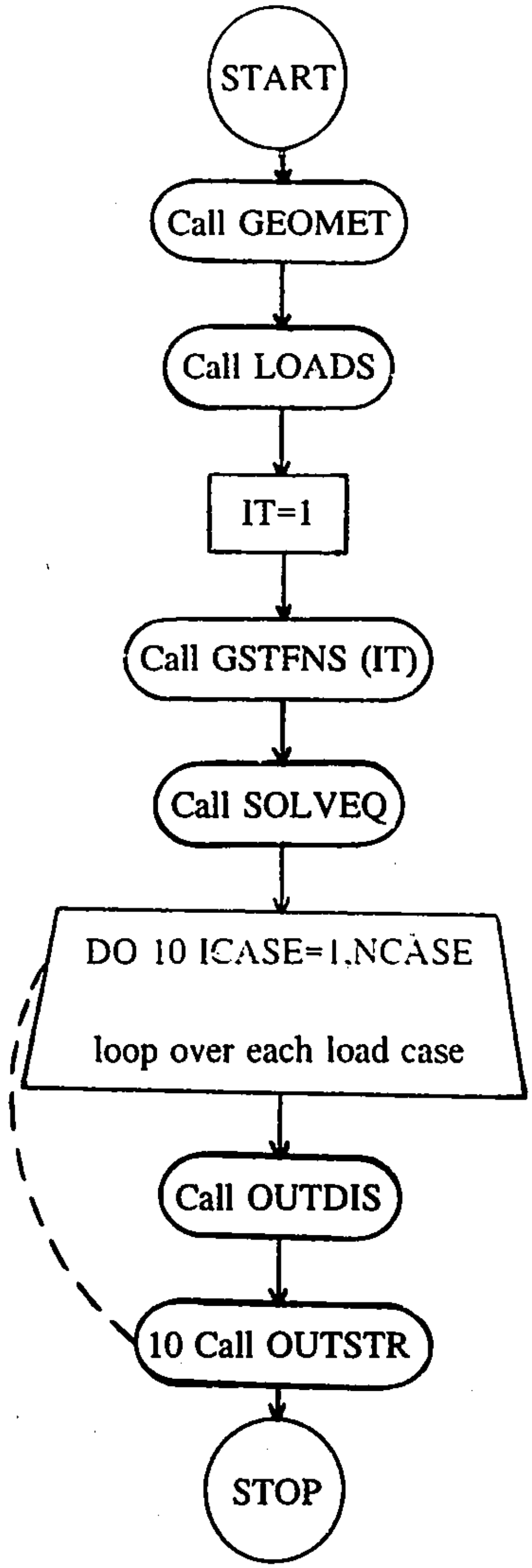


Fig 4.5 Flow chart of main program

Chapter 5

APPLICATION AND RESULTS

5.1 General

Several numerical examples are investigated in this chapter to show the practical application of the concepts presented in previous chapters. For convenience, the present program is called SAMFSS (Structural Analysis by Mindlin Flat Shell Superelements). All examples considered are in S.I. units. However, the program could be used in any other system of units provided that all values are consistent throughout. The SAMFSS program in FORTRAN is listed in Appendix C.

5.2 Comparison of Results

First, the results for plate bending and membrane behaviours are investigated and verified in two dimensional examples. Then a tank and two caisson structures are analyzed using the three dimensional model made up of flat shell elements.

Plate bending and membrane results are compared with analytical and closed form solutions given by others. Whereas the results of the 3-D examples are compared with those of another finite element program, SAP90, using 4-node flat shell elements.

5.2.1 Plate Bending Results

Two different examples are solved by SAMFSS. First a cantilever beam is modeled by the plate bending elements. Then a square plate under two different types of loading; i.e. a uniformly distributed load and a central point load, is analyzed. Because of the symmetry in geometry and loading, only a quarter of the square plate is considered.

5.2.1.1 Cantilever beam model

The beam characteristics are as follows:

length	$L = 4.0$	
width	$W = 1.0$	
thickness	$t = 0.1$	
modulus of elasticity	$E = 10^6$	
Poisson's ratio	$\nu = 0.$	
tip uniformly distributed load	$q = 1.0$	$(P = q\ w = 1.0)$

From the beam theory we have:

$$\text{tip deflection} \qquad w_t = \frac{PL^3}{3EI}$$

$$I = \frac{Wt^3}{12} = \frac{1.0(0.1)^3}{12} = 8.33\times10^{-5}$$

$$w_t = \frac{1.0(4.0)^3}{3(10)^6(8.33\times10^{-5})} = 0.256$$

$$\text{end moment} \qquad M = PL = 1.0(4.0) = 4.0$$

$$m = \frac{M}{W} = \frac{4.0}{1.0} = 4.0 \qquad \text{moment per unit width}$$

$$\text{end shear} \qquad V = P = 1.0$$

$$v = \frac{V}{W} = \frac{1.0}{1.0} = 1.0 \qquad \text{shear per unit width}$$

Using a coarse mesh made up of one element, the SAMFSS gives the following results:

$w_t = 0.25610$

$m = 4.0$

$v = 1.0$

which are similar to the values obtained by the beam theory.

5.2.1.2 Square plate model

Two different thickness ratios t/L = 0.01 and t/L = 0.1 are considered to represent a thin plate and a moderately thick plate, respectively. For a simply supported square plate under a UDL, a wide range of thickness ratios are also tested.

a) thin plate example

Two types of boundary conditions; 1) simply supported, and 2) clamped at all four edges, are considered.

The plate characteristics are as follows:

length	$L = 1.0$
width	$W = 1.0$
thickness	$t = 0.1$
modulus of elasticity	$E = 10.92 \times 10^6$
Poisson's ratio	$\nu = 0.3$
flexural rigidity	$D = E\,t^3/12(1-\nu^2) = 1.0$

For such a thin plate ($t/L = 0.01$), we can compare the results with those of the classic thin plate theory.

a.1) simply supported thin plate

Timoshenko [37] gives the following results for a simply supported square plate;

i) under a UDL, q=1;

center deflection	$w_o = 0.00406 \dfrac{qa^4}{D} = 0.00406$
center bending moment	$m_{ob} = 0.0479\, qa^2 = 0.0479$
corner torsional moment	$m_{ct} = -0.0325\, qa^2 = -0.0325$
edge center shear	$v_{eo} = -0.338\, qa = -0.338$

ii) under a central load, P=1;

center deflection $w_o = 0.01160 \dfrac{Pa^2}{D} = 0.01160$

center bending moment $m_{ob} = 0.298\ P = 0.298$

center torsional moment $m_{ct} = -0.061\ P = -0.061$

edge center shear $v_{co} = -0.417 \dfrac{P}{a} = -0.417$

where a is the size of square plate.

Also, Hinton [38] has given a closed form solution in terms of a Fourier series for a simply supported plate. He has included a program which gives the following results for the above example;

i) UDL

$w_o = 0.0040645$
$m_{ob} = 0.047886$
$m_{ct} = -0.032482$
$v_{co} = -0.33756$

ii) central point load

$w_o = 0.011648$
$m_{ob} = 1.0785$
$m_{ct} = -0.060953$
$v_{co} = -0.66731$

For a simply supported plate under a UDL, the above results are quite similar to those obtained by the classic thin plate theory. But under a central point load, the closed form solution gives a center bending moment and an edge center shear much higher than those of the thin plate theory. Whereas the results for the central displacement and corner torsional moment are again in good agreement.

It should be noted that in fact Timoshenko [37] gives the results for a square plate under a UDL over a very small area instead of a central point load. This area is a circle having a diameter of 0.1a. Running the program given by Hinton [38] for a patch load uniformly distributed over a small square area having the size of 0.0866L, which has equivalent area to a 0.1a diameter circle, leads to the following results:

$$W_o = 0.011427$$
$$m_{bb} = 0.29664$$
$$m_{tt} = -0.060703$$
$$v_{eo} = -0.41737$$

which are in good agreement with those given by Timoshenko [37].

Tables 5.1 and 5.2 compare the SAMFSS results with those of Timoshenko [37] and Hinton [38] for a simply supported thin plate under a UDL and a central load, respectively. In table 5.2, the star superscript for Hinton and 6x6 mesh results indicates a UDL over a small area instead of a central point load.

a.2) clamped thin plate

Again Timoshenko [37] gives the following results for a clamped square plate:

i)under a UDL, q=1;

central deflection $\quad w_o = 0.00126 \dfrac{q a^4}{D} = 0.00126$

center bending moment $m_{ob} = 0.0231 \; qa^2 = 0.0231$

edge center bending momnt $m_{eob} = -0.0513 \; qa^2 = -0.0513$

ii) under a central load, P=1;

$$w_o = 0.00560 \; \frac{Pa^2}{D} = 0.00560$$

$$m_{eob} = -0.1257 \; qa^2 = -0.1257$$

Tables 5.3 and 5.4 compare the above values with the SAMFSS results for a clamped thin plate under a UDL and a central load, respectively.

b) thick plate example

A simply supported thick plate under the two types of loading is analyzed. The plate characteristics are as follows:

length	L = 1.0
width	W = 1.0
thickness	t = 0.1
modulus of elasticity	E = 10920
Poisson's ratio	$\nu = 0.3$
flexural rigidity	$D = E\,t^3/12(1-\nu^2) = 1.0$

Applying the classic thin plate theory for this moderately thick plate, we expect the same results as the thin plate example since the flexural rigidity is equal for both cases.

Closed form solution [38] leads to the following results:

i) under a UDL, q=1;

$$w_o = 0.0042728$$
$$m_{bb} = 0.047886$$
$$m_{tt} = -0.032482$$
$$v_{eo} = -0.33756$$

ii) under a central load, P=1;

- for a central point load:

$$w_o = 0.016341$$
$$m_{bb} = 1.0785$$
$$m_{tt} = -0.060953$$
$$v_{eo} = -0.66731$$

- for an equivalent central UDL over a small square area:

$$w_o = 0.012718$$
$$m_{bb} = 0.29664$$
$$m_{tt} = -0.060703$$
$$v_{eo} = -0.41737$$

Comparing the above values with those of the thin plate example (t/L=0.01), we noticed that the central deflection is increased about 5% and 11% for a UDL and an equivalent central load, respectively. Whereas the internal forces (moments and shears) have not changed. Therefore, it can be concluded that for this moderately thick plate, the thin plate theory still gives acceptable results, especially for a UDL.

Tables 5.5 and 5.6 compare the SAMFSS results with the above values for a moderately thick plate under a UDL and a central load, respectively. Again in table 5.6, the star superscript for Hinton and 6x6 mesh results indicate a UDL over a small area instead of a central point load.

c) Simply supported plate (different thickness ratios)

The same square plate with different thicknesses but equal flexural rigidity (D=1) is tested under a UDL. The SAMFSS results for center displacement are compared with those of the closed form solution [38] in table 5.7 and diagram 5.1.

5.2.2 Membrane Results

Two membrane examples are investigated. First a cantilever deep beam model under an in-plane tip load is analyzed. Then a square panel under a uniform stretching load at one edge is considered.

5.2.2.1 Cantilever deep beam model

The beam characteristics are the same as the cantilever model for the plate bending example (5.2.1.1). The only difference is that the tip load P=1 is an in plane UDL. Again, from the beam theory, we have:

$$tip \quad deflection \qquad w_t = \frac{PL^3}{3EI}$$

$$I = \frac{t\,W^3}{12} = \frac{0.1(1)^3}{12} = 0.00833$$

$$w_t = \frac{1.0(4)^3}{3(10^6)(0.00833)} = 0.00256$$

end moment $M = PL = 1.0(4.0) = 4.0$

$$(\sigma_b)_{\max} = \frac{Mc}{I} = \frac{6M}{t\,W^2}$$

$$(\sigma_b)_{\max} = \frac{6(4.0)}{0.1(1)^2} = 240.0$$

Using a coarse mesh made up of one element leads to the following results:

$w_t = 0.00264$

$\sigma_{11} = 240.0$

which are similar to the theoretical values.

5.2.2.2 Square panel model

The SAMFSS is used to analyze a square panel fixed at one edge carrying an in-plane load uniformly distributed along the opposite edge. The panel characteristics are as follows:

length	$L = 1.0$
width	$W = 1.0$
thickness	$t = 0.1$
Poisson's ratio	$\nu = 0.$
modulus of elasticity	$E = 10^6$
UDL over edge	$q = 1$

We can easily analyze this simple example:

$$\sigma = \frac{P}{A} = \frac{qW}{Wt} = \frac{q}{t}$$

$$\sigma = \frac{1}{0.1} = 10.0$$

$$\epsilon = \frac{\sigma}{E} = \frac{10}{10^6} = 1 \times 10^{-5}$$

$$tip \quad elongation \qquad w_t = \epsilon L = 1 \times 10^{-5}(1.0) = 1 \times 10^{-5}$$

SAMFSS gives the following results for a coarse mesh made up of one element:

w_t = 0.00001

σ_{11} = 10.0

which is quite accurate.

5.2.3 A Cubic Tank

A cubic tank is analyzed under inside hydrostatic pressure. To evaluate the results of SAMFSS, a set of results obtained by SAP90 are presented. The tank characteristics are as follows:

length	$L = 3.0$
width	$W = 3.0$
height	$H = 3.0$
thickness	$t = 0.1$
modulus of elasticity	$E = 2.188 \times 10^9$
Poissons's ratio	$\nu = 0.15$

The tank is assumed to be seated on the ground. The underground soil is modeled as an elastic foundation. Soil and water data are as follows:

subgrade modulus (in three directions) $k = 10^6$

water specific gravity $\gamma_w = 1000$

head of water (inside the tank) 2.95

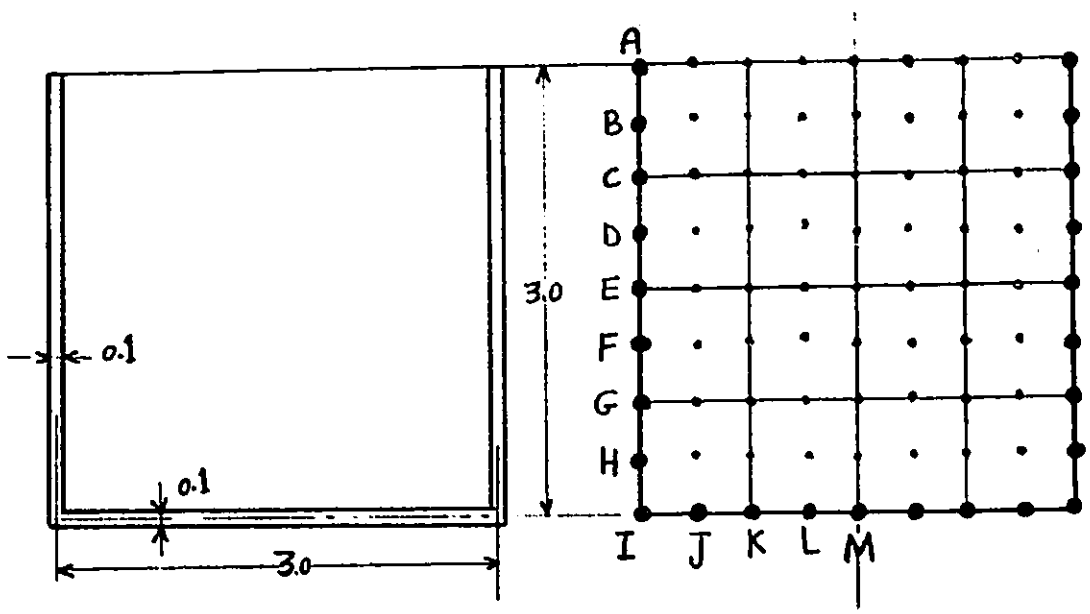


Fig. 5.1 The cubic tank

All four vertical walls and horizontal bottom of the tank are divided into the same mesh size. A 4×4 mesh of 9-node elements or a 8×8 mesh of 4-node elements is used for the SAMFSS and SAP90 models, respectively.

Table 5.8 and diagram 5.2 compare three sets of results at 13 points (from A to M) on midsection of the tank. These three sets are: 1) midsection in-plane displacements (δ_x , δ_z) , 2) bending moments parallel to the edge along the midsection (m^*_b) , and 3) axial forces normal to the midsection (f^*_a).

For further study of the convergence of the results, 2×2 and 3×3 mesh sizes are tested for the SAMFSS model compared with equivalent mesh sizes of 4×4 and 6×6 for the SAP90 model. Only the maximum values, which occur at point E (Fig. 5.1) at center of the wall, are compared in table 5.9 and diagram 5.3.

5.2.4 A Two Compartment Caisson

A caisson with two equal compartments (named Caisson 2C) is analyzed under three different load cases. The characteristics of Caisson 2C are as follows:

length	$L = 12.0$
width	$W = 6.0$
height	$h = 3.0$
thickness (all panels)	$t = 0.2$
Poisson's ratio	$\nu = 0.15$
modulus of elasticity	$E = 2.188 \times 10^9$
specific gravity	$\gamma = 2400$

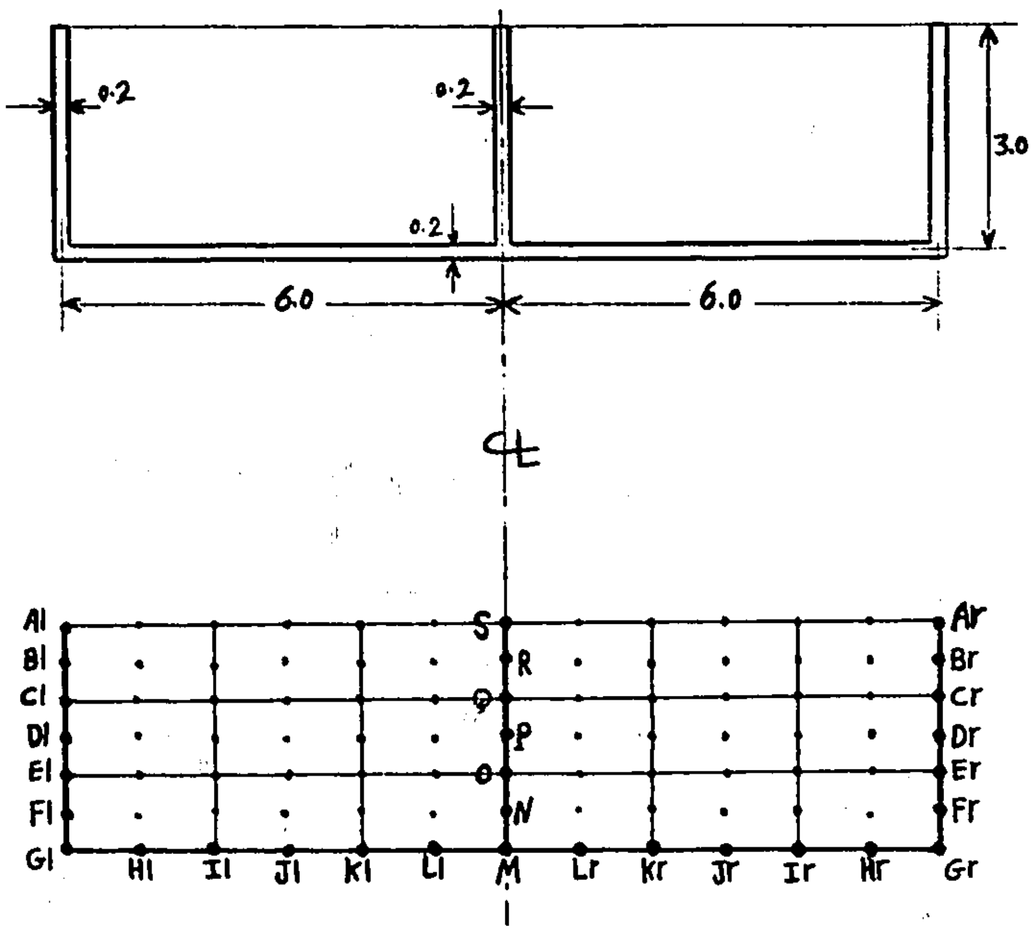


Fig. 5.2 A two-compartment caisson (Caisson 2C)

Caisson 2C is seating on the ground which is modeled as an elastic foundation. The same soil is considered for the base subgrade and backfills. The soil and water data are:

specific gravity of dry soil	$\gamma_{ds} = 2000$
specific gravity of wet soil	$\gamma_{ws} = 2500$
subgrade modulus (in three directions)	$k = 10^6$
water specific gravity	$\gamma_w = 1000$

All the heads of water and soil pressure are measured from the top of the base. Deadweight of structure is considered in all three load cases.

-load case 1

No earth pressure is considered in this load case.

head of water outside Caisson 2C 1.9

head of water inside two compartments 2.9

-load case 2

Two concentrated loads, each equal to 15000 acting at top of the middle transverse panel, are considered. There is no hydrostatic or earth pressure.

-load case 3

The left compartment is filled with soil whereas the right one contains water. Besides the hydrostatic pressure of water, there is a backfill just at the left side of the left compartment in longitudinal view.

head of water outside Caisson 2C 1.9

head of water inside the right compartment 1.9

head of soil inside the left compartment 2.9

head of soil outside Caisson 2C (only at left) ... 2.9

The same mesh size is selected for all the panels of Caisson 2C. A 3×3 mesh of 9 node elements and a 6×6 mesh of 4 node elements is used for the SAMFSS and SAP90 models, respectively.

The same set of results as for the cubic tank (5.2.3) is investigated along the longitudinal midsection. For the first and second load cases, because of the symmetric geometry and loading about the middle transverse panel, the results are given at 19 points from Al to S. But for the third load case, the results are presented along the whole longitudinal midsection consisting of 31 points from Al to Ar (Fig. 5.2).

Tables 5.10, 5.11, and 5.12 and diagrams 5.4, 5.5, and 5.6 compare the results of the two programs for Caisson 2C under the three considered load cases, respectively.

5.2.5 A Three Compartment Two Layer Caisson

Named Caisson 3C and consisting of two layers, this caisson includes three equal compartments. The base and vertical panels of the lower layer are thicker than the upper panels. Only one load case including the dead weight of structure and hydrostatic pressure is considered. Caisson 3C characteristics are as follows:

length	L = 12.0
width	W = 4.0
height	h = 3.0
thickness of lower layer panels	0.2
thickness of upper layer panels	0.1

The material characteristics of Caisson 3C for both layers, water specific gravity and subgrade modulus are assumed to be the same as those of the previous example (Caisson 2C, 5.2.4). Again head of water is measured from the top of the base.

head of water outside Caisson 3C 1.4
head of water inside the middle compartment ... 2.9
The left and right compartments are assumed to be empty.

Each base panel is divided into a 2×2 mesh of 9 node elements and a 4×4 mesh of 4 node elements for the SAMFSS and SAP90 models, respectively. But in vertical direction, each layer is divided into one 9 node element and two 4 node elements for the two models.

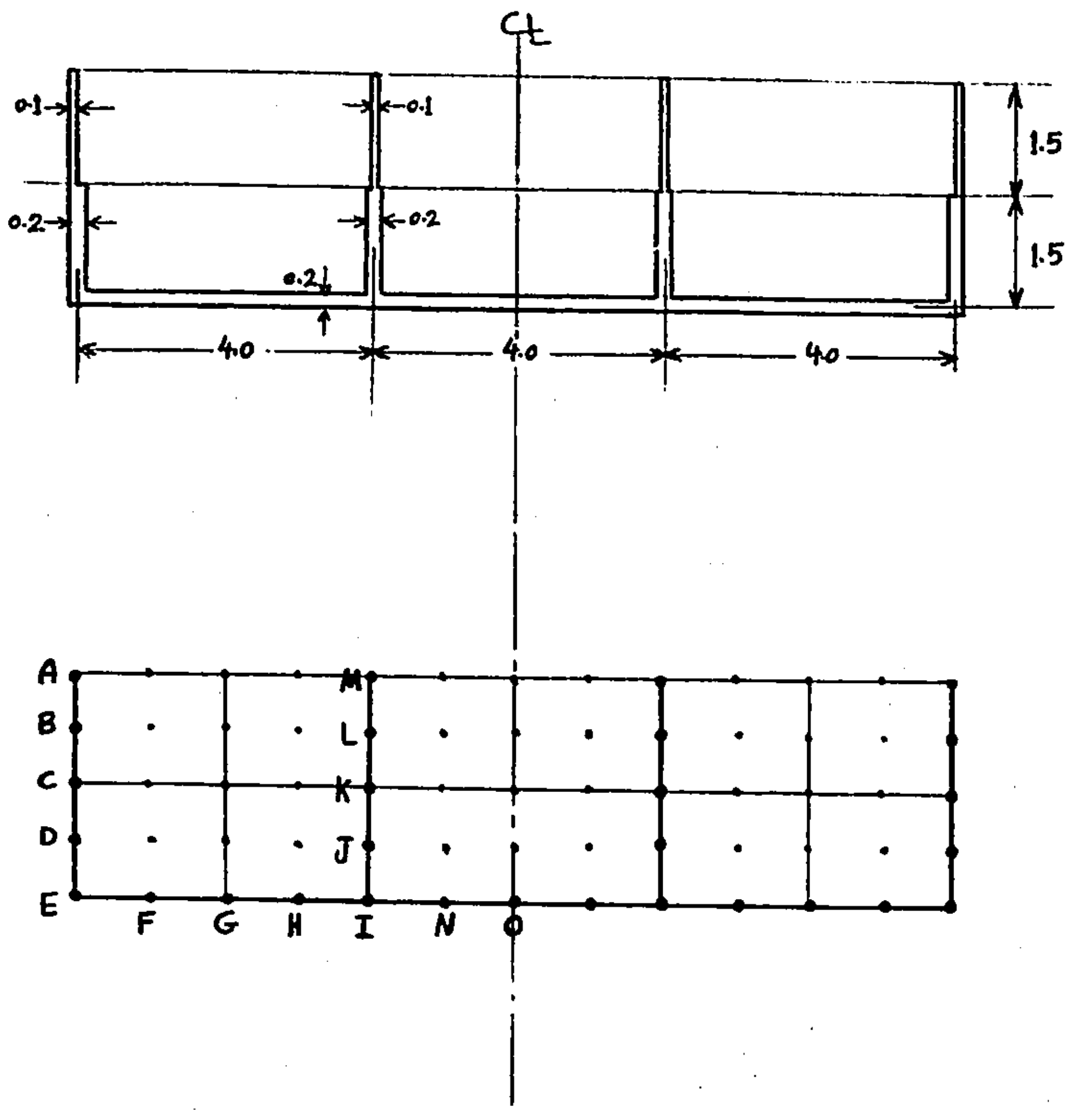


Fig. 5.3 A two-layer three-compartment caisson (Caisson 3C)

The same set of results, as for the previous example, is given at 15 points from A to O along the longitudinal midsection. Table 5.13 and diagram 5.7 compare the results of the two programs for Caisson 3C.

5.3 Convergence of the Results

This section investigates the convergence of presented results in the previous section. For the plate bending and membrane examples, it can be easily studied since their analytical or closed form solutions are available. But for the 3-D examples, the SAMFSS results are only compared with those of the SAP90 program using a specific mesh size for each example.

However, the mesh fineness of each 3-D example is selected differently, and it can help to assess the convergence of the results. A 4×4 mesh is used to model each panel of the cubic tank (5.2.3). Whereas the mesh size of panels for Caisson 2C is 3×3, and a 2×2 or 2×1 mesh is adopted to model the base or vertical panels of Caisson 3C, respectively (5.2.4 and 5.2.5).

It should be noted that all the comparisons and discussions for the 3-D examples are based on the longitudinal midsection results which could be a good sample of the behaviour of the whole structure.

5.3.1 The Plate Bending Results

Here the results of the simply supported examples are compared with those of the closed form solution [38]. Whereas the results of the clamped examples are compared with those of the thin plate theory [37].

a) Under a UDL

The results for a UDL converge rapidly. For a coarse 2×2 mesh, the error involved in center displacement is less than; 0.2% for the simply supported plate; and 1.5% for the clamped one. For the same mesh size, the maximum errors involved in moments and shear forces are about 4.2% and 16% for the two boundary conditions, respectively. These errors reduce to; 1.6% and 4.6% for a 4×4 mesh; and 0.9% and 2.1% for a 6×6 mesh (tables 5.1, 5.3, and 5.5).

In general, the results for a UDL converge faster for the simply supported plate than the clamped one. However, there is no significant difference in convergence rate between the thin and thick plate examples ($t/L=0.01$ and $t/L=0.1$). For the simply supported plate, the rapid convergence of the center displacement is shown for a wide range of thickness ratios in table 5.7 and digram 5.1.

b) Under a central point load

In contrast to the UDL case, under a central point load the results converge very slowly.

b.1) The simply supported plate

Using a 4×4 mesh leads to about 0.2% and 12% error in the center displacement of the thin and thick plate examples, respectively. These errors reduce to 0.1% and 9.3% for a 6×6 mesh (tables 5.2 and 5.6).

In contrast to the errors for w_o, the errors involved in internal forces are almost the same for the thin and thick plate examples. For a 6×6 mesh, they are about; 0.4% for the corner torsional moment m_{ct}; 65% for the edge center shear v_{eo}; and 161% for the center bending moment m_{ob}. However, using a UDL over a small area instead of a central point load, equivalent to what Timoshenko [37] utilized, leads to better results for practical applications. In this case, the errors for a 6×6 mesh are limited to about; .06% for m_{ct}; .1% for v_{eo}; and 3% for m_{ob}.

b.2) The clamped plate

Classic thin plate theory [37] gives a center displacement, w_o=0.00560 for the clamped plate example under a central load. The SAMFSS results converge to a higher value of about 0.00565. Besides w_o, Timoshenko [37] gives the edge center bending moment, m_{cob} = -0.1257. But he has not presented the center bending moment m_{ob}, or the edge center shear v_{eo}, for the clamped plate under a central load. The SAMFSS results for m_{cob} converge fairly well (table 5.4).

5.3.2 The Membrane Results

Both of the presented membrane examples (5.2.2) give some accurate results even for a coarse mesh made up of one element.

5.3.3 The Cubic Tank Results

Displacement results are in complete agreement for the two programs (table 5.8-a and diagram 5.2-a). Difference between the two output values is less than 0.8% in vertical displacements (in z direction), and 1.4% in horizontal displacements (in x or y direction). The maximum 1.4% difference in horizontal displacements occurs at top of the wall (point A of the midsection), and it reduces to 0.2% near the corner (point H).

For maximum m^*_b , the two programs results differ about; 7% at point E at the center of the vertical wall; and 6% at point J near the corner of the base. For maximum f^*_a, the difference between the two programs results is about; 5% at point E and 9% at point F (both on the wall); and 11% at point I_b at the corner of the base. The differences are a little more significant at point A at top of the vertical wall and at point I at the corner of the midsection (table 5.7-b and diagram 5.1-b). In general, the two outputs are in good agreement.

Regarding the convergence study of the maximum results (table 5.9 and diagram 5.3), it may be concluded that the SAMFSS program gives enough accuracy for design purposes provided a 4×4 mesh division is utilized under distributed loading.

5.3.4 The Results of Caisson 2C

In general, the differences between the results of the two programs are higher than those of the cubic tank example. This is due to the fact that a finer mesh is used to analyze the cubic tank.

Results of the two programs have better agreement for the first load case which includes hydrostatic pressure. For the second load case, which includes two concentrated loads, there are some significant differences between the two programs results even in horizontal displacements. Finally, for the third load case, which includes hydrostatic and earth pressure, differences between the two programs results are fairly significant. This is due to the different ways of applying the lateral earth pressure for the two programs. The SAMFSS program uses a quadratic (second order) interpolation to calculate the distributed loads, whereas a linear interpolation is used in hand calculation of input loading for the SAP90 model. Since the lateral earth pressure distribution is parabolic, quadratic interpolation evaluates it more accurately.

a) Load case 1

Maximum difference between the displacement results of the two programs is less than 1.5% (table 5.10-a and diagram 5.4).

For the maximum values of m^*_b , the two programs results differ about; 19% at point A1 at top of the external vertical panel; and 7% at point I1 of the base panel. For maximum f^*_a , they differ about; 3% at point Al; 7% at point S at top of the internal vertical panel; and 5% at point Gl_b at the corner end of the base panel (table 5.10-b and diagram 5.4).

b) Load case 2

Maximum difference between the two programs results is about 2% in vertical displacements and 19% in horizontal displacements (table 5.11-a and diagram 5.5).

For maximum m^*_b, they differ about; 3% at point Dl at the center of the external vertical panel; and 5% at point Jl at the center of the base panel. For maximum f^*_a, the difference is about; 64% and 23% at points Al and Gl_w at the top and bottom of the external vertical panel; 36% and 8% at points Gl_b and Ml_b at the two ends of the base panel; and 26% and 8% at points M_w and S at the bottom and top of the internal vertical panel (table 5.11-b and diagram 5.5).

c) Load case 3

At this load case, the results of the two programs have some significant differences, especially for the right compartment. As it was mentioned before, this is due to the fact that the input loading of the two programs are not exactly similar.

For the left compartment, i.e. from Al to M, the maximum difference of the two programs results is about; 4% in vertical displacements; and 15% in horizontal displacements. For the right compartment, the differences are much more significant (table 5.12-a and diagram 5.6).

For maximum m^*_b, the difference is about; 1% at point Fl of the left vertical panel; 12% at point Il of the left base panel; 30% at point S of the internal vertical panel; 15% at point Jr of the right base panel; and 54% at point Dr of the right vertical panel (table

5.12-b and diagram 5.6).

For maximum f^*_a, the difference is about; 11% at point Al of the left vertical panel; 27% and 31% at points Jl and Ml_b of the left base panel; 16% at point S of the internal vertical panel; 28% at point Mr_b of the right base panel; and 21% and 5% at points Gr_w and Dr of the right vertical panel.

5.3.5 The Results of Caisson 3C

Since the selected meshes are coarse (2×2 and 2×1 for the SAMFSS model), the differences between the two programs results are higher than the previous examples. These differences are more significant in horizontal displacements and bending moments compared with vertical displacements and axial forces, respectively.

Only for maximum values, the difference is about; 4% in horizontal displacements; and 5% in vertical displacements. Internal forces, specially bending moments, have more significant differences (table 5.13 and diagram 5.7). To obtain reliable results for design purposes, a finer mesh should be adopted.

mesh size	w_o	m_{ob}	m_{ct}	v_{co}
1	.0041960	.05933	-.03546	-.3634
2	.0040712	.04934	-.03385	-.3378
3	.0040657	.04843	-.03328	-.3373
4	.0040649	.04817	-.03301	-.3375
5	.0040646	.04806	-.03286	-.3377
6	.0040645	.04801	-.03277	-.3377
7	.0040645	.04797	-.03271	-.3377
Timoshenko	.00406	.0479	-.0325	-.338
Hinton	.0040645	.04789	-.03248	-.3376

Table 5.1
Simply supported thin plate under UDL (t/L = 0.01)

mesh size	w_o	m_{ob}	m_{ct}	v_{co}
1	.011693	.2368	-.08546	-.0162
2	.011602	.2992	-.06399	-.2808
3	.011617	.3414	-.06197	-.3589
4	.011624	.3715	-.06149	-.3858
5	.011628	.3948	-.06128	-.3979
6	.011631	.4138	-.06118	-.4041
7	.011634	.4298	-.06112	-.4076
Hinton	.011648	1.0785	-.06095	-.6673
6*	.011310	.2876	-.06074	-.4172
Hinton*	.011427	.2966	-.06070	-.4174
Timoshenko	.01160	.298	-.061	-.417

Superscript * indicates a UDL over a small area instead of a central point load.

Table 5.2
Simply supported thin plate under central load (t/L = 0.01)

mesh size	w_o	m_{ob}	m_{eob}	v_{eo}
1	.0015431	.04801	-.03693	-.5043
2	.0012786	.02534	-.04420	-.4371
3	.0012698	.02378	-.04547	-.4402
4	.0012685	.02335	-.04903	-.4411
5	.0012681	.02318	-.04979	-.4411
6	.0012680	.02310	-.05023	-.4410
7	.0012679	.02305	-.05050	-.4409
Timoshenko	.00126	.0231	-.0513	?

Table 5.3

Clamped thin plate under UDL (t/L = 0.01)

mesh size	w_o	m_{ob}	m_{eob}	v_{eo}
1	.0061809	.1923	-.1479	-.0206
2	.0056358	.2457	-.1203	-.4082
3	.0056333	.2882	-.1210	-.6474
4	.0056377	.3180	-.1222	-.7295
5	.0056413	.3412	-.1231	-.7601
6	.0056440	.3601	-.1238	-.7733
7	.0056460	.3761	-.1242	-.7795
Timoshenko	.00560	?	-.1257	?

Table 5.4

Clamped thin plate under central load (t/L = 0.01)

mesh size	w_o	m_{ob}	m_{ct}	v_{eo}
1	.0044083	.05949	-.03529	-.3658
2	.0042798	.04936	-.03376	-.3406
3	.0042742	.04843	-.03324	-.3383
4	.0042733	.04817	-.03299	-.3378
5	.0042730	.04806	-.03285	-.3377
6	.0042729	.04801	-.03276	-.3377
7	.0042729	.04797	-.03270	-.3377
Timoshenko	.00406	.0479	-.0325	-.338
Hinton	.0042728	.04789	-.03248	-.3376

Table 5.5
Simply supported moderately thick plate under UDL (t/L = 0.1)

mesh size	w_o	m_{ob}	m_{ct}	v_{eo}
1	.013390	.2370	-.08521	-.0129
2	.013932	.2994	-.06391	-.2798
3	.014312	.3416	-.06193	-.3588
4	.014579	.3717	-.06147	-.3857
5	.014784	.3950	-.06128	-.3975
6	.014951	.4140	-.06118	-.4037
7	.015092	.4300	-.06112	-.4074
Hinton	.016341	1.0785	-.06095	-.6673
6^*	.012517	.2880	-.06074	-.4169
Hinton^*	.012718	.2966	-.06070	-.4174
Timoshenko	.01160	.298	-.061	-.417

Superscript * indicates a UDL over a small area instead of a central point load.

Table 5.6
Simply supported moderately thick plate under central load (t/L = 0.1)

mesh size	t/L				
	.1	.01	.001	.0001	.00001
1	440827	419596	419384	419382	419382
2	427979	407117	406908	406906	406907
3	427415	406573	406364	406362	406362
4	427325	406485	406277	406275	406277
5	427301	406462	406253	406251	406281
6	427292	406453	406245	406243	406287
7	427289	406450	406242	406240	406302
Hinton	427284	406446	406237	406235	406235

Displacements are multiplied by 10^8

Table 5.7

Convergence of the center displacement of simply supported square plate under UDL for different thickness ratios

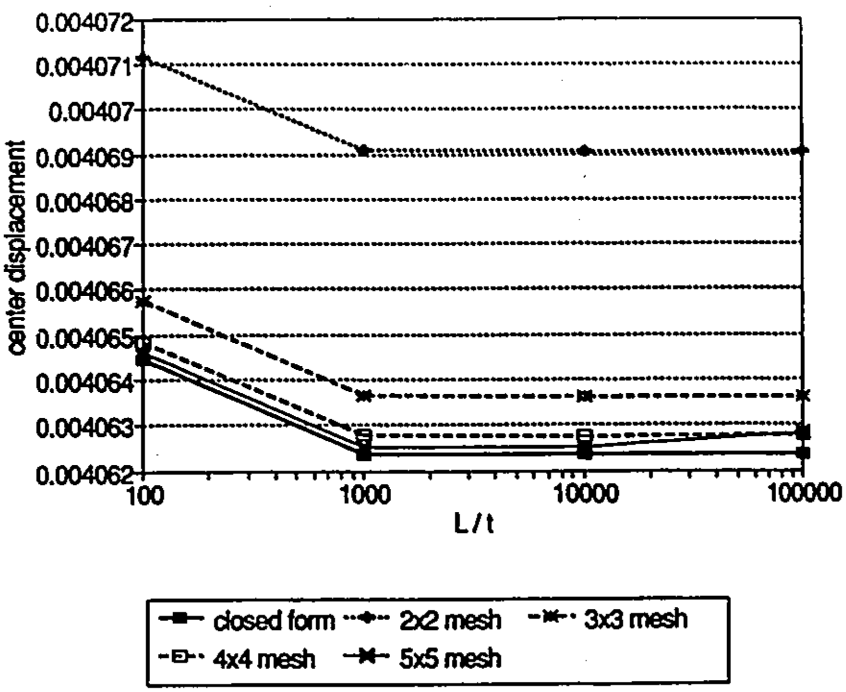


Diagram 5.1

point	δ_x		δ_z	
	SAMFSS	SAP90	SAMFSS	SAP90
A	-.000733	-.000723	-.003261	-.003238
B	-.000871	-.000864	-.003261	-.003239
C	-.001032	-.001030	-.003261	-.003239
D	-.001185	-.001186	-.003260	-.003238
E	-.001270	-.001272	-.003259	-.003237
F	-.001218	-.001219	-.003258	-.003236
G	-.000977	-.000971	-.003258	-.003235
H	-.000522	-.000523	-.003257	-.003234
I	-.000011	-.000012	-.003258	-.003234
J	-.000008	-.000008	-.002925	-.002916
K	-.000005	-.000005	-.002756	-.002752
L	-.000002	-.000002	-.002669	-.002673
M	0.0	0.0	-.002645	-.002649

Table 5.8-a

Midsection in-plane displacements of

the cubic tank (5.2.3)

point	m_y^*		f_a^*	
	SAFMSS	SAP90	SAFMSS	SAP90
A	298.4	272.9	652.6	517.7
B	331.2	302.3	910.0	903.5
C	379.8	351.2	1396	1389
D	428.8	401.1	1689	1823
E	457.5	428.2	2017	2115
F	428.2	407.8	1997	2171
G	343.4	319.0	1809	1927
H	154.4	156.5	1625	1517
I_w	-36.8	-52.2	1876	1424
I_b	51.4	54.1	1986	1792
J	88.8	83.7	1254	1051
K	82.4	80.3	1130	1112
L	73.6	72.2	1335	1398
M	71.0	68.7	1376	1528

Table 5.8-b

Midsection bending moments and axial forces of

the cubic tank (5.2.3)

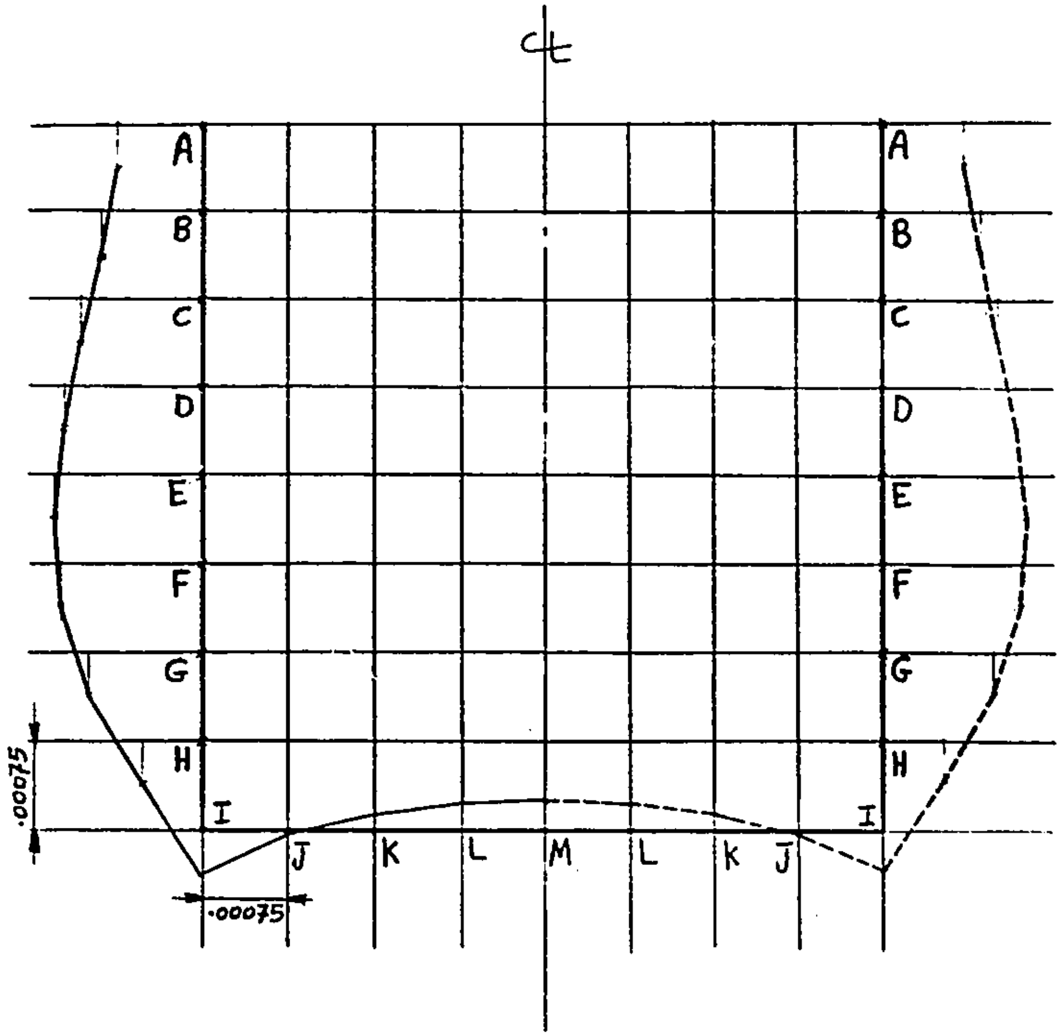


Diagram 5.2-a

Midsection in-plane displacements of the cubic tank (5.2.3)

——————— SAMFSS (left)

- - - - - - - SAP90 (right)

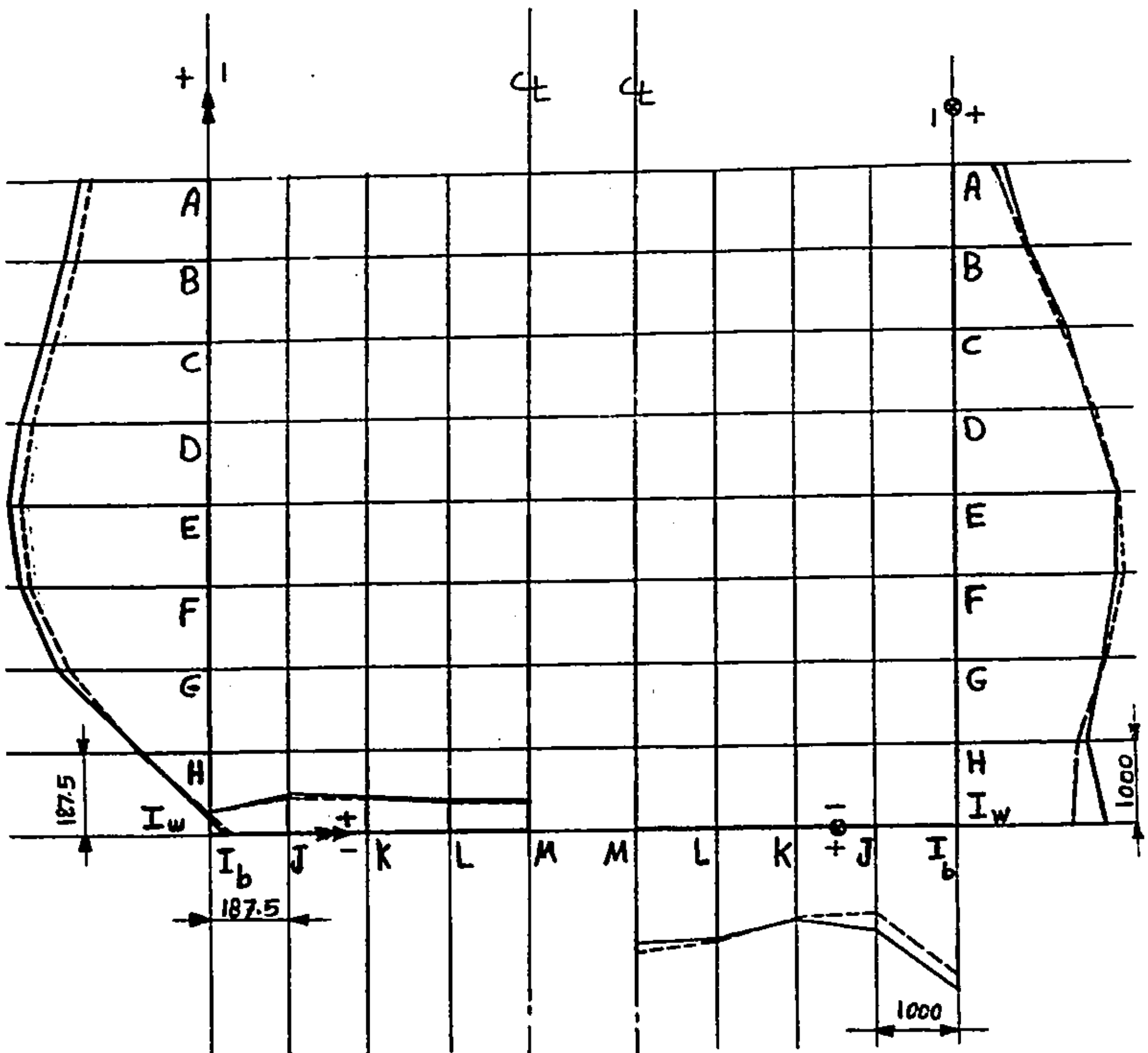


Diagram 5.2-b

Midsection bending moments (left) and axial forces (right) of

the cubic tank (5.2.3)

——————— SAMFSS

– – – – – – SAP90

mesh size	output values					
	δ_x		m^*_b		f^*_a	
	SAMFSS	SAP90	SAMFSS	SAP90	SAMFSS	SAP90
2	-.001370	-.001332	710.6	482.3	3610	2200
3	-.001265	-.001288	376.5	442.0	2087	2145
4	-.001270	-.001272	457.5	428.2	2017	2115

Table 5.9

Convergence of the results at center of the wall (point E) of the cubic tank

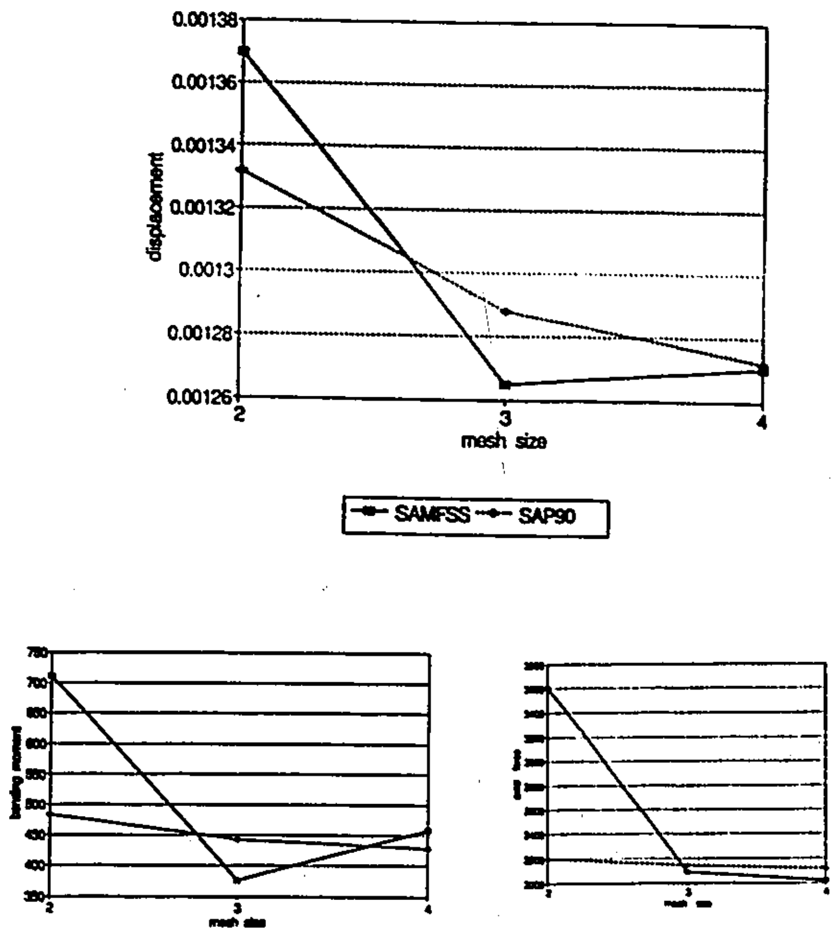


Diagram 5.3

point	δ_x		δ_z	
	SAMFSS	SAP90	SAMFSS	SAP90
Al	-.001536	-.001535	-.002699	-.002669
Bl	-.001398	-.001399	-.002698	-.002668
Cl	-.001250	-.001248	-.002697	-.002667
Dl	-.001050	-.001045	-.002697	-.002666
El	-.000775	-.000767	-.002695	-.002665
Fl	-.000417	-.000412	-.002694	-.002663
Gl	-.000007	-.000008	-.002691	-.002661
Hl	-.000004	-.000005	-.001935	-.001920
Il	-.000002	-.000002	-.001468	-.001479
Jl	-.000002	-.000002	-.001389	-.001395
Kl	-.000002	-.000002	-.001639	-.001645
Ll	-.000001	-.000001	-.002184	-.002140
M	0.0	0.0	-.002532	-.002495
N	0.0	0.0	-.002537	-.002500
O	0.0	0.0	-.002541	-.002504
P	0.0	0.0	-.002545	-.002507
Q	0.0	0.0	-.002547	-.002510
R	0.0	0.0	-.002550	-.002513
S	0.0	0.0	-.002552	-.002515

Table 5.10-a

Midsection in-plane displacements of Caisson 2C

at load case 1

point	m_b^*		f_a^*	
	SAMFSS	SAP90	SAMFSS	SAP90
Al	812.6	966.9	2804	2898
Bl	780.9	887.8	2454	2626
Cl	688.7	806.5	2235	2293
Dl	598.7	677.7	1995	2047
El	434.3	493.2	1661	1642
Fl	223.2	254.4	1337	1243
Gl_w	6.7	-2.7	843.8	1086
Gl_b	27.2	16.8	1622	1550
Hl	308.9	318.9	663.8	623.1
Il	464.2	433.3	748.4	853.3
Jl	461.9	429.6	770.2	832.5
Kl	356.2	314.9	401.5	395.0
Ll	69.0	90.5	-292.0	-485.5
Ml_b	-200.0	-206.4	-1060	-1409
M_w	0.0	0.0	-1982	-2137
N	0.0	0.0	451.8	234.8
O	0.0	0.0	2535	2495
P	0.0	0.0	4452	4602
Q	0.0	0.0	6292	6601
R	0.0	0.0	8612	9071
S	0.0	0.0	11158	11957

Table 5.10-b

Midsection bending moments and axial forces of Caisson 2C

at load case 1

point	δ_x		δ_z	
	SAMFSS	SAP90	SAMFSS	SAP90
Al	-.000368	-.000309	-.002622	-.002574
Bl	-.000383	-.000331	-.002620	-.002574
Cl	-.000401	-.000353	-.002620	-.002572
Dl	-.000403	-.000362	-.002618	-.002570
El	-.000367	-.000333	-.002616	-.002568
Fl	-.000251	-.000230	-.002614	-.002565
Gl	-.000005	-.000004	-.002611	-.002562
Hl	-.000004	-.000004	-.001777	-.001751
I!	-.000004	-.000003	-.001087	-.001111
Jl	-.000002	-.000002	-.000961	-.000972
Kl	-.000001	0.0	-.001360	-.001384
Ll	-.000002	-.000002	-.002286	-.002245
M	0.0	0.0	-.002901	-.002894
N	0.0	0.0	-.002916	-.002906
O	0.0	0.0	-.002925	-.002915
P	0.0	0.0	-.002934	-.002923
Q	0.0	0.0	-.002941	-.002930
R	0.0	0.0	-.002946	-.002935
S	0.0	0.0	-.002947	-.002934

Table 5.11-a

Midsection in-plane displacements of Caisson 2C

at load case 2

point	m_b^*		f_a^*	
	SAMFSS	SAP90	SAMFSS	SAP90
Al	248.3	262.8	2622	4303
Bl	249.3	261.5	2248	2830
Cl	250.2	266.3	1241	1742
Dl	259.8	267.7	651.2	769.9
El	245.7	256.3	-323.4	-370.1
Fl	217.1	221.3	-1395	-1834
Gl_w	174.0	159.4	-2916	-3597
Gl_b	-110.3	-149.3	-2328	-3176
Hl	341.4	393.8	-1528	-1787
Il	748.7	715.5	-797.5	-874.2
Jl	753.9	790.2	-179.1	-18.6
Kl	675.2	596.6	1248	1143
Ll	37.6	125.4	2744	3190
Ml_b	-372.4	-682.7	7200	7746
M_w	0.0	0.0	5010	6312
N	0.0	0.0	2942	2865
O	0.0	0.0	979.5	742.8
P	0.0	0.0	-1446	-1982
Q	0.0	0.0	-5193	-5625
R	0.0	0.0	-10058	-11283
S	0.0	0.0	-21460	-23209

Table 5.11-b

Midsection bending moments and axial forces of Caisson 2C

at load case 2

point	δ_x		δ_z	
	SAMFSS	SAP90	SAMFSS	SAP90
Al	-.003026	-.002712	-.012561	-.012092
Bl	-.002628	-.002355	-.012561	-.012091
Cl	-.002235	-.002004	-.012558	-.012089
Dl	-.001823	-.001630	-.012557	-.012086
El	-.001311	-.001169	-.012552	-.012081
Fl	-.000582	-.000505	-.012546	-.012074
Gl	.000570	.000543	-.012534	-.012064
Hl	.000563	.000537	-.009074	-.008794
Il	.000556	.000531	-.006118	-.006121
Jl	.000548	.000524	-.004907	-.004919
Kl	.000541	.000518	-.004875	-.004905
Ll	.000536	.000514	-.005534	-.005507
M	.000535	.000514	-.005645	-.005680
N	.000248	.000273	-.005658	-.005691
O	-.000017	.000058	-.005666	-.005699
P	-.000368	-.000233	-.005674	-.005706
Q	-.000803	-.000611	-.005678	-.005710
R	-.001290	-.001046	-.005683	-.005713
S	-.001790	-.001497	-.005685	-.005716
Lr	.000526	.000507	-.003980	-.003967
Kr	.000518	.000500	-.001957	-.002042
Jr	.000513	.000495	-.000875	-.000920
Ir	.000508	.000491	-.000347	-.000444
Hr	.000505	.000488	.000009	-.000212
Gr	.000503	.000487	.000550	.000164
Fr	.000090	.000144	.000547	.000162
Er	-.000389	-.000283	.000548	.000161
Dr	-.000907	-.000759	.000546	.000161
Cr	-.001444	-.001258	.000547	.000160
Br	-.001987	-.001766	.000546	.000160
Ar	-.002530	-.002273	.000547	.000160

Table 5.12-a

Midsection in-plane displacements of Caisson 2C

at load case 3

point	m^*_b		f^*_a	
	SAMFSS	SAP90	SAMFSS	SAP90
Al	448.0	509.7	4652	5163
Bl	484.1	541.4	3818	4471
Cl	519.9	593.4	3888	4128
Dl	602.8	637.3	3688	3698
El	596.4	653.5	3168	2761
Fl	625.1	617.7	2136	1376
Gl_w	600.3	561.8	429.4	278.6
Gl_b	-381.2	-534.7	2014	1277
Hl	841.4	932.8	1684	1496
Il	1797	1599	2263	2573
Jl	1690	1606	2308	2938
Kl	1201	1127	1506	2108
Ll	328.1	359.7	-203.6	1175
Ml_b	-472.5	-477.9	-2482	-1898
M_w	169.8	217.2	-4546	-3383
N	-69.8	-128.9	-222.8	71.4
O	-333.8	-390.2	3105	3041
P	-464.7	-569.5	5816	5466
Q	-543.0	-681.7	8408	7635
R	-603.8	-745.8	11760	10360
S	-622.3	-810.5	16070	13910
Mr_b	-562.0	-698.6	-3090	-2411
Lr	97.4	245.8	-2348	-2040
Kr	799.2	786.8	-1986	-1706
Jr	742.3	850.1	-1219	-1232
Ir	478.3	596.7	-512.9	-562.8
Hr	164.5	222.7	827.8	481.1
Gr_b	-107.4	-121.4	2402	2214
Gr_w	-93.1	-121.8	1797	2166
Fr	-121.4	-162.3	1746	1507
Er	-120.6	-181.8	973.0	928.8
Dr	-126.5	-185.6	325.8	351.5
Cr	-114.8	-181.1	-393.4	-313.0
Br	-114.4	-174.6	-1305	-1088
Ar	-106.8	-172.9	-1944	-2038

table 5.12-b
Midsection bending moments and axial forces of
Caisson 2C at load case 3

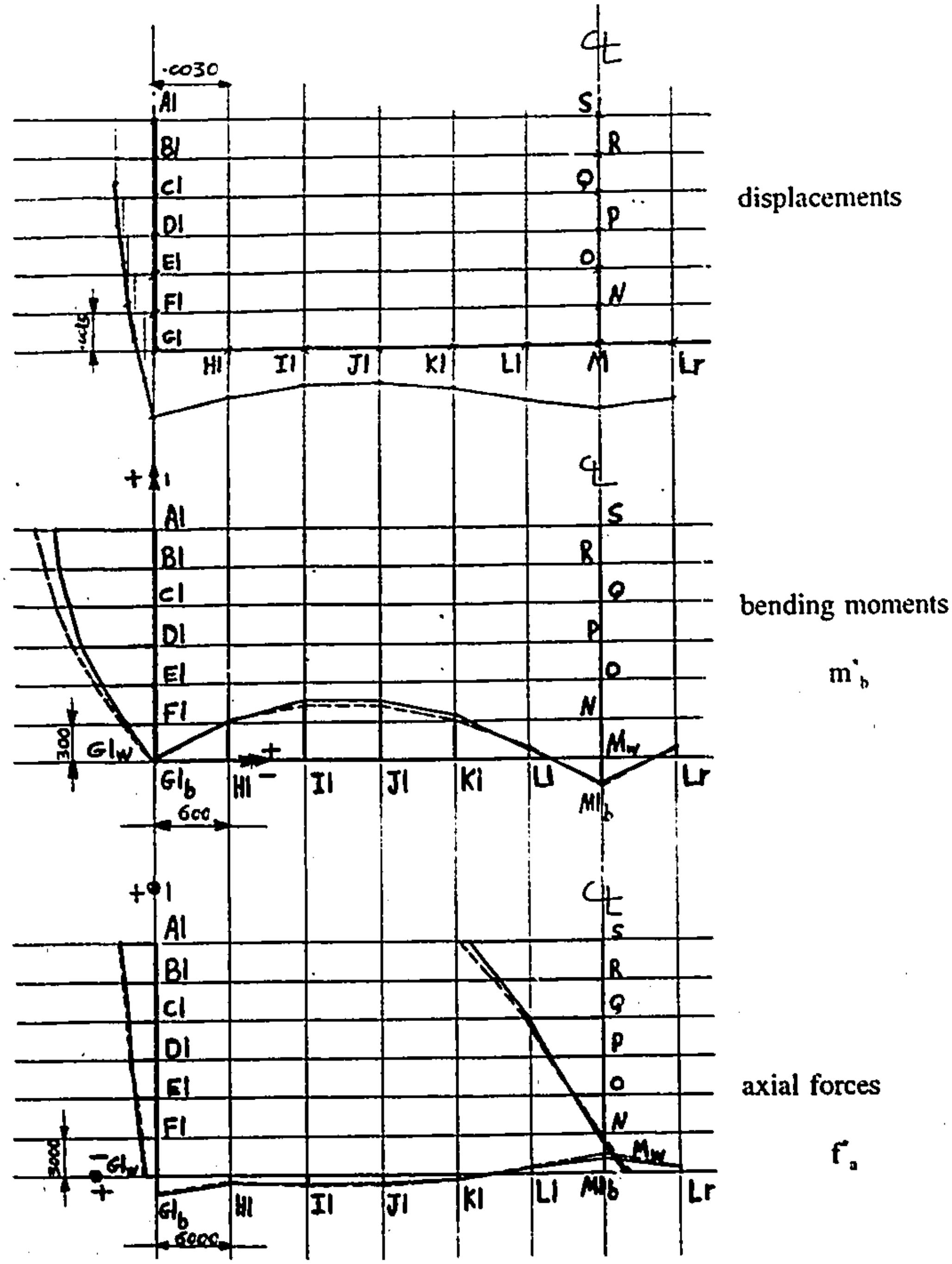


Diagram 5.4

Midsection results of Caisson 2C (5.2.4) at load case 1

——————— SAMFSS

- - - - - - SAP90

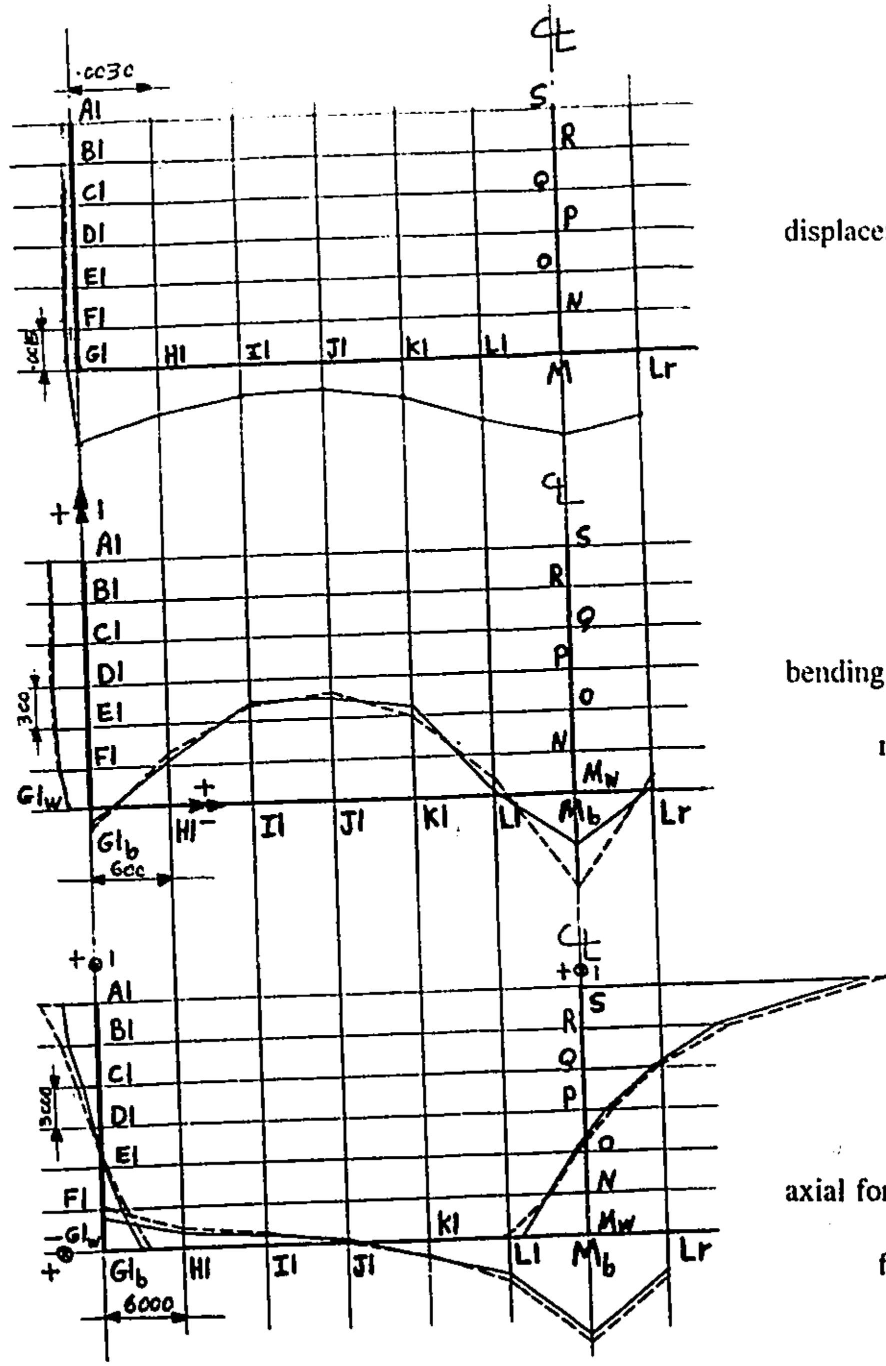


Diagram 5.5

Midsection results of Caisson 2C (5.2.4) at load case 2

——————— SAMFSS

- - - - - - - SAP90

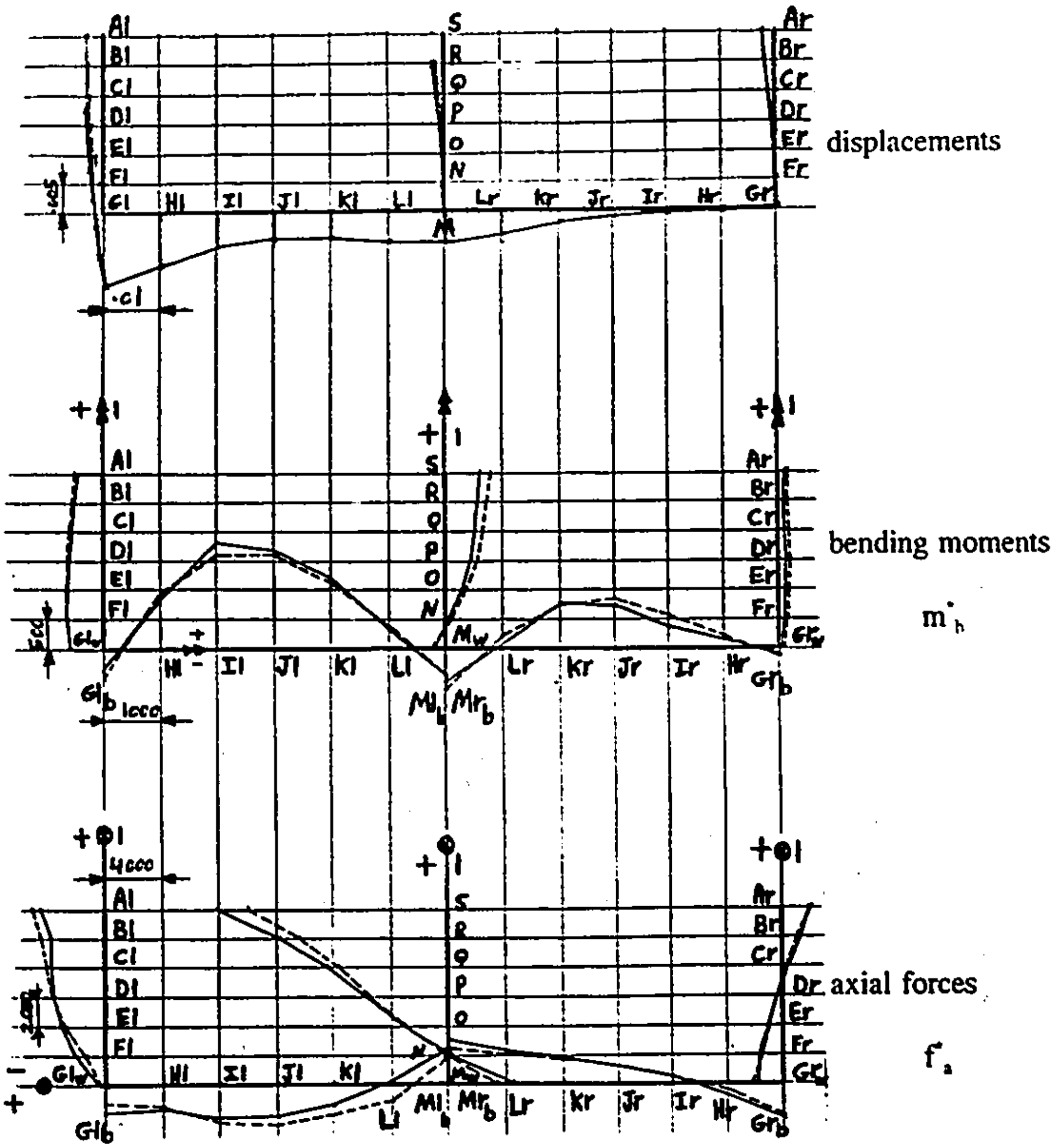


Diagram 5.6

Midsection results of Caisson 2C (5.2.4) at load case 3

——————— SAMFSS

– – – – – – SAP90

point	δ_x		δ_z	
	SAMFSS	SAP90	SAMFSS	SAP90
A	.000034	.000065	-.000779	-.000770
B	.000006	.000038	-.000777	-.000769
C	-.000017	.000002	-.000775	-.000766
D	-.000060	-.000040	-.000771	-.000763
E	-.000011	-.000012	-.000769	-.000759
F	-.000013	-.000014	-.000440	-.000438
G	-.000014	-.000015	-.000257	-.000305
H	-.000015	-.000015	-.000559	-.000552
I	-.000015	-.000015	-.000878	-.000872
J	-.000234	-.000247	-.000880	-.000876
K	-.000677	-.000660	-.000883	-.000879
L	-.001102	-.001127	-.000885	-.000882
M	-.001389	-.001441	-.000888	-.000884
N	-.000008	-.000007	-.000999	-.000981
O	0.0	0.0	-.001089	-.001042

Table 5.13-a

Midsection in-plane displacements of

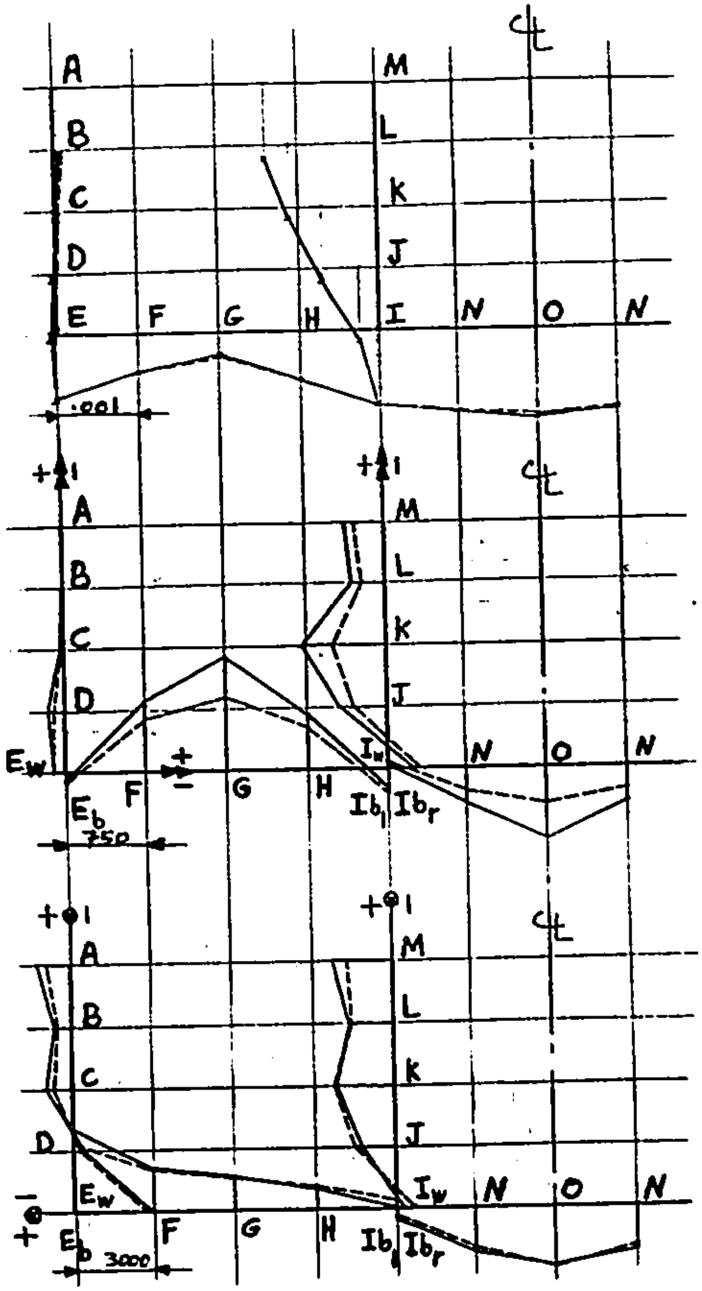
Caisson 3C (5.2.5)

point	m^*_b		f^*_a	
	SAMFSS	SAP90	SAMFSS	SAP90
A	4.6	-2.2	1142	829.5
B	4.3	0.0	555.9	474.6
C	24.6	22.8	898.4	633.2
D	141.7	83.0	-414.4	-292.4
E_w	111.3	117.9	-2876	-2707
E_b	-80.3	-112.1	-2988	-2472
F	622.3	458.8	-1586	-1459
G	1038	677.2	-1311	-1247
H	489.4	374.4	-767.0	-837.9
I_{bl}	-161.5	-218.4	-74.9	-345.1
I_w	-233.8	-287.5	-238.8	-658.7
J	443.1	308.8	1348	1578
K	802.1	501.8	2248	2267
L	324.3	222.0	1670	1632
M	388.7	291.2	2245	1751
I_{br}	42.3	72.5	365.2	205.7
N	-314.4	-198.0	1534	1390
O	-650.2	-324.1	2132	2196

Table 5.13-b

Midsection bending moments and axial forces of

Caisson 3C (5.2.5)



displacements

bending moments

m^*_b

axial forces

f^*_x

Diagram 5.7

Midsection results of Caisson 3C (5.2.5)

——————— SAMFSS

— — — — — SAP90

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The multilevel superelement substructuring method presented in this thesis has been successfully utilized to analyze boxlike caissons. Based on this approach, a computer program, SAMFSS has been developed for caisson structures with different numbers of layers and compartments.

Two types of structures can be analyzed using this computer program. The first type includes two-dimensional structures like membranes, deep beams, and thin and thick plate bending problems. The second type is three-dimensional boxlike structures like cubic tanks and marine caissons.

For two-dimensional problems, the accuracy of this method and convergence of results have been tested and verified on a variety of examples. SAMFSS program results have been compared with exact analytical or closed form solutions. For special case of square simply supported plate under a UDL, the accuracy and convergence of results have been verified for a wide range of thickness ratios. In general, the results converge very rapidly under a UDL, but more slowly under concentrated load.

For three-dimensional problems, the results of SAMFSS program have been compared with those of SAP90 program. In these comparisons, mesh fineness has been found to play the most important role in the accuracy and convergence of the results. The finer is the adopted mesh, the closer are the results of the two programs. Also, it should be noted that the two programs use different plate theories. For a caisson structure with fairly thick panels, SAMFSS program gives better results since it accounts for shear deformations.

This superelement substructuring approach significantly reduces data preparation time. Profiting by the repetition of similar substructures and generating mesh and load data, this method minimizes the required input file. Besides the time saving, it prevents human mistakes in mesh generation and load computation.

Moreover, this method reduces the required storage capacity and C.P.U. time of computer. This is achieved by processing several smaller stiffness matrices instead of an entire large stiffness matrix. Therefore, the multilevel superelement substructuring approach has some economical advantages over the conventional finite element method.

6.2 Recommendations for Further Study

The improvements which could be made to the present study and some features of future expansion are presented in this section.

a) Present study development

The following improvements may be added to the present program to enhance its capabilities:

1) developing other prismatic shapes including both cylindrical or polygon plans, e.g. hexagon or octagon, and stepped or conical forms of these caisson shapes,

2) developing other types of shell elements, specially a triangular element, which could be efficient for modelling other caisson shapes,

3) developing for caissons having some other convenient structural system or material like stiffened steel shell or composite steel frames with reinforced concrete slabs,

4) accounting for dynamic loads including wave, ice, and seismic forces as quasi-static loads in a linear static analysis, as provided by CAISSON software [34],

and 5) utilizing some sophisticated manipulation like block partitioning or skyline storage for stiffness matrix and improved equation solving techniques to increase the mesh fineness limit.

b) Future expansion

A comprehensive nonlinear dynamic analysis using multilevel superelement substructuring method for caissons may be developed after the following steps are performed:

1) applying the above mentioned improvements for the present study, specially items 4 and 5,

2) utilizing other general purpose programs for nonlinear and dynamic analysis of caissons,

and 3) comparing the results of different programs and those of full scale tests and investigating the performance of previously built caissons.

References

[1] *R. E. White*, *"FOUNDATION ENGINEERING"*, edited by G. A. Leonards, McGraw-Hill Book Company, 1962, pp 894-897, and 914-917.

[2] *R. D. Mindlin*, **"Influence of Rotary Inertia and Shear on Flexural Motions of Isotropic, Elastic Plates"**, Journal of Applied Mechanics, Vol. 18, 1951, pp 31-38.

[3] *E. D. L. Pugh, E. Hinton, and O. C. Zienkiewicz*, **"A Study of Quadrilateral Plate Bending Elements with Reduced Integration"**, International Journal for Numerical Methods in Engineering, Vol. 12, No. 7, 1978, pp 1059-1079.

[4] *O. C. Zienkiewicz, R. L. Taylor, and J. M. Too*, **"Reduced Integration Technique in General Analysis of Plates and Shells"**, International Journal for Numerical Methods in Engineering, Vol. 3, 1971, pp 275-290.

[5] *S. E. Pawsey and R. W. Clough*, **"Improved Numerical Integration of Thick Shell Finite Elements"**, International Journal for Numerical Methods in Engineering, Vol. 3, 1971, pp 575-586.

[6] *N. Bicanic and E. Hinton*, **"Spurious Modes in Two- Dimensional Isoparametric Elements"**, International Journal for Numerical Methods in Engineering, Vol. 14, 1979, pp 1545-1577.

[7] *T. J. R. Hughes and M. Cohen*, **"The Heterosis Finite Element for Plate Bending"**, Computers & Structures, Vol. 9, 1978, pp 445-450.

[8] *T. J. R. Hughes and Tezduyar*, **"Finite Elements Based upon Mindlin Plate Theory with Particular Reference to the Four-Node Bilinear Isoparametric Element"**, Journal of Applied Mechanics, Vol. 48, 1981, pp 587-596.

[9] *E. Hinton and H. C. Huang*, **"A Family of Quadrilateral Mindlin Plate Elements with Substitute Shear Strain Fields"**, Computers & Structures, Vol. 23, No. 3, 1986, pp 409-431.

[10] *E. N. Dvorkin and K. J. Bathe*, **"Continuum Mechanics Based Four-Node Shell Elements for General Nonlinear Analysis"**, Engineering Computation, Vol. 1, 1984, pp 77-88.

[11] *K. J. Bathe and E. N. Dvorkin*, **"A Four-Node Plate Bending Element Based on Mindlin/Reissner Plate Theory and Mixed Interpolation"**, International Journal for Numerical Methods in Engineering, Vol. 21, No. 2, 1985, pp 367-383.

[12] *H. C. Huang and E. Hinton*, **"A Nine-Node Lagrangian Mindlin Plate Element with Enhanced Shear Interpolation"**, Engineering Computation, Vol. 1, 1984, pp 369-379.

[13] *E. L. Wilson, R. L. Taylor, W. P. Doherty, and J. Ghaboussi*, **"Incompatible Displacement Modes"**, O N R Symposium on Numerical and Computer Methods in Structural Mechanics, University of Illinois, Urbana, 1971.

[14] *R. D. Cook*, **"Improved Two Dimensional Finite Element"**, Journal of the Structural Division, Proceedings of the ASCE, Vol. 100, No. ST9, September 1974, pp 1851-1863.

[15] *R. L. Taylor, P. J. Beresford, and E. L. Wilson*, **"A Nonconforming Element for Stress Analysis"**, International Journal for Numerical Methods in Engineering, Vol. 10, 1976, pp 1211-1219.

[16] *C. K. Choi and S. H. Kim*, **"Coupled Use of Reduced Integration and Non-Conforming Modes in Quadratic Mindlin Plate Element"**, International Journal for Numerical Methods in Engineering, Vol. 28, 1989, pp 1909-1928.

[17] *S. H. Kim and C. K. Choi*, **"Improvement of Quadratic Finite Element for Mindlin Plate Bending "**, International Journal for Numerical Methods in Engineering, Vol. 34, 1992, pp 197-208.

[18] *E. L. Wilson*, **"The Static Condensation Algorithm"**, International Journal for Numerical Methods in Engineering, Vol. 18, 1975, pp 198-203.

[19] *H. GangaRao, A. Abd Elmeged, and V. Chaudhary*, **"Macro Approach for Ribbed and Grid Plate Systems"**, Journal of the Engineering Mechanics Division, Proceedings of the ASCE, Vol. 101, No. EM1, February 1975, pp 25-43.

[20] *R. M. Gutkowsky and C. K. Wang*, **"Continuous Plate Analysis by Finite Panel Method"**, Journal of the Structural Division, Proceedings of the ASCE, Vol. 102, No. ST3, March 1976, pp 629-643.

[21] *R. M. Gutkowsky*, **"Finite Panel Analysis of Orthotropic Plate Systems"**, Journal of the Structural Division, Proceedings of the ASCE, Vol. 103, No. ST11, November 1977, pp 2211-2233.

[22] *T. Wah*, **"Elastic Quadrilateral Plates"**, Computers & Structures, Vol. 10, 1979, pp 457-466.

[23] *K. P. Jacobsen*, **"Fully Integrated Superelements: A Database Approach to Finite Element Analysis"**, Computers & Structures, Vol. 16, No. 1-4, 1983, pp 307-315.

[24] *T. J. Parsons, H. GangaRao, and W. C. Peterson*, "Macro-Element Analysis", Computers & Structures, Vol. 20, No. 5, 1985, pp 877-883.

[25] *Y. Liu and K. Chen*, **"A Two-Dimensional Mesh Generator for Variable Order Triangular and Rectangular Elements"**, Computers & Structures, Vol. 29, No. 6, 1988, pp 1033-1053.

[26] *M. Li*, **"Analysis of Box-Girder Bridges by Macro-Element Substructural Method"**, Master Thesis, University of Ottawa, 1991.

[27] *P. A. F. Martins and M. J. M. Barata Marques*, **"A Three-Dimensional Mesh Generator"**, Computers & Structures, Vol. 42, No. 4, 1992, pp 511-529.

[28] *P. Anderson*, **"SUBSTRUCTURAL ANALYSIS AND DESIGN"**, Ronald Press Company, 1956, pp 230-231 and 242-249.

[29] *D. H. Lee*, **"DEEP FOUNDATIONS AND SHEET PILING"**, London Concrete Publications Limited, 1961, pp 169 and 207-213.

[30] *L. J. Goodman and R. H. Karol*, **" THEORY AND PRACTICE OF FOUNDATION ENGINEERING"**, Macmillan Company, 1968, pp 233-244.

[31] *M. J. Tomlinson*, **"FOUNDATION DESIGN AND CONSTRUCTION"**, third edition by London Pitman, 1976, pp 310-311 and 321.

[32] *R. M. Koerner*, **"CONSTRUCTION AND GEOTECHNICAL METHODS IN FOUNDATION ENGINEERING"**, McGraw-Hill Book Company, 1984, pp 79-86.

[33] *US Army Corps of Engineers,* " **SHORE PROTECTION MANUAL** ", Coastal Engineering Research Centre, Department of the Army, Vol. I and II, fourth edition 1984, pp 5.56, 6.93, and 8.75-8.85.

[34] *Acres International Limited, Niagara Falls, Ontario,* " **CAISSON - AN INTEGRATED SOFTWARE SYSTEM FOR THE ANALYSIS AND DESIGN OF MARINE CAISSONS**", Vol. 1 & 2, Public Works Canada, Architectural and Engineering Services, July 1988.

[35] ***K. Terzaghi,*** **"THEORETICAL SOIL MECHANICS"**, John Wiley & Sons, 1954.

[36] *R. D. Cook, D. S. Malkus, and M. E. Plesha,* " **CONCEPTS AND APPLICATIONS OF FINITE ELEMENT ANALYSIS**", John Wily & Sons, Inc., third edition 1989, pp 228-231, 250-252, 257-260, and 314-328.

[37] ***S. P. Timoshenko and S. W. Kreiger,*** **"THEORY OF PLATES AND SHELLS"**, McGraw-Hill Book Company, third edition 1970, pp 116-120, 143-150, and 197-205.

[38] ***E. Hinton and D. R. J. Owen,*** **"FINITE ELEMENT SOFTWARE FOR PLATES AND SHELLS"**, Pineridge Press Limited, Swansea, U.K., 1984, pp 1-42 and 157-218.

Appendix A

PRINCIPAL SUBROUTINES

All subroutines called by the main program, except OUTDIS, are briefly explained. The latter one only writes nodal displacements and support reactions. A simple flow chart for main subroutines is presented after each explanation.

a) Subroutine GEOMET

1. Read general input data including; length, width, and height; number of layers, longitudinal and transverse panels, material types, and load cases; and material properties.
2. Call BASETH to read subgrade modulus and thickness of each base panel.
3. Read height and number of elements in height for each layer.
4. Read number of elements in each longitudinal and transverse panel.
5. Call GEOLYR to determine the thickness and span of all longitudinal and transverse panels of each layer.
6. Call NOPEPS to find; number of elements in length, width, and height; and total number of nodes and elements.
7. Call NODCOR to find the Cartesian coordinates of all nodes.
8. Call ELMNOD to determine the material type and node numbering of each element, and also find whether each d o f is boundary or internal in panel, layer, and caisson structural levels.

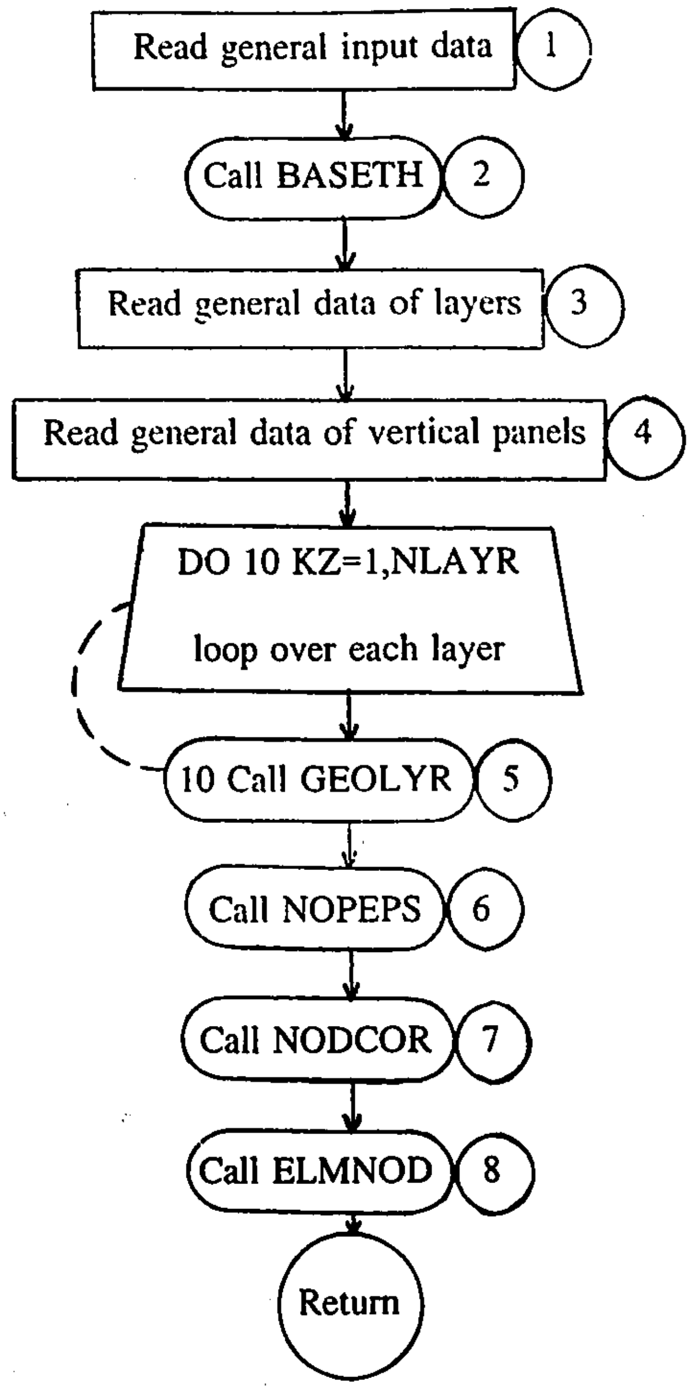


Fig A.1 Flow chart of subroutine GEOMET

b) Subroutine LOADS

1. Initialize the load matrix of the caisson structure, $\underline{P}_c = \underline{0}$.

2. Call LOADIN to read input loading data for each load case including water and soil level of all compartments and point load data.

3. Initialize the load matrix of elements, $\underline{P}_e = \underline{0}$.

4. Call HEAD to find water and soil head at every layer of each compartment.

5. Call LODBAS to find the contribution of base panel(s) to $\underline{P}_e$ due to water and soil pressure and gravity load.

6. Call LODWAL to find the contribution of vertical panels to $\underline{P}_e$ due to water and soil pressure and gravity load.

7. Call CONLOC to find which element is subjected to each point load.

8. Call CONLOD to add the contribution of each point load to $\underline{P}_e$.

9. Write the nodal loads of each element and form the load matrix of the caisson structure $\underline{P}_c$.

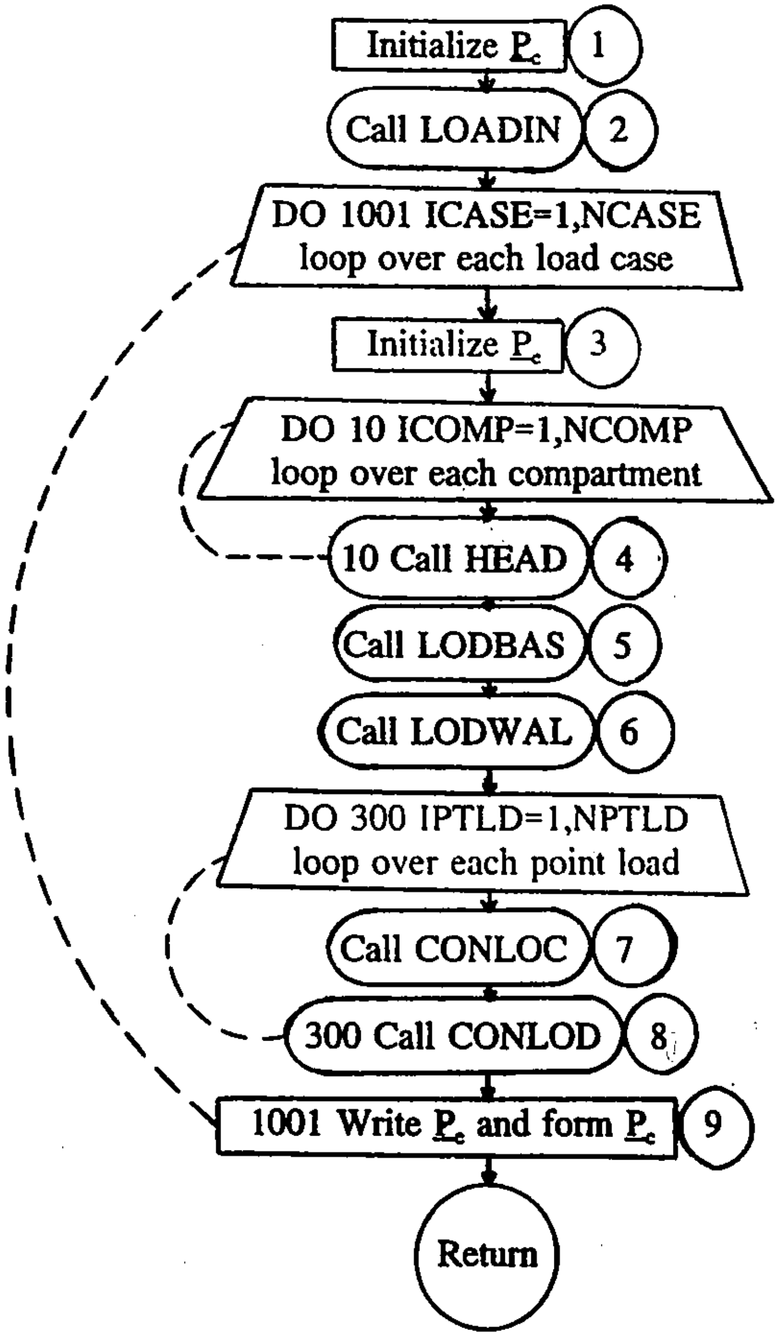


Fig. A.2 Flow chart of subroutine LOADS

c) subroutine GSTFNS (IT)

1. Initialize the stiffness matrix for; boundary dof's, $\underline{S}_{cbb}$; and internal dof's, [1]$\underline{S}_{ci}$, and the load matrix $\underline{P}_c$ of the caisson structure.

2. Initialize the stiffness matrix of elastic support $\underline{K}_{cs}$, and the stiffness and load matrices of the layer substructure: $\underline{S}_{lbb}$, [2]$\underline{S}_{li}$, and $\underline{P}_l$.

3. Call STBASE to process the stiffness and load matrices concerning the element(s) of base panel(s) as follows:

I-Sum the load and stiffness matrices of individual elements into the load and stiffness matrices of the relevant panel substructure; $\underline{P}_p$, $\underline{S}_{pbb}$ and [3]$\underline{S}_{pi}$.
II-Condense the internal dof's of both $\underline{P}_p$ and $\underline{S}_{pbb}$ to yield to the panel superelement load and stiffness matrices $\underline{P}_p^*$ and $\underline{S}_p^*$.
III-Sum $\underline{P}_p^*$ and $\underline{S}_p^*$ into the load and stiffness matrices of the relevant layer substructure; $\underline{P}_l$, $\underline{S}_{lbb}$ and $\underline{S}_{li}$.

4. Call STTRPL to do the same process as the above step concerning the elements of longitudinal panels.

5. Call STLNPL to do the same process as step 3 concerning the elements of transverse panels.

6. Call LYRCSN to process the stiffness and load matrices of the layer substructure(s) as follows:

I-Condense the internal dof's of both $\underline{P}_l$ and $\underline{S}_{lbb}$ to yield to the layer superelement load and stiffness matrices $\underline{P}_l^*$ and $\underline{S}_l^*$.
II-Sum $\underline{P}_l^*$ and $\underline{S}_l^*$ into the entire caisson load and stiffness matrices; $\underline{P}_c$, $\underline{S}_{cbb}$ and $\underline{S}_{ci}$.

7. Initialize $\underline{P}_l$, $\underline{S}_{lbb}$ and $\underline{S}_{li}$ to repeat the above steps 4~6 for the next possible layer.

8. Complete the symmetric lower triangles of $\underline{S}_{cbb}$ and $\underline{S}_{ci}$.

---

[1] $\underline{S}_{ci} = [\underline{S}_{cib} \; , \; \underline{S}_{cii}]$

[2] $\underline{S}_{li} = [\underline{S}_{lib} \; , \; \underline{S}_{lii}]$

[3] $\underline{S}_{pi} = [\underline{S}_{pib} \; , \; \underline{S}_{pii}]$

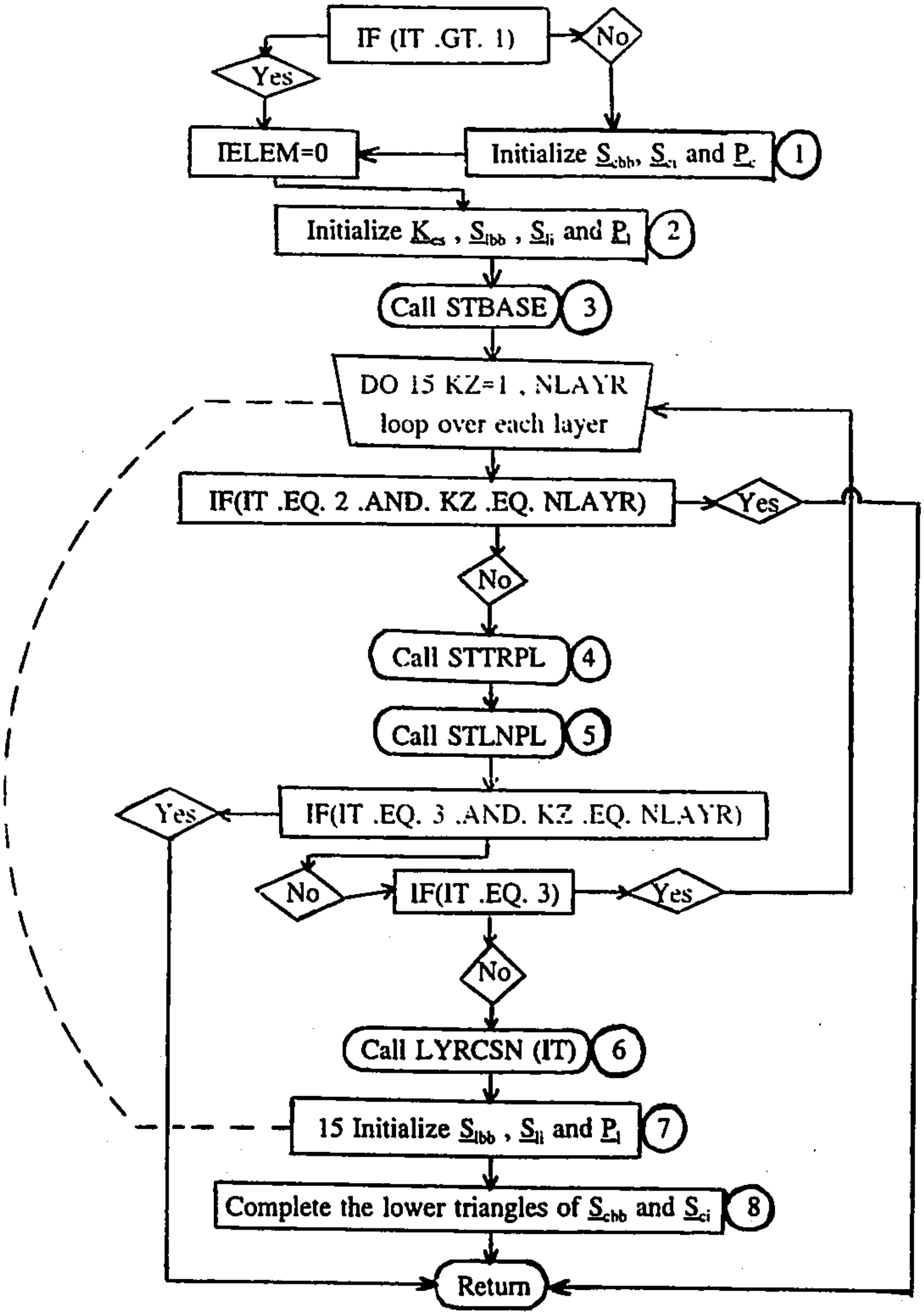


Fig. A:3 Flow chart of subroutine GSTFNS

d) **Subroutine SOLVEQ**

1. Initialize general nodal displacements $\underline{\delta}$ and caisson nodal displacements $\underline{\delta} = [\underline{\delta}_{cb}, \underline{\delta}_{ci}]$.

2. Condense $\underline{P}_c$ and $\underline{S}_{cbb}$ to the load and stiffness matrices of the caisson superelement $\underline{P}_c^*$ and $\underline{S}_c^*$.

3. Call MATINV to inverse $\underline{S}_c^*$ into $\underline{S}^{*}{}_c^{-1}$

4. Find the boundary nodal displacements of the entire caisson structure $\underline{\delta}_{cb}$.

5. Recover the internal nodal displacements of the entire caisson structure $\underline{\delta}_{ci}$.

6. Rearrange caisson nodal displacements in their proper places.

7. Call SOLVLY to compute the internal nodal displacements of the highest layer substructure.

8. Call GSTFNS (IT=2,3) to find the internal nodal displacements; $\underline{\delta}_{li}$ of other layer substructures; and $\underline{\delta}_{pi}$ of the panel substructures.

9. Compute base reactions.

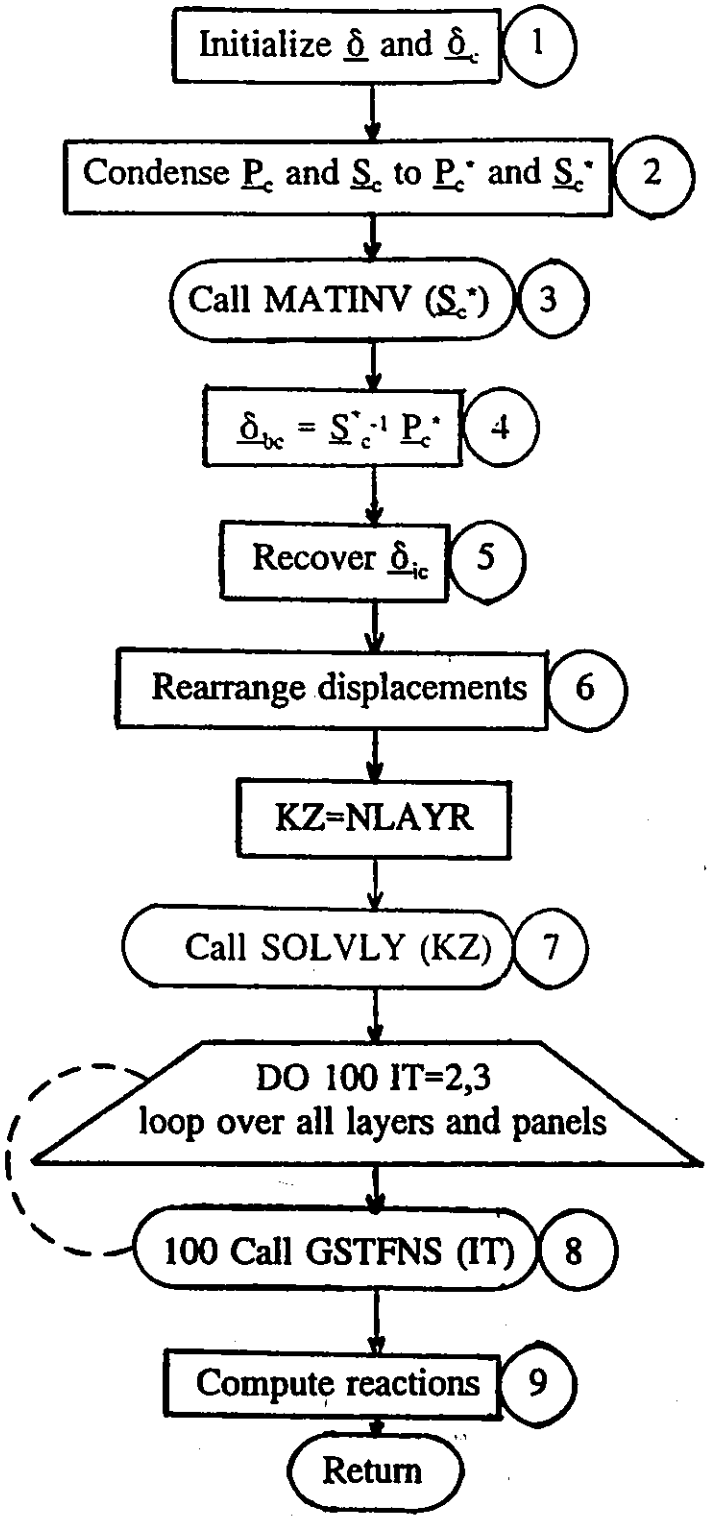


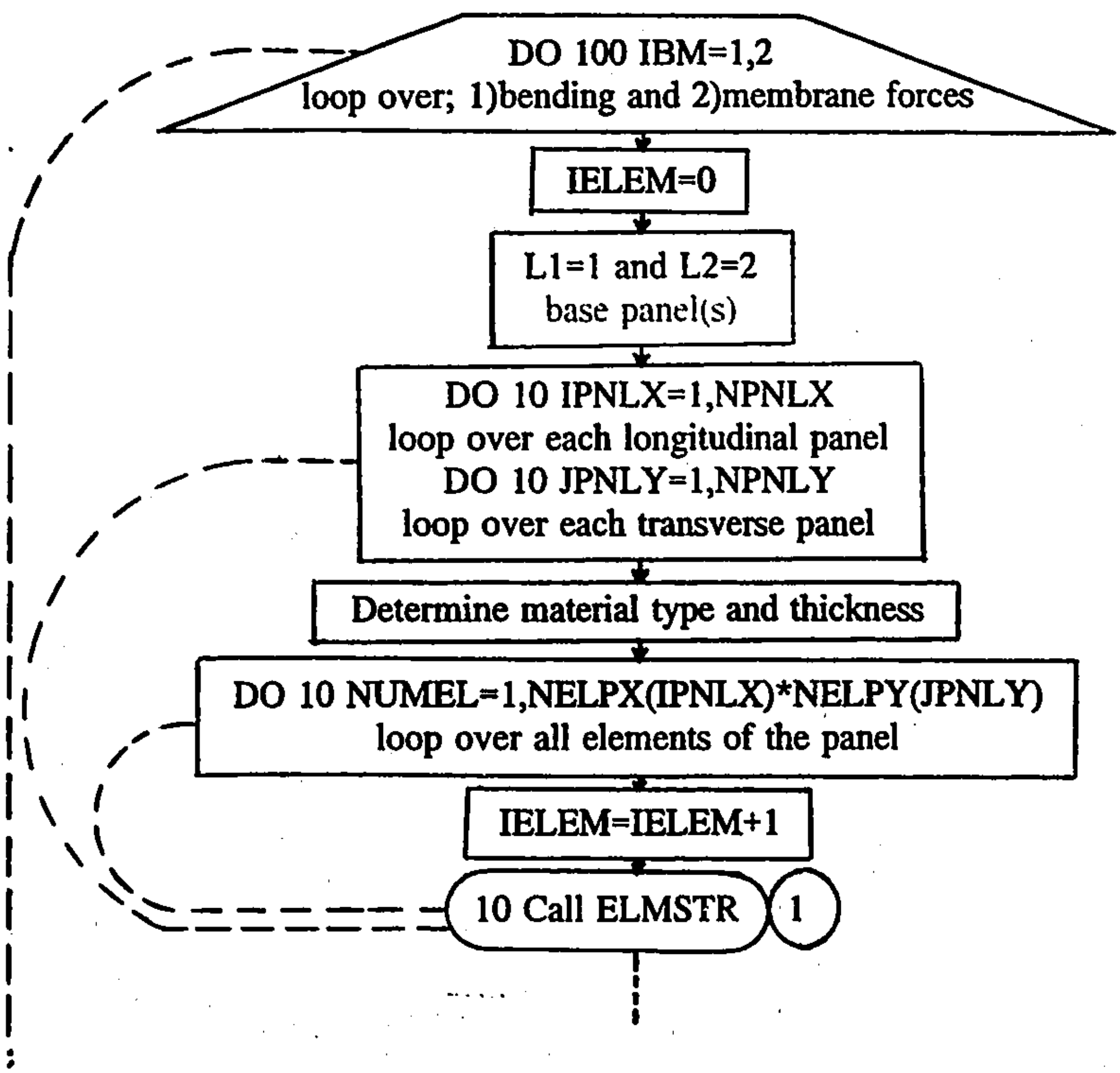
Fig. A.4 Flow chart of subroutine SOLVEQ

e) Subroutine OUTSTR

1. Compute internal forces (bending forces and membrane stresses) for all elements os base panel(s).

2. Compute internal forces for individual elements of longitudinal vertical panels.

3. Compute internal forces for individual elements of transverse vertical panels.



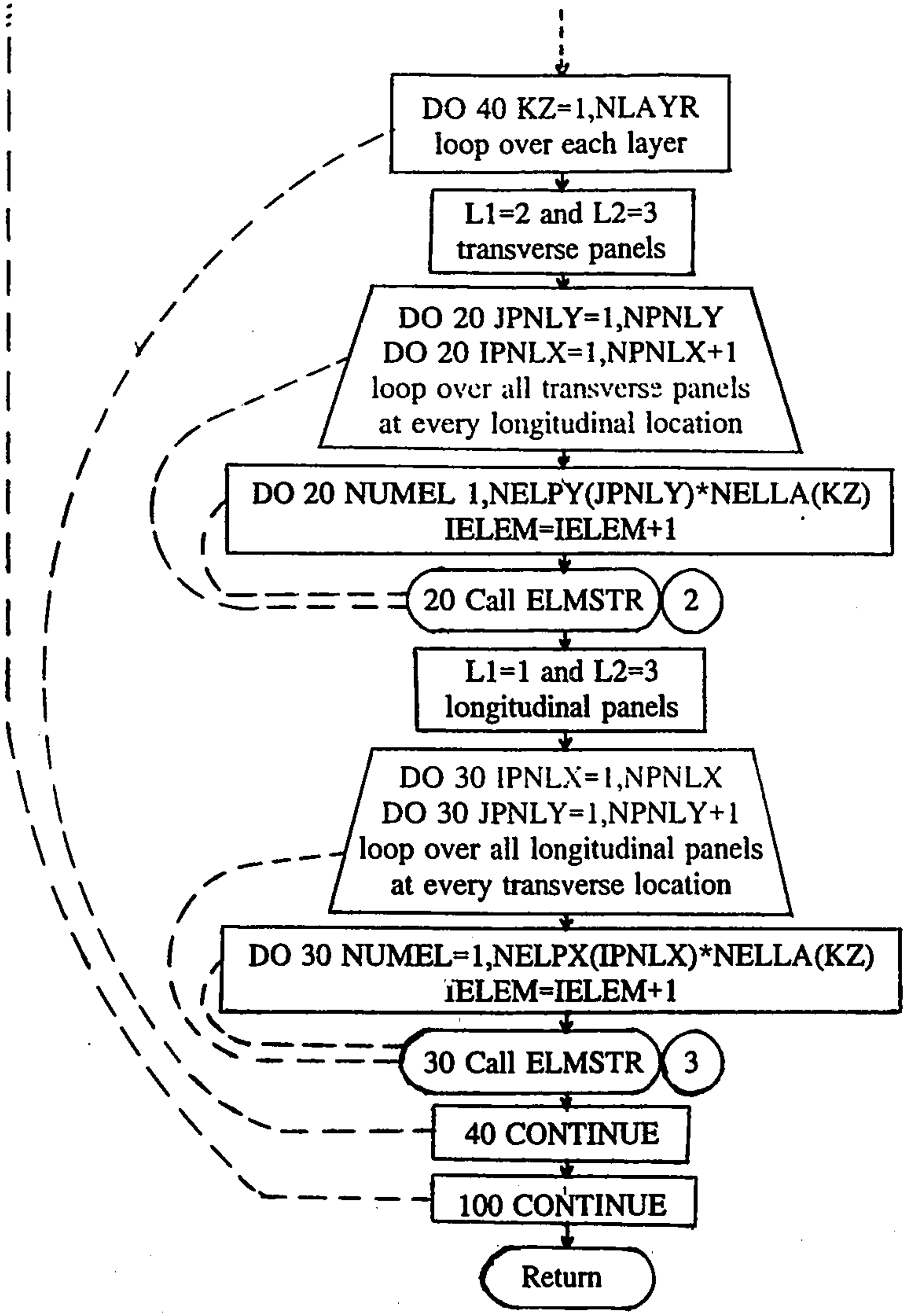


Fig. A.5 Flow chart of subroutine OUTSTR

Appendix B

SAMPLE INPUT DATA FILES

Input data files of the cubic tank example for both SAMFSS and SAP90 programs are presented. Because of the symmetry in two horizontal directions (x and y), only a quarter of structure is modelled in SAP90 program.

a) SAMFSS Input File

```
Cubic Tank
3.1,3.1,2.95
1,1,1
0,0,0,0,1,1
1,2.188E9,.15,2400.
1e6
1,.1
2.95,1
4
4
1,.1,1,.1
1,.1,1,.1
1,1,1000.
1,2000.,2500.,0.,30.,0.,0.,30.
0.,0.,0.,0.,2.95
0.,0.,0.,0.,0.
1,1,1,1,1
1,0.,1.,0.
```

b) SAP90 Input File

Cubic Tank

```
system
l=1

joints
1   x=0.     y=0.    z=0.
9                    z=3.
37           y=1.5   z=0.
45                   z=3.   q=1,9,37,45,1,9
46  x=.375   y=0.    z=0.
54                   z=3.
73  x=1.5            z=0.
81                   z=3.   q=46,54,73,81,1,9
82  x=.375   y=.375  z=0.
85           y=1.5
94  x=1.5    y=.375
97           y=1.5           q=82,85,94,97,1,4

springs
1           k=35156.25,35156.25,35156.25,0,0,0
10 28 9     k=70312.5,70312.5,70312.5,0,0,0
46 64 9     k=70312.5,70312.5,70312.5,0,0,0
37          k=35156.25,0,35156.25,0,0,0
73          k=0,35156.25,35156.25,0,0,0
85 93 4     k=70312.5,0,70312.5,0,0,0
94 96       k=0,70312.5,70312.5,0,0,0
```

```
82 84          k=140625,140625,140625,0,0,0
86 88          k=140625,140625,140625,0,0,0
90 92          k=140625,140625,140625,0,0,0
97             k=0,0,35156.25,0,0,0

restraints
37 45    r=0,1,0,1,0,1
85 93 4  r=0,1,0,1,0,1
73 81    r=1,0,0,0,1,1
94 96    r=1,0,0,0,1,1
97       r=1,1,0,1,1,1

potential
1  97  w=1000.,3.0

shell
nm=1  p=-1.
1  e=2.188e9  u=.15
1  jq=1,46,10,82  etype=0   m=1  th=.1,.1
2  jq=46,55,82,86
3  jq=55,64,86,90
4  jq=64,73,90,94
5  jq=10,82,19,83
6  jq=19,83,28,84
7  jq=28,84,37,85
8  jq=82,86,83,87                   g=3,3
17 jq=1,10,2,11                     g=4,8
49 jq=1,2,46,47                     g=8,1
57 jq=46,47,55,56                   g=8,3
```

loads

```
1  73  36  l=1  f=0.,0.,1.7578125
97              f=0.,0.,1.7578125
10 28  9        f=0.,0.,3.515625
46 64  9        f=0.,0.,3.515625
85 93  4        f=0.,0.,3.515625
94 96           f=0.,0.,3.515625
82 84           f=0.,0.,7.03125
86 88           f=0.,0.,7.03125
90 92           f=0.,0.,7.03125
```

Appendix C

SAMFSS COMPUTER PROGRAM

```
C
C
C
C     SSSS        AAA        MM       MM     FFFFFFFF      SSSS         SSSS
C   SS    SS     AA AA      MMMM     MMMM    FF          SS    SS     SS    SS
C  SS           AA   AA    MM MM   MM MM    FF         SS           SS
C  SS          AA     AA   MM  MM MM  MM    FF         SS           SS
C    SS        AAAAAAAAA   MM  MM  MM  MM    FFFFFFF     SS           SS
C      SS      AA     AA   MM   MM MM  MM    FF            SS           SS
C        SS    AA     AA   MM    MMM   MM    FF              SS           SS
C         SS   AA     AA   MM     M    MM    FF               SS           SS
C SS    SS     AA     AA   MM          MM    FF         SS    SS     SS    SS
C   SSSS       AA     AA   MM          MM    FF           SSSS         SSSS
C
C
C
C
C
C
C      STRUCTURAL ANALYSIS BY MINDLIN FLAT SHELL SUPERELEMETS
C
C
C      A MULTILEVEL SUPERELEMENT SUBSTRUCTURE FOR THE ANALYSIS OF
C                 RECTANGULAR BOXLIKE CAISSONS USING
C      RECTANGULAR 9-NODE LAGRANGIAN MINDLIN FLAT SHELL ELEMENTS
C
C
C
C      PREPARED BY:
C      SAEED ARONI HESARI
C
C      AS A PART OF THE M.SC. THESIS
C
C      DEPARTMENT OF CIVIL ENGINEERING
C      UNIVERSITY OF OTTAWA
C
C      APRIL 94
C
C
C
C
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON/STFRR/ STFRR
       COMMON/STFCV/ STFCV
       COMMON/CLOAD/ CLOAD
       COMMON/CCSLD/ CCSLD
       COMMON/ELOAD/ ELOAD
       COMMON/COORD/ COORD
       DIMENSION       ASDIS(918,5),COORD(153,3),ELOAD(36,54),
     +                 LNODS(36,9),MATNO(36),PROPS(10,3),STREL(9,8),
     +                 TREAC(25,3,5),CLOAD(918,5),STFRR(252,252),
     +                 STFCV(126,378),NELLA(3),NELPX(3),NELPY(3),
     +                 HIGHT(3),THPNX(3,4),THPNY(4,3),THBAS(3,3),
     +                 SPNXL(3),SPNYW(3),SKNOD(153),CCSLD(378,5),
     +                 INDPL(153),INDLY(153,3),NLPRM(3),NLPCN(3),
     +                 NDLYR(42,2),NDLYC(16,2),INDCS(153)
       DATA  NDOFN/6/,NNODE/9/,NGAUB/3/,NGAUS/2/,NGAUM/2/,NPOIN/153/
       NTOTV=NPOIN*NDOFN
       NEVAB=NNODE*NDOFN
       CALL     GEOMET  (NPNLX,NPNLY,NLAYR,NELPX,NELPY,NELLA,
     +                   LNODS,PROPS,MATNO,NCASE,INDPL,
     +                   NPOIN,NDOFN,NELEM,NTOTV,THPNX,THPNY,
     +                   THBAS,HIGHT,ZHIGH,THCBS,XLNGT,YWDTH,
     +                   SPNXL,SPNYW,XLNOD,YWNOD,NCPRM,NCPCN,
     +                   NNODE,NLPRM,NLPCN,NDLYR,NDLYC,INDLY,
     +                   INDCS,SK)
       CALL     LOADS   (NCASE,LNODS,MATNO,NELEM,
     +                   NEVAB,NGAUB,NNODE,PROPS,NDOFN,
     +                   NPNLX,NPNLY,NLAYR,NELPX,NELPY,
```

```
     +                 NELLA,THPNX,THPNY,HIGHT,ZHIGH,THCBS,
     +                 SPNXL,SPNYW,XLMOD,YWMOD)
      IT=1
      CALL    GSTFNS  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                 NGAUM,NNODE,PROPS,NDOFN,NCPRM,NCPCN,
     +                 NPNLX,NPNLY,NLAYR,NELLA,NELPX,NELPY,
     +                 SPNXL,SPNYW,THBAS,THPNX,THPNY,SKNOD,
     +                 NPOIN,NCASE,INDPL,INDLY,NLPRM,NLPCN,
     +                 NDLYR,NDLYC,INDCS,SK,IT)
      CALL    SOLVEQ  (TREAC,NPOIN,NCPRM,NCPCN,
     +                 ASDIS,LNODS,NEVAB,NGAUB,NGAUS,NGAUM,
     +                 NNODE,NDOFN,MATNO,PROPS,INDPL,NPNLX,
     +                 NPNLY,NLAYR,NELPX,NELPY,NELLA,THBAS,
     +                 THPNX,THPNY,SPNXL,SPNYW,INDLY,NLPRM,
     +                 NLPCN,NDLYR,NDLYC,INDCS,SKNOD,NCASE,
     +                 SK)
      DO 10 ICASE=1,NCASE
      CALL    OUTDIS  (ICASE,ASDIS,NDOFN,NPOIN,TREAC,SKNOD)
      CALL    OUTSTR  (ICASE,ASDIS,LNODS,MATNO,NNODE,NDOFN,
     +                 NGAUS,PROPS,STREL,NPNLX,NPNLY,NLAYR,
     +                 NELPX,NELPY,NELLA,THBAS,THPNX,THPNY)
 10   CONTINUE
      STOP
      END

      SUBROUTINE      GSTFNS  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                         NGAUM,NNODE,PROPS,NDOFN,NCPRM,NCPCN,
     +                         NPNLX,NPNLY,NLAYR,NELLA,NELPX,NELPY,
     +                         SPNXL,SPNYW,THBAS,THPNX,THPNY,SKNOD,
     +                         NPOIN,NCASE,INDPL,INDLY,NLPRM,NLPCN,
     +                         NDLYR,NDLYC,INDCS,SK,IT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/STFRR/ STFRR
      COMMON/STFCV/ STFCV
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/ESTIF/ ESTIF
      COMMON/GSTIF/ GSTIF
      COMMON/CLOAD/ CLOAD
      COMMON/CCSLD/ CCSLD
      COMMON/CLYLD/ CLYLD
      COMMON/COORD/ COORD
      DIMENSION       COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                ESTIF(54,54),GSTIF(54,54),STFRR(252,252),
     +                STFCV(126,378),INDPL(153),NELLA(3),NELPX(3),
     +                NELPY(3),SKNOD(153),THBAS(3,3),THPNX(3,4),
     +                THPNY(4,3),SPNXL(3),SPNYW(3),KONTE(153),
     +                CLOAD(918,5),CCSLD(378,5),ASDIS(918,5),
     +                INDLY(153,3),INDCS(153),NLPRM(3),NLPCN(3),
     +                NDLYR(42,2),NDLYC(16,2),SLYRR(252,252),
     +                SLYCV(36,288),CLYLD(288,5)
      WRITE(8,111)
111   FORMAT(' subroutine gstfns')
      IF(IT .NE. 1) GO TO 5
      CALL    RAZERO  (STFRR,252,252)
      CALL    RAZERO  (STFCV,126,378)
      CALL    RAZERO  (CCSLD,378,5)
      NCVRM=NCPRM*NDOFN
      NCVCN=NCPCN*NDOFN
  5   IELEM=0
      CALL    RVZERO  (SKNOD,153)
      CALL    IVZERO  (KONTE,153)
      CALL    RAZERO  (SLYRR,252,252)
      CALL    RAZERO  (SLYCV,36,288)
      CALL    RAZERO  (CLYLD,288,5)
      CALL    STBASE  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                 NGAUM,NNODE,PROPS,NDOFN,INDPL,NPNLX,
     +                 NPNLY,NELPX,NELPY,THBAS,SPNXL,SPNYW,
     +                 SKNOD,SK,IELEM,IT,KONTE,NLPRM,NLPCN,
```

```
     +                 NPOIN,NCASE,INDLY)
      DO 15 KZ=1,NLAYR
      IF(IT .EQ. 2 .AND. KZ .EQ. NLAYR) GO TO 16
      CALL    STTRPL  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                 NGAUM,NNODE,PROPS,NDOFN,INDPL,NPNLX,
     +                 NPNLY,NELLA,NELPY,THPNY,SPNYW,SKNOD,
     +                 KZ,IELEM,IT,KONTE,NPOIN,NLPRM,NLPCN,
     +                 NCASE,INDLY)
      CALL    STLNPL  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                 NGAUM,NNODE,PROPS,NDOFN,INDPL,NPNLX,
     +                 NPNLY,NELLA,NELPX,THPNX,SPNXL,SKNOD,
     +                 KZ,IELEM,IT,KONTE,NPOIN,NLPRM,NLPCN,
     +                 NCASE,INDLY)
      IF(IT .EQ. 3) GO TO 15
      NLKPR=NLPRM(KZ)
      NLKPC=NLPCN(KZ)
      CALL    LYRCSN  (IT,KZ,NDOFN,NCASE,NLKPR,NLKPC,
     +                 NCVRM,INDLY,NDLYR,NDLYC,INDCS,ASDIS)
      IF(KZ .EQ. NLAYR) GO TO 15
10    CALL    RAZERO  (SLYRR,252,252)
      CALL    RAZERO  (CLYLD,288,5)
      CALL    RAZERO  (SLYCV,36,288)
15    CONTINUE
16    DO 20 ICVRM=2,NCVRM
      DO 20 JCVRM=1,ICVRM-1
20    STFRR(ICVRM,JCVRM)=STFRR(JCVRM,ICVRM)
      DO 30 ICVCN=2,NCVCN
      DO 30 JCVCN=1,ICVCN-1
30    STFCV(ICVCN,JCVCN+NCVRM)=STFCV(JCVCN,ICVCN+NCVRM)
      RETURN
      END

      SUBROUTINE      STBASE  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                         NGAUM,NNODE,PROPS,NDOFN,INDPL,NPNLX,
     +                         NPNLY,NELPX,NELPY,THBAS,SPNXL,SPNYW,
     +                         SKNOD,SK,IELEM,IT,KONTE,NLPRM,NLPCN,
     +                         NPOIN,NCASE,INDLY)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/PNSRR/ PNSRR
      COMMON/PNSCV/ PNSCV
      COMMON/ESTIF/ ESTIF
      COMMON/GSTIF/ GSTIF
      COMMON/CLOAD/ CLOAD
      COMMON/CLYLD/ CLYLD
      COMMON/CPNLD/ CPNLD
      COMMON/COORD/ COORD
      DIMENSION       COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                ESTIF(54,54),GSTIF(54,54),INDPL(153),NELPX(3),
     +                NELPY(3),SSTF(9),SKNOD(153),ELCOD(2,9),SHAPE(9),
     +                THBAS(3,3),SPNXL(3),SPNYW(3),KONTP(153),
     +                IPNOD(153),CPNLD(150,5),KONTE(153),CLOAD(918,5),
     +                ASDIS(918,5),PNSRR(96,96),
     +                PNSCV(54,150),INDLY(153,3),NDPRF(10),NDPCF(6),
     +                SLYRR(252,252),SLYCV(36,288),CLYLD(288,5),
     +                NLPRM(3),NLPCN(3)
      WRITE(8,111)
111   FORMAT(' subroutine stbase')
      L1=1
      L2=2
      KZ=1
      NLKPR=NLPRM(KZ)
      NLKPC=NLPCN(KZ)
      NLVRM=NLKPR*NDOFN
      DO 140 JPNLY=1,NPNLY
      DO 130 IPNLX=1,NPNLX
      NPPRM=0
      NPPCN=0
```

```
      CALL     IVZERO  (KONTP,153)
      CALL     IVZERO  (IPNOD,153)
      DO 5 NUMEL=1,NELPX(IPNLX)*NELPY(JPNLY)
      NOELM=IELEM+NUMEL
 5    CALL     PNLNDV  (NOELM,NNODE,LNODS,INDPL,IPNOD,NPPRM,
     +                  NPPCN,KONTP)
      NPVRM=NPPRM*NDOFN
      NPVCN=NPPCN*NDOFN
      LPROP=MATNO(IELEM+1)
      THICK=THBAS(IPNLX,JPNLY)
      IF(JPNLY .EQ. 1 .AND. IPNLX .EQ. 1) GO TO 20
      NUM=0
      IF(IPNLX .GT. 1) GO TO 10
      IF(NELPY(JPNLY) .NE. NLPY0) NUM=NUM+1
      IF(SPNYW(JPNLY) .NE. SPNY0) NUM=NUM+1
 10   IF(NELPX(IPNLX) .NE. NLPX0) NUM=NUM+1
      IF(LPROP .NE. LPRP0)        NUM=NUM+1
      IF(THICK .NE. THCK0)        NUM=NUM+1
      IF(SPNXL(IPNLX) .NE. SPNX0) NUM=NUM+1
      IF(NUM .EQ. 0) GO TO 90
 20   NUM=1
      CALL     STIFMA  (THICK,LNODS,LPROP,NGAUM,
     +                  NNODE,PROPS,NDOFN,IELEM,L1,L2)
      CALL     STIFPB  (THICK,LNODS,LPROP,NGAUB,NGAUS,
     +                  NNODE,PROPS,NDOFN,IELEM,L1,L2)
      DO 30 IELVA=1,NEVAB-1
      DO 30 JELVA=IELVA+1,NEVAB
 30   ESTIF(JELVA,IELVA)=ESTIF(IELVA,JELVA)
      DO 40 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      DO 40 IDIME=1,2
 40   ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
      RAREA=(ELCOD(1,3)-ELCOD(1,1))*(ELCOD(2,5)-ELCOD(2,3))
      DO 50 INODE=1,7,2
 50   SHAPE(INODE)=1/36.
      DO 60 INODE=2,8,2
 60   SHAPE(INODE)=1/9.
      SHAPE(9)=4/9.
      DO 70 INODE=1,NNODE
      SSTF(INODE)=SHAPE(INODE)*RAREA*SK
      DO 70 IDOFN=1,3
      IPOSN=(INODE-1)*NDOFN+IDOFN
 70   ESTIF(IPOSN,IPOSN)=ESTIF(IPOSN,IPOSN)+SSTF(INODE)
      DO 80 I=1,NEVAB
      DO 80 J=1,NEVAB
 80   GSTIF(I,J)=ESTIF(I,J)
      IELEM=IELEM-1
 90   LPRP0=LPROP
      THCK0=THICK
      NLPX0=NELPX(IPNLX)
      SPNX0=SPNXL(IPNLX)
      CALL     RAZERO  (CPNLD,150,5)
      CALL     RAZERO  (PNSRR,96,96)
      CALL     RAZERO  (PNSCV,54,150)
      DO 100 NUMEL=1,NELPX(IPNLX)*NELPY(JPNLY)
      IELEM=IELEM+1
      CALL     PNLASM  (1,IELEM,LNODS,NNODE,NDOFN,
     +                  IPNOD,SSTF,SKNOD,NPVRM,KONTE,
     +                  NCASE)
 100  CONTINUE
      DO 110 IPVRM=2,NPVRM
      DO 110 JPVRM=1,IPVRM-1
 110  PNSRR(IPVRM,JPVRM)=PNSRR(JPVRM,IPVRM)
      DO 115 IPVCN=2,NPVCN
      DO 115 JPVCN=1,IPVCN-1
 115  PNSCV(IPVCN,JPVCN+NPVRM)=PNSCV(JPVCN,IPVCN+NPVRM)
      CALL     CNDNSP  (NCASE,NPVRM,NPVCN)
      IF(IT .NE. 3) GO TO 120
      CALL     SOLVPN  (NCASE,NPPRM,NPPCN,NPOIN,NDOFN,IPNOD,
```

```
     +                    ASDIS)
      GO TO 130
120   CALL     LYRNDV  (KZ,NPOIN,NPPRM,IPNOD,INDLY,NPPRF,NPPCF,
     +                    NDPRF,NDPCF)
      CALL     LYRASM  (KZ,NLVRM,NCASE,NDOFN,NPPRF,NPPCF,NDPRF,
     +                    NDPCF,IPNOD,INDLY)
130   CONTINUE
      SPNY0=SPNYW(JPNLY)
      NLPY0=NELPY(JPNLY)
140   CONTINUE
      RETURN
      END

      SUBROUTINE      STTRPL  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                          NGAUM,NNODE,PROPS,NDOFN,INDPL,NPNLX,
     +                          NPNLY,NELLA,NELPY,THPNY,SPNYW,SKNOD,
     +                          KZ,IELEM,IT,KONTE,NPOIN,NLPRM,NLPCN,
     +                          NCASE,INDLY)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/PNSRR/ PNSRR
      COMMON/PNSCV/ PNSCV
      COMMON/ESTIF/ ESTIF
      COMMON/GSTIF/ GSTIF
      COMMON/CLOAD/ CLOAD
      COMMON/CLYLD/ CLYLD
      COMMON/CPNLD/ CPNLD
      COMMON/COORD/ COORD
      DIMENSION       COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                ESTIF(54,54),GSTIF(54,54),INDPL(153),NELLA(3),
     +                NELPY(3),SSTF(9),SKNOD(153),THPNY(4,3),SPNYW(3),
     +                KONTP(153),IPNOD(153),CPNLD(150,5),KONTE(153),
     +                CLOAD(918,5),ASDIS(918,5),PNSRR(96,96),
     +                PNSCV(54,150),INDLY(153,3),NDPRF(10),
     +                NDPCF(6),SLYRR(252,252),SLYCV(36,288),
     +                CLYLD(288,5),NLPRM(3),NLPCN(3)
      WRITE(8,111)
111   FORMAT(' subroutine sttrpl')
      CALL     RVZERO  (SSTF,9)
      L1=2
      L2=3
      NLKPR=NLPRM(KZ)
      NLKPC=NLPCN(KZ)
      NLVRM=NLKPR*NDOFN
      NLVCN=NLKPC*NDOFN
      DO 100 IPNLX=1,NPNLX+1
      DO 100 JPNLY=1,NPNLY
      NPPRM=0
      NPPCN=0
      CALL     IVZERO  (KONTP,153)
      CALL     IVZERO  (IPNOD,153)
      DO 10 NUMEL=1,NELPY(JPNLY)*NELLA(KZ)
      NOELM=IELEM+NUMEL
 10   CALL     PNLNDV  (NOELM,NNODE,LNODS,INDPL,IPNOD,NPPRM,
     +                   NPPCN,KONTP)
      NPVRM=NPPRM*NDOFN
      NPVCN=NPPCN*NDOFN
      LPROP=MATNO(IELEM+1)
      THICK=THPNY(IPNLX,JPNLY)
      IF(IPNLX .EQ. 1 .AND. JPNLY .EQ. 1) GO TO 30
      NUM=0
      IF(NELPY(JPNLY) .NE. NLPY0) NUM=NUM+1
      IF(LPROP .NE. LPRP0)        NUM=NUM+1
      IF(THICK .NE. THCK0)        NUM=NUM+1
      IF(SPNYW(JPNLY) .NE. SPNY0) NUM=NUM+1
      IF(NUM .EQ. 0) GO TO 60
 30   NUM=1
      CALL    STIFMA  (THICK,LNODS,LPROP,NGAUM,
```

```
     +                   NNODE,PROPS,NDOFN,IELEM,L1,L2)
      CALL     STIFFB   (THICK,LNODS,LPROP,NGAUB,NGAUS,
     +                   NNODE,PROPS,NDOFN,IELEM,L1,L2)
      DO 50 IELVA=1,NEVAB-1
      DO 50 JELVA=IELVA+1,NEVAB
50    ESTIF(JELVA,IELVA)=ESTIF(IELVA,JELVA)
      CALL     TRANSM   (NNODE,NEVAB)
      IELEM=IELEM-1
60    LPRPO=LPROP
      THCKO=THICK
      NLPYO=NELPY(JPNLY)
      SPNYO=SPNYN(JPNLY)
      CALL     RAZERO   (CPNLD,150,5)
      CALL     RAZERO   (PNSRR,96,96)
      CALL     RAZERO   (PNSCV,54,150)
      DO 70 NUMEL=1,NELPY(JPNLY)*NELLA(KZ)
      IELEM=IELEM+1
      CALL     PNLASM   (0,IELEN,LNODS,NNODE,NDOFN,
     +                   IPNOD,SSTF,SKNOD,NPVRM,KONTE,
     +                   NCASE)
70    CONTINUE
      DO 80 IPVRM=2,NPVRM
      DO 80 JPVRM=1,IPVRM-1
80    PNSRR(IPVRM,JPVRM)=PNSRR(JPVRM,IPVRM)
      DO 85 IPVCN=2,NPVCN
      DO 85 JPVCN=1,IPVCN-1
85    PNSCV(IPVCN,JPVCN+NPVRM)=PNSCV(JPVCN,IPVCN+NPVRM)
      CALL     CNDNSP   (NCASE,NPVRM,NPVCN)
      IF(IT .NE. 3) GO TO 90
      CALL     SOLVPN   (NCASE,NPPRM,NPPCN,NPOIN,NDOFN,IPNOD,
     +                   ASDIS)
      GO TO 100
90    CALL     LYRNDV   (KZ,NPOIN,NPPRM,IPNOD,INDLY,NPPRF,NPPCF,
     +                   NDPRF,NDPCF)
      CALL     LYRASM   (KZ,NLVRM,NCASE,NDOFN,NPPRF,NPPCF,NDPRF,
     +                   NDPCF,IPNOD,INDLY)
100   CONTINUE
      RETURN
      END

      SUBROUTINE      STLNPL  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                         NGAUM,NNODE,PROPS,NDOFN,INDPL,NPNLX,
     +                         NPNLY,NELLA,NELPX,THPNX,SPNXL,SKNOD,
     +                         KZ,IELEM,IT,KONTE,NPOIN,NLPRM,NLPCN,
     +                         NCASE,INDLY)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/PNSRR/ PNSRR
      COMMON/PNSCV/ PNSCV
      COMMON/ESTIF/ ESTIF
      COMMON/GSTIF/ GSTIF
      COMMON/CLOAD/ CLOAD
      COMMON/CLYLD/ CLYLD
      COMMON/CPNLD/ CPNLD
      COMMON/COORD/ COORD
      DIMENSION       COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                ESTIF(54,54),GSTIF(54,54),INDPL(153),NELLA(3),
     +                NELPX(3),SSTF(9),SKNOD(153),THPNX(3,4),SPNXL(3),
     +                KONTP(153),IPNOD(153),CPNLD(150,5),KONTE(153),
     +                CLOAD(918,5),ASDIS(918,5),PNSRR(96,96),
     +                PNSCV(54,150),INDLY(153,3),NDPRF(10),
     +                NDPCF(6),SLYRR(252,252),SLYCV(36,288),
     +                CLYLD(288,5),NLPRM(3),NLPCN(3)
      WRITE(8,111)
111   FORMAT(' subroutine stlnpl')
      CALL     RVZERO   (SSTF,9)
      L1=1
      L2=3
```

```
      NLKPR=NLPRM(KZ)
      NLKPC=NLPCN(KZ)
      NLVRM=NLKPR*NDOFN
      NLVCN=NLKPC*NDOFN
      DO 100 JPNLY=1,NPNLY+1
      DO 100 IPNLX=1,NPNLX
      NPPRM=0
      NPPCN=0
      CALL    IVZERO  (KONTP,153)
      CALL    IVZERO  (IPNOD,153)
      DO 10 NUMEL=1,NELPX(IPNLX)*NELLA(KZ)
      NOELM=IELEM+NUMEL
10    CALL    PNLNDV  (NOELM,NNODE,LNODS,INDPL,IPNOD,NPPRM,
     +                 NPPCN,KONTP)
      NPVRM=NPPRM*NDOFN
      NPVCN=NPPCN*NDOFN
      LPROP=MATNO(IELEM+1)
      THICK=THPNX(IPNLX,JPNLY)
      IF(JPNLY .EQ. 1 .AND. IPNLX .EQ. 1) GO TO 30
      NUM=0
      IF(NELPX(IPNLX) .NE. NLPX0) NUM=NUM+1
      IF(LPROP .NE. LPRP0)        NUM=NUM+1
      IF(THICK .NE. THCK0)        NUM=NUM+1
      IF(SPNXL(IPNLX) .NE. SPNX0) NUM=NUM+1
      IF(NUM .EQ. 0) GO TO 60
30    NUM=1
      CALL    STIFMA  (THICK,LNODS,LPROP,NGAUM,
     +                 NNODE,PROPS,NDOFN,IELEM,L1,L2)
      CALL    STIFPB  (THICK,LNODS,LPROP,NGAUB,NGAUS,
     +                 NNODE,PROPS,NDOFN,IELEM,L1,L2)
      DO 50 IELVA=1,NEVAB-1
      DO 50 JELVA=IELVA+1,NEVAB
50    ESTIF(JELVA,IELVA)=ESTIF(IELVA,JELVA)
      CALL    TRANSL  (NNODE,NEVAB)
      IELEM=IELEM-1
60    LPRP0=LPROP
      THCK0=THICK
      NLPX0=NELPX(IPNLX)
      SPNX0=SPNXL(IPNLX)
      CALL    RAZERO  (CPNLD,150,5)
      CALL    RAZERO  (PNSRR,96,96)
      CALL    RAZERO  (PNSCV,54,150)
      DO 70 NUMEL=1,NELPX(IPNLX)*NELLA(KZ)
      IELEM=IELEM+1
      CALL    PNLASM  (0,IELEM,LNODS,NNODE,NDOFN,
     +                 IPNOD,SSTF,SKNOD,NPVRM,KONTE,
     +                 NCASE)
70    CONTINUE
      DO 80 IPVRM=2,NPVRM
      DO 80 JPVRM=1,IPVRM-1
80    PNSRR(IPVRM,JPVRM)=PNSRR(JPVRM,IPVRM)
      DO 85 IPVCN=2,NPVCN
      DO 85 JPVCN=1,IPVCN-1
85    PNSCV(IPVCN,JPVCN+NPVRM)=PNSCV(JPVCN,IPVCN+NPVRM)
      CALL    CNDNSP  (NCASE,NPVRM,NPVCN)
      IF(IT .NE. 3) GO TO 90
      CALL    SOLVPN  (NCASE,NPPRM,NPPCN,NPOIN,NDOFN,IPNOD,
     +                 ASDIS)
      GO TO 100
90    CALL    LYRNDV  (KZ,NPOIN,NPPRM,IPNOD,INDLY,NPPRF,NPPCF,
     +                 NDPRF,NDPCF)
      CALL    LYRASM  (KZ,NLVRM,NCASE,NDOFN,NPPRF,NPPCF,NDPRF,
     +                 NDPCF,IPNOD,INDLY)
100   CONTINUE
      RETURN
      END

      SUBROUTINE      LYRCSN  (IT,KZ,NDOFN,NCASE,NLKPR,NLKPC,
     +                         NCVRM,INDLY,NDLYR,NDLYC,INDCS,ASDIS)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/STFRR/ STFRR
      COMMON/STFCV/ STFCV
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/CCSLD/ CCSLD
      COMMON/CLYLD/ CLYLD
      DIMENSION       SLYRR(252,252),SLYCV(36,288),CLYLD(288,5),
     +                STFRR(252,252),STFCV(126,378),CCSLD(378,5),
     +                INDLY(153,3),INDCS(153),NDLYR(42,2),NDLYC(16,2),
     +                NDLRF(32),NDLCF(21),ASDIS(918,5)
      WRITE(8,111)
  111 FORMAT(' subroutine lyrcsn')
      NLVRM=NLKPR*NDOFN
      NLVCN=NLKPC*NDOFN
      DO 10 ILVRM=2,NLVRM
      DO 10 JLVRM=1,ILVRM-1
   10 SLYRR(ILVRM,JLVRM)=SLYRR(JLVRM,ILVRM)
      DO 20 ILVCN=2,NLVCN
      DO 20 JLVCN=1,ILVCN-1
   20 SLYCV(ILVCN,JLVCN+NLVRM)=SLYCV(JLVCN,ILVCN+NLVRM)
      CALL    CNDNSL  (NCASE,NLVRM,NLVCN)
      IF(IT .EQ. 2) GO TO 30
      CALL    CSNNDV  (KZ,NLKPR,NDLYR,INDCS,NLPRF,NLPCF,NDLRF,
     +                 NDLCF)
      CALL    CSNASN  (KZ,NCVRM,NCASE,NDOFN,NLPRF,NLPCF,NDLRF,
     +                 NDLCF,INDLY,INDCS)
      RETURN
   30 CALL    SOLVLY  (KZ,NCASE,NLKPR,NLKPC,NDOFN,NDLYR,NDLYC,
     +                 ASDIS)
      RETURN
      END

      SUBROUTINE      PNLNDV  (IELEM,NNODE,LNODS,INDPL,IPNOD,NPPRM,
     +                         NPPCN,KONTP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       LNODS(36,9),KONTP(153),IPNOD(153),INDPL(153)
      DO 20 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      KONTP(LNODE)=KONTP(LNODE)+1
      IF(KONTP(LNODE) .GT. 1) GO TO 20
      IVELE=INDPL(LNODE)
      IF(IVELE .GT. 0) GO TO 10
      NPPCN=NPPCN+1
      IPNOD(LNODE)=-NPPCN
      GO TO 20
   10 NPPRM=NPPRM+1
      IPNOD(LNODE)=NPPRM
   20 CONTINUE
      RETURN
      END

      SUBROUTINE      PNLASM  (IBASE,IELEM,LNODS,NNODE,NDOFN,
     +                         IPNOD,SSTF,SKNOD,NPVRM,KONTE,
     +                         NCASE)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PNSRR/ PNSRR
      COMMON/PNSCV/ PNSCV
      COMMON/GSTIF/ GSTIF
      COMMON/CLOAD/ CLOAD
      COMMON/CPNLD/ CPNLD
      DIMENSION       LNODS(36,9),IPNOD(153),SSTF(9),SKNOD(153),
     +                KONTE(153),PNSRR(96,96),PNSCV(54,150),
     +                CPNLD(150,5),CLOAD(918,5),GSTIF(54,54),
     +                IVREL(54),IEVRM(54),IVCEL(54),IEVCN(54)
      WRITE(8,111) IELEM
  111 FORMAT(' assembling element no. ',I3)
      NEVRM=0
      NEVCN=0
```

```
      IELVA=0
      DO 70 INODE=1,NNODE
      IF(IBASE .EQ. 0) GO TO 5
      IF(INODE .EQ. 1 .OR. INODE .EQ. 3 .OR. INODE .EQ. 5 .OR. INODE
     + .EQ. 7) SS=SSTF(INODE)*2.25
      IF(INODE .EQ. 2 .OR. INODE .EQ. 4 .OR. INODE .EQ. 6 .OR. INODE
     + .EQ.8) SS=SSTF(INODE)*1.125
      IF(INODE .EQ. 9) SS=SSTF(INODE)*.5625
    5 LNODE=LNODS(IELEM,INODE)
      IPOSN=(LNODE-1)*NDOFN
      KONTE(LNODE)=KONTE(LNODE)+1
      IF(IBASE .EQ. 1) SKNOD(LNODE)=SKNOD(LNODE)+SS
      IVPNL=IPNOD(LNODE)
      IF(IVPNL .GT. 0) GO TO 30
      IVPNL=(-IVPNL-1)*NDOFN
      DO 20 IDOFN=1,NDOFN
      IVPNL=IVPNL+1
      IPOSN=IPOSN+1
      DO 10 ICASE=1,NCASE
   10 CPNLD(IVPNL+NPVRM,ICASE)=CLOAD(IPOSN,ICASE)
      IELVA=IELVA+1
      NEVCN=NEVCN+1
      IVCEL(IELVA)=IVPNL
      IEVCN(NEVCN)=IELVA
   20 CONTINUE
      GO TO 70
   30 IVPNL=(IVPNL-1)*NDOFN
      DO 60 IDOFN=1,NDOFN
      IVPNL=IVPNL+1
      IPOSN=IPOSN+1
      IF(KONTE(LNODE) .GT. 1) GO TO 50
      DO 40 ICASE=1,NCASE
   40 CPNLD(IVPNL,ICASE)=CLOAD(IPOSN,ICASE)
   50 IELVA=IELVA+1
      NEVRM=NEVRM+1
      IVREL(IELVA)=IVPNL
      IEVRM(NEVRM)=IELVA
   60 CONTINUE
   70 CONTINUE
      DO 110  IEVAR=1,NEVRM
      IELVA=IEVRM(IEVAR)
      IVPNL=IVREL(IELVA)
      DO 90 JEVAR=1,IEVAR
      JELVA=IEVRM(JEVAR)
      JVPNL=IVREL(JELVA)
      IF(IVPNL .LE. JVPNL) GO TO 80
      PNSRR(JVPNL,IVPNL)=PNSRR(JVPNL,IVPNL)+GSTIF(IELVA,JELVA)
      GO TO 90
   80 PNSRR(IVPNL,JVPNL)=PNSRR(IVPNL,JVPNL)+GSTIF(IELVA,JELVA)
   90 CONTINUE
      DO 100  JEVAR=1,NEVCN
      JELVA=IEVCN(JEVAR)
      JVPNL=IVCEL(JELVA)
  100 PNSCV(JVPNL,IVPNL)=PNSCV(JVPNL,IVPNL)+GSTIF(IELVA,JELVA)
  110 CONTINUE
      DO 130  IEVAR=1,NEVCN
      IELVA=IEVCN(IEVAR)
      IVPNL=IVCEL(IELVA)
      DO 130  JEVAR=1,IEVAR
      JELVA=IEVCN(JEVAR)
      JVPNL=IVCEL(JELVA)
      IF(IVPNL .LE. JVPNL) GO TO 120
      IVPNO=IVPNL+NPVRM
      PNSCV(JVPNL,IVPNO)=PNSCV(JVPNL,IVPNO)+GSTIF(IELVA,JELVA)
      GO TO 130
  120 JVPNO=JVPNL+NPVRM
      PNSCV(IVPNL,JVPNO)=PNSCV(IVPNL,JVPNO)+GSTIF(IELVA,JELVA)
  130 CONTINUE
      RETURN
```

```
      END

      SUBROUTINE      CNDNSP  (NCASE,NPVRM,NPVCN)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PNSRR/ PNSRR
      COMMON/PNSCV/ PNSCV
      COMMON/CPNLD/ CPNLD
      DIMENSION       PNSRR(96,96),PNSCV(54,150),CPNLD(150,5)
      NPVAR=NPVRM+NPVCN
      DO 80 K=1,NPVCN
      LL=NPVAR-K
      KK=LL+1
      DO 70 L=1,LL
      IF(KK .LE. NPVRM) GO TO 10
      DUM=PNSCV(KK-NPVRM,L)/PNSCV(KK-NPVRM,KK)
      GO TO 20
10    DUM=PNSRR(KK,L)/PNSRR(KK,KK)
20    DO 60 M=1,L
      IF(L .LE. NPVRM .AND. KK .GT. NPVRM) GO TO 30
      IF(L .GT. NPVRM .AND. KK .GT. NPVRM) GO TO 40
      IF(L .LE. NPVRM .AND. KK .LE. NPVRM) GO TO 50
      PNSCV(L-NPVRM,M)=PNSCV(L-NPVRM,M)-PNSRR(KK,M)*DUM
      GO TO 60
30    PNSRR(L,M)=PNSRR(L,M)-PNSCV(KK-NPVRM,M)*DUM
      GO TO 60
40    PNSCV(L-NPVRM,M)=PNSCV(L-NPVRM,M)-PNSCV(KK-NPVRM,M)*DUM
      GO TO 60
50    PNSRR(L,M)=PNSRR(L,M)-PNSRR(KK,M)*DUM
60    CONTINUE
      DO 65 ICASE=1,NCASE
65    CPNLD(L,ICASE)=CPNLD(L,ICASE)-CPNLD(KK,ICASE)*DUM
70    CONTINUE
80    CONTINUE
      DO 90 K=2,LL
      DO 90 L=1,K-1
90    PNSRR(L,K)=PNSRR(K,L)
      RETURN
      END

      SUBROUTINE      LYRNDV  (KZ,NPOIN,NPPRM,IPNOD,INDLY,NPPRF,NPPCF,
     +                         NDPRF,NDPCF)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       IPNOD(153),INDLY(153,3),NDPRF(10),NDPCF(6)
      NPPRF=0
      NPPCF=0
      DO 40 IPPRM=1,NPPRM
      DO 10 IPOIN=1,NPOIN
10    IF(IPNOD(IPOIN) .EQ. IPPRM) GO TO 20
20    IF(INDLY(IPOIN,KZ) .GT. 0) GO TO 30
      NPPCF=NPPCF+1
      NDPCF(NPPCF)=IPOIN
      GO TO 40
30    NPPRF=NPPRF+1
      NDPRF(NPPRF)=IPOIN
40    CONTINUE
      RETURN
      END

      SUBROUTINE      LYRASM  (KZ,NLVRM,NCASE,NDOFN,NPPRF,NPPCF,NDPRF,
     +                         NDPCF,IPNOD,INDLY)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/PNSRR/ PNSRR
      COMMON/CLYLD/ CLYLD
      COMMON/CPNLD/ CPNLD
      DIMENSION       IPNOD(153),CPNLD(150,5),PNSRR(96,96),
     +                CLYLD(288,5),INDLY(153,3),NDPRF(10),NDPCF(6),
     +                SLYRR(252,252),SLYCV(36,288)
```

```
      DO 60 IPPRF=1,NPPRF
      IPOIN=NDPRF(IPPRF)
      IPPRM=IPNOD(IPOIN)
      IPVRM=(IPPRM-1)*NDOFN
      ILPRM=INDLY(IPOIN,KZ)
      ILVRM=(ILPRM-1)*NDOFN
      DO 60 IDOFN=1,NDOFN
      IPVRM=IPVRM+1
      ILVRM=ILVRM+1
      DO 10 ICASE=1,NCASE
10    CLYLD(ILVRM,ICASE)=CLYLD(ILVRM,ICASE)+CPNLD(IPVRM,ICASE)
      DO 40 JPPRF=1,IPPRF
      JPOIN=NDPRF(JPPRF)
      JPPRM=IPNOD(JPOIN)
      JPVRM=(JPPRM-1)*NDOFN
      JLPRM=INDLY(JPOIN,KZ)
      JLVRM=(JLPRM-1)*NDOFN
      DO 30 JDOFN=1,NDOFN
      JPVRM=JPVRM+1
      IF(JPVRM .GT. IPVRM) GO TO 40
      JLVRM=JLVRM+1
      IF(ILVRM .LE. JLVRM) GO TO 20
      SLYRR(JLVRM,ILVRM)=SLYRR(JLVRM,ILVRM)+PNSRR(IPVRM,JPVRM)
      GO TO 30
20    SLYRR(ILVRM,JLVRM)=SLYRR(ILVRM,JLVRM)+PNSRR(IPVRM,JPVRM)
30    CONTINUE
40    CONTINUE
      DO 50 JPPCF=1,NPPCF
      JPOIN=NDPCF(JPPCF)
      JPPRM=IPNOD(JPOIN)
      JPVRM=(JPPRM-1)*NDOFN
      JLPCN=-INDLY(JPOIN,KZ)
      JLVCN=(JLPCN-1)*NDOFN
      DO 50 JDOFN=1,NDOFN
      JPVRM=JPVRM+1
      JLVCN=JLVCN+1
50    SLYCV(JLVCN,ILVRM)=SLYCV(JLVCN,ILVRM)+PNSRR(IPVRM,JPVRM)
60    CONTINUE
      DO 100 IPPCF=1,NPPCF
      IPOIN=NDPCF(IPPCF)
      IPPRM=IPNOD(IPOIN)
      IPVRM=(IPPRM-1)*NDOFN
      ILPCN=-INDLY(IPOIN,KZ)
      ILVCN=(ILPCN-1)*NDOFN
      DO 100 IDOFN=1,NDOFN
      IPVRM=IPVRM+1
      ILVCN=ILVCN+1
      DO 70 ICASE=1,NCASE
70    CLYLD(ILVCN+NLVRM,ICASE)=CLYLD(ILVCN+NLVRM,ICASE)+
     +                         CPNLD(IPVRM,ICASE)
      DO 90 JPPCF=1,IPPCF
      JPOIN=NDPCF(JPPCF)
      JPPRM=IPNOD(JPOIN)
      JPVRM=(JPPRM-1)*NDOFN
      JLPCN=-INDLY(JPOIN,KZ)
      JLVCN=(JLPCN-1)*NDOFN
      DO 90 JDOFN=1,NDOFN
      JPVRM=JPVRM+1
      IF(JPVRM .GT. IPVRM) GO TO 90
      JLVCN=JLVCN+1
      IF(ILVCN .LE. JLVCN) GO TO 80
      SLYCV(JLVCN,ILVCN+NLVRM)=SLYCV(JLVCN,ILVCN+NLVRM)+
     +                         PNSRR(IPVRM,JPVRM)
      GO TO 90
80    SLYCV(ILVCN,JLVCN+NLVRM)=SLYCV(ILVCN,JLVCN+NLVRM)+
     +                         PNSRR(IPVRM,JPVRM)
90    CONTINUE
100   CONTINUE
      RETURN
```

```
      END

      SUBROUTINE      CNDNSL  (NCASE,NLVRM,NLVCN)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SLYRR/ SLYRR
      COMMON/SLYCV/ SLYCV
      COMMON/CLYLD/ CLYLD
      DIMENSION       SLYRR(252,252),SLYCV(36,288),CLYLD(288,5)
      WRITE(8,111)
111   FORMAT(' subroutine cndnsl')
      NLVAR=NLVRM+NLVCN
      DO 80 K=1,NLVCN
      LL=NLVAR-K
      KK=LL+1
      DO 70 L=1,LL
      IF(KK .LE. NLVRM) GO TO 10
      DUM=SLYCV(KK-NLVRM,L)/SLYCV(KK-NLVRM,KK)
      GO TO 20
 10   DUM=SLYRR(KK,L)/SLYRR(KK,KK)
 20   DO 60 M=1,L
      IF(L .LE. NLVRM .AND. KK .GT. NLVRM) GO TO 30
      IF(L .GT. NLVRM .AND. KK .GT. NLVRM) GO TO 40
      IF(L .LE. NLVRM .AND. KK .LE. NLVRM) GO TO 50
      SLYCV(L-NLVRM,M)=SLYCV(L-NLVRM,M)-SLYRR(KK,M)*DUM
      GO TO 60
 30   SLYRR(L,M)=SLYRR(L,M)-SLYCV(KK-NLVRM,M)*DUM
      GO TO 60
 40   SLYCV(L-NLVRM,M)=SLYCV(L-NLVRM,M)-SLYCV(KK-NLVRM,M)*DUM
      GO TO 60
 50   SLYRR(L,M)=SLYRR(L,M)-SLYRR(KK,M)*DUM
 60   CONTINUE
      DO 65 ICASE=1,NCASE
 65   CLYLD(L,ICASE)=CLYLD(L,ICASE)-CLYLD(KK,ICASE)*DUM
 70   CONTINUE
 80   CONTINUE
      DO 90 K=2,LL
      DO 90 L=1,K-1
 90   SLYRR(L,K)=SLYRR(K,L)
      RETURN
      END

      SUBROUTINE      CSNNDV  (KZ,NLKPR,NDLYR,INDCS,NLPRF,NLPCF,NDLRF,
     +                         NDLCF)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       NDLYR(42,2),INDCS(153),NDLRF(32),NDLCF(21)
      WRITE(8,111)
111   FORMAT(' subroutine csnndv')
      NLPRF=0
      NLPCF=0
      DO 40 ILKPR=1,NLKPR
      IPOIN=NDLYR(ILKPR,KZ)
      IF(INDCS(IPOIN) .GT. 0) GO TO 30
      NLPCF=NLPCF+1
      NDLCF(NLPCF)=IPOIN
      GO TO 40
 30   NLPRF=NLPRF+1
      NDLRF(NLPRF)=IPOIN
 40   CONTINUE
      RETURN
      END

      SUBROUTINE      CSNASM  (KZ,NCVRM,NCASE,NDOFN,NLPRF,NLPCF,NDLRF,
     +                         NDLCF,INDLY,INDCS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/STFRR/ STFRR
      COMMON/STFCV/ STFCV
      COMMON/SLYRR/ SLYRR
      COMMON/CCSLD/ CCSLD
      COMMON/CLYLD/ CLYLD
```

```
      DIMENSION       INDLY(153,3),CLYLD(288,5),SLYRR(252,252),
     +                CCSLD(378,5),STFRR(252,252),STFCV(126,378),
     +                INDCS(153),NDLRF(32),NDLCF(21)
      WRITE(8,111)
111   FORMAT(' subroutine csnasm')
      DO 60 ILPRF=1,NLPRF
      IPOIN=NDLRF(ILPRF)
      ILPRM=INDLY(IPOIN,KZ)
      ILVRM=(ILPRM-1)*NDOFN
      ICPRM=INDCS(IPOIN)
      ICVRM=(ICPRM-1)*NDOFN
      DO 60 IDOFN=1,NDOFN
      ILVRM=ILVRM+1
      ICVRM=ICVRM+1
      DO 10 ICASE=1,NCASE
 10   CCSLD(ICVRM,ICASE)=CCSLD(ICVRM,ICASE)+CLYLD(ILVRM,ICASE)
      DO 40 JLPRF=1,ILPRF
      JPOIN=NDLRF(JLPRF)
      JLPRM=INDLY(JPOIN,KZ)
      JLVRM=(JLPRM-1)*NDOFN
      JCPRM=INDCS(JPOIN)
      JCVRM=(JCPRM-1)*NDOFN
      DO 30 JDOFN=1,NDOFN
      JLVRM=JLVRM+1
      IF(JLVRM .GT. ILVRM) GO TO 40
      JCVRM=JCVRM+1
      IF(ICVRM .LE. JCVRM) GO TO 20
      STFRR(JCVRM,ICVRM)=STFRR(JCVRM,ICVRM)+SLYRR(ILVRM,JLVRM)
      GO TO 30
 20   STFRR(ICVRM,JCVRM)=STFRR(ICVRM,JCVRM)+SLYRR(ILVRM,JLVRM)
 30   CONTINUE
 40   CONTINUE
      DO 50 JLPCF=1,NLPCF
      JPOIN=NDLCF(JLPCF)
      JLPRM=INDLY(JPOIN,KZ)
      JLVRM=(JLPRM-1)*NDOFN
      JCPCN=-INDCS(JPOIN)
      JCVCN=(JCPCN-1)*NDOFN
      DO 50 JDOFN=1,NDOFN
      JLVRM=JLVRM+1
      JCVCN=JCVCN+1
 50   STFCV(JCVCN,ICVRM)=STFCV(JCVCN,ICVRM)+SLYRR(ILVRM,JLVRM)
 60   CONTINUE
      DO 100 ILPCF=1,NLPCF
      IPOIN=NDLCF(ILPCF)
      ILPRM=INDLY(IPOIN,KZ)
      ILVRM=(ILPRM-1)*NDOFN
      ICPCN=-INDCS(IPOIN)
      ICVCN=(ICPCN-1)*NDOFN
      DO 100 IDOFN=1,NDOFN
      ILVRM=ILVRM+1
      ICVCN=ICVCN+1
      DO 70 ICASE=1,NCASE
 70   CCSLD(ICVCN+NCVRM,ICASE)=CCSLD(ICVCN+NCVRM,ICASE)+
     +                           CLYLD(ILVRM,ICASE)
      DO 95 JLPCF=1,ILPCF
      JPOIN=NDLCF(JLPCF)
      JLPRM=INDLY(JPOIN,KZ)
      JLVRM=(JLPRM-1)*NDOFN
      JCPCN=-INDCS(JPOIN)
      JCVCN=(JCPCN-1)*NDOFN
      DO 90 JDOFN=1,NDOFN
      JLVRM=JLVRM+1
      IF(JLVRM .GT. ILVRM) GO TO 95
      JCVCN=JCVCN+1
      IF(ICVCN .LE. JCVCN) GO TO 80
      STFCV(JCVCN,ICVCN+NCVRM)=STFCV(JCVCN,ICVCN+NCVRM)+
     +                           SLYRR(ILVRM,JLVRM)
      GO TO 90
```

```
80    STFCV(ICVCN,JCVCN+NCVRM)=STFCV(ICVCN,JCVCN+NCVRM)+
     +                             SLYRR(ILVRM,JLVRM)
90    CONTINUE
95    CONTINUE
100   CONTINUE
      RETURN
      END

      SUBROUTINE      BMATMA (BMEMB,CARTD,KNODE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION      BMEMB(3,2),CARTD(2,9)
      DNKD1=CARTD(1,KNODE)
      DNKD2=CARTD(2,KNODE)
      CALL    RAZERO  (BMEMB,3,2)
      BMEMB(1,1)=DNKD1
      BMEMB(2,2)=DNKD2
      BMEMB(3,1)=DNKD2
      BMEMB(3,2)=DNKD1
      RETURN
      END

      SUBROUTINE      BMATPB (BFLEX,BSHER,CARTD,KNODE,SHAPE,
     +                        IFFLE,IFSHE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION      BFLEX(3,3),BSHER(2,3),CARTD(2,9),SHAPE(9)
      DNKD1=CARTD(1,KNODE)
      DNKD2=CARTD(2,KNODE)
10    IF(IFFLE .EQ. 0) GO TO 20
      CALL    RAZERO  (BFLEX,3,3)
      BFLEX(1,3)=DNKD1
      BFLEX(2,2)=-DNKD2
      BFLEX(3,2)=-DNKD1
      BFLEX(3,3)=DNKD2
20    IF(IFSHE .EQ. 0) RETURN
      CALL    RAZERO  (BSHER,2,3)
      BSHER(1,1)=DNKD1
      BSHER(1,3)=SHAPE(KNODE)
      BSHER(2,1)=DNKD2
      BSHER(2,2)=-SHAPE(KNODE)
      RETURN
      END

      SUBROUTINE IVZERO(IVECT,NSIZE)
      DIMENSION IVECT(NSIZE)
      DO 5 ISIZE=1,NSIZE
5     IVECT(ISIZE)=0
      RETURN
      END

      SUBROUTINE IAZERO(IVECT,NSIZE,MSIZE)
      DIMENSION IVECT(NSIZE,MSIZE)
      DO 5 ISIZE=1,NSIZE
      DO 5 JSIZE=1,MSIZE
5     IVECT(ISIZE,JSIZE)=0
      RETURN
      END

      SUBROUTINE RVZERO(RVECT,NSIZE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RVECT(NSIZE)
      DO 5 ISIZE=1,NSIZE
5     RVECT(ISIZE)=0.
      RETURN
      END

      SUBROUTINE RAZERO(RVECT,NSIZE,MSIZE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RVECT(NSIZE,MSIZE)
      DO 5 ISIZE=1,NSIZE
```

```
      DO 5 JSIZE=1,MSIZE
 5    RVECT(ISIZE,JSIZE)=0.
      RETURN
      END

      SUBROUTINE      STIFMA  (THICK,LNODS,LPROP,NGAUM,
     +                         NNODE,PROPS,NDOFN,IELEM,L1,L2)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ESTIF/ ESTIF
      COMMON/COORD/ COORD
      DIMENSION   CARTD(2,9),COORD(153,3),DERIV(2,9),ELCOD(2,9),
     +            ESTIF(54,54),GPCOD(2,9),LNODS(36,9),POSGP(3),
     +            PROPS(10,3),SHAPE(9),WEIGP(3),DMEMB(3,3),
     +            BMEMI(3,3),BMEMJ(3,3)
      IELEM=IELEM+1
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      ELCOD(1,INODE)=COORD(LNODE,L1)
      IF(L1 .EQ. 2) ELCOD(1,INODE)=-COORD(LNODE,L1)
 10   ELCOD(2,INODE)=COORD(LNODE,L2)
      CALL    RAZERO  (ESTIF,54,54)
      KGASP=0
      CALL    GAUSSQ  (NGAUM,POSGP,WEIGP)
      DO 40 IGAUS=1,NGAUM
      EXISP=POSGP(IGAUS)
      DO 40 JGAUS=1,NGAUM
      KGASP=KGASP+1
      ETASP=POSGP(JGAUS)
      CALL    SFR2    (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL    JACOB2  (CARTD,DERIV,DJACB,ELCOD,GPCOD,IELEM,
     +                 KGASP,NNODE,SHAPE)
      DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      CALL    MODMA   (DMEMB,LPROP,PROPS)
      DO 30 INODE=1,NNODE
      CALL    BMATMA  (BMEMI,CARTD,INODE)
      DO 30 JNODE=INODE,NNODE
      CALL    BMATMA  (BMEMJ,CARTD,JNODE)
 30   CALL    SUBMA   (NDOFN,BMEMI,BMEMJ,DAREA,DMEMB,
     +                 INODE,JNODE,2,3,2,THICK)
 40   CONTINUE
      DAREA=(ELCOD(1,3)-ELCOD(1,1))*(ELCOD(2,7)-ELCOD(2,1))
      YOUNG=PROPS(LPROP,1)
      CONST=.3*YOUNG*DAREA*THICK
      DO 50 INODE=1,NNODE
      IELVA=INODE*NDOFN
      DO 50 JNODE=INODE,NNODE
      JELVA=JNODE*NDOFN
      ESTIF(IELVA,JELVA)=-.125*CONST
      IF(IELVA .EQ. JELVA) ESTIF(IELVA,JELVA)=CONST
 50   CONTINUE
      RETURN
      END

      SUBROUTINE      STIFFB  (THICK,LNODS,LPROP,NGAUB,NGAUS,
     +                         NNODE,PROPS,NDOFN,IELEM,L1,L2)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ESTIF/ ESTIF
      COMMON/COORD/ COORD
      DIMENSION   CARTD(2,9),COORD(153,3),DERIV(2,9),ELCOD(2,9),
     +            ESTIF(54,54),GPCOD(2,9),LNODS(36,9),POSGP(3),
     +            PROPS(10,3),SHAPE(9),WEIGP(3),DFLEX(3,3),DSHER(2,2),
     +            BFLEI(3,3),BFLEJ(3,3),BSHEI(2,3),BSHEJ(2,3)
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      ELCOD(1,INODE)=COORD(LNODE,L1)
      IF(L1 .EQ. 2) ELCOD(1,INODE)=-COORD(LNODE,L1)
 10   ELCOD(2,INODE)=COORD(LNODE,L2)
      KGASP=0
      CALL    GAUSSQ  (NGAUB,POSGP,WEIGP)
```

```
      DO 40 IGAUS=1,NGAUB
      EXISP=POSGP(IGAUS)
      DO 40 JGAUS=1,NGAUB
      KGASP=KGASP+1
      ETASP=POSGP(JGAUS)
      CALL     SFR2     (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL     JACOB2   (CARTD,DERIV,DJACB,ELCOD,GPCOD,IELEM,
     +                   KGASP,NNODE,SHAPE)
      DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      CALL     MODPB    (THICK,DFLEX,DSHER,LPROP,PROPS,1,0)
      DO 30 INODE=1,NNODE
      CALL     BMATPB   (BFLEI,BSHEI,CARTD,INODE,SHAPE,1,0)
      DO 30 JNODE=INODE,NNODE
      CALL     BMATPB   (BFLEJ,BSHEJ,CARTD,JNODE,SHAPE,1,0)
30    CALL     SUBPB    (NDOFN,BFLEI,BFLEJ,DAREA,DFLEX,
     +                   INODE,JNODE,3,3,3)
40    CONTINUE
      KGASP=0
      CALL     GAUSSQ   (NGAUS,POSGP,WEIGP)
      DO 60 IGAUS=1,NGAUS
      EXISP=POSGP(IGAUS)
      DO 60 JGAUS=1,NGAUS
      KGASP=KGASP+1
      ETASP=POSGP(JGAUS)
      CALL     SFR2     (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL     JACOB2   (CARTD,DERIV,DJACB,ELCOD,GPCOD,IELEM,
     +                   KGASP,NNODE,SHAPE)
      DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      CALL     MODPB    (THICK,DFLEX,DSHER,LPROP,PROPS,0,1)
      DO 50 INODE=1,NNODE
      CALL     BMATPB   (BFLEI,BSHEI,CARTD,INODE,SHAPE,0,1)
      DO 50 JNODE=INODE,NNODE
      CALL     BMATPB   (BFLEJ,BSHEJ,CARTD,JNODE,SHAPE,0,1)
      CALL     SUBPB    (NDOFN,BSHEI,BSHEJ,DAREA,DSHER,
     +                   INODE,JNODE,3,2,3)
50    CONTINUE
60    CONTINUE
      RETURN
      END

      SUBROUTINE       GAUSSQ   (NGAUS,POSGP,WEIGP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION POSGP(3),WEIGP(3)
      GO TO (1,2,3) NGAUS
1     POSGP(1)=0.
      WEIGP(1)=2.
      RETURN
2     POSGP(2)=1./SQRT(3.)
      POSGP(1)=-POSGP(2)
      WEIGP(1)=1.
      WEIGP(2)=1.
      RETURN
3     POSGP(3)=SQRT(.6)
      POSGP(2)=0.
      POSGP(1)=-POSGP(3)
      WEIGP(1)=5./9.
      WEIGP(2)=8./9.
      WEIGP(3)=5./9.
      RETURN
      END

      SUBROUTINE    SFR2    (DERIV,ETASP,EXISP,SHAPE,NNODE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DERIV(2,9),SHAPE(9)
      S=EXISP
      T=ETASP
      S2=S*2.
      T2=T*2.
      TT=T*T
```

```
      SS=S*S
      ST=S*T
      SST=S*S*T
      STT=S*T*T
      ST2=S*T*2.
      SHAPE(1)=(-1.+ST+SS+TT-SST-STT)/4.
      SHAPE(2)=(1.-T-SS+SST)/2.
      SHAPE(3)=(-1.-ST+SS+TT-SST+STT)/4.
      SHAPE(4)=(1.+S-TT-STT)/2.
      SHAPE(5)=(-1.+ST+SS+TT+SST+STT)/4.
      SHAPE(6)=(1.+T-SS-SST)/2.
      SHAPE(7)=(-1.-ST+SS+TT+SST-STT)/4.
      SHAPE(8)=(1.-S-TT+STT)/2.
      DERIV(1,1)=(T+S2-ST2-TT)/4.
      DERIV(1,2)=-S+ST
      DERIV(1,3)=(-T+S2-ST2+TT)/4.
      DERIV(1,4)=(1.-TT)/2.
      DERIV(1,5)=(T+S2+ST2+TT)/4.
      DERIV(1,6)=-S-ST
      DERIV(1,7)=(-T+S2+ST2-TT)/4.
      DERIV(1,8)=(-1.+TT)/2.
      DERIV(2,1)=(S+T2-SS-ST2)/4.
      DERIV(2,2)=(-1.+SS)/2.
      DERIV(2,3)=(-S+T2-SS+ST2)/4.
      DERIV(2,4)=-T-ST
      DERIV(2,5)=(S+T2+SS+ST2)/4.
      DERIV(2,6)=(1.-SS)/2.
      DERIV(2,7)=(-S+T2+SS-ST2)/4.
      DERIV(2,8)=-T+ST
      SHAPE(9)=(1.-SS)*(1.-TT)
      DERIV(1,9)=-S2*(1.-TT)
      DERIV(2,9)=-T2*(1.-SS)
      DO 10 I=1,7,2
      SHAPE(I)=SHAPE(I)+SHAPE(9)/4.
      DO 10 J=1,2
10    DERIV(J,I)=DERIV(J,I)+DERIV(J,9)/4.
      DO 20 I=2,8,2
      SHAPE(I)=SHAPE(I)-SHAPE(9)/2.
      DO 20 J=1,2
20    DERIV(J,I)=DERIV(J,I)-DERIV(J,9)/2.
      RETURN
      END

      SUBROUTINE      JACOB2 (CARTD,DERIV,DJACB,ELCOD,GPCOD,IELEM,
     +                        KGASP,NNODE,SHAPE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       CARTD(2,9),DERIV(2,9),ELCOD(2,9),
     +                GPCOD(2,9),SHAPE(9),XJACI(2,2),XJACM(2,2)
      DO 2 IDIME=1,2
      GPCOD(IDIME,KGASP)=0.
      DO 2 INODE=1,NNODE
      GPCOD(IDIME,KGASP)=GPCOD(IDIME,KGASP)+ELCOD(IDIME,INODE)*
     + SHAPE(INODE)
2     CONTINUE
      DO 4 IDIME=1,2
      DO 4 JDIME=1,2
      XJACM(IDIME,JDIME)=0.
      DO 4 INODE=1,NNODE
      XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME)+DERIV(IDIME,INODE)*
     + ELCOD(JDIME,INODE)
4     CONTINUE
      DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1)
      IF(DJACB) 6,6,8
6     WRITE(6,900) IELEM
      STOP
8     CONTINUE
      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
      DO 10 IDIME=1,2
      DO 10 INODE=1,NNODE
      CARTD(IDIME,INODE)=0.
      DO 10 JDIME=1,2
      CARTD(IDIME,INODE)=CARTD(IDIME,INODE)+XJACI(IDIME,JDIME)*
     + DERIV(JDIME,INODE)
10    CONTINUE
900   FORMAT(//,' PROGRAM HALTED IN SUBROTINE JACOB2',/,11X,
     + ' ZERO OR NEGATIVE AREA',/,10X,' ELEMENT NUMBER ',I5)
      RETURN
      END

      SUBROUTINE     MODMA   (DMEMB,LPROP,PROPS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION      DMEMB(3,3),PROPS(10,3)
      YOUNG=PROPS(LPROP,1)
      POISS=PROPS(LPROP,2)
      CALL    RAZERO (DMEMB,3,3)
      CONST=YOUNG/(1-POISS*POISS)
      DMEMB(1,1)=CONST
      DMEMB(2,2)=CONST
      DMEMB(1,2)=CONST*POISS
      DMEMB(2,1)=CONST*POISS
      DMEMB(3,3)=CONST*(1.-POISS)/2.
      RETURN
      END

      SUBROUTINE     MODPB  (THICK,DFLEX,DSHER,LPROP,PROPS,
     +                        IFFLE,IFSHE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION      DFLEX(3,3),DSHER(2,2),PROPS(10,3)
      YOUNG=PROPS(LPROP,1)
      POISS=PROPS(LPROP,2)
10    IF(IFFLE .EQ. 0) GO TO 20
      CALL    RAZERO (DFLEX,3,3)
      CONST=(YOUNG*THICK**3)/(12.*(1-POISS*POISS))
      DFLEX(1,1)=CONST
      DFLEX(2,2)=CONST
      DFLEX(1,2)=CONST*POISS
      DFLEX(2,1)=CONST*POISS
      DFLEX(3,3)=CONST*(1.-POISS)/2.
20    IF(IFSHE .EQ. 0) RETURN
      CALL    RAZERO  (DSHER,2,2)
      DSHER(1,1)=(YOUNG*THICK)/(2.4+2.4*POISS)
      DSHER(2,2)=DSHER(1,1)
      RETURN
      END

      SUBROUTINE     SUBMA   (NDOFN,BIMAT,BJMAT,DAREA,DMATX,
     +                         INODE,JNODE,NCOLI,NROIJ,NCOLJ,
     +                         THICK)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ESTIF/ ESTIF
      DIMENSION      BIMAT(NROIJ,NCOLI),BJMAT(NROIJ,NCOLJ),
     +               DMATX(NROIJ,NROIJ),DBMAT(3,3),
     +               ESTIF(54,54),SMSTF(2,2)
      DO 10 J=1,NCOLJ
      DO 10 I=1,NROIJ
      DBMAT(I,J)=0.
      DO 10 K=1,NROIJ
10    DBMAT(I,J)=DBMAT(I,J)+DMATX(I,K)*BJMAT(K,J)
      DO 20 J=1,NCOLJ
      DO 20 I=1,NCOLI
      SMSTF(I,J)=0.
      DO 20 K=1,NROIJ
20    SMSTF(I,J)=SMSTF(I,J)+BIMAT(K,I)*DBMAT(K,J)
      IFROW=(INODE-1)*NDOFN
      JFCOL=(JNODE-1)*NDOFN
```

```
      DO 30 I=1,NCOLI
      IRSUB=IFROW+I
      DO 30 J=1,NCOLJ
      JCSUB=JFCOL+J
30    ESTIF(IRSUB,JCSUB)=ESTIF(IRSUB,JCSUB)+SMSTF(I,J)*DAREA*THICK
      RETURN
      END

      SUBROUTINE   SUBPB     (NDOFN,BIMAT,BJMAT,DAREA,DMATX,
     +                        INODE,JNODE,NCOLI,NROIJ,NCOLJ)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ESTIF/ ESTIF
      DIMENSION       BIMAT(NROIJ,NCOLI),BJMAT(NROIJ,NCOLJ).
     +                DMATX(NROIJ,NROIJ),DBMAT(3,3),
     +                ESTIF(54,54),SBSTF(3,3)
      DO 10 J=1,NCOLJ
      DO 10 I=1,NROIJ
      DBMAT(I,J)=0.
      DO 10 K=1,NROIJ
10    DBMAT(I,J)=DBMAT(I,J)+DMATX(I,K)*BJMAT(K,J)
      DO 20 J=1,NCOLJ
      DO 20 I=1,NCOLI
      SBSTF(I,J)=0.
      DO 20 K=1,NROIJ
20    SBSTF(I,J)=SBSTF(I,J)+BIMAT(K,I)*DBMAT(K,J)
      IFROW=(INODE-1)*NDOFN+2
      JFCOL=(JNODE-1)*NDOFN+2
      DO 30 I=1,NCOLI
      IRSUB=IFROW+I
      DO 30 J=1,NCOLJ
      JCSUB=JFCOL+J
30    ESTIF(IRSUB,JCSUB)=ESTIF(IRSUB,JCSUB)+SBSTF(I,J)*DAREA
      RETURN
      END

      SUBROUTINE      TRANSL  (NNODE,NEVAB)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ESTIF/ ESTIF
      COMMON/GSTIF/ GSTIF
      DIMENSION       ESTIF(54,54),GSTIF(54,54)
      IROW=1
      DO 20 INODE=1,NNODE
      DO 20 IK=1,2
      JCOL=IROW
      DO 10 JNODE=INODE,NNODE
      DO 10 JK=1,2
      IF(IK .EQ. 2 .AND. JK .EQ. 2 .AND. JNODE .EQ. NNODE) GO TO 15
      GSTIF(IROW,JCOL)=      ESTIF(IROW,JCOL)
      GSTIF(IROW,JCOL+1)=  -ESTIF(IROW,JCOL+2)
      GSTIF(IROW,JCOL+2)=   ESTIF(IROW,JCOL+1)
      GSTIF(IROW+1,JCOL)=  -ESTIF(IROW+2,JCOL)
      GSTIF(IROW+1,JCOL+1)= ESTIF(IROW+2,JCOL+2)
      GSTIF(IROW+1,JCOL+2)=-ESTIF(IROW+2,JCOL+1)
      GSTIF(IROW+2,JCOL)=   ESTIF(IROW+1,JCOL)
      GSTIF(IROW+2,JCOL+1)=-ESTIF(IROW+1,JCOL+2)
      GSTIF(IROW+2,JCOL+2)= ESTIF(IROW+1,JCOL+1)
      JCOL=JCOL+3
10    CONTINUE
15    IROW=IROW+3
20    CONTINUE
      DO 30 IROW=4,NEVAB
      DO 30 JCOL=1,IROW-1
30    GSTIF(IROW,JCOL)=GSTIF(JCOL,IROW)
      RETURN
      END

      SUBROUTINE      TRANSW  (NNODE,NEVAB)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ESTIF/ ESTIF
```

```
      COMMON/GSTIF/ GSTIF
      DIMENSION       ESTIF(54,54),GSTIF(54,54)
      IROW=1
      DO 20 INODE=1,NNODE
      DO 20 IK=1,2
      JCOL=IROW
      DO 10 JNODE=INODE,NNODE
      DO 10 JK=1,2
      IF(IK .EQ. 2 .AND. JK .EQ. 2 .AND. JNODE .EQ. NNODE) GO TO 15
      GSTIF(IROW,JCOL)=     ESTIF(IROW+2,JCOL+2)
      GSTIF(IROW,JCOL+1)=   ESTIF(IROW+2,JCOL)
      GSTIF(IROW,JCOL+2)=  -ESTIF(IROW+2,JCOL+1)
      GSTIF(IROW+1,JCOL)=   ESTIF(IROW,JCOL+2)
      GSTIF(IROW+1,JCOL+1)= ESTIF(IROW,JCOL)
      GSTIF(IROW+1,JCOL+2)=-ESTIF(IROW,JCOL+1)
      GSTIF(IROW+2,JCOL)=  -ESTIF(IROW+1,JCOL+2)
      GSTIF(IROW+2,JCOL+1)=-ESTIF(IROW+1,JCOL)
      GSTIF(IROW+2,JCOL+2)= ESTIF(IROW+1,JCOL+1)
      JCOL=JCOL+3
10    CONTINUE
15    IROW=IROW+3
20    CONTINUE
      DO 30 IROW=4,NEVAB
      DO 30 JCOL=1,IROW-1
30    GSTIF(IROW,JCOL)=GSTIF(JCOL,IROW)
      RETURN
      END

      SUBROUTINE      LOADS   (NCASE,LNODS,MATNO,NELEM,
     +                          NEVAB,NGAUB,NNODE,PROPS,NDOFN,
     +                          NPNLX,NPNLY,NLAYR,NELPX,NELPY,
     +                          NELLA,THPNX,THPNY,HIGHT,ZHIGH,THCBS,
     +                          SPNXL,SPNYW,XLMOD,YWMOD)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CLOAD/ CLOAD
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
      DIMENSION       COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                ELOAD(36,54),CLOAD(918,5),WBTHD(3,13),
     +                SBTHD(3,13),WATLV(13,5),SOILV(13,5),ISOTP(13,5),
     +                FORCE(8,6),FCORD(8,3),FACT1(5),FACT2(5),
     +                FACT3(5),FACT4(5,8),NELPX(3),NELPY(3),NELLA(3),
     +                HIGHT(3),THPNX(3,4),THPNY(4,3),
     +                SPNXL(3),SPNYW(3),SOPRP(10,7)
      WRITE(8,111)
111   FORMAT(' subroutine loads')
      CALL     RAZERO  (CLOAD,918,5)
      CALL     LOADIN  (NPNLX,NPNLY,NCASE,NDOFN,NCOMP,WGAMA,
     +                  WATLV,SOILV,ISOTP,FORCE,FCORD,FACT1,
     +                  FACT2,FACT3,FACT4,SOPRP,NPTLD)
      DO 1001 ICASE=1,NCASE
      CALL     RAZERO  (ELOAD,36,54)
      DWFAC=FACT1(ICASE)
      WPFAC=FACT2(ICASE)
      SPFAC=FACT3(ICASE)
      DO 10 ICOMP=1,NCOMP
10    CALL     HEAD    (ICOMP,ICASE,NLAYR,HIGHT,ZHIGH,WATLV,SOILV,
     +                  THCBS,WBTHD,SBTHD)
      CALL     LODBAS  (ICASE,LNODS,MATNO,PROPS,
     +                  NGAUB,NNODE,NDOFN,THCBS,NPNLX,NPNLY,
     +                  NELPX,NELPY,DWFAC,WPFAC,SPFAC,WATLV,
     +                  SOILV,WGAMA,ISOTP,SOPRP,IELEM)
      CALL     LODWAL  (ICASE,LNODS,MATNO,PROPS,
     +                  NGAUB,NNODE,NDOFN,NPNLX,NPNLY,NLAYR,
     +                  NELPX,NELPY,NELLA,THPNX,THPNY,HIGHT,
     +                  DWFAC,WPFAC,SPFAC,WBTHD,SBTHD,ISOTP,
     +                  SOPRP,WGAMA,IELEM)
      NELNB=IELEM
      DO 300 IPTLD=1,NPTLD
```

```
      FACT=FACT4(ICASE,IPTLD)
      IF(FACT .EQ. 0.) GO TO 300
      CALL     CONLOC  (IPTLD,NPNLX,NPNLY,NLAYR,NELPX,NELPY,
     +                  NELLA,SPNXL,SPNYW,HIGHT,XLNOD,YWNOD,
     +                  ZHIGH,THCBS,FCORD,NELMB,IELEM,IIX)
      CALL     CONLOD  (IELEM,IIX,IPTLD,NNODE,NDOFN,LNODS,
     +                  FORCE,FCORD,FACT)
  300 CONTINUE
      WRITE(6,925) ICASE
  925 FORMAT(//10X,'LOAD CASE NO.',I2,/10X,15(1H*),
     + //5X,' TOTAL NODAL FORCES FOR EACH ELEMENT')
      DO 115 IELEM=1,NELEM
  115 WRITE(6,935) IELEM,(ELOAD(IELEM,IELVA),IELVA=1,NEVAB)
  935 FORMAT(/1X,I4,3X,6E12.4/(8X,6E12.4))
      DO 125 IELEM=1,NELEM
      IELVA=0
      DO 125 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      IPOSN=(LNODE-1)*NDOFN
      DO 125 IDOFN=1,NDOFN
      IELVA=IELVA+1
      IPOSN=IPOSN+1
      IF(ELOAD(IELEM,IELVA) .EQ. 0.) GO TO 125
      CLOAD(IPOSN,ICASE)=CLOAD(IPOSN,ICASE)+ELOAD(IELEM,IELVA)
  125 CONTINUE
 1001 CONTINUE
      RETURN
      END

      SUBROUTINE       LODBAS  (ICASE,LNODS,MATNO,PROPS,
     +                          NGAUB,NNODE,NDOFN,THCBS,NPNLX,NPNLY,
     +                          NELPX,NELPY,DWFAC,WPFAC,SPFAC,WATLV,
     +                          SOILV,WGAMA,ISOTP,SOPRP,IELEM)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
      DIMENSION        COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                 ELOAD(36,54),NELPX(3),NELPY(3),
     +                 WATLV(13,5),SOILV(13,5),ISOTP(13,5),SOPRP(10,7)
      IELEM=0
      L1=1
      L2=2
      ICOMP=4
      DO 30 JY=1,NPNLY
      DO 30 IX=1,NPNLX
      ICOMP=ICOMP+1
      LPROP=MATNO(IELEM+1)
      BGAMA=PROPS(LPROP,3)
      CALL     BASEPR  (ICOMP,ICASE,THCBS,WATLV,SOILV,ISOTP,SOPRP,
     +                  PRNET,WGAMA,BGAMA,DWFAC,WPFAC,SPFAC)
      DO 20 I=1,NELPX(IX)
      DO 20 J=1,NELPY(JY)
      IELEM=IELEM+1
      CALL     GRAVLD  (IELEM,L1,L2,PRNET,LNODS,NGAUB,NNODE,
     +                  NDOFN)
   20 CONTINUE
   30 CONTINUE
      RETURN
      END

      SUBROUTINE       LODWAL  (ICASE,LNODS,MATNO,PROPS,
     +                          NGAUB,NNODE,NDOFN,NPNLX,NPNLY,NLAYR,
     +                          NELPX,NELPY,NELLA,THPNX,THPNY,HIGHT,
     +                          DWFAC,WPFAC,SPFAC,WBTHD,SBTHD,ISOTP,
     +                          SOPRP,WGAMA,IELEM)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
      DIMENSION        COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
```

```
     +                    ELOAD(36,54),NELPX(3),NELPY(3),NELLA(3),
     +                    HIGHT(3),THPNX(3,4),THPNY(4,3),ISOTP(13,5),
     +                    SOPRP(10,7),WBTHD(3,13),SBTHD(3,13)
      NELMB=IELEM
      IIX=1
      JJY=1
      DO 100 IDW=1,2
      IF(IDW .EQ. 1) GO TO 37
      IF(WBTHD(1,1) .EQ. 0.) GO TO 100
      NUMBR=0
      DO 33 ICOMP=1,4
33    IF(SBTHD(1,ICOMP) .GT. 0.) NUMBR=NUMBR+1
      IF(NUMBR .EQ. 0) GO TO 100
37    DO 80 KZ=1,NLAYR
      DELTH=HIGHT(KZ)/NELLA(KZ)
      ICMPO=4
      L1=2
      L2=3
      DO 50 IX=1,NPNLX+1,IIX
      ICMPO=ICMPO+IIX
      DO 50 JY=1,NPNLY
      ICOMP=ICMPO+(JY-1)*NPNLX
      LRDIF=1
      IL=1
      IR=2
      LX=1
      THICK=THPNY(IX,JY)
      CALL     WALLHD  (ICASE,LRDIF,IL,IR,ICOMP,KZ,IX,NPNLX,WBHDL,
     +                  WBHDR,SBHDL,SBHDR,ISOTP,ISTPL,ISTPR,WBTHD,
     +                  SBTHD)
      LPROP=MATNO(IELEM+1)
      GAMMA=PROPS(LPROP,3)
      UDLOD=-THICK*GAMMA*DWFAC
      DO 40 J=1,NELPY(JY)
      DO 40 K=1,NELLA(KZ)
      IELEM=IELEM+1
      CALL     WALELD  (IX,LX,IDW,IELEM,K,WBHDL,WBHDR,SBHDL,SBHDR,
     +                  SOPRP,ISTPL,ISTPR,DELTH,L1,L2,NNODE,
     +                  NDOFN,LNODS,NGAMA,NGAUB,WPFAC,SPFAC)
      IF(IDW .EQ. 2) GO TO 40
      CALL     GRAVLD  (IELEM,L1,L2,UDLOD,LNODS,NGAUB,NNODE,
     +                  NDOFN)
40    CONTINUE
50    CONTINUE
      ICMPO=4
      L1=1
      L2=3
      DO 70 JY=1,NPNLY+1,JJY
      ICMPO=ICMPO+(JY-1)*NPNLX
      DO 70 IX=1,NPNLX
      ICOMP=ICMPO+IX
      LRDIF=NPNLX
      IL=3
      IR=4
      LY=2
      THICK=THPNX(IX,JY)
      CALL     WALLHD  (ICASE,LRDIF,IL,IR,ICOMP,KZ,JY,NPNLY,WBHDL,
     +                  WBHDR,SBHDL,SBHDR,ISOTP,ISTPL,ISTPR,WBTHD,
     +                  SBTHD)
      LPROP=MATNO(IELEM+1)
      GAMMA=PROPS(LPROP,3)
      UDLOD=-THICK*GAMMA*DWFAC
      DO 60 I=1,NELPX(IX)
      DO 60 K=1,NELLA(KZ)
      IELEM=IELEM+1
      CALL     WALELD  (JY,LY,IDW,IELEM,K,WBHDL,WBHDR,SBHDL,SBHDR,
     +                  SOPRP,ISTPL,ISTPR,DELTH,L1,L2,NNODE,
     +                  NDOFN,LNODS,NGAMA,NGAUB,WPFAC,SPFAC)
      IF(IDW .EQ. 2) GO TO 60
```

```
      CALL     GRAVLD  (IELEM,L1,L2,UDLOD,LNODS,NGAUB,NNODE,
     +                  NDOFN)
 60   CONTINUE
 70   CONTINUE
 80   CONTINUE
      IIX=NPNLX
      JJY=NPNLY
      IELEM=NELMB
100   CONTINUE
      RETURN
      END

      SUBROUTINE      LOADIN  (NPNLX,NPNLY,NCASE,NDOFN,NCOMP,WGAMA,
     +                         WATLV,SOILV,ISOTP,FORCE,FCORD,FACT1,
     +                         FACT2,FACT3,FACT4,SOPRP,NPTLD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       WATLV(13,5),SOILV(13,5),ISOTP(13,5),SOPRP(10,7),
     +                FORCE(8,6),FCORD(8,3),FACT1(5),FACT2(5),
     +                FACT3(5),FACT4(5,8)
      NCOMP=NPNLX*NPNLY+4
      WRITE(6,1000) NCOMP
      READ(5,*) NSOTP,NPTLD,WGAMA
      WRITE(6,1010) WGAMA,NSOTP
      DO 10 JSOTP=1,NSOTP
      READ(5,*)      I,(SOPRP(I,J),J=1,7)
 10   WRITE(6,1020) I,(SOPRP(I,J),J=1,7)
      DO 20 I=1,NCASE
      READ(5,*) (WATLV(ICOMP,I),ICOMP=1,NCOMP)
      READ(5,*) (SOILV(ICOMP,I),ICOMP=1,NCOMP)
 20   READ(5,*) (ISOTP(ICOMP,I),ICOMP=1,NCOMP)
      WRITE(6,1025)
      DO 30 I=1,NCASE
 30   WRITE(6,1030) I,(ICOMP,WATLV(ICOMP,I),SOILV(ICOMP,I),
     + ISOTP(ICOMP,I),ICOMP=1,NCOMP)
      DO 40 IPTLD=1,NPTLD
 40   READ(5,*) I,(FCORD(I,J),J=1,3),(FORCE(I,JDOFN),JDOFN=1,NDOFN)
      WRITE(6,1040)(I,(FCORD(I,J),J=1,3),(FORCE(I,JDOFN),
     + JDOFN=1,NDOFN),I=1,NPTLD)
      DO 50 ICASE=1,NCASE
 50   READ(5,*) I,FACT1(I),FACT2(I),FACT3(I)
      WRITE(6,1050) (I,FACT1(I),FACT2(I),FACT3(I),I=1,NCASE)
      DO 60 ICASE=1,NCASE
 60   READ(5,*) I,(FACT4(I,J),J=1,NPTLD)
      DO 70 J=1,NPTLD
 70   WRITE(6,1060) J,(I,FACT4(I,J),I=1,NCASE)
1000  FORMAT(/10X,'TOTAL NO. OF COMPARTMENTS =',I2)
1010  FORMAT(/10X,'WATER SPECIFIC GRAVITY = ',F10.3,
     + //10X,'NO. OF SOIL TYPES =',I2)
1020  FORMAT(//,'  SOIL NO.',1X,'DRY DENS.',1X,'WET DENS.',2X,
     + 'COHESION',1X,'PHI-ANGLE',1X,'BTA-ANGLE',1X,'SURCHARGE',1X,
     + 'DLT-ANGLE',/2X,8(1H-),2(1X,9(1H-)),2X,8(1H-),4(1X,9(1H-)),
     + /(I10,7F10.2))
1025  FORMAT(/' LOAD CASE',1X,'COMPRTMNT',4X,'WATER LEVEL',
     + 5X,'SOIL LEVEL',6X,'SOIL TYPE',/2(1X,9(1H-)),4X,11(1H-),
     + 5X,10(1H-),6X,9(1H-))
1030  FORMAT(/2I10,2F15.3,I15,/(10X,I10,2F15.3,I15))
1040  FORMAT(/2X,'NO.',1X,'X-COOR',1X,'Y-COOR',1X,'Z-COOR',2X,
     + 'X-FORC.',2X,'Y-FORC.',2X,'Z-FORC.',2X,'XX-MOM.',2X,'YY-MOM.',
     + 2X,'ZZ-MOM.',/2X,3(1H-),3(1X,6(1H-)),6(2X,7(1H-)),
     + /(I5,3F7.3,6F9.3))
1050  FORMAT(//,' LOAD COMBINATION FACTORS FOR DIFFERENT LOAD CASES',
     + /1X,50(1H*),// ' LOAD CASE',4X,'DEAD WEIGHT',1X,
     + 'WATER PRESSURE',2X,'SOIL PRESSURE',/1X,9(1H-),4X,11(1H-),
     + 1X,14(1H-),2X,13(1H-),/(I10,3F15.2))
1060  FORMAT(/10X,' LOAD CASE',4X,'POINT LOAD NO.',I2,/11X,9(1H-),
     + 4X,16(1H-),/(10X,I10,F20.2))
      RETURN
      END
```

```
      SUBROUTINE      HEAD    (ICOMP,ICASE,NLAYR,HIGHT,ZHIGH,WATLV,
     +                         SOILV,THCBS,WBTHD,SBTHD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       WATLV(13,5),SOILV(13,5),WBTHD(3,13),SBTHD(3,13),
     +                HIGHT(3)
      WBHED=WATLV(ICOMP,ICASE)
      IF(WBHED .EQ. 0.) GO TO 40
      WTHED=WBHED-ZHIGH
      IF(WTHED .GT. 0.)GO TO 10
      CALL    HEADLY  (ICOMP,NLAYR,WBHED,WBTHD,HIGHT,THCBS)
      GO TO 20
   40 DO 50 I=1,NLAYR
   50 WBTHD(I,ICOMP)=0.
      GO TO 20
   10 WRITE(6,1000) ICOMP,ICASE
 1000 FORMAT(/' CHECK WATER LEVEL FOR COMPARTMENT NO.',I2,' AT LOAD ',
     + 'CASE NO.',I2,/)
      STOP
   20 SBHED=SOILV(ICOMP,ICASE)
      IF(SBHED .EQ. 0.) GO TO 60
      STHED=SBHED-ZHIGH
      IF(STHED .GT. 0.)GO TO 30
      CALL    HEADLY  (ICOMP,NLAYR,SBHED,SBTHD,HIGHT,THCBS)
      RETURN
   30 WRITE(6,1010) ICOMP,ICASE
 1010 FORMAT(/' CHECK SOIL LEVEL FOR COMPARTMENT NO.',I2,' AT LOAD ',
     + 'CASE NO.',I2,/)
      STOP
   60 DO 70 I=1,NLAYR
   70 SBTHD(I,ICOMP)=0.
      RETURN
      END

      SUBROUTINE      HEADLY  (ICOMP,NLAYR,BOTMH,BOTHD,HIGHT,THCBS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       BOTHD(3,13),HIGHT(3)
      BOTHD(1,ICOMP)=BOTMH+THCBS/2.
      IF(NLAYR .EQ. 1) RETURN
      DO 10 ILAYR=2,NLAYR
      BOTHD(ILAYR,ICOMP)=BOTHD(ILAYR-1,ICOMP)-HIGHT(ILAYR-1)
      IF(BOTHD(ILAYR,ICOMP) .GT. 0.) GO TO 10
      BOTHD(ILAYR,ICOMP)=0.
   10 CONTINUE
      RETURN
      END

      SUBROUTINE      BASEPR  (ICOMP,ICASE,THCBS,WATLV,SOILV,ISOTP,
     +                         SOPRP,PRNET,WGAMA,BGAMA,DWFAC,WPFAC,
     +                         SPFAC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       WATLV(13,5),SOILV(13,5),ISOTP(13,5),SOPRP(10,7)
      BASDN=THCBS*BGAMA
      WAHDI=WATLV(ICOMP,ICASE)
      WAHDO=WATLV(1,ICASE)+THCBS
      IF(WATLV(1,ICASE) .EQ. 0.) WAHDO=0.
      WATPR=(WAHDO-WAHDI)*WGAMA
      SOHED=SOILV(ICOMP,ICASE)
      ISOLT=ISOTP(ICOMP,ICASE)
      SGAMA=SOPRP(ISOLT,1)
      SOLPR=SOHED*SGAMA
      PRNET=-DWFAC*BASDN+WPFAC*WATPR-SPFAC*SOLPR
      RETURN
      END

      SUBROUTINE      GRAVLD  (IELEM,L1,L2,UDLOD,LNODS,NGAUB,
     +                         NNODE,NDOFN)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
```

```
      DIMENSION       LNODS(36,9),COORD(153,3),ELOAD(36,54),
     +                POSGP(3),WEIGP(3),CARTD(2,9),DERIV(2,9),
     +                ELCOD(2,9), SHAPE(9),GPCOD(2,9)
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      ELCOD(1,INODE)=COORD(LNODE,L1)
      IF(L1 .EQ. 2) ELCOD(1,INODE)=-COORD(LNODE,L1)
   10 ELCOD(2,INODE)=COORD(LNODE,L2)
      KGASP=0
      CALL    GAUSSQ  (NGAUB,POSGP,WEIGP)
      DO 30 IGAUS=1,NGAUB
      EXISP=POSGP(IGAUS)
      DO 30 JGAUS=1,NGAUB
      ETASP=POSGP(JGAUS)
      KGASP=KGASP+1
      CALL    SFR2    (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL    JACOB2  (CARTD,DERIV,DJACB,ELCOD,GPCOD,IELEM,KGASP,
     +                 NNODE,SHAPE)
      DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      DO 20 INODE=1,NNODE
      NPOSN=(INODE-1)*NDOFN+3
      ELOAD(IELEM,NPOSN)=ELOAD(IELEM,NPOSN)+SHAPE(INODE)*UDLOD*DAREA
   20 CONTINUE
   30 CONTINUE
      RETURN
      END

      SUBROUTINE      WALLHD  (ICASE,LRDIF,IL,IR,ICOMP,KZ,IXJY,NPLXY,
     +                         WBHDL,WBHDR,SBHDL,SBHDR,ISOTP,ISTPL,
     +                         ISTPR,WBTHD,SBTHD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       WBTHD(3,13),SBTHD(3,13),ISOTP(13,5)
      IF(IXJY .EQ. 1) GO TO 10
      WBHDL=WBTHD(KZ,ICOMP-LRDIF)
      SBHDL=SBTHD(KZ,ICOMP-LRDIF)
      ISTPL=ISOTP(ICOMP-LRDIF,ICASE)
      GO TO 20
   10 WBHDL=WBTHD(KZ,IL)
      SBHDL=SBTHD(KZ,IL)
      ISTPL=ISOTP(IL,ICASE)
   20 IF(IXJY .EQ. NPLXY+1) GO TO 30
      WBHDR=WBTHD(KZ,ICOMP)
      SBHDR=SBTHD(KZ,ICOMP)
      ISTPR=ISOTP(ICOMP,ICASE)
      RETURN
   30 WBHDR=WBTHD(KZ,IR)
      SBHDR=SBTHD(KZ,IR)
      ISTPR=ISOTP(IR,ICASE)
      RETURN
      END

      SUBROUTINE      WALELD  (IXJY,LXY,IDW,IELEM,K,WBHDL,WBHDR,SBHDL,
     +                         SBHDR,SOPRP,ISTPL,ISTPR,DELTH,L1,
     +                         L2,NNODE,NDOFN,LNODS,WGAMA,NGAUB,
     +                         WPFAC,SPFAC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
      DIMENSION       ELOAD(36,54),SOPRP(10,7),LNODS(36,9),
     +                COORD(153,3)
      IF(IDW .GT. 1) GO TO 10
      EWBHL=WBHDL-(K-1)*DELTH
      CALL    ELEMHD  (EWBHL,EWTHL,EWOHL,DELTH)
      IWS=1
      ILR=1
      CALL    ELEMLD  (LXY,IDW,IWS,ILR,IELEM,EWBHL,EWTHL,EWOHL,
     +                 ISOLT,SOPRP,WPFAC,SPFAC,L1,L2,NNODE,
     +                 NDOFN,LNODS,DELTH,WGAMA,NGAUB)
      EWBHR=WBHDR-(K-1)*DELTH
```

```
      CALL     ELEMHD  (EWBHR,EWTHR,EWOHR,DELTH)
      IWS=1
      ILR=-1
      CALL     ELEMLD  (LXY,IDW,IWS,ILR,IELEM,EWBHR,EWTHR,EWOHR,
     +                  ISOLT,SOPRP,WPFAC,SPFAC,L1,L2,NNODE,
     +                  NDOFN,LNODS,DELTH,WGAMA,NGAUB)
10    IF(IXJY .GT. 1 .AND. IDW .GT. 1) GO TO 20
      ESBHL=SBHDL-(K-1)*DELTH
      CALL     ELEMHD  (ESBHL,ESTHL,ESOHL,DELTH)
      IWS=2
      ILR=1
      ISOLT=ISTPL
      CALL     ELEMLD  (LXY,IDW,IWS,ILR,IELEM,ESBHL,ESTHL,ESOHL,
     +                  ISOLT,SOPRP,WPFAC,SPFAC,L1,L2,NNODE,
     +                  NDOFN,LNODS,DELTH,WGAMA,NGAUB)
      IF(IDW .GT. 1) RETURN
20    ESBHR=SBHDR-(K-1)*DELTH
      CALL     ELEMHD  (ESBHR,ESTHR,ESOHR,DELTH)
      IWS=2
      ILR=-1
      ISOLT=ISTPR
      CALL     ELEMLD  (LXY,IDW,IWS,ILR,IELEM,ESBHR,ESTHR,ESOHR,
     +                  ISOLT,SOPRP,WPFAC,SPFAC,L1,L2,NNODE,
     +                  NDOFN,LNODS,DELTH,WGAMA,NGAUB)
      RETURN
      END

      SUBROUTINE      ELEMHD  (EBHED,ETHED,EOHED,DELTH)
      IMPLICIT REAL*8(A-H,O-Z)
      IF(EBHED .GT. 0.) GO TO 10
      EBHED=0.
      ETHED=0.
      EOHED=0.
      RETURN
10    ETHED=EBHED-DELTH
      IF(ETHED .GE. 0.) GO TO 20
      EOHED=-ETHED
      ETHED=0.
      RETURN
20    EOHED=0.
      RETURN
      END

      SUBROUTINE      ELEMLD  (LXY,IDW,IWS,ILR,IELEM,EBHED,ETHED,
     +                          EOHED,ISOLT,SOPRP,WPFAC,
     +                          SPFAC,L1,L2,NNODE,NDOFN,LNODS,
     +                          DELTH,WGAMA,NGAUB)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
      DIMENSION       ELOAD(36,54),SOPRP(10,7),WEIGP(3),POSGP(3),
     +                DERIV(2,9),CARTD(2,9),SHAPE(9),GPCOD(2,9),
     +                ELCOD(2,9),LNODS(36,9),COORD(153,3)
      IF(EBHED .EQ. 0.) RETURN
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      ELCOD(1,INODE)=COORD(LNODE,L1)
      IF(L1 .EQ. 2) ELCOD(1,INODE)=-COORD(LNODE,L1)
10    ELCOD(2,INODE)=COORD(LNODE,L2)
      KGASP=0
      CALL     GAUSSQ  (NGAUB,POSGP,WEIGP)
      DO 50 IGAUS=1,NGAUB
      EXISP=POSGP(IGAUS)
      DO 50 JGAUS=1,NGAUB
      ETASP=POSGP(JGAUS)
      KGASP=KGASP+1
      CALL     SFR2    (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL     JACOB2  (CARTD,DERIV,DJACB,ELCOD,GPCOD,IELEM,KGASP,
     +                  NNODE,SHAPE)
```

```
      DAREA=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      GPHED=EBHED-(ETASP+1.)*(EBHED-ETHED)/(2.-2.*EOHED/DELTH)
      IF(GPHED .LE. 0.) GO TO 50
      IF(IDW .GT. 1 .OR. IWS .GT. 1) GO TO 20
      GPPRS=GPHED*WGAMA*ILR
      FACT=WPFAC
      GO TO 30
20    CALL    SOILPR  (IDW,ILR,ISOLT,SOPRP,GPHED,GPPRS)
      FACT=SPFAC
30    DO 40 INODE=1,NNODE
      NPOSN=(INODE-1)*NDOFN+LXY
      ELOAD(IELEM,NPOSN)=ELOAD(IELEM,NPOSN)+SHAPE(INODE)*DAREA*GPPRS*
     + FACT
40    CONTINUE
50    CONTINUE
      RETURN
      END

      SUBROUTINE      SOILPR  (IDW,ILR,ISOLT,SOPRP,GPHED,GPPRS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION        SOPRP(10,7)
      PI=3.141592654
      SGAMA=SOPRP(ISOLT,IDW)
      IF(IDW .EQ. 2) SGAMA=SGAMA-SOPRP(ISOLT,1)
      COHES=SOPRP(ISOLT,3)
      PHI=SOPRP(ISOLT,4)*PI/180.
      BETA=SOPRP(ISOLT,5)*PI/180.
      SURCH=SOPRP(ISOLT,6)
      DELTA=SOPRP(ISOLT,7)*PI/180.
      ALPHA=PI/2.
      SA=SIN(ALPHA)
      SA1B=SIN(ALPHA+BETA)
      FACN=SA/SA1B
      SA1P=SIN(ALPHA+PHI)
      CD=COS(DELTA)
      SA2D=SIN(ALPHA-DELTA)
      SP1D=SIN(PHI+DELTA)
      SP2B=SIN(PHI-BETA)
      FACKA=(SA1P*SA1P*CD)/(SA*SA2D*(1.+SQRT(SP1D*SP2B/SA2D*SA1B))
     + *(1.+SQRT(SP1D*SP2B/SA2D*SA1B)))
      GPPRS=0.5*(SGAMA+2.*FACN*SURCH/GPHED)*GPHED*GPHED*FACKA/(SA*CD)*
     + ILR
      RETURN
      END

      SUBROUTINE      CONLOC  (IPTLD,NPNLX,NPNLY,NLAYR,NELPX,NELPY,
     +                         NELLA,SPNXL,SPNYW,HIGHT,XLMOD,YWMOD,
     +                         ZHIGH,THCBS,FCORD,NELMB,IELEM,IIX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       FCORD(8,3),NELPX(3),NELPY(3),NELLA(3),SPNXL(3),
     +                SPNYW(3),HIGHT(3)
      PREC=1.E-15
      IIX=0
      JJY=0
      XCORD=0.
      FXCOR=FCORD(IPTLD,1)
      IF(FXCOR-XLMOD .GT. PREC) GO TO 700
      IF(FXCOR .GT. 0.) GO TO 5
      IIX=1
      IIPNX=1
      GO TO 30
 5    DO 10 IX=1,NPNLX
      XCORD=XCORD+SPNXL(IX)
      IF(FXCOR-XCORD .GT. PREC) GO TO 10
      GO TO 20
10    CONTINUE
20    IIPNX=IX
      IF(DABS(FXCOR-XCORD) .GT. PREC) GO TO 30
      IIX=1
```

```
      IIPNX=IX+1
 30   YCORD=0.
      FYCOR=FCORD(IPTLD,2)
      IF(FYCOR-YWMOD .GT. PREC) GO TO 700
      IF(FYCOR .GT. 0.) GO TO 35
      JJY=1
      JJPNY=1
      IF(IIX .EQ. 1) GO TO 69
      GO TO 65
 35   DO 40 JY=1,NPNLY
      YCORD=YCORD+SPNYW(JY)
      IF(FYCOR-YCORD .GT. PREC) GO TO 40
      GO TO 50
 40   CONTINUE
 50   JJPNY=JY
      IF(DABS(FYCOR-YCORD) .GT. PREC) GO TO 60
      JJY=1
      JJPNY=JY+1
      GO TO 65
 60   IF(IIX .EQ. 0) GO TO 700
 65   IF(IIX .EQ. 0 .OR. JJY .EQ. 0) GO TO 75
 69   JJY=0
      IF(JJPNY .GT. 1) JJPNY=JJPNY-1
 75   IF(JJY .EQ. 0) GO TO 33
      IF(XCORD .GT. 0.) GO TO 85
      INELX=1
      GO TO 800
 85   XCORD=XCORD-SPNXL(IIPNX)
      DELTX=SPNXL(IIPNX)/NELPX(IIPNX)
      DO 90 I=1,NELPX(IIPNX)
      XCORD=XCORD+DELTX
      IF(FXCOR-XCORD .GT. PREC) GO TO 90
      GO TO 120
 90   CONTINUE
120   INELX=I
      GO TO 800
 33   IF(YCORD .GT. 0.) GO TO 37
      JNELY=1
      GO TO 800
 37   YCORD=YCORD-SPNYW(JJPNY)
      DELTY=SPNYW(JJPNY)/NELPY(JJPNY)
      DO 91 J=1,NELPY(JJPNY)
      YCORD=YCORD+DELTY
      IF(FYCOR-YCORD .GT. PREC) GO TO 91
      GO TO 121
 91   CONTINUE
121   JNELY=J
      GO TO 800
700   WRITE(6,1000) IPTLD
1000  FORMAT(' CHECK FOR POINT OF APPLYING CONCENTRATED LOAD NO.',I2)
      STOP
800   ZCORD=0.
      FZCOR=FCORD(IPTLD,3)
      ZHIGB=ZHIGH+THCBS/2.
      IF(FZCOR-ZHIGB .GT. PREC) GO TO 700
      DO 70 KZ=1,NLAYR
      ZCORD=ZCORD+HIGHT(KZ)
      IF(FZCOR-ZCORD .GT. PREC) GO TO 70
      GO TO 80
 70   CONTINUE
 80   KKZ=KZ
      ZCORD=ZCORD-HIGHT(KKZ)
      DELTZ=HIGHT(KKZ)/NELLA(KKZ)
      DO 100 K=1,NELLA(KKZ)
      ZCORD=ZCORD+DELTZ
      IF(FZCOR-ZCORD .GT. PREC) GO TO 100
      GO TO 110
100   CONTINUE
110   KNELZ=K
```

```
      CALL    CONELM  (IELEM,NELMB,NPNLX,NPNLY,NLAYR,NELPX,
     +                 NELPY,NELLA,IIX,JJY,IIPNX,JJPNY,KKZ,
     +                 INELX,JNELY,KNELZ)
      RETURN
      END

      SUBROUTINE      CONELM  (IELEM,NELMB,NPNLX,NPNLY,NLAYR,NELPX,
     +                         NELPY,NELLA,IIX,JJY,IIPNX,JJPNY,KKZ,
     +                         INELX,JNELY,KNELZ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       NELPX(3),NELPY(3),NELLA(3)
      IELEM=NELMB
      DO 50 KZ=1,NLAYR
      DO 20 IX=1,NPNLX+1
      DO 20 JY=1,NPNLY
      DO 10 J=1,NELPY(JY)
      DO 10 K=1,NELLA(KZ)
      IELEM=IELEM+1
      IF(IIX .EQ. 0)    GO TO 10
      IF(KZ .LT. KKZ)   GO TO 10
      IF(IX .LT. IIPNX) GO TO 10
      IF(JY .LT. JJPNY) GO TO 10
      IF(J .LT. JNELY)  GO TO 10
      IF(K .LT. KNELZ)  GO TO 10
      RETURN
10    CONTINUE
20    CONTINUE
      DO 40 JY=1,NPNLY+1
      DO 40 IX=1,NPNLX
      DO 30 I=1,NELPX(IX)
      DO 30 K=1,NELLA(KZ)
      IELEM=IELEM+1
      IF(JJY .EQ. 0)    GO TO 30
      IF(KZ .LT. KKZ)   GO TO 30
      IF(JY .LT. JJPNY) GO TO 30
      IF(IX .LT. IIPNX) GO TO 30
      IF(I .LT. INELX)  GO TO 30
      IF(K .LT. KNELZ)  GO TO 30
      RETURN
30    CONTINUE
40    CONTINUE
50    CONTINUE
      RETURN
      END

      SUBROUTINE      CONLOD  (IELEM,IIX,IPTLD,NNODE,NDOFN,LNODS,
     +                         FORCE,FCORD,FACT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ELOAD/ ELOAD
      COMMON/COORD/ COORD
      DIMENSION       LNODS(36,9),COORD(153,3),ELOAD(36,54),
     +                FORCE(8,6), FCORD(8,3),DERIV(2,9),SHAPE(9),
     +                ELCOD(2,9)
      L2=3
      IF(IIX .EQ. 1) GO TO 10
      L1=1
      GO TO 20
10    L1=2
20    DO 30 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      ELCOD(1,INODE)=COORD(LNODE,L1)
      ELCOD(2,INODE)=COORD(LNODE,L2)
30    CONTINUE
      EXIFP=(FCORD(IPTLD,L1)-ELCOD(1,2))/(ELCOD(1,2)-ELCOD(1,1))
      ETAFP=(FCORD(IPTLD,L2)-ELCOD(2,4))/(ELCOD(2,4)-ELCOD(2,3))
      CALL    SFR2    (DERIV,ETAFP,EXIFP,SHAPE,NNODE)
      IEVAB=0
      DO 40 INODE=1,NNODE
      DO 40 IDOFN=1,NDOFN
```

```
      IEVAB=IEVAB+1
      FORCI=FORCE(IPTLD,IDOFN)
      IF(FORCI .EQ. 0.) GO TO 40
      ELOAD(IELEM,IEVAB)=ELOAD(IELEM,IEVAB)+FACT*SHAPE(INODE)*FORCI
 40   CONTINUE
      RETURN
      END

      SUBROUTINE      OUTDIS  (ICASE,ASDIS,NDOFN,NPOIN,TREAC,SKNOD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       ASDIS(918,5),TREAC(25,3,5),SKNOD(153)
      WRITE(8,111)
111   FORMAT(' subroutine outdis')
      WRITE(6,900) ICASE
 900  FORMAT(///,5X,19(1H*),/5X,'* LOAD CASE NO.',I2,' *',/5X,19(1H*),
     + //5X,'DISPLACEMENTS',/5X,13(1H-))
      WRITE(6,905)
 905  FORMAT(4X,'NODE',3X,'DISP-X',6X,'DISP-Y',6X,'DISP-Z',6X,
     + 'XX-ROT',6X,'YY-ROT',6X,'ZZ-ROT')
      DO 5 IPOIN=1,NPOIN
      NGASH=IPOIN*NDOFN
      NGISH=NGASH-NDOFN+1
 5    WRITE(6,910) IPOIN,(ASDIS(IGASH,ICASE),IGASH=NGISH,NGASH)
 910  FORMAT(I8,6E12.4)
      WRITE(6,915)
 915  FORMAT(//5X,'REACTIONS',/5X,9(1H-))
      WRITE(6,920)
 920  FORMAT(4X,'NODE',3X,'FORC-X',6X,'FORC-Y',6X,'FORC-Z')
      ISPRT=0
      DO 10 IPOIN=1,NPOIN
      IF(SKNOD(IPOIN) .EQ. 0.) GO TO 10
      ISPRT=ISPRT+1
      WRITE(6,910) IPOIN,(TREAC(ISPRT,IDOFN,ICASE),IDOFN=1,3)
 10   CONTINUE
      RETURN
      END

      SUBROUTINE      OUTSTR  (ICASE,ASDIS,LNODS,MATNO,NNODE,NDOFN,
     +                         NGAUS,PROPS,STREL,NPNLX,NPNLY,NLAYR,
     +                         NELPX,NELPY,NELLA,THBAS,THPNX,THPNY)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COORD/ COORD
      DIMENSION       ASDIS(918,5),COORD(153,3),LNODS(36,9),MATNO(36),
     +                PROPS(10,3),STREL(9,8),NELPX(3),NELPY(3),
     +                NELLA(3),THBAS(3,3),THPNX(3,4),THPNY(4,3)
      WRITE(8,111)
111   FORMAT(' subroutine outstr')
      WRITE(6,900)
 900  FORMAT(//5X,'STRESS RESULTANTS',/5X,17(1H*))
      DO 100 IBM=1,2
      IF(IBM .EQ. 2) GO TO 3
      WRITE(6,901)
 901  FORMAT(//1X,'PLATE BENDING FORCES',/1X,20(1H-),//)
      WRITE(6,905)
 905  FORMAT(1X,'ELEMENT NO.',4X,'NODE',2X,'MOMENT 1-1',2X,
     + 'MOMENT 2-2',2X,'MOMENT 1-2',2X,' SHEAR 1-3',2X,' SHEAR 2-3',
     + /1X,11(1H-),4X,4(1H-),5(2X,10(1H-)))
      GO TO 7
  3   WRITE(6,930)
930   FORMAT(//1X,'IN PLANE STRESSES',/1X,17(1H-),//)
      WRITE(6,935)
935   FORMAT(1X,'ELEMENT NO.',4X,'NODE',4X,'STRESS 1-1',4X,
     + 'STRESS 2-2',4X,'STRESS 1-2',/1X,11(1H-),4X,4(1H-),
     + 3(4X,10(1H-)))
  7   IELEM=0
      L1=1
      L2=2
      DO 10 IPNLX=1,NPNLX
      DO 10 JPNLY=1,NPNLY
```

```
      LPROP=MATNO(IELEM+1)
      THICK=THBAS(IPNLX,JPNLY)
      DO 10 NUMEL=1,NELPX(IPNLX)*NELPY(JPNLY)
      IELEM=IELEM+1
 10   CALL     ELMSTR  (ICASE,IBM,IELEM,L1,L2,LNODS,ASDIS,
     +                  NNODE,NDOFN,NGAUS,PROPS,STREL,LPROP,THICK)
      DO 40 KZ=1,NLAYR
      L1=2
      L2=3
      DO 20 JPNLY=1,NPNLY
      DO 20 IPNLX=1,NPNLX+1
      LPROP=MATNO(IELEM+1)
      THICK=THPNY(IPNLX,JPNLY)
      DO 20 NUMEL=1,NELPY(JPNLY)*NELLA(KZ)
      IELEM=IELEM+1
 20   CALL     ELMSTR  (ICASE,IBM,IELEM,L1,L2,LNODS,ASDIS,
     +                  NNODE,NDOFN,NGAUS,PROPS,STREL,LPROP,THICK)
      L1=1
      L2=3
      DO 30 IPNLX=1,NPNLX
      DO 30 JPNLY=1,NPNLY+1
      LPROP=MATNO(IELEM+1)
      THICK=THPNX(IPNLX,JPNLY)
      DO 30 NUMEL=1,NELPX(IPNLX)*NELLA(KZ)
      IELEM=IELEM+1
 30   CALL     ELMSTR  (ICASE,IBM,IELEM,L1,L2,LNODS,ASDIS,
     +                  NNODE,NDOFN,NGAUS,PROPS,STREL,LPROP,THICK)
 40   CONTINUE
100   CONTINUE
      RETURN
      END

      SUBROUTINE      ELMSTR  (ICASE,IBM,IELEM,L1,L2,LNODS,ASDIS,
     +                         NNODE,NDOFN,NGAUS,PROPS,STREL,LPROP,
     +                         THICK)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COORD/ COORD
      DIMENSION       ASDIS(918,5),COORD(153,3),ELCOD(2,9),SPCOD(2,9),
     +                ELDIS(6,9),LNODS(30,9),CARTD(2,9),PROPS(10,3),
     +                POSGP(3),SHAPE(9),DGRAD(6),POSSP(3),SHEAV(9,2),
     +                STRES(8),STREL(9,8),STRSG(4,2),WEIGP(3),
     +                ICODE(9),DERIV(2,9),DFLEX(3,3),DSHER(2,2),
     +                DMEMB(3,3)
      DATA    POSSP/-1.,0.,1./,ICODE/1,8,7,2,9,6,3,4,5/
      CALL    RAZERO  (STREL,9,8)
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      ELCOD(1,INODE)=COORD(LNODE,L1)
      IF(L1 .EQ. 2) ELCOD(1,INODE)=-COORD(LNODE,L1)
      ELCOD(2,INODE)=COORD(LNODE,L2)
      NPOSN=(LNODE-1)*NDOFN
      DO 10 IDOFN=1,NDOFN
      NPOSN=NPOSN+1
      ELDIS(IDOFN,INODE)=ASDIS(NPOSN,ICASE)
 10   CONTINUE
      IF(L1 .EQ. 1 .AND. L2 .EQ. 2) GO TO 18
      DO 14 INODE=1,NNODE
      ELDIS(2,INODE)=-ELDIS(2,INODE)
 14   ELDIS(5,INODE)=-ELDIS(5,INODE)
      IF(L1 .EQ. 1) GO TO 18
      DO 16 INODE=1,NNODE
      ELDIS(1,INODE)=-ELDIS(1,INODE)
 16   ELDIS(4,INODE)=-ELDIS(4,INODE)
 18   IF(IBM .EQ. 2) GO TO 35
      KSAMP=0
      DO 20 ISAMP=1,3
      EXISP=POSSP(ISAMP)
      DO 20 JSAMP=1,3
      ETASP=POSSP(JSAMP)
```

```
      CALL     BENDM    (ETASP,EXISP,IELEM,KSAMP,LPROP,ELCOD,
     +                   PROPS,NNODE,DFLEX,DSHER,SPCOD,L1,L2,
     +                   CARTD,SHAPE,ELDIS,STRES,DGRAD,DERIV,
     +                   THICK)
      INODE=ICODE(KSAMP)
      DO 20 ISTRE=1,3
      STREL(INODE,ISTRE)=STREL(INODE,ISTRE)+STRES(ISTRE)
   20 CONTINUE
      KSAMP=0
      CALL     GAUSSQ  (NGAUS,POSGP,WEIGP)
      DO 25 IGAUS=1,NGAUS
      EXISP=POSGP(IGAUS)
      DO 25 JGAUS=1,NGAUS
      ETASP=POSGP(JGAUS)
      CALL     SHEAR    (ETASP,EXISP,IELEM,KSAMP,LPROP,DERIV,
     +                   ELCOD,PROPS,NNODE,DFLEX,
     +                   DSHER,SPCOD,CARTD,SHAPE,ELDIS,STRES,
     +                   STRSG,DGRAD,L1,L2,THICK)
   25 CONTINUE
      CALL     EXTRA    (SHEAV,STRSG)
      DO 30 ISAMP=1,9
      INODE=ICODE(ISAMP)
      DO 30 ISTRE=1,2
      KSTRE=ISTRE+3
   30 STREL(INODE,KSTRE)=STREL(INODE,KSTRE)+SHEAV(ISAMP,ISTRE)
      WRITE(6,1000) IELEM,(LNODS(IELEM,INODE),(STREL(INODE,ISTRE),
     + ISTRE=1,5),INODE=1,NNODE)
 1000 FORMAT(/9X,I3,I8,5E12.4,/(12X,I8,5E12.4))
      RETURN
   35 KSAMP=0
      DO 40 ISAMP=1,3
      EXISP=POSSP(ISAMP)
      DO 40 JSAMP=1,3
      ETASP=POSSP(JSAMP)
      CALL     MEMBR    (ETASP,EXISP,IELEM,KSAMP,LPROP,DMEMB,ELCOD,
     +                   PROPS,NNODE,SPCOD,CARTD,SHAPE,ELDIS,STRES,
     +                   DERIV,L1,L2)
      INODE=ICODE(KSAMP)
      DO 40 ISTRE=6,8
      STREL(INODE,ISTRE)=STREL(INODE,ISTRE)+STRES(ISTRE)
   40 CONTINUE
      WRITE(6,1010) IELEM,(LNODS(IELEM,INODE),(STREL(INODE,ISTRE),
     + ISTRE=6,8),INODE=1,NNODE)
 1010 FORMAT(/9X,I3,I8,3E14.4,/(12X,I8,3E14.4))
      RETURN
      END

      SUBROUTINE       BENDM    (ETASP,EXISP,IELEM,KSAMP,LPROP,ELCOD,
     +                           PROPS,NNODE,DFLEX,DSHER,SPCOD,L1,L2,
     +                           CARTD,SHAPE,ELDIS,STRES,DGRAD,DERIV,
     +                           THICK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION        CARTD(2,9),DERIV(2,9),DFLEX(3,3),DGRAD(6),
     +                 DSHER(2,2),ELCOD(2,9),ELDIS(6,9),PROPS(10,3),
     +                 SHAPE(9),SPCOD(2,9),STRES(8)
      KSAMP=KSAMP+1
      CALL     SFR2     (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL     JACOB2   (CARTD,DERIV,DJACB,ELCOD,SPCOD,
     +                   IELEM,KSAMP,NNODE,SHAPE)
      CALL     MODPB    (THICK,DFLEX,DSHER,LPROP,PROPS,1,0)
      CALL     STRPB    (CARTD,DFLEX,DGRAD,DSHER,ELDIS,
     +                   NNODE,SHAPE,STRES,1,0,L1,L2)
      RETURN
      END

      SUBROUTINE       STRPB    (CARTD,DFLEX,DGRAD,DSHER,ELDIS,
     +                           NNODE,SHAPE,STRES,IFFLE,IFSHE,L1,L2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION        CARTD(2,9),DFLEX(3,3),DGRAD(6),DSHER(2,2),
```

```
     +                  ELDIS(6,9),SHAPE(9),STRES(8)
      CALL    RAZERO   (STRES,8,1)
      CALL    GRADPB  (CARTD,DGRAD,ELDIS,NNODE,L1,L2)
      IF(IFSHE .EQ. 0) GO TO 20
      ROT11=0.
      ROT22=0.
      DO 10 INODE=1,NNODE
      ROT11=ROT11+SHAPE(INODE)*ELDIS(L1+3,INODE)
   10 ROT22=ROT22+SHAPE(INODE)*ELDIS(L2+3,INODE)
   20 IF(IFFLE .EQ. 0) GO TO 30
      EFL11=DGRAD(3)
      EFL22=-DGRAD(5)
      EFL12=DGRAD(6)-DGRAD(2)
      STRES(1)=DFLEX(1,1)*EFL11+DFLEX(1,2)*EFL22
      STRES(2)=DFLEX(2,1)*EFL11+DFLEX(2,2)*EFL22
      STRES(3)=DFLEX(3,3)*EFL12
   30 IF(IFSHE .EQ. 0) RETURN
      ESH13=DGRAD(1)+ROT22
      ESH23=DGRAD(4)-ROT11
      STRES(4)=DSHER(1,1)*ESH13
      STRES(5)=DSHER(2,2)*ESH23
      RETURN
      END

      SUBROUTINE   GRADPB   (CARTD,DGRAD,ELDIS,NNODE,L1,L2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       CARTD(2,9),DGRAD(6),ELDIS(6,9),CONST(3)
      CALL    RAZERO  (DGRAD,6,1)
      IF(L2 .EQ. 3) GO TO 20
      DO 10 INODE=1,NNODE
      DNID1=CARTD(1,INODE)
      DNID2=CARTD(2,INODE)
      DO 10 IDOFN=1,3
      IPOSN=IDOFN+3
      CONS=ELDIS(IDOFN+2,INODE)
      DGRAD(IDOFN)=DGRAD(IDOFN)+DNID1*CONS
   10 DGRAD(IPOSN)=DGRAD(IPOSN)+DNID2*CONS
      RETURN
   20 IF(L1 .EQ. 1) GO TO 40
      DO 30 INODE=1,NNODE
      CONST(1)=ELDIS(1,INODE)
      CONST(2)=ELDIS(5,INODE)
      CONST(3)=ELDIS(6,INODE)
      DNID1=CARTD(1,INODE)
      DNID2=CARTD(2,INODE)
      DO 30 IDOFN=1,3
      IPOSN=IDOFN+3
      DGRAD(IDOFN)=DGRAD(IDOFN)+DNID1*CONST(IDOFN)
   30 DGRAD(IPOSN)=DGRAD(IPOSN)+DNID2*CONST(IDOFN)
      RETURN
   40 DO 50 INODE=1,NNODE
      CONST(1)=ELDIS(2,INODE)
      CONST(2)=ELDIS(4,INODE)
      CONST(3)=ELDIS(6,INODE)
      DNID1=CARTD(1,INODE)
      DNID2=CARTD(2,INODE)
      DO 50 IDOFN=1,3
      IPOSN=IDOFN+3
      DGRAD(IDOFN)=DGRAD(IDOFN)+DNID1*CONST(IDOFN)
   50 DGRAD(IPOSN)=DGRAD(IPOSN)+DNID2*CONST(IDOFN)
      RETURN
      END

      SUBROUTINE      SHEAR   (ETASP,EXISP,IELEM,KSAMP,LPROP,DERIV,
     +                         ELCOD,PROPS,NNODE,DFLEX,
     +                         DSHER,SPCOD,CARTD,SHAPE,ELDIS,STRES,
     +                         STRSG,DGRAD,L1,L2,THICK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       CARTD(2,9),DERIV(2,9),DFLEX(3,3),DGRAD(6),
```

```
     +                 DSHER(2,2),ELCOD(2,9),ELDIS(6,9),PROPS(10,3),
     +                 SHAPE(9),SPCOD(2,9),STRES(8),STRSG(4,2)
      KSAMP=KSAMP+1
      CALL     SFR2    (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL     JACOB2  (CARTD,DERIV,DJACB,ELCOD,SPCOD,IELEM,
     +                  KSAMP,NNODE,SHAPE)
      CALL     MODPB   (THICK,DFLEX,DSHER,LPROP,PROPS,0,1)
      CALL     STRPB   (CARTD,DFLEX,DGRAD,DSHER,ELDIS,
     +                  NNODE,SHAPE,STRES,0,1,L1,L2)
      STRSG(KSAMP,1)=STRES(4)
      STRSG(KSAMP,2)=STRES(5)
      RETURN
      END

      SUBROUTINE  EXTRA  (SHEAV,STRSG)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION        SHEAV(9,2),STRSG(4,2)
      FACT1=1.+0.5*SQRT(3.)
      FACT2=-0.5
      FACT3=-0.5
      FACT4=1.-0.5*SQRT(3.)
      DO 5 ISTRE=1,2
      SHEAV(1,ISTRE)=STRSG(1,ISTRE)*FACT1+STRSG(2,ISTRE)*FACT2+
     +               STRSG(3,ISTRE)*FACT3+STRSG(4,ISTRE)*FACT4
      SHEAV(3,ISTRE)=STRSG(2,ISTRE)*FACT1+STRSG(1,ISTRE)*FACT2+
     +               STRSG(4,ISTRE)*FACT3+STRSG(3,ISTRE)*FACT4
      SHEAV(7,ISTRE)=STRSG(3,ISTRE)*FACT1+STRSG(1,ISTRE)*FACT2+
     +               STRSG(4,ISTRE)*FACT3+STRSG(2,ISTRE)*FACT4
      SHEAV(9,ISTRE)=STRSG(4,ISTRE)*FACT1+STRSG(2,ISTRE)*FACT2+
     +               STRSG(3,ISTRE)*FACT3+STRSG(1,ISTRE)*FACT4
      SHEAV(2,ISTRE)=0.5*(SHEAV(1,ISTRE)+SHEAV(3,ISTRE))
      SHEAV(4,ISTRE)=0.5*(SHEAV(1,ISTRE)+SHEAV(7,ISTRE))
      SHEAV(6,ISTRE)=0.5*(SHEAV(3,ISTRE)+SHEAV(9,ISTRE))
      SHEAV(8,ISTRE)=0.5*(SHEAV(7,ISTRE)+SHEAV(9,ISTRE))
      SHEAV(5,ISTRE)=0.25*(STRSG(1,ISTRE)+STRSG(2,ISTRE)+
     +                     STRSG(3,ISTRE)+STRSG(4,ISTRE))
    5 CONTINUE
      RETURN
      END

      SUBROUTINE      MEMBR   (ETASP,EXISP,IELEM,KSAMP,LPROP,DMEMB,
     +                         ELCOD,PROPS,NNODE,SPCOD,CARTD,SHAPE,
     +                         ELDIS,STRES,DERIV,L1,L2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION        CARTD(2,9),DERIV(2,9),DMEMB(3,3),ELCOD(2,9),
     +                 ELDIS(6,9),PROPS(10,3),SHAPE(9),SPCOD(2,9),
     +                 STRES(8)
      KSAMP=KSAMP+1
      CALL     SFR2    (DERIV,ETASP,EXISP,SHAPE,NNODE)
      CALL     JACOB2  (CARTD,DERIV,DJACB,ELCOD,SPCOD,IELEM,
     +                  KSAMP,NNODE,SHAPE)
      CALL     MODMA   (DMEMB,LPROP,PROPS)
      CALL     STRMA   (DMEMB,CARTD,NNODE,STRES,ELDIS,L1,L2)
      RETURN
      END

      SUBROUTINE      STRMA   (DMEMB,CARTD,NNODE,STRES,ELDIS,L1,L2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION        CARTD(2,9),DMEMB(3,3),ELDIS(6,9),STRES(8)
      EMB11=0.
      EMB22=0.
      EMB12=0.
      DO 10 INODE=1,NNODE
      DNID1=CARTD(1,INODE)
      DNID2=CARTD(2,INODE)
      DISP1=ELDIS(L1,INODE)
      DISP2=ELDIS(L2,INODE)
      EMB11=EMB11+DNID1*DISP1
      EMB22=EMB22+DNID2*DISP2
```

```
 10     EMB12=EMB12+DNID1*DISP2+DNID2*DISP1
        STRES(6)=DMEMB(1,1)*EMB11+DMEMB(1,2)*EMB22
        STRES(7)=DMEMB(2,1)*EMB11+DMEMB(2,2)*EMB22
        STRES(8)=DMEMB(3,3)*EMB22
        RETURN
        END

        SUBROUTINE      SOLVEQ  (TREAC,NPOIN,NCPRM,NCPCN,
     +                           ASDIS,LNODS,NEVAB,NGAUB,NGAUS,NGAUM,
     +                           NNODE,NDOFN,MATNO,PROPS,INDPL,NPNLX,
     +                           NPNLY,NLAYR,NELPX,NELPY,NELLA,THBAS,
     +                           THPNX,THPNY,SPNXL,SPNYW,INDLY,NLPRM,
     +                           NLPCN,NDLYR,NDLYC,INDCS,SKNOD,NCASE,
     +                           SK)
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/STFRR/ STFRR
        COMMON/STFCV/ STFCV
        COMMON/ESTIF/ ESTIF
        COMMON/GSTIF/ GSTIF
        COMMON/CLOAD/ CLOAD
        COMMON/CCSLD/ CCSLD
        COMMON/COORD/ COORD
        DIMENSION       ASDIS(918,5),TREAC(25,3,5),DISPL(378,5),
     +                  COORD(153,3),LNODS(36,9),MATNO(36),PROPS(10,3),
     +                  ESTIF(54,54),GSTIF(54,54),STFRR(252,252),
     +                  INDPL(153),NELLA(3),NELPX(3),NELPY(3),
     +                  THBAS(3,3),THPNX(3,4),THPNY(4,3),SPNXL(3),
     +                  SPNYW(3),CLOAD(918,5),CCSLD(378,5),
     +                  STFCV(126,378),SKNOD(153),INDLY(153,3),
     +                  INDCS(153),NLPRM(3),NLPCN(3),NDLYR(42,2),
     +                  NDLYC(16,2)
        WRITE(8,111)
111     FORMAT(' subroutine solveq')
        CALL    RAZERO  (DISPL,378,5)
        CALL    RAZERO  (ASDIS,918,5)
        NCVRM=NCPRM*NDOFN
        NCVCN=NCPCN*NDOFN
C*******STATIC CONDENSATION**********************************************
        WRITE(8,112)
112     FORMAT(' static condensation')
        NCVAR=NCVCN+NCVRM
        DO 35 K=1,NCVCN
        LL=NCVAR-K
        KK=LL+1
        DO 25 L=1,LL
        IF(KK .LE. NCVRM) GO TO 105
        DUM=STFCV(KK-NCVRM,L)/STFCV(KK-NCVRM,KK)
        GO TO 115
  105   DUM=STFRR(KK,L)/STFRR(KK,KK)
  115   DO 15 M=1,L
        IF(L .LE. NCVRM .AND. KK .GT. NCVRM) GO TO 125
        IF(L .GT. NCVRM .AND. KK .GT. NCVRM) GO TO 135
        IF(L .LE. NCVRM .AND. KK .LE. NCVRM) GO TO 145
        STFCV(L-NCVRM,M)=STFCV(L-NCVRM,M)-STFRR(KK,M)*DUM
        GO TO 15
  125   STFRR(L,M)=STFRR(L,M)-STFCV(KK-NCVRM,M)*DUM
        GO TO 15
  135   STFCV(L-NCVRM,M)=STFCV(L-NCVRM,M)-STFCV(KK-NCVRM,M)*DUM
        GO TO 15
  145   STFRR(L,M)=STFRR(L,M)-STFRR(KK,M)*DUM
  15    CONTINUE
        DO 10 JCASE=1,NCASE
  10    CCSLD(L,JCASE)=CCSLD(L,JCASE)-CCSLD(KK,JCASE)*DUM
  25    CONTINUE
  35    CONTINUE
        DO 45 K=2,LL
        DO 45 L=1,K-1
  45    STFRR(L,K)=STFRR(K,L)
C*******SOLUTION PHASE***************************************************
```

```
      WRITE(8,113)
113   FORMAT(' solution phase')
      CALL    MATINV  (NCVRM)
      DO 99 ICASE=1,NCASE
      DO 50 ICVRM=1,NCVRM
      DO 50 JCVRM=1,NCVRM
 50   DISPL(ICVRM,ICASE)=DISPL(ICVRM,ICASE)+CCSLD(JCVRM,ICASE)*
     + STFRR(ICVRM,JCVRM)
C******BACK SUBSTITUTION FOR CONDENSED D.O.F.**************************
      WRITE(8,114)
114   FORMAT(' back substitution')
      DO 85 J=1,NCVCN
      JJ=NCVRM+J
      DUM=0.
      K=JJ-1
      DO 75 L=1,K
 75   DUM=DUM+STFCV(J,L)*DISPL(L,ICASE)
 85   DISPL(JJ,ICASE)=(CCSLD(J+NCVRM,ICASE)-DUM)/STFCV(J,JJ)
C*******RE-ARRANGEMENT OF NODAL DISPLACEMENTS**************************
      WRITE(8,1015)
1015  FORMAT(' rearrangement of displacements')
      DO 60 ICPRM=1,NCPRM
      ICVRM=(ICPRM-1)*NDOFN
      DO 53 IPOIN=1,NPOIN
 53   IF(INDCS(IPOIN) .EQ. ICPRM) GO TO 57
 57   ITOTV=(IPOIN-1)*NDOFN
      DO 60 IDOFN=1,NDOFN
      ITOTV=ITOTV+1
      ICVRM=ICVRM+1
 60   ASDIS(ITOTV,ICASE)=DISPL(ICVRM,ICASE)
      DO 95 ICPCN=1,NCPCN
      ICVCN=(ICPCN-1)*NDOFN
      DO 63 IPOIN=1,NPOIN
 63   IF(INDCS(IPOIN) .EQ. -ICPCN) GO TO 67
 67   ITOTV=(IPOIN-1)*NDOFN
      DO 95 IDOFN=1,NDOFN
      ITOTV=ITOTV+1
      ICVCN=ICVCN+1
 95   ASDIS(ITOTV,ICASE)=DISPL(ICVCN+NCVRM,ICASE)
 99   CONTINUE
      KZ=NLAYR
      NLKPR=NLPRM(KZ)
      NLKPC=NLPCN(KZ)
      CALL    SOLVLY  (KZ,NCASE,NLKPR,NLKPC,NDOFN,NDLYR,NDLYC,
     +                 ASDIS)
      DO 100 IT=2,3
      CALL    GSTFNS  (ASDIS,LNODS,MATNO,NEVAB,NGAUB,NGAUS,
     +                 NGAUM,NNODE,PROPS,NDOFN,NCPRM,NCPCN,
     +                 NPNLX,NPNLY,NLAYR,NELLA,NELPX,NELPY,
     +                 SPNXL,SPNYN,THBAS,THPNX,THPNY,SKNOD,
     +                 NPOIN,NCASE,INDPL,INDLY,NLPRM,NLPCN,
     +                 NDLYR,NDLYC,INDCS,SK,IT)
100   CONTINUE
C*******REACTION CALCULATION*******************************************
      WRITE(8,116)
116   FORMAT(' reaction calculation')
      DO 110 ICASE=1,NCASE
      NSPRT=0
      DO 90 IPOIN=1,NPOIN
      IF(SKNOD(IPOIN) .EQ. 0.) GO TO 90
      ITOTV=(IPOIN-1)*NDOFN
      NSPRT=NSPRT+1
      DO 80 IDOFN=1,3
      ITOTV=ITOTV+1
      TREAC(NSPRT,IDOFN,ICASE)=SKNOD(IPOIN)*ASDIS(ITOTV,ICASE)
 80   CONTINUE
 90   CONTINUE
110   CONTINUE
      RETURN
```

```
      END

      SUBROUTINE      SOLVLY  (KZ,NCASE,NLKPR,NLKPC,NDOFN,NDLYR,NDLYC,
     +                         ASDIS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SLYCV/ SLYCV
      COMMON/CLYLD/ CLYLD
      DIMENSION       NDLYR(42,2),NDLYC(16,2),DISLY(288,5),
     +                ASDIS(918,5),SLYCV(36,288),CLYLD(288,5)
      WRITE(8,111)
111   FORMAT(' subroutine solvly')
      NLPRM=NLKPR
      NLPCN=NLKPC
      NLVRM=NLPRM*NDOFN
      NLVCN=NLPCN*NDOFN
      DO 30 ILPRM=1,NLPRM
      IPOIN=NDLYR(ILPRM,KZ)
      ILVRM=(ILPRM-1)*NDOFN
      IPOSN=(IPOIN-1)*NDOFN
      DO 30 IDOFN=1,NDOFN
      ILVRM=ILVRM+1
      IPOSN=IPOSN+1
      DO 20 ICASE=1,NCASE
 20   DISLY(ILVRM,ICASE)=ASDIS(IPOSN,ICASE)
 30   CONTINUE
      DO 50 ICASE=1,NCASE
      DO 50 J=1,NLVCN
      JJ=NLVRM+J
      DUM=0.
      K=JJ-1
      DO 40 L=1,K
 40   DUM=DUM+SLYCV(J,L)*DISLY(L,ICASE)
 50   DISLY(JJ,ICASE)=(CLYLD(J+NLVRM,ICASE)-DUM)/SLYCV(J,JJ)
      DO 90 ILPCN=1,NLPCN
      IPOIN=NDLYC(ILPCN,KZ)
      ILVCN=(ILPCN-1)*NDOFN
      IPOSN=(IPOIN-1)*NDOFN
      DO 80 IDOFN=1,NDOFN
      ILVCN=ILVCN+1
      IPOSN=IPOSN+1
      DO 80 ICASE=1,NCASE
 80   ASDIS(IPOSN,ICASE)=DISLY(ILVCN+NLVRM,ICASE)
 90   CONTINUE
      RETURN
      END

      SUBROUTINE      SOLVPN  (NCASE,NPPRM,NPPCN,NPOIN,NDOFN,IPNOD,
     +                         ASDIS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PNSCV/ PNSCV
      COMMON/CPNLD/ CPNLD
      DIMENSION       IPNOD(153),PNLDS(150,5),ASDIS(918,5),
     +                PNSCV(54,150),CPNLD(150,5)
      NPVRM=NPPRM*NDOFN
      NPVCN=NPPCN*NDOFN
      DO 30 IPPRM=1,NPPRM
      DO 10 IPOIN=1,NPOIN
 10   IF(IPNOD(IPOIN) .EQ. IPPRM) GO TO 20
 20   IPVRM=(IPPRM-1)*NDOFN
      IPOSN=(IPOIN-1)*NDOFN
      DO 30 IDOFN=1,NDOFN
      IPVRM=IPVRM+1
      IPOSN=IPOSN+1
      DO 30 ICASE=1,NCASE
 30   PNLDS(IPVRM,ICASE)=ASDIS(IPOSN,ICASE)
      DO 50 ICASE=1,NCASE
      DO 50 J=1,NPVCN
      JJ=NPVRM+J
      DUM=0.
```

```
      K=JJ-1
      DO 40 L=1,K
40    DUM=DUM+PNSCV(J,L)*PNLDS(L,ICASE)
50    PNLDS(JJ,ICASE)=(CPNLD(J+NPVRM,ICASE)-DUM)/PNSCV(J,JJ)
      DO 90 IPPCN=1,NPPCN
      DO 60 IPOIN=1,NPOIN
60    IF(IPNOD(IPOIN) .EQ. -IPPCN) GO TO 70
70    IPVCN=(IPPCN-1)*NDOFN+NPVRM
      IPOSN=(IPOIN-1)*NDOFN
      DO 80 IDOFN=1,NDOFN
      IPVCN=IPVCN+1
      IPOSN=IPOSN+1
      DO 80 ICASE=1,NCASE
80    ASDIS(IPOSN,ICASE)=PNLDS(IPVCN,ICASE)
90    CONTINUE
      RETURN
      END

      SUBROUTINE      MATINV  (N)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/STFRR/ ST
      DIMENSION       ST(252,252)
      WRITE(8,111)
111   FORMAT(' subroutine matinv')
      DO 19 I=1,N
      Z=ST(I,I)
      ST(I,I)=1.
      DO 60 J=1,N
60    ST(I,J)=ST(I,J)/Z
      DO 19 K=1,N
      IF(K-I)3,19,3
3     Z=ST(K,I)
      ST(K,I)=0.
      DO 4 J=1,N
4     ST(K,J)=ST(K,J)-Z*ST(I,J)
19    CONTINUE
      RETURN
      END

      SUBROUTINE      GEOMET  (NPNLX,NPNLY,NLAYR,NELPX,NELPY,NELLA,
     +                         LNODS,PROPS,MATNO,NCASE,INDPL,
     +                         NPOIN,NDOFN,NELEM,NTOTV,THPNX,THPNY,
     +                         THBAS,HIGHT,ZHIGH,THCBS,XLNGT,YWDTH,
     +                         SPNXL,SPNYW,XLMOD,YWMOD,NCPRM,NCPCN,
     +                         NNODE,NLPRM,NLPCN,NDLYR,NDLYC,INDLY,
     +                         INDCS,SK)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COORD/ COORD
      DIMENSION       NELPX(3),NELPY(3),NELLA(3),SPNXL(3),
     +                SPNYW(3),HIGHT(3),THBAS(3,3),THPNX(3,4),
     +                THPNY(4,3),COORD(153,3),LNODS(36,9),PROPS(10,3),
     +                MATNO(36),NMATB(3,3),NMATL(3,4,3),NMATW(4,3,3),
     +                TITLE(20),INDPL(153),INDLY(153,3),NLPRM(3),
     +                NLPCN(3),NDLYR(42,2),NDLYC(16,2),INDCS(153)
      WRITE(8,111)
111   FORMAT(' subroutine geomet')
      READ(5,920)  TITLE
      WRITE(6,921) TITLE
      READ(5,*)       XLNGT,YWDTH,ZHIGH
      WRITE(6,1000) XLNGT,YWDTH,ZHIGH
      READ(5,*)       NPNLX,NPNLY,NLAYR
      WRITE(6,1010) NPNLX,NPNLY,NLAYR
      READ(5,*)       ILSPN,INSPN,IXTHC,IYTHC,NMATS,NCASE
      WRITE(6,1020) ILSPN,INSPN,IXTHC,IYTHC,NMATS,NCASE
      WRITE(6,910)
      DO 12 INATS=1,NMATS
      READ(5,*)     NUMAT,(PROPS(NUMAT,IPROP),IPROP=1,3)
12    WRITE(6,912) NUMAT,(PROPS(NUMAT,IPROP),IPROP=1,3)
      CALL    BASETH  (NMATB,IXTHC,IYTHC,THBAS,NPNLX,NPNLY,THCBS,SK)
```

```
      WRITE(6,99) SK
      WRITE(6,1030) THCBS
      WRITE(6,1040) ((IX,JY,NMATB(IX,JY),THBAS(IX,JY),JY=1,NPNLY),
     + IX=1,NPNLX)
      WRITE(6,1049)
      READ(5,*)         (HIGHT(I),NELLA(I),I=1,NLAYR)
      WRITE(6,1050) (I,HIGHT(I),NELLA(I),I=1,NLAYR)
      WRITE(6,1059)
      READ(5,*)          (NELPX(IX),IX=1,NPNLX)
      WRITE(6,1060) (IX,NELPX(IX),IX=1,NPNLX)
      WRITE(6,1069)
      READ(5,*)          (NELPY(JY),JY=1,NPNLY)
      WRITE(6,1070) (JY,NELPY(JY),JY=1,NPNLY)
      DO 10 KZ=1,NLAYR
      CALL    GEOLYR  (KZ,NPNLX,NPNLY,XLNGT,YWDTH,IXTHC,IYTHC,
     +                 ILSPN,IWSPN,THPNX,THPNY,SPNXL,SPNYW,NMATL,
     +                 NMATW,XLMOD,YWMOD)
      WRITE(6,1079)
      WRITE(6,1080) ((KZ,IX,JY,NMATL(IX,JY,KZ),THPNX(IX,JY),
     + IX=1,NPNLX),JY=1,NPNLY+1)
      WRITE(6,1089)
      WRITE(6,1090) ((KZ,JY,IX,NMATW(IX,JY,KZ),THPNY(IX,JY),
     + JY=1,NPNLY),IX=1,NPNLX+1)
 10   CONTINUE
      CALL    NOPEPS  (NPNLX,NPNLY,NLAYR,NELPX,NELPY,NELLA,NPOIN,
     +                 NELBX,NELBY,NELMZ,NELEM)
      NTOTV=NPOIN*NDOFN
      WRITE(6,1100) NELBX,NELBY,NELMZ
      WRITE(6,1110) NELEM,NPOIN
      CALL    NODCOR  (NPNLX,NPNLY,NLAYR,NELPX,NELPY,NELLA,
     +                 SPNXL,SPNYW,HIGHT,THCBS)
      WRITE(6,1119)
      WRITE(6,1120) (IPOIN,(COORD(IPOIN,I),I=1,3),IPOIN=1,NPOIN)
      NLPYT=0
      NLPXT=0
      CALL    ELMNOD  (NNODE,NPOIN,NCPRM,NCPCN,NLAYR,NPNLX,
     +                 NPNLY,NELBX,NELBY,NELMZ,NELLA,NELPX,
     +                 NELPY,LNODS,MATNO,NMATB,NMATL,NMATW,
     +                 INDPL,NLPRM,NLPCN,INDLY,NDLYR,NDLYC,
     +                 INDCS,HIGHT)
      WRITE(6,1129) (I,I=1,9)
      WRITE(6,1130) (IELEM,(LNODS(IELEM,INODE),INODE=1,9)
     + ,IELEM=1,NELEM)
 99   FORMAT(//10X,'ELASTIC MODULUS OF SUBGRADE REACTION =',E12.4)
 920  FORMAT(20A4)
 921  FORMAT(/////,1X,20A4)
 910  FORMAT(//10X,'ELEMENT PROPERTIES',/10X,18(1H*),//2X,'NUMBER',
     + 5X,'YOUNG MODULUS',5X,'POISSON RATIO',2X,
     + 'SPECIFIC GRAVITY',/2X,6(1H-),2(5X,13(1H-)),2X,16(1H-))
 912  FORMAT(I8,3E18.6)
 907  FORMAT(//,' NODE',6X,'CODE',6X,'FIXED VALUES')
1000  FORMAT(//10X,'TOTAL  LENGTH',5X,'TOTAL   WIDTH',5X,
     + 'TOTAL  HEIGHT',/10X,3(13(1H-),5X),/10X,3(F13.3,5X))
1010  FORMAT(//10X,'NO. OF PANELS',5X,'NO. OF PANELS',5X,
     + 'NO. OF LAYERS',/10X,' X direction ',5X,' Y direction ',5X,
     + ' Z direction ',/10X,3(13(1H-),5X),/10X,3(6X,I2,10X))
1020  FORMAT(//10X,'ILSPN IWSPN IXTHC IYTHC',/10X,4(5(1H-),1X),/10X,
     + 4(4X,I1,1X),////20X,'NO. OF MATERIAL TYPES=',I2,//,//20X,'NO. ',
     + 'OF LOAD CASE(S) =',I2,//)
1030  FORMAT(//10X,'MINIMUM THICKNESS OF BASE PANELS =',F6.3,//10X,
     + 'IX',5X,'JY',5X,'MATERIAL TYPE',5X,'THBAS (IX,JY)',/10X,2(2(1H-)
     + ,5X),2(13(1H-),5X))
1040  FORMAT(10X,I2,5X,I2,16X,I2,12X,F6.3)
1049  FORMAT(//10X,'LAYER #',5X,'HEIGHT OF LAYER',
     + 5X,'# OF ELEMENTS IN HEIGHT',/10X,7(1H-),5X,15(1H-),
     + 5X,23(1H-))
1050  FORMAT(15X,I2,14X,F6.3,26X,I2)
1059  FORMAT(//10X,'PANEL #',5X,'# OF ELEMENTS LENGTHWISE',/10X,
     + 7(1H-),5X,24(1H-))
```

```
1060    FORMAT(15X,I2,16X,I2)
1069    FORMAT(//10X,'PANEL #',5X,'# OF ELEMENTS WIDTHWISE',/10X,
     +  7(1H-),5X,23(1H-))
1070    FORMAT(15X,I2,16X,I2)
1079    FORMAT(//10X,'LAYER #',5X,'LONGT. PANEL #',5X,'TRANS.'
     +  ,' LOCATION #',5X,'MATERIAL TYPE',5X,'PANEL THICKNESS',
     +  /10X,7(1H-),5X,14(1H-),5X,17(1H-),5X,13(1H-),5X,15(1H-))
1080    FORMAT(15X,I2,17X,I2,20X,I2,16X,I2,15X,F5.3)
1089    FORMAT(//10X,'LAYER #',5X,'TRANS. PANEL #',5X,'LONGT.'
     +  ,' LOCATION #',5X,'MATERIAL TYPE',5X,'PANEL THICKNESS',
     +  /10X,7(1H-),5X,14(1H-),5X,17(1H-),5X,13(1H-),5X,15(1H-))
1090    FORMAT(15X,I2,17X,I2,20X,I2,16X,I2,15X,F5.3)
1100    FORMAT(//5X,'# OF ELMNTS LENGTHWISE',3X,'# OF ELMNTS',
     +  '  WIDTHWISE',3X,'# OF ELMNTS IN  HEIGHT',/5X,3(22(1H-),3X),
     +  /5X,3(11X,I2,12X))
1110    FORMAT(///10X,'TOTAL NO. OF ELEMENTS =',I5,//10X,'TOTAL  NO.'
     +  ,'  OF  NODES =',I5)
1119    FORMAT(//10X,'NODE',5X,'X COORD',5X,'Y COORD',5X,'Z COORD',/10X,
     +  4(1H-),5X,3(7(1H-),5X))
1120    FORMAT(10X,I4,5X,F7.3,5X,F7.3,5X,F7.3)
1129    FORMAT(//10X,'ELM #',9(1X,'NODE',I1,1X),/10X,5(1H-),9(1X,5(1H-),
     +  1X))
1130    FORMAT(10X,I4,9I7)
        RETURN
        END

        SUBROUTINE      BASETH  (NMATB,IXTHC,IYTHC,THBAS,NPNLX,NPNLY,
     +                           THCBS,SK)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION       THBAS(3,3),NMATB(3,3)
        READ(5,*) SK
        IF(IXTHC .NE. 0 .OR. IYTHC .NE. 0) GO TO 20
        READ(5,*) IBMAT,THCBS
        DO 10 IX=1,NPNLX
        DO 10 JY=1,NPNLY
        NMATB(IX,JY)=IBMAT
  10    THBAS(IX,JY)=THCBS
        RETURN
  20    IF(IXTHC .NE. 0 .AND. IYTHC .NE. 0) GO TO 60
        IF(IXTHC .NE. 0) GO TO 40
        READ(5,*) (NMATB(1,JY),THBAS(1,JY),JY=1,NPNLY)
        DO 30 IX=2,NPNLX
        DO 30 JY=1,NPNLY
        NMATB(IX,JY)=NMATB(1,JY)
  30    THBAS(IX,JY)=THBAS(1,JY)
        GO TO 80
  40    READ(5,*) (NMATB(IX,1),THBAS(IX,1),IX=1,NPNLX)
        DO 50 JY=2,NPNLY
        DO 50 IX=1,NPNLX
        NMATB(IX,JY)=NMATB(IX,1)
  50    THBAS(IX,JY)=THBAS(IX,1)
        GO TO 80
  60    DO 70 IX=1,NPNLX
  70    READ(5,*) (NMATB(IX,JY),THBAS(IX,JY),JY=1,NPNLY)
  80    THCBS=THBAS(1,1)
        DO 90 IX=1,NPNLX
        DO 90 JY=1,NPNLY
  90    IF(THBAS(IX,JY) .LT. THCBS) THCBS=THBAS(IX,JY)
        RETURN
        END

        SUBROUTINE      GEOLYR  (KZ,NPNLX,NPNLY,XLNGT,YWDTH,IXTHC,
     +                           IYTHC,ILSPN,IWSPN,THPNX,THPNY,SPNXL,
     +                           SPNYW,NMATL,NMATW,XLNOD,YWNOD)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION       THPNX(3,4),THPNY(4,3),SPNXL(3),SPNYW(3),
     +                  NMATL(3,4,3),NMATW(4,3,3)
        IF(IXTHC .NE. 0) GO TO 20
        READ(5,*) (NMATL(1,JY,KZ),THPNX(1,JY),JY=1,NPNLY+1)
```

```
      DO 10 IX=2,NPNLX
      DO 10 JY=1,NPNLY+1
      NMATL(IX,JY,KZ)=NMATL(1,JY,KZ)
 10   THPNX(IX,JY)=THPNX(1,JY)
      GO TO 40
 20   DO 30 IX=1,NPNLX
 30   READ(5,*) (NMATL(IX,JY,KZ),THPNX(IX,JY),JY=1,NPNLY+1)
 40   IF(IYTHC .NE. 0) GO TO 60
      READ(5,*) (NMATW(IX,1,KZ),THPNY(IX,1),IX=1,NPNLX+1)
      DO 50 JY=2,NPNLY
      DO 50 IX=1,NPNLX+1
      NMATW(IX,JY,KZ)=NMATW(IX,1,KZ)
 50   THPNY(IX,JY)=THPNY(IX,1)
      GO TO 80
 60   DO 70 JY=1,NPNLY
 70   READ(5,*) (NMATW(IX,JY,KZ),THPNY(IX,JY),IX=1,NPNLX+1)
 80   IF(KZ .GT. 1) RETURN
      THMYL=THPNY(1,1)
      DO 90 JY=2,NPNLY
 90   IF(THPNY(1,JY) .LT. THMYL) THMYL=THPNY(1,JY)
      THMYR=THPNY(NPNLX+1,1)
      DO 100 JY=2,NPNLY
 100  IF(THPNY(NPNLX+1,JY) .LT. THMYR) THMYR=THPNY(NPNLX+1,JY)
      IF(ILSPN .NE. 0) GO TO 120
      DO 110 IX=1,NPNLX
      XLMOD=XLNGT-(THMYL+THMYR)/2.
 110  SPNXL(IX)=XLMOD/NPNLX
      GO TO 140
 120  READ(5,*) (SPNXL(IX),IX=1,NPNLX)
      XLMOD=0.
      DO 130 IX=1,NPNLX
 130  XLMOD=XLMOD+SPNXL(IX)
      XLMTR=XLNGT-(THMYL+THMYR)/2.
      IF(XLMOD .EQ. XLMTR) GO TO 140
      WRITE(*,1000) XLMOD,XLMTR
 1000 FORMAT('CHECK FOR LENGTH OF PANELS IN X DIRECTION. XLMOD=',
     + F10.3,'OR',F10.3)
 140  THMXL=THPNX(1,1)
      DO 150 IX=2,NPNLX
 150  IF(THPNX(IX,1) .LT. THMXL) THMXL=THPNX(IX,1)
      THMXR=THPNX(1,NPNLY+1)
      DO 160 IX=2,NPNLX
 160  IF(THPNX(IX,NPNLY+1) .LT. THMXR) THMXR=THPNX(IX,NPNLY+1)
      IF(IWSPN .NE. 0) GO TO 180
      DO 170 JY=1,NPNLY
      YWMOD=YWDTH-(THMXL+THMXR)/2.
 170  SPNYW(JY)=YWMOD/NPNLY
      RETURN
 180  READ(5,*) (SPNYW(JY),JY=1,NPNLY)
      YWMOD=0.
      DO 200 JY=1,NPNLY
 200  YWMOD=YWMOD+SPNYW(JY)
      YWMTR=YWDTH-(THMXL+THMXR)/2.
      IF(YWMOD .EQ. YWMTR) RETURN
      WRITE(*,1010) YWMOD,YWMTR
 1010 FORMAT('CHECK FOR LENGTH OF PANELS IN Y DIRECTION. YWMOD=',
     + F10.3,'OR',F10.3)
      RETURN
      END

      SUBROUTINE      NOPEPS  (NPNLX,NPNLY,NLAYR,NELPX,NELPY,NELLA,
     +                         NPOIN,NELBY,NELBY,NELMZ,NELEM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       NELPX(3),NELPY(3),NELLA(3)
      NOPLX=NPNLX*(NPNLY+1)
      NOPLY=NPNLY*(NPNLX+1)
      NPLYR=NOPLX+NOPLY
      NOPLB=NPNLX*NPNLY
      NLPXT=0
```

```
      NLPYT=0
      NELMZ=0
      NELBX=0
      NELBY=0
      DO 10 IX=1,NPNLX
 10   NELBX=NELBX+NELPX(IX)
      DO 20 JY=1,NPNLY
 20   NELBY=NELBY+NELPY(JY)
      DO 30 KZ=1,NLAYR
      NELMZ=NELMZ+NELLA(KZ)
      NLPXT=NLPXT+NELBX*(NPNLY+1)*NELLA(KZ)
      NLPYT=NLPYT+NELBY*(NPNLX+1)*NELLA(KZ)
 30   CONTINUE
      NELMB=NELBX*NELBY
      NPTPX=(NELBX*2+1)*(NPNLY+1)
      NPTPY=(NELBY*2+1)*(NPNLX+1)
      NPTHR=NPTPX+NPTPY-(NPNLX+1)*(NPNLY+1)
      NPTWL=2*NPTHR*NELMZ
      NPOIN=NPTWL+(2*NELBX+1)*(2*NELBY+1)
      GO TO 70
 70   NPLTT=NOPLB+NLAYR*NPLYR
      NELEM=NLPXT+NLPYT+NELMB
      RETURN
      END

      SUBROUTINE      NODCOR  (NPNLX,NPNLY,NLAYR,NELPX,NELPY,NELLA,
     +                         SPNXL,SPNYW,HIGHT,THCBS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COORD/ COORD
      DIMENSION       NELPX(3),NELPY(3),NELLA(3),SPNXL(3),SPNYW(3),
     +                HIGHT(3),COORD(153,3)
      HIGHT(1)=HIGHT(1)+THCBS/2.
      XCOOR=0.
      NODEP=0
      DO 100 IX=1,NPNLX
      XINCR=SPNXL(IX)/(2.*NELPX(IX))
      NPNTX=2*NELPX(IX)+1
      I1=2
      IF(IX .EQ. 1) I1='
      DO 200 I=I1,NPNTX
      XCOOR=XCOOR+XINCR
      IF(I .EQ. 1) XCOOR=XCOOR-XINCR
      YCOOR=0.
      DO 300 JY=1,NPNLY
      YINCR=SPNYW(JY)/(2.*NELPY(JY))
      NPNTY=2*NELPY(JY)+1
      J1=2
      IF(JY .EQ. 1) J1=1
      DO 400 J=J1,NPNTY
      YCOOR=YCOOR+YINCR
      IF(J .EQ. 1) YCOOR=YCOOR-YINCR
      NODEP=NODEP+1
      COORD(NODEP,1)=XCOOR
      COORD(NODEP,2)=YCOOR
      COORD(NODEP,3)=0.
      IF(I .EQ. NPNTX .OR. J .EQ. NPNTY) GO TO 10
      IF(I .GT. 1 .AND. J .GT. 1) GO TO 400
 10   DO 500 KZ=1,NLAYR
      KK=2*NELLA(KZ)
      ZINCR=HIGHT(KZ)/KK
      DO 600 K=1,KK
      NODEN=NODEP+K
      COORD(NODEN,1)=XCOOR
      COORD(NODEN,2)=YCOOR
600   COORD(NODEN,3)=COORD(NODEP,3)+K*ZINCR
      NODEP=NODEN
500   CONTINUE
400   CONTINUE
300   CONTINUE
```

```
200     CONTINUE
100     CONTINUE
        RETURN
        END

        SUBROUTINE      ELMNOD  (NNODE,NPOIN,NCPRM,NCPCN,NLAYR,NPNLX,
     +                           NPNLY,NELBX,NELBY,NELMZ,NELLA,NELPX,
     +                           NELPY,LNODS,MATNO,NMATB,NMATL,NMATH,
     +                           INDPL,NLPRM,NLPCN,INDLY,NDLYR,NDLYC,
     +                           INDCS,HIGHT)
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/COORD/ COORD
        DIMENSION        NELLA(3),NELPX(3),NELPY(3),LNODS(36,9),
     +                   MATNO(36),NMATB(3,3),NMATL(3,4,3),NMATH(4,3,3),
     +                   KONTP(153),INDPL(153),INDLY(153,3),INDCS(153),
     +                   KONTC(153),NDLYR(42,2),NDLYC(16,2),NLPRM(3),
     +                   NLPCN(3),COORD(153,3),HIGHT(3)
        CALL     IVZERO  (INDPL,153)
        CALL     ELMPB1  (NPNLX,NPNLY,NELBY,NELMZ,NELPX,NELPY,NLPBT,
     +                    INPYY,INPYX,LNODS,MATNO,NMATB,KONTP)
        NELMI=1
        NELMF=NLPBT
        CALL     NODVPL  (NELMI,NELMF,NNODE,LNODS,INDPL,KONTP)
        DO 10 KZ=1,NLAYR
        CALL     ELMPY1  (KZ,NPNLX,NPNLY,NELBY,NELMZ,NELLA,NELPX,
     +                    NELPY,NLPBT,NLPYT,NLPXT,INPYY,INPYX,LNODS,
     +                    MATNO,NMATH,KONTP)
        NELEM=NELLA(KZ)*NELBY*(NPNLX+1)
        NELMF=NLPBT+NLPYT+NLPXT
        NELMI=NELMF-NELEM+1
        CALL     NODVPL  (NELMI,NELMF,NNODE,LNODS,INDPL,KONTP)
        CALL     ELMPX1  (KZ,NPNLX,NPNLY,NELMZ,NELLA,NELPX,NELPY,
     +                    NLPBT,NLPYT,NLPXT,INPYY,INPYX,LNODS,
     +                    MATNO,NMATL,KONTP)
        NELEM=NELLA(KZ)*NELBX*(NPNLY+1)
        NELMF=NLPBT+NLPYT+NLPXT
        NELMI=NELMF-NELEM+1
        CALL     NODVPL  (NELMI,NELMF,NNODE,LNODS,INDPL,KONTP)
 10     CONTINUE
        NPPRM=0
        NPPCN=0
        DO 30 IPOIN=1,NPOIN
        IVNOD=INDPL(IPOIN)
        IF(IVNOD .GT. 0) GO TO 20
        NPPCN=NPPCN+1
        INDPL(IPOIN)=-NPPCN
        GO TO 30
 20     NPPRM=NPPRM+1
        INDPL(IPOIN)=NPPRM
 30     CONTINUE
        IELEM=0
        CALL     IAZERO  (INDLY,153,3)
        CALL     IVZERO  (INDCS,153)
        CALL     IVZERO  (KONTP,153)
        CALL     BASELM  (IELEM,NPNLX,NPNLY,NELPX,NELPY,LNODS,
     +                    KONTP,INDPL,INDLY,INDCS)
        ZBOT=0.
        DO 60 KZ=1,NLAYR
        ZTOP=ZBOT+HIGHT(KZ)
        CALL     TRPELM  (IELEM,KZ,ZBOT,ZTOP,NPNLX,NPNLY,
     +                    NLAYR,NELPY,NELLA,LNODS,KONTP,INDPL,
     +                    INDLY,INDCS)
        CALL     LNPELM  (IELEM,KZ,ZBOT,ZTOP,NPNLX,NPNLY,
     +                    NLAYR,NELPX,NELLA,LNODS,KONTP,INDPL,
     +                    INDLY,INDCS)
        NLPRM(KZ)=0
        NLPCN(KZ)=0
        DO 50 IPOIN=1,NPOIN
        IF(INDLY(IPOIN,KZ) .EQ.  0) GO TO 50
```

```
      IF(INDLY(IPOIN,KZ) .EQ. -1) GO TO 40
      NLPRM(KZ)=NLPRM(KZ)+1
      INDLY(IPOIN,KZ)=NLPRM(KZ)
      NDLYR(NLPRM(KZ),KZ)=IPOIN
      IF(INDCS(IPOIN) .EQ. 0) INDCS(IPOIN)=-1
      GO TO 50
40    NLPCN(KZ)=NLPCN(KZ)+1
      INDLY(IPOIN,KZ)=-NLPCN(KZ)
      NDLYC(NLPCN(KZ),KZ)=IPOIN
50    CONTINUE
      ZBOT=ZTOP
      CALL    IVZERO  (KONTP,153)
60    CONTINUE
      NCPRM=0
      NCPCN=0
      CALL    IVZERO  (KONTC,153)
      DO 80 IPOIN=1,NPOIN
      IF(INDCS(IPOIN) .EQ. 0) GO TO 80
      KONTC(IPOIN)=KONTC(IPOIN)+1
      IF(KONTC(IPOIN) .GT. 1) GO TO 80
      IF(INDCS(IPOIN) .EQ. -1) GO TO 70
      NCPRM=NCPRM+1
      INDCS(IPOIN)=NCPRM
      GO TO 80
70    NCPCN=NCPCN+1
      INDCS(IPOIN)=-NCPCN
80    CONTINUE
      RETURN
      END

      SUBROUTINE      ELMPB1  (NPNLX,NPNLY,NELBY,NELMZ,NELPX,NELPY,
     +                         NLPBT,INPYY,INPYX,LNODS,MATNO,NMATB,
     +                         KONTP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       KNODE(9),NELPX(3),NELPY(3),LNODS(36,9),
     +                MATNO(36),NMATB(3,3),KONTP(153)
      CALL    IVZERO  (KONTP,153)
      NLPBT=0
      INPYY=(2*NELMZ+1)*(2*NELBY+1)
      INPYX=0
      DO 10 JY=1,NPNLY
10    INPYX=INPYX+2*NELPY(JY)-1
      INPYX=INPYX+(2*NELMZ+1)*(NPNLY+1)
      KNODE(1)=1
      KNODE(2)=INPYY+1
      KNODE(3)=INPYY+INPYX+1
      DO 400 JY=1,NPNLY
      DO 300 IX=1,NPNLX
      DO 200 I=1,NELPX(IX)
      DO 100 J=1,NELPY(JY)
      NLPBT=NLPBT+1
      MATNO(NLPBT)=NMATB(IX,JY)
      IF(J .GT. 1) GO TO 30
      KNOD1=KNODE(3)
      KNODE(4)=KNODE(3)+2*NELMZ+1
      DO 20 INODE=8,9
20    KNODE(INODE)=KNODE(INODE-7)+2*NELMZ+1
      GO TO 60
30    DO 40 INODE=1,3
40    KNODE(INODE)=KNODE(8-INODE)
      KNODE(4)=KNODE(3)+1
      DO 50 INODE=8,9
50    KNODE(INODE)=KNODE(INODE-7)+1
      IF(I .EQ. 1)          KNODE(8)=KNODE(8)+2*NELMZ
      IF(I .EQ. NELPX(IX)) KNODE(4)=KNODE(4)+2*NELMZ
60    KNODE(5)=KNODE(4)+1
      KNODE(6)=KNODE(9)+1
      KNODE(7)=KNODE(8)+1
      IF(I .EQ. 1)          KNODE(7)=KNODE(7)+2*NELMZ
```

```
      IF(I .EQ. NELPX(IX)) KNODE(5)=KNODE(5)+2*NELMZ
      DO 70 INODE=1,9
      LNODE=KNODE(INODE)
      LNODS(NLPBT,INODE)=LNODE
70    KONTP(LNODE)=KONTP(LNODE)+1
100   CONTINUE
      IF(IX .EQ. NPNLX .AND. I .EQ. NELPX(IX)) GO TO 210
      KNODE(1)=KNOD1
      KNODE(2)=KNODE(1)+INPYX
      KNODE(3)=KNODE(1)+2*INPYX
      IF(I-NELPX(IX)+1) 200,110,120
110   IF(JY .EQ. 1) GO TO 200
      INPYB=0
      DO 11 JJY=1,JY-1
 11   INPYB=INPYB+(2*NELPY(JJY)-1)*2*NELMZ
      KNODE(3)=KNODE(3)+INPYB
      GO TO 200
120   INPYA=0
      DO 12 JJY=JY,NPNLY
 12   INPYA=INPYA+(2*NELPY(JJY)-1)*2*NELMZ
      DO 13 INODE=2,3
 13   KNODE(INODE)=KNODE(INODE)+INPYA
      IF(NELPX(IX+1) .GT. 1 .OR. JY .EQ. 1) GO TO 200
      INPYB=0
      DO 14 JJY=1,JY-1
 14   INPYB=INPYB+(2*NELPY(JJY)-1)*2*NELMZ
      KNODE(3)=KNODE(3)+INPYB
200   CONTINUE
210   CONTINUE
300   CONTINUE
      IF(JY .EQ. NPNLY) RETURN
      INPYI=0
      INPXM=0
      DO 320 JJY=1,JY
      INPYI=INPYI+2*NELPY(JJY)*(2*NELMZ+1)
320   INPXM=INPXM+2*NELPY(JJY)-1
      INPXM=INPXM+JY*(2*NELMZ+1)
      KNODE(1)=INPYI+1
      KNODE(2)=INPYI+INPXM+1
      KNODE(3)=KNODE(2)+INPYX
      IF(NELPX(1) .EQ. 1) KNODE(3)=INPYY+INPYX+INPYI+1
400   CONTINUE
      RETURN
      END

      SUBROUTINE      ELMPY1  (KZ,NPNLX,NPNLY,NELBY,NELMZ,NELLA,NELPX,
     +                          NELPY,NLPBT,NLPYT,NLPXT,INPYY,INPYX,
     +                          LNODS,MATNO,NMATN,KONTP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       KNODE(9),NELPX(3),NELPY(3),NELLA(3),
     +                LNODS(36,9),MATNO(36),NMATN(4,3,3),KONTP(153)
      CALL    IVZERO  (KONTP,153)
      NODIN=0
      DO 400 IX=1,NPNLX+1
      KNODE(1)=4*NELMZ+1
      KNODE(2)=2*NELMZ
      KNODE(3)=-1
      IF(KZ .EQ. 1) GO TO 30
      INCNZ=0
      DO 10 KKZ=1,KZ-1
 10   INCNZ=INCNZ+2*NELLA(KKZ)
      DO 20 INODE=1,3
 20   KNODE(INODE)=KNODE(INODE)+INCNZ
 30   DO 40 INODE=1,3
 40   KNODE(INODE)=KNODE(INODE)+NODIN
      DO 300 JY=1,NPNLY
      DO 200 J=1,NELPY(JY)
      DO 100 K=1,NELLA(KZ)
      NLPYT=NLPYT+1
```

```
      IELEM=NLPBT+NLPYT+NLPXT
      MATNO(IELEM)=NMATH(IX,JY,KZ)
      KNODE(4)=KNODE(3)+1
      KNODE(5)=KNODE(3)+2
      KNODE(6)=KNODE(2)+2
      KNODE(7)=KNODE(1)+2
      KNODE(8)=KNODE(1)+1
      KNODE(9)=KNODE(2)+1
      DO 50 INODE=1,9
      KNODE(INODE)=KNODE(INODE)+2
      LNODE=KNODE(INODE)
      LNODS(IELEM,INODE)=LNODE
 50   KONTP(LNODE)=KONTP(LNODE)+1
100   CONTINUE
      IF(JY .EQ. NPNLY .AND. J .EQ. NELPY(JY)) GO TO 210
      DO 110 INODE=1,3
110   KNODE(INODE)=KNODE(INODE)+4*NELMZ-2*NELLA(KZ)+2
200   CONTINUE
210   CONTINUE
300   CONTINUE
      DO 310 INODE=1,3
310   KNODE(INODE)=KNODE(INODE)-(NELBY-1)*(4*NELMZ+2)
      IF(IX .EQ. NPNLX+1) RETURN
      NODIN=NODIN+INPYY+(2*NELPX(IX)-1)*INPYX
400   CONTINUE
      RETURN
      END

      SUBROUTINE      ELMPX1  (KZ,NPNLX,NPNLY,NELMZ,NELLA,NELPX,NELPY,
     +                         NLPBT,NLPYT,NLPXT,INPYY,INPYX,LNODS,
     +                         MATNO,NMATL,KONTP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION       KNODE(9),NELPX(3),NELPY(3),NELLA(3),
     +                LNODS(36,9),MATNO(36),NMATL(3,4,3),KONTP(153)
      CALL   IVZERO  (KONTP,153)
      KNODE(1)=-1
      KNODE(2)=INPYY-1
      KNODE(3)=INPYY+INPYX-1
      INCNZ=0
      IF(KZ .EQ. 1) GO TO 20
      DO 10 KKZ=1,KZ-1
 10   INCNZ=INCNZ+2*NELLA(KKZ)
 20   DO 400 JY=1,NPNLY+1
      DO 30 INODE=1,3
 30   KNODE(INODE)=KNODE(INODE)+INCNZ
      DO 300 IX=1,NPNLX
      DO 200 I=1,NELPX(IX)
      DO 100 K=1,NELLA(KZ)
      NLPXT=NLPXT+1
      IELEM=NLPBT+NLPYT+NLPXT
      MATNO(IELEM)=NMATL(IX,JY,KZ)
      DO 40 INODE=1,3
 40   KNODE(INODE)=KNODE(INODE)+2
      KNODE(4)=KNODE(3)+1
      KNODE(5)=KNODE(3)+2
      KNODE(6)=KNODE(2)+2
      KNODE(7)=KNODE(1)+2
      KNODE(8)=KNODE(1)+1
      KNODE(9)=KNODE(2)+1
      DO 50 INODE=1,9
      LNODE=KNODE(INODE)
      LNODS(IELEM,INODE)=LNODE
 50   KONTP(LNODE)=KONTP(LNODE)+1
100   CONTINUE
      IF(IX .EQ. NPNLX .AND. I .EQ. NELPX(IX)) GO TO 210
      KNODE(1)=KNODE(3)-2*NELLA(KZ)
      KNODE(2)=KNODE(1)+INPYX
      KNODE(3)=KNODE(1)+2*INPYX
      IF(I-NELPX(IX)+1) 200,110,120
```

```
110     IF(JY .EQ. 1) GO TO 200
        INPYB=0
        DO 11 JJY=1,JY-1
 11     INPYB=INPYB+(2*NELPY(JJY)-1)*2*NELMZ
        KNODE(3)=KNODE(3)+INPYB
        GO TO 200
120     IF(JY .EQ. NPNLY+1) GO TO 14
        INPYA=0
        DO 12 JJY=JY,NPNLY
 12     INPYA=INPYA+(2*NELPY(JJY)-1)*2*NELMZ
        DO 13 INODE=2,3
 13     KNODE(INODE)=KNODE(INODE)+INPYA
 14     IF(NELPX(IX+1) .GT. 1 .OR. JY .EQ. 1) GO TO 200
        INPYB=0
        DO 15 JJY=1,JY-1
 15     INPYB=INPYB+(2*NELPY(JJY)-1)*2*NELMZ
        KNODE(3)=KNODE(3)+INPYB
200     CONTINUE
210     CONTINUE
300     CONTINUE
        IF(JY .EQ. NPNLY+1) RETURN
        INPYI=0
        INPXM=0
        DO 340 JJY=1,JY
        INPYI=INPYI+2*NELPY(JJY)*(2*NELMZ+1)
340     INPXM=INPXM+2*NELPY(JJY)-1
        INPXM=INPXM+JY*(2*NELMZ+1)
        KNODE(1)=INPYI-1
        KNODE(2)=INPYY+INPXM-1
        KNODE(3)=KNODE(2)+INPYX
        IF(NELPX(1) .EQ. 1) KNODE(3)=INPYY+INPYX+INPYI-1
400     CONTINUE
        RETURN
        END

        SUBROUTINE      NODVPL  (NELMI,NELMF,NNODE,LNODS,INDPL,KONTP)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION       LNODS(36,9),INDPL(153),KONTP(153)
        DO 10 IELEM=NELMI,NELMF
        DO 10 INODE=1,NNODE
        LNODE=LNODS(IELEM,INODE)
        KOUNT=KONTP(LNODE)
        IF(KOUNT .EQ. 4) GO TO 10
        ICODE=INODE-(INODE/2)*2
        IF(ICODE .EQ. 0 .AND. KOUNT .GT. 1) GO TO 10
        IF(INODE .EQ. 9) GO TO 10
        INDPL(LNODE)=1
 10     CONTINUE
        RETURN
        END

        SUBROUTINE      BASELM  (IELEM,NPNLX,NPNLY,NELPX,NELPY,LNODS,
     +                           KONTP,INDPL,INDLY,INDCS)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION       NELPX(3),NELPY(3),LNODS(36,9),KONTP(153),
     +                  INDPL(153),INDLY(153,3),INDCS(153)
        DO 20 JPNLY=1,NPNLY
        DO 20 IPNLX=1,NPNLX
        DO 10 NUMEL=1,NELPY(JPNLY)*NELPX(IPNLX)
        IELEM=IELEM+1
        DO 10 INODE=1,8
        LNODE=LNODS(IELEM,INODE)
        IF(INDPL(LNODE) .LT. 0) GO TO 10
        KONTP(LNODE)=KONTP(LNODE)+1
        IF(KONTP(LNODE) .GT. 1) GO TO 10
        INDLY(LNODE,1)=1
        INDCS(LNODE)=1
 10     CONTINUE
 20     CONTINUE
```

```
      RETURN
      END

      SUBROUTINE      TRPELM  (IELEM,KZ,ZBOT,ZTOP,NPNLX,NPNLY,
     +                         NLAYR,NELPY,NELLA,LNODS,KONTP,INDPL,
     +                         INDLY,INDCS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COORD/ COORD
      DIMENSION       NELPY(3),NELLA(3),LNODS(36,9),COORD(153,3),
     +                KONTP(153),INDPL(153),INDLY(153,3),INDCS(153)
      PREC=1.E-15
      DO 30 IPNLX=1,NPNLX+1
      DO 30 JPNLY=1,NPNLY
      DO 20 NUMEL=1,NELPY(JPNLY)*NELLA(KZ)
      IELEM=IELEM+1
      DO 20 INODE=1,8
      LNODE=LNODS(IELEM,INODE)
      IF(INDPL(LNODE) .LT. 0) GO TO 20
      KONTP(LNODE)=KONTP(LNODE)+1
      IF(KONTP(LNODE) .GT. 1) GO TO 20
      ZCOOR=COORD(LNODE,3)
      IF(DABS(ZCOOR-ZBOT) .LT. PREC .OR. DABS(ZCOOR-ZTOP) .LT. PREC)
     + GO TO 10
      INDLY(LNODE,KZ)=-1
      GO TO 20
10    INDLY(LNODE,KZ)=1
      IF(KZ .EQ. 1   .AND. DABS(ZCOOR-ZBOT) .LT. PREC) INDCS(LNODE)=1
      IF(KZ.EQ.NLAYR .AND. DABS(ZCOOR-ZTOP) .LT. PREC) INDCS(LNODE)=1
20    CONTINUE
30    CONTINUE
      RETURN
      END

      SUBROUTINE      LNPELM  (IELEM,KZ,ZBOT,ZTOP,NPNLX,NPNLY,
     +                         NLAYR,NELPX,NELLA,LNODS,KONTP,INDPL,
     +                         INDLY,INDCS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COORD/ COORD
      DIMENSION       NELPX(3),NELLA(3),LNODS(36,9),COORD(153,3),
     +                KONTP(153),INDPL(153),INDLY(153,3),INDCS(153)
      PREC=1.E-15
      DO 30 JPNLY=1,NPNLY+1
      DO 30 IPNLX=1,NPNLX
      DO 20 NUMEL=1,NELPX(IPNLX)*NELLA(KZ)
      IELEM=IELEM+1
      DO 20 INODE=1,8
      LNODE=LNODS(IELEM,INODE)
      IF(INDPL(LNODE) .LT. 0) GO TO 20
      KONTP(LNODE)=KONTP(LNODE)+1
      IF(KONTP(LNODE) .GT. 1) GO TO 20
      ZCOOR=COORD(LNODE,3)
      IF(DABS(ZCOOR-ZBOT) .LT. PREC .OR. DABS(ZCOOR-ZTOP) .LT. PREC)
     + GO TO 10
      INDLY(LNODE,KZ)=-1
      GO TO 20
10    INDLY(LNODE,KZ)=1
      IF(KZ .EQ. 1   .AND. DABS(ZCOOR-ZBOT) .LT. PREC) INDCS(LNODE)=1
      IF(KZ.EQ.NLAYR .AND. DABS(ZCOOR-ZTOP) .LT. PREC) INDCS(LNODE)=1
20    CONTINUE
30    CONTINUE
      RETURN
      END
```