FINITE ELEMENT DYNAMIC SHAKEDOWN ANALYSIS OF A GRAVITY TYPE OFFSHORE STRUCTURE-FOUNDATION SYSTEM

VOLUME I

A. K. HALDAR

FINITE ELEMENT DYNAMIC SHAKEDOWN ANALYSIS OF A GRAVITY TYPE OFFSHORE STRUCTURE-FOUNDATION SYSTEM

by

(C) A. K. Haldar, B.E., M.Eng, P.Eng.

A thesis submitted to the School
of Graduate Studies in partial
fulfillment of the requirements
for the degree of Doctor
of Philosophy

VOLUME - I

Faculty of Engineering
Memorial University of Newfoundland
December 1984

St. John's Newfoundland

ABSTRACT

The shakedown deformation of a gravity type offshore foundation subjected to dynamic cyclic wave loading is determined based on the two dimensional finite element formulation of a fluid saturated porous medium. The soil is assumed to behave in an undrained manner and modelled as a two-phase material. The undrained characteristics of the soil leads the Poisson's ratio to be one-half and thus results in a singular stiffness matrix. The reduced integration technique is used in order to circumvent the problem. The dynamic shakedown load factor is obtained by first computing the elastodynamic response of the caisson-foundation (saturated) medium for some adjusted initial conditions (e.g. displacement and velocity) and then solving a linear programming problem based on the dual form of kinematic shakedown theory.

A finite element computer code OPFA (Offshore Platform Foundation Analysis) is developed which computes the dynamic shakedown load factor of a gravity type offshore foundation and then determines the response quantities such as displacements and stresses in the saturated soil medium. The nonlinear stress-strain characteristics of the soil medium is considered in the analysis by use of an equivalent linear and an elastic-perfectly plastic model.

Several problems are solved in order to check the accuracy of the computer programme. The results are in excellent agreement with the available analytical solutions. The shakedown analysis for a flexible foundation shows that limit pressures depend on the inclination angle and the eccentricity of the load. The shakedown limit pressure decreases as the eccentricity and the angle of inclination of the load increase. Also, shakedown analyses indicate that the footing pressures are below those predicted by the approximate bearing capacity formula. These shakedown pressures are obtained from an upper bound solution and therefore the use of the above formula will not produce a conservative estimate of the bearing capacity of the foundation when the loading is cyclic in nature.

The dynamic shakedown load factor for the gravity foundation is only 10% lower than its static counterpart. The reason for the decrease of this load factor is due to the amplification of stresses under dynamic loading condition. In computing the response quantities for three different soil models, it is observed that the equivalent linear analysis predicts lower deformation as opposed to other linear and nonlinear models. However, for the quasi-static analysis of the gravity foundation, the permanent vertical shakedown deformation is within 0.085% of the footing width. In all cases it is observed that shakedown occurs with respect to the deformation of the foundation.

ACKNOWLEDGEMENTS

The author is very grateful to his former supervisor, Dr. D. V. Reddy, Professor, Department of Ocean Engineering, Florida Atlantic University, for his excellent guidance, valuable suggestions, encouragement and continuous support with the provisions of reference material throughout the project. The author would also like to express his appreciation to Dr. M. Arockiasamy, his present supervisor, who has reviewed the manuscript critically and given various suggestions.

Special thanks are due to Dean Peters, Associate Dean Chari, and Dr. D. B. Muggeridge, Chairman, Ocean Research Group, Faculty of Engineering and Applied Science, Dean Aldrich, School of Graduate Studies and members of the thesis supervisory committee for their constant advice and encouragement.

Thanks are due to Mr. G. Bennett of the Computer Service Center at Memorial University for his valuable help during the programming period and Mrs. Janet Morris of Newfoundland and Labrador Hydro, who during many evening hours, carefully typed the draft and the final manuscript.

Finally, I am especially thankful to my wife Pratima and my two sons, Anirban and Anindam whose understandings and sacrifices made it possible for me to complete this dissertation.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Cont'd.)

Page

CHAPTER III - FINITE ELEMENT FORMULATION FOR FLUID SATURATED POROUS MEDIUM 29

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

Page

TABLE OF CONTENTS (Cont'd.)

LIST OF TABLES

LIST OF TABLES (Cont'd.)

LIST OF FIGURES

LIST OF FIGURES (Cont'd.)

LIST OF FIGURES (Cont'd.)

LIST OF FIGURES (Cont'd.)

LIST OF FIGURES (Cont'd.)

LIST OF FIGURES (Cont'd.)

LIST OF FIGURES (Cont'd.)

Figure Page

LIST OF SYMBOLS

ENGLISH LETTERS

t = time parameter

$\bar{x}$ = cartesian coordinate system (e.g. x and y)

p = excess pore pressure over hydrostatic

n = porosity

k = permeability

X_B and Y_B = body force of bulk solid in x and y directions

X_F and Y_F = body force of fluid in x and y directions

u and w = displacement of solid and fluid with respect to solid

K_w = bulk modulus of water

M_o = modified bulk modulus of water

$\hat{}$ = prescribed value

V = volume

S = total surface area

$\hat{T}(\bar{x},t)$ = prescribed traction

$\hat{p}(\bar{x},t)$ = prescribed fluid pressure

LIST OF SYMBOLS (Cont'd.)

l amd m = direction cosines in x and y

h_i = i-th interpolation function

r and s = local coordinate system

x and y = global coordinate system

$\underset{\approx}{U}_x$ and $\underset{\approx}{U}_y$ = nodal solid displacement vectors in x and y directions

$\underset{\approx}{W}_x$ and $\underset{\approx}{W}_y$ = nodal fluid displacement vectors in x and y directions

$\underline{U}$ and $\underline{W}$ = displacement vectors for solid and fluid at nodes

D_{ss} , D_{sf} and D_{ff} = stress-strain matrix for solid, solid-fluid coupling and fluid respectively

$\underset{\approx}{D}$ = matrix containing D_{ss}, D_{sf} and D_{ff}

M_{ss} , M_{sf} and M_{ff} = solid, solid-fluid and fluid mass matrices for an element

C_{ss} and C_{ff} = solid and fluid damping matrices for an element

K_{ss} , K_{sf} and K_{ff} = solid, solid-fluid and fluid stiffness matrices for an element

D_{sd} = drained soil skeleton constitutive matrix

p_B and p_f = bulk solid and fluid load vectors for an element

P_B = assembled global load vector

LIST OF SYMBOLS (Cont'd.)

K_c = modified bulk modulus

E_u and E_d = undrained and drained Young's moduli respectively

D^s and D^v = deviatoric and volumetric constitutive matrices

S^e = strain energy

K_1 and K_2 = drained stiffness matrix and stiffness matrix due to volumetric strain respectively

B_1 and B_2 = Rayleigh damping coefficients

s_u = undrained shear strength

f_y = yield function

$\hat{T}_i$ = prescribed surface traction

B_i = body force

u_i and $\dot{u}_i$ = displacement and velocity field

$Q(.)$ = plastic dissipation density function

N = total number of cycles

$\hat{C}_{kl}$ = coefficients of elasticity matrix

$F(\hat{T}_i)$ and $F_o(\hat{T}_i)$ = arbitrary loading and unit loading domain respectively

LIST OF SYMBOLS (Cont'd.)

$\bar{T}_s(\bar{x})$ = surface traction in s-th load mode

$\tilde{N}_i$ = stress gradient matrix containing unit normal vector for i-th yield plane

f_{yi} = linearized i-th yield plane

H = hardening parameter

K_{oi} = vector containing the distance of the i-th yield plane from the origin

$\tilde{N}_i^j$ = unit normal vector relatred to i-th yield plane of j-th element

R = total number of yield planes

V_j = subvolume for element, j

M = total number of elements

N^o = total number of degrees of freedom (D.O.F.)

W_{int} = internal work

$[G_o]$ = global compatibility matrix

W_{ext} = external work

$\underline{U}(\bar{x},t^*)$ and $\dot{\underline{U}}(\bar{x},t^*)$ = fictitious displacement and velocity

t^* = arbitrary fictitious time

LIST OF SYMBOLS (Cont'd.)

$U(\bar{x},0)$ and $\dot{U}(\bar{x},0)$ = actual initial conditions; displacement and velocity

$\tilde{M}_i^j$ = maximum stress vector of j-th element with respect to i-th yield plane

$\bar{U}^*$ and $\dot{\bar{U}}^*$ = displacement and velocity computed at a finite time

T_D = dead load

G = secant shear modulus

G_{max} = maximum shear modulus

D_{max} = maximum damping ratio

m_j, c_j, $\bar{k}_j$ = mass, damping and stiffness matrices for element, j

D_j = damping ratio for element, j

M^*, C^* and K^* = assembled mass, damping and stiffness matrices respectively

a and b = material parameters for soil

Y = hardening parameter

$[D_{sd}^{Ep}]$ = undrained elastic-plastic constitutive matrix

c' = cohesion

J_1 and J_2 = first and second stress invariant

S_x, S_y and S_z = deviatoric stress components

LIST OF SYMBOLS (Cont'd.)

$\bar{e}^p$ = effective plastic strain

$\tilde{R}$ = residual load vector

ΔU = incremental displacement vector

$\hat{K}_t^{(i)}$ = effective undrained stiffness matrix for time t, at a particular iteration, i

$\hat{P}_B$ = effective load vector

$\|\cdot\|$ = norm

RTOL = specified tolerance value

Δt = incremental time step

F_V = submerged or dead weight of the structure

q_{ult} = ultimate bearing pressure

N_c = bearing capacity factor = $(2+\Pi)$

e = eccentricity of the loading

B and A = footing width and area of footing

B' = effective width of the footing
= $B(1-2\,e/B)$

F_H and $\bar{F}_H$ = horizontal loading and prescribed value

Q' = inclined loading

a_0 = height of the caisson structure

LIST OF SYMBOLS (Cont'd.)

F_M and $\bar{\bar{F}}_M$ = overturning moment caused by the horizontal component of the wave loading and its prescribed value

t_s = stiffness ratio of the caisson structure to foundation soil

GREEK LETTERS

σ_x, σ_y and τ_{xy} = total stress components

ρ and ρ_f = solid and fluid weight densities

ε_x, ε_y and γ_{xy} = strain components

ξ = volumetric strain of fluid with respect to solid

α = grain compressibility

λ and μ = Lame's constants

λ_c = modified Lame constant

ϕ' and ψ' = arbitrary functions

δW = virtual work

σ and ε = stress and strain vectors

$\underset{\sim}{\phi}_m$ and $\underline{\phi}_m$ = matrix containing interpolation functions

$\underset{\sim}{0}$ = null vector

LIST OF SYMBOLS (Cont'd.)

$\underline{\underline{\psi}}$ = strain displacement matrix

ψ_{11} = solid strain-displacement matrix

ψ_{22} = fluid volumetric strain displacement matrix

$\delta\underline{U}$ and $\delta\underline{W}$ = virtual displacement of solid and fluid with respect to solid respectively

ε_v = volumetric strain

ν_u and ν_d = undrained and drained Poisson's ratios respectively

ϕ_u = angle of internal friction

σ^s_{ij} = safe soil stress tensor

σ^a_{ij} = admissible soil stress tensor

ε^p_{ij} and $\dot{\varepsilon}^p_{ij}$ = plastic strain tensor and its rate

σ_{ij} = elastic plastic stress

σ^E_{ij} = elastic stress tensor

σ^R_{ij} = residual stress tensor

τ = time interval

β = load factor

β_s = shakedown limit load

LIST OF SYMBOLS (Cont'd.)

β_E = elastic limit load

ω and ω^* = period and frequency ratio

$\Delta\varepsilon_{ij}$ = incremental strain tensor

$\Delta\varepsilon^E_{ij}$ and $\Delta\varepsilon^p_{ij}$ = incremental strain tensor; elastic and plastic respectively

γ_s = s-th load parameter

σ^{kE}_{ij} = elastic stress field corresponding to load parameter,k

σ^T = transpose of stress vector

Δ = plastic multiplier

σ^j and ε^j = stress and strain for j-th element

σ^E_{max} = maximum elastic stress response vector

σ^{*E}_{max} = fictitious maximum elastic stress response vector

τ_{max} = maximum shear stress

γ_{max} = maximum shear strain

σ_{ii} = hydrostatic stress

S_{ij} = deviatoric stress tensor

δ_{ij} = Kronecker delta

LIST OF SYMBOLS (Cont'd.)

σ'_{ij} and σ' = effective stress vectors

σ_y and $\bar{\sigma}$ = yield stress

Θ = parameter normally equals to 1.4

δ_H and δ_V = horizontal and vertical deformations of the footing

α_A = load inclination angle

$\underset{\sim}{\Delta}^j$ = plastic multiplier for an element, j

$\underset{\widetilde{=}}{\Delta}$ = plastic multiplier for all elements in assembled form

σ_v = normal vertical pressure

CHAPTER I

INTRODUCTION

1.0 Nature of the Problem

The continuous demand for oil and gas has challenged ocean engineers to undertake the design and construction of large gravity type offshore structures in hostile ocean environments. The gravity type offshore structure has an obvious advantage over the steel jacket type platform because of its shorter installation time and ability to provide the oil storage facilities in its cellular compartments. The large magnitude of wave forces which are transmitted by the structure to the surrounding soil cannot be compared with a similar land based structure. As the platforms are being installed in deeper water, the complexity of the problem has increased enormously. Conventional methods of analyses are no longer applicable and new techniques need to be developed for complete dynamic analysis of the structure-foundation system.

A typical offshore gravity structure will always encounter two situations: a) calm water condition and b) storm condition. During the passage of large waves, the net force on the structure will vary and reach its maximum value as the wave crest strikes the structure. Fig. 1.1 depicts the principal forces that will act upon the structure as the wave passes. It is seen from Fig. 1.1, that the structure is subjected to the cyclic nature of loading, which in turn will generate cyclic stresses in the foundation soil. The duration of this cyclic loading can be for several hours with a relatively low frequency range of about 0.05 - 0.5 Hz. The dynamic nature of the wave forces on the foundation causes great concern in the geotechnical design aspect and therefore requires the development of a reliable method for the analysis of offshore structure-foundation problems.

The soil behaviour beneath the foundation of an offshore structure subjected to dynamic cyclic loading may be separated into a short-term undrained condition with no drainage allowed and a long-term condition in which drainage occurs. In the undrained short term condition, the excess pore pressure leads to either a reduction or an increase of 'effective stress' with accompanying reduction or increase in soil stiffness and strength. For the platform this means an increase or decrease in the natural period of vibration. The magnitude of

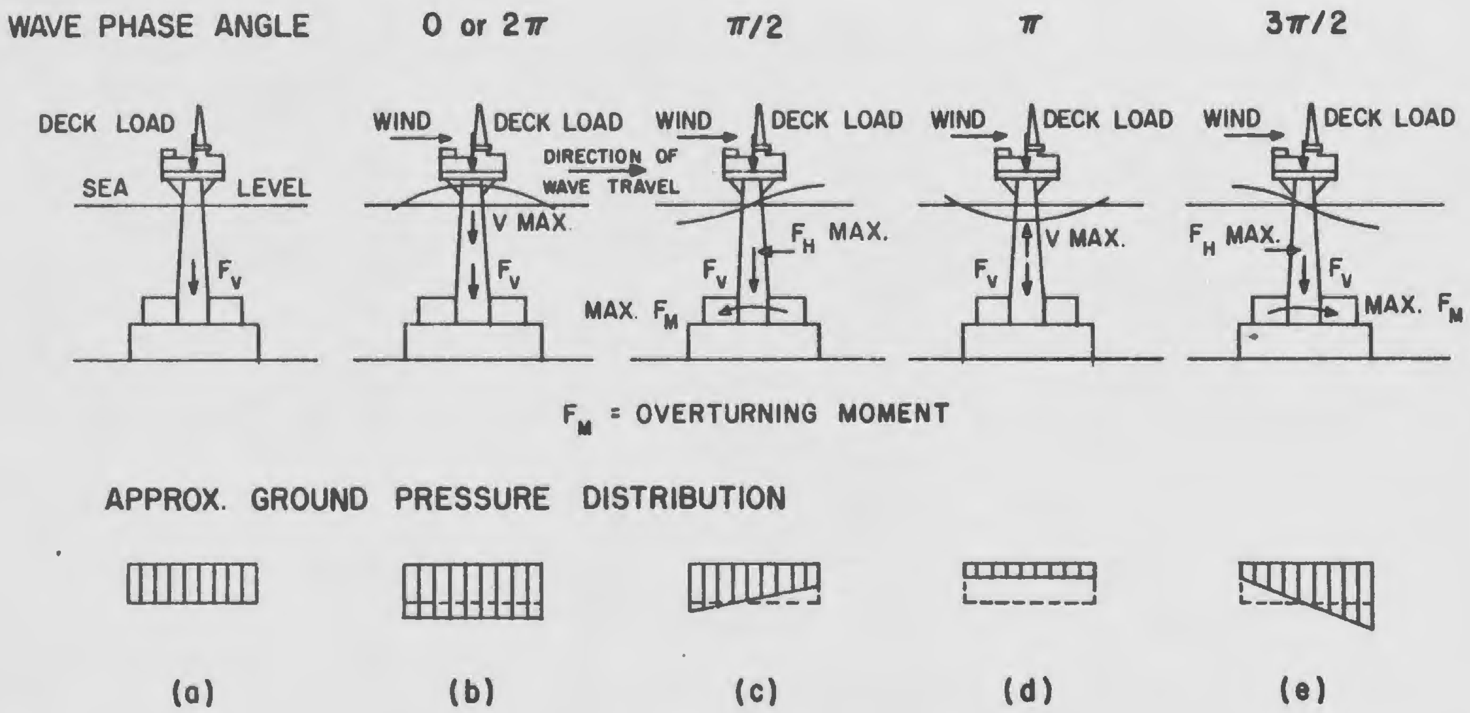


FIG. 1.1 CYCLIC LOADING ON THE FOUNDATION OF A GRAVITY TYPE OFFSHORE STRUCTURE DUE TO WAVE ACTION (STUBBS, 1976)

changes in 'effective stresses' due to the change in average pore pressure (increase/decrease) will depend on the rate of dissipation of pore water. It is seen from the above argument that the 'effective stress' is responsible for all major deformations, linear or nonlinear, and failure states can only be adequately modelled in terms of such 'effective streess'. It has been shown that the soil properties, in particular the stiffness, are among the most important factors influencing the dynamic behaviour of typical offshore gravity platforms.

For foundations subjected to loads varying in time in a nonproportional manner within prescribed limits, the classical limit theorems can give unsafe estimates of the collapse load, as failure can occur at loads well below the static collapse values. Shakedown theorems, which are generalizations of the limit theorems, can provide appropriate safe bounds for such complex loading programmes. For load reversal varying between prescribed limits, a foundation is said to shakedown when a state is reached such that all subsequent load applications produce only elastic changes. In recent years, very few attempts have been made to find practical solutions to the shakedown problem related to marine foundations resting on saturated soils. As the nonlinear dynamic response analysis of an offshore foundation to wave loading can only be carried out for a very limited number of cycles due to constraints on computing costs, the shakedown analysis becomes more practical and relevant to the overall design process.

Research in the field of earthquake engineering has shown that the soil deformation can vary from small to very large amplitude depending on the nature of the dynamic loading. Cyclic loading due to wave causes permanent deformation and thus strain under both drained and undrained conditions. The permanent strains are cumulative in nature i.e. carried over from one cycle to another and remain even if the cyclic loading has been terminated.

In predicting the permanent deformation of an offshore foundation system, one has to incorporate the appropriate soil model in a finite element analysis. Current research in soil stress-strain relationship is progressing towards the development of a three dimensional soil model based on the principles of plasticity as well as linear or nonlinear elasticity. Several sophisticated constitutive relationships have been proposed in recent years although none of them completely describe the behaviour of real soil.

1.2 Object of the Research

The complete dynamic response analysis of an offshore gravity foundation-soil system with the evaluation of the shakedown limit load which will further enable prediction of the permanent deformation of the foundation is beyond the present state-of-the-art.

The object of this research is to develop an analytical method for the prediction of the permanent deformation of an offshore foundation with particular reference to soil shakedown. A dynamic finite element formulation of the shakedown problem is presented for a fluid saturated porous medium.

1.3 Organization of the Thesis

Chaper 2 reviews the literatures related to the present investigation. These are described under three sections: 1) fluid saturated porous medium 2) constitutive modelling of soil and 3) shakedown analysis related to structural mechanics and geomechanics.

Chapter 3 presents the general dynamic finite element formulation of a saturated soil medium in which the soil is modelled as a two-phase material. The undrained characteristics of the soil lead the Poisson's ratio to be one-half and thus results in a singular stiffness matrix. The reduced integration technique is used in order to circumvent the problem. The two types of problems e.g. a strip footing and a cantilever beam are chosen as test cases in order to check the computer programme based on the present formulation.

Chapter 4 presents the shakedown formulation of the offshore foundation problem under cyclic loading conditions. Basic background on shakedown related theorems is presented in the first part whereas the finite element formulation of the dynamic shakedown problem is described in the second part. Typical boundary value problems are solved in order to validate the especially developed computer code.

Chapter 5 describes the various types of soil models. The general formulations for the equivalent linearized soil model and the incremental nonlinear elastic-plastic model are presented. Finally the chapter concludes with the computation of the shakedown displacement of a strip footing under repeated vertical loading condition.

Chapter 6 describes the series of engineering problems where the accumulation of permanent displacements are studied. Three types of analyses are carried out. The first analysis presents the elastic and shakedown limit loads of a flexible foundation and the effects of the load eccentricity and the angle of load inclination on these limit loads. The second analysis presents the permanent deformation of a gravity type offshore foundation considering the static soil shakedown. In the third analysis, the dynamic shakedown load factor is first determined and the responses are then computed for various types of soil models. In the second and third analyses, the effect of the caisson structure is taken into account by modelling it as a stiff footing.

Chapter 7 reviews the results of the research and recommends possible extensions of the present work.

Appendix A presents the proofs of the shakedown theorems. Appendix B summarizes the intermediate steps in deriving the elastic-plastic constitutive matrix based on the formulation presented in

Chapter 5. Appendices C and D describe the computer code OPFA (Offshore Platform Foundation Analysis) in a detailed manner together with the preparation of input data. Appendix E in Vol. II of this thesis presents the listing of the computer program 'OPFA'.

CHAPTER II

REVIEW OF LITERATURE

2.1 Introduction

This chapter reviews the literature related to the present investigation. For convenience, the reviews are divided in three sections. These are a) fluid saturated porous media b) constitutive modelling of soil medium and c) shakedown analysis with particular reference to structural mechanics and geomechanics.

2.2 Fluid-saturated Porous Media

A systematic development of the mechanics of deformation of fluid saturated (fully) porous solids was presented by Biot (1941, 1955, 1956a, 1956b, 1956c, 1962). The basic concept in the development of the theory was to treat the saturated soil as a two-phase media with a matrix of solid particles (the soil skeleton) and the water filling the space not occupied by the solids (pore fluid). Biot's theory was based on pure mechanical considerations. Later Green and Naghdi (1965) established the formal basis of the dynamic theory of mixtures from which Green and Steel (1966) formulated a theory of poroelasticity. Dereshewicz and Skalak (1963) have shown the uniqueness in dynamic poroelasticity using Neumann's uniqueness theorem of classical elasticity to liquid-saturated porous solids within the framework of Biot's theory. Recently Firoozbakhsh (1976) presented a unified treatment of the mechanics of deformation of a nondissipative liquid-filled saturated porous elastic medium based on Mindlin's work of microstructure in linear elasticity. Crochet and Naghdi (1966) developed the theoretical basis for the constitutive theory for flow of fluid through an elastic solid.

Since 1960, finite element approximations in conjunction with appropriate variational principles have shown great potential for the solution of the initial boundary value problems in different branches of engineering. Sandhu and Wilson (1969) developed a finite element scheme for theoretical field equations of fluid flow in saturated porous elastic medium based on Biot's theory using Gurtin's (1964) variational principle. The constitutive equation used for the solid phase was based on an elastic, isotropic, linear material behaviour. Later, Ghaboussi and Wilson (1972) extended the quasistatic formulation of Sandhu and Wilson (1969) to the dynamic problem.

Finite element formulation of the wave propagation problem using Biot's field equations for dynamical condition was carried out by Melian and Brebbia (1972). Biot's equations were reformulated following a Galerkin type formulation to discretize the governing differential equations of motion. Taga and Togashi (1975, 1977) studied the similar wave propagation problem based on the Hamilton's Principle. Applying the bimixture theory and the finite element method, free vibration and steady state responses were obtained. Ishihara (1970) furnished an approximate form of the wave equations based on Biot's theory and tried to correlate these wave equations with those normally obtained from the ordinary elastic theory.

A rigorous finite element formulation based on Gurtin's variational principle was made for the governing equations in Biot's consolidation theory by Yokoo, Yamagata and Nagoka (1971a, 1971b, 1971c). Their analyses included idealization of nonhomogeneous, anisotropic soil as a one, two or three dimensional consolidation problem.

2.3 Constitutive Modelling

Cyclic loadings due to waves induce permanent strains in the soil foundation system. The permanent strains often termed as cumulative or residual strains will remain even if the loading has been terminated. Accumulation of permanent strains in the soil due to repetitive nature of loading such as wave or earthquake will depend primarily on the soil type and density, permeability, overconsolidation ratio, initial stresses, and finally number of frequencies and magnitudes of the loading cycles.

The current state-of-the-art on computational methods for permanent displacements of offshore gravity platform can be categorized by the following two approaches.

In the first approach an incremental analysis is carried out for the entire loading cycles which involves consideraton of large number of time steps. Prevost (1977) has illustrated the use of a complex elasto-plastic soil model in order to predict the quasi-static response of an offshore foundation to wave loading. Prevost's model was based on the anisotropic theory of plasticity and has often been criticized for using large number of material parameters. Prevost introduced a set of nested yield surfaces and the associated plastic moduli to model the undrained characteristics of soil such as clay. Mroz, Norris and Zienkiewicz (1978) have also presented a soil model similar to Prevost's model based on two surface anisotropic plasticity theory. The basic concepts behind the Mroz's theory are: a) the consolidation history is bounded and defined by Roscoe-Burland (1968) surface b) yield surface defines an elastic domain within the bounding surface and c) the plastic modulus varies from a value H_0 near the yield surface to a value H_b on the bounding surface. The Mroz's model is often considered as extension of the work of Dafalias and Popov (1976) who introduced the concept of vanishing 'yield surface' in modelling the soil behavior based on plasticity theory. Finn and Martin (1980) evaluated the merits of the above two models e.g. Prevost's model and Mroz's model and concluded that anisotropic plasticity model predicts the static response of soil in a reasonable manner. However, the verification of the model to cyclic loading problem was very limited at the time of their study.

Van Eekelen (1980) stated that the anisotropic plasticity model of soil can only produce some features associated with cyclic loading and that the verification of the model with respect to laboratory soil tests is very limited. He illustrated that the Mroz's model (1978) will predict a pore pressure under undrained loading which is too small for normally consolidated clays and will show the incorrect sign for heavily overconsolidated soil.

It is seen from the above review, that there is no unified soil constitutive modelling techniques available at this time to study the effects of load reversal under two and three dimensional stress conditions.

The second approach for predicting the permanent deformation of foundation under repetitive nature of loading has been described by several authors e.g. Anderson et al (1978), Dumas and Lee (1980), Marr and Christian (1981).

Dumas and Lee (1980) presented a step by step approach for the computation of permanent deformation of foundation to a small parcel of uniform cycles of wave loading. The loading was treated as a linear static one and the following steps describe the procedure:

1. First the initial displacements, static stresses and cyclic stresses are computed.
2. From the laboratory test data, the strains are computed in each element based on the magnitude of stress level and number of cycles.
3. A pseudo-plastic modulus, E_p, is defined as the ratio between the cyclic stress, τ_e, to the strain, γ_e, computed in step 2.
4. Combined secant modulus, E_{ip}, is determined based on the initial modulus, E_i, and the plastic modulus, E_p, as $1/E_i + 1/E_p$.
5. Permanent displacements are then computed as the difference between the nodal displacements computed using secant modulus, E_{ip}, and the initial displacements based on tangent modulus, E_i.

Anderson (1978) first presented a graphical method based on the cyclic stress-strain data on saturated clay under undrained cyclic loading. The stress strain behaviour of a clay element subjected to two way cyclic loading in a simple shear apparatus under undrained condition is shown in Fig. 2.1. During the test, the cyclic shear stress, τ_{hc}, was kept constant (stress controlled test) at a

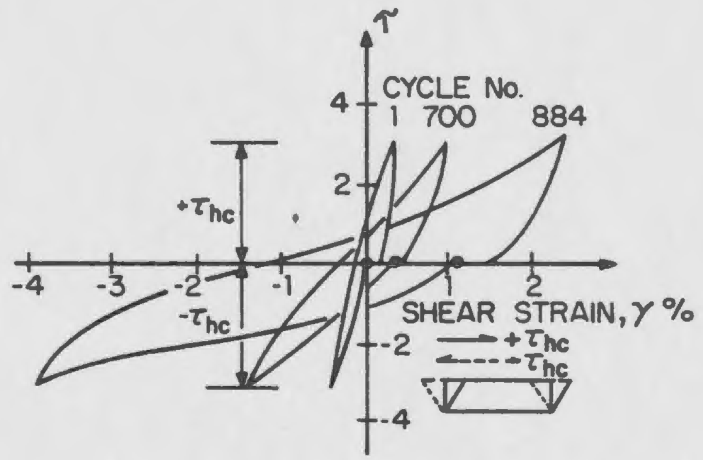


FIG. 2.1 STRESS-STRAIN CURVES FROM SIMPLE SHEAR TESTS WITH UNDRAINED TWO-WAY CYCLIC LOADING (ANDERSON, 1976)

level of ± 14.2 KN/m^2 which corresponds to 45% of the horizontal static shear stress, τ_{hf}, at failure under undrained condition. It can be seen from the Fig. 2.1, that the cyclic shear strain increases from +0.35% (0.0035) in the first cycle to +3.1% (.031) after 884 cycles. Based on these tests, a strain contour diagram, Fig. 2.2, is first constructed which shows the variation of cyclic shear strain with a given stress level and number of cycles. While using this method, waves are arranged in blocks, each representing wave loads of approximately equal amplitude. Each block of waves gives two contributions to the cyclic shear strain level. These are a) immediate cyclic strain level caused by the increase in stress level from one block to the next and b) increase in cyclic shear strain level due to the number of cycles at the particular stress level. The cylic shear strain at the end of a block is defined as the initial value for the next block and is expressed as

$$\gamma_{c,N+\Delta N} = \gamma_{c,N} + \Delta\gamma_{c,i} + \Delta\gamma_{c,N} \qquad \ldots (2.1)$$

where $\gamma_{c,N+\Delta N}$ = cyclic shear strain after, $N+\Delta N$ cycles (Fig. 2.3)

$\gamma_{c,N}$ = cyclic shear strain in cycle N with a cyclic shear stress, $\tau_{c,N}$

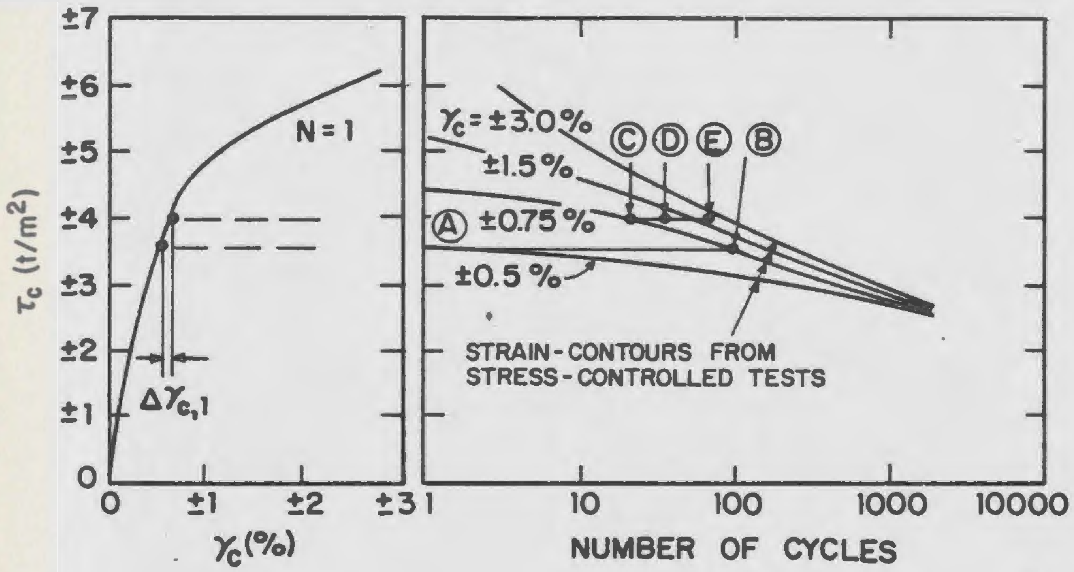


FIG. 2.2 PREDICTION OF CYCLIC SHEAR STRAINS FOR SOIL ELEMENT SUBJECTED TO VARYING CYCLIC SHEAR STRESSES. (ANDERSON, 1976)

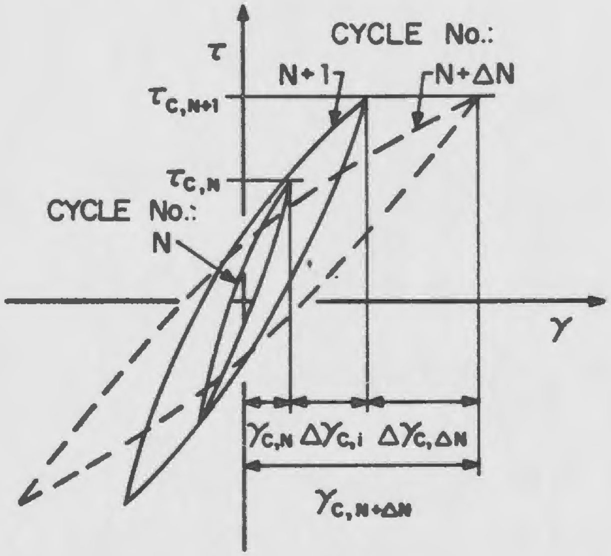


FIG. 2.3 STRESS-STRAIN BEHAVIOR OF SOIL ELEMENT SUBJECTED TO VARYING CYCLIC SHEAR STRESSES (ANDERSON,1976)

$\Delta\gamma_{c,i}$ = immediate change in cyclic shear strain due to change in cyclic shear stress level from $\tau_{c,N}$ to $\tau_{c,N+1}$

and $\Delta\gamma_{c,\Delta N}$ = increase in cyclic shear strain due to ΔN cycles with a cyclic shear stress, $\tau_{c,N+1}$

Marr and Christian (1981) described a method based on the concept of 'no net load'. The permanent strains for each element were evaluated during particular loading cycle and their effects represented by unbalanced equivalent nodal loads at the end of the cycle. Before the computation proceeds further, this unbalanced nodal load was redistributed resulting in additional deformations in the elements. The permanent deformations were then computed from the difference between the above two deformations. The constitutive relationship for soil was obtained based on a series of tests carried out on clay. They obtained a nearly log-linear relationship among cumulative strains, number of cycles, porosity, stress level, and cyclic shear stress.

2.4 Shakedown Analysis

2.4.1 Structural Mechanics

Bleich (1932) presented the static shakedown theorem with a limited proof. However, general proof of static shakedown theorem with the illustration of a truss problem was first presented by Melan (1936). Applications of static shakedown theory for continuous medium have been presented by Melan (1938), Symonds (1951) and Koiter (1960). Koiter (1960) also extended the work of Melan to the kinematic characterization of inadaptation (non shakedown) and presented the basic proof for kinematic shakedown theorem. The extension of Melan's static shakedown theorem to temperature load cycles was formulated by Prager (1956).

Considerable work has been done on the application of shakedown theory for discrete structures (frames, trusses, etc.). Symonds (1951) was the first to present the application of shakedown theorem to a continuum problem. A circular shaft subjected to combined axial force and twisting moment was analysed and the shakedown domain for the axial force and twisting moment obtained. Leckie and Penny (1969) carried out a shakedown analysis for a complex structure based on Melan's theorem. They evaluated the lower bound estimates of shakedown load factors for pressure, twist and moment applied to a radial flush of nozzle in a spherical pressure vessel. The residual stress field was approximated based on an available elastic solution and shakedown load factor finally evaluated using linear programming technique. Combining the loading and the temperature, Leckie and Penny (1969) showed how a shakedown analysis can help in making the important design decisions of a pressure vessel.

The matrix formulation of the shakedown problem was first presented by Maier (1969) with the use of piecewise linearization of the yield surface. Maier (1969) proposed that the shakedown load factor can be obtained as the optimal value of a linear programming problem. Melan's theorem was presented as a primal linear programming problem whereas Koiter's theorem was presented in its dual form. Koiter's theorem was also extended in order to allow for variable dislocations with associated flow laws. Maier (1969) also presented the shakedown theory based on nonassociated flow laws (non normality condition) and showed the procedures for obtaining the lower and upper bounds to the shakedown load factor.

Ceradini (1969, 1980) first presented the dynamic shakedown theory as an extension to Melan's and Koiter's formulations. Dynamic shakedown theory is applicable to the situation where repeated loadings are varying so rapidly that the inertia and damping forces cannot be neglected (e.g. seismic-structure or wave-structure-foundation interaction problem).

The first dynamic shakedown theorem states that shakedown will occur if the yield condition is satisfied at every point and time by the sum of a constant residual stress field and the linear elastodynamic response evaluated on the basis of some arbitrary initial condition. Corradi and Maier (1972) extended the Koiter's inadaptation theorem to the dynamic problem and stated this as a second dynamic shakedown theorem. Ho (1972) formulated the dynamic shakedown theory of elasto-plastic systems as an extension to Koiter's inadaptation theorem (second shakedown theorem). It was shown that shakedown design of a system under dynamic loading can be transformed into a quasi-static elastic counterpart with the proper application of dynamic amplification factor over its elastic response. Maier (1972) extended Ceradini's dynamic shakedown theorem to systems which obey general hardening rule, using the dual relationship between the static and kinematic shakedown theorems; Koiter's theorem was also extended to allow for work hardening. Maier (1972) also included the effect of geometric nonlinearity in the classical shakedown theorem of Melan and Koiter and presented a proof for the new second order shakedown theory. König (1969) developed the shakedown theory of plates based on Melan's theorem using the yield locus concept; shakedown load factors were evaluated for simply supported and clamped plates by using the Tresca yield condition.

Although the shakedown theory based on finite element formulation has been developed for a broad range of problems, e.g. work hardening, dynamic and geometric nonlinearity, very few numerical solutions applicable to a continuum problem do exist in the literature. Belytschko (1972) presented the first numerical solution for shakedown analysis in continuum problem under plane stress condition. The problem was formulated with equilibrated finite element modelling coupled with a nonlinear programming approach. The method was applied to analyze the shakedown of a square plate with a circular hole subjected to biaxial variable repeated loadings. The material was assumed to be elastic-perfecly plastic obeying Von Mises yield condition. Corradi and Zavelani (1974) used the approach of Maier, based on piecewise linearization of the yield surface coupled with compatible finite element displacement model, and analyzed the same problem as that solved by Belytschko (1972).

The same thin plate, plane stress problem was also analysed by Hung and Palgen (1979) based on the theory as formulated by Hung and König (1976). The formulation was a static one with the yield criterion defined as the 'yield criterion of the mean'. Borkowski and Kleiber (1980) presented the numerical shakedown analyses for a hyperstatic truss and for a cantilever beam under plane stress conditions. Kleiber, König, and Sawczuk (1982) presented an excellent report on studies related to plastic structures. Special topics which were covered in the above presentation were: stability, anisotropy, hardening and cyclic loads. Several example problems were illustrated showing the results regarding the computer analyses of non-linear structural mechanics problems.

2.4.2 Geomechanics

Very few applications of shakedown criteria for soil mechanics problems are available at present contrary to the structural mechanics problem as presented earlier.

Rowe (1975) carried out experimental work on model studies of an offshore gravity platform foundation resting on saturated clays. The foundation was subjected to inclined, eccentric, and cyclic quasi-static loads to simulate the wave loading on an actual platform. The 'amount of shakedown' was considered as a function of the size of the wave force and the number of wave cycles. The load factor was defined as the value corresponding to 'equilibrium shakedown' of the vertical displacements of the foundation.

Zienkiewicz (1976, 1978) presented the effect of wave action on the foundation of an offshore structure using finite element analysis. The primary conclusion was that even such a relatively slow cyclic loading may lead to 'progressive deformation' (incremental collapse) at load level well below the static collapse value. In the earlier publications of Zienkiewicz (1976, 1978), 'ratchetting' (progressive collapse) was defined as the shakedown phenomenon. In fact, this is contrary to the actual definition given by Bleich (1932), Melan (1936), Neal (1956) and later by Maier (1969).

Aboustitt and Reddy (1980) developed a computer program for the application of the kinematic static shakedown theorem applicable to dry soil (drained) situation. The application was illustrated for a footing subjected to variable repeated loading under a plane strain condition. As there was no available solution for shakedown analysis in plane strain, the software developed was checked with the only available solution for plane stress: a square plate with a central circular hole, subjected to biaxial variable repeated loading. By making use of the fact that the shakedown theorems are generalizations of the limit theorems for a complex loading programme, the formulation was checked with available solutions of plane strain limit analysis. The footing analysis showed that the shakedown load varies almost linearly with the uniaxial compressive strength. At the same time Pande (1980) presented approximate solution of shakedown load for a strip footing under quasi-static loading situation. The traditional method of incremental nonlinear (elasto-plastic) analysis was used to obtain the lower bound on shakedown load factor. The soil was assumed to be purely cohesive and to obey the Mohr-Coulomb yield criterion. It was identified that shakedown analysis of offshore foundation problem will be of great interest to the geotechnical engineering field even though the clear procedures for carrying out such an analysis were lacking.

CHAPTER III

FINITE ELEMENT FORMULATION FOR FLUID SATURATED POROUS MEDIUM

3.1 Introduction

The general dynamic finite element formulation of saturated soil medium is described. The soil is modelled as a two-phase material whose state is described by the stresses and displacements in each phase. The undrained behaviour of fully saturated soil is accompanied by the condition of 'no volume' change and therefore the soil deforms like an incompressible material. The traditional displacement based finite element formulation does not apply when a material becomes truly incompressible. The incompressibility leads the Poisson's ratio to be one-half and thus results in a singular stiffness matrix. Reduced integration technique is used in order to circumvent the singularity problem.

The chapter is divided in three sections. The first section describes the general equations of motion of a fluid saturated porous medium based on Biot's approach. The second section presents the displacement based finite element formulation of the dynamic equations of motion using reduced integration technique. Two types of loading, e.g. undrained and drained states of loading, are discussed with respect to soil foundation interaction. A computer code for the elastodynamic analysis is developed and its correctness checked by analysing two problems: a) a strip footing under,vertical loading and b) a cantilever beam under periodic vertical loading.

3.2 Deformation Characteristics of Saturated Soil

The deformation analysis of saturated soil is normally carried out based on the assumption of the soil as a two-phase medium. The grains of the soil skeleton constitue the solid phase while the pore fluid filling the voids makes up the fluid phase. Normally, it is assumed that the soil skeleton behaves linearly or nonlinearly whereas the pore fluid is incompressible. The stresses in the soil skeleton are conventionally termed as "effective stresses" and those in the fluid (water) phase called the pore pressure.

During the loading process the two-phase characteristics of the saturated soil causes it to behave in two widely different ways. When the loading is applied in an instantaneous manner, excess pore pressure develops and the pore fluid then tends to flow due to the difference in the pressure gradient within the medium. The flow then continues for a finite time depending on the permeability of the soil and the drainage boundary condition. At the instant of the loading the pore fluid has little time to dissipate and because the pore fluid is incompressible relative to the soil skeleton, the soil deforms as an incompressible material. The deformation of soil with 'zero volume change' is often defined as undrained deformation. When the soil is loaded with free drainage condition,such that excess pore pressures dissipate, the deformation is defined to be drained. However, in intermediate state where the flow can take place, interaction between the skeleton and the flow of the pore water must be considered. Terzaghi first observed the interaction in 1925, and he assumed that the water filling the pore volume behaves as an incompressible liquid and the deformation of the skeleton is elastic. The individual grain is considered as incompressible. Darcy's law is applicable due to the pore fluid flow being laminar. Terzaghi's one dimensional consolidation theory, however, is not applicable for most practical problems where the soil is loaded in multi directions. Biot (1941) presented an alternative approach based on multidimensional theory. Biot's theory is used in the present study for the finite element formulation of the equations of motion for soil deformation analysis under dynamic loading. The basic assumptions in Biot's theory are:

1. Soil is modelled as a two-phase material.
2. The solid skeleton behaves in a linearly elastic form.
3. The fluid (water) is compressible and of the Newtonian type.
4. The individual grain constituting the soil skeleton also deforms elastically.
5. The total stress can be separated into two parts; a) effective stress and b) pore pressure.
6. Volume change due to shear deformation is not considered here.

3.3 Equations of Motion for a Saturated Porous Medium

The general equations of motion for fluid saturated porous medium were first presented by Biot and the equations are for i) bulk fluid-solid mixture,

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \rho b_x = \rho \ddot{u}_x + \rho_f \ddot{w}_x \qquad \ldots (3.1)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \rho b_y = \rho \ddot{u}_y + \rho_f \ddot{w}_y \qquad \ldots (3.2)$$

and ii) for the fluid with respect to the solid,

$$\frac{\partial p}{\partial x} + \rho_f b_x = \rho_f \ddot{u}_x + \frac{1}{n} \rho_f \ddot{w}_x + \frac{1}{k} \dot{w}_x \qquad \ldots (3.3)$$

$$\frac{\partial p}{\partial y} + \rho_f b_y = \rho_f \ddot{u}_y + \frac{1}{n} \rho_f \ddot{w}_y + \frac{1}{k} \dot{w}_y \qquad \ldots (3.4)$$

where σ_x, σ_y and τ_{xy} = total stresses of bulk solid in x and y directions

$X_B = \rho b_x$ = body force of bulk solid in x direction

$Y_B = \rho b_y$ = body force of bulk solid in y direction

u_x, $\dot{u}_x$ and $\ddot{u}_x$ = displacement, velocity and acceleration of bulk solid in x direction

u_y, $\dot{u}_y$ and $\ddot{u}_y$ = displacement, velocity and acceleration of bulk solid in y direction

w_x, $\dot{w}_x$ and $\ddot{w}_x$ = displacement, velocity and acceleration of fluid relative to solid in x direction

w_y, $\dot{w}_y$ and $\ddot{w}_y$ = displacement, velocity and acceleration of fluid relative to solid in y direction

p = excess pore pressure over hydrostatic

$X_F = \rho_f b_x$ = fluid body force in x direction

$Y_F = \rho_f b_y$ = fluid body force in y direction

n = porosity

and k = permeability

3.4 Strain-displacement Relationship

The strain displacement relationships for the solid and the fluid are:

$$\varepsilon_x = \frac{\partial u_x}{\partial x} ; \qquad \ldots (3.5a)$$

$$\varepsilon_y = \frac{\partial u_y}{\partial y} ; \qquad \ldots (3.5b)$$

and
$$\gamma_{xy} = \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) ; \qquad \ldots (3.5c)$$

The volumetric strain of fluid with respect to solid is given as

ξ = volumetric strain of fluid with respect to solid

$$= \frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y} \qquad \ldots (3.5d)$$

where ε_x, ε_y and γ_{xy} = solid strain components in x and y directions

and ξ = volumetric strain of fluid with respect to solid

3.5 Constitutive Relationships

Stress strain relationship for saturated porous media is expressed in terms of

$$\sigma_x = (\lambda_c + 2\mu)\frac{\partial u_x}{\partial x} + \lambda_c \frac{\partial u_y}{\partial y} + \alpha M_o \xi \qquad ..(3\cdot6a)$$

$$\sigma_y = \lambda_c \frac{\partial u_x}{\partial x} + (\lambda_c + 2\mu)\frac{\partial u_y}{\partial y} + \alpha M_o \xi \qquad ..(3\cdot6b)$$

$$\tau_{xy} = \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \qquad ..(3\cdot6c)$$

$$p = \alpha M_o \frac{\partial u_x}{\partial x} + \alpha M_o \frac{\partial u_y}{\partial y} + M_o \xi \qquad ..(3\cdot6d)$$

The above Eqns. (3.6a - 3.6d) can be written as

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ p \end{bmatrix} = \begin{bmatrix} (\lambda_c + 2\mu) & \lambda_c & 0 & \alpha M_o \\ \lambda_c & (\lambda_c + 2\mu) & 0 & \alpha M_o \\ 0 & 0 & \mu & 0 \\ \alpha M_o & \alpha M_o & 0 & M_o \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \xi \end{bmatrix}$$

$$= \begin{bmatrix} \lambda + 2\mu & \lambda & 0 & \alpha M_o \\ \lambda & \lambda + 2\mu & 0 & \alpha M \\ 0 & 0 & \mu & 0 \\ \alpha M_o & \alpha M_o & 0 & M_o \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \xi \end{bmatrix} + \begin{bmatrix} \alpha^2 M_o & \alpha^2 M_o & 0 & 0 \\ \alpha^2 M_o & \alpha^2 M_o & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \xi \end{bmatrix}$$

$$..(3.6e)$$

where λ_c = modified Lame's constant

$$= \lambda + \alpha^2 M_o$$

λ and μ = the Lame's constants defining the properties of soil skeleton under drained condition

α = compressibility of grains ($n \leq \alpha \leq 1$)

$$M_o = \frac{K_w}{n}$$

where K_w = bulk modulus of water

3.6 Boundary Conditions

The boundary conditions are given as (Fig. 3.1)

$$\left.\begin{aligned} u_x(x,t) &= \hat{u}_x(x,t) \\ u_y(y,t) &= \hat{u}_y(y,t) \\ w_x(x,t) &= \hat{w}_x(x,t) \\ w_y(y,t) &= \hat{w}_y(y,t) \end{aligned}\right\} \quad \text{For displacements on 'S}_1\text{'} \qquad \ldots(3.7a)$$

$$\left.\begin{aligned} \sigma_x \ell + \tau_{xy} m &= \hat{T}_x(x,t) \\ \tau_{xy} \ell + \sigma_y m &= \hat{T}_y(y,t) \\ p\ell &= \hat{p}_x(x,t) \\ pm &= \hat{p}_y(y,t) \end{aligned}\right\} \quad \text{For forces on 'S}_2 \qquad \ldots(3.7b)$$

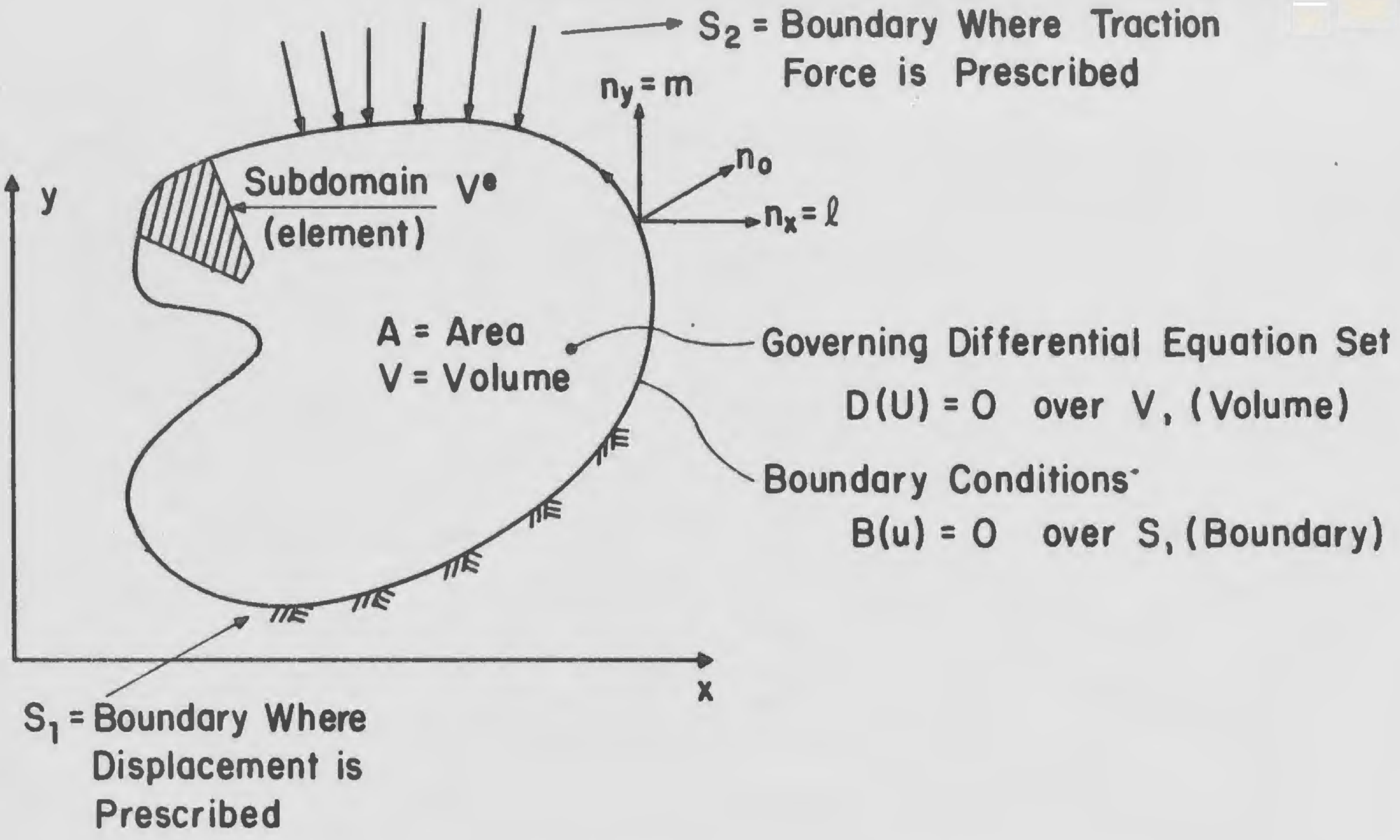


FIG. 3.1 GENERAL BOUNDARY VALUE PROBLEM

where $\hat{u}_x$, $\hat{u}_y$ = prescribed displacement - for bulk solid on S_1

$\hat{w}_x$, $\hat{w}_y$ = prescribed displacement for fluid with respect to solid on S_1

$\hat{T}_x(x,t)$ & $\hat{T}_y(y,t)$ = prescribed traction force on S_2 in x and y direction.

$\hat{p}_x(x,t)$ & $\hat{p}_y(y,t)$ = pressure on S_2 in x and y direction.

ℓ and m are the direction cosines in x and y directions.

Applying the principle of virtual work to equations of motion following Haldar (1978) and Haldar and Reddy (1979)

$$\begin{aligned}
\delta W = & \int_V \left\{ \left(\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \rho b_x - \rho \ddot{u}_x - \rho_f \ddot{w}_x \right) \delta u_x \right\} dV + \\
& \int_V \left\{ \left(\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \rho b_y - \rho \ddot{u}_y - \rho_f \ddot{w}_y \right) \delta u_y \right\} dV + \\
& \int_V \left\{ \left(\frac{\partial p}{\partial x} + \rho_f b_x - \rho_f \ddot{u}_x - \frac{1}{n} \rho_f \ddot{w}_x - \frac{1}{k} \dot{w}_x \right) \delta w_x \right\} dV + \\
& \int_V \left\{ \left(\frac{\partial p}{\partial y} + \rho_f b_y - \rho_f \ddot{u}_y - \frac{1}{n} \rho_f \ddot{w}_y - \frac{1}{k} \dot{w}_y \right) \delta w_y \right\} dV = 0
\end{aligned}$$

... (3.8)

Each term of Eqn. (3.8) is integrated using Green's theorem as follows:

$$\iiint_V \phi' \frac{\partial \Psi'}{\partial x} dV = - \iiint_V \frac{\partial \phi'}{\partial x} \Psi' dV + \int_S \phi' \Psi' \ell dS \qquad \ldots(3.9)$$

Where ϕ', and Ψ' are two functions and ℓ is the direction cosine in x-direction (Fig. 3.1).

Setting $\phi' = \delta u_x$ and $\Psi' = \sigma_x$ in the first integral of Eqn. (3.8)

$$\int_V \frac{\partial \sigma_x}{\partial x} \delta u_x \; dV = \int_V \phi' \frac{\partial \Psi'}{\partial x} dV$$

$$= - \int_V \frac{\partial \phi'}{\partial x} \Psi' dV + \int_S \phi' \Psi' \ell dS \qquad \ldots (3.10)$$

$$= - \int_V \frac{\partial}{\partial x} (\delta u_x) \; \sigma_x \; dV + \int_S \sigma_x \; (\ell \delta u_x) \; dS$$

$\ldots$ (3.11a)

Similarly the remaining terms are integrated by parts as

$$\int_V \frac{\partial \tau_{xy}}{\partial y} \delta u_x \; dV = - \int_V \tau_{xy} \frac{\partial}{\partial y} (\delta u_x) \; dV + \int_S \tau_{xy} \; (m \delta u_x) \; dS$$

$\ldots$ (3.11b)

Similarly, for the first term of the second integral in Eqn. (3.8)

$$\int_V \frac{\partial \tau_{xy}}{\partial x} \delta u_y \, dV = - \int_V \tau_{xy} \frac{\partial}{\partial x} (\delta u_y) \, dV + \int_S \tau_{xy} (\ell \delta u_y) \, dS \quad \ldots (3.11c)$$

$$\int_V \frac{\partial \sigma_y}{\partial y} \delta u_y \, dV = - \int_V \sigma_y \frac{\partial}{\partial y} (\delta u_y) \, dV + \int_S \sigma_y (m \delta u_y) \, dS \quad \ldots (3.11d)$$

Also, the first two terms in third and fourth integral of Eqn. (3.8) become

$$\int_V \frac{\partial p}{\partial x} \delta w_x \, dV = - \int_V p \frac{\partial}{\partial x} (\delta w_x) \, dV + \int_S p (\ell \delta w_x) \, dS \quad \ldots (3.11e)$$

and

$$\int_V \frac{\partial p}{\partial y} \delta w_y \, dV = - \int_V p \frac{\partial}{\partial y} (\delta w_y) \, dV + \int_S p (m \delta w_y) \, dS \quad \ldots (3.11f)$$

Eqn. (3.8) is rewritten in a new form using the above Eqns. (3.11a - 3.11f) as

$$\delta W = \{-\int_V \sigma_x \frac{\partial}{\partial x} (\delta u_x) \, dV + \int_S \sigma_x (\ell \delta u_x) \, dS\} + \{- \int_V \tau_{xy} \frac{\partial}{\partial y}$$

$$(\delta u_x) \, dV + \int_S \tau_{xy} (m \delta u_x) \, dS\} + \int_V \{(\rho b_x - \rho \ddot{u}_x - \rho_f \ddot{w}_x) \delta u_x\} \, dV$$

$$+ \{-\int_V \tau_{xy} \frac{\partial}{\partial x} (\delta u_y) \, dV + \int_S \tau_{xy} (\ell \delta u_y) \, dS\} + \{-\int_V \sigma_y$$

$$\frac{\partial}{\partial y} (\delta u_y) \, dV + \int_S \sigma_y (m \delta u_y) \, dS\} + \int_V \{(\rho b_y - \rho \ddot{u}_y - \rho_f \ddot{w}_y)$$

$$\delta u_y\} \, dV + \{-\int_V p \frac{\partial}{\partial x} (\delta w_x) \, dV + \int_S p \, (\ell \delta w_x) \, dS\}$$

$$+ \int_V \{\rho_f b_x - \rho_f \ddot{u}_x - \frac{1}{n} \rho_f \ddot{w}_x - \frac{1}{k} \dot{w}_x\} \, \delta w_x \, dV$$

$$+ \{- \int_V p \frac{\partial}{\partial y} (\delta w_y) \, dV + \int_S p \, (m \delta w_y) \, dS\}$$

$$+ \{\int_V (\rho_f b_y - \rho_f \ddot{u}_y - \frac{1}{n} \rho_f \ddot{w}_y - \frac{1}{k} \dot{w}_y) \, \delta w_y\} \, dV \qquad \ldots (3.12)$$

Using the relations expressed in Eqn. (3.5), gives the following relationships

$$\sigma_x \frac{\partial}{\partial x} (\delta u_x) = \sigma_x \delta \, (\frac{\partial u_x}{\partial x}) = \sigma_x \, \delta \varepsilon_x \qquad \ldots (3 \cdot 13a)$$

Similarly,

$$\sigma_y \frac{\partial}{\partial y} (\delta u_y) = \sigma_y \delta \, (\frac{\partial u_y}{\partial y}) = \sigma_y \, \delta \varepsilon_y \qquad \ldots (3 \cdot 13b)$$

and

$$P \frac{\partial}{\partial x} (\delta w_x) + p \frac{\partial}{\partial y} (\delta w_y) = p \, \delta (\frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y}) = p \delta \xi \qquad \ldots (3.13c)$$

Finally, the following relationship is obtained after substituting Eqn. (3.13) in Eqn. (3.12)

$$\delta W = - \int_V (\sigma_x \delta \varepsilon_x + \sigma_y \delta \varepsilon_y + \tau_{xy} \delta \gamma_{xy} + p \delta \xi) \, dV + \int_{S2} \{(\sigma_x \ell$$

$$+ \tau_{xy} m) \delta u_x \, dS - \int_V \{(\rho \ddot{u}_y + \rho_f \ddot{w}_y) \, \delta u_y\} \, dV + \int_{S2} \{(\tau_{xy} \ell$$

$$+ \sigma_y m) \, \delta u_y\} \, dS + \int_V (\rho b_x \delta u_x) \, dV + \int_V (\rho b_y \delta u_y) \, dV - \int_V dV \, \delta u_x$$

$$\{(\rho \ddot{u}_x + \rho_f \ddot{w}_x) + \int_{S2} (p \ell \delta w_x + p m \delta w_y) dS + \int_V (\rho_f b_x \; \delta w_x) dV$$

$$+ \int_V (\rho_f b_y \; \delta w_y) \, dV - \int_V \{(\rho_f \ddot{u}_x + \frac{\rho_f}{n} \ddot{w}_x) \; \delta w_x\} \, dV - \int_V \frac{1}{k}$$

$$(\dot{w}_x \delta w_x) \, dV - \int_V \{\rho_f \ddot{u}_y + \frac{1}{n} \rho_f \ddot{w}_y\} \delta u_y \, dV - \int_V \frac{1}{k} \, (\dot{w}_y \; \delta w_y) \, dV$$

... (3.14)

$$= - \int_V \langle \delta\varepsilon_x \; \delta\varepsilon_y \; \delta\gamma_{xy} \; \delta\xi \rangle \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ p \end{Bmatrix} dV + \int_{S2} \langle \delta u_x \; \delta u_y \rangle \begin{Bmatrix} \hat{T}_x \\ \hat{T}_y \end{Bmatrix} dS$$

$$+ \int_V \langle \delta u_x \; \delta u_y \rangle \begin{Bmatrix} X_B \\ Y_B \end{Bmatrix} dV - \int_V \langle \delta u_x \; \delta u_y \rangle \begin{Bmatrix} \rho \ddot{u}_x \\ \rho \ddot{u}_y \end{Bmatrix} dV - \int_V \langle \delta u_x \; \delta u_y \rangle$$

$$\begin{Bmatrix} \rho_f \ddot{w}_x \\ \rho_f \ddot{w}_y \end{Bmatrix} dV + \int_{S2} \langle \delta w_x \; \delta w_y \rangle \begin{Bmatrix} \hat{p}_x \\ \hat{p}_y \end{Bmatrix} dS - \int_V \langle \delta w_x \; \delta w_y \rangle \begin{Bmatrix} \rho_f \ddot{u} \\ \rho_f \ddot{u}_y \end{Bmatrix} dV$$

$$- \int_V \langle \delta w_x \; \delta w_y \rangle \begin{Bmatrix} \frac{\rho_f}{n} \; \ddot{w}_x \\ \frac{\rho_f}{n} \; \ddot{w}_y \end{Bmatrix} dV - \int_V \langle \delta w_x \; \delta w_y \rangle \begin{Bmatrix} \frac{1}{k} \; \dot{w}_x \\ \frac{1}{k} \; \dot{w}_y \end{Bmatrix} dV$$

$$+ \int_V \langle \delta w_x \; \delta w_y \rangle \begin{Bmatrix} \rho_f b_x \\ \rho_f b_y \end{Bmatrix} dV$$

... (3.15)

$$= - \int_V \delta\varepsilon^T \; \sigma \; dV + \int_{S2} \delta u^T \; \{\hat{T}\} \; dS + \int_V \delta u^T \begin{Bmatrix} X_B \\ Y_B \end{Bmatrix} dV$$

$$- \int_V \rho \delta u^T \begin{Bmatrix} \ddot{u}_x \\ \ddot{u}_y \end{Bmatrix} dV - \int_V \rho_f \; \delta u^T \begin{Bmatrix} \ddot{w}_x \\ \ddot{w}_y \end{Bmatrix} dV + \int_{S2} \delta w^T \; \{\hat{p}\} \; dS$$

$$-\int_V \delta w^T \, dV \, \rho_f \begin{Bmatrix} \ddot{u}_x \\ \ddot{u}_y \end{Bmatrix} - \int_V \frac{\rho_f}{n} \, \delta w^T \begin{Bmatrix} \ddot{w}_x \\ \ddot{w}_y \end{Bmatrix} dV - \int_V \frac{1}{k} \, \delta w^T \begin{Bmatrix} \dot{w}_x \\ \dot{w}_y \end{Bmatrix} dV$$

$$-\int_V \delta w^T \begin{Bmatrix} X_F \\ Y_F \end{Bmatrix} dV \qquad \ldots (3.16)$$

where $\delta \varepsilon^T = < \delta \varepsilon_x \; \delta \varepsilon_y \; \delta \gamma_{xy} \; \delta \xi >$

$$\sigma^T = < \sigma_x \; \sigma_y \; \tau_{xy} \; p >$$

and T denotes the transpose

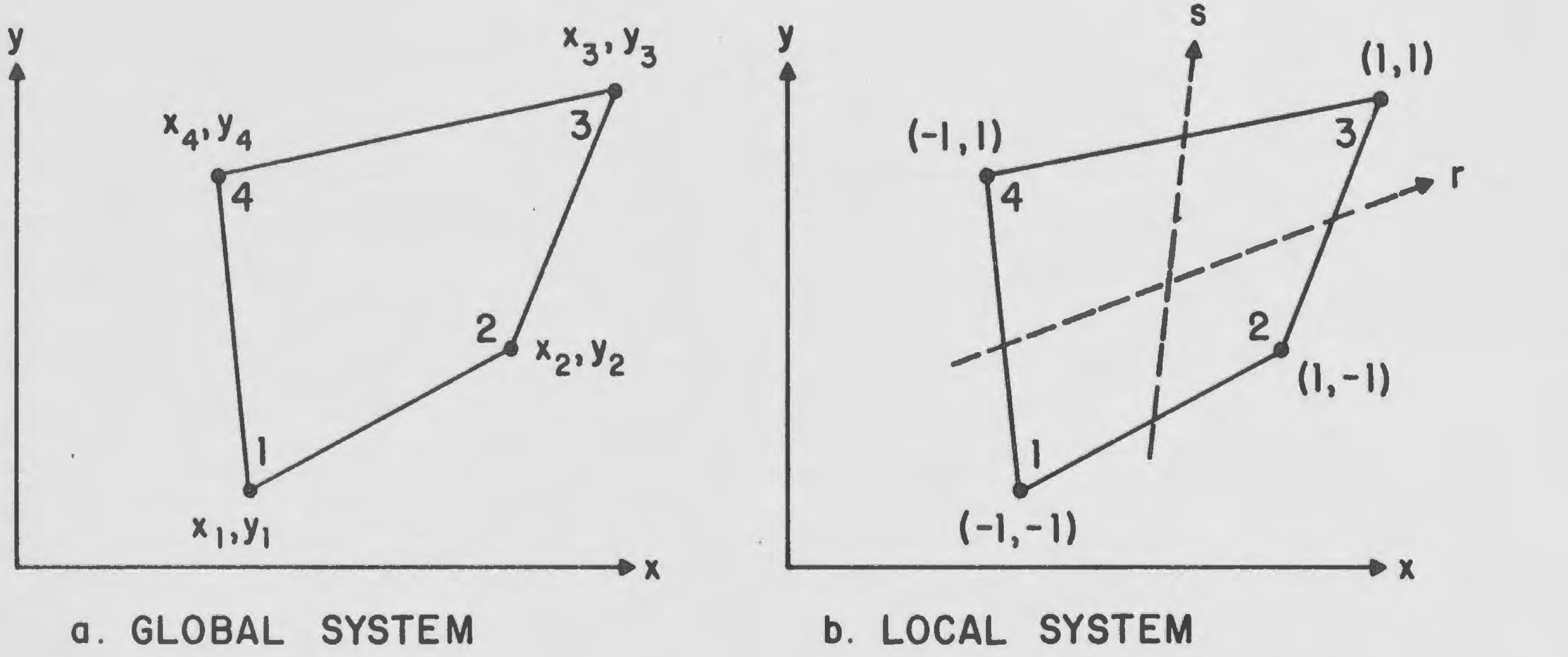
Here, S2, represents that part of the total surface, S, for which stresses are prescribed, and not displacements (Fig. 3.1)

3.7 Two Dimensional Finite Element Formulation

A four noded isoparametric element shown in Fig. 3.2 is used in which a linear displacement field is assumed for the solid and the fluid. The continuum is discretized using four noded elements. The soil is assumed to be homogeneous within each element.

For a four noded isoparametric element, the coordinate interpolations are assumed following Haldar and Reddy (1979) as

$$x = \sum_{i=1}^{4} h_i x_i \qquad \ldots (3.17a)$$



a. GLOBAL SYSTEM b. LOCAL SYSTEM

FIG. 3.2 TWO-DIMENSIONAL ISOPARAMETRIC PLANE STRAIN ELEMENT

$$y = \sum_{i=1}^{4} h_i y_i \quad .. (3.17b)$$

The displacement interpolation functions for the solid and fluid are

$$u_x = \sum_{i=1}^{4} h_i U_{xi} \quad .. (3.18a)$$

$$u_y = \sum_{i=1}^{4} h_i U_{yi} \quad .. (3.18b)$$

and

$$w_x = \sum_{i=1}^{4} h_i W_{xi} \quad .. (3.18c)$$

$$w_y = \sum_{i=1}^{4} h_i W_{yi} \quad .. (3.18d)$$

where

$$\begin{aligned} h_1 &= \tfrac{1}{4} (1 + r)(1 + s) \\ h_2 &= \tfrac{1}{4} (1 - r)(1 + s) \\ h_3 &= \tfrac{1}{4} (1 - r)(1 - s) \\ h_4 &= \tfrac{1}{4} (1 + r)(1 - s) \end{aligned} \quad .. (3.19)$$

Here, r, s, and x,y represent the local and global coordinate systems respectively (Fig. 3.2). The solid displacements, within an element are expressed in terms of nodal displacements vectors $\underset{\approx}{U}_x = (U_{x1}, U_{x2}, U_{x3}, U_{x4})$ and $\underset{\approx}{U}_y = (U_{y1}, U_{y2}, U_{y3}$ and $U_{y4})$ as

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & h_4 \end{bmatrix} \begin{bmatrix} \underset{\approx}{U}_x \\ \underset{\approx}{U}_y \end{bmatrix}$$

$$= \begin{bmatrix} \underset{\approx}{\phi}_m & \underset{\approx}{0} \\ \underset{\approx}{0} & \underset{\approx}{\phi}_m \end{bmatrix} \begin{bmatrix} \underset{\approx}{U}_x \\ \underset{\approx}{U}_y \end{bmatrix} \qquad .. \ (3.20a)$$

or $u = \underline{\phi}_m \underline{U}$,

where u is solid displacement vector at any point within the element

and $\underline{U}$ is displacement vector at nodal points

Similarly, the fluid displacement with respect to solid within an element can also be rewritten in terms of fluid nodal displacement vectors as

$$\begin{bmatrix} w_x \\ w_y \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & h_4 \end{bmatrix} \begin{bmatrix} \underset{\approx}{W}_x \\ \underset{\approx}{W}_y \end{bmatrix}$$

$$= \begin{bmatrix} \underset{\approx}{\phi}_m & \underset{\approx}{0} \\ \underset{\approx}{0} & \underset{\approx}{\phi}_m \end{bmatrix} \begin{bmatrix} \underset{\approx}{W}_x \\ \underset{\approx}{W}_y \end{bmatrix} \qquad .. \ (3.20b)$$

or, $w = \underline{\phi}_m \underline{W}$

where w is fluid displacement vector at any point within the element

and $\underline{W}$ is displacement vector at nodal points

From Eqns. (3.20a) and (3.20b) the following relationships are obtained:

$$u^T = \underline{U}^T \underline{\phi}_m^T \quad \text{and} \quad w^T = \underline{W}^T \underline{\phi}_m^T \qquad .. \ (3.21a)$$

$$\ddot{u} = \underline{\phi}_m \ddot{\underline{U}} \quad \text{and} \quad \ddot{w} = \underline{\phi}_m \ddot{\underline{W}} \qquad .. \ (3.21b)$$

$$\dot{u} = \underline{\phi}_m \dot{\underline{U}} \quad \text{and} \quad \dot{w} = \underline{\phi}_m \dot{\underline{W}} \qquad .. \ (3.21c)$$

where $\dot{u}$, $\ddot{u}$ = velocity and acceleration vectors for solid

and $\dot{w}$, $\ddot{w}$ = velocity and acceleration vectors for fluid with respect to solid.

The strain displacement relationships given by Eqn. (3.5) are established as follows in terms of the derivatives of the interpolation function as

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \xi \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & 0 \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ w_x \\ w_y \end{bmatrix} \qquad .. \ (3.22)$$

After substituting Eqns. (3.20a and 3.20b), Eqn. (3.22) becomes

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \xi \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & 0 \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \underset{\sim}{\phi}_m & \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\phi}_m & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\phi}_m & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\phi}_m \end{bmatrix} \begin{bmatrix} \underset{\sim}{U}_x \\ \underset{\sim}{U}_y \\ \underset{\sim}{W}_x \\ \underset{\sim}{W}_y \end{bmatrix} \qquad .. \ (3.23)$$

$$= \begin{bmatrix} \frac{\partial}{\partial x}(\underset{\sim}{\phi}_m) & \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \frac{\partial}{\partial y}(\underset{\sim}{\phi}_m) & \underset{\sim}{0} & \underset{\sim}{0} \\ \frac{\partial}{\partial y}(\underset{\sim}{\phi}_m) & \frac{\partial}{\partial x}(\underset{\sim}{\phi}_m) & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \frac{\partial}{\partial x}(\underset{\sim}{\phi}_m) & \frac{\partial}{\partial y}(\underset{\sim}{\phi}_m) \end{bmatrix} \begin{bmatrix} \underset{\sim}{U}_x \\ \underset{\sim}{U}_y \\ \underset{\sim}{W}_x \\ \underset{\sim}{W}_y \end{bmatrix} \qquad .. \ (3.24)$$

$$= \begin{bmatrix} \psi_{11} & \psi_{12} \\ \psi_{21} & \psi_{22} \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{W} \end{bmatrix} = \left[\underset{\sim}{\psi}\right] \begin{bmatrix} \underline{U} \\ \underline{W} \end{bmatrix} \qquad .. \ (3.25)$$

where

$$\psi_{11} = \begin{bmatrix} \frac{\partial}{\partial x}\,(\underset{=}{\phi}_m) & \underset{\sim}{0} \\ \underset{\sim}{0} & \frac{\partial}{\partial y}\,(\underset{=}{\phi}_m) \\ \frac{\partial}{\partial y}\,(\underset{=}{\phi}_m) & \frac{\partial}{\partial x}\,(\underset{=}{\phi}_m) \end{bmatrix} \quad \text{.. (3. 25a)}$$

and

$$\underline{U} = \begin{bmatrix} \underset{\sim}{U}_x \\ \underset{\sim}{U}_y \end{bmatrix}, \quad \underline{W} = \begin{bmatrix} \underset{\sim}{W}_x \\ \underset{\sim}{W}_y \end{bmatrix} \quad \text{.. (3.25b)}$$

$$\psi_{12} = \begin{bmatrix} \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} \end{bmatrix} \quad \text{.. (3.25c)}$$

$$\psi_{21} = \begin{bmatrix} \underset{\sim}{0} & \underset{\sim}{0} \end{bmatrix} \quad \text{.. (3.25d)}$$

$$\psi_{22} = \begin{bmatrix} \frac{\partial}{\partial x}\,(\underset{=}{\phi}_m) & \frac{\partial}{\partial y}\,(\underset{=}{\phi}_m) \end{bmatrix} \quad \text{.. (3.25e)}$$

The stress-strain relation can be rewritten from Eqns. (3.6e) in the following form

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ p \end{bmatrix} = \begin{bmatrix} \lambda_c + 2\mu & \lambda_c & 0 & \alpha M_o \\ \lambda_c & \lambda_c + 2\mu & 0 & \alpha M_o \\ 0 & 0 & \mu & \alpha M_o \\ \alpha M_o & \alpha M_o & 0 & M_o \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \xi \end{bmatrix}$$

or,

$$\sigma = \begin{bmatrix} \sigma_s \\ \\ p \end{bmatrix} = \begin{bmatrix} D_{ss} & D_{sf} \\ \\ D_{fs} & D_{ff} \end{bmatrix} \begin{bmatrix} \varepsilon_s \\ \\ \xi \end{bmatrix} \qquad \ldots (3.26)$$

where D_{ss} = solid stress-strain matrix (undrained)

$D_{sf} = D_{fs}$= solid-fluid coupling stress-strain matrix

D_{ff} = fluid stress-strain matrix

and ${\sigma_s}^T = <\sigma_x \; \sigma_y \; \tau_{xy}>$

${\varepsilon_s}^T = <\varepsilon_x \; \varepsilon_y \; \gamma_{xy}>$

Substitution of Eqn. (3.25) in stress-strain relation of Eqn. (3.26) furnishes

$$\sigma = \begin{bmatrix} D_{ss} & D_{sf} \\ \\ D_{fs} & D_{ff} \end{bmatrix} \begin{bmatrix} \psi_{11} & \psi_{12} \\ \\ \psi_{21} & \psi_{22} \end{bmatrix} \begin{bmatrix} \underline{U} \\ \\ \underline{W} \end{bmatrix} \qquad \ldots (3.27)$$

or

$$\sigma = \left[\underset{\sim}{\underline{D}}\right] \left[\underset{\sim}{\underline{\psi}}\right] \begin{bmatrix} \underline{U} \\ \underline{W} \end{bmatrix}$$

$$= \begin{bmatrix} D_{ss}\psi_{11} & (D_{ss}\psi_{12} + D_{sf}\psi_{22}) \\ \\ D_{fs}\psi_{11} & D_{ff}\psi_{22} \end{bmatrix} \begin{bmatrix} \underline{U} \\ \\ \underline{W} \end{bmatrix} \qquad \ldots (3.28)$$

where $D_{ss}\psi_{12} = 0$

From Eqn. (3.25) the strain displacement relation, is obtained in partitioned form as

$$<\varepsilon_x \ \varepsilon_y \ \gamma_{xy}> = \underline{U}^T \ \psi_{11}^T$$

$$\xi = \underline{W}^T \psi_{22}{}^T \qquad .. \ (3.29)$$

or,

$$<\delta\varepsilon_x \ \delta\varepsilon_y \ \delta\gamma_{xy}> = \delta\underline{U}^T \ \psi_{11}^T \qquad .. \ (3.30)$$

and

$$\delta\xi = \delta\underline{W}^T \ \psi_{22}{}^T \qquad .. \ (3.31)$$

Rewriting Eqn. (3.27) in the following form

$$\sigma = \begin{bmatrix} D_{ss} & D_{sf} \\ D_{fs} & D_{ff} \end{bmatrix} \begin{bmatrix} \psi_{11} \underline{U} \\ \psi_{22} \underline{W} \end{bmatrix} = \begin{bmatrix} D_{ss}\psi_{11}\underline{U} + D_{sf}\psi_{22}\underline{W} \\ D_{fs}\psi_{11}\underline{U} + D_{ff}\psi_{22}\underline{W} \end{bmatrix} \qquad .. \ (3.32)$$

and substitution of the above relations in Eqn. (3.16) gives

$$\delta W = - \int_V \delta\underline{U}^T \ \psi_{11}^T \ (D_{ss}\psi_{11}\underline{U} + D_{sf}\psi_{22}\underline{W}) \ dV - \int_V \delta\underline{W}^T \ \psi_{22}^T$$

$$(D_{fs}\psi_{11}\underline{U} + D_{ff}\psi_{22}\underline{W}) \ dV + \int_{S2} \delta\underline{U}^T \ \underline{\phi}_m^T \ \{\hat{T}\} \ dS$$

$$+ \int_V \delta\underline{U}^T \ \underline{\phi}_m^T \begin{Bmatrix} X_B \\ Y_B \end{Bmatrix} dV - \int_V \rho\delta\underline{U}^T \ \underline{\phi}_m^T \ \underline{\phi}_m \ \ddot{\underline{U}} \ dV$$

$$- \int_{V} \rho_f \, \delta \underline{U}^T \, \underline{\phi}_m^T \, \underline{\phi}_m \ddot{\underline{W}} \, dV + \int_{S2} \delta \underline{W}^T \, \underline{\phi}_m^T \, \{\hat{p}\} \, dS$$

$$- \int_{V} \delta \underline{W}^T \, \rho_f \, \underline{\phi}_m^T \, \underline{\phi}_m \, \ddot{\underline{U}} \, dV - \int_{V} \delta \underline{W}^T \, \underline{\phi}_m^T \, \frac{\rho_f}{n} \, \underline{\phi}_m \, \ddot{\underline{W}} \, dV$$

$$- \int_{V} \delta \underline{W}^T \, \underline{\phi}_m^T \, \frac{1}{k} \, \underline{\phi}_m \, \dot{\underline{W}} \, dV + \int_{V} \delta \underline{W}^T \, \underline{\phi}_m^T \begin{Bmatrix} X_F \\ Y_F \end{Bmatrix} dV \qquad .. \; (3.33)$$

As $\delta W = 0$ and $\delta \underline{U}^T$ and $\delta \underline{W}^T$ are arbirary variations in displacements for solids and fluids with respect to solids, two sets of equations, which are independently zero, are obtained as:

$$\Big(\int_{V} \underline{\phi}_m^T \, \rho \underline{\phi}_m \, \ddot{\underline{U}} + \int_{V} \underline{\phi}_m^T \, \rho_f \, \underline{\phi}_m \, \ddot{\underline{W}} + \int_{V} \psi_{11}^T D_{ss} \, \psi_{11} \, \underline{U} + \int_{V} \psi_{11}^T D_{sf}$$

$$\psi_{22} \, \underline{W} \Big) \, dV = \int_{V} \underline{\phi}_m^T \begin{Bmatrix} X_B \\ Y_B \end{Bmatrix} dV + \int_{S2} \underline{\phi}_m^T \, \{\hat{T}\} \, dS$$

$$.. \; (3.34a)$$

and

$$\Big(\int_{V} \underline{\phi}_m^T \, \rho_f \, \underline{\phi}_m \, \ddot{\underline{U}} + \int_{V} \underline{\phi}_m^T \, \frac{\rho_f}{n} \, \underline{\phi}_m \, \ddot{\underline{W}} + \int_{V} \underline{\phi}_m^T \, \frac{1}{k} \, \underline{\phi}_m \, \dot{\underline{W}} + \int_{V}$$

$$\psi_{22}^T D_{sf} \, \psi_{11} \, \underline{U} + \int_{V} \psi_m^T \, D_{ff} \, \psi_{22} \, \underline{W} \Big) \, dV = \int_{V} \underline{\phi}_m^T \begin{Bmatrix} X_F \\ Y_F \end{Bmatrix} dV$$

$$+ \int_{S2} \underline{\phi}_m^T \, \{\hat{p}\} \, dS \qquad .. \; (3.34b)$$

or

$$\begin{bmatrix} M_{ss} & M_{sf} \\ M_{fs} & M_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\underline{U}} \\ \ddot{\underline{W}} \end{bmatrix} + \begin{bmatrix} \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & C_{ff} \end{bmatrix} \begin{bmatrix} \dot{\underline{U}} \\ \dot{\underline{W}} \end{bmatrix} + \begin{bmatrix} K_{ss} & K_{sf} \\ K_{fs} & K_{ff} \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{W} \end{bmatrix} = \begin{bmatrix} p_B \\ p_f \end{bmatrix} \quad .. \ (3.35)$$

The structural damping which accounts for the energy dissipation of solid skeleton, may be represented in terms of Rayleigh damping coefficients (B_1 and B_2) as a combination of mass and stiffness matrices and expressed in the following form as

$$C_{ss} = B_1 \ (\ M_{ss} \ - n^2 \ M_{ff} \ ) + B_2 \ (\ K_{ss} \ - \alpha^2 \ K_{ff} \ ) \quad .. \ (3.36)$$

Substitution of Eqn. (3.36) in (3.35) gives

$$\begin{bmatrix} M_{ss} & M_{sf} \\ M_{fs} & M_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\underline{U}} \\ \ddot{\underline{W}} \end{bmatrix} + \begin{bmatrix} C_{ss} & \underset{\sim}{0} \\ \underset{\sim}{0} & C_{ff} \end{bmatrix} \begin{bmatrix} \dot{\underline{U}} \\ \dot{\underline{W}} \end{bmatrix} + \begin{bmatrix} K_{ss} & K_{sf} \\ K_{fs} & K_{ff} \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{W} \end{bmatrix} = \begin{bmatrix} p_B \\ p_f \end{bmatrix} \quad .. \ (3.37)$$

where

M_{ss} = solid consistent mass matrix

$$= \int_V \underset{\sim}{\phi}_m^T \ \rho \ \underset{\sim}{\phi}_m \ dV, \quad .. \ (3.38a)$$

M_{sf} = coupled solid fluid matrix

$$= \int_V \underline{\phi}_m^T \rho_f \underline{\phi}_m \, dV \, , \qquad .. \, (3.38b)$$

M_{ff} = fluid consistent mass matrix

$$= \int_V \underline{\phi}_m^T \frac{\rho_f}{n} \underline{\phi}_m \, dV \, , \qquad .. \, (3.38c)$$

C_{ff} = fluid damping matrix

$$= \int_V \underline{\phi}_m^T \frac{1}{k} \underline{\phi}_m \, dV \, , \qquad .. \, (3.38d)$$

K_{ss} = solid stiffness matrix (undrained)

$$= \int_V \psi_{11}^T \, D_{ss} \psi_{11} \, dV \, , \qquad .. \, (3.38e)$$

K_{sf} = K_{fs} = solid fluid coupled stiffness matrix

$$= \int_V \psi_{11}^T D_{sf} \psi_{22} = \int_V \psi_{22}^T D_{fs} \psi_{11} dV \qquad .. \, (3.38f)$$

K_{ff} = fluid stiffness matrix

$$= \int_V \psi_{22}^T \, D_{ff} \psi_{22} \, dV \, , \qquad .. \, (3.38g)$$

Using Eqn. (3.6e), the term, D_{ss}, in Eqn. (3.38e) can further be rewritten as

$$K_{ss} = \int_V \psi_{11}^T \left[D_{sd} + D_{ff}\right] \psi_{11} \, dV \qquad \ldots (3.38h)$$

where D_{sd} = drained elasticity matrix for the soil skeleton

$$= \begin{bmatrix} \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda+2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

and D_{ff} = elasticity matrix for the pore fluid

$$= \begin{bmatrix} \alpha^2 M_o & \alpha^2 M_o & 0 \\ \alpha^2 M_o & \alpha^2 M_o & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \ldots (3.38i)$$

Substituting Eqn. (3.25a) in Eqn. (3.38h) one can obtain

$$K_{ss} = \int_V \psi_{11}^T D_{sd} \psi_{11} \, dV + \int_V \alpha^2 M_o \psi_{22}^T \psi_{22} \, dV \qquad \ldots (3.38j)$$

where $\psi_{22} = [h_{1,x}, h_{2,x}, \ldots h_{3,y}, h_{4,y}]$

and $h_{i,x}$ = differentiation of the i-th interpolaton function with respect to x-coordinate

The load vectors in Eqn. (3.37) are expressed as

$$p_B = \text{bulk load vector including gravity}$$

$$= \int_V \underline{\phi}_m^T \begin{Bmatrix} X_B \\ Y_B \end{Bmatrix} dV + \int_{S2} \underline{\phi}_m^T \{\hat{T}\}\, dS \quad \ldots (3.38l)$$

and

$$p_f = \text{fluid load vector including gravity}$$

$$= \int_V \underline{\phi}_m^T \begin{Bmatrix} X_F \\ Y_F \end{Bmatrix} dV + \int_{S2} \underline{\phi}_m^T \{\hat{p}\}\, dS \quad \ldots (3,38m)$$

3.8 Condition for Incompressibility

The redistribution of the stresses takes place just after the loading is applied to the soil medium. The stresses here are termed as total and effective and are accompanied by strains and displacements. The pore water pressure will also develop and the magnitude and distribution of the pore water pressure depends not only on the load but also on the soil type and its past stress history. If the loading is applied in an instantaneous manner the soil behaves under undrained condition, since the pore water has little time to dissipate. The condition of undrained state for fully saturated soil is accompanied by the condition of 'no volume' change and as such may be expressed in terms of strain components as

$$\varepsilon_x + \varepsilon_y + \varepsilon_z = 0 \qquad \ldots (3.39)$$

The strain components are

$$\varepsilon_x = \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E} - \nu \frac{\sigma_z}{E} \qquad \ldots (3.40a)$$

$$\varepsilon_y = \frac{\sigma_y}{E} - \nu \frac{\sigma_x}{E} - \nu \frac{\sigma_z}{E} \qquad \ldots (3.40b)$$

$$\varepsilon_z = \frac{\sigma_z}{E} - \nu \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E} \qquad \ldots (3.40c)$$

and $\gamma_{xy} = \frac{\tau_{xy}}{\mu}$ where ν = Poisson's ratio

Substitution of Eqns. (3.40a-c) into Eqn. (3.39) yields

$$(1-2\nu) \frac{\sigma_x}{E} + (1-2\nu) \frac{\sigma_y}{E} + (1-2\nu) \frac{\sigma_z}{E} = 0 \qquad \ldots (3.41)$$

Since σ_x, σ_y and σ_z are not equal to zero and hence $(1-2\nu) = 0$ which results in $\nu = 0.50$

The conventional displacement based finite element formulation cannot be applied to the solution of undrained deformation of soil because of the difficulty in expressing the stresses solely in terms of the strains. Even when the finite element procedure is applied to nearly incompressible solid $(\nu \simeq 0.5)$ the solution will lead to very large errors.

However, the elasticity matrix, $\left[\underset{\approx}{D}\right]$, in Eqn. (3.28) can be separated in the following form for undrained condition since $\xi = 0$ from Eqn. (3.5d) indicating no flow of fluid with respect to solid.

$$\left[\underset{\approx}{D}\right] = (K_c - \tfrac{2}{3}\mu)\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \mu \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \ldots (3.42)$$

where the shear modulus μ and the bulk modulus K_c represent the elasticity constants in terms of total stress, σ_{ij}, and are defined as

$$\mu = \frac{E_u}{2\ (1 + \nu_u)} \doteq \frac{E_d}{2\ (1 + \nu_d)} \quad \ldots (3.43a)$$

and $$K_c = \frac{E_u}{3\ (1 - 2\nu_u)} \quad \ldots (3.43b)$$

in which E_u and E_d = undrained and drained Young's moduli respectively.

ν_u and ν_d = undrained and drained Poisson's ratios respectively.

Eqn. (3.42) can be rewritten as

$$[\underset{\approx}{D}] = \mu \ [D]^{s} + 2\Theta \ [D]^{v} \qquad \ldots (3.44)$$

where $[D]^{s}$ = constitutive matrix related to deviatoric component of stress-strain part

and $[D]^{v}$ = constitutive matrix related to volumetric component of stress-strain part

As $\nu_{\mu} \rightarrow 0.5$, μ remains finite when the term $2\Theta \equiv (K_{c}-\frac{2}{3}\mu)\rightarrow\infty$

The strain energy, S^{e}, can then be expressed as

$$S^{e} = \frac{1}{2} \int_{V} \varepsilon^{T} \ [D]^{s} \ \mu \ \varepsilon \ dV + \frac{1}{2} \int_{V} \varepsilon^{T} \ [D]^{v} \ 2\Theta \ \varepsilon \ dV \qquad \ldots (3.45)$$

The volumetric strain, ε_{v}, is expressed as

$$\varepsilon_{v} = \varepsilon_{x} + \varepsilon_{y} = \frac{\partial u_{x}}{\partial x} + \frac{\partial u_{y}}{\partial y} \qquad \ldots (3.46)$$

$$= \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_{x} \\ \varepsilon_{y} \\ \gamma_{xy} \end{bmatrix} \qquad \ldots (3.47a)$$

and

$$S^{e} = S_{s} + S_{v} = \frac{1}{2} \int_{V} \varepsilon^{T} \ [D]^{s} \ \mu \ \varepsilon \ dV + \int_{V} \Theta \ \varepsilon_{v}^{2} \ dV \qquad \ldots (3.47b)$$

Therefore, the term S_s represents the distortional energy whereas, the term S_v, represents the volumetric strain energy. The incompressibility condition is satisfied when $\Theta \rightarrow \infty$.

In the undrained condition, the fluid has no movement with respect to the solid ie. $\underline{W} = 0$ and Eqn. (3.37) reduces to

$$\left[M_{ss}\right]\left\{\ddot{\underline{U}}\right\} + \left[C_{ss}\right]\left\{\dot{\underline{U}}\right\} + \left[K_{ss}\right]\left\{\underline{U}\right\} = \left\{P_B\right\} \quad .. (3.48)$$

for the dynamical condition. For the quasistatic condition for undrained loading $\ddot{\underline{U}} = \dot{\underline{U}} = 0$ and Eqn. (3.48) after assembling reduces to

$$\left[K^*\right]\left\{\underline{U}\right\} = \left\{P_B\right\} \quad .. (3.49)$$

and can be expressed in an explicit form as $\left[\text{Eqn. (3.38j)}\right]$

$$\sum^{M}\left(\int_V \psi_{11}^T D_{sd} \psi_{11} dV + \int_V \alpha^2 M_o \psi_{22}^T \psi_{22} dV\right)\left\{\underline{U}\right\} = \left\{P_B\right\} \quad .. (3.50)$$

where P_B = assembled load vector. We write above as

$$(K_1 + \alpha^2 M_o K_2)\left\{\underline{U}\right\} = \left\{P_B\right\} \quad .. (3.51)$$

where K_1 and $\alpha^2 M_o K_2$ = assembled stiffness matrix for solid phase under drained condition and that due to the effect of the degree of incompressibility of the pore fluid respectively.

The second term in Eqn. (3.51) represents the stiffness related to the volumetric strain, ε_v. For $\varepsilon_v = 0$, $\alpha^2 M_o$ should tend to infinity under undrained condition. Eqn. (3.51) is solved with finite values for the matrices K_1 and K_2.

It is clear that as $\alpha^2 M_o K_2$ tends to be very large when compared to K_1 ,Eqn. (3.51) becomes

$$\left(\alpha^2 M_o K_2\right)\left\{\underline{U}\right\} = \left\{P_B\right\} \qquad .. \text{ (3.52)}$$

and

$$\left[K_2\right]\left\{\underline{U}\right\} = \frac{1}{\alpha^2 M_o} \left\{P_B\right\} \rightarrow 0$$

Thus the incompressibility constraint takes a dominant role and will lead to a trivial solution if the matrix K_2 is non-singular;

Eqn. (3.52) shows the difficulty of obtaining results for the incompressible case with the displacement based finite element formulation, where the value of the Poisson's ratio tends to one-half (0.50) but is not equal to one-half. In order to circumvent this problem, the matrix $[K_2]$ is made singular and therefore

$$\{\underline{U}\} \neq 0$$

whereas

$$[K_2]\{\underline{U}\} = 0$$

The singularily in the $[K_2]$ matrix is introduced by use of lower order integration (Zienkiewiez, 1977) and therefore the reduced integration technique is particularly recommended for nearly incompressible problems.

The assembled undrained stiffness matrix, $[K^*]$, in global form should be non-singular, although the matrix, $[K_2]$ is singular. For linear quadrilateral element, the 2x2 integration is carried out for the matrix, $[K_1]$ and one point integration at centroid is performed for the matrix $[K_2]$. Therefore, in forming the matrix $[K_1]$ three independent relations (ε_x, ε_y, γ_{xy}) are introduced at each Gauss point but for the matrix $[K_2]$ only one such relation (volumetric strain, ε_v) is required. The condition of non-singularily of the assembled matrix, from Eqn. (3.49) is guaranteed if the number of such relations introduced at all the integrating points is greater than that of the available degrees of freedom.

3.9 Checking of Computer Programme

Two types of problems are studied in this section based on the formulation presented before. These are: a) deformation analysis of a strip footing under static vertical loading and b) dynamic response analysis of a cantilever beam under periodic loading. In both analyses, the results obtained are compared with the published values to ensure the accuracy and reliability of the computer programme.

3.9.1 Deformation Analysis of a Strip Footing

The application of the finite element formulation as described before is illustrated by the case of a strip footing underlaid by a shallow stratum of clay under undrained condition. The ratio of the half-width of the footing to the depth of the stratum is 0.50. The loading is assumed to act vertically downwards with a constant intensity and the soil medium is assumed to be homogeneous and isotropic. The finite element mesh of the strip footing is shown in Fig. (3.3). It consists of 56 nodes and 42 four noded rectangular isoparametric elements. Vertical side boundary is placed at a distance four times the width of the footing from

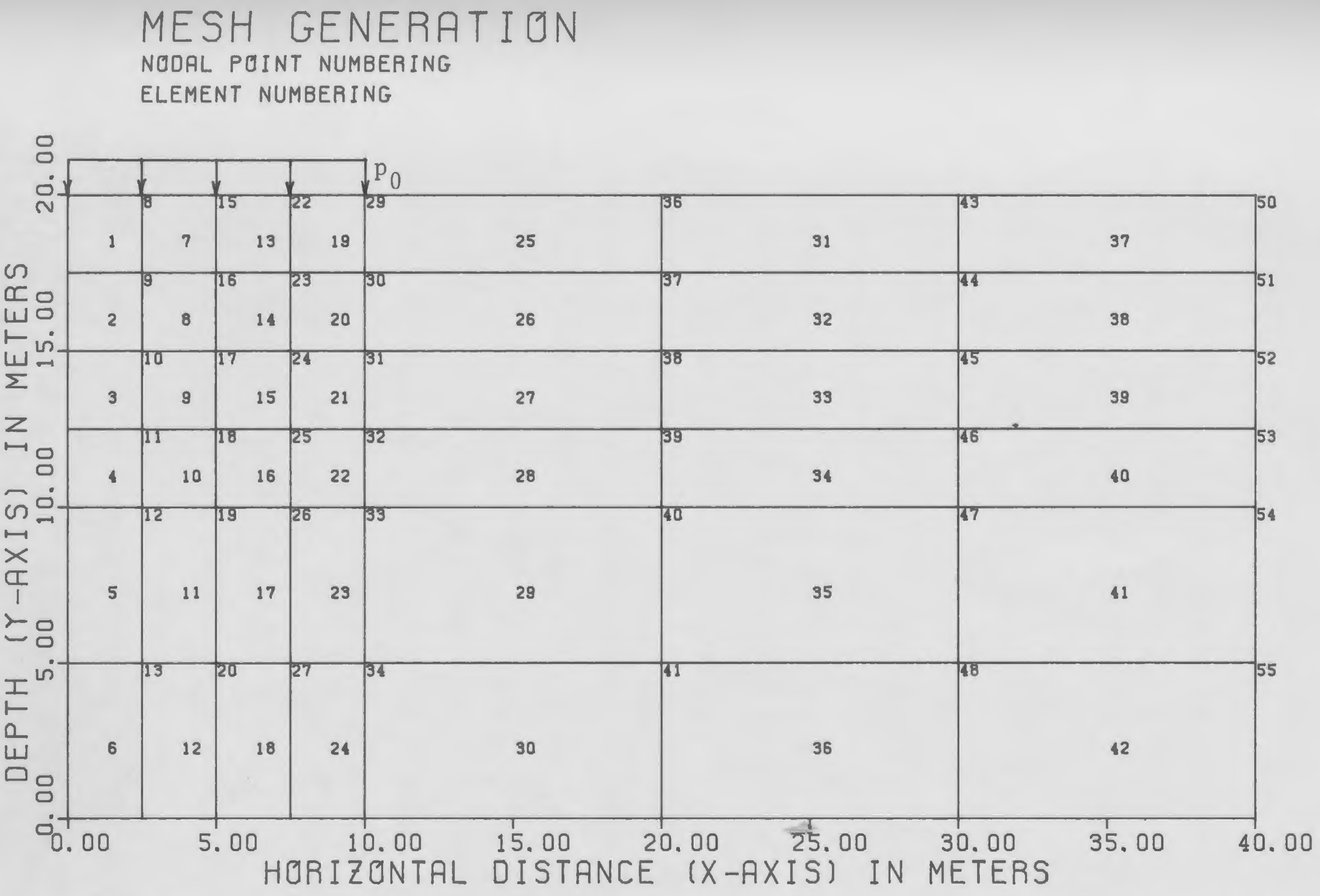


FIG. 3.3 FINITE ELEMENT MESH FOR STRIP FOUNDATION

the footing centerline and assumed to be smooth. The bottom boundary is assumed to be rough and rigid. The footing is assumed to deform under plane strain condition. Reduced integration technique is used to ensure the nearly incompressibility of the saturated soil medium.

In the initial undrained analysis the soil is modelled as a weightless two-phase material with $s_u = 100$ Kpa and $\phi_u = 0$, where s_u = undrained shear strength and ϕ_u = angle of internal friction. The drained elastic Lame's parameters $\lambda = 150\ s_u$, $\mu = 100\ s_u$ and drained Poisson's ratio, $\nu_d = 0.30$ are assumed for the soil skeleton whereas the bulk modulus for pore water is assigned as $K_w = 2000$ MPa. This gives an equivalent undrained Poisson's ratio, $\nu_u \cong 0.499$ approximately ensuring the nearly incompressibility of the soil medium. For the drained analysis, K_w is only set equal to zero while the other parameters remain unaltered. The drained and undrained deformations are computed based on Eqn. (3.49) and compared with the solution presented by Booker et al (1976). Fig. 3.4 depicts the comparison of the foundation deformation at surface along horizontal direction.

Figs. 3.5 to 3.7 present the comparisons of the distribution of the stress components such as p, (pore pressure), σ_x and σ_y along the depth. The results obtained from the present analysis are in excellent agreement with those obtained by Booker et al (1976). It is interesting to note that Booker et al (1976) obtained the solution of an undrained problem based on a different approach such as defining the problem in terms of a single parameter and then expanding the solution in terms of that parameter. Figs. 3.8 and 3.9 present the stress contour plots for vertical normal stress component, σ_y, and pore pressure, p, under undrained condition whereas Fig. 3.10 depicts only the normal stress component, σ_y, for drained conditon.

3.9.2 Dynamic Response Analysis of a Cantilever Beam

For the dynamic response analysis, a cantilever beam is chosen as an example problem, and is shown in Fig. 3.11. The beam is discretized with four noded isoparametric rectanglular elements. The mesh consists of 3 elements and 8 nodes. Nodes 1 and 2 are fixed and the beam is subjected to equal periodic forces of 1.0 cos (50t) at the free nodes 7 and 8.

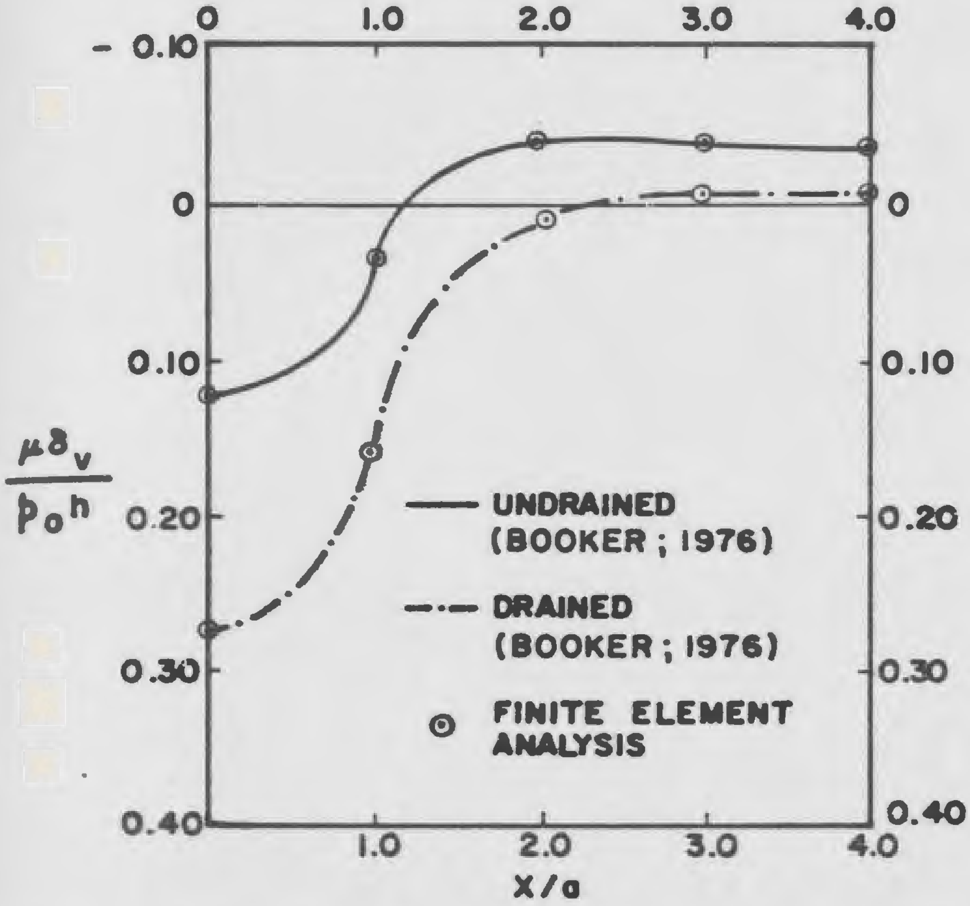


FIG. 3.4 SETTLEMENT OF THE FOUNDATION AT THE SURFACE

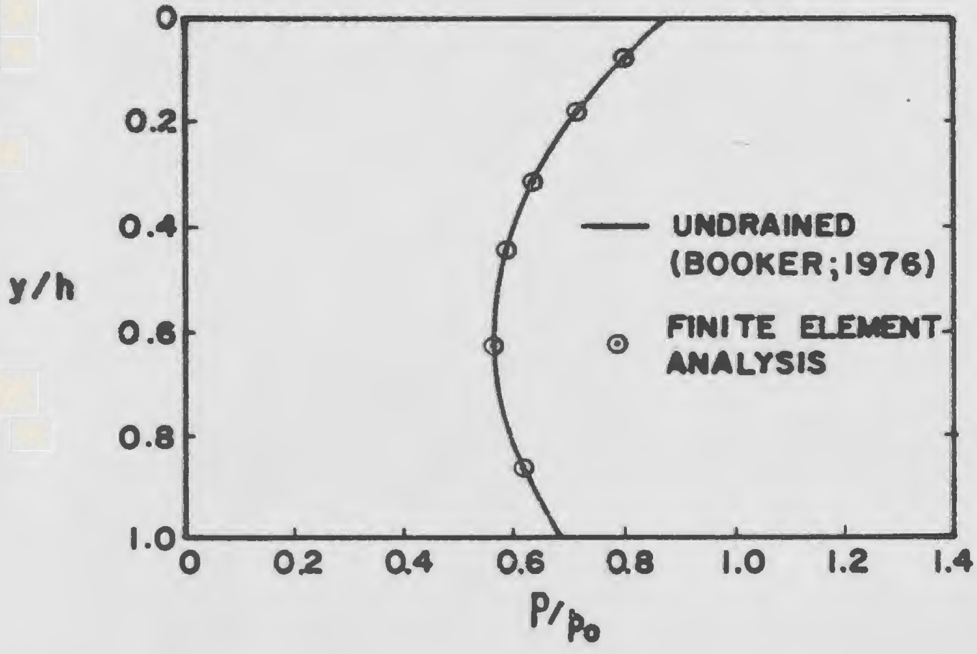


FIG. 3.5 DISTRIBUTION OF PORE PRESSURE ALONG THE CENTER LINE

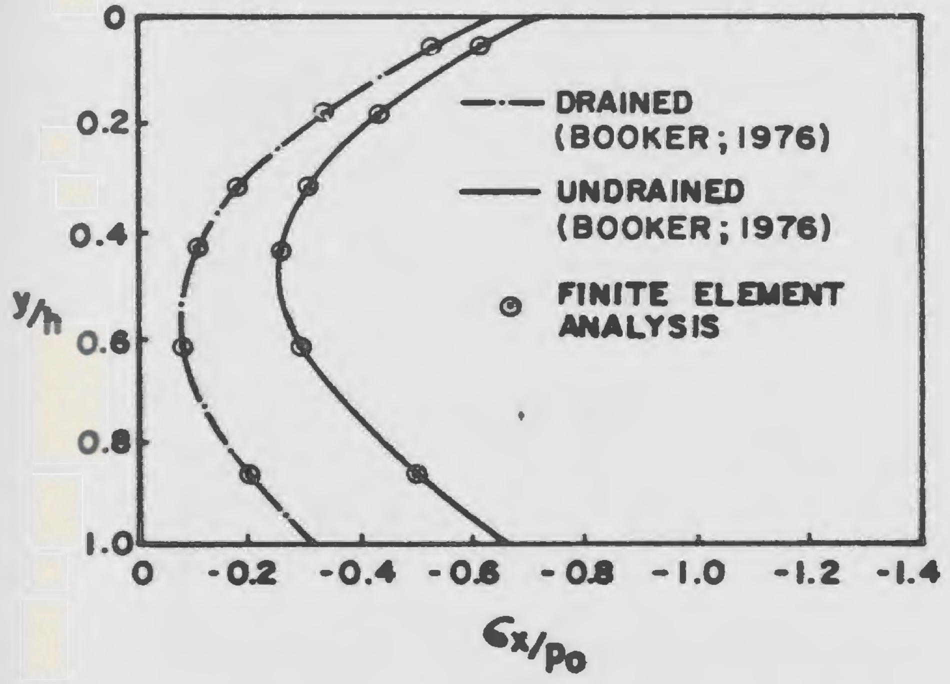


FIG. 3.6 DISTRIBUTION OF HORIZONTAL NORMAL STRESS

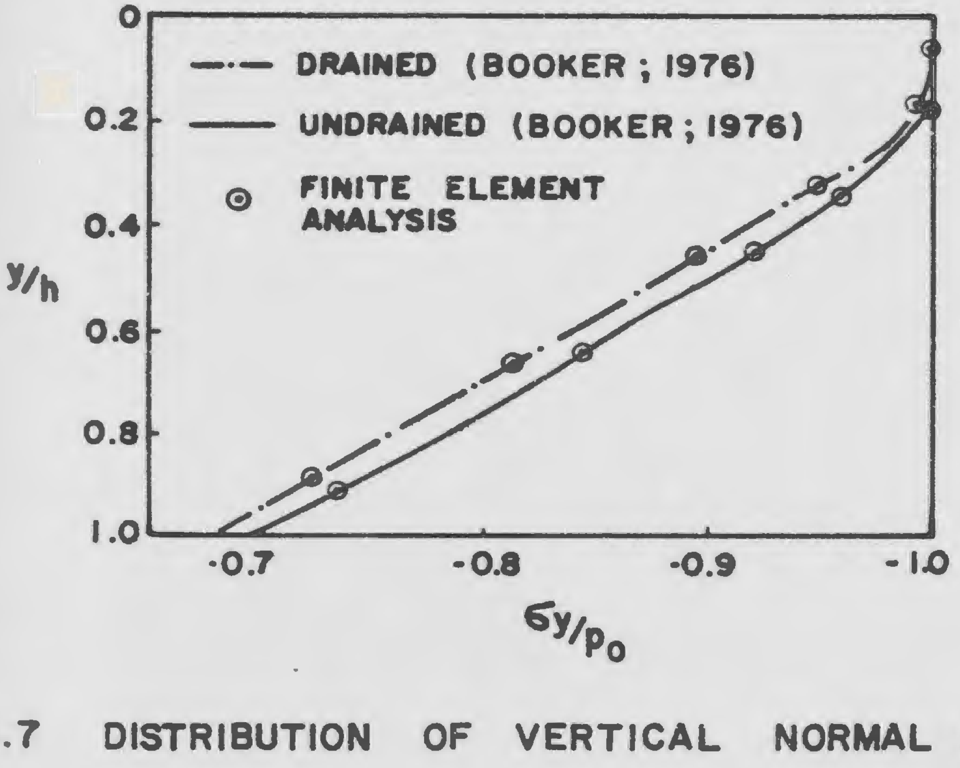


FIG. 3.7 DISTRIBUTION OF VERTICAL NORMAL STRESS

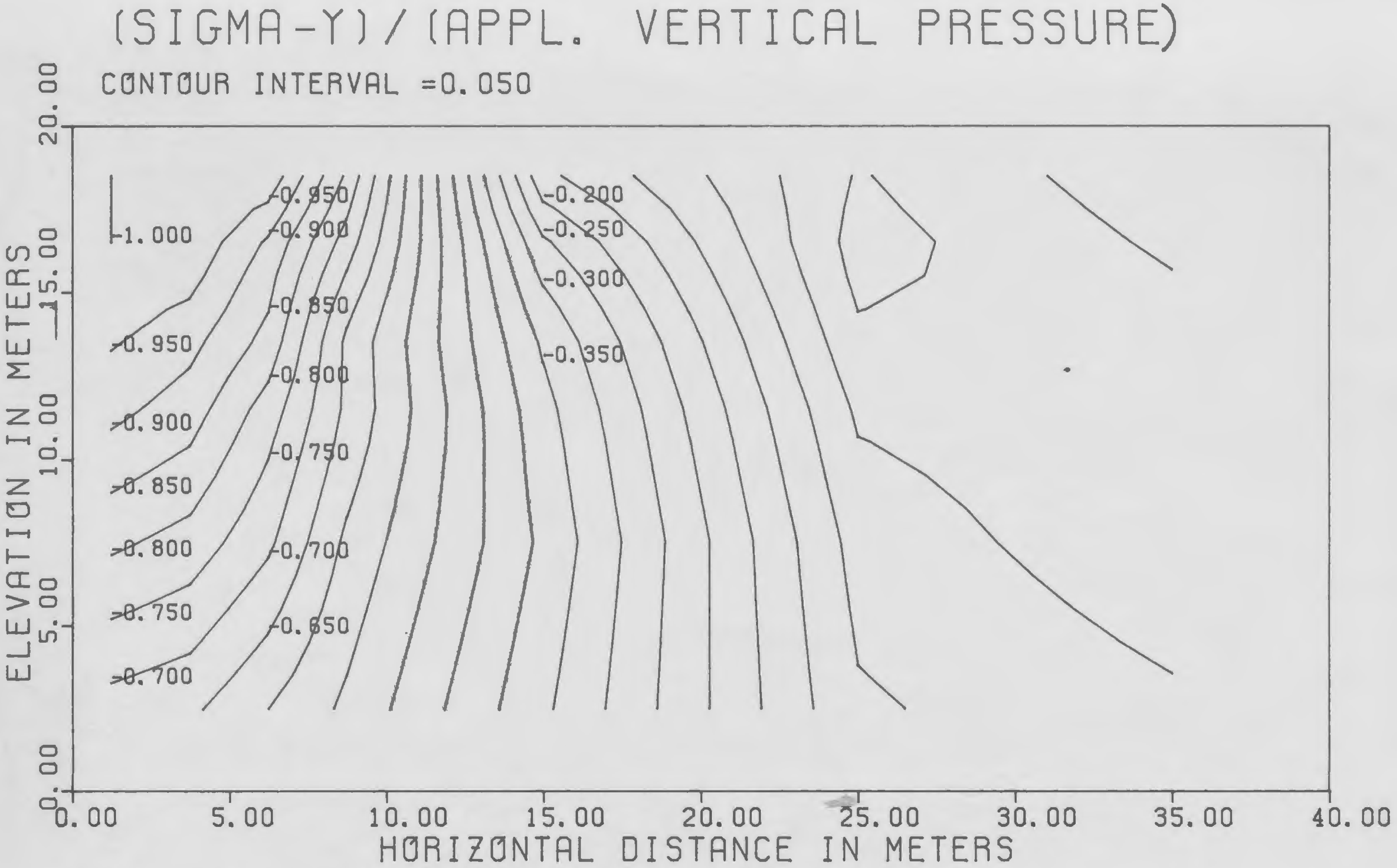


FIG. 3.8 STRESS CONTOURS FOR SIGMA-Y
(UNDRAINED CONDITION)

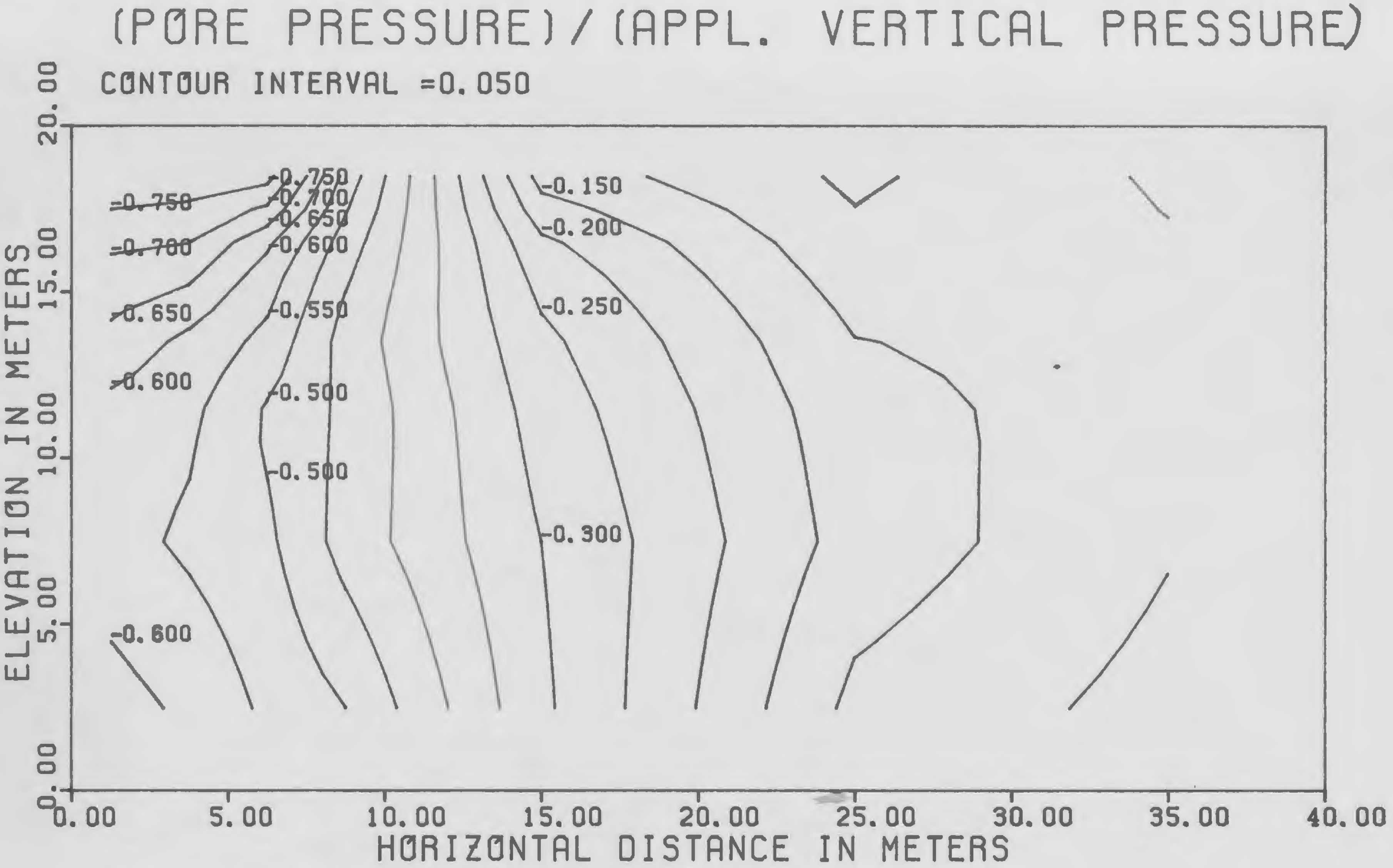


FIG. 3.9 PORE PRESSURE CONTOURS
(UNDRAINED CONDITION)

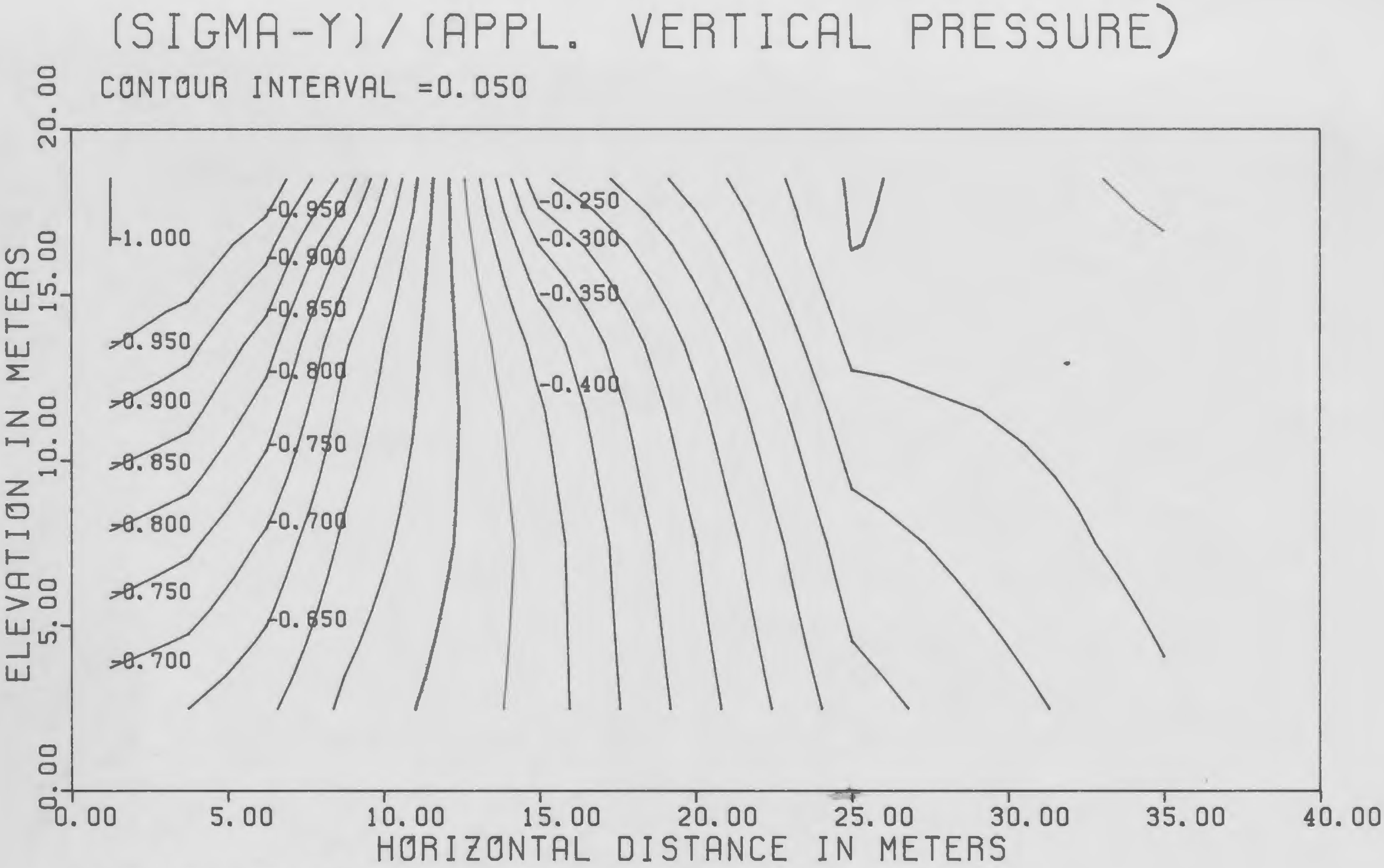


FIG. 3.10 STRESS CONTOURS FOR SIGMA-Y
(DRAINED CONDITION)

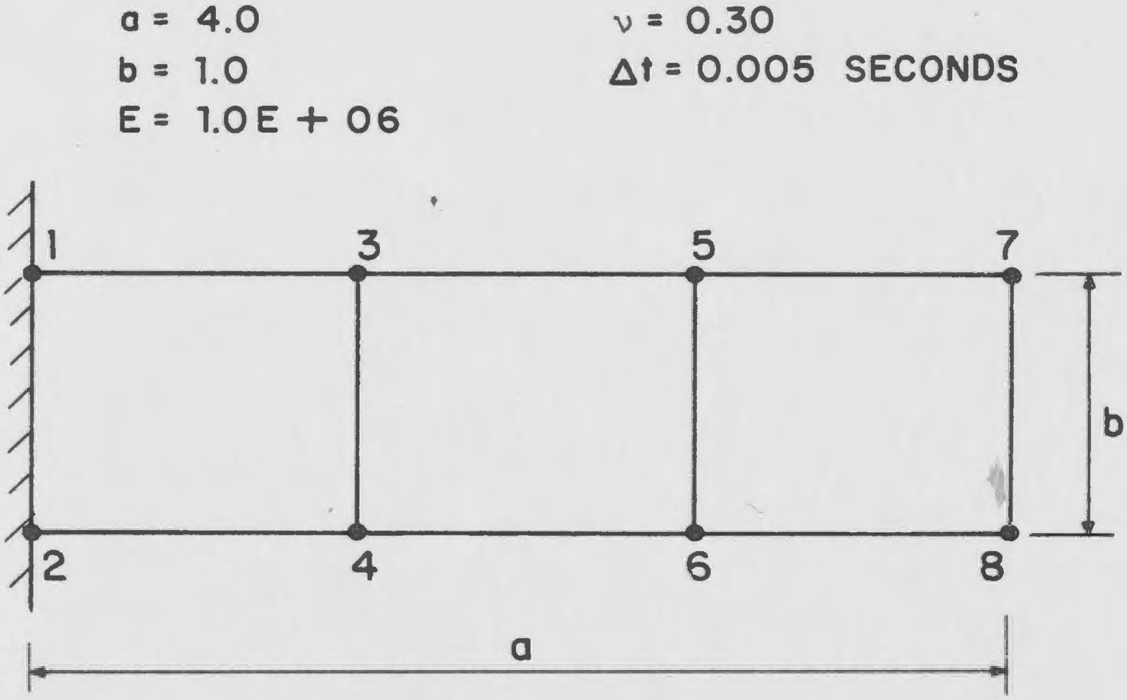


FIG. 3.11 CANTILEVER BEAM UNDER PERIODIC VERTICAL LOADING AT NODES 7 AND 8 (SMITH, 1982)

An eigenvalue analysis is first performed and the natural frequencies are shown in Table 3.1 and compared to those obtained by Smith (1982). The Rayleigh damping coefficients, $B_1 = 12.36$ and $B_2 = 1.2 \times 10^{-5}$ are obtained by specifying a modal damping ratios of 6.5% and 1.7% in first and second modes respectively. The response analysis is carried out for a few time steps and also compared in Table 3.2. The responses computed from the present programme compared reasonably well with those obtained by Smith (1982).

TABLE 3.1 - COMPARISON OF FREQUENCIES FOR CANTILEVER BEAM

Mode Numbers	Present Analysis	Smith (1982)	Difference in Percent
1	96	102	6.25
2	431	458	6.26
3	496	526	6.05
4	903	919	1.77
5	1350	1422	5.33
6	1490	1575	5.70
7	1600	1630	1.88
8	1710	1800	5.26
9	1780	1810	1.69
10	1870	2030	8.56
11	2290	2580	12.66
12	2640	2960	12.12

TABLE 3.2 - COMPARISON OF RESPONSE QUANTITIES FOR CANTILEVER BEAM

	Displacement		Velocity		Acceleration	
Time (Seconds)	Present Analysis	Smith (1982)	Present Analysis	Smith (1982)	Present Analysis	Smith (1982)
.005	0.483×10^{-5}	0.510×10^{-5}	0.290×10^{-2}	0.300×10^{-2}	1.160	1.200
.010	0.318×10^{-4}	0.330×10^{-4}	0.749×10^{-2}	0.776×10^{-2}	0.678	0.700
.015	0.755×10^{-4}	0.782×10^{-4}	0.955×10^{-2}	0.987×10^{-2}	0.146	0.144
.020	0.123×10^{-3}	0.127×10^{-3}	0.897×10^{-2}	0.912×10^{-2}	-0.390	-0.444
.025	0.161×10^{-3}	0.164×10^{-3}	0.573×10^{-2}	0.531×10^{-2}	-0.960	-1.077

CHAPTER IV

DYNAMIC SHAKEDOWN ANALYSIS OF OFFSHORE FOUNDATION SUBJECTED TO CYCLIC LOADING

4.1 Introduction

This chapter presents the dynamic shakedown analysis of a gravity type offshore foundation under cyclic loading conditions. Although the shakedown analysis is relatively new to the field of geotechnical engineering, its existence in literature related to structural mechanics has been well known for more than four decades.

For foundations subjected to loads varying in time in a nonproportional manner within prescribed limits, the classical limit theorems can give unsafe estimates of the collapse load, as failure can occur at loads well below the static collapse values. Shakedown theorems, which are generalizations of the limit theorems, would provide appropriate safe bounds for such complex loading programmes. For load reversal varying between prescribed limits, a foundation is said to shakedown when a state is reached such that all subsequent load applications produce only elastic changes. In recent years, very few attempts have been made to find practical solutions to the shakedown problem related to marine foundations resting on saturated soils. As the nonlinear dynamic response analysis of an offshore foundation to wave loading can only be carried out for a very limited number of cycles due to constraints on computing costs, the shakedown analysis becomes more practical and relevant to the overall design process.

This chapter is divided into two parts. The first part introduces the basic background and concepts on shakedown related theorems following Martin (1975). The second part describes the finite element formulation of the dynamic shakedown analysis of offshore foundation and its solution as a linear programming problem.

A computer programme has been developed based on the shakedown theory and typical solutions of two dimensional boundary value problems are obtained and the validity of the computer code is checked by comparing these solutions with the analytical results.

4.2 Assumptions

The basic assumptions of classical shakedown theory are as follows (König and Maier, 1981);

(a) The material obeys the elastic-perfectly plastic constitutive laws. The hardening (or softening) phenomenon in the shakedown process is neglected.

(b) The plastic flow rule is governed by the associative flow theory of plasticity i.e. normality rule is applicable. The yield surface is convex in nature and the material obeys the Drucker's postulate (1950) of material stability criteria.

(c) Effects of configuration changes do not influence the equilibrium condition of the body i.e. geometric nonlinearity is not being considered. Hence, the classical shakedown theorems are based on 'first-order' theory.

(d) The loading process is assumed to vary slowly in time i.e. inertia and damping forces are neglected and therefore the response process is quasistatic.

(e) The elastic properties governing the constitutive laws and the yield stresses are considered to be independent of temperature.

(f) The phenomenon of creep is not considered.

(g) The quantities involved in shakedown analysis can be expressed in deterministic form.

The assumption of elastic-perfectly plastic material implies that the yield surface remains unaltered over the deformation process. The stress vector can either be inside the yield surface (elastic domain) causing the plastic strain rates to be equal to zero or can touch the yield surface thus generating non-vanishing plastic strain rates (plastic domain).

The safe soil stress state, σ^s_{ij}, may be defined by the stress state corresponding to a point lying inside the yield surface whereas the admissible soil stress state, σ^a_{ij}, may be defined by the stress state corresponding to points including those on the yield surface. Here, σ_{ij}, denotes the cartesian stress tensor. The yield condition, f_y, can be written in the following forms:

$$f_y\ (\sigma^s_{ij}) < 0 \qquad \text{(elastic)} \qquad \ldots\ (4.1a)$$

and

$$f_y\ (\sigma^a_{ij}) \leq 0 \qquad \text{(elastic-plastic)} \qquad \ldots\ (4.1b)$$

The Drucker postulate of stability criteria states that the work of additional stresses on a closed stress path is non-negative, or that no energy can be recovered from the initially stressed element. The postulate is presented in the form

$$\left(\sigma_{ij} - \sigma^{s}_{ij}\right) \dot{\varepsilon}^{p}_{ij} > 0 \qquad \ldots (4.2a)$$

and

$$\left(\sigma_{ij} - \sigma^{a}_{ij}\right) \dot{\varepsilon}^{p}_{ij} \geq 0 \qquad \ldots (4.2b)$$

where σ_{ij} = total elastic-plastic stress state of bulk solid corresponding to strain, ε_{ij}.

4.3 Shakedown Analysis

Shakedown analysis is the generalization of the limit analysis in a sense that it provides information whether a body will shakedown or not under a given cyclic loading programme in contradistinction to the limit analysis which provides the information whether or not flow will take place in the same body under given loads.

4.3.1 Static Shakedown

Based on Melan's theorem, when an elastic-perfectly plastic body is subjected to a load, $\hat{T}_i\ (\bar{x},t)$, varying slowly between prescribed limits, it will shakedown if a time independent residual stress field, $\sigma^R_{ij}(\bar{x})$, can be found such that

$$f_y \left\{ \sigma^E_{ij}\ (\bar{x},\ t) + \sigma^R_{ij}\ (\bar{x}) \right\} \leq 0 \qquad \ldots (4.3)$$

for every $\bar{x}$ and t. $f_y\ (\sigma_{ij}) \leq 0$ is the yield condition and, $\sigma^E_{ij}\ (\bar{x},\ t)$, is the linear elastic stress response of the body to the loading programme, $\hat{T}_i\ (\bar{x},t)$. In the above expressions, σ_{ij}, $\bar{x}$, and t denote the cartesian stress tensor, the coordinate system and time respectively.

4.3.2 Kinematic Shakedown

According to Koiter's theorem, the continuum or the body will not shakedown i.e., it will fail ultimately by cyclic plastic deformation, if any admissible plastic strain rate cycle, $\dot{\varepsilon}^p_{ij}$, can be found over some time interval, τ, such that the following inequality holds:

$$\int_o^\tau dt \left(\int_{S_2} \hat{T}_i \dot{u}_i\ dS + \int_V B_i \dot{u}_i\ dV \right) > \int_o^\tau dt \int_V Q(\dot{\varepsilon}^p_{ij})\ dV \qquad \ldots (4.4)$$

where $\hat{T}_i$ = surface tractions prescribed over, S_2 (Fig. 3.1)

B_i = body force over the entire volume, V

$\dot{u}_i$ = admissible velocity field

and

$Q(\dot{\varepsilon}^p_{ij})$ = plastic dissipation density function uniquely determined by the kinematically admissible strain rate distribution, $\dot{\varepsilon}^p_{ij}$

Conversely, if shakedown occurs for all admissible cycles of plastic strain rates and arbitrary loads varying slowly between prescribed limits, then it is possible to find a scalar multiplier, $\beta_s \geq 1$, such that

$$\beta_s \int_o^\tau dt \left(\int_{S_2} \hat{T}_i \dot{u}_i \, dS + \int_V B_i \dot{u}_i \right) dV = \int_o^\tau dt \int_V Q(\dot{\varepsilon}^p_{ij}) \, dV \qquad \ldots (4.5)$$

The maximum value of, β_s, is then obviously the factor of safety with respect to shakedown.

4.3.3 Dynamic Shakedown

Based on Ceradini's dynamic shakedown theorem, the continuum subjected to dynamic loading will shakedown if among all the systems of initial conditions (including the self-equilibrating stress state as well as the distribution of initial displacements and velocities) there exist both a residual stress field and an elastic solution of the dynamic problem so that their sum gives stresses within the plastic limits. The elastic dynamic response has often been called as a fictitious process which is determined by imposing the arbitrary initial conditions such as displacement and velocity at some time which need not coincide with the actual ones.

In the case of a periodic dynamic loading, the possibility of shakedown is in fact determined by the forced vibration part of the response which is independent of the initial conditions of motion. The free vibration will decay in the early stage due to viscous properties of the system. Therefore, the dynamic shakedown problem may be reduced to a static case with the application of the proper amplification factor on the response.

For the sake of completeness and clarity, the proofs of the above theorems are presented in Appendix A.

4.4 Cyclic Loading

The foundation of a gravity type offshore structure is quite often subjected to repeated or cyclic loading (Fig. 1.1). This loading is primarily caused by wave action. A typical cyclic loading programme can be defined as follows:

$$\hat{T}_i\ (\bar{x},t) = \hat{T}_i\ (\bar{x},t + \omega) \qquad \ldots\ (4.6)$$

where t is any parameter which increases monotonically with real time, and, ω, is its period.

The soil stress distribution can be written as the sum of the elastic and the residual stress distributions as

$$\sigma_{ij}\ (\bar{x},t) = \sigma_{ij}^{E}\ (\bar{x},t) + \sigma_{ij}^{R}\ (\bar{x},t) \qquad \ldots\ (4.7)$$

The elastic soil stress distribution is also a function of cyclic loading and therefore it depends on the instantaneous value of the external loads. Then, σ_{ij}^{E}, can be expressed as

$$\sigma_{ij}^{E}\ (\bar{x},t) = \sigma_{ij}^{E}\ (\bar{x},t + \omega) \qquad \ldots\ (4.8)$$

Because of the cyclic nature of the loading programme, the total soil stress distributions, $\sigma_{ij}(\bar{x},t)$ and $\sigma_{ij}(\bar{x},t + \omega)$, are in equilibrium with the external loadings, $\hat{T}_i(\bar{x},t)$ and $\hat{T}_i(\bar{x},t + \omega)$, respectively. The stress field differences are also self-equilibrating i.e. in equilibrium with zero external forces.

4.4.1 Shakedown

Shakedown is said to occur when the plastic strain rates are zero. The foundation will attain a cyclic steady state after N cycles i.e.

$$\dot{\varepsilon}^{p}_{ij}(\bar{x},t) = 0 \quad \text{for } t > N\omega \qquad \ldots (4.9)$$

and the displacements, $U_i(\bar{x},t)$, are also cyclic. The time independent residual stress field, $\sigma^{R}_{ij}(\bar{x})$, is uniquely determined by the plastic strain, $\varepsilon^{p}_{ij}(\bar{x},t)$. The total stress is expressed from Eqn. (4.7) as

$$\sigma_{ij}(\bar{x},t) = \sigma^{E}_{ij}(\bar{x},t) + \sigma^{R}_{ij}(\bar{x})$$

The behaviour of the foundation can be considered in terms of the response to one cycle of loading such that $N\omega < t < (N + 1)\,\omega$. The elastic strain rates are given by

$$\dot{\varepsilon}^{E}_{ij}\ (\bar{x},t) = \hat{C}_{\kappa\ell}\left[\dot{\sigma}^{E}_{ij}\ (\bar{x},t) + \dot{\sigma}^{R}_{ij}\ (\bar{x},t)\right] \quad \ldots\ (4.10)$$

where, $\hat{C}_{\kappa\ell}$, represents the coefficients of the elasticity matrix.

The change in elastic strain field over a cycle is zero and therefore

$$\Delta\varepsilon^{E}_{ij}(\bar{x}) = 0 \quad \ldots\ (4.11)$$

However, the change in the total strains

$$\Delta\varepsilon_{ij}\ (\bar{x}) = \varepsilon_{ij}\ \{\bar{x},\ (N + 1)\omega\} - \varepsilon_{ij}\ \{\bar{x}, N\omega\} \quad \ldots\ (4.12)$$

must be kinematically admissible, since it represents the difference between two kinematically admissible fields. The change in the plastic strains over a cycle is defined here as

$$\Delta\varepsilon^{p}_{ij}\ (\bar{x}) = \int_{N\omega}^{(N+1)\omega} \dot{\varepsilon}^{p}_{ij}\ (\bar{x},t)\ dt \quad \ldots\ (4.13)$$

$\Delta\varepsilon^{p}_{ij}$ $(\bar{x})$ will also be kinematically admissible when the foundation reaches a cyclic steady state which determines the condition for shakedown.

4.4.2 Alternating Plasticity

For the situation when, $\Delta\varepsilon^p_{ij}\ (\bar{x}) = 0$, it is implied that the plastic strains, $\varepsilon^p_{ij}\ (\bar{x},t)$, and the displacements are cyclic in nature as well as their rates. This condition leads to the situation where the elements in the system undergo plastic strains which are alternating in nature i.e. change in sign with each cycle.

4.4.3 Incremental Collapse

When, $\Delta\varepsilon^p_{ij}\ (\bar{x},t) \neq 0$, the foundation undergoes incremental changes in displacement, $\Delta\dot{u}_i$, over each cycle. The work done by the external loading programme in each cycle may be written as

$$\int_{N\omega}^{(N+1)\omega} \hat{T}_i\dot{u}_i dt = \int_{N\omega}^{(N+1)\omega} dt \left(\int_V \sigma_{ij}\dot{\varepsilon}^E_{ij} dV + \int_V \sigma_{ij}\dot{\varepsilon}^p_{ij} dV \right) \quad \ldots (4.14)$$

Using Eqn. (4.11), Eqn. (4.14) reduces to

$$= \int_{N\omega}^{(N+1)\omega} dt \left(\int_V \sigma_{ij}\dot{\varepsilon}^p_{ij} dV \right) > 0 \quad \ldots (4.15)$$

Thus it can be seen the foundation will undergo increasing deformation with each cycle and if the cycling loading continues for a long period of time, the foundation will eventually fail as a result of unbounded deformation. This type of behaviour is often termed as progressive deformation or incremental collapse.

From the offshore foundation design point of view, a foundation subjected to a combination of cyclic loading (e.g. vertical, horizontal etc.), in which shakedown does not occur represents a situation where failure is almost certain.

4.5 Nonlinear Programming Approach to Shakedown Analysis

Consider a body subjected to cyclic loads, $\hat{T}\ (\bar{x},t)$ with an elastic solution, $\sigma_{ij}^{E}\ (\bar{x},t)$. Due to the linearity of the elastic solution, if the loads are magnified by β_s times i.e. $\beta_s\ \hat{T}_i\ (\bar{x},t)$, (where β_s is a scalar multiplier and $\beta_s > 1$), the elastic stress field will be, $\beta_s\ \sigma_{ij}^{E}\ (\bar{x},t)$. From Melan's theorem, if shakedown occurs for a multiplier, $\beta_s > 1$, there will be a shakedown residual stress field, $\sigma_{ij}^{R}\ (\bar{x})$, such that

$$f_y\ \{\beta_s\ \sigma_{ij}^{E}\ (\bar{x},t) + \sigma_{ij}^{R}\ (\bar{x})\} \leq 0 \qquad \dots (4.16)$$

The vector, $\{\beta_s\sigma_{ij}^{E}\ (\bar{x},t) + \sigma_{ij}^{R}\ (\bar{x})\}$, must not exceed the yield limit at any point in the body during the loading cycle.

As the yield function, $f_y\ (\sigma_{ij}) \leq 0$, is normally a convex quadratic function in stress space, the evaluation of the shakedown load factor, β_s, based on Melan's theorem leads to the following programming problem:

$$\beta = \text{Maximize } \beta_s$$

$$\text{subject to } f_y\ \{\beta_s\ \sigma^E_{ij}\ (\bar{x},t) + \sigma^R_{ij}\ (\bar{x})\} \leq 0 \qquad \ldots (4.17)$$

The determination of β thus becomes a nonlinear mathematical programming problem.

In order to apply Koiter's theorem for the evaluation of the shakedown load factor, β , the external work is equated with internal work for any admissible plastic strain rate cycle, $\dot{\varepsilon}^p_{ij}\ (\bar{x},t)$. Therefore, β, can be obtained by equating the external work to the internal energy.

From Eqn. (4.5) one obtains

$$\beta = \frac{\int_o^\tau dt \int_V Q\ (\dot{\varepsilon}^p_{ij})\ dV}{\int_o^\tau dt \left(\int_{S2} \hat{T}_i(\bar{x},t)\ \dot{u}_i(\bar{x},t)\ dS + \int_V B_i\ \dot{u}_i(\bar{x},t)\ dV \right)} \qquad \ldots (4.18)$$

It is obvious then from Koiter's theorem that β is the smallest multiplier obtained by equating external and internal work. Thus any value obtained from Koiter's theorem provides an upper bound on the shakedown load factor whereas the value obtained from Melan's theorem is lower bound.

4.6 Evaluation of Shakedown Load Factor

The determination of the shakedown load factor often requires the numerical solution of the optimization problems given by Eqn. (4.17) or Eqn. (4.18). Therefore, the constraints given by Eqn. (4.17) or Eqn. (4.18) must be solved by suitably discretizing the loading programme, stress field and the yield condition.

4.6.1 Discretization of the Loading Programme

The loading domain is represented in load space by, $F(\hat{T}_i) = 0$, and shown in Fig. 4.1. It can also be represented as an unit loading domain, $F_o(\hat{T}_i) = 0$, multiplied by the appropriate load factor, β_s. The actual loading history, $\hat{T}_i\ (\bar{x}, t)$, can pass through any point inside the loading domain, $F(\hat{T}_i) = 0$, as a certain function of time, t. As the foundation is subjected to a cyclic loading programme which passes

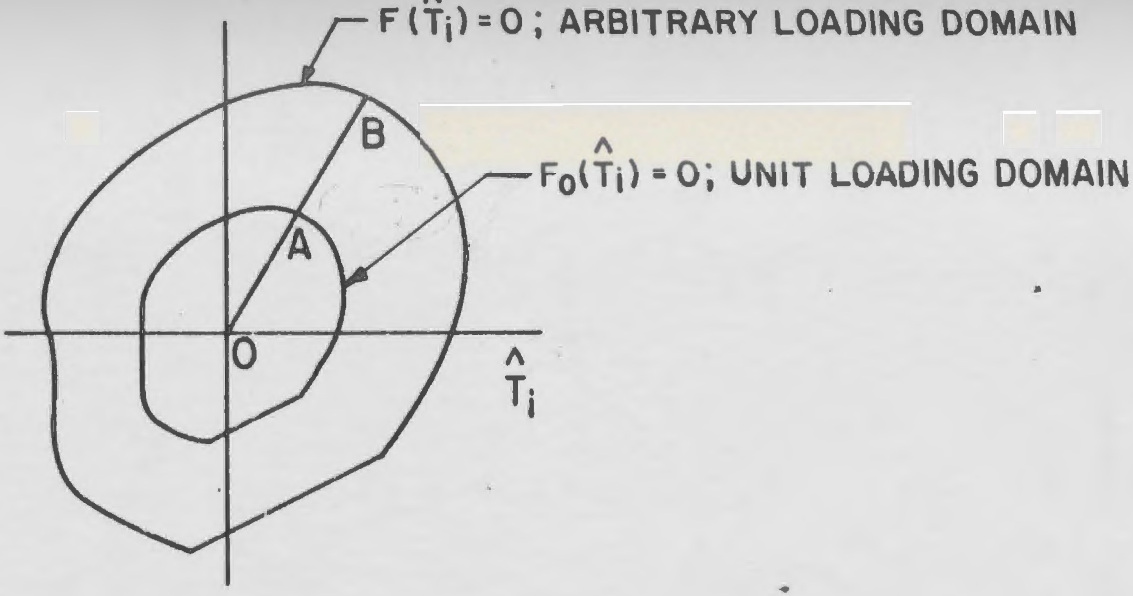


FIG. 4.1 GENERALIZED LOADING DOMAIN $\left(\frac{OB}{OA} = \beta\right.$; HO, 1972)

beyond the initial yield surface, the yield surface will start moving and form a subsequent yield surface which changes with time. Shakedown will occur if a yield surface can be set up such that it contains the loading cycle, $\hat{T}_i\ (\bar{x},t)$, which means the response thereafter will be purely elastic. On the other hand, if shakedown does not occur, the yield surface will continue to translate each time the load cycle is repeated with increased plastic deformation leading to incremental collapse.

The time dependent variable repeated loading function, $\hat{T}_i\ (\bar{x},\ t)$, can be expressed as

$$\hat{T}_i\ (\bar{x},t) = \sum_{s=1}^{n} \gamma_s(t)\ \bar{T}_s\ (\bar{x}) \qquad \ldots\ (4.19)$$

where $\bar{x} \in S_2$ (Fig. 3.1)

$\gamma_s\ (t)$ = s-th load parameter

$\bar{T}_s\ (\bar{x})$ = constant surface traction (reference) in the s-th unit load mode

The simplest load domain Ω in n-dimensional space is shown in Fig. 4.2 and it is assumed that the load factor, $\gamma_s(t)$, can vary within the load space. The load factor domain is also assumed to be convex.

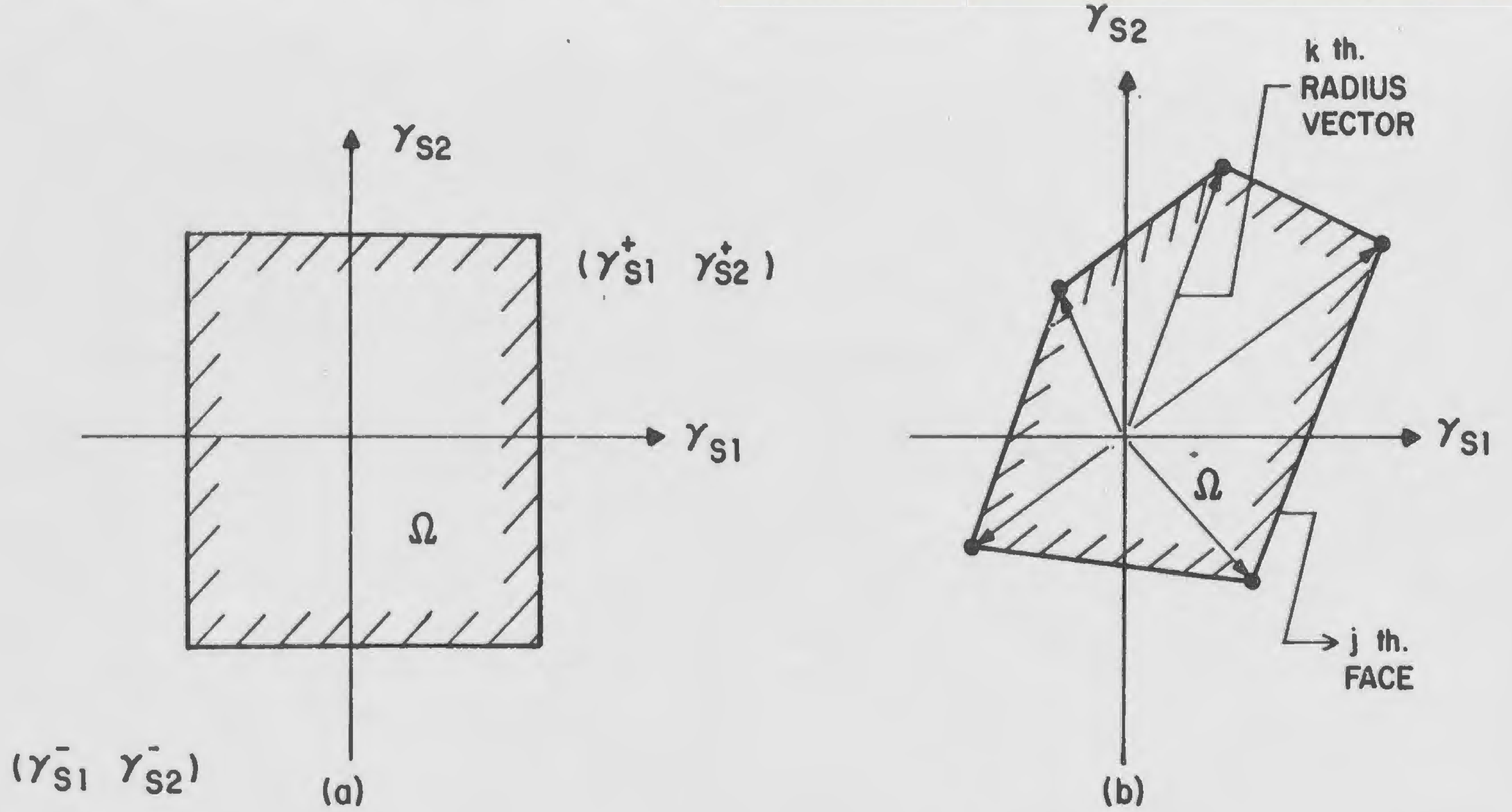


FIG. 4.2 DIFFERENT LOADING DOMAINS FOR SHAKEDOWN ANALYSIS (a) RECTANGULAR LOADING DOMAIN (b) ARBITRARY LOADING DOMAIN (BORKOWSKI AND KLEIBER, 1980)

Denoting the corners of the load domain by $\gamma_{s1}, \ldots \gamma_{sn}$, and the unique respective elastic stress responses corresponding to these load factors as $\sigma_{ij}^{1E}(\bar{x}), \ldots \sigma_{ij}^{nE}(\bar{x})$, it can be shown that a given system shakes down to a certain cyclic loading process which contains all the vertices of a given load domain Ω, (e.g. $\gamma_{s1}, \ldots \gamma_{sn}$), then it shakes down to any load path contained within Ω. Therefore, the mathematical problem of Eqn. (4.17) is expressed as follows:

$$\beta = \text{Max } \beta_s \qquad \ldots (4.20)$$

subject to

$$f_y \left\{ \beta_s \, \sigma_{ij}^{kE} (\bar{x}) + \sigma_{ij}^{R} (\bar{x}) \right\} \leq 0 \text{ in } V$$

where k = 1, 2, ... n

4.6.2 Discretization of the Stress Field

In order to compute, σ_{ij}^{kE} $(\bar{x},t)$, in the continuum formulation, it is necessary to discretize the stress field by using finite difference or finite element method. It is also necessary to express the residual stress field, σ_{ij}^{R} $(\bar{x})$, in terms of a discrete number of parameters. Finally, the yield condition is applied at discrete number of points in the body defined here as 'check points'.

4.6.3 Piecewise Linearization of the Yield Surface

The evaluation of shakedown load factor, β, from Eqn. (4.20) may be carried out as a nonlinear optimization problem by satisfying the yield condition at the selected 'check points' in the discretized body. However, this may lead to computational difficulty for large class of continuum problems. Therefore, the piecewise linearization of the yield function is used here in order to transfer a nonlinear mathematical programming problem to a linear programming one. The piecewise linearizaton of the convex yield function requires that the yield domain be a polyhedron in the stress space and represented by a finite number of yield planes. This means that instead of satisfying nonlinear constraints at selected 'check points' in the discretized body, it is sufficient to satisfy a large number of linear inequalities at those 'check points'. Therefore, the shakedown load factor, β_s, can be obtained by solving a linear programming problem (Maier, 1969).

Assuming, $f_y(\sigma_{ij}, \varepsilon^p_{ij}) \leq 0$, a differentiable yield function such that, $f_y(\sigma_{ij}) \leq 0$ denotes the current elastic domain in the, σ_{ij}, space, one may write

$$\dot{f}_y = \frac{\partial f_y}{\partial \sigma_{ij}} \dot{\sigma}_{ij} + \frac{\partial f_y}{\partial \varepsilon^p_{ij}} \dot{\varepsilon}^p_{ij} \qquad \ldots (4.21)$$

The plastic strain rate satisfying the normality condition is defined by

$$\dot{\varepsilon}^p_{ij} = \dot{\Delta} \frac{\partial f_y}{\partial \sigma_{ij}} \qquad \ldots (4.22)$$

where Δ = plastic multiplier.

The plastic flow rule at the yield limit, $f_y = 0$, requires $\dot{\Delta} \geq 0$, $f_y \leq 0$, $\dot{f}_y \dot{\Delta} = 0$ and $f_y \dot{\Delta} = 0$. The last condition is often termed as a complementary condition, which implies that plastic yielding (loading process) is active only when the current stress, σ_{ij}, is in contact with yield surface, $f_y = 0$, and loss of contact of the stress point with the yield surface ($\dot{f}_y \leq 0$, unloading process) are events which are mutually exclusive. Writing the above equations as follows, one obtains

$$\dot{f}_y = \tilde{N}^T \dot{\sigma}_{ij} - H\dot{\Delta} \qquad \ldots (4.23a)$$

where

$$\tilde{N} = \frac{\partial f_y}{\partial \sigma_{ij}} \qquad \ldots (4.23b)$$

and

$$H = - \frac{\partial f_y}{\partial \varepsilon^p_{ij}} \cdot \frac{\partial f_y}{\partial \sigma_{ij}} \qquad \ldots (4{\cdot}23c)$$

The scalar H is defined here as the 'hardening modulus', where $H > 0$ represents work hardening, $H = 0$ elastic-perfectly plastic and $H < 0$ softening behaviour. Of course, $H < 0$, is not permitted because of the Drucker's stability postulate and associative nature of the flow rule (normality condition).

Linearizing the continuously differentiable yield function, $f_y\ (\sigma_{ij}, \varepsilon^p_{ij}) \leq 0$, into a suitable number of yield planes, R, one can write in matrix form the following

$$\left\{\dot{f}_{yi}\right\} = \left[\tilde{N}_i\right]^T \left\{\dot{\sigma}_{rs}\right\} - \left[H_i\right] \left\{\dot{\Delta}_i\right\} - \left\{K_{oi}\right\} \leq 0 \qquad \ldots (4.24a)$$

where

$$\left[\tilde{N}_i\right]^T = \frac{\partial f_{yi}}{\partial \sigma_{rs}} \qquad \ldots (4.24b)$$

$$\left[H_i\right] = - \frac{\partial f_{yi}}{\partial \varepsilon^p_{rs}} \cdot \frac{\partial f_{yi}}{\partial \sigma_{rs}} \qquad \ldots (4.24c)$$

$i = (1, 2, \ldots R)$

and

$\left\{K_{oi}\right\}$ = each element in the vector defines the distance of each yield plane, $f_{yi} = 0$, from the origin (Fig. 4.3)

Therefore, Eqn. (4.22) is rewritten as

$$\dot{\varepsilon}^{p}_{rs} = \sum_{i=1}^{R} \frac{\partial f_{yi}}{\partial \sigma_{rs}} \dot{\Delta}_i \qquad \ldots (4.25)$$

The graphical interpretation of Eqn. (4.24) is shown in Fig. 4.3.

Representing the yield function, f_{yi}, for a typical 'check point', j, within the body for ideal plasticity (H = 0) one can write

$$\left\{f^{j}_{yi}\right\} = \left[\tilde{N}^{j}_{i}\right]^{T} \left\{\dot{\sigma}^{j}_{rs}\right\} - \left\{K^{j}_{oi}\right\} \leq 0 \qquad \ldots (4.26)$$

Eqn. (4.26) describes the inequality that represents the elastic polyhedron at 'check point' j, in which each of its i = 1, 2, ... R yield planes is identified with an unit normal vector in, $\left[\tilde{N}^{j}_{i}\right]$, and a element in vector, $\left\{K^{j}_{oi}\right\}$, which represents the distance of each yield plane from the origin (Fig. 4.3).

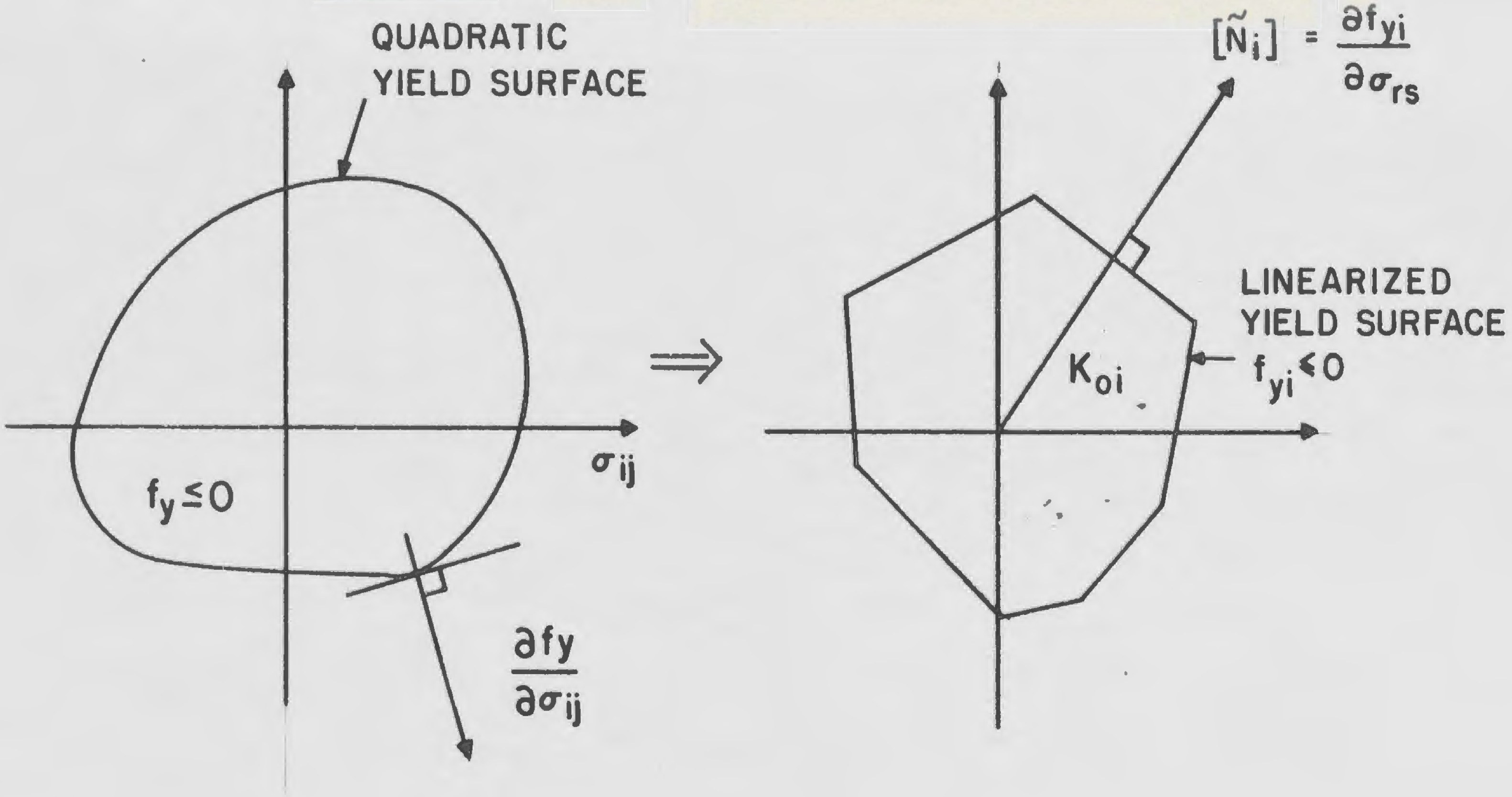


FIG. 4.3 PIECEWISE LINEARIZED YIELD SURFACE (MAIER, 1976)

4.7 Finite Element Formulation for Dynamic Shakedown Analysis

This section describes the application of the finite element method to determine the shakedown load for the foundation of a gravity type offshore structure subjected to dynamic cyclic loading. For the sake of numerical computation, the foundation is discretized into an assemblage of four noded isoparametric elements following Haldar et al (1982a and 1982b). In the present finite element model, displacement compatibility across the boundary between two adjacent elements is satisfied whereas the equilibrium condition will be satisfied only in a global sense.

The foundation is discretized with M number of isoparametric finite elements having N^o number of unconstrained displacements and a finite number of subvolumes, V_j, (j = 1, 2, ... M).

4.7.1 Evaluation of Internal Work

The plastic energy dissipation rate, $Q\ (\dot{\varepsilon}^{pj}_{rs})$, associated with a subvolume, V_j, can now be expressed as

$$Q\ (\dot{\varepsilon}^{pj}_{rs}) = \int_{V_j} \left\{\dot{\varepsilon}^{pj}_{rs}\right\}^T \left\{\sigma^j_{rs}\right\} dV \qquad \ldots\ (4.27)$$

and substituting Eqn. (4.25) in Eqn. (4.27) results in

$$Q\ (\dot{\varepsilon}^{pj}_{rs}) = \int_{V_j} \sum_{i=1}^{R} \frac{\partial f_{yi}}{\partial \sigma_{rs}} \dot{\Delta}^j_i \left\{\sigma^j_{rs}\right\} dV$$

$$= \left\{\dot{\underset{\sim}{\Delta}}{}^j_i\right\}^T \left[\tilde{N}^j_i\right]^T \left\{\sigma^j_{rs}\right\} \qquad \ldots\ (4.28)$$

where $\left\{\dot{\underset{\sim}{\Delta}}{}^j\right\}^T = \int_{V_j} (\dot{\Delta}^j_1 \ \ldots\ \dot{\Delta}^j_R)\ dV$

and $\left[\tilde{N}\right]$ = matrix whose columns contain $\frac{\partial f_{yi}}{\partial \sigma_{rs}}$

Therefore the total plastic energy dissipation rate for the entire foundation is obtained by assembling $Q\ (\dot{\varepsilon}^{pj}_{rs})$ over all elements as

$$\dot{Q} = \sum_{j=1}^{M} Q\ (\dot{\varepsilon}^{pj}_{rs})$$

$$= \sum_{j=1}^{M} \left\{\dot{\underset{\sim}{\Delta}}{}^j\right\}^T \left[N^j_i\right]^T \left\{\sigma^j_{rs}\right\}$$

$$= \sum_{j=1}^{M} \left\{\dot{\underset{\sim}{\Delta}}{}^j\right\}^T \left\{f^j_{yi} + K^j_{oi}\right\}$$

$$= \left\{K_o\right\}^T \left\{\dot{\underset{\approx}{\Delta}}\right\} \qquad \ldots\ (4.29)$$

where $\left\{K_o\right\}$ and $\left\{\dot{\underset{\approx}{\Delta}}\right\}$ collect all the components of $\left\{K^j_{oi}\right\}$ and $\left\{\dot{\underset{\sim}{\Delta}}{}^j\right\}$ for all elements (j = 1, 2, ... M), and every yielding mode, (i = 1, 2, ... R). Correspondingly, the vector, $\left\{\hat{f}_y\right\}$, collects all $\left\{f_{yi}^j\right\}$.

The internal work is then evaluated by integrating Eqn. (4.29) over one cycle, τ,

$$W_{int} = \int_0^{\tau} \dot{Q}\, d\tau = \int_0^{\tau} \left\{K_o\right\}^T \left\{\dot{\underset{\approx}{\Delta}}\right\} d\tau$$

$$= \left\{K_o\right\}^T \int_0^{\tau} \left\{\dot{\underset{\approx}{\Delta}}\right\} d\tau$$

$$= \left\{K_o\right\}^T \left\{\underset{\approx}{\Delta}\right\} \qquad \ldots (4.30)$$

4.7.2 Kinematically Admissible Plastic Strain

From Eqn (3.381), the bulk load vector, $\left\{p_B\right\}$, for an element, j, can be written in the following form:

$$\left\{p_B^j\right\} = \int_V \underline{\phi}_m^T \left\{\begin{matrix} X_B \\ Y_B \end{matrix}\right\} dV + \int_{S2} \underline{\phi}_m^T \left\{\hat{T}\right\} dS$$

Assembling over all elements, and using Eqn. (3.25), it can now be written as

$$\left[G_o\right]^T \left\{\sigma\right\} = \left\{P_B\right\} \qquad \ldots (4.31)$$

where $\left[G_o\right]^T$ collects all the integrals, $\overset{M}{\Sigma} \int_{V_j} \left[\underset{\approx}{\psi}^j\right]^T dV$,

$\left\{P_B\right\}$ collects all bulk load vectors, $\overset{M}{\Sigma} \left\{p_B^j\right\}$

for the entire assemblage and therefore, $\left\{\sigma\right\}$, represents the collection of all stress components. If no loads are applied to the body, only self equilibrating residual stress field will be present and equilibrium is provided by

$$\left[G_o\right]^T \left\{\sigma\right\} = 0 \qquad \ldots (4.32)$$

From Eqn. (4.5), for an element j, the external work is equated with the internal energy dissipated during one cycle, τ, and therefore

$$\int_o^\tau \left\{p_B^j\right\}^T \left\{\dot{\underline{U}}^j\right\} dt = \int_o^\tau \left(\int_{V_j} \left\{\sigma^j\right\}^T \left\{\dot{\varepsilon}^{pj}\right\} dV_j\right) dt \qquad \ldots (4.33)$$

Substituting Eqn. (3.25) in Eqn. (4.33) for element, j

$$\int_\tau \int_{V_j} \left\{\sigma^j\right\}^T \left[\underset{\approx}{\psi}^j\right] \left\{\dot{\underline{U}}^j\right\} dV_j = \int_\tau \left(\int_{V_j} \left\{\sigma^j\right\}^T \left\{\dot{\varepsilon}^{pj}\right\} dV_j\right)$$

and using Eqn. (4.28)

$$\left(\int_{V_j} \left[\underset{\approx}{\psi}^j\right] dV_j\right) \left\{\underline{U}^j\right\} = \left[\tilde{N}^j\right] \left\{\underset{\sim}{\Delta}^j\right\} \qquad \ldots (4.34)$$

Assembling over all elements j = 1, 2, ... M, one can write

$$\left[G_o\right] \left\{\underline{U}\right\} = \left[\tilde{N}\right] \left\{\underset{\approx}{\Delta}\right\} \qquad \ldots (4.35)$$

where $\left[G_o\right]$ is known as the assembled compatibility matrix of the entire foundation system. $\left[\tilde{N}\right] = \left(\text{diag.} \; \tilde{N}^1 \; \tilde{N}^2 \; \ldots \; \tilde{N}^M\right)$ represents the matrix containing gradient submatrices $\left[\tilde{N}^j\right]$ along its main diagonal. Each of the block diagonal matrices collects the column vectors $\left[\tilde{N}^j_i\right]$ of all the yield planes of each element as

$$\left[\tilde{N}^j_i\right] = \left[\tilde{N}^j_1, \; \tilde{N}^j_2 \; \ldots \; \tilde{N}^j_R\right] \qquad \ldots (4.36)$$

Correspondingly, the vector, $\left\{\underline{U}\right\}$ represents the nodal displacements of the discretized system.

4.7.3 Evaluation of External Work

Using the virtual work principle and assembling over all elements, the total external work during one cycle can be expressed as

$$W_{ext} = \sum_{j=1}^{M} \int_{o}^{\tau} \left\{\underline{\dot{U}}^j\right\}^T \left\{p^j_B\right\} d\tau = \sum_{j=1}^{M} \int_{o}^{\tau} d\tau \int_{V_j} \left\{\sigma^{Ej}\right\}^T \left\{\dot{\varepsilon}^{pj}\right\} dV$$

$$\ldots (4.37)$$

Assume that $\dot{\varepsilon}^p$ is a vector which has a fixed direction in strain space. Let $\left\{\sigma^E\right\}_{max}$ indicate the value of $\left\{\sigma^E\right\}$ which has the largest component in the direction of $\left\{\dot{\varepsilon}^{p}\right\}$. Then the following inequality holds

$$\left\{\sigma^{E}\right\}^{T}_{max} \left\{\dot{\varepsilon}^{p}\right\} \geq \left\{\sigma^{E}\right\}^{T} \left\{\dot{\varepsilon}^{p}\right\} \qquad \ldots (4.38)$$

Substituting Eqn. (4.38) into Eqn. (4.37) and using Eqn. (4.28) gives'

$$W_{ext} = \int_{\tau} d\tau \sum_{j=1}^{M} \left\{\sigma^{Ej}\right\}^{T}_{max} \left[\tilde{N}^{j}\right] \left\{\dot{\underset{\sim}{\Delta}}^{j}\right\} \qquad \ldots (4.39)$$

Eqn. (4.39) is rewritten in the following form as

$$W_{ext} = \int_{\tau} d\tau \sum_{j=1}^{M} \left\{\dot{\underset{\sim}{\Delta}}^{j}\right\}^{T} \left[\tilde{N}^{j}\right]^{T} \left\{\sigma^{Ej}\right\}_{max} \qquad \ldots (4.39a)$$

Using Eqn. (4.30) and Eqn. (4.39a), Eqn. (4.18) can be rewritten as a programming problem:

$$\beta = \text{minimize} \left\{K_{o}\right\}^{T} \left\{\dot{\underset{\approx}{\Delta}}\right\}$$

subject to

$$\int_{\tau} d\tau \sum_{j=1}^{M} \left\{\dot{\underset{\sim}{\Delta}}^{j}\right\}^{T} \left[\tilde{N}^{j}\right]^{T} \left\{\sigma^{Ej}\right\}_{max} = 1$$

where $\dot{\underset{\sim}{\Delta}} \geq 0$

and $\dot{\varepsilon}^{pj}$ is compatible $\qquad \ldots (4.40)$

As described in Chapter 3, the soil beneath the gravity foundation will be in a state of undrained condition during the event of a storm which may typically last for several hours and therefore, the dynamic equations of motion for the discretized system under undrained state can be written in assembled form following Eqn. (3.48) as

$$\left[M^*\right]\left\{\underline{\ddot{U}}\right\}+\left[C^*\right]\left\{\underline{\dot{U}}\right\}+\left[K^*\right]\left\{\underline{U}\right\}=\left\{P_B\right\}$$

Here, the matrix [K*], represents the undrained stiffness matrix of the entire assemblage defined in Eqn. (3.50) and is expressed as

$$\left[K^*\right] = \sum^{M} \left(\int_V \psi_{11}^T \quad D_{sd} \quad \psi_{11} \quad dV + \int_V \alpha^2 M_o \quad \psi_{22}^T \quad \psi_{22} \quad dV\right)$$

4.8 Condition For Dynamic Shakedown

The necessary condition of shakedown for a discretized system under dynamical situation can be stated as follows:

"If the system shakes down, then some constant plastic multiplier, Δ, some initial conditions $\underline{U}^*(\bar{x},t^*)$ $\underline{\dot{U}}^*(\bar{x},t^*)$ and a time t*, exist such that the yield condition can be satisfied at any time $t \geq t^*$ as follows:

$$\left\{f_y^*\right\} = \left[\tilde{N}\right]^T \left\{\overset{*}{\sigma}(\bar{x},t)\right\} - \left\{K_o\right\} \leq 0 \quad \ldots (4.41)$$

The asterisk sign represents the fictitious process with suitably adjusted initial conditions so that the response $\overset{*}{\sigma}(t)$, becomes periodic (cyclic) after time $t \geq t^*$."

Substituting the value of, $\overset{*}{\sigma}(\bar{x},t)$,from Eqn. (4.7) into Eqn. (4.41), one obtains

$$\left\{f_y^*\right\} = \left[\tilde{N}\right]^T \left\{\overset{*}{\sigma}^E(\bar{x},t) + \sigma^R(\bar{x})\right\} - \left\{K_o\right\} < 0 \quad \ldots (4.42)$$

where, $\sigma^R(\bar{x})$, defines the time-independent residual stress field.

$\left\{\overset{*}{\sigma}^E(\bar{x},t)\right\}$ is defined here as the fictitious linear elastic dynamic stress response of the undrained soil medium to the given cyclic loading, $\left\{P_B(t)\right\}$, after a finite number of cycles. The stress, $\overset{*}{\sigma}^E(\bar{x},t)$, is based on initial conditions of displacement, $\underline{U}^*$, and velocity $\dot{\underline{U}}^*$, which need not coincide with the normally assumed zero initial conditions $\underline{U}(\bar{x},o) = 0.0$ and $\dot{\underline{U}}(\bar{x},o) = 0.0$.

The fictitious elastodynamic stress response of each element, j, defined as, $\overset{*}{\sigma}^{Ej}(t,\underline{U}^*,\dot{\underline{U}}^*)$, is projected on the outward normal vector, $\left[\tilde{N}_i^j\right]$, of each yield plane, i, (Fig. 4.4). Maximizing these projections with respect to time, $t \geq t^*$, gives

$$\tilde{M}_i^j = \max_{t \geq t^*} \left[\tilde{N}_i^j\right]^T \cdot \overset{*}{\sigma}^{Ej} (t, \underline{U}^*, \underline{\dot{U}}^*) \qquad \ldots (4.43)$$

where

$$\left\{\tilde{M}_i^j\right\}^T = \left[\tilde{M}_1^j,\ \tilde{M}_2^j\ \ldots\ \tilde{M}_R^j\right]^T_{max}$$

and assembling all these $\tilde{M}_i^j$ vectors in one place, one obtains

$$\left\{\tilde{M}\right\} = \left\{\tilde{M}_i^j\right\} \qquad \ldots (4.44)$$

where i = 1, 2, ... R

and $\quad$ j = 1, 2, ... M

Based on the elastic dynamic response computed from Eqn. (3.48), using initial conditions e.g. displacement $\underline{U}^* = \underline{\bar{U}}^*$ and velocity $\underline{\dot{U}}^* = \underline{\dot{\bar{U}}}^*$, the $\tilde{M}$-vector is evaluated from Eqn. (4.44). The maximum value of, β, of the programming problem is then expressed as

$$\beta = \max \beta_s$$

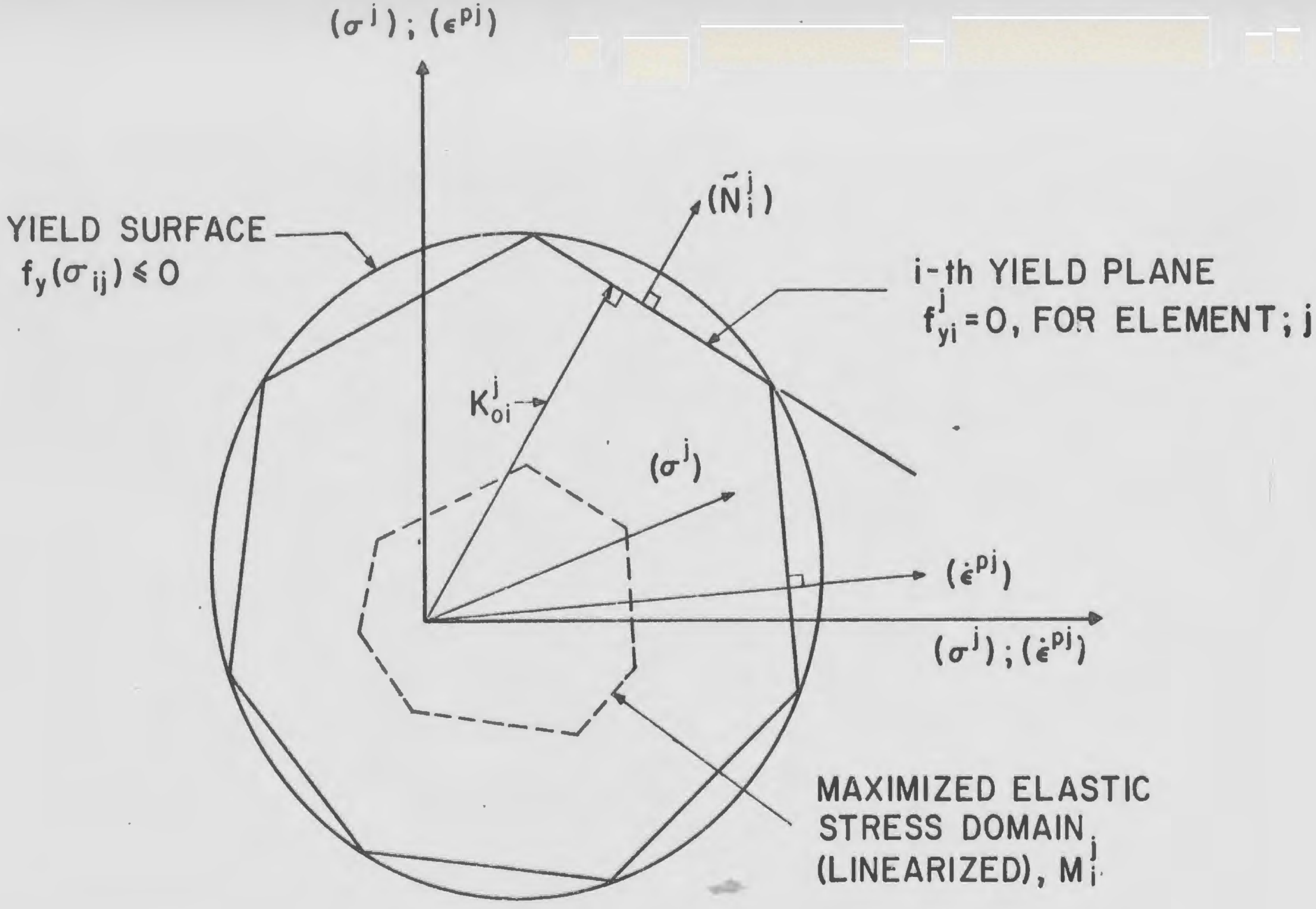


FIG. 4.4 GRAPHICAL REPRESENTATION OF LINEARIZED ELASTIC STRESS DOMAIN, MAXIMIZED WITH RESPECT TO TIME, FOR A TYPICAL ELEMENT, j.

subject to

$$\beta_s \left\{\tilde{M}\right\} + \left[\tilde{N}\right]^T \left\{\sigma^R\right\} \leq \left\{K_o\right\} \qquad \ldots (4.45)$$

where β_s is a common multiplier to elastic dynamic (or static) stress response and also a function of t^*, $\underline{\bar{U}}^*$, $\underline{\dot{\bar{U}}}^*$ through $\tilde{M}$. Assume $\underline{\bar{U}}^*$ and $\underline{\dot{\bar{U}}}^*$ as the particular initial conditions such as displacement and velocity which make the fictitious elastic response process to be periodic with respect to the external cyclic loading. For a general loading history, it may be very difficult to obtain an optimum value of β as a function of Δ, $\underline{\bar{U}}^*$ and $\underline{\dot{\bar{U}}}^*$. However, the optimization problem can greatly be simplified if the external loading process is periodic. In this case, the free vibration part of the response will decay after a finite time i.e. $t > N\omega$ due to damping and therefore the periodic response is described as

$$\overset{*}{\sigma}{}^E (\bar{x},t) = \overset{*}{\sigma}{}^E (\bar{x},t+\omega) \qquad \ldots (4.45a)$$

$\underline{\bar{U}}^*(t)$, $\underline{\dot{\bar{U}}}^*(t)$ represent the displacement and velocity at time $t = N\omega$, defined here as fictitious initial conditions, which make the response to be periodic i.e. independent of the free vibration part. By imposing these initial conditions at $t = 0$, the conditions for periodicity as expressed by Eqn. (4.46) can automatically be verified for t > 0.

4.9 Linear Programming Formulation

4.9.1 Primal Problem

The scalar multiplier factor, β_s, is called statically admissible if Eqn. (4.45) is satisfied for some σ^R. The search for, β, is a solution of the following linear programming problem

$$\beta = \max_{\beta_s,\ \sigma^R} \beta_s \qquad \dots (4.46)$$

subject to

1) $\left[G_o\right]^T \left\{\sigma^R\right\} = 0$

2) $\beta_s \left\{\tilde{M}\right\} + \left[\tilde{N}\right]^T \left\{\sigma^R\right\} \leq \left\{K_o\right\}$

and 3) $\beta_s > 0$

which can be written in the following 'tableau' form

$$\beta = \text{Max} \quad \begin{array}{|c|c|c|c} \beta_s & \{\sigma^R\} & & \\ \hline 1 & \{0\}^T & & \\ \hline 0 & [G_o]^T & \{0\} & = \\ \hline \{\tilde{M}\} & [\tilde{N}]^T & \{K_o\} & \leq \\ \hline \end{array} \qquad \dots (4.47)$$

In order to ensure the nonnegativity of the variable $\{\sigma^R\}$, one can write

$$\{\sigma^R\} = \{\sigma^R\}^+ - \{\sigma^R\}^- \qquad \dots (4.48)$$

with $\{\sigma^R\}^+ \geq 0$ and $\{\sigma^R\}^- \geq 0$

Then the modified 'tableau' becomes

$$\beta = \text{Max} \quad \begin{array}{|c|c|c|c|c} \beta_s & \{\sigma^R\}^+ & \{\sigma^R\}^- & & \\ \hline 1 & \{0\}^T & \{0\}^T & & \\ \hline 0 & [G_o]^T & -[G_o]^T & \{0\} & = \\ \hline \{\tilde{M}\} & [\tilde{N}]^T & -[\tilde{N}]^T & \{K_o\} & \leq \\ \hline \end{array} \qquad \dots (4.49)$$

4.9.2 Dual Problem

In the linear programming formulation of an optimization problem, any maximization problem can be related to its counterpart problem such as miminization having the same vectors and matrices and connected with the former by some duality properties.

The matrix formulation of Eqn. (4.40) is furnished through Eqn. (4.30) for internal work, Eqn. (4.35) for the condition of kinematically admissible plastic strain rate cycle and Eqn. (4.39a) for external work. Therefore, the dual programme is written in the following matrix form:

$$\beta = \min_{\underline{U},\ \underset{\simeq}{\Delta}} \left\{K_o\right\}^T \left\{\underset{\simeq}{\Delta}\right\} \qquad \ldots (4.50)$$

subject to

a) $\left[G_o\right] \{\underline{U}\} - \left[\tilde{N}\right] \{\underset{\simeq}{\Delta}\} = 0$

b) $\{\tilde{M}\}^T \{\underset{\simeq}{\Delta}\} = 1$

In order to ensure the nonnegativity of the variable, $\{\underline{U}\}$, one can write

$$\{\underline{U}\} = \{\underline{U}\}^+ - \{\underline{U}\}^- \qquad \ldots (4.51)$$

with $\{\underline{U}\}^+ \geq 0$ and $\{\underline{U}\}^- \geq 0$

Therefore the above Eqn. (4.51) is expressed in the 'tableau' form as

$$\beta = \text{Min} \quad \begin{array}{|c|c|c|c|c} \multicolumn{1}{c}{\{\underline{U}\}^{+}} & \multicolumn{1}{c}{\{\underline{U}\}^{-}} & \multicolumn{1}{c}{\{\underset{\approx}{\Delta}\}} & \multicolumn{1}{c}{} & \\ \hline \{0\}^T & \{0\}^T & \{K_o\}^T & \multicolumn{1}{c}{} & \\ \hline [G_o] & -[G_o] & -[\tilde{N}] & \{0\} & = \\ \hline \{0\}^T & \{0\}^T & \{\tilde{M}\} & 1 & = \\ \hline \end{array} \qquad \ldots (4.52)$$

The dual problem, Eqn. (4.52), has as many variables as Eqn. (4.49), the primal problem, has number of constraints. The computation time is proportional only to the number of variables but vary to the cube of the number of constraints. Therefore, formulation as described by Eqn. (4.52) is more attractive from computational standpoint.

4.10 Dead Load

If the foundation is subjected to non-repeated loading such as gravity loading, T_D , below the bearing capacity of the foundation in addition to the variable repeated or cyclic loading, $\{\hat{T}\}$, the loading programme then may be written as

$$\{p_B(t)\} = \beta_s \{\hat{T}(t)\} + \{T_D\} \qquad \ldots (4.53)$$

The energy balance of the external and internal work may be written as

$$\{K_o\}^T \{\underset{\approx}{\Delta}\} = \beta_s \{\tilde{M}\}^T \{\underset{\approx}{\Delta}\} + \int^{\tau} \{T_D\}^T \{\underline{\dot{U}}\}\, dt$$

or

$$\{K_o\}^T \{\underset{\approx}{\Delta}\} - \{T_D\}^T \{\underline{U}\} = \beta_s \{\tilde{M}\}^T \{\underset{\approx}{\Delta}\} \quad \ldots (4.54)$$

Therefore the modified 'tableau of Eqn. (4.52) taking into account the effects of non cyclic loading (e.g submerged weight of the structure) becomes

$$\begin{array}{r|c|c|c|c|l} & \{\underline{U}\}^+ & \{\underline{U}\}^- & \{\underset{\approx}{\Delta}\} & & \\ \hline \beta = \text{Min} & -\{T_D\}^T & \{T_D\}^T & \{K_o\}^T & & \\ \hline & [G_o] & -[G_o] & -[\tilde{N}] & \{0\} & = \\ \hline & \{0\}^T & \{0\}^T & \{\tilde{M}\}^T & 1 & = \\ \hline \end{array} \quad \ldots (4.55)$$

4.11 Summary of Steps to Evaluate the Shakedown Load Factor for the Offshore Foundation Problem

Step 1. Carry out the dynamic analysis with prescribed initial conditons, or static analysis depending on the nature of the problem for the prescribed load domain. For static response, consider the peaks of the loads (maxima), and for the dynamic response, adjust the initial conditions so that the response becomes periodic with respect to the cyclic wave loading.

Step 2. Maximize these stress response vectors with respect to the assumed linearized yield planes, so that the vector, $\left\{\tilde{M}\right\}$, may be constructed.

Step 3. Construct the global compatibility $\left[G_o\right]$ and gradient, $\left[\tilde{N}\right]$, matrices for the entire structure-foundation system. Also, compute the global $\{K_o\}$ vector based on the individual yield plane, (i = 1, 2, ... R) and assembling over all elements (j = 1, 2, ... M).

Step 4. Carry out the minimization problem based on the linear programming 'tableau' given by Eqn. (4.55).

4.12 Checking of Computer Programme

Two types of problems are chosen in order to compute the shakedown load factors based on the kinematic formulation given in Eqn. (4.55). These problems are i) a thick cylinder and ii) a strip footing. Both problems are analysed assuming a plane strain condition.

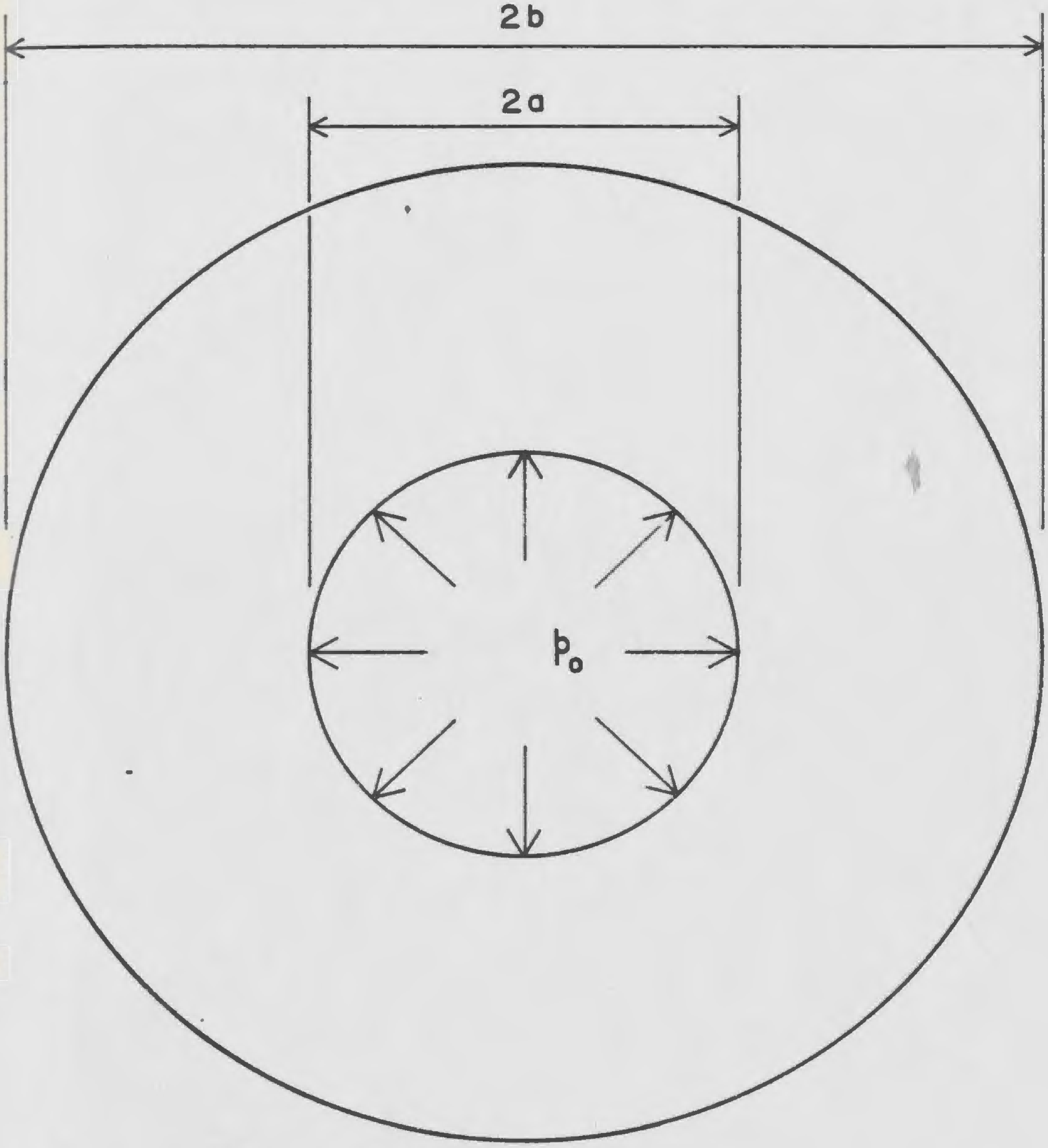
4.12.1 Shakedown Load For A Thick Cylinder

The thick cylinder shown in Fig. 4.5(a) is subjected to an internal pressure and has a ratio of outer to inner radius of 2.50. The quarter of the cylinder is discretized because of the symmetry. Three types of finite element meshes are considered for the analyses and shown in Figs. 4.5(b), 4.5(c) and 4.5(d). The internal pressure, P_o, is assumed to vary repeatedly between zero and some prescribed values, P^*. The material properties used in all the analyses are $\frac{E}{\sigma_o} = 0.50 \times 10^3$, $\nu = 0.30$ and $\sigma_o = 0.418$ MPa where E = Young's modulus, ν = Poisson's ratio and σ_o = uniaxial yield stress in tension.

The material is assumed to be elastic-perfectly plastic and obey Von Mises yield condition. Six linearized planes are used to describe the yield surface.

The theoretical solutions for elastic and limit pressures are given by the following two equations (Hill, 1950) as

$$p_e = \frac{c}{\alpha_1} \quad 3\left(\frac{G}{G+3K_c}\right)^2 + \left(\frac{b}{a}\right)^4 \quad {}^{-1/2}$$



$0 < p_o < p^*$

FIG. 4.5(a) THICK CYLINDER UNDER REPEATED INTERNAL PRESSURE.

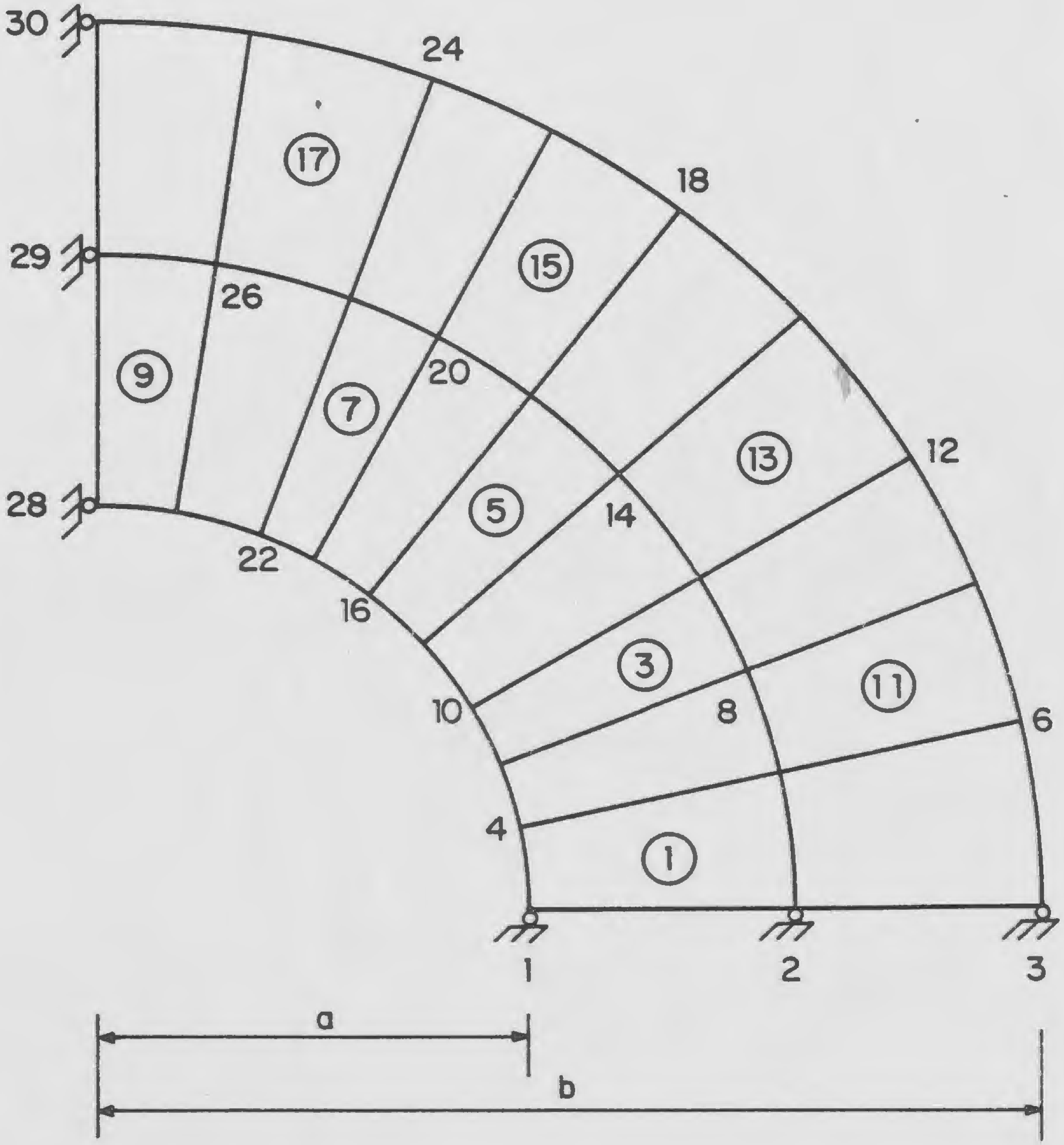


FIG. 4.5(b) FINITE ELEMENT IDEALIZATION OF A THICK CYLINDER (18 ELEMENTS)

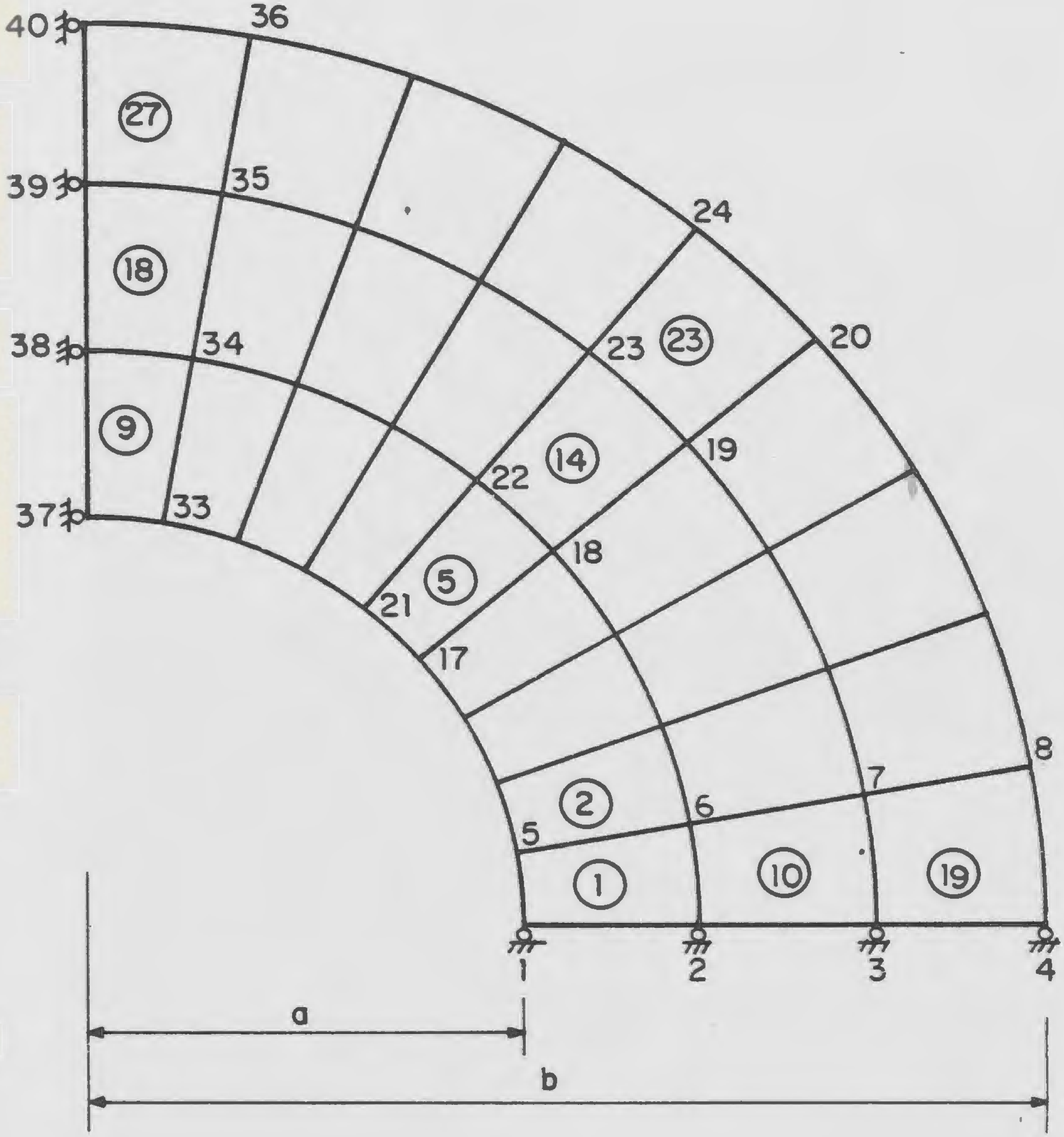


FIG. 4.5(c) FINITE ELEMENT IDEALIZATION OF A THICK CYLINDER (27 ELEMENTS)

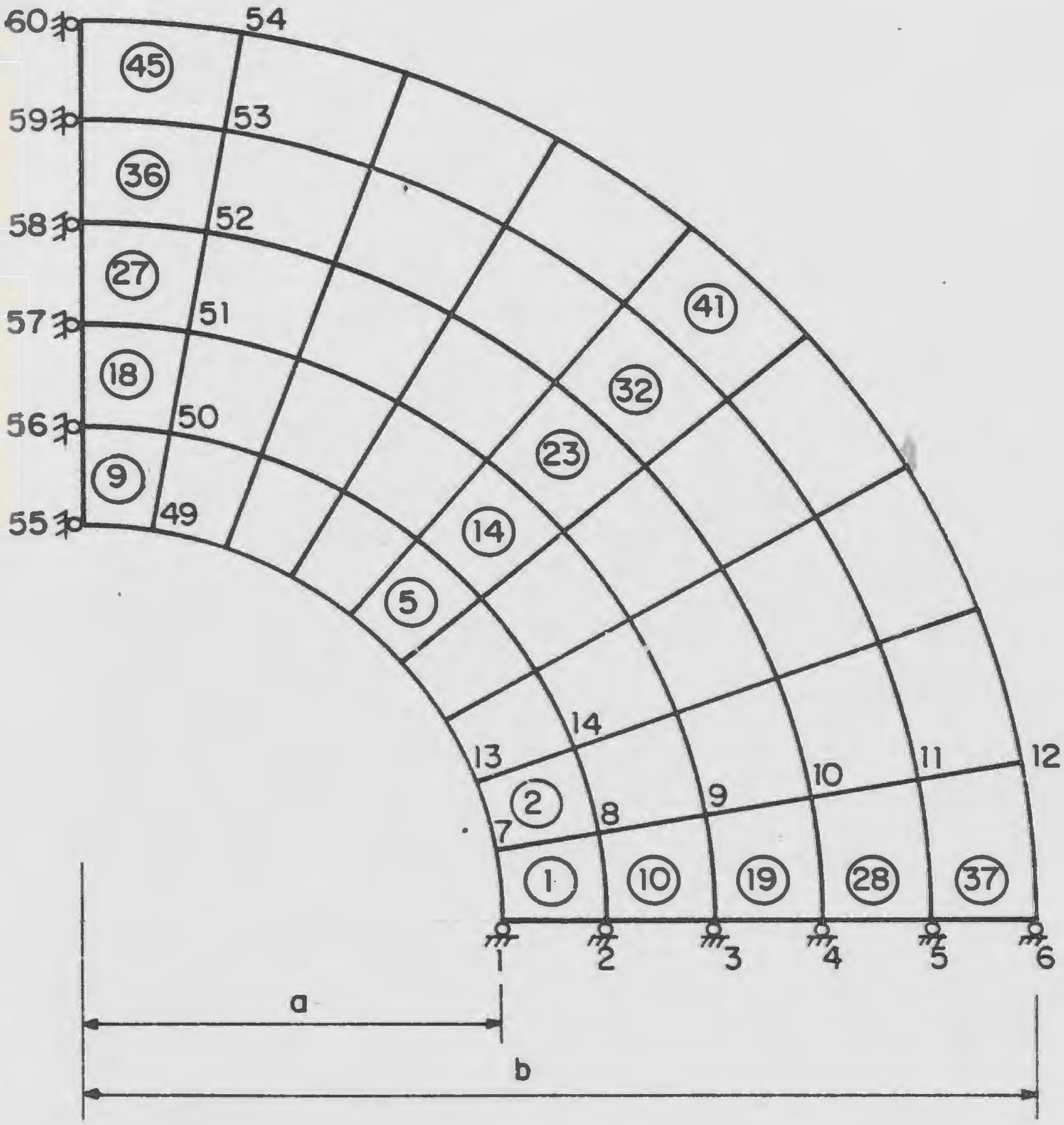


FIG. 4.5 (d) FINITE ELEMENT IDEALIZATION OF A THICK CYLINDER (45 ELEMENTS)

where

$$\alpha_1 = \frac{a^2}{(b^2-a^2)}$$

and

p_c = limit pressure

$= 2c \ln (\frac{b}{a})$

where

b = outer radius

a = inner radius

c = $\sigma_o / \sqrt{3}$

σ_o = yield pressure for uniaxial tension

K_c = bulk modulus = $\frac{E}{3(1-2\nu)}$

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

ν = Poisson's ratio

Because of the one parameter loading arrangement, theoretical limit pressure, p_c is equal to the shakedown pressure P_s. Table 4.1 provides limit pressures for three types of mesh arrangements. It is seen from Fig. 4.6 that as the number of elements increase the solution converges quite rapidly to the theoretical values. In Fig. 4.6 the elastic shakedown and collapse pressures are expressed in nondimensional form such as elastic pressure = $^{p}e/c$, shakedown pressure = $^{p}s/c$ and collapse pressure = $^{p}c/c$.

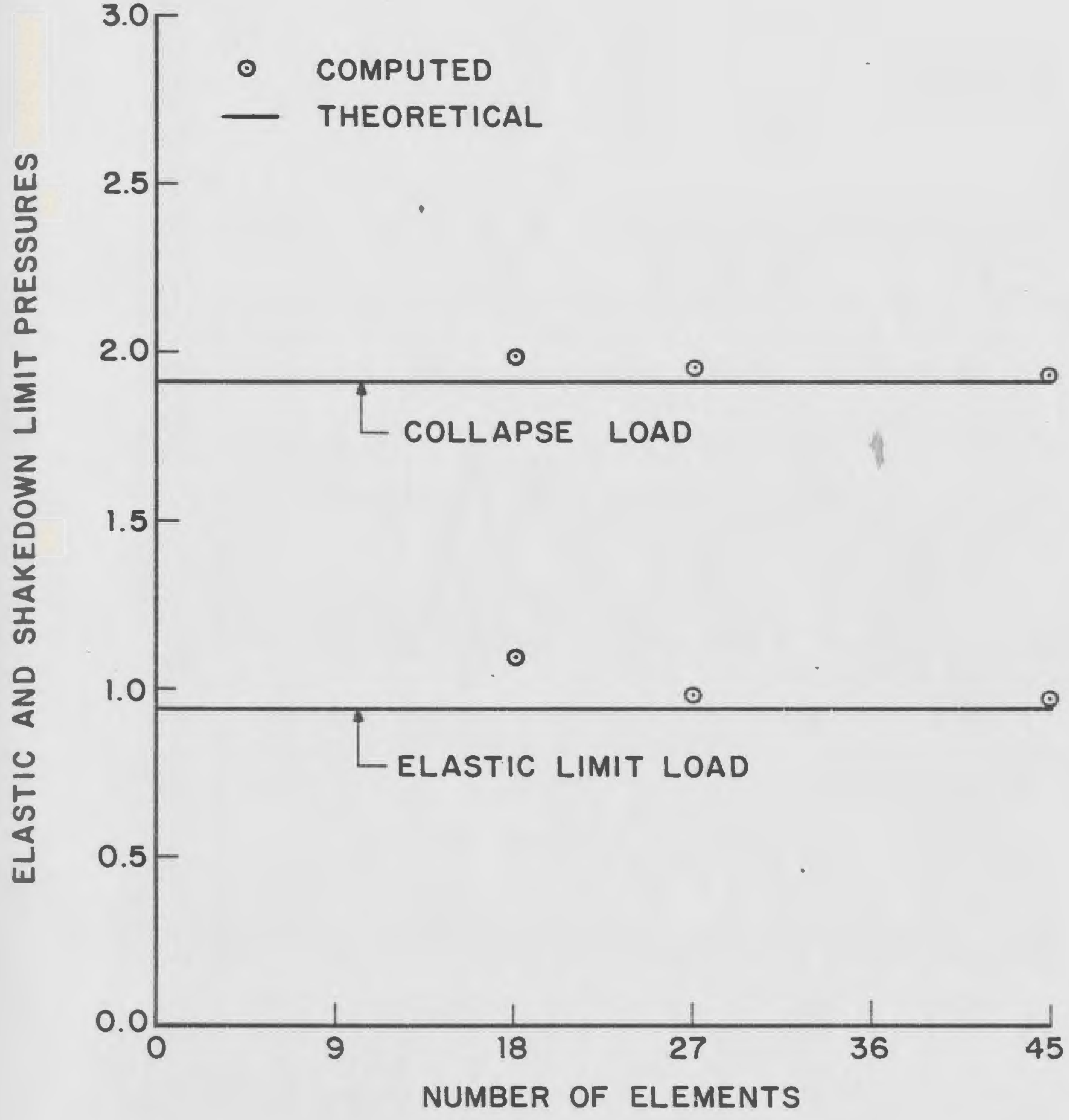


FIG. 4.6 INFLUENCE OF NUMBER OF ELEMENTS ON SHAKEDOWN LIMIT PRESSURE

TABLE 4.1 - INFLUENCE OF NUMBER OF ELEMENTS ON SHAKEDOWN LIMIT PRESSURES

Number of Elements	Number of Nodal Points	Number of Equations	Elastic Pressure (p_e/c)	Shakedown Pressure (p_s/c)	Theoretical Collapse Pressure (p_c/c)
18	30	54	1.22	1.94	1.83
27	40	72	0.96	1.88	1.83
45	60	108	0.92	1.84	1.83

4.12.2 Shakedown Load For A Strip Footing

The application of the kinematic shakedown formulation as described before is illustrated by the case of a strip footing underlaid by a shallow stratum of clay under undrained condition. The loading magnitude is assumed to vary repeatedly downwards between zero and some prescribed value and is defined here by a dimensionless parameter, F, equal to the ratio of the applied load per unit area to the undrained shear strength (i.e. $F = \frac{W}{As_u}$ where W = load, A = area and s_u = undrained shear strength of the soil).

The soil medium is assumed to be homogeneous and isotropic. The finite element mesh of the strip footing is shown in Fig. 4.7. It consists of 45 nodes and 32 four noded rectangular isoparametric elements. The ratio of the footing width to the depth of the stratum is 0.80. Due to symmetry, only half of the footing geometry and the mesh is shown. Vertical side boundary is placed at a distance eight and a half times the width of the footing and assumed to be smooth. The bottom boundary is assumed to be rough and rigid. The footing is assumed to deform under plane strain condition. Reduced integration technique is used to ensure the nearly incompressibility of the soil medium.

In the undrained analysis the soil is modelled as a weightless two-phase material with s_u = 100 KPa and $\phi_u = 0$, where s_u = undrained shear strength and ϕ_u = angle of internal friction. The drained elastic Lame's parameters $\lambda = 150\ s_u$, $\mu = 100\ s_u$ and drained Poisson's ratio, $\nu_d = 0.30$ are assumed for the soil skeleton whereas the bulk modulus for pore water is assigned as K_w = 2000 MPa. This gives an equivalent undrained Poisson's ratio, $\nu_u \cong 0.499$ approximately ensuring the nearly incompressibility of the soil medium. The soil behaviour is assumed to be elastic-perfectly plastic and obey the Von Mises yield condition. Six linearized planes are used here to describe the yield condition.

The repeated vertical pressure is applied at nodes 1, 6 and 11 (refer Fig. 4.7). The stress fields comprising of σ_x, σ_y, σ_z and τ_{xy} for all the elements are obtained from which the elastic limit pressure is computed by observing the pressure at which any one of the elements becomes first plastic. The shakedown limit pressure is obtained by first normalizing the elastic stress responses for all

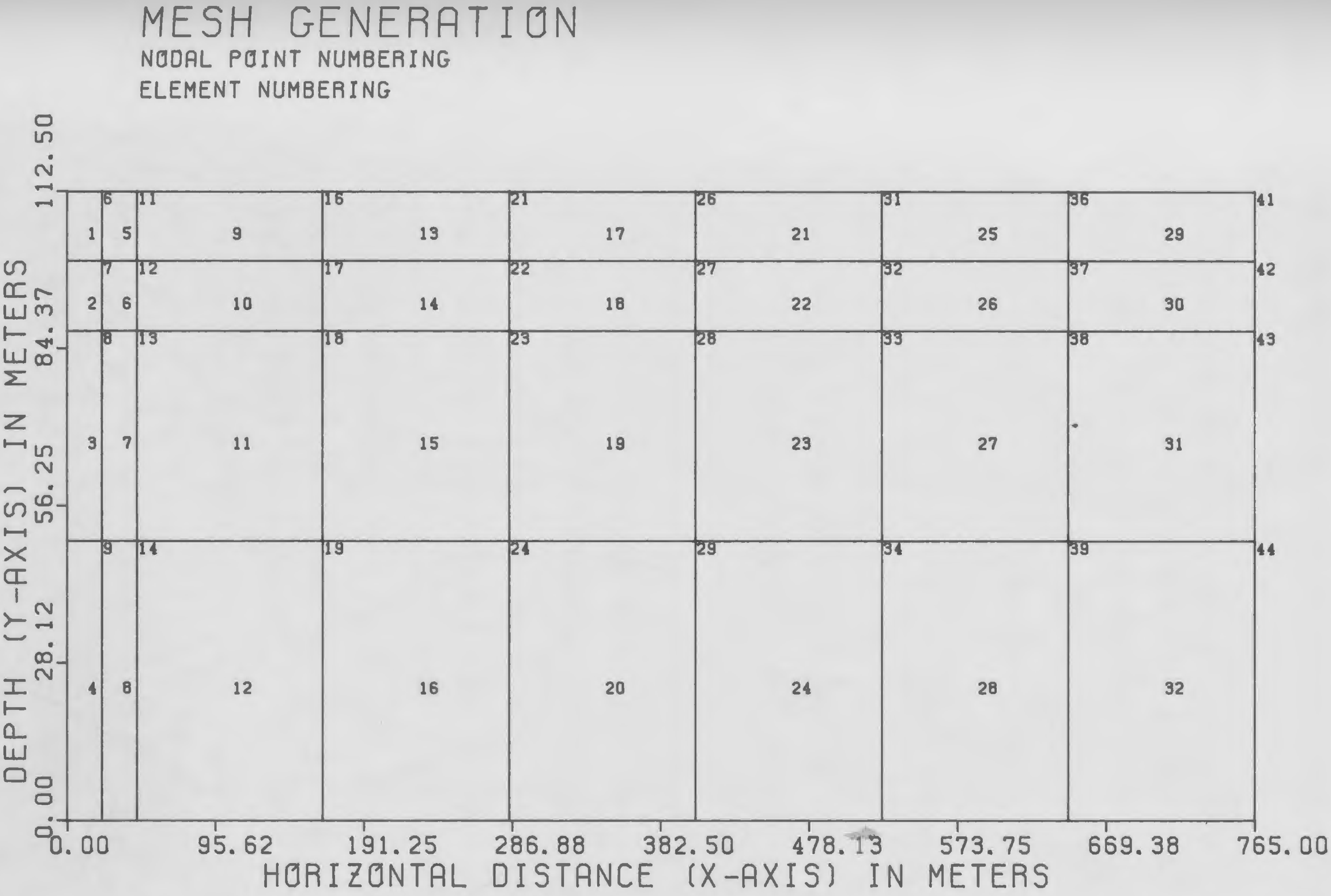


FIG. 4.7 FINITE ELEMENT MESH FOR STRIP FOUNDATION (SHAKEDOWN ANALYSIS)

elements following Eqn. (4.43) and then solving the mathematical programming problem as given by Eqn. (4.55) using the maxima of these elastic responses. Total number of constraints and variables for this problem are 97 and 320 respectively. The typical CPU time required is 20 minutes and the major portion of this time is spent in solving the programming problem.

Table 4.2 shows the elastic and shakedown limit pressures in dimensionless form. Because of the one parameter loading situation, the shakedown pressure, F^s, will be equal to the collapse load, F^c, which is given theoretically as $F^c = (2+\Pi)s_u$ in which F^c is the failure pressure (Hill, 1950). For a strip footing the above pressure is a true load because the applications of the two limit theorems (e.g. upper bound and lower bound) provide the same results. However, the shakedown pressure computed from Eqn. (4.55) will give an upper bound solution which is 4.65% higher than the theoretical collapse pressure. The elastic limit pressure, F^e, is only 4.50% higher than the theoretical value (Jumikis, 1969).

The strip footing problem is also run with a free lateral boundary and smooth rigid bottom. In this case the first yield occurs at a lower applied pressure because of the reduction in horizontal normal stress. The computed value of the first yield in dimensionless form is 2.18 (Table 4.3). The shakedown pressure shown in Table 4.3 is 2.42. Hill (1950) computed the ultimate load for the same problem as 2.57. Therefore, the comparison shows excellent agreement with the published values thus ensuring the accuracy of the computer programme.

TABLE 4.2 - COMPUTATION OF ELASTIC AND SHAKEDOWN LIMIT PRESSURES FOR A STRIP FOOTING UNDER VARIABLE REPEATED LOADING

$\frac{\text{Footing Width}}{\text{Depth of Foundation}}$	Number of Elements	Number of Nodal Points	Number of Equations	Elastic Limit Pressure (F^e/s_u)	Shakedown Limit Pressure (F^s/s_u)
0.80	32	45	64	3.28	5.38

TABLE 4.3 - COMPUTATION OF ELASTIC AND SHAKEDOWN PRESSURES FOR A STRIP FOOTING (HILL'S PROBLEM, 1950)

$\frac{\text{Footing Width}}{\text{Depth of Foundation}}$	Number of Elements	Number of Nodal Points	Number of Equations	Elastic Limit Pressure (F^e/s_u)	Shakedown Limit Pressure (F^s/s_u)
0.80	32	45	77	2.18	2.42

CHAPTER V

DYNAMIC RESPONSE ANALYSIS USING NONLINEAR SOIL MODELS

5.1 Introduction

This chapter presents the formulation of three different types of soil constitutive relationships which will be used in the dynamic response analysis of a gravity type offshore foundation. Research in the field of earthquake engineering has shown that the soil deformation can vary from small to very large amplitude depending on the nature of the dynamic loading. Cyclic loading causes permanent deformation and thus strain, under both drained and undrained conditions. The permanent strains are cumulative in nature i.e. carried over from one cycle to another and remain even if the cyclic loading has been terminated. The amount of straining depends primarily on the soil type, average effective confining pressure, void ratio, degree of saturation, frequency of vibration, over consolidation ratio (OCR), etc;

The chapter is divided in three sections. The first section describes the formulation of the three types of soil modellings: These are a) elastic b) equivalent linear and c) incremental nonlinear elastic-plastic. Although the equivalent linear model is quite popular in the field of earthquake engineering, it fails to provide any information on the permanent deformation which is of great concern to the designer of an offshore platform. Nevertheless, the analysis can still provide some valuable information on strain compatible soil properties which will provide a realistic soil stress distribution. On the other hand, the incremental elastic-plastic analysis coupled with direct time step integration approach is capable of predicting the permanent deformation history. However, in an incremental stress-strain relationship, the solution has to proceed in small time steps thus limiting the analysis to a very few number of cycles.

The second section presents the incremental formulation of dynamic equations of motion in time domain. A computer code has been developed for the dynamic response analysis of an offshore gravity foundation based on the above soil modellings. An example problem is illustrated with particular reference to the prediction of permanent deformation of a strip footing under repeated vertical loading condition.

5.2 Soil Models

The mechanical behaviour of saturated soils leads to very complicated stress-strain relationship. Therefore, the successful modelling must at least consider the two-phase characteristics where the behaviour is primarily affected by the hydrostatic pressure, fully controlled by the effective stresses, the applied total stress and its past history. The undrained behaviour of the soil-foundation system is given by Eqn. (3.38j) where, $[K_{ss}]$, represents the undrained stiffness matrix and expressed as

$$K_{ss} = \int_V \Psi_{11}{}^T \; D_{sd} \; \Psi_{11} \; dV + \int_V \alpha^2 M_o \; \Psi_{22}{}^T \; \Psi_{22} \; dV$$

If the stress-strain relationship cannot be expressed uniquely and is somewhat dependent on loading history or strain, the constitutive matrix, $[D_{sd}]$, must be updated in order to make it either strain compatible for equivalent linear model or function of existing state of stress incremental nonlinear model. However, for an elastic model, $[D_{sd}]$, remains constant during the analysis.

5.2.1 Linear Elastic Model

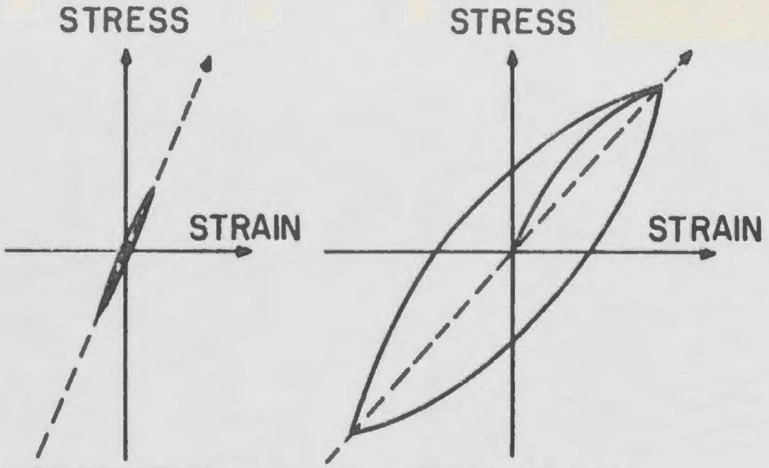
Among the soil models proposed in the literature, the elastic model is the simplest, where the stress-strain relations are reversible and have no time-delay. The elastic model still can provide some useful results, which gives the starting point of developing other more complex models. The elastic response analysis requires the elastic properties of the soil and structure e.g. usually defined by Lame's constants, as the input.

5.2.2 Equivalent Linear Model

The equivalent linear model is used here to study the dynamic response of the soil-foundation system to wave excitation. The method was originally developed for the dynamic response analysis of soil deposits to seismic excitation. The basic assumptions which were made in formulating the equivalent linear model were: a) seismic excitation is primarily due to shear waves propagating vertically and b) the strong nonlinear characteristics of the soil may be modelled by a viscously damped linear oscillator whose properties such as stiffness and damping are strain dependent.

Extensive experimental research on dynamic soil properties has shown that dynamic soil properties such as shear modulus and damping are strain dependent. The shear modulus decreases with increasing strain amplitude whereas damping increases with the strain amplitude of the loading as shown in Fig. 5.1.

From the extensive series of reversed torsional tests on cohesive and cohesionless soils and the information already available in the literature, it is reasonable to assume that a modified hyperbolic curve will satisfactorily represent the shearing stress-shearing strain relationship throughout the range of strain amplitude up to failure. Fig. 5.2 shows the factor which governs the basic hyperbolic shearing stress-strain curve. At zero shearing strain, the tangent to the curve represents the maximum shear modulus, G_{max}. The secant modulus at any point along the curve for example, point A is designed as, G, and the maximum shearing stress as, τ_{max}, (determined from a simple shear test). The horizontal line at the ordinate of, τ_{max}, is the second asymptote to the hyperbolic curve as shown in Fig. 5.2. G_{max} and τ_{max} can be obtained from field and/or laboratory tests.



LOWER STRAIN
LOWER DAMPING
HIGHER MODULUS

HIGHER STRAIN
HIGHER DAMPING
LOWER MODULUS

FIG. 5.1 ILLUSTRATION OF STRAIN DEPENDENCY OF MODULI AND DAMPING IN SOILS (SEED AND IDRISS, 1969)

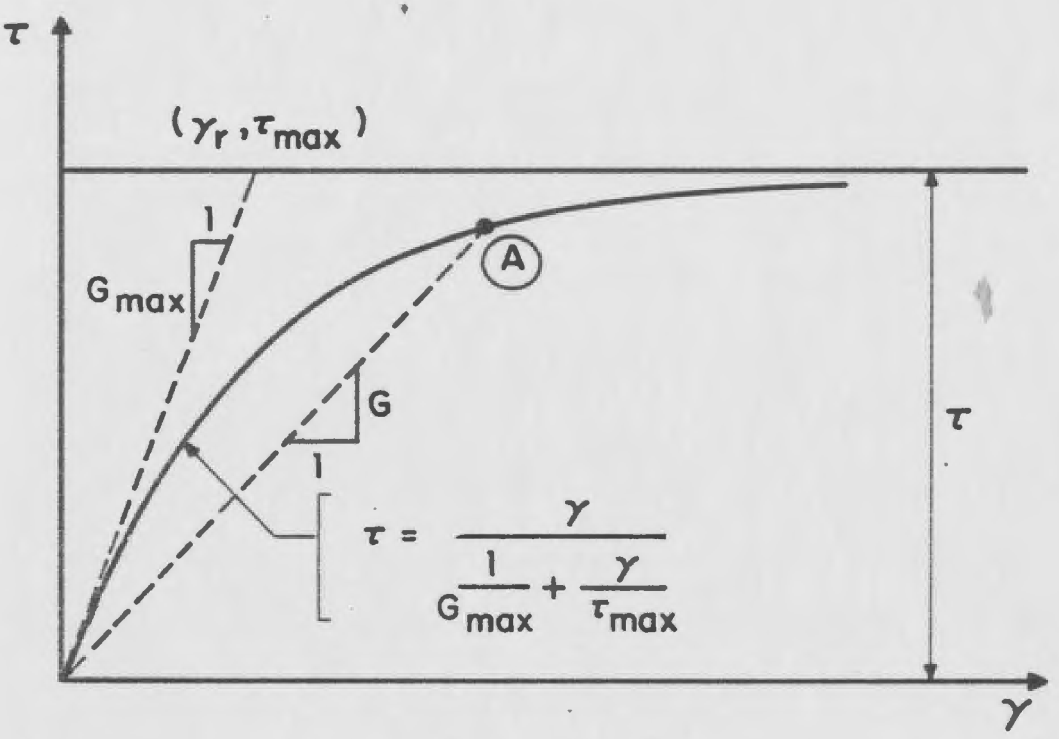


FIG. 5.2 BASIC PARAMETERS FOR HYPERBOLIC SHEARING STRESS-SHEARING STRAIN CURVES (SEED AND IDRISS, 1969)

5.2.2.1 Variable Stiffness and Damping Solution

The basic procedure adopted in an equivalent linear model is that the dynamic response of the soil-foundation system may be approximated sufficiently by a damped linear viscous oscillator if the properties of this oscillator (e.g. stiffness and damping) are chosen properly. An iterative method is normally used in order to derive these properties. While using this method in the finite element analysis, the stress-strain characteristics of the soil are assigned in each element with the following four properties; a) maximum shear modulus, G_{max}, defined as the shear modulus at low strain 10^{-4}% level b) maximum damping, D_{max}, (= 29%) c) Poisson's ratio and d) estimated shear modulus, G, for the first iteration. The average in-situ shear moduli for saturated clays are shown in Fig. 5.3. These curves are also tabulated in Table 5.1 and are shown in Fig. 5.4 in terms of G/G_{max} and damping ratios, D.

In the first cycle of the analysis, initial values of shear moduli and damping ratios are chosen and a dynamic linear elastic response analysis is performed for the entire time

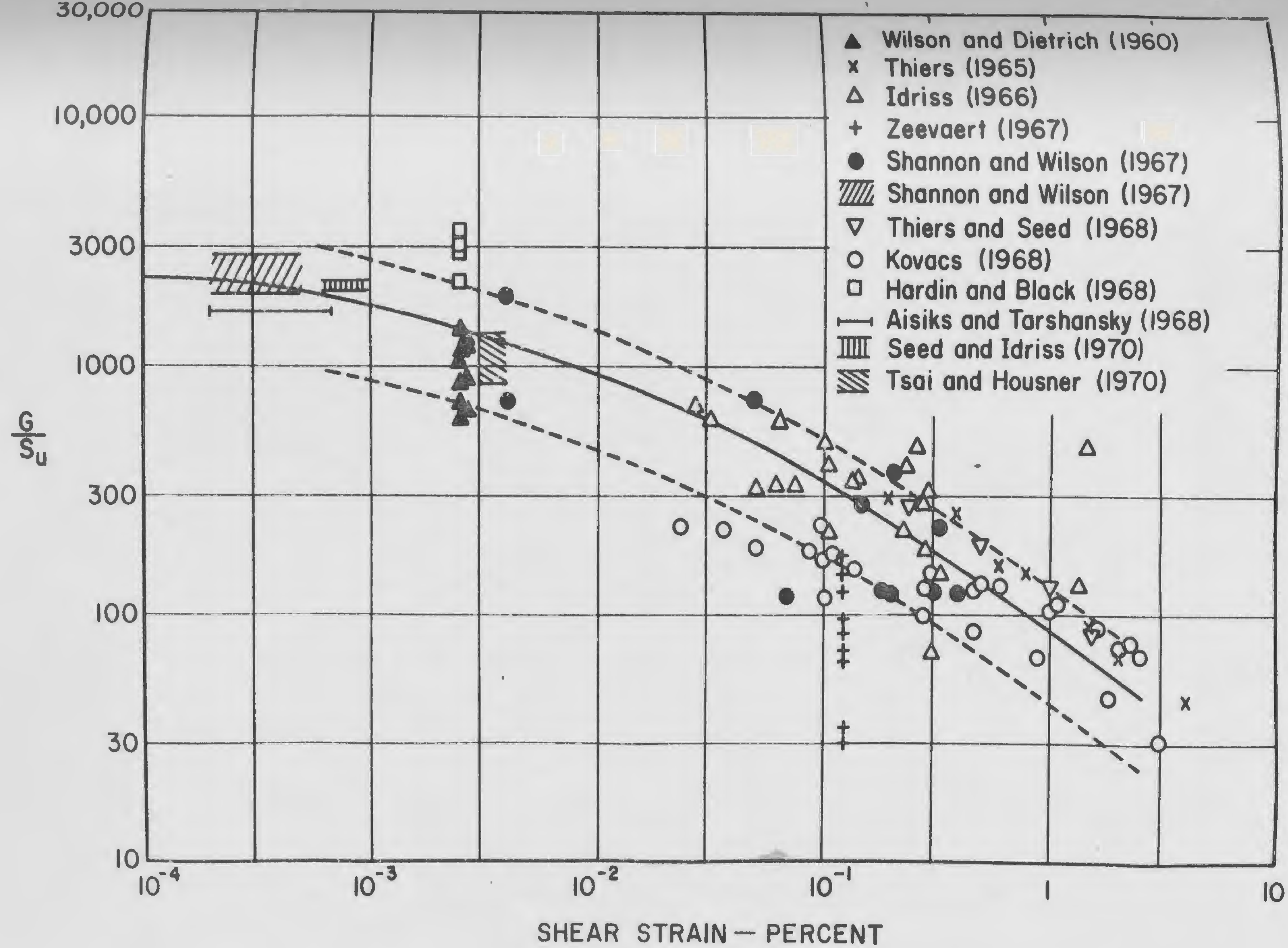


FIG. 5.3 IN-SITU SHEAR MODULI FOR SATURATED CLAYS
(SEED AND IDRISS, 1969)

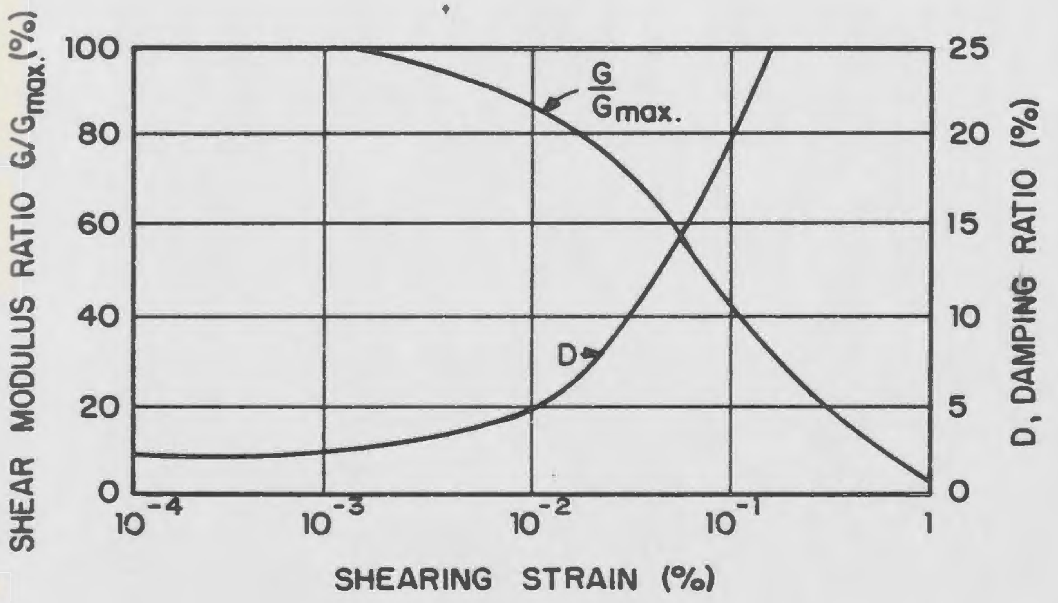


FIG. 5.4 MODULUS AND DAMPING RATIOS USED FOR DYNAMIC RESPONSE ANALYSIS (SEED AND IDRISS, 1969)

TABLE 5.1 - STRAIN-COMPATIBLE SOIL PROPERTIES

(SEED AND IDRISS, 1969)

Effective Shear Strain (%)	log (γ_{eff})	Shear Modulus Reduction Factor*	Fraction of Critical Damping (%)
		Clay	Clay
$1. \times 10^{-4}$	-4.0	1.000	2.50
3.16×10^{-4}	-3.5	0.913	2.50
1.00×10^{-3}	-3.0	0.761	2.50
3.16×10^{-3}	-2.5	0.565	3.50
1.00×10^{-2}	-2.0	0.400	4.75
3.16×10^{-2}	-1.5	0.261	6.50
1.00×10^{-1}	-1.0	0.152	9.25
0.316	-0.5	0.076	13.8
1.00	0.	0.037	20.0
3.16	0.5	0.013	26.0
10.00	1.0	0.004	29.0

* This is the factor which has to be applied to the shear modulus at low shear strain amplitudes (here defined as 10^{-4} percent) to obtain the modulus at higher strain levels.

histories of the wave force which may contain several number of cycles. The maximum shear strain, γ_{max}, is selected for each element in order to compute the new strain compatible shear modulus and damping ratio. Having computed the new value of shear modulus and damping ratio for each element depending on the maximum shear strain level, a second analysis is carried out for the next iteration. The procedure is repeated until no significant changes in moduli or damping ratios are observed for all elements. The final response obtained from the last iteration is assumed to be a reasonable approximation for the true nonlinear response.

In the variable stiffness and damping solution, the stiffness and damping submatrices are formed for each element and then all element submatrices are assembled in global form in a conventional manner. The global stiffness and mass matrices are computed and an eigenvalue analysis is performed. In order to compute the damping submatrix of an element, j, the Rayleigh type damping is assumed. For an element, j,

$$[c]_j = B_1[m]_j + B_2[\bar{k}]_j \qquad \text{... (5.1)}$$

in which $[c]_j$, $[m]_j$ and $[\bar{k}]_j$ are the damping, mass and stiffness submatrices respectively and B_1 and B_2 are parameters which are functions of the damping ratios and stiffness characteristics. The Rayleigh parameters B_1 and B_2 are expressed as

$$B_1 = D_j \omega_1$$

$$\text{and} \qquad B_2 = D_j/\omega_1 \qquad \ldots (5.2)$$

where, D_j, represents the strain dependent damping ratio for element j, obtained from Fig. 5.4 and Table 5.1 and, ω_1, represents the fundamental frequency of the soil-foundation system. $[m]$ and $[\bar{k}]$ are given by Eqn. (3.36).

The global damping matrix, C^*, for the entire assemblage is formed by appropriate addition of the damping submatrices of all finite elements in the discretized model. Thus if, c^j_{kl}, represents (kl)-th term of the damping submatrix $[c]_j$ of a typical element j, the kl-th term of the damping matrix of the entire system of M-elements is given by

$$C^*_{kl} = \sum_{j=1}^{M} c^j_{kl} \qquad \ldots (5.3)$$

The resulting damping matrix is symmetric and normally sparsely populated. The response (displacement) from Eqn. (3.48) is obtained by a step-by-step integration procedure described in the next section. Haldar et al (1980) have used the equivalent linear model in the dynamic response analysis of a submarine pipe line to seismic excitation and observed the significant effect on the stresses in the porous bed around the pipe.

5.2.3 Elastic Plastic Model

The primary assumption of the incremental elastic-plastic soil model is that the total strain increments are separable into elastic and plastic components, in which the elastic part is linearly related to the stress increment. The plastic component of the strain increment is defined by means of a plastic potential, g, and yield functions, f_y. The stress level at which plastic strain develops is defined by the yield function whereas the plastic potential function defines the distribution of the plastic strain increment during the plastic flow. The magnitude of the plastic strain is determined by a particular work-hardening relationship. If the yield surface is defined by a continuous differentiable function, f_y, the three possible loading states, which will be encountered during an incremental load path, are as follows:

1) During the loading state the stress increments are directed outwards from the yield surface and as such

$$\frac{\partial f_y}{\partial \sigma_{ij}} d\sigma_{ij} > 0 \qquad \ldots (5.4a)$$

2) During the neutral loading state, the stress increments will remain tangent to the yield surface and as such

$$\frac{\partial f_y}{\partial \sigma_{ij}} d\sigma_{ij} = 0 \qquad \ldots (5.4b)$$

and

3) During the unloading state, the stress increments are directed inwards and as such

$$\frac{\partial f_y}{\partial \sigma_{ij}} d\sigma_{ij} \leq 0 \qquad \ldots (5.4c)$$

In the associative flow theory of plasticity, the yield function, f_y, is assumed to coincide with the plastic potential function, g. In deriving the elastic-plastic stress-strain relationship in an incremental form, a single continuous yield function, f_y, is assumed and the total strain, $d\varepsilon_{ij}$, given by

$$d\varepsilon_{ij} = d\varepsilon^E_{ij} + d\varepsilon^p_{ij} \qquad \ldots (5.5)$$

where

$d\varepsilon^E_{ij}$ = elastic strain component
$d\varepsilon^p_{ij}$ = plastic strain component
and $d\varepsilon_{ij}$ = total strain

The strain increments for three dimensional stress state is represented in vector form as

$$\{d\varepsilon_{ij}\}^T = \langle d\varepsilon_x \quad d\varepsilon_y \quad d\varepsilon_z \quad d\gamma_{xy} \quad d\gamma_{yz} \quad d\gamma_{zx} \rangle \qquad \ldots (5.6a)$$

and in two dimensional stress state

$$\{d\varepsilon_{ij}\}^T = \langle d\varepsilon_x \quad d\varepsilon_y \quad d\varepsilon_z \quad d\gamma_{xy} \rangle \qquad \ldots (5.6b)$$

5.2.3.1 Elastic Strain Increment

The elastic strain increments are related to the effective stress increments as

$$\{d\sigma'_{ij}\} = \left[D_{sd}\right] \{d\varepsilon^E_{ij}\} \qquad \ldots (5.7)$$

where $\{d\sigma'_{ij}\}$ = effective stress increment vector

and

$[D_{sd}]$ = drained elastic stress-strain matrix as defined by Eqn. (3.38i)

5.2.3.2 Plastic Strain Increment

The initial and subsequent yield condition for any work hardening model can be written as

$$f_y\ (\sigma'_{ij},\ \varepsilon^p_{ij},\ b) = 0 \qquad \ldots\ (5.8)$$

in which, σ'_{ij}, ε^p_{ij}, and b are the effective stress components, the plastic strain components and a material constant respectively. The constant, b, is a function of effective plastic strain, $\bar{e}^p$.

The flow rule, which relates the relative magnitude of the plastic strain increments to the stresses, is

$$d\varepsilon^p_{ij} = \Delta\ \frac{\partial f_y}{\partial \sigma'_{ij}} \qquad \ldots\ (5.9)$$

and the equivalent plastic strain

$$d\bar{\varepsilon}^p = \Delta\sqrt{d\varepsilon^p_{ij} \cdot d\varepsilon^p_{ij}} \qquad \ldots (5.10)$$

The vector of stress gradient is expressed as

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T = \left[\frac{\partial f_y}{\partial \sigma'_x} \quad \frac{\partial f_y}{\partial \sigma'_y} \quad \frac{\partial f_y}{\partial \sigma'_z} \quad \frac{\partial f_y}{\partial \tau_{xy}}\right] \qquad \ldots (5.11)$$

5.2.3.3 Work Hardening Law

The work hardening law determines the actual magnitude of the plastic strain increment caused by a given stress increment. Therefore, the work hardening law is defined here as the relationship between the effective stress level and the plastic work done per unit volume and expressed as a function of equivalent plastic work as, $Q(\varepsilon^p_{ij})$ and in matrix form.

$$Q(\varepsilon^p_{ij}) = \left\{\sigma'_{ij}\right\}^T \left\{\varepsilon^p_{ij}\right\} \qquad \ldots (5.12)$$

5.2.3.4 Incremental Formulation

The incremental method assumes that during yielding, the function, f_y, should satisfy the consistency condition and therefore

$$df_y = 0$$

or

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \left\{d\sigma'_{ij}\right\} + \left\{\frac{\partial f_y}{\partial \varepsilon^p_{ij}}\right\}^T \left\{d\varepsilon^p_{ij}\right\} + \frac{\partial f_y}{d\bar{e}^p}\, d\bar{e}^p = 0 \qquad \ldots (5.13)$$

where

$$\left\{\frac{\partial f_y}{\partial \varepsilon^p_{ij}}\right\}^T = \left[\frac{\partial f_y}{\partial \varepsilon^p_x} \quad \frac{\partial f_y}{\partial \varepsilon^p_y} \quad \frac{\partial f_y}{\partial \varepsilon^p_z} \quad \frac{\partial f_y}{\partial \gamma^p_{xy}}\right] \qquad \ldots (5.14)$$

Using the elastic strain to effective stress relationship as defined by Eqn. (5.7), Eqn. (5.13) becomes

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \left[D_{sd}\right] \left\{d\varepsilon^E_{ij}\right\} + \left\{\frac{\partial f_y}{\partial \varepsilon^p_{ij}}\right\}^T \left\{d\varepsilon^p_{ij}\right\} + Y = 0 \qquad \ldots (5.15)$$

where $Y = \dfrac{\partial f_y}{\partial \bar{e}^p}\, d\bar{e}^p$

Substituting Eqn. (5.5) in Eqn. (5.15), one obtains

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \left[D_{sd}\right] \left(\left\{d\varepsilon_{ij}\right\} - \left\{d\varepsilon^p_{ij}\right\}\right) + \left\{\frac{\partial f_y}{\partial \varepsilon^p_{ij}}\right\}^T \left\{d\varepsilon^p_{ij}\right\} + Y = 0 \qquad \ldots (5.16)$$

Substituting the flow rule from Eqn. (5.9) into Eqn. (5.16), one obtains

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \left[D_{sd}\right] \left(\{d\varepsilon_{ij}\} - \Delta\{\frac{\partial f_y}{\partial \sigma'_{ij}}\}\right) + \left\{\frac{\partial f_y}{\partial \varepsilon^p_{ij}}\right\}^T \left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\} + Y = 0 \qquad \ldots (5.17)$$

Writing the Eqn. (5.17) in the following form, we get

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \left[D_{sd}\right] \left(\{d\varepsilon_{ij}\} - \Delta\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}\right) + H\Delta = 0 \qquad \ldots (5.18)$$

where

$$H = \left\{\frac{\partial f_y}{\partial \varepsilon^p_{ij}}\right\}^T \left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\} + \frac{\partial f_y}{\partial \bar{e}^p} \sqrt{d\varepsilon^p_{ij} \cdot d\varepsilon^p_{ij}} \qquad \ldots (5.19)$$

Solving for Δ from Eqn. (5.18), one obtains

$$\Delta = \frac{\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \left[D_{sd}\right] \{d\varepsilon_{ij}\}}{\left\{\frac{\partial f_y}{\partial \sigma_{ij}}\right\}^T \left[D_{sd}\right] \left\{\frac{\partial f_y}{\partial \sigma_{ij}}\right\} - H} \qquad \ldots (5.20)$$

Substituting the value of Δ in Eqn. (5.7), one obtains

$$\left\{d\sigma'_{ij}\right\} = \left[D_{sd}\right]\left(\left\{d\varepsilon_{ij}\right\} - \Delta\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}\right)$$

$$= \left(\left[D_{sd}\right] - \frac{\left[D_{sd}\right]\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T\left[D_{sd}\right]}{\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T\left[D_{sd}\right]\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\} - H}\right)\left\{d\varepsilon_{ij}\right\} \qquad \ldots (5.21)$$

The term in the bracket (.) is defined as the desired elastic plastic stress-strain matrix, which is symmetric because of the associative nature of flow rule. If, H, is zero, no work hardening is present and therefore the assumption of elastic-perfectly plastic model is valid. The strain increment is defined as

$$\left\{d\bar{\varepsilon}\right\} = \bar{\Delta}\left\{\frac{\partial f_y}{\partial \sigma'}\right\} \qquad \ldots (5.22)$$

where $\bar{\Delta}$ is arbitrary

and

$$\{d\sigma\} = \left[D_{sd}\right]\left\{d\bar{\varepsilon}\right\} - \left[D_{sd}\right]\left\{\frac{\partial f_y}{\partial \sigma'}\right\} L^{-1}\left\{\frac{\partial f_y}{\partial \sigma'}\right\}^T \times$$

$$\left[D_{sd}\right]\left\{\frac{\partial f_y}{\partial \sigma'}\right\}\bar{\Delta} \qquad \ldots (5.23a)$$

where

$$\left[L\right] = \left\{\frac{\partial f_y}{\partial \sigma'}\right\}^T \left[D_{sd}\right] \left\{\frac{\partial f_y}{\partial \sigma'}\right\} - H \qquad \ldots (5.23b)$$

If H = 0, one obtains

$$\left\{d\sigma\right\} = \left[D_{sd}\right]\left\{d\bar{\varepsilon}\right\} - \left[D_{sd}\right] \left[L\right]^{-1} \left[L\right] \left\{d\bar{\varepsilon}\right\} = 0$$

$$\ldots (5.24)$$

which implies that the plastic deformation will progress at constant stress level.

5.2.3.5 Drucker Prager Model

The Drucker Prager yield criteria is quite often used in the elastic-plastic finite element analysis and may be considered as a first attempt to approximate the well-known Mohr Coulomb failure criteria by a smooth surface (Fig. 5.5). The yield criteria is expressed in terms of the two stress invariants (first and second) together with two material constants a and b. The yield function is expressed as

$$f_y = a\, J_1 + \sqrt{J_2} - b = 0 \qquad \ldots (5.25a)$$

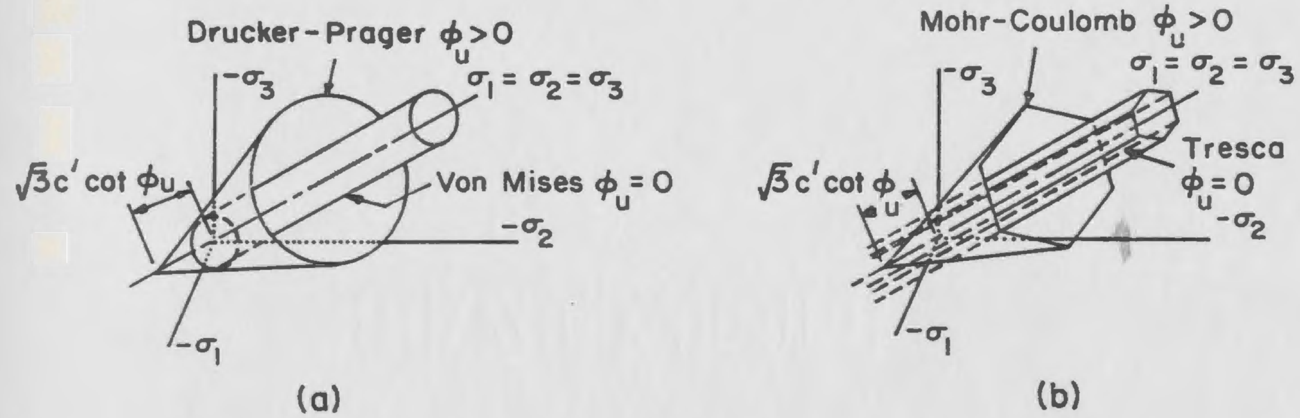


FIG. 5.5 TYPICAL YIELD FUNCTIONS IN GENERALISED STRESS-SPACE (ZIENKIEWICZ , 1977)

The constants a and b are called the material parameters which are related, in the case of plane strain, to the Coulomb's constants, c' and ϕ_u as

$$a = \frac{\sin^2\phi_u}{\sqrt{3\left(3 + \sin^2\phi_u\right)}} \qquad \ldots (5.25b)$$

and

$$b = \frac{\left(\sqrt{3}c' \cos\phi_u\right)^2}{3 + \sin^2\phi_u} \qquad \ldots (5.25c)$$

The characteristics of undrained clays satisfying 'zero volume change' condition can be modelled mathematically by Von Mises yield criteria. This model is only applicable to undrained clays which have a constant shear strength. The undrained shear strength of clay, s_u, is then simply related to second invariant of stress, J_2.

The equation of the Von Mises yield surface is expressed as

$$f_y = \sqrt{J_2} - b = 0 \qquad \ldots (5.25d)$$

The terms J_1 and J_2 are the first and second invariant of stresses and expressed as

$$J_1 = \sigma'_{ij} = (\sigma'_x + \sigma'_y + \sigma'_z) \qquad \ldots (5.26a)$$

$$J_2 = 1/2\ S_{ij}\ S_{ij} = 1/2\ (S_x{}^2 + S_y{}^2 + S_z{}^2) + \tau_{xy}{}^2$$

$$= 1/6\ \ (\sigma_x' - \sigma_y')^2 + (\sigma_y' - \sigma_z')^2 + (\sigma_z' - \sigma_x')^2 + 6\tau_{xy}{}^2 \qquad \ldots (5.26b)$$

where $S_x = \sigma_x' - \frac{J_1}{3}$

$$S_y = \sigma_y' - \frac{J_1}{3} \qquad \ldots (5.26c)$$

and $S_z = \sigma_z' - \frac{J_1}{3}$

The stress tensor, σ_{ij}', can then be expressed in terms of J_1 and S_{ij} as

$$\sigma_{ij}' = 1/3\ J_1\ \delta_{ij} + S_{ij} = \sigma_{mm}' + S_{ij} \qquad \ldots (5.26d)$$

where, δ_{ij}, is Kronecker delta. The mean hydrostatic pressure, σ_{mm}', causes volumetric change whereas the deviatoric stress, S_{ij}, causes shape change in a soil element.

For perfect plasticity, b, is constant and can be expressed uniquely. However, for isotropic hardening, b, is not constant and is a function of accumulated increments of effective plastic strain and expressed as

$$b = b\,(\bar{e}^p) \qquad \ldots (5.27)$$

where $\bar{e}^p$ = effective plastic strain

For yielding under an uniaxial state of stress on a soil sample in vertical direction, σ_y, is active and all other stress components are zero. The yield function given by Eqn. (5.25a) becomes

$$a\sigma_y' + \frac{\sigma_y'}{\sqrt{3}} - b = 0$$

or

$$b = \left(a + \frac{1}{\sqrt{3}}\right)\sigma_y' \qquad \ldots (5.28)$$

Substituting Eqn. (5.28) in Eqn. (5.25a), the effective stress, $\bar{\sigma}$, is expressed as

$$\bar{\sigma} = \frac{aJ_1 + \sqrt{J_2}}{a + \frac{1}{\sqrt{3}}} \qquad \ldots (5.29)$$

The yielding occurs when $\bar{\sigma} = \sigma_y'$. By putting a = 0, one obtains the Von Mises yield condition and the effective stress.

The incremental effective plastic work is given by

$$dQ(\varepsilon^p_{ij}) = \left\{\sigma'_{ij}\right\}^T \; d\varepsilon^p_{ij}$$

$$= \bar{\sigma}\; d\bar{\varepsilon}^p \qquad \ldots (5.30)$$

Using Eqn. (5.25) and Eqn. (5.26), it can be shown that

$$\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\} = \frac{\partial f_y}{\partial J_1} \cdot \left\{\frac{\partial J_1}{\partial \sigma'_{ij}}\right\} + \frac{\partial f_y}{\partial J_2} \cdot \left\{\frac{\partial J_2}{\partial \sigma'_{ij}}\right\}$$

$$= a\delta_{ij} + S_{ij}/2\sqrt{J_2} \qquad \ldots (5.31)$$

Substituting Eqn. (5.9), (5.29) and (5.31) in Eqn. (5.30) and using the relationship

$$\sigma'_{ij}\, S_{ij} = (S_{ij} + \frac{\sigma'_{mm}}{3}\,\delta_{ij})\, S_{ij} = 2J_2 \qquad \ldots (5.32)$$

one obtains

$$dQ(\varepsilon^p_{ij}) = \bar{\sigma}\; d\bar{\varepsilon}^p = \Delta\left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\}^T \sigma'_{ij}$$

$$= \Delta\left(a\delta_{ij} + \frac{S_{ij}}{2\sqrt{J_2}}\right)\sigma'_{ij}$$

$$= \Delta\left(aJ_1 + \sqrt{J_2}\right)$$

$$= \Delta\left(a + \frac{1}{\sqrt{3}}\;\bar{\sigma}\right) \qquad \ldots (5.33)$$

From Eqn. (5.9) one may write

$$\left\{d\,\varepsilon^{p}_{ij}\right\}\cdot\left\{d\,\varepsilon^{p}_{ij}\right\} = \Delta^{2}\left\{\frac{\partial f_{y}}{\partial\sigma'_{ij}}\right\}\cdot\left\{\frac{\partial f_{y}}{\partial\sigma'_{ij}}\right\}$$

$$= \Delta^{2}\ (3a^{2} + 1/2) \qquad \ldots (5.34a)$$

Therefore

$$\sqrt{\left\{d\varepsilon^{p}_{ij}\right\}\cdot\left\{d\varepsilon^{p}_{ij}\right\}} = \Delta\sqrt{(3a^{2} + 1/2)} \qquad \ldots (5.34b)$$

From Eqn. (5.33)

$d\bar{e}^{p} = \Delta\left(a + \frac{1}{\sqrt{3}}\right)$ and using Eqn. (5.34b) one obtains

$$d\bar{e}^{p} = \sqrt{d\varepsilon^{p}_{ij}\ .\ d\varepsilon^{p}_{ij}}\ \frac{\left(a + \frac{1}{\sqrt{3}}\right)}{\sqrt{3a^{2} + 1/2}} \qquad \ldots (5.34c)$$

Finally if a = 0, one obtains the Von Mises effective plastic strain increment from Eqn. (5.10).

The elastic plastic constitutive relationship expressed by Eqn. (5.21) can be presented after using Eqn. (5.31) and Eqn. (5.32) as follows

$$\left\{d\sigma'_{ij}\right\} = \left[D_{sd}\right]\left(1 - \frac{\left[a\delta_{ij} + \frac{S_{ij}}{2\ J_{2}}\right]\left[a\delta_{ij} + \frac{S_{ij}}{2\ J_{2}}\right]^{T}\left[D_{sd}\right]}{\left[a\delta_{ij} + \frac{S_{ij}}{2\ J_{2}}\right]^{T}\left[D_{sd}\right]\left[a\delta_{ij} + \frac{S_{ij}}{2\ J_{2}}\right] - H}\right)\left\{d\varepsilon_{ij}\right\}$$

$$= \left[D^{Ep}_{sd}\right]\left\{d\varepsilon_{ij}\right\} \qquad \ldots (5.35)$$

where $[D_{sd}^{Ep}]$ is the desired elastic-plastic constitutive matrix based on effective stress vector.

From Eqn. (5.19)

$$H = \left\{\frac{\partial f_y}{\partial \varepsilon_{ij}^p}\right\}^T \left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\} + \frac{\partial f_y}{\partial \bar{e}^p} \sqrt{d\varepsilon_{ij}^p \cdot d\varepsilon_{ij}^p}$$

$$= \left\{\frac{\partial f_y}{\partial \varepsilon_{ij}^p}\right\}^T \left\{\frac{\partial f_y}{\partial \sigma'_{ij}}\right\} + \frac{\partial f_y}{\partial b} \cdot \frac{\partial b}{\partial \bar{e}^p} \qquad \ldots (5.36)$$

For elastic-perfectly plastic material

$$\frac{\partial b}{\partial \bar{e}^p} = 0 \text{ and } \frac{\partial f_y}{\partial \varepsilon_{ij}^p} = 0 \text{ and hence } H = 0$$

5.2.3.6 Undrained Elastic-Plastic Stress-Strain Relationship

For the solution of general boundary value problem under undrained condition, the total stress is expressed in terms of effective stress, σ'_{ij}, and pore pressure, p, from Eqn. (3.6e) as

$$\sigma_{ij} = \sigma'_{ij} + \delta_{ij}\alpha p \qquad \ldots (5.37)$$

Using Eqn. (5.35) and Eqn. (3.6e) under undrained condition, $(\xi = 0)$ one can write,

$$\left\{\dot{\sigma}_{ij}\right\} = \left[D^{Ep}_{sd}\right] \left\{\dot{\varepsilon}_{ij}\right\} + \alpha^2 M_o \left\{\dot{\varepsilon}_{ij}\right\} \qquad \ldots (5.38)$$

where $[D^{Ep}_{sd}]$ is defined as solid skeleton constitutive matrix. It has been pointed out before that the constitutive matrix, $[D^{Ep}_{sd}]$, is responsible for all deformations due to material nonlinearity. The elasto-plastic form of the $[D^{Ep}_{sd}]$ matrix is given by Eqn. (5.38) and after some algebric manipulation one can obtain (Appendix B)

$$\left\{\dot{\sigma}_{ij}\right\} = \begin{bmatrix} \lambda+2\mu+\alpha^2 M_o - \frac{E_{11}^2}{L} & \lambda+\alpha^2 M_o - \frac{E_{11}\cdot E_{22}}{L} & \lambda - \frac{E_{11}\cdot E_{33}}{L} & - \frac{E_{11}\cdot E_{12}}{L} \\ & \lambda+2\mu+\alpha^2 M_o - \frac{E_{22}^2}{L} & \lambda - \frac{E_{22}\cdot E_{33}}{L} & - \frac{E_{22}\cdot E_{12}}{L} \\ \text{SYM.} & & \lambda+2\mu+\alpha^2 M_o - \frac{E_{33}^2}{L} & - \frac{E_{12}\cdot E_{33}}{L} \\ & & & \mu - \frac{E_{12}^2}{L} \end{bmatrix} \left\{\dot{\varepsilon}_{ij}\right\}$$

$$\ldots (5.39)$$

where

$$a = \frac{\partial f_y}{\partial J_1}$$

$$B = \frac{\partial f_y}{\partial J_2} = \frac{\mu}{2\sqrt{J_2}}$$

$$L = 3a^2 (3\lambda + 2\mu) + \mu$$

$$E_{11} = a (3\lambda + 2\mu) + 2 BS_x$$

$$E_{12} = 2 B\tau_{xy}$$

$$E_{22} = a (3\lambda + 2\mu) + 2 BS_y$$

$$E_{33} = a (3\lambda + 2\mu) + 2 BS_z \quad \ldots (5.40)$$

5.3 Solution for Equations of Motion

The general dynamic equations of motion of a gravity type offshore structure-foundation system resting on saturated soils can be written in a global assembled form as

$$[M^*] \{\ddot{\underline{U}}\} + [C^*] \{\dot{\underline{U}}\} + [K^*] \{\underline{U}\} = \{P_B\} \quad \ldots (5.41)$$

where, [M*], [C*] and [K*] are the assembled global mass, damping and stiffness matrices respectively. $\{P_B\}$ is the global load vector which takes into account the effect of wave loading.

For, linear systems, Eqn. (5.41) can be solved by any one of the following methods.

a. Modal analysis or modal superpositon.

b. Direct integration.

c. Fourier analysis or complex response.

In linear analysis the response can be computed either in the time domain or in the frequency domain, but for a true nonlinear analysis (geometric or material) time domain is the most suitable one.

5.3.1 Equations of Motion in Incremental Form

In the case of a nonlinear system, the incremental form of the equations of motion can be expressed in the following form for time, $t+\tau$, as:

$$[K^*]\,\{\Delta \underline{U}\} = \{P_B\}_{t+\tau} - \{\tilde{R}\}_t - [M^*]\,\{\ddot{\underline{U}}\}_{t+\tau} - [C^*]\,\{\dot{\underline{U}}\}_{t+\tau} \quad \ldots (5.42a)$$

where

$$\{\Delta \underline{U}\} = \{\underline{U}\}_{t+\tau} - \{\underline{U}\}_t \quad \ldots (5.42b)$$

= incremental displacement vector

$$R_t = \sum^{M} \int_V \left[\underset{\approx}{\psi}\right]^T \left\{\sigma\right\}_t \, dV \qquad \ldots (5.42c)$$

= residual load vector computed from the stresses vector $(\sigma)_t$, at time t for all elements and $\left[\underset{\approx}{\psi}\right]^T$ obtained from Eqn. (3.25).

and

τ = increment of time step.

Several methods are available for solving the above equations of motion using direct integration scheme. In this investigation, an implicit integration scheme is used where the solution is advanced by one time step interval, Δt. The values of the displacement and its derivatives at one instant of time is then used to determine the values at subsequent time steps by means of recurrence relationships.

5.3.1.1 Wilson - Θ Method

In this method, the integration is advanced by first linearly interpolating to a hypothetical time step increment, $\tau = \Theta \Delta t$, $(\Theta = 1.37)$ and thus obtaining the displacement and its derivatives at $\tau = \Theta \Delta t$, time point. Later the displacement, and its derivatives at $t = t + \Delta t$ are computed by the averaging procedure.

The linear acceleration method assumes that the velocities and displacements are related to the acceleration in the following form:

$$\dot{\underline{U}}_{t+\tau} = \dot{\underline{U}}_t + \frac{\tau}{2}(\ddot{\underline{U}}_{t+\tau} + \ddot{\underline{U}}_t) \qquad \ldots (5.43a)$$

$$\underline{U}_{t+\tau} = \underline{U}_t + \tau\dot{\underline{U}}_t + \frac{\tau^2}{6}(\ddot{\underline{U}}_{t+\tau} + 2\ddot{\underline{U}}_t) \qquad \ldots (5.43b)$$

which gives

$$\underline{U}_{t+\tau} = \frac{6}{\tau^2}\Delta\underline{U} - \frac{6}{\tau}\dot{\underline{U}}_t - 2\ddot{\underline{U}}_t \qquad \ldots (5.43c)$$

and

$$\dot{\underline{U}}_{t+\tau} = \frac{3}{\tau}\Delta\underline{U} - 2\dot{\underline{U}}_t - \frac{\tau}{2}\ddot{\underline{U}}_t \qquad \ldots (5.43d)$$

Substituting Eqns. (5.43c) and (5.43d) in Eqn. (5.42a) one obtains

$$[\hat{K}]_t\,\{\Delta\underline{U}\} = \{\hat{P}_B\}_{t+\tau} \qquad \ldots (5.44)$$

where

$[\hat{K}]$ = effective undrained stiffness matrix

$$= [\overset{*}{K}]_t + \frac{6}{\tau^2}[M^*]_{t+\tau} + \frac{3}{\tau}[C^*]_{t+\tau} \qquad \ldots (5.44a)$$

and

$$\{\hat{P}_B\}_{t+\tau} = \text{effective load vector}$$

$$= \{P_B\}_{t+\tau} + [M^*] \left(\frac{6}{\tau} \dot{\underline{U}}_t + 2\ddot{\underline{U}}_t\right) + [C^*]$$

$$\left(2\dot{\underline{U}}_t + \frac{\tau}{2} \ddot{\underline{U}}_t\right) - \{\tilde{R}\}_t \qquad \ldots (5.44b)$$

5.3.2 Incremental Analysis

In an elasto-plastic analysis, the stress-strain relationship is generally formulated in an incremental form. The basis of the incremental formulation is to subdivide the real loading function into a finite number of small increments which are then prescribed one at a time. During the load increment, the stiffness matrix remains constant for that loading step. As the load is increased, the foundation reaches its elastic limit when the highest stressed element begins to yield following a specific yield model. Once the material has yielded, the stress-strain relationship can only be defined in terms of existing state of stress and can only be evaluated in an incremental form. The stiffness matrix is recomputed at every load step by substituting the appropriate elastic-plastic stiffness matrices for yielded elements. During each time of load step increment, an iterative approach in combination with the well known modified Newton-Raphson (MNR) technique is used with the total applied load subdivided into a finite number of load increments. For a dynamic analysis this means the evaluation of the effective load vector at each time step.

For each load/time increment, the incremental deflection is determined using the modified Newton-Raphson technique. If the solution at the start of a load increment is known, the tangent stiffness matrix at that point can be computed and used to obtain an initial linear estimate of the deflection at the end of the increment. However, since the load-deflection or stress-strain curve of the soil is non-linear, the actual load the system can support at this deflection is less than the applied load and thus a load unbalance exists at the end of the first iteration. The unbalanced loads are applied to the system and a revised estimate of the incremental deflection is determined using the same tangent stiffness matrix evaluated at the previous point. This process is repeated until a saw-tooth type convergence is achieved.

5.3.2.1 Iteration of Equilibrium Equations

Eqn. (5.44) was obtained by linearizing the system response about the configuration at time $t = t+\tau$. It may be noted here that the errors in the linearization can be expected to be small provided the load increments or time steps are very small. As the time step increment becomes very small, the cost of the analysis will increase tremendously and thus it is quite often necessary to imply larger time step increment for dynamic analysis or load step increment for static analysis. This large increment can introduce gross instability by slowly drifting away the computed solution from the exact solution. This is much more pronounced in the dynamic analysis than in the static analysis, since in the dynamic analysis the solution is always dependent on the past history. Therefore, in order to obtain a reasonable accurate solution, equilibrium iteration is necessary within a load and/or time step with a required tolerence criteria.

The basic equations for equilibrium iteration are obtained by using modified Newton-Raphson (MNR) technique and are given in matrix form as

$$[\hat{K}]_t^{(i-1)} \; \{\Delta\underline{U}\}^{(i)} = \{P_B\}_{t+\tau}^{(i-1)} \qquad \ldots (5.45a)$$

where

$$\{\Delta\underline{U}\}^{(i)} = \{\underline{U}\}_{t+\tau}^{(i)} - \{\underline{U}\}_{t+\tau}^{(i-1)} \qquad \ldots (5.45b)$$

and i = number of iterations within a particular time or/load step.

Eqn. (5.45a) forms the basis for the iterative solution of Eqn. (5.44) with the following initial conditons as

$$[\hat{K}]_{t+\tau}^{(o)} = [\hat{K}]_t, \; \{\tilde{R}\}_{t+\tau}^{(o)} = \{\tilde{R}\}_t \text{ and } \{\Delta\underline{U}\}_{t+\tau}^{(o)} = \{\Delta\underline{U}\} \qquad \ldots (5.46)$$

The iteration procedure is continued until a suitable convergence has been achieved in the solution process during a particular time/load step increment. The suitable convergence criteria is obtained by specifying a fixed tolerance (RTOL) at the beginning of the solution and by checking the ratio of the norm of the increment of displacement to the total displacement as

$$\left\| \frac{\Delta\underline{U}^{(i)}}{\underline{U}^{(i)} + \underline{U}_t} \right\|_2 < \text{RTOL} \qquad \ldots (5.47)$$

5.4 Permanent Deformation of A Strip Footing

The incremental elastic-plastic analysis is used here to compute the permanent shakedown displacement of a strip footing under cyclic vertical loading condition. The analysis is carried out for a first few cycles of the loading until a steady-state condition is reached.

The simple finite element model of a strip footing underlaid by a shallow stratum of clay under undrained condition is shown in Fig. 5.6. The mesh consists of 32 elements and 45 nodes. The ratio of the footing width to the depth of the stratum is 0.80. Because of the symmetry of the vertical loading, the mesh for half of the footing is shown in Fig. 5.6. The loading diagram is similar to a triangular pulse with a period of 1 second. The loading magnitude is defined by a dimensionless parameter, F, equal to the ratio of weight per unit area to the undrained shear strength (i.e. $F = W/As_u$ in which W is the vertical load, A = area of the strip footing and s_u = undrained shear strength of the soil). Five cycles of loading (5 seconds duration) are considered in the analysis and a total of eighty (80) load steps are used in order to trace the entire load path.

Vertical side boundary is placed at a distance of

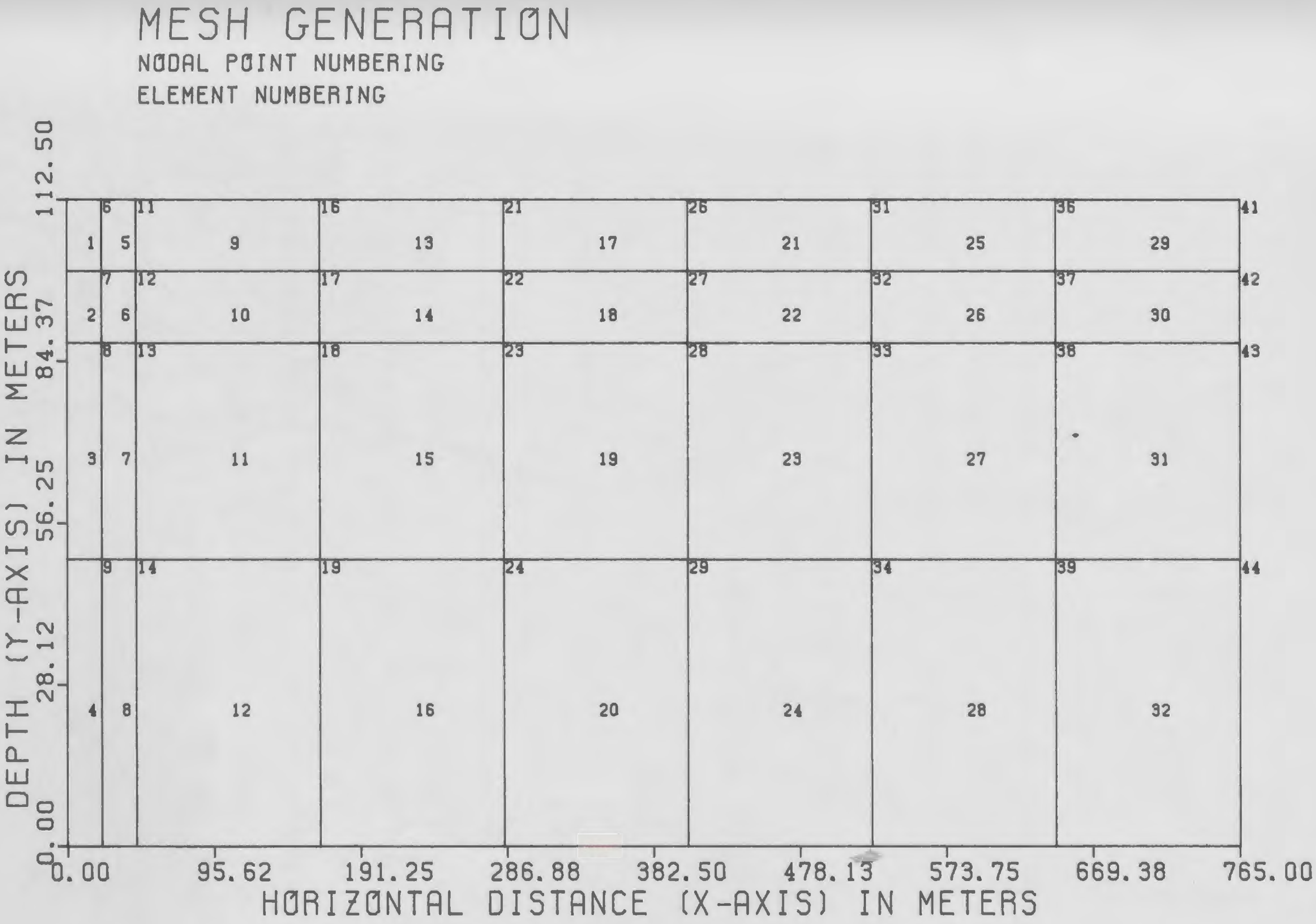


FIG. 5.6 FINITE ELEMENT MESH FOR STRIP FOUNDATION (INCREMENTAL ANALYSIS)

eight and a half times the width of the footing and assumed to be smooth. The bottom boundary is assumed to be rough and rigid. The footing is assumed to deform under plane strain condition. Reduced integration technique is used to ensure the nearly incompressibility of the undrained soil medium.

In the elastic-plastic undrained analysis the soil is modelled as a weightless two-phase material with $s_u = 100$ KPa and $\phi_u = 0$, where s_u = undrained shear strength and ϕ_u = angle of internal friction. The drained elastic Lame's parameters $\lambda = 150\ s_u$, $\mu = 100 s_u$ and drained Poisson's ratio, $\nu_d = 0.30$ are assumed for the soil skeleton whereas the bulk modulus for pore water is assigned as $K_w = 2000$ MPa. This gives an equivalent undrained Poisson's ratio, $\nu_u = 0.499$ approximately.

In order to compute the permanment deformation of the strip footing, the elastic and shakedown limit pressures obtained in Chapter 4 (Refer Table 4.2) are used here. It is found that for $F^e < 3.28$ (dimensionless elastic limit pressure) the behaviour of the soil foundation is purely elastic. However, for $F = F^s = 5.31$, the foundation undergoes cyclic inelastic deformation for few cycles and then reaches a steady state situation. The reason for applying a shakedown pressure of $F^s = 5.31$ was to obtain the boundedness of the deformation. Table 5.2 shows the elastic and inelastic deformation for the first five cycles of loading. For $F^s = 5.31$, the permanent (inelastic) deformation increases for the first two cycles of loading and then remains unchanged implying that the plastic work is bounded in time and a limited amount of plastic flow has occurred. The foundation develops residual stresses due to the initial plastic flow and therefore the stresses obtained from the subsequent cyclic elastic-plastic analysis can be superimposed on these residual stresses without violating the yield criterion (shakedown condition). Thus, the foundation shakes down to elastic behaviour and any load $3.28 \leq F^s < 5.31$ will be a safe load in order to achieve a shakedown condition. The inelastic (permanent) deformation computed for the footing is 0.82% of the footing width and shown in Table 5.2.

TABLE 5.2 - PERMANENT DEFORMATION OF A STRIP FOOTING SUBJECTED TO REPEATED VERTICAL LOADING

Cycle Number	(Vertical Displacement/Footing Width) in Percent		
	Elastic	Permanent	Total
1/4	0.38	-	0.38
1/2	0.76	0.79	1.56
3/4	0.38	0.81	1.19
1	0.00	0.81	0.81
1 1/4	0.38	0.81	1.19
1 1/2	0.76	0.82	1.58
1 3/4	0.38	0.82	1.20
2	0.00	0.82	0.82
2 1/4	0.38	0.82	1.20
2 1/2	0.76	0.82	1.58
2 3/4	0.38	0.82	1.20
3	0.00	0.82	0.82
3 1/4	0.38	0.82	1.20
3 1/2	0.76	0.82	1.58
3 3/4	0.38	0.82	1.20
4	0.00	0.82	0.82
4 1/4	0.38	0.82	1.20
4 1/2	0.76	0.82	1.58
4 3/4	0.38	0.82	1.20
5	0.00	0.82	0.82

CHAPTER VI

RESPONSE ANALYSES OF A GRAVITY TYPE OFFSHORE FOUNDATION CONSIDERING SOIL SHAKEDOWN

6.1 Introduction

Three types of analyses are carried out in order to compute the permanent deformations beneath a gravity type offshore foundation. The first analysis investigates the effects of eccentricity and angle of inclination of the load on the shakedown load factor for a flexible foundation.

In the second analysis, the permanent deformation of the foundation of a gravity type offshore platform is studied considering the static soil shakedown. In the third analysis the dynamic shakedown load factor is first determined and the responses are then computed and compared for various types of soil modelling. In the second and third analyses the effect of the caisson structure is taken into account by modelling it as a stiff footing.

6.2 Method of Approach

The above analyses are carried out using the two-dimensional finite element plane strain formulation described in Chapters 3, 4 and 5. The detailed descriptions of the computer programme especially developed for the present investigation and the necessary input data requirements are presented in Appendices C and D.

6.2.1 Limitations of the Method

The limitations of the present approach are:

a. the effect of the change in volume due to shear deformation is neglected.

b. the soil beneath the foundation is assumed to be linearly elastic-perfectly plastic material and is modelled with the Von Mises yield criteria following associated flow theory of plasticity. The assumption of constant soil shear strength which is independent of the normal stress components is a valid approximation for undrained saturated clay, the so called $\phi_u = 0$ condition (Skempton, 1941).

c. Present analyses are based on small strain formulation i.e. they do not consider the nonlinearity caused by the change in geometry.

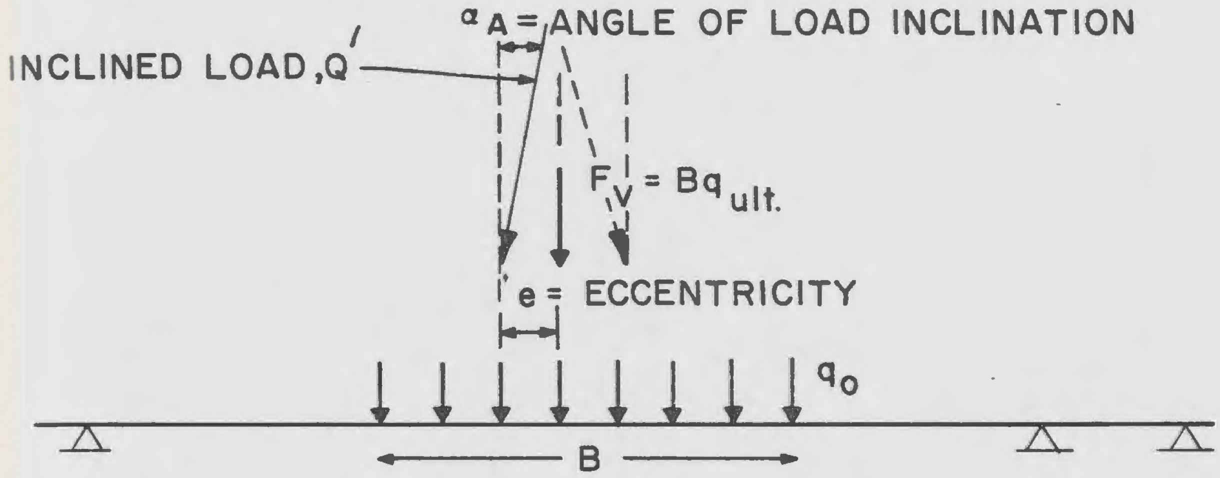
6.3 Flexible Foundation Analysis

Several analyses are carried out in order to determine the effects of the eccentricity and the angle of inclination of the cyclic wave loading on the static elastic and shakedown limit pressures.

6.3.1 Geometry and Finite Element Mesh

Fig. 6.1 shows the geometry of the foundation resting on saturated cohesive soil. Based on the bearing capacity formula, two load inclination angles, α_A, and five ratios of eccentricity to footing width, e/B, have been chosen for computing the elastic and shakedown limit pressures. Von Mises yield condition is used in all analyses and represented by six linearized yield planes.

Fig. 6.2 depicts the finite element mesh for the flexible strip foundation. The foundation has a loaded width of ninety meters. The soil medium is assumed to be homogeneous and isotropic. The



FOOTING WIDTH, B = 90.0 m

UNDRAINED SHEAR STRENGTH, S_u = 100.0 kN/m^2

BULK WEIGHT, ρ = 18.6 kN/m^3

SPECIFIC WEIGHT OF WATER, ρ_f = 9.8 kN/m^3

$$q_{ult} = \frac{F_V}{B'L} = N_c \times S_u \times \left(1 - \frac{\alpha_A}{90}\right)^2 \left(1 - 2\frac{e}{B}\right)$$

WHERE $\alpha = \tan^{-1}\left(\frac{F_H}{F_V}\right)$

$N_c = (2 + \pi)$

B' = EFFECTIVE WIDTH

FIG. 6.1 BEARING CAPACITY FOR FLEXIBLE FOOTING ON COHESIVE SOIL (MEYERHOF, 1963)

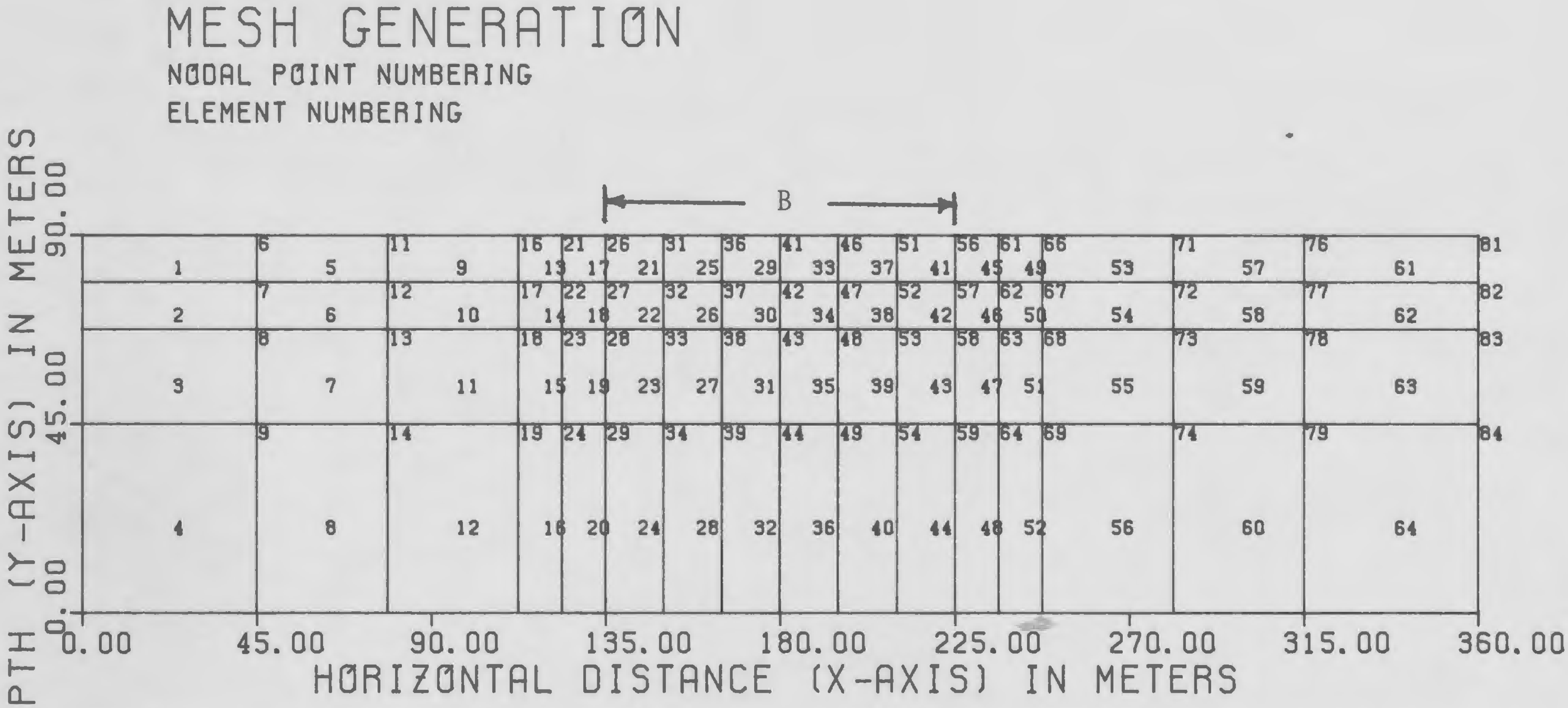


FIG. 6.2 FINITE ELEMENT MESH FOR STRIP FOUNDATION

mesh consists of 85 nodes and 64 four noded rectangular isoparametric elements. Vertical side boundary is placed at a distance two times the width of the footing from the footing centerline and assumed to be smooth. The bottom boundary is assumed to be rough and rigid. The footing is assumed to deform under plane strain condition. Reduced integration technique is used to implement effectively the nearly incompressibility condition for undrained deformation.

In the undrained analysis the soil is modelled as a weightless two-phase material with $s_u = 100.0$ kPa and $\phi_u = 0.0$. The drained elastic Lame's parameters $\lambda = 150\ s_u$, $\mu = 100\ s_u$ and drained Poison's ratio, $\nu_d = 0.30$ are assumed for the soil skeleton whereas the bulk modulus for pore water is assigned as $K_w = 2000$ MPa. This gives an equivalent undrained Poisson's ratio, $\nu_u \simeq 0.499$.

The submerged weight, F_V, of the gravity type offshore structure is assumed to act with an eccentricity, $\pm e$, which causes a cyclic moment, $F_M = F_V.(\pm e)$ at the base level (Fig. 6.1). The foundation is also subjected to the horizontal force component, $\pm F_H$, of the cyclic inclined loading, Q', as shown in Fig. 6.1. The stress distributions due to loadings, F_V, F_H and F_M are assumed to be uniform and linear respectively.

6.3.2 Evaluation of Elastic Limit Load

For the vertical load alone (e/B = 0.0, α_A = 0.0), the vertical pressure is applied at each of the seven nodes 26, 31, 36, 41, 46, 51, and 56.

The stress fields comprising of σ_x, σ_y, σ_z and τ_{xy} for all the elements are obtained by applying the vertical pressure and by observing the pressure at which one/or more of the elements will first become plastic.

The elastic limit load factor, β_E is obtained from Eqn. (5.25d) and Eqn. (5.26b) as

$$\beta_E = \frac{s_u}{\sqrt{\tfrac{1}{2}(S_x^2 + S_y^2 + S_z^2) + \tau_{xy}^2}} \qquad \ldots(6.1)$$

and computed for each element. From the elastic half space theory the elastic limit pressure is given as 3.14 s_u (Jumikis, 1969). This pressure is obtained based on Tresca failure surface. However, the present analysis indicates a value of

3.08 s_u which is within the 2.0 % of the theoretical elastic limit pressure solution. Element No. 31, at a depth of 34.0 meters from the surface indicates a minimum load factor, β_E, of 1.29. The vertical displacement at center node 41, is compared with that obtained based on Kelvin's equation for a line load acting within an infinite solid (Taylor and Matyas; 1983).

For a clay layer with constant Young's modulus the vertical settlement may be estimated following Taylor and Matyas (1983) as

$$\delta_V = \frac{\sigma_V \, d_o}{\Pi E} \cdot \frac{1 + \nu_u}{1 - \nu_u} \left[\alpha_o + (1 - 2\nu_u)\, \alpha_1\right] \quad \ldots(6.2)$$

where

δ_V = vertical settlement in meter

σ_V = applied vertical pressure = 233 kPa

d_o = depth of the foundation = 90 m

B = foundation width = 90 m (Fig. 6.1 and Fig. 6.2)

ν_u = Poisson's ratio = 0.499

E = Young's modulus = 29,980 kPa

and α_0 and α_1 are taken from Fig. 5 of the above reference; for $d_o/B = 1.0$, $\alpha_0 = 0.3571$ and $\alpha_1 = 2.5833$.

Based on the above parameters, δ_V is computed as 0.2415 meter as opposed to the vertical displacement of 0.2120 meter obtained from the finite element analysis. The difference in the result can be explained due to the effect of the side boundary.

6.3.3 Evaluation of Shakedown Limit Load

The computer programme has been checked in Chapter 4 for a strip footing with the available solutions under vertical loading condition. In this section, the static shakedown analysis of a flexible foundation is carried out under cyclic inclined eccentric loading condition. Fig. 6.1 depicts the loading system and notation.

The piecewise linearized Von Mises yield criterion with its associative nature of flow rule condition is used in shakedown analyses. The finite element mesh of the soil stratum is shown in Fig. 6.2 and the material properties are described in Section 6.3.1.

For the case of centered vertical loading (i.e. $\alpha_A = 0.0$ and $e/B = 0.0$), the shakedown limit pressure, q_s, is obtained as 5.22 s_u from Eqn. (4.55) and shown in Table 6.1. The above pressue is 1.5% higher than the true collapse pressure. The typical central processing unit (cpu) time required is 180 minutes and major portion of this time is spent in solving the linear programming problem involving 193 constraints and 640 variables.

6.3.4 Influence of Eccentricity and Load Inclination Angle

The parametric studies are carried out in order to determine the influence of the various, e/B, ratios and the angle of inclination of the cyclic wave loading, α_A, on the elastic and shakedown limit pressures.

The elastic limit pressures, q_e, for eccentricity ratios of $e/B = 0.05$ to 0.16 are also computed at increments of 0.05 for $\alpha_A = 0$ and shown in

TABLE 6.1 - COMPARISON OF ELASTIC AND SHAKEDOWN LIMIT PRESSURES FOR e/B = 0.0 - 0.16 AND $\alpha_A = 0.0^\circ$

e/B	Number of elements	Number of Nodal Points	Number of Equations	Elastic Limit Pressure (q_e/s_u)	Shakedown Limit Pressure (q_s/s_u)
0.0	64	85	128	3.08	5.22
0.05	64	85	128	2.75	4.10
0.10	64	85	128	2.28	3.60
0.15	64	85	128	2.10	3.15
0.16	64	85	128	1.98	3.10

Table 6.1. The elastic limit pressure, q_e, is reduced as much as 35% for an eccentricity ratio e/B = 0.16 as compared to that corresponding to e/B = 0.0. For $\alpha_A = 10^o$, the uniform horizontal pressure is also applied in addition to the distributed vertical pressure to the nodes at the top of the caisson. The elastic limit pressure, q_e, for e/B = 0.0 is computed as $2.98s_u$. In this case, the minimum value of β_E, is obtained in element No. 35. Table 6.2 illustrates the values of the elastic limit pressure, q_e, for all e/B ratios.

For $\alpha_A = 0.0$, Table 6.1 compares the values of the static shakedown limit pressures for various e/B ratios from 0.0 to 0.16, whereas Table 6.2 compares the same for $\alpha_A = 10.0^o$. Results show that the shakedown limit pressures decrease as the e/B ratio and the inclination angle increase. The reduction could be as high as 54% e.g. for e/B = 0.16 and $\alpha_A = 10^o$ as opposed to e/B = 0.0 and $\alpha_A = 0.0$. Figs. 6.3(a) and (b) depict the comparison of these values with those obtained from semi-empirical bearing capacity relationship (Fig. 6.1) given by Meyerhof (1963). The shakedown pressures are obtained from an upperbound solution and therefore the use of the semi-empirical formula will

TABLE 6.2 - COMPARISON OF ELASTIC AND SHAKEDOWN LIMIT PRESSURES FOR e/B = 0.0 - 0.16 AND $\alpha_A = 10.0^\circ$

e/B	Number of elements	Number of Nodal Points	Number of Equations	Elastic Limit Pressure (q_e/s_u)	Shakedown Limit Pressure (q_s/s_u)
0.0	64	85	128	2.98	3.92
0.05	64	85	128	2.63	3.52
0.10	64	85	128	2.23	3.00
0.15	64	85	128	1.90	2.50
0.16	64	85	128	1.80	2.40

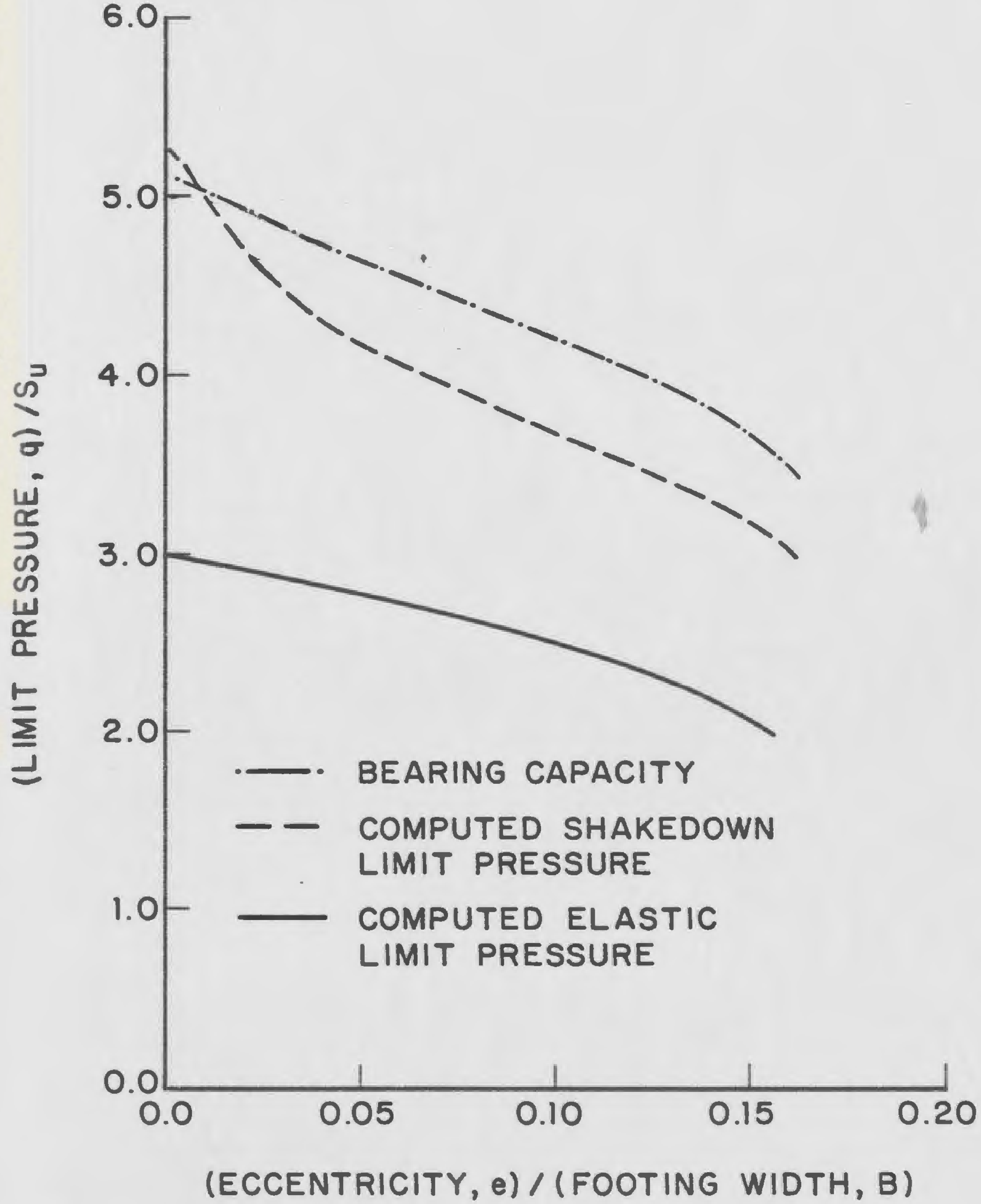


FIG. 6.3 (a) ELASTIC AND SHAKEDOWN LIMIT PRESSURES FOR VARIOUS ECCENTRICITIES ($\alpha_A = 0.0°$)

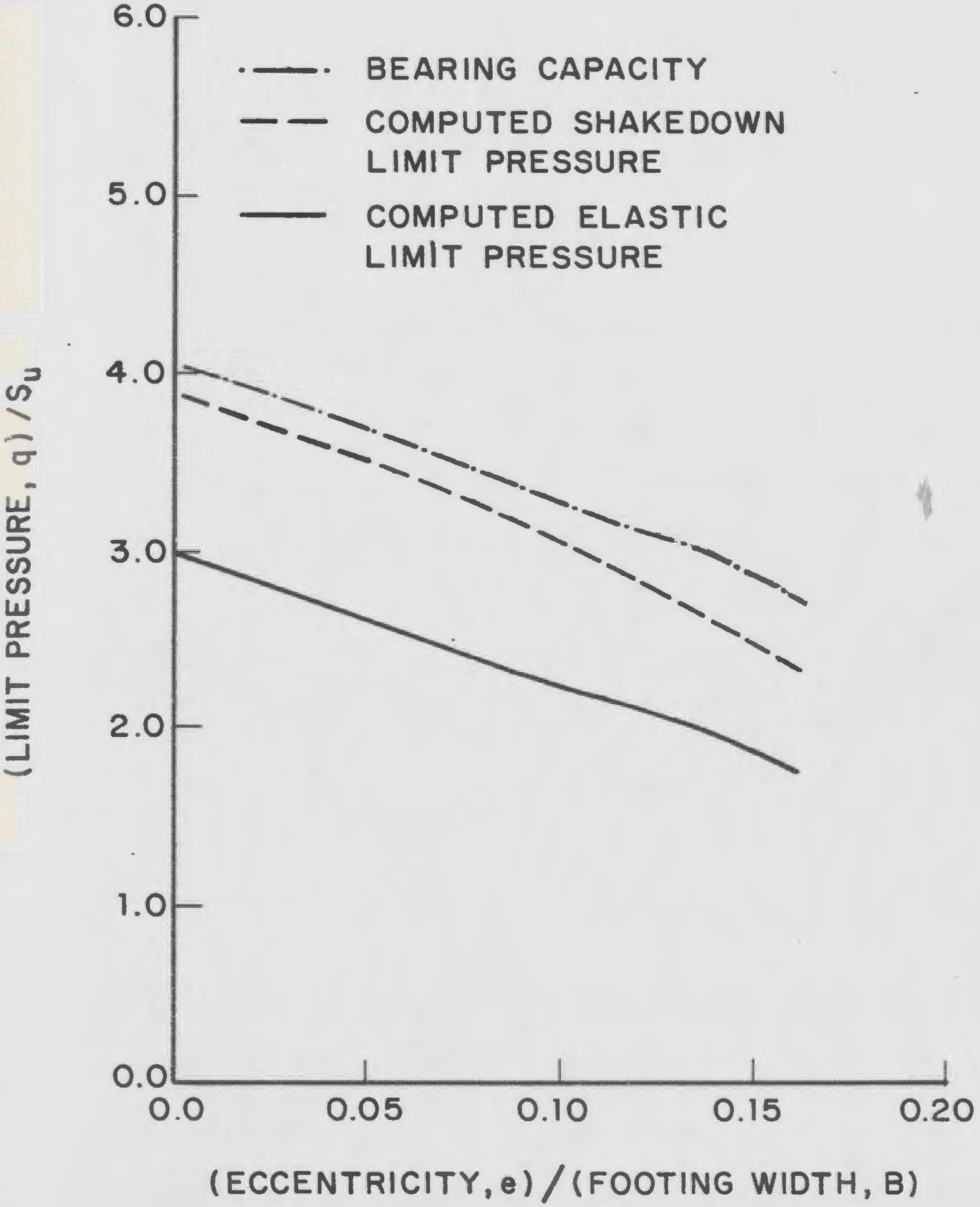


FIG. 6.3(b) ELASTIC AND SHAKEDOWN LIMIT PRESSURES FOR VARIOUS ECCENTRICITIES ($\alpha_A = 10^\circ$)

not produce a conservative estimate of the bearing capacity of the foundation when the loading is repetitive in nature.

6.4 Static Analysis of a Gravity Type Offshore Foundation

In this section the formulation presented in the previous chapters is used to analyze the action of an offshore gravity structure with its soil foundation which consists of saturated clays resting on a rigid base. Particular attention is given to the static nature of the wave loading and the soil foundation is assumed to behave in an undrained manner during one particular storm. The reduced integration technique is used in order to implement effectively the nearly incompressibility condition for undrained deformation.

The permament deformation of the caisson structure is computed by using an elastic-plastic finite element formulation as described in Section 5.2.3. A static shakedown analysis is first carried out for the entire caisson-foundation-soil system. Having computed the static shakedown load factor, the loading domain at shakedown is first established and the responses are then computed and compared for two types of soil behaviour: a) linear and b) nonlinear.

6.4.1 Loading Programme

The magnitude of the wave loading on the structure-foundation depends on the wave period, T_i, length, L_i, and height, H_i and the dimensions of the structure. In a typical storm situation there are varying number of waves having similar characteristics and can be grouped together. However, it has been observed that the complete storm in a deep ocean environment consists of wave groups having smaller heights followed by wave groups having larger heights until the peak of the storm has been reached. In the present analysis a single wave with the largest height with maximum return period has been considered. The individual wave has a wave length of 218.0 meters, height 30.0 meters and period 12.0 seconds. Before the storm, structure is in calm water and as such the total vertical force, F_V, acting on the structure is the buoyant (submerged) weight of the structure.

Fig. 6.4(a) shows a wave profile in the free field and Fig. 6.4(b) shows the structure standing in

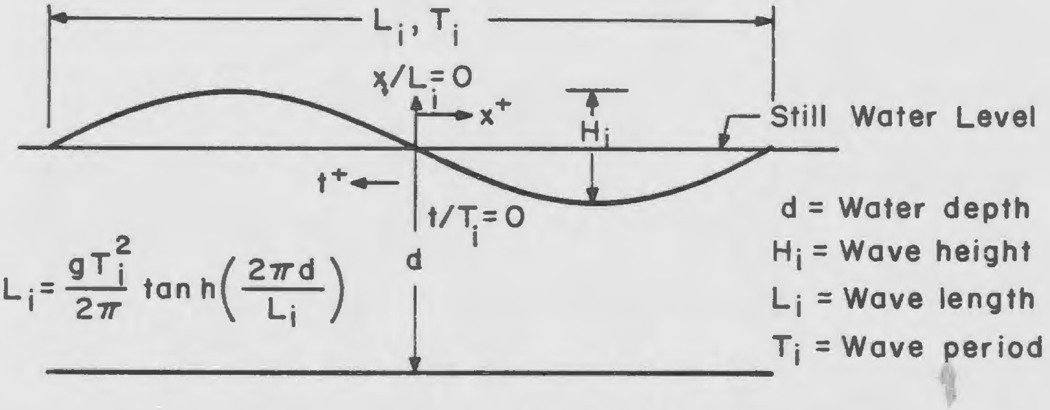


FIG. 6.4(a) CHARACTERISTICS OF A LINEAR WAVE

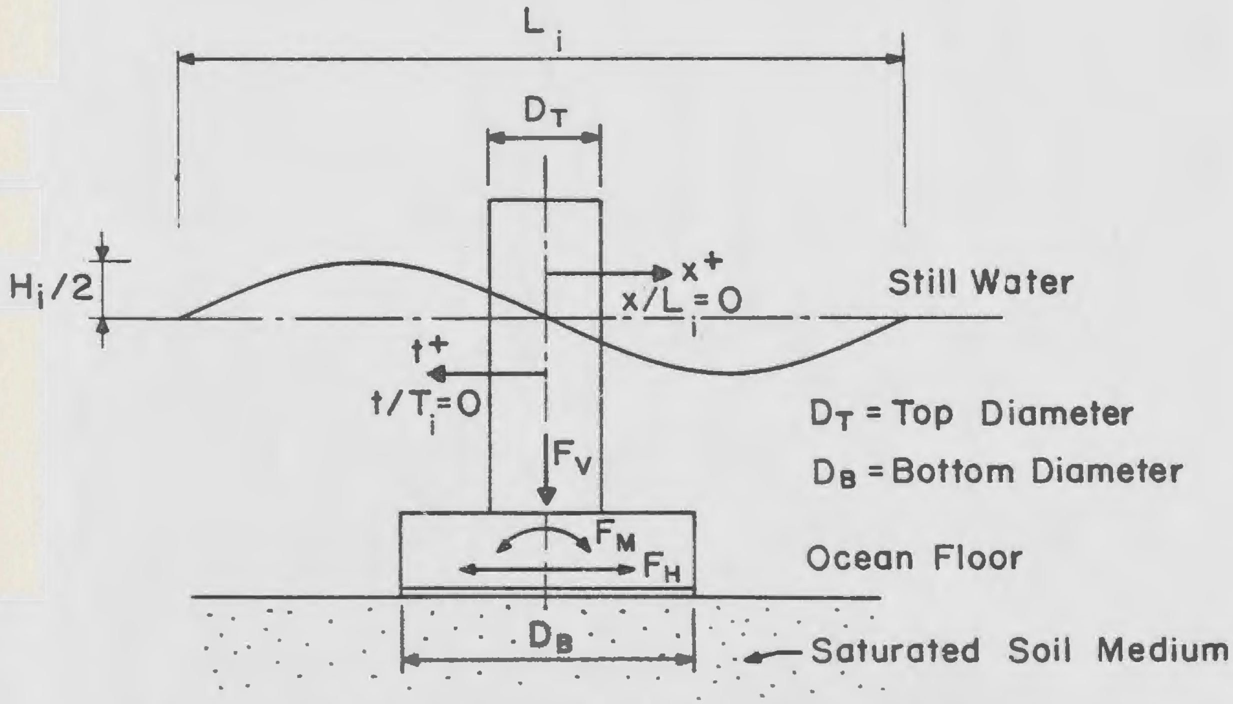


FIG. 6.4(b) WAVE FORCES ON STRUCTURE

water during a storm with its axis perpendicular to the line of the wave propagation. If the lateral dimension of the caisson is comparable to the wave length, the wave forces are primarily due to diffraction effects i.e. including inertial forces. The drag force is comparatively small and therefore can be neglected. The forces acting on the caisson due to a single wave is computed from diffraction theory and the values obtained have been taken from a problem given by Pool (1976). The net effect of the wave force is shown in Fig. 6.4(c) and represented by a cyclic horizontal force, $\pm F_H$, and a cyclic moment, $\pm F_M$. The general cyclic wave loading domain is therefore expressed as

$$-\beta\bar{F}_H \leq F_H \leq \beta\bar{F}_H$$

and $$-\beta\bar{F}_M \leq F_M \leq \beta\bar{F}_M$$

where $\bar{F}_H$ and $\bar{F}_M$ are the nondimensional parameters and defined as

$$\bar{F}_H = \frac{F_H}{As_u}\text{ , and } \bar{F}_M = F_H . a_o ;$$

where, β is the appropriate load factor, A is the area of the foundation under plane strain condition, a_o is the height of the caisson structure

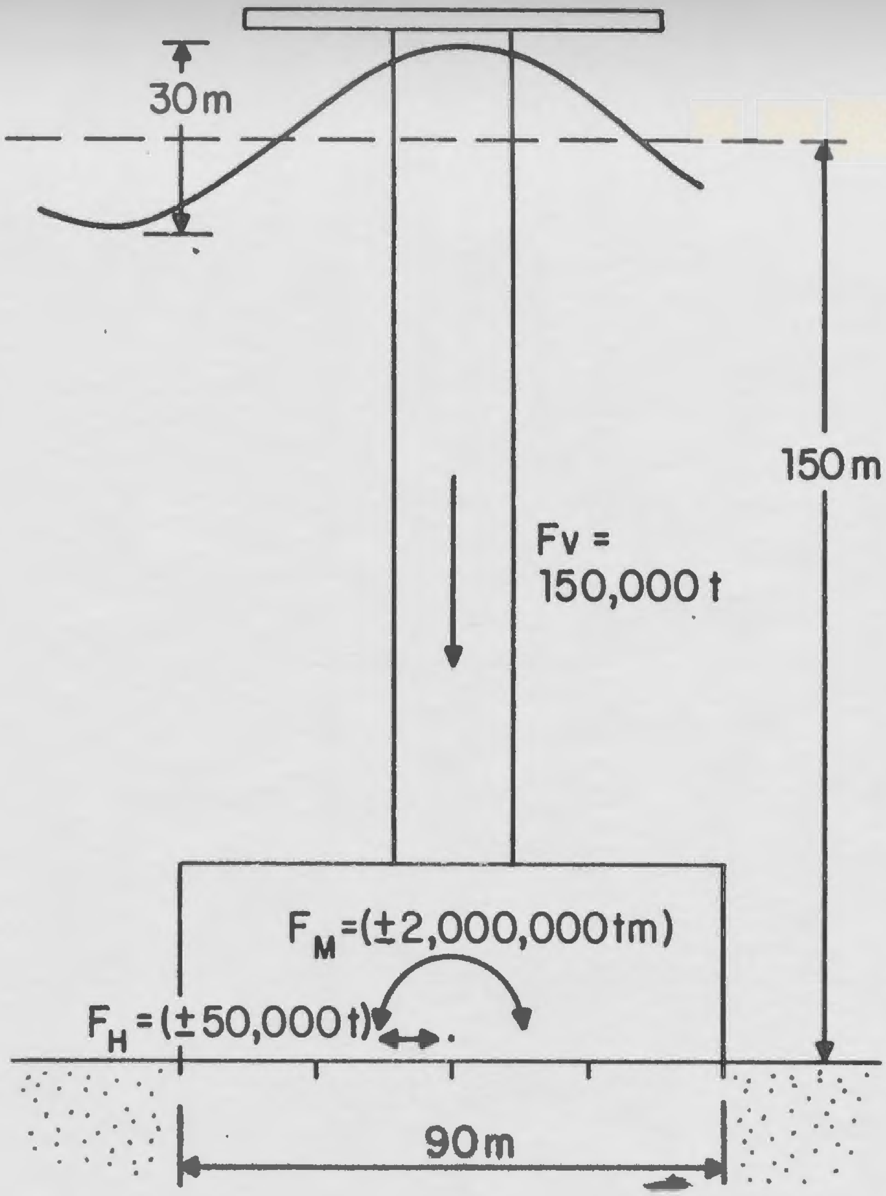


FIG. 6.4(c) TYPICAL LOADS ON A GRAVITY STRUCTURE (POOL, 1976)

above the mud line (a_o = 40.0 meters) and $\bar{F}_H$ and $\bar{F}_M$ are prescribed values of F_H and F_M.

6.4.2 Modelling of the Caisson Structure and the Foundation Soil

The circular platform base shown in Fig. 6.4(c) has a diameter of 90 meters and is modelled as an equivalent rectangle with the same area. The rectangular base is later idealized as a plane strain model with the width equal to that of the structure. The total wave force is distributed per unit meter width of the structure. A two dimensional plane strain finite element model is used in carrying out the analysis. The caisson structure is modelled by two-dimensional four noded linear element whereas the foundation is modelled by the same type of elements using linear and nonlinear soil properties representing the characteristics of saturated cohesive soil.

The finite element mesh of the caisson structure-foundation system is shown in Fig. 6.5. The mesh consists of five structural elements (elements 21, 26, 31, 36, and 41) and sixty soil elements. The shear strength is assumed to be constant over the depth and the behaviour of the soil is assumed to

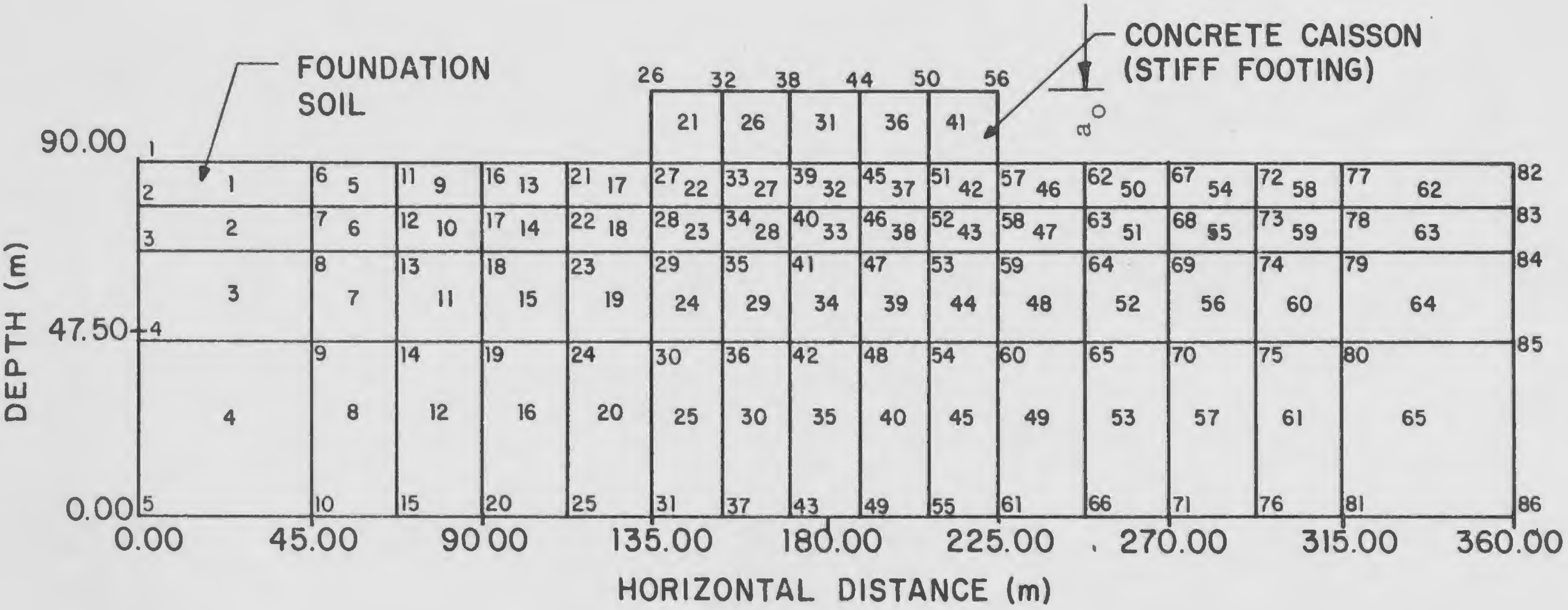


FIG. 6.5 FINITE ELEMENT MESH FOR STIFF FOUNDATION (NOT TO SCALE)

be undrained in nature with respect to the cyclic loading condition i.e. there is no flow of fluid with respect to soil.

In order to model the caisson foundation a very high stiffness is assigned to the structural elements 21, 26, 31, 36 and 41. The loads are applied to the foundation through the nodes 26, 32, 38, 44, 50 and 56. The stiffness ratio of the structure to the foundation soil, t_s, is computed from the following equation (Jumikis, 1969) as

$$t_s = \frac{1.25\ E_L\ B^3}{E_f\ a_o^3} \qquad \ldots(6.3)$$

where E_L = Young's modulus of the soil in the finite layer

E_f = Young's modulus of the footing

a_o = thickness of the footing

$B/2$ = half width of the footing

The semi-empirical method for obtaining the settlement at the center of the footing is given by Jumikis (1969) as follows:

$$\delta_V = [K(t) + 2K_r] \frac{\sigma_V B (1 - \nu_u)^2}{2\Pi E_L} \qquad \ldots(6.4)$$

where $K(t) = 0$ for $t_s = 0.0$ implying the rigid footing condition. The values of K_r for different, $\frac{2d_o}{B}$, ratios and variation of $K(t)$ with respect to t_s have been given by Jumikis (1969).

In the undrained analysis, the soil is treated as a weightless two phase material with $s_u = 100.0$ kPa, and $\phi_u = 0.0$ throughout the depth of the layer. The drained elastic parameters $\lambda = 150\ s_u$, $\mu = 100\ s_u$ and $\nu_d = 0.30$ are assumed for the soil skeleton and a high value of bulk modulus, 2000 MPa is assigned to the pore water.

The ratio of the Young's modulus for the caisson foundation to the soil medium is assumed as $\frac{E_f}{E_L} = 1000$. Therefore the stiffness ratio, t_s computed from Eqn. (6.3) becomes 0.0142, where $B/2 = 45.0$ meters and $a_o = 90.0$ meters. Because t_s is relatively small, it is reasonable to assume that the footing is stiff compared to the foundation soil. The computed vertical displacement at the center of the caisson is compared with that given by Jumikis (1969) and found to be in agreement. It is interesting to note that the vertical displacement of the stiff footing is found to be 25%

less than that compared with the flexible one.

6.4.3 Evaluation of Elastic Limit Load

The stress fields comprising of σ_x, σ_y, σ_z and τ_{xy} for all the elements are obtained by applying the cyclic horizontal loading, $\bar{F}_H$, ($\bar{F}_H = \frac{F_H}{As_u} = 0.78$) coupled with the vertical submerged weight of the structure, $\frac{F_V}{As_u} = 2.33$, at nodes 26, 32, 38, 44, 50 and 56. The elastic limit load factor, β_E, is then computed from Eqn. (6.1) which translates into the following elastic loading domain; $\frac{F_V}{As_u} = 2.33$ and $\frac{F_H}{As_u} = 0.50$.

6.4.4 Evaluation of Shakedown Limit Load

In this section, the static shakedown analysis of a caisson-structure foundation system is carried out under cyclic inclined eccentric loading condition. The cyclic horizontal loading due to the wave action and the vertical submerged weight of the structure are applied at nodes which are located at the top of the caisson structure. The upper bound value of 1.28 for shakedown load factor is obtained from Eqn. (4.55). Therefore shake-domain is described as follows; $\frac{F_V}{As_u} = 2.33$ and $\pm \frac{F_H}{As_u} = 0.64$. The total number of constraints and variables for this problem are 196 and 654 respectively. The central processing unit (cpu) time for this problem is 210 minutes and the major portion of this time is again utilised for solving the linear programming problem.

6.4.5 Elastic Analysis

The linear stress analysis of the structure-foundation system in calm water is carried out with a net submerged vertical force, F_V. The analysis is based on the assumption of undrained soil behaviour and the contour plots for normal stress σ_y, and pore pressure, p, are shown in Figs. 6.6(a) and 6.6(b). Assuming the cyclic horizontal force is at its maximum value (i.e. wave phase angle is either 90^o or 270^o refer Fig. 1.1) the elastic stress analysis is also carried out with the cyclic horizontal force, F_H, coupled with the submerged weight, F_V, acting at the top nodes of the caisson structure.

The distributions of horizontal and vertical displacements at selected nodes along the horizontal direction are shown in Figs. 6.7(a) and 6.7(b). In this case, the horizontal force is assumed to act from left to the right of the

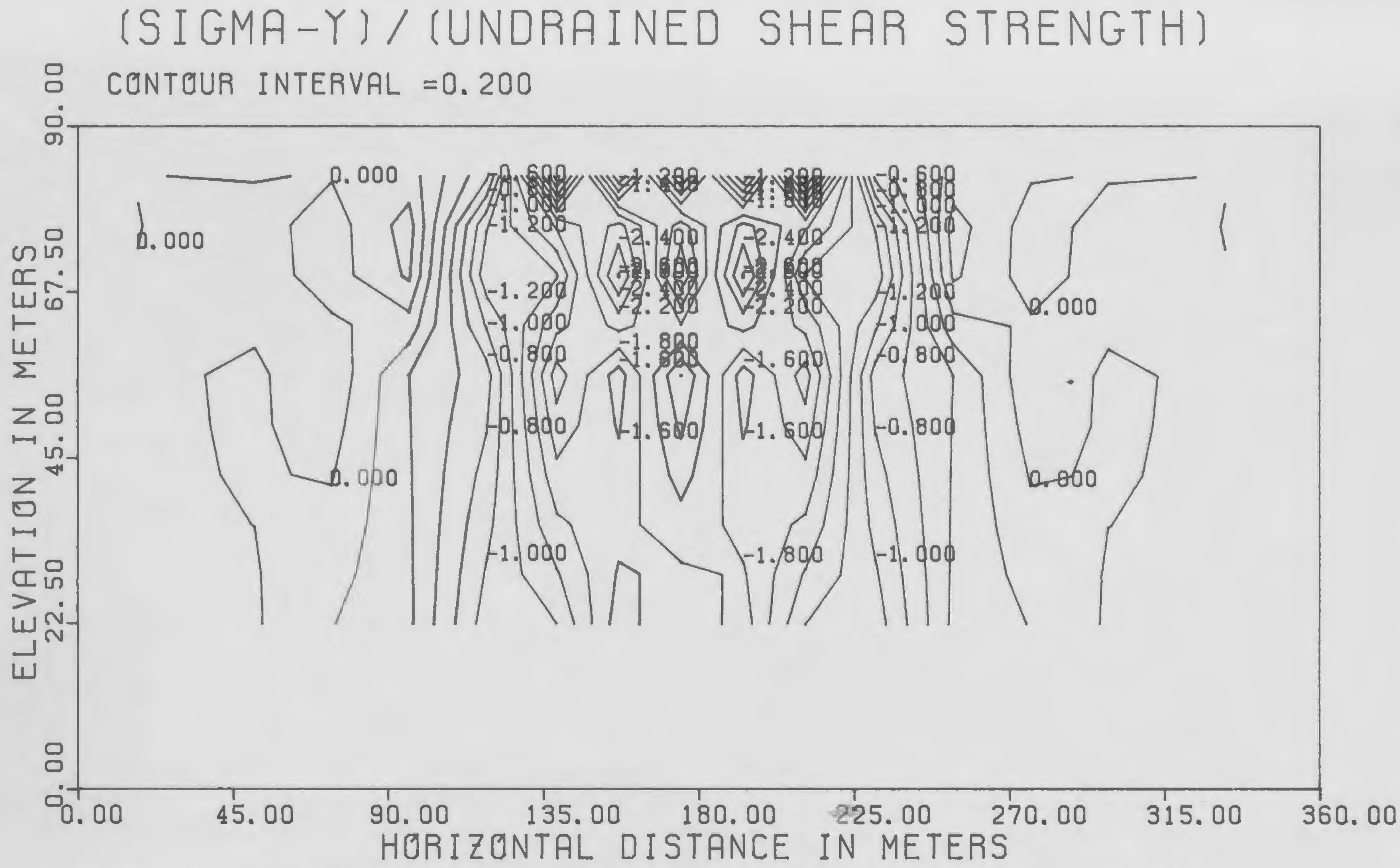


FIG. 6.6(a) STRESS CONTOURS PLOT FOR (SIGMA-Y)
(WITH CAISSON; SUBMERGED WEIGHT ONLY)

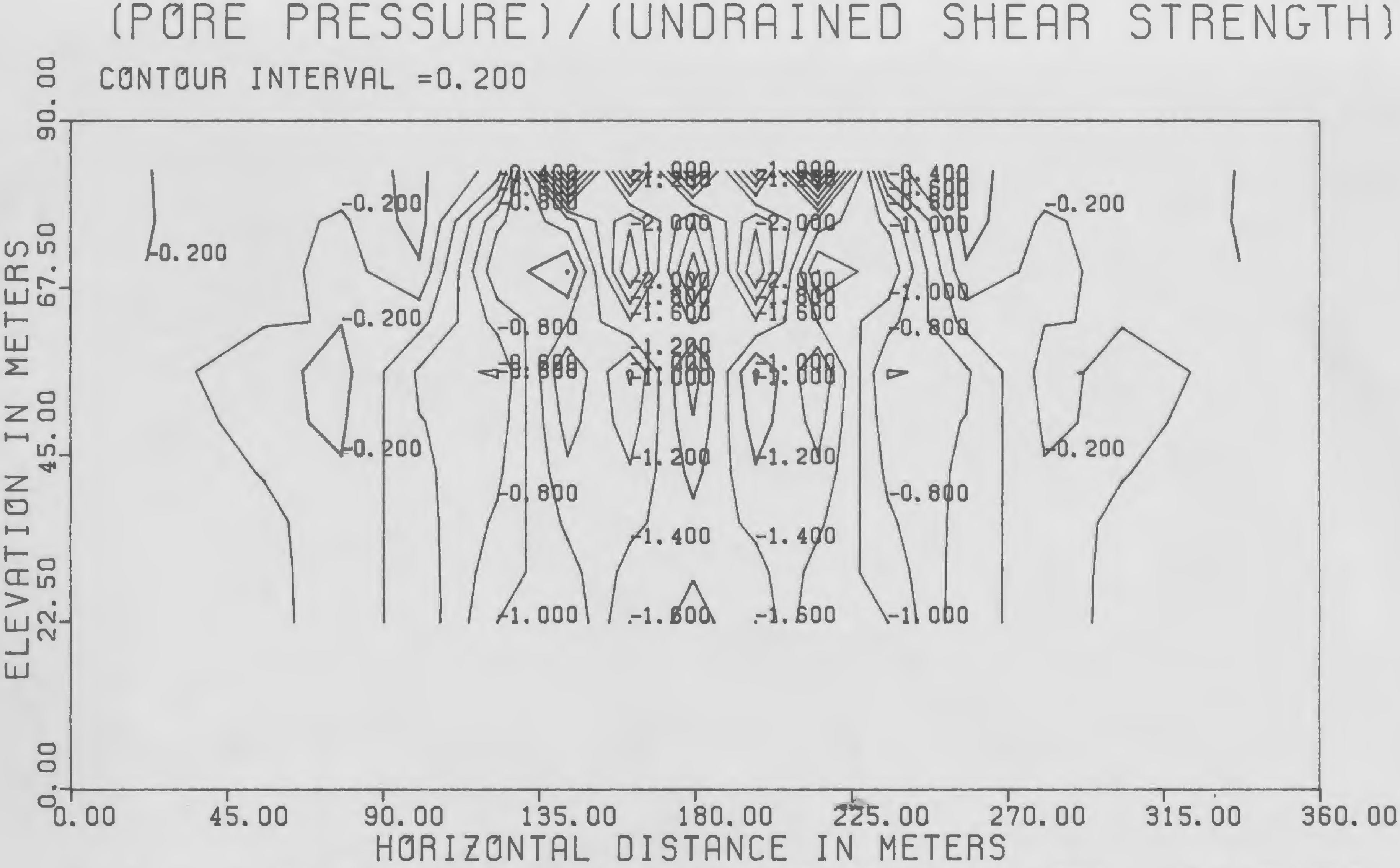


FIG. 6.6(b) STRESS CONTOURS PLOT FOR PORE PRESSURE (WITH CAISSON; SUBMERGED WEIGHT ONLY)

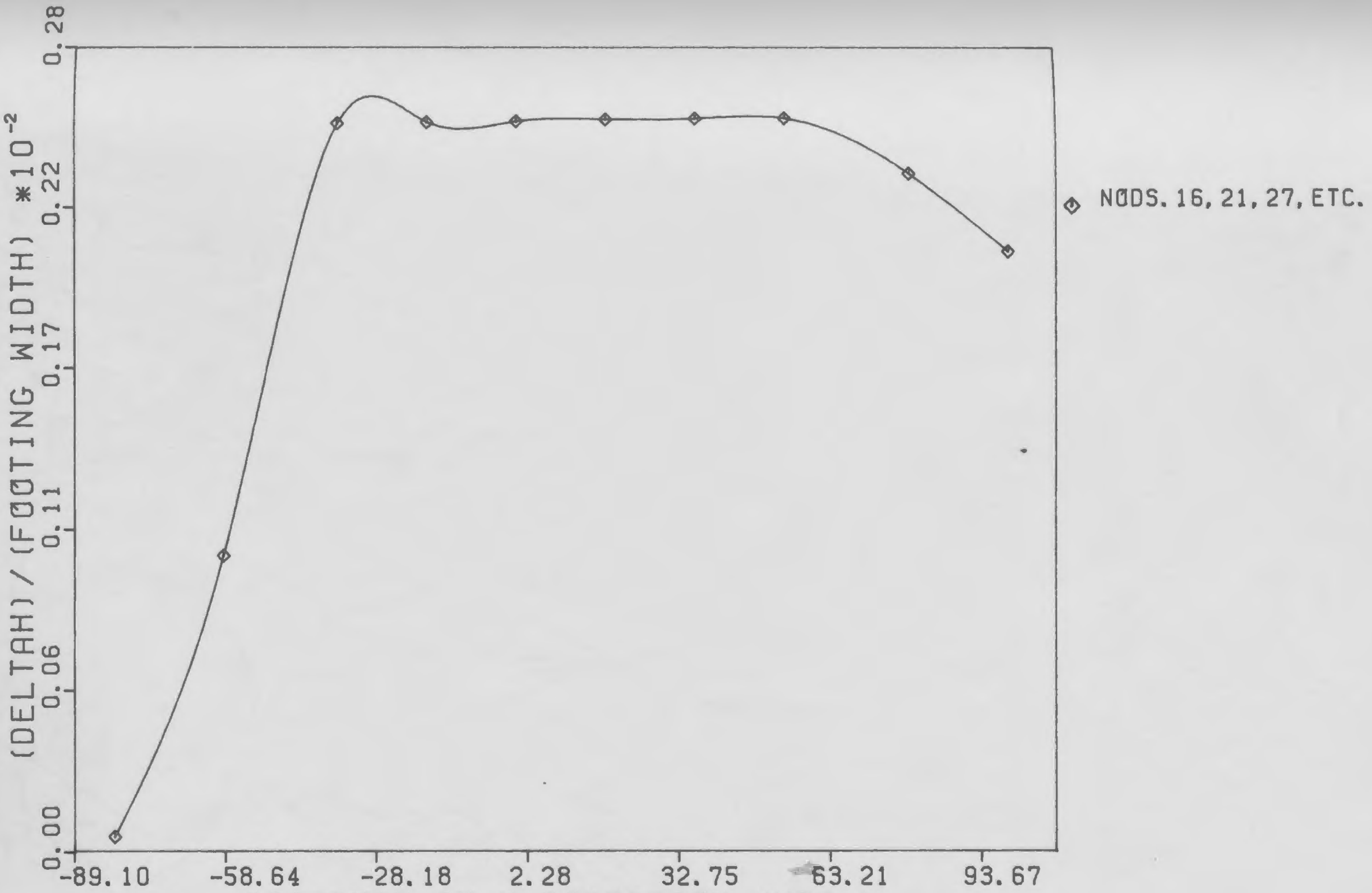


FIG. 6.7(a) DISTRIBUTION OF HOR. DISPLACEMENT
(WITH CAISSON; INCLINED LOADING)

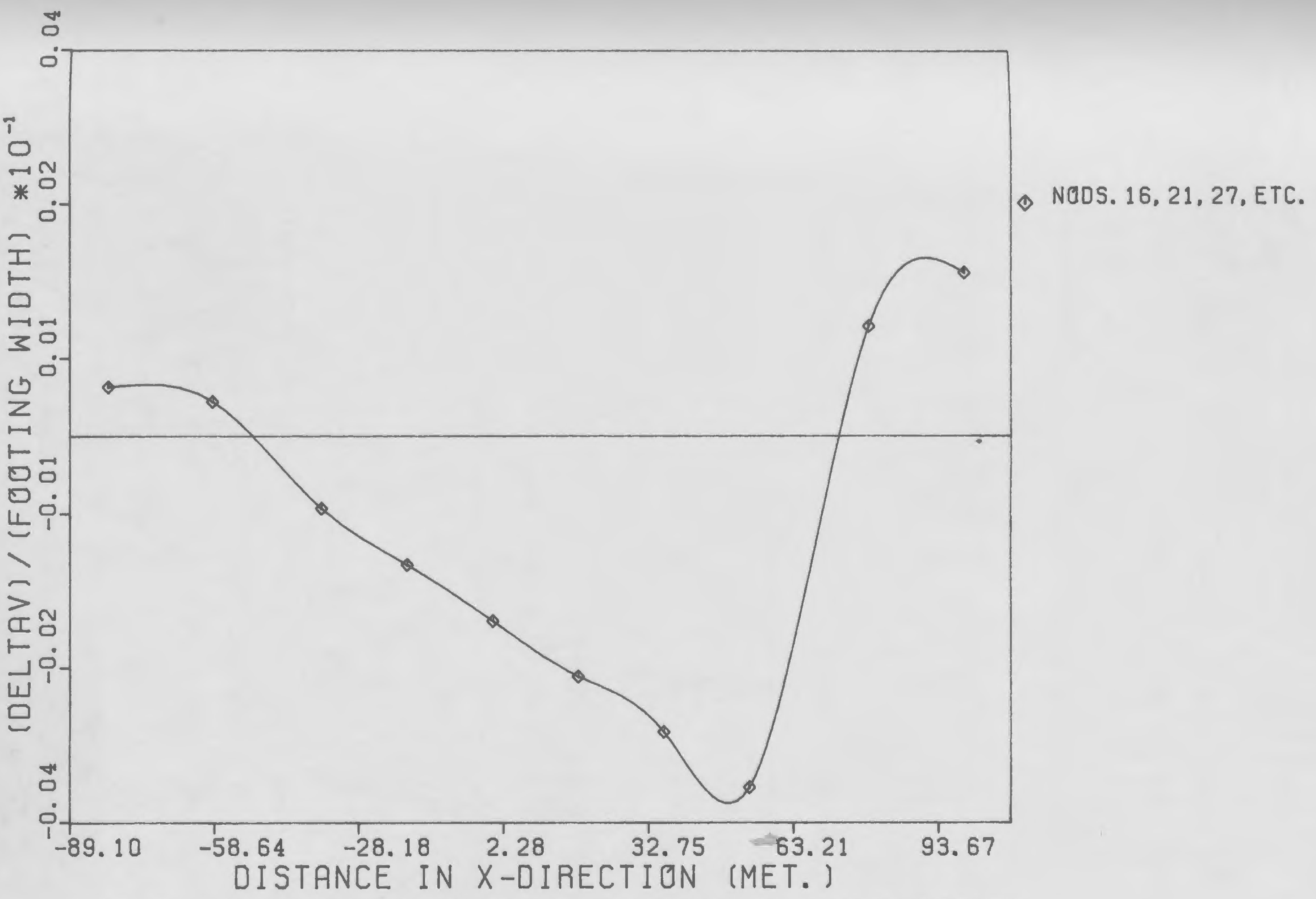


FIG. 6.7(b) DISTRIBUTION OF VER. DISPLACEMENT
(WITH CAISSON; INCLINED LOADING)

footing. The vertical deformations, δ_V, at nodes 39 and 57 below the casisson are shown in Table 6.3 for zero cycle i.e. due to the submerged weight of the structure. However, the vertical deformation, δ_V, at node 57 becomes almost twice that of node 39 for 1/4 cycle. This is due to the fact that at 1/4 cycle, the peak horizontal force, F_H, acts at the top nodes of the caisson structure causing a moment at the base level. This moment induces additional vertical compressive stresses at elements below the right edge of the caisson which result in substantial increase in the vertical deformation. The horizontal deformations at nodes e.g. 27, 33, 39, 45, 51 and 57 beneath the caisson are 0.25% of the footing width. However, the deformations at nodes, 62 and 67, are 11% and 27% less as compared to the deformation at node, 39. The distribution of shear stress beneath the foundation is shown in Fig. 6.8. It is observed that shear stress intensity is high in elements near the right edge of the footing. For example, the shear stress at element, 46 is 2.25 times higher than that compared to element, 32. This is because of the additional shear stress which is induced near the edge due to the static moment, F_M, besides the effect of the horizontal force, F_H. The comparison of normal stress, σ_y,

TABLE 6.3 - COMPUTED ELASTIC DEFORMATIONS
AT VARIOUS LOAD CYCLES

Nodes	Loading Cycles	Deformation in meters	
		Horizontal, δ_H	Vertical, δ_V
39	0	-	-0.1580
	1/4	0.2262	-0.1780
57	0	-	-0.1580
	1/4	0.2271	-0.3295

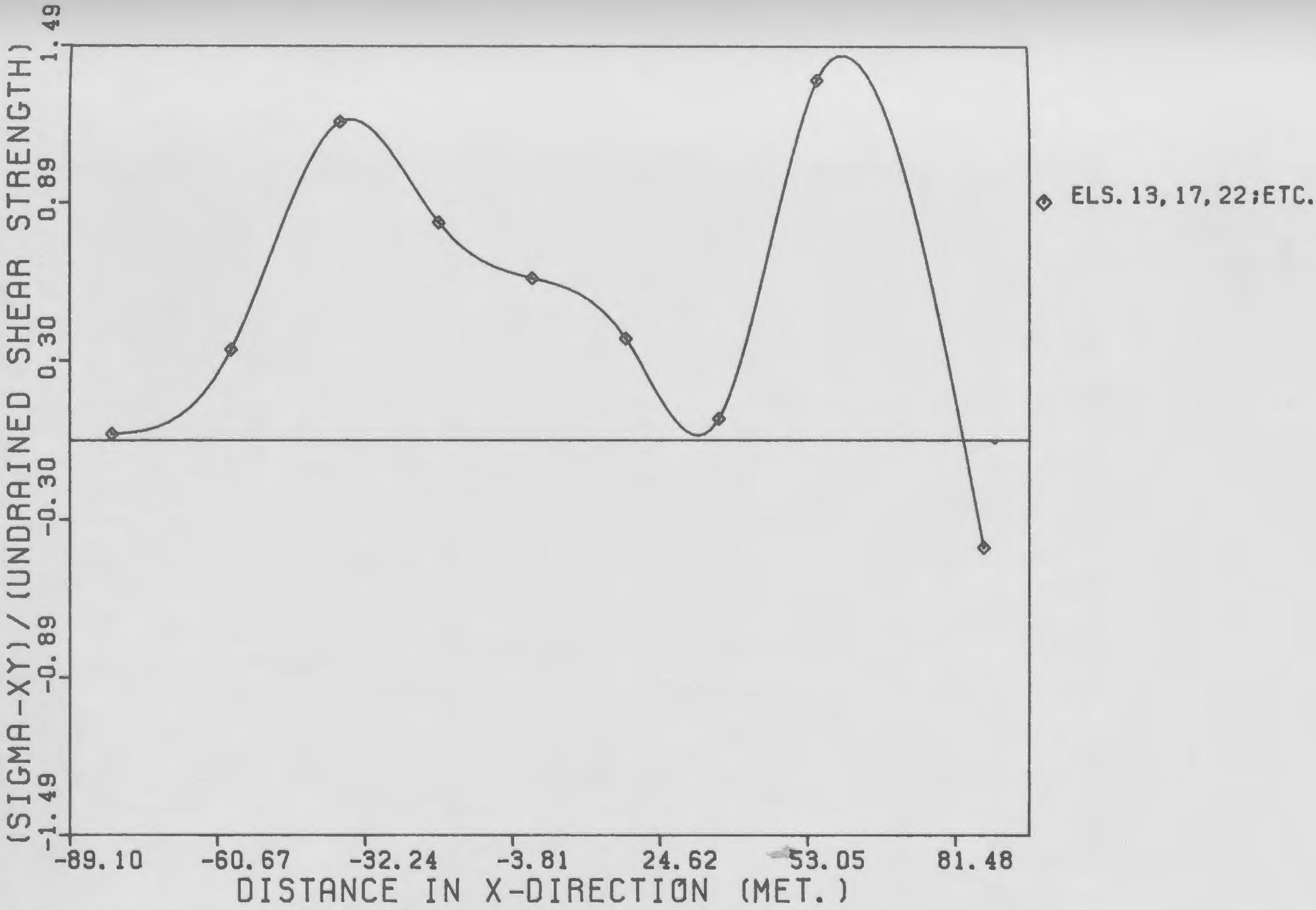


FIG. 6.8 DISTRIBUTION OF SHEAR STRESS
(WITH CAISSON; INCLINED LOADING)

is also made along the depth for the element groups lying on the center line and left of the right edge of the footing. It is seen from Fig. 6.9(a) that the normal stress is higher in elements near the right edge of the caisson. Fig. 6.9(b) depicts the comparison of the distributions of shear stress for the same groups of elements. The contours for shear stress component τ_{xy} are shown in Fig. 6.10 for combined vertical and peak horizontal force which act at the top of the caisson.

6.4.6 Elastic-Plastic Analysis

The nonlinear stress analysis is carried out based on the assumption that the soil behaves as an elastic-perfectly plastic material obeying the Von Mises criteria. The static shakedown load factor obtained in Section 6.4.4 provides the loading domain at shakedown. This load is then applied uniformly at the top nodes of the caisson. The static vertical load, F_V is also applied in a distributed manner at those nodes.

The entire analysis is carried out for 5 cycles of loading i.e. for 60 seconds; 420 load steps are used in order to trace the complete loading path. The uniaxial compressive strength, $\bar{\sigma}$, of soil

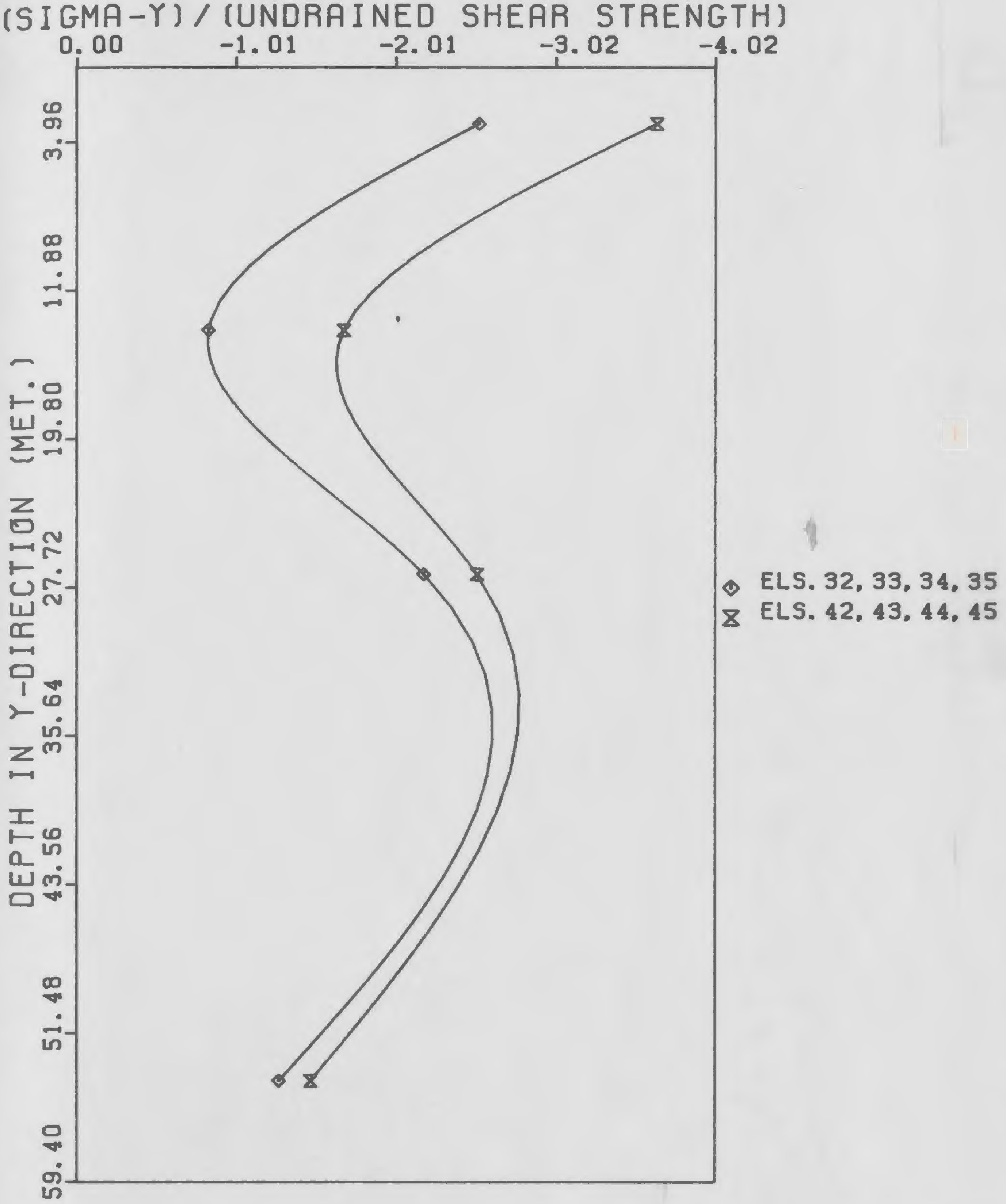


FIG. 6. 9 (a) DISTRIBUTION OF NORMAL STRESS (WITH CAISSON; INCLINED LOADING)

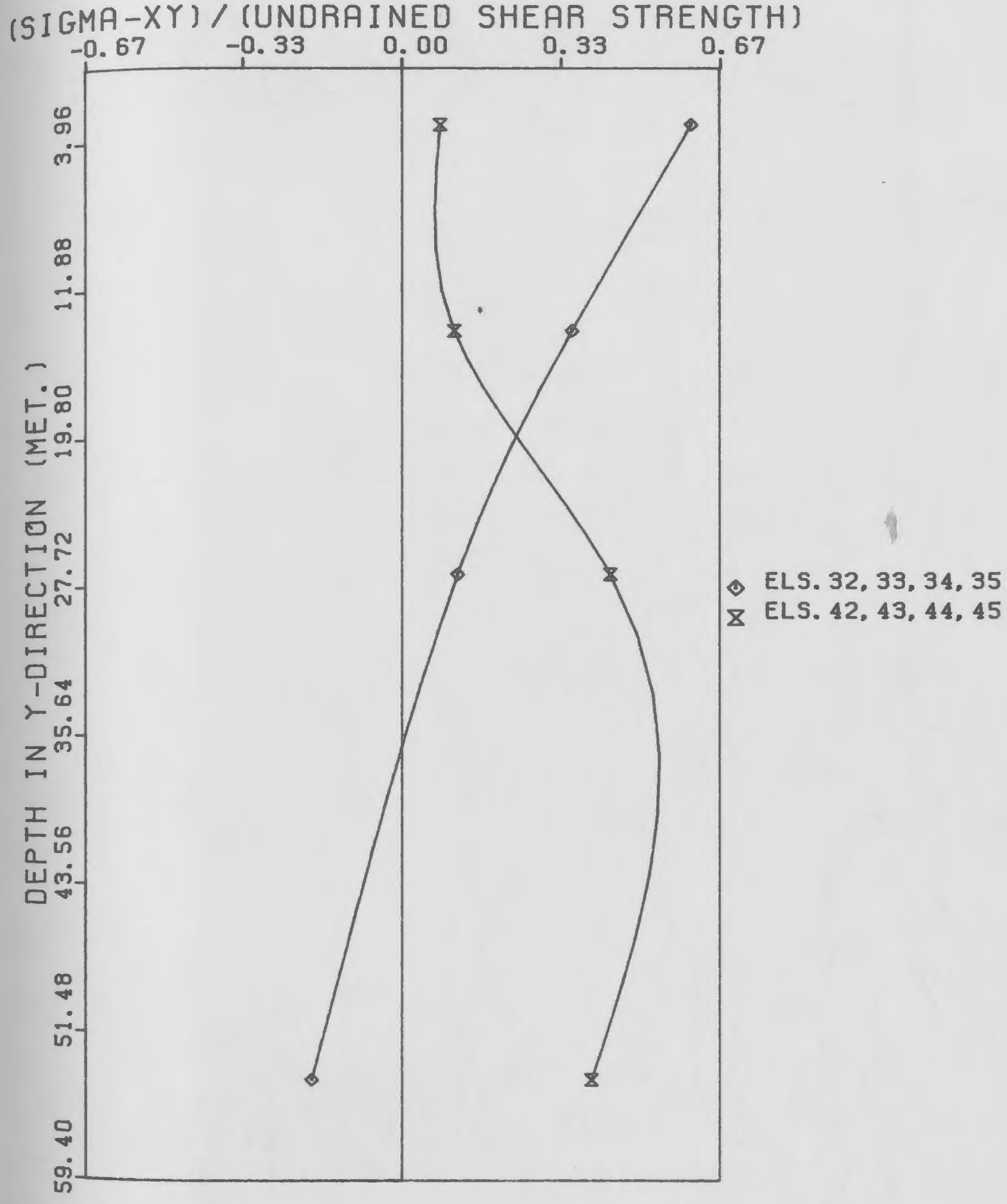


FIG. 6. 9 (b) DISTRIBUTION OF SHEAR STRESS (WITH CAISSON; INCLINED LOADING)

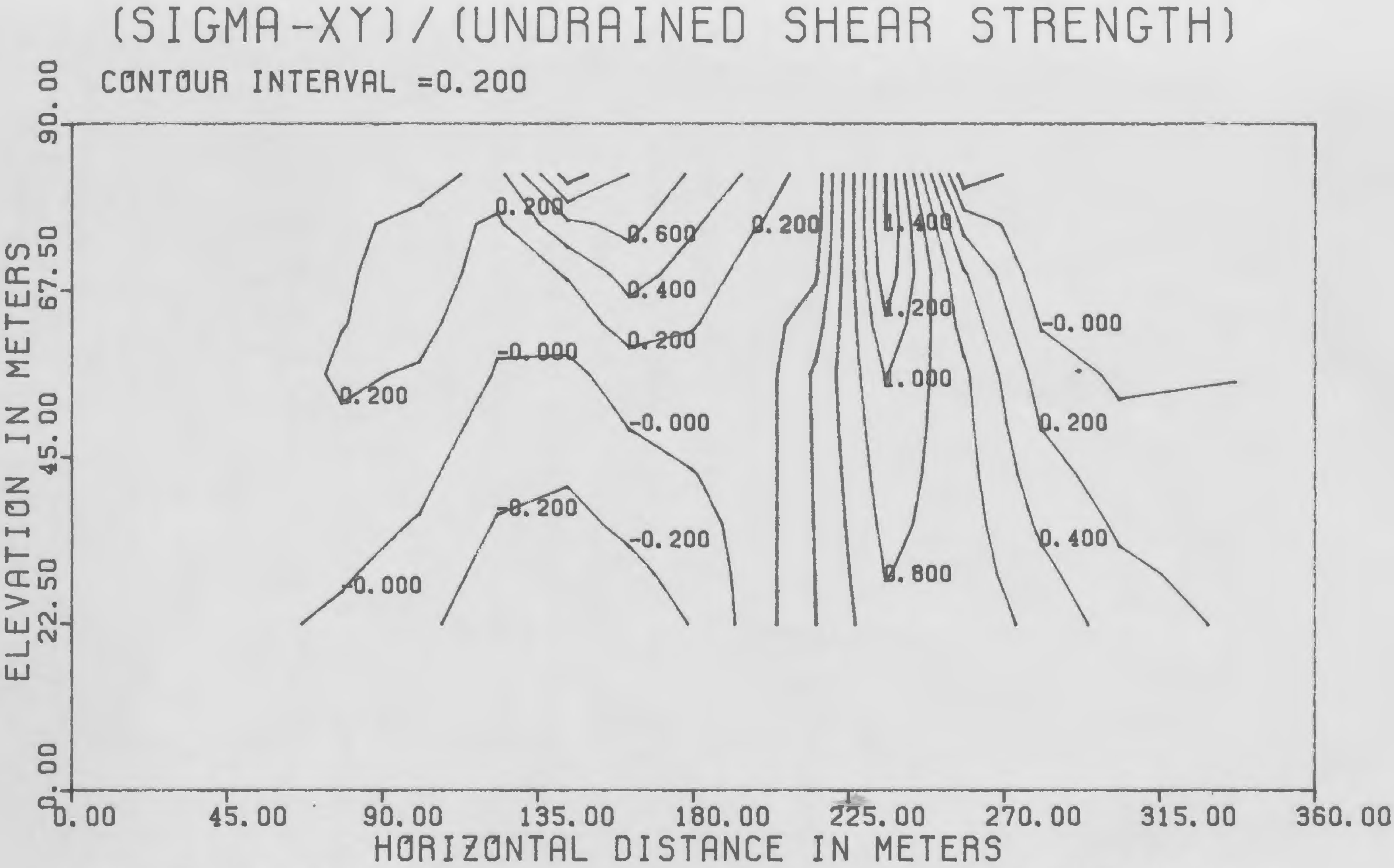


FIG. 6.10 STRESS CONTOURS PLOT FOR (SIGMA-XY)
(WITH CAISSON; INCLINED LOADING)

beneath the foundation is related to the undrained unsoftened static shear strength, s_u, as $\sqrt{3}\, s_u$ for plane strain condition. The caisson is modelled with the compressive strength of concrete and a low value of angle of internal friction, ϕ_u. The properties of the soil and the concrete are shown in Table 6.4.

The horizontal and vertical deformations beneath the caisson due to the cyclic wave loading is shown in Figs. 6.11 and 6.12 for various loading cycles. They show clearly that the deformations along horizontal and vertical directions have stabilized (shakedown) after second loading cycle. Virtually there is no increase in the deformations in the subsequent cycles. This is in agreement qualitatively with the experimental observation made by Rowe (1975) on small scale model tests and described in Section 2.4.2.

Table 6.5 shows the horizontal and vertical deformations for two nodes, 39 and 57, at cycles 0, 1/4, 1, 3 1/4, 4, 4 1/4 and 5 cycles. The vertical deformation, δ_V, due to the submerged weight of the structure is 0.1580 meter ie. 0.18% of the width of the footing. It is seen from the above Table 6.5 that vertical deformations at nodes 39

TABLE 6.4 - MATERIAL PROPERTIES FOR THE CAISSON STRUCTURE AND THE SATURATED SOIL MEDIUM

Notations	Caisson	Soil
λ	4200.0 MPa	14.7 MPa
μ	9800.0 MPa	9.80 MPa
α	-	1.0
K_w	-	2000.0 Mpa
ρ	22.0 KN/m^3	19.0 KN/m^3
ρ_f	-	9.8 KN/m^3
n	-	0.30
ν_d	0.15	0.30
ν_u	-	0.499
c'	15.00 MPa	100.0 KPa
ϕ_u	10^o	-

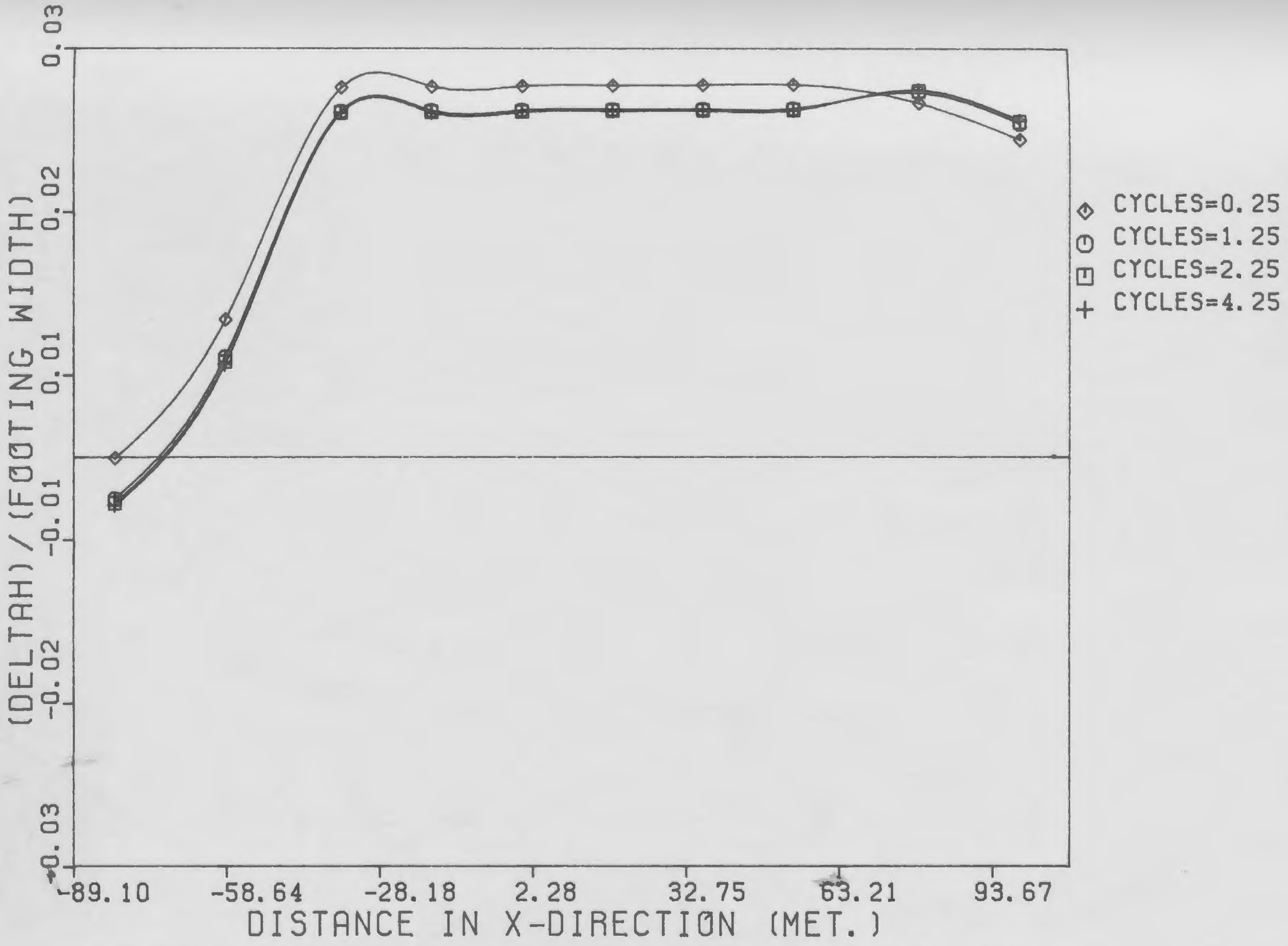


FIG. 6.11 DISTRIBUTION OF HOR. DISPLACEMENT
(STIFF FOOTING; ELASTIC PLASTIC ANALYSIS)

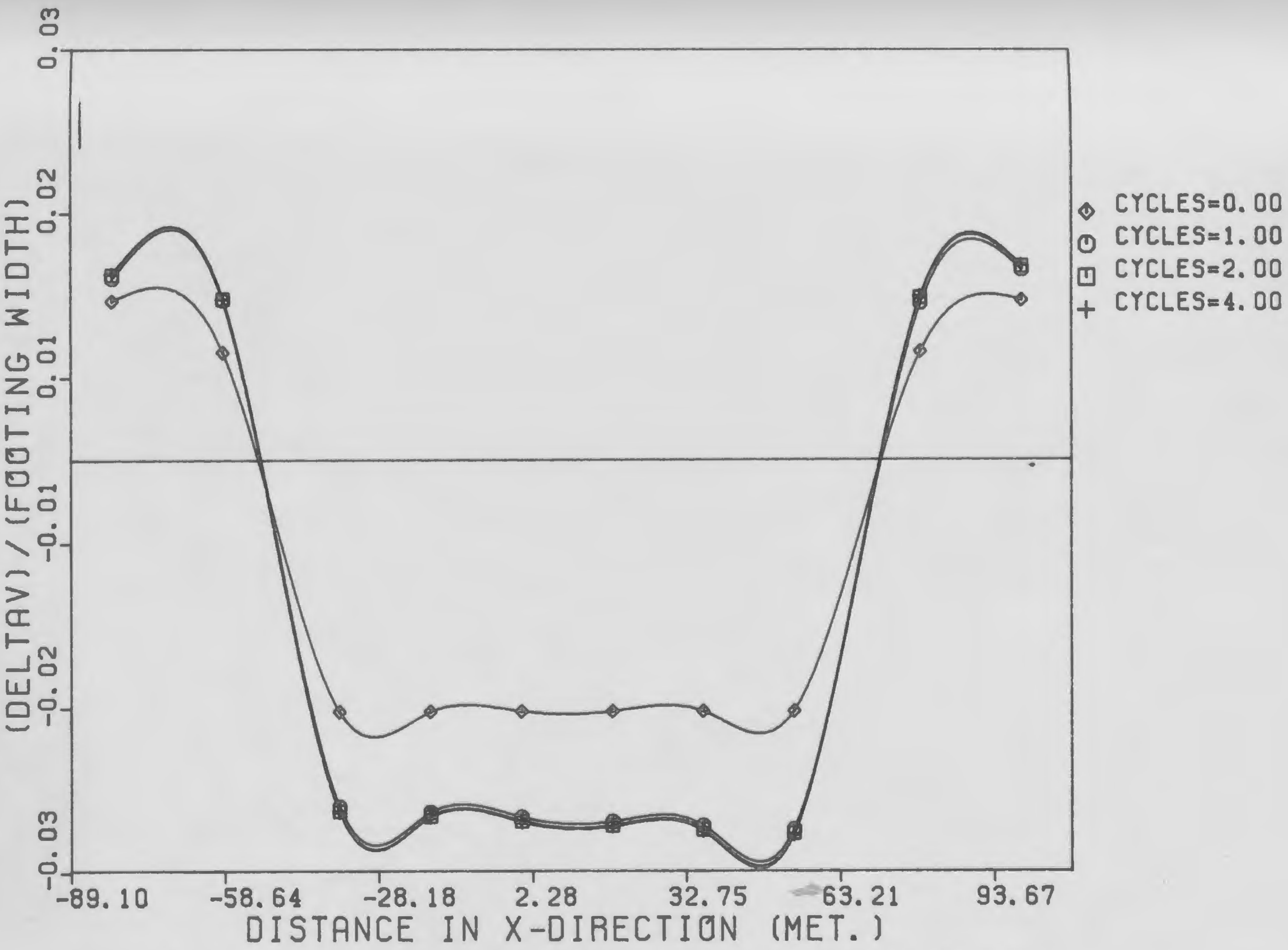


FIG. 6.12 DISTRIBUTION OF VER. DISPLACEMENT
(STIFF FOOTING; ELASTIC PLASTIC ANALYSIS;)

TABLE 6.5 - COMPUTED DEFORMATIONS (IN METERS) FOR LINEAR AND NONLINEAR ANALYSES

		N = 0		N = 1/4		N = 1		N = 3 1/4		N = 4		N = 4 1/4		N = 5	
	Types of Analyses	δ_H	δ_V	δ_H	δ_V	δ_H	δ_V	δ_H	δ_V	δ_H	δ_V	δ_H	δ_V	δ_H	δ_V
39	Elastic	-	-0.1580	.2262	-0.1780	-	-	-	-	-	-	-	-	-	-
	Elastic -Plastic	-	-0.1580	.2500	-0.1850	-	-0.2300	.2324	-0.2000	-	-0.2280	.2322	-.2021	-	-.2300
57	Elastic	-	-0.1580	.2271	- .3295	-	-	-	-	-	-	-	-	-	-
	Elastic -Plastic	-	-0.1580	.2500	- .3800	-	-0.2327	.2332	- .3660	-	-0.2361	0.2330	-.3663	-	-.2360

and 57 have stabilized to a value of 0.2291 meter and 0.2314 meter respectively. These deformations are 45% higher than the inital settlement at zero load cycle.

The soil elements 22, 27 and 46 beneath the caisson yielded at peak horizontal load. The distribution of shear stress beneath the caisson for selected elements is shown in Fig. 6.13. Fig. 6.14 depicts the distribution of shear stress along the depth for the elements lying just on the centerline. Fig. 6.15 shows similar distributions along the depth for those elements near the left of the right edge of the caisson structure. It is seen that shakedown has taken place with a redistribution of shear stress once the elements have yielded and as the load passes from first cycle to the subsequent cycle.

6.4.7 Comparison of Elastic and Elastic-Plastic Analyses

The comparison of the two static analyses, viz elastic and elastic-plastic are presented at 1/4 load cycle when the horizontal force is at its peak value. Figs. 6.16(a) and (b) show the comparisons for horizontal and vertical deformations for the two analyses. For node 57, the horizontal

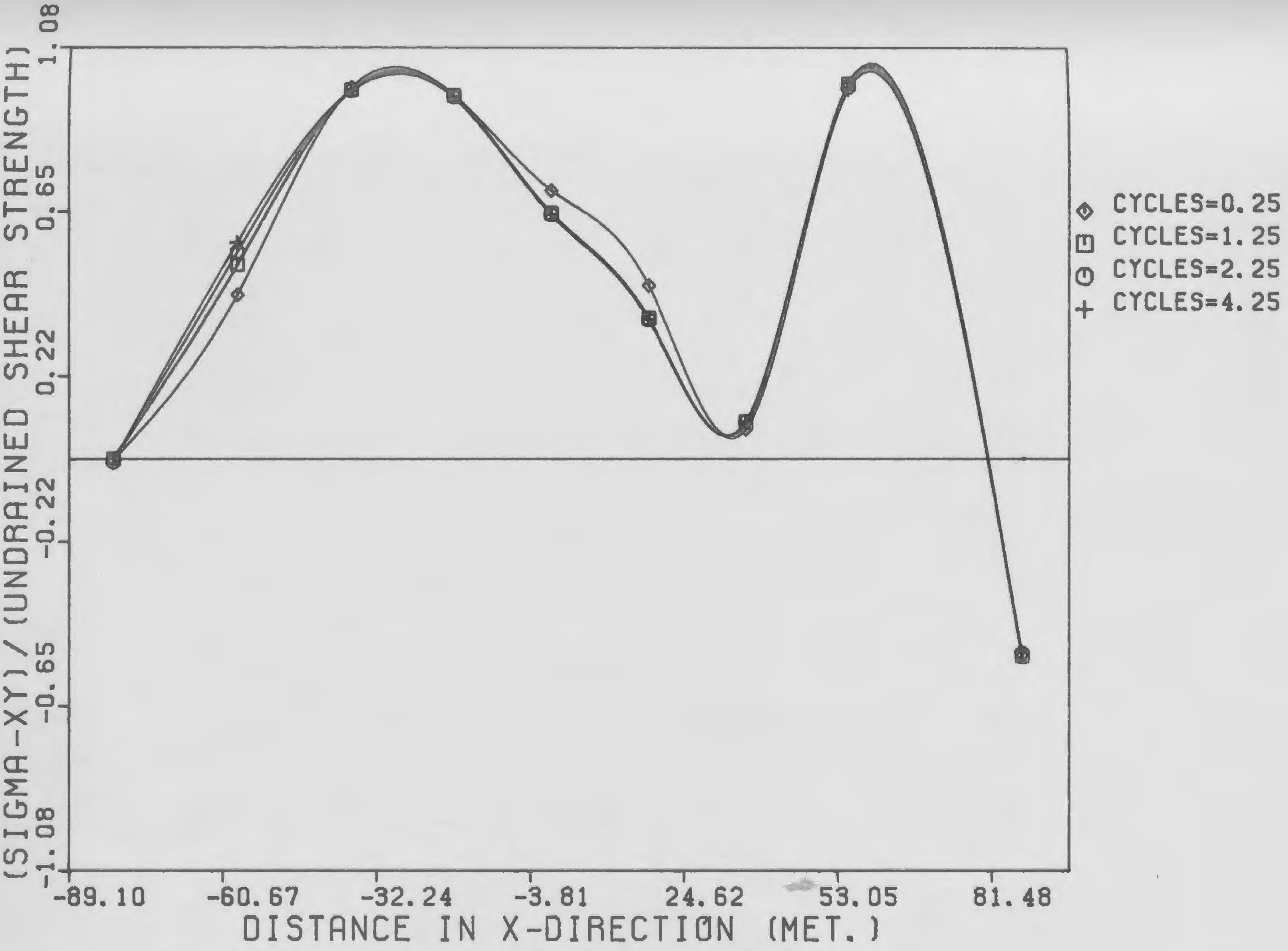


FIG. 6.13 DISTRIBUTION OF SHEAR STRESS
(WITH CAISSON; ELASTIC PLASTIC ANALYSIS)

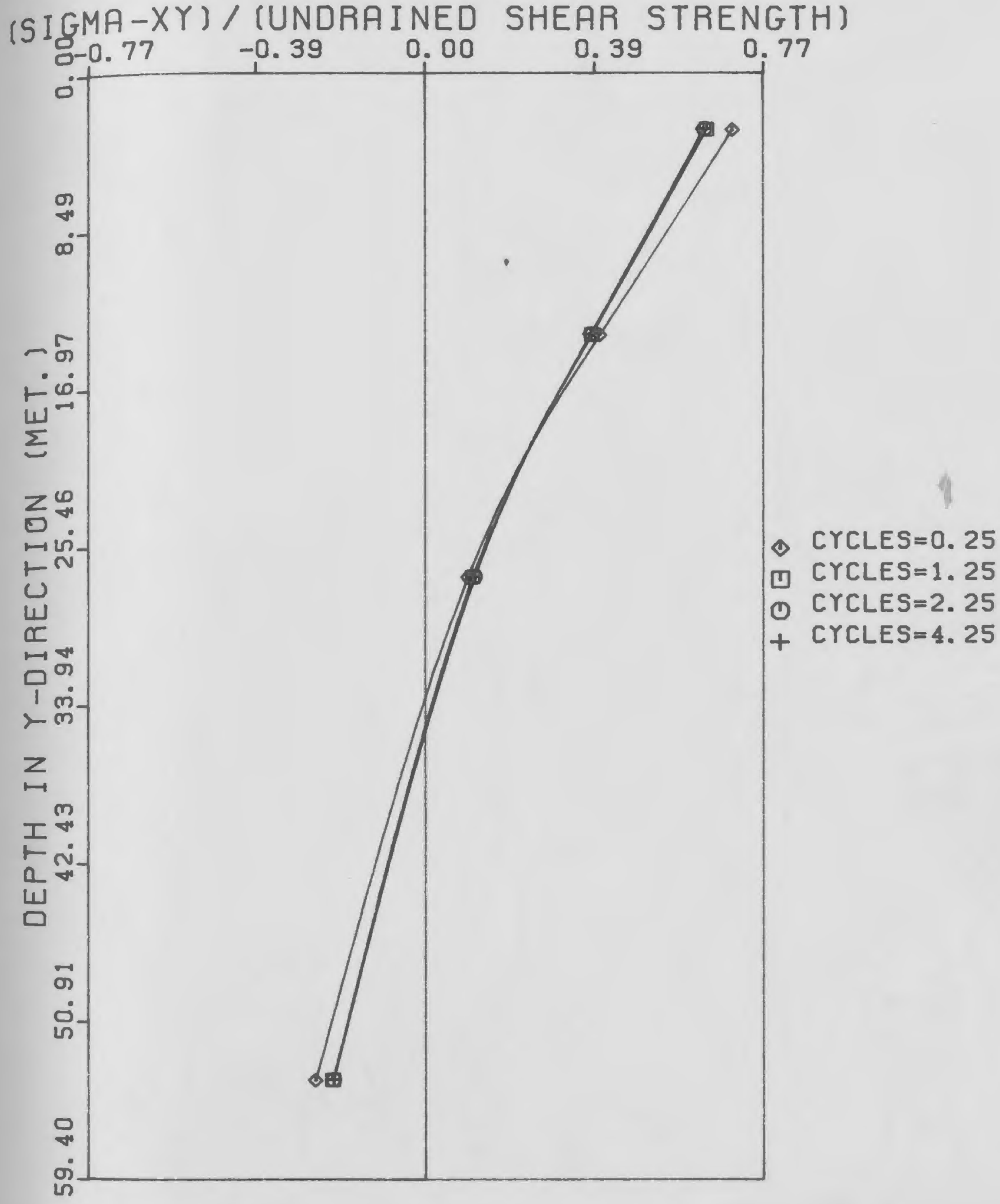


FIG. 6.14 DISTRIBUTION OF SHEAR STRESS
(WITH CAISSON; NONLINEAR ANALYSIS)
(ELEMENTS 32, 33, 34, 35)

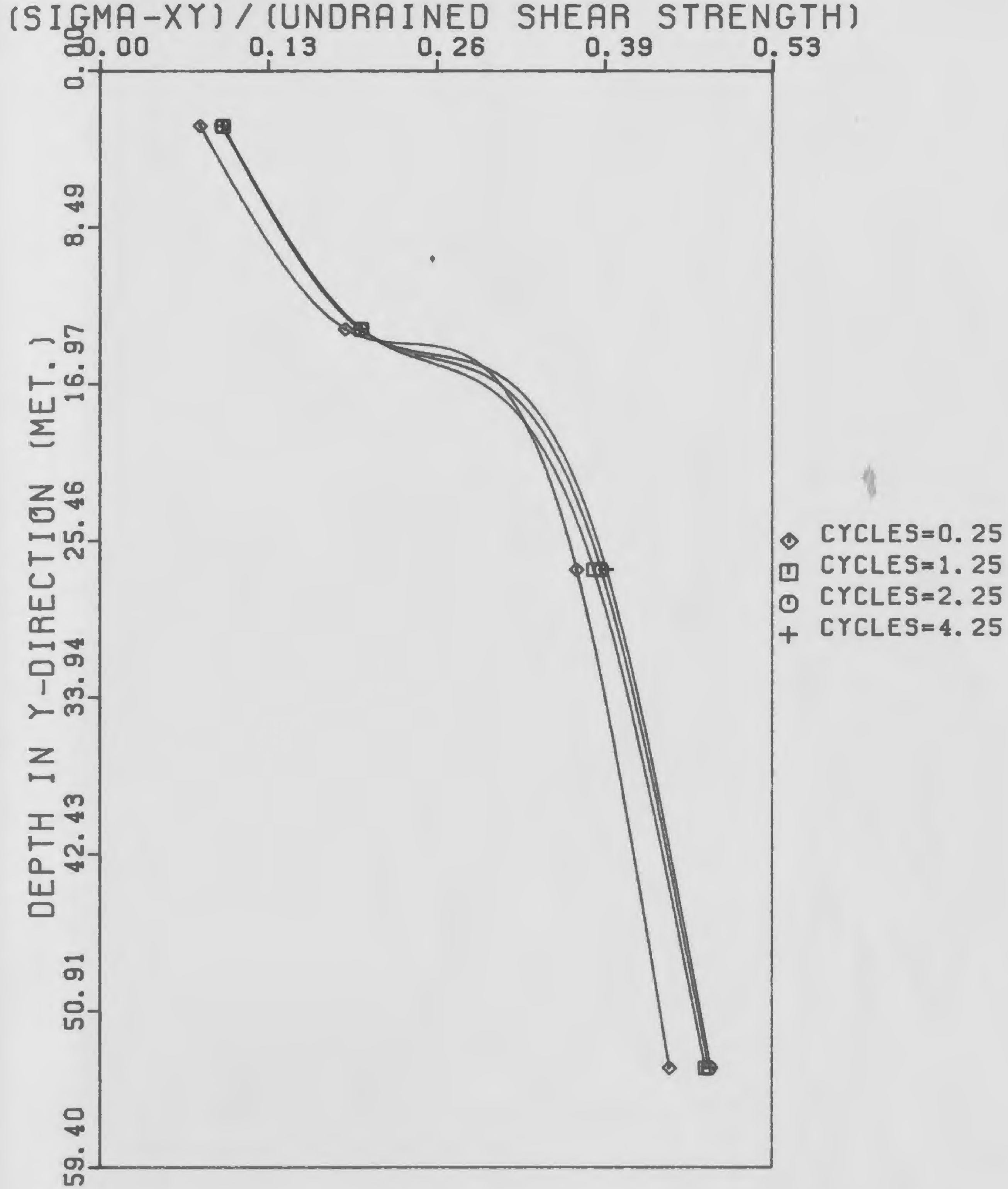


FIG. 6. 15 DISTRIBUTION OF SHEAR STRESS
(WITH CAISSON; NONLINEAR ANALYSIS)
(ELEMENTS 42, 43, 44, 45)

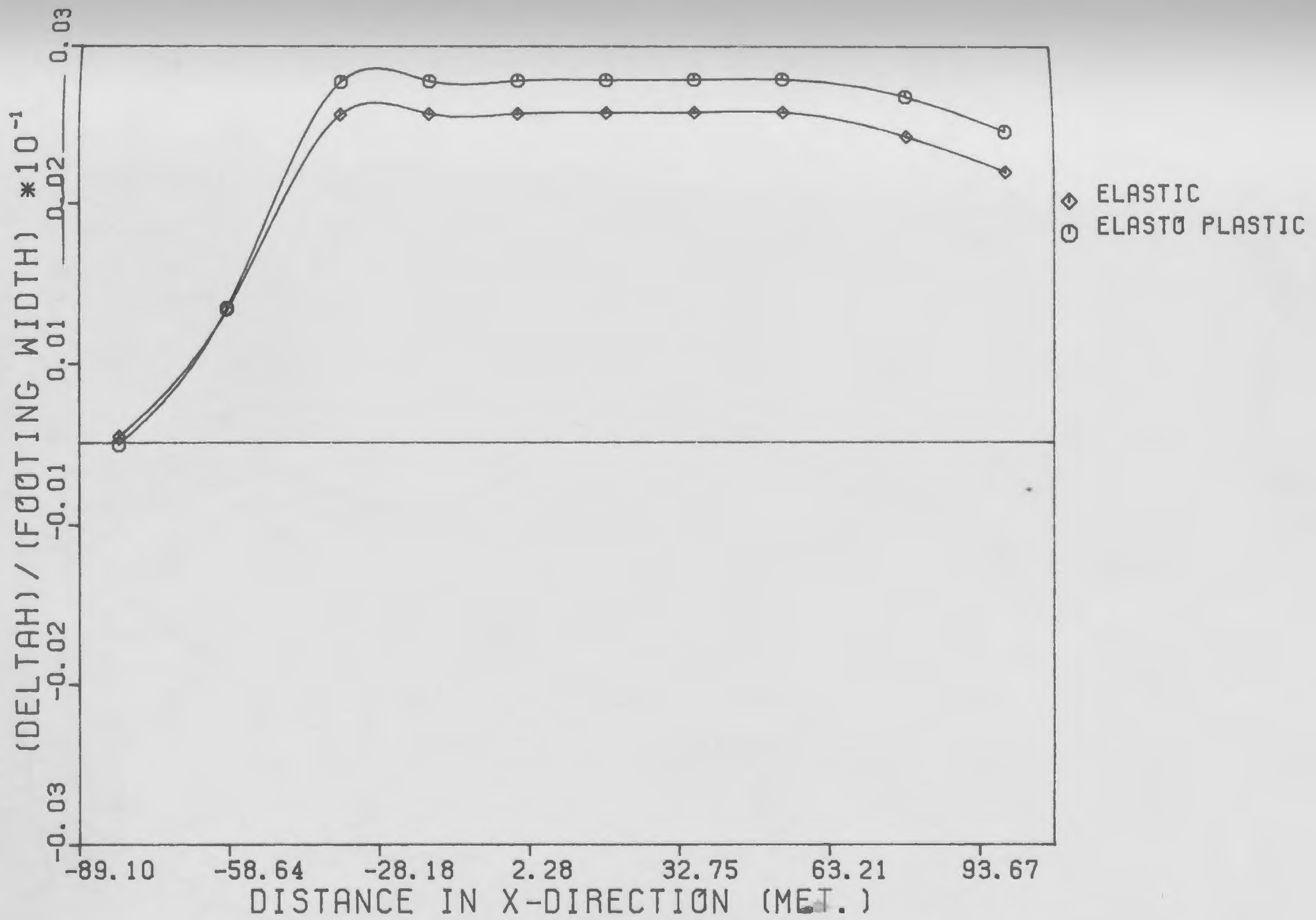


FIG. 6.16(a) DISTRIBUTION OF HOR. DISPLACEMENT
(WITH CAISSON; COMPARISON OF ANALYSES)
(AT 1/4 CYCLE)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC)

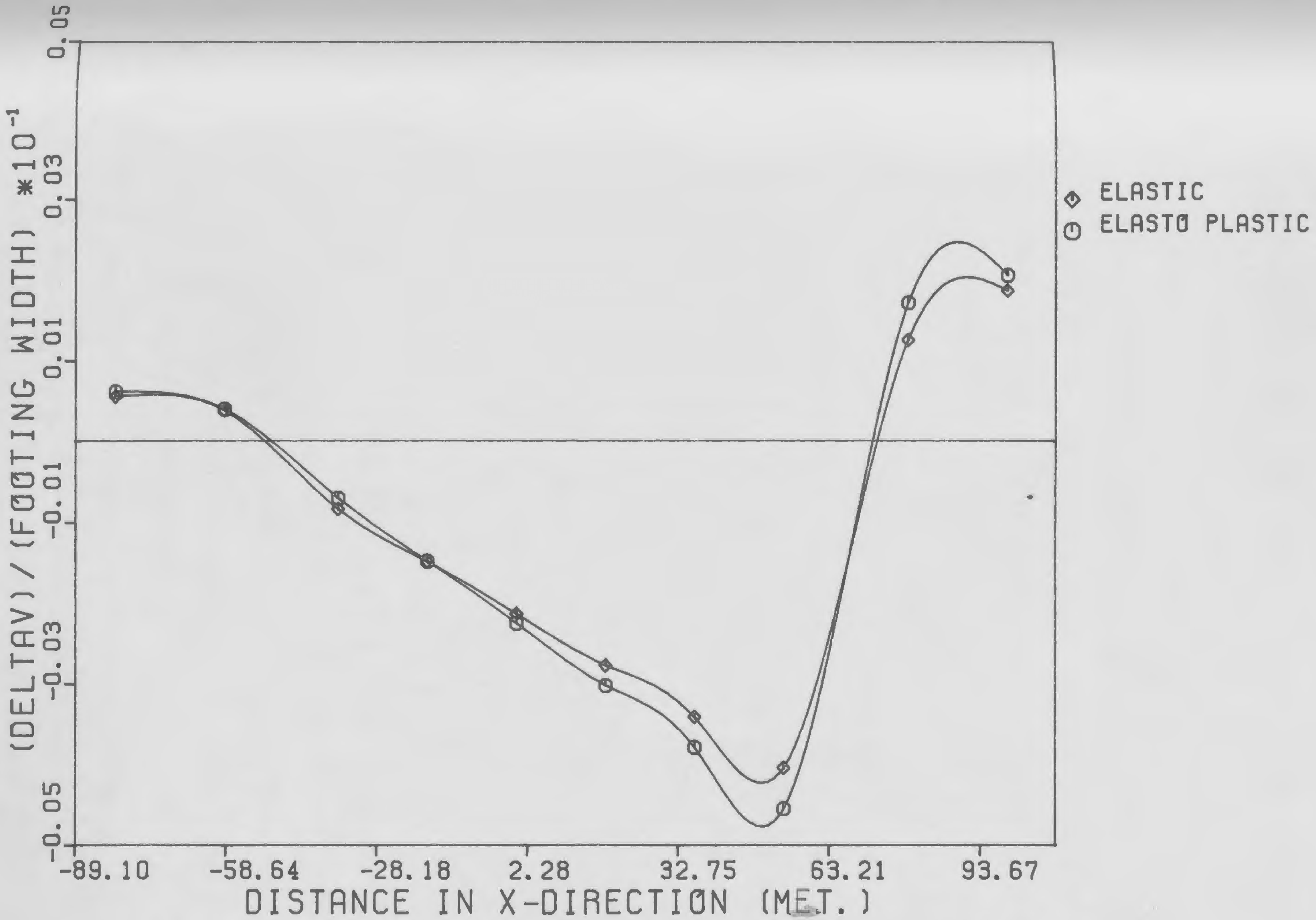


FIG. 6.16(b) DISTRIBUTION OF VER. DISPLACEMENT
(WITH CAISSON; COMPARISON OF ANALYSES)
(AT 1/4 CYCLE)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC)

and vertical deformations from elastic-plastic analysis are 12% and 13% higher than those obtained by elastic analysis, since some soil elements beneath the caisson have yielded. Fig. 6.17 shows the distribution of δ_H along the depth. Fig. 6.18 compares the shear stress distibution beneath the foundation for the two analyses. The shear stress predicted by elastic-plastic analysis is lower than that predicted by the elastic analysis. For example, the elastic shear stress at element, 46 is 45% higher than obtained from a nonlinear analysis. Fig. 6.19 compares the shear stress along the depths for elements lying along the centerline.

6.5 Dynamic Analysis of a Gravity Type Offshore Foundation

The present investigation considers the dynamic interaction between the caisson structure of a gravity base offshore platform and its foundation soil which consists of saturated clay underlaid by a rigid base. The caisson structure is assumed to be linear whereas the nonlinear behaviour of the soil is taken into account by two different types of analyses: a) equivalent linear model and b) elastic-perfectly plastic model. The cyclic wave

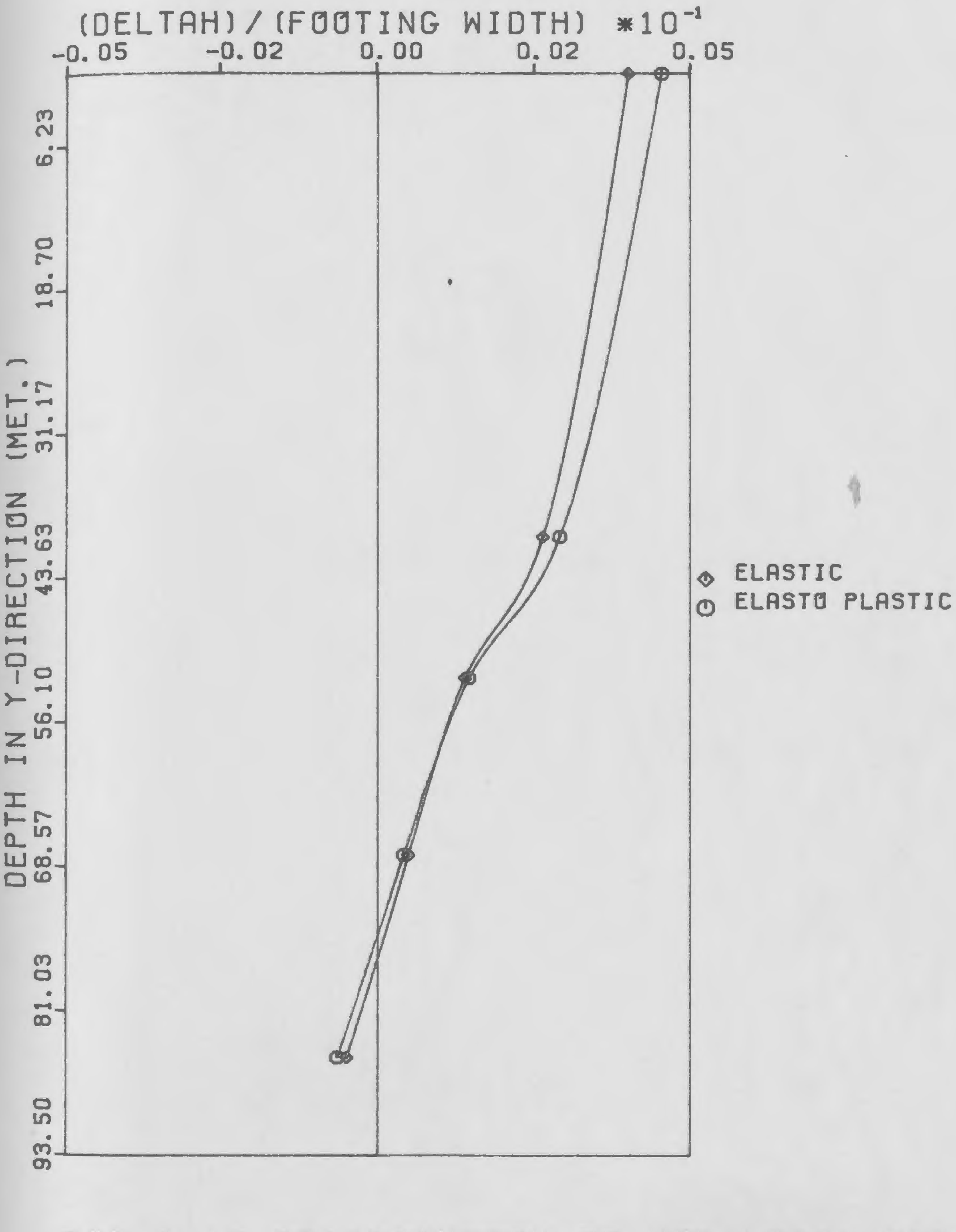


FIG. 6.17 DISTRIBUTION OF HOR. DISPLACEMENT
(WITH CAISSON; COMPARISON OF ANALYSES)
(NODES 38, 39, 40, 41, AND 42)

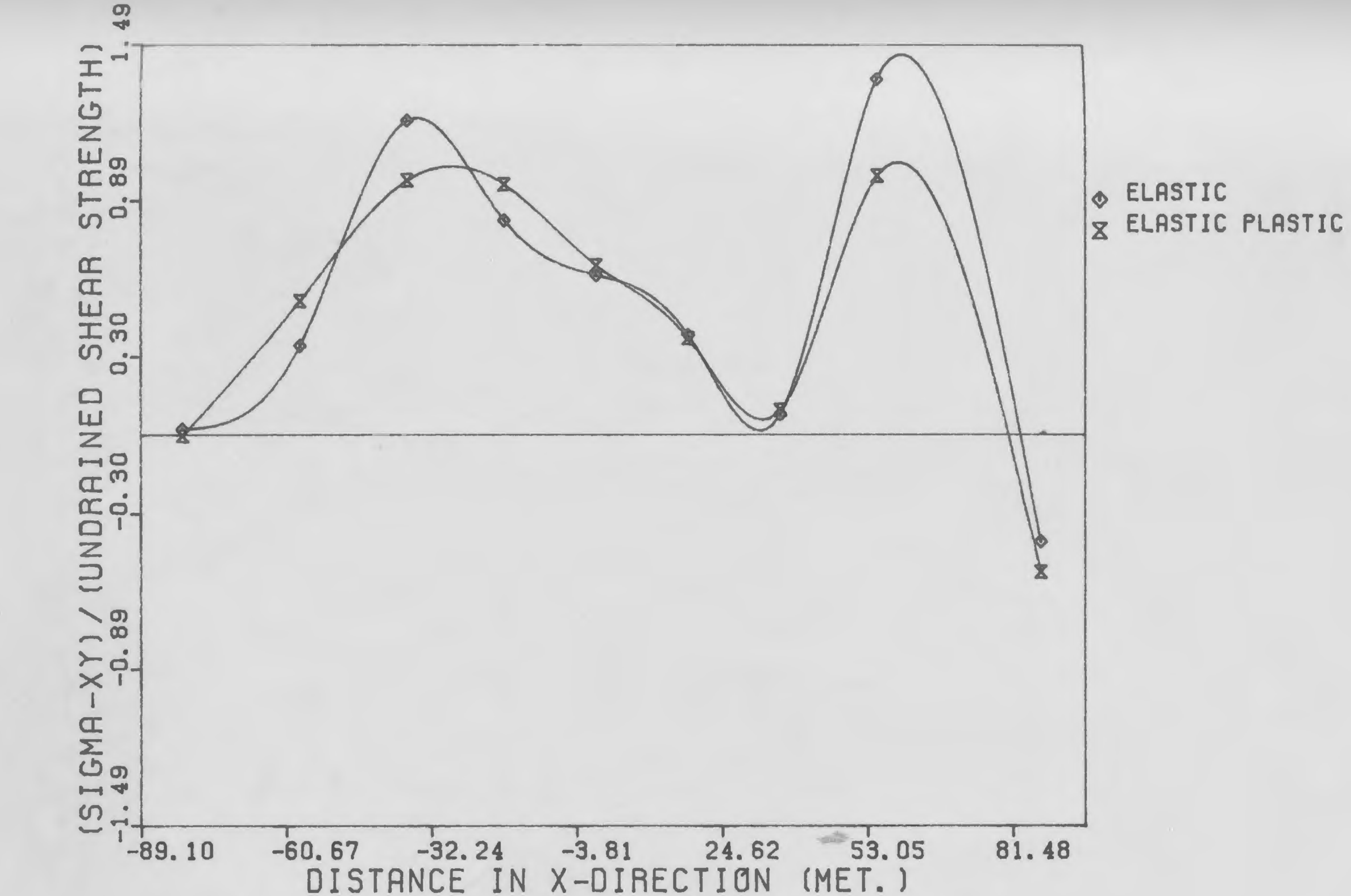


FIG. 6. 18 DISTRIBUTION OF SHEAR STRESS
(WITH CAISSON; COMPARISON OF ANALYSES)
(AT 1/4 CYCLE)
(ELEMENTS 13, 17, 22, 27, 32, 37, 42, 46, 50)

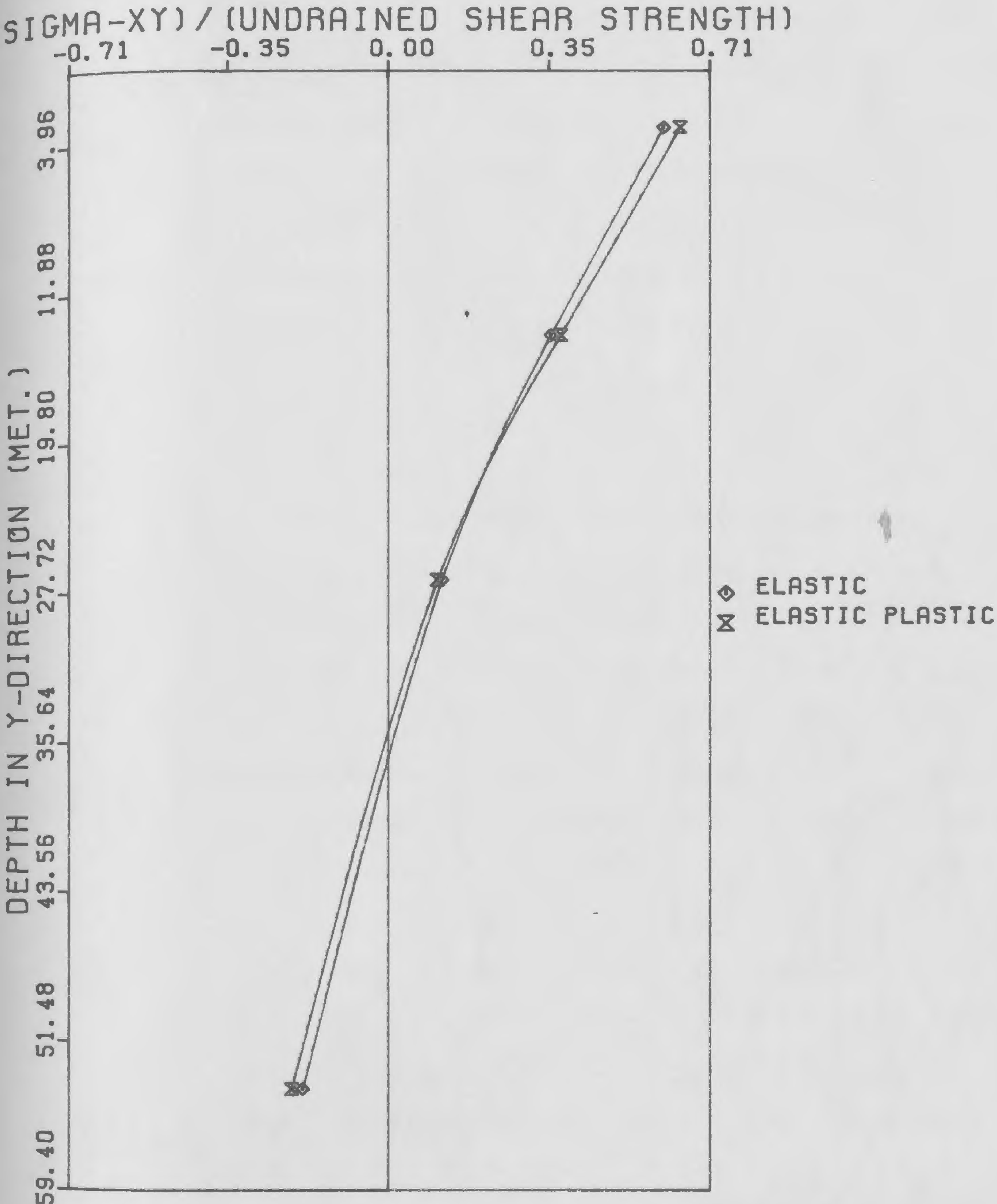


FIG. 6.19 DISTRIBUTION OF SHEAR STRESS
(WITH CAISSON: COMPARISON OF ANALYSES)
(32, 33, 34, 35; CYCLE=0.25)

loading function is simulated here as sinusoidal forcing function, with a 12 seconds period, acting at the top of the caisson's nodes thus causing a dynamic cyclic moment at the foundation level. The submerged weight, F_V, is assumed to be static and remains constant during the entire loading cycle.

6.5.1 Dynamic Shakedown Analysis

In order to carry out the dynamic shakedown analysis, the elastic stress responses of the structure-foundation system have to be determined. The time history of response quantities are computed by an implicit step-by-step integration scheme which is primarily dependent on the time step interval (Δt) chosen. The time step interval is a function of the system frequencies and Table 6.6 shows the computed frequencies of the caisson-foundation system for first ten modes under drained and undrained conditions. The drained condition is simulated by setting the bulk modulus of water as zero. The fundamental frequency in the drained state is only 12% less than that in undrained state. However, for higher modes, the difference is considerably large. For example, the frequency at eighth mode for drained state is 31% less than

TABLE 6.6 - UNDRAINED AND DRAINED FREQUENCIES OF THE CAISSON-FOUNDATION SYSTEM

MODE NUMBERS	UNDRAINED FREQUENCIES (rad/sec)	DRAINED FREQUENCIES (rad/sec)
1	1.53	1.34
2	2.23	1.86
3	2.24	1.94
4	2.93	2.42
5	2.96	2.44
6	3.20	2.75
7	3.73	2.88
8	3.80	2.90
9	4.33	3.14
10	4.41	3.63

that in undrained state. This is because of the fact that under undrained condition, incompressibility constraint on volumetric mode of deformation introduces additional stiffness.

Although theoretically one should take the period associated with the highest mode, it is known from practice that higher modes contribute very little to the response process. Besides, soil foundation system has a typical frequency ratio

$$\omega^* = \frac{\omega_o}{\omega_1} = 0.34$$

where

ω_0 = wave frequency = 0.523 rad/sec.

and ω_1 = fundamental frequency of the structure-foundation system

= 1.53 rad/sec. (Refer Table 6.6)

The time step of 0.1429 second is chosen based on one tenth of the period associated with the tenth mode of the structure-foundation system. This will result in 420 time steps in order to describe the five cycles of loading (60 seconds duration) adequately.

The Rayleigh damping coefficients are computed based on the specified modal damping ratios in first and second mode. The critical damping values are assumed as 15% and 20% in mode 1 and mode 2 respectively. Based on the above values, the Rayleigh damping coefficients, B_1 and B_2, for the caisson and the soil medium are computed and shown in Table 6.7.

6.5.1.1 Dynamic Elastic Analysis

The elastic dynamic response is computed first with zero initial conditions for five cycles of loading and the displacements and velocities for each node are saved at the end of the loading process.

Computed displacements and velocities which have been saved are used as fictitious initial conditions, $\underline{\bar{U}}^*$, and $\underline{\dot{\bar{U}}}^*$ and the normalized maximum stress response vector is computed from Eqn. (4.43). Finally, the minimization of Eqn. (4.55) is carried out once again using the above stress vector and the shakedown load factor, β_s, obtained as 1.15. This is 10% lower than the coresponding value obtained from the static analysis (1.28). This is because of the fact that stress amplification has occurred at the frequency ratio of $\omega^* = 0.34$ and as such the shakedown load factor has decreased. Based on this load factor, $\beta_s =$

TABLE 6.7 - RAYLEIGH DAMPING COEFFICIENTS FOR CAISSON STRUCTURE AND SOIL MEDIUM

Rayleigh Damping Coefficients	Caisson	Soil
B_1	-0.340×10^{-2}	0.739×10^{-1}
B_2	0.276×10^{-1}	0.1650

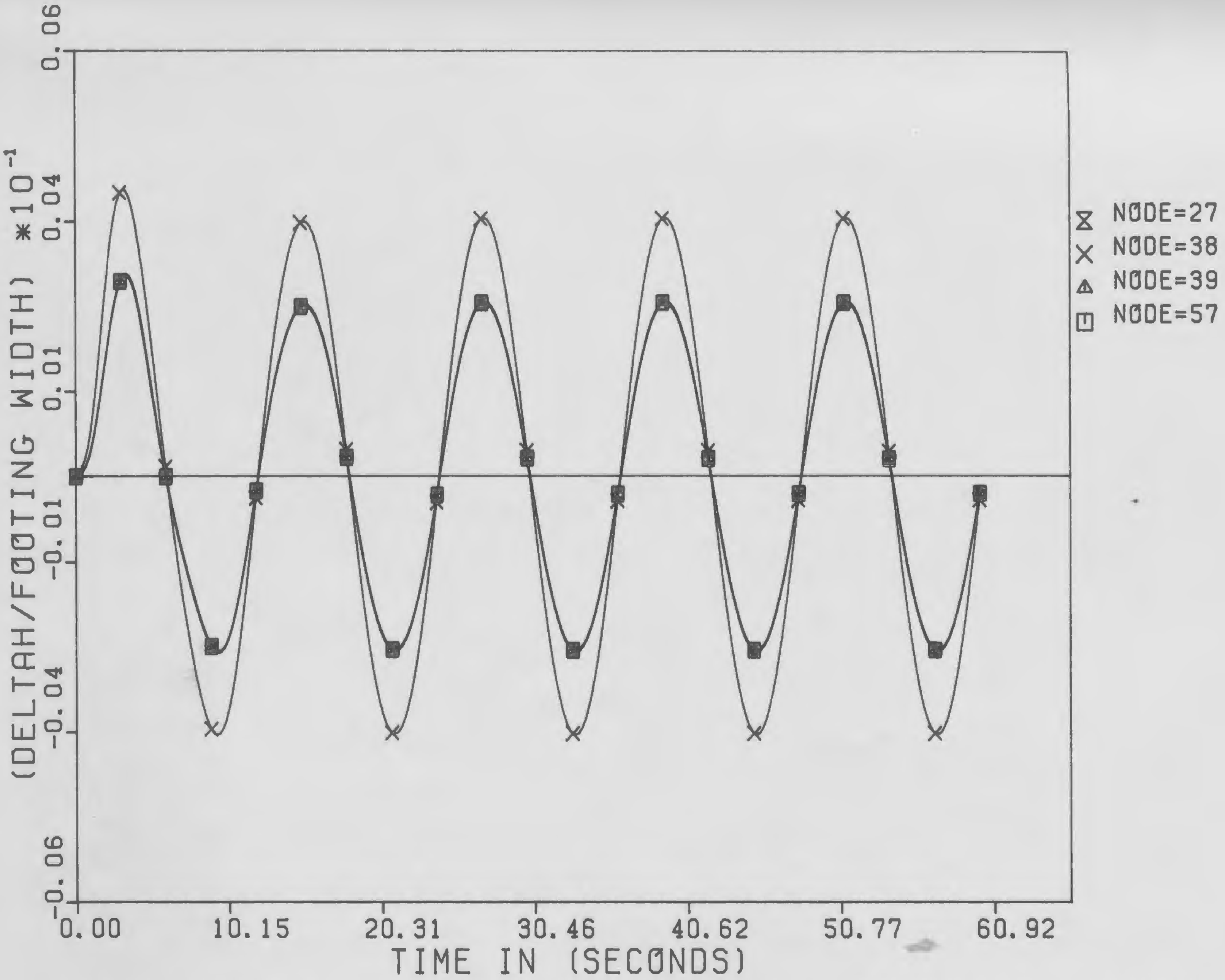


FIG. 6. 20 (a) TIME HISTORY PLOT OF HOR. DISPLACEMENT
(WITH CAISSON; DYNAMIC ELASTIC ANALYSIS)

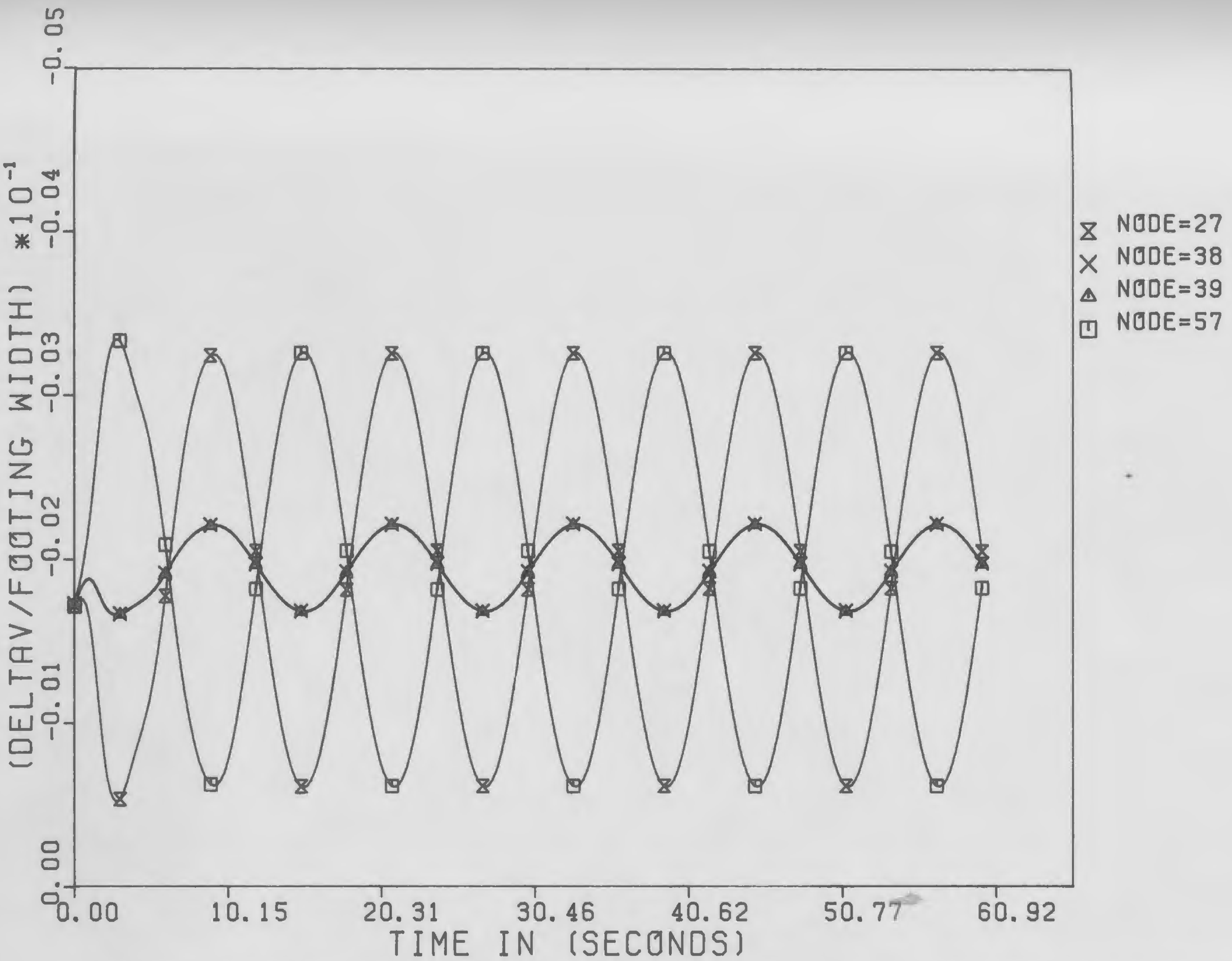


FIG. 6. 20 (b) TIME HISTORY PLOT OF VER. DISPLACEMENT
(WITH CAISSON; DYNAMIC ELASTIC ANALYSIS)

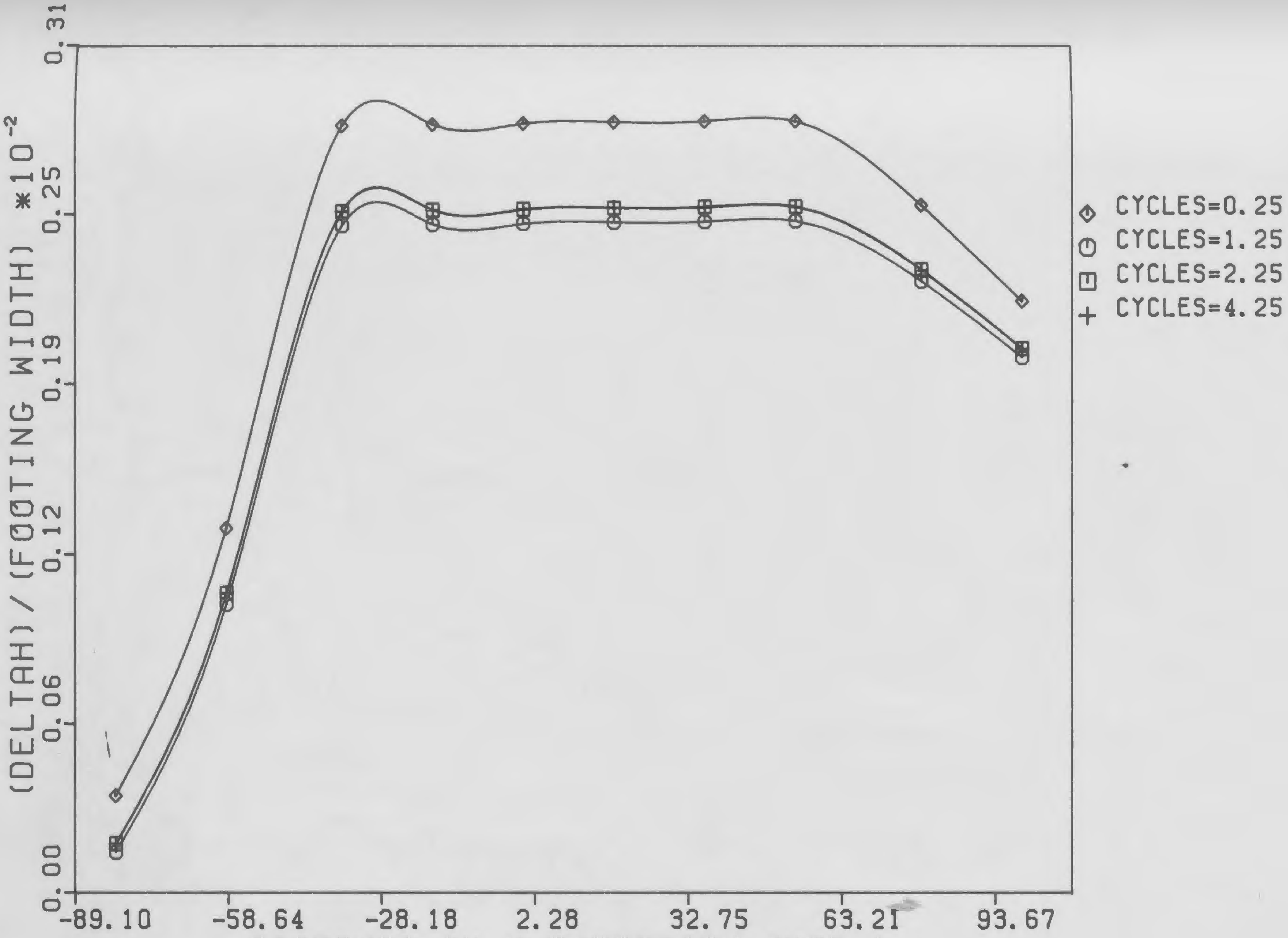


FIG. 6.21 (a) DISTRIBUTION OF HOR. DISPLACEMENT
(WITH CAISSON; DYNAMIC ELASTIC ANALYSIS)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC;)

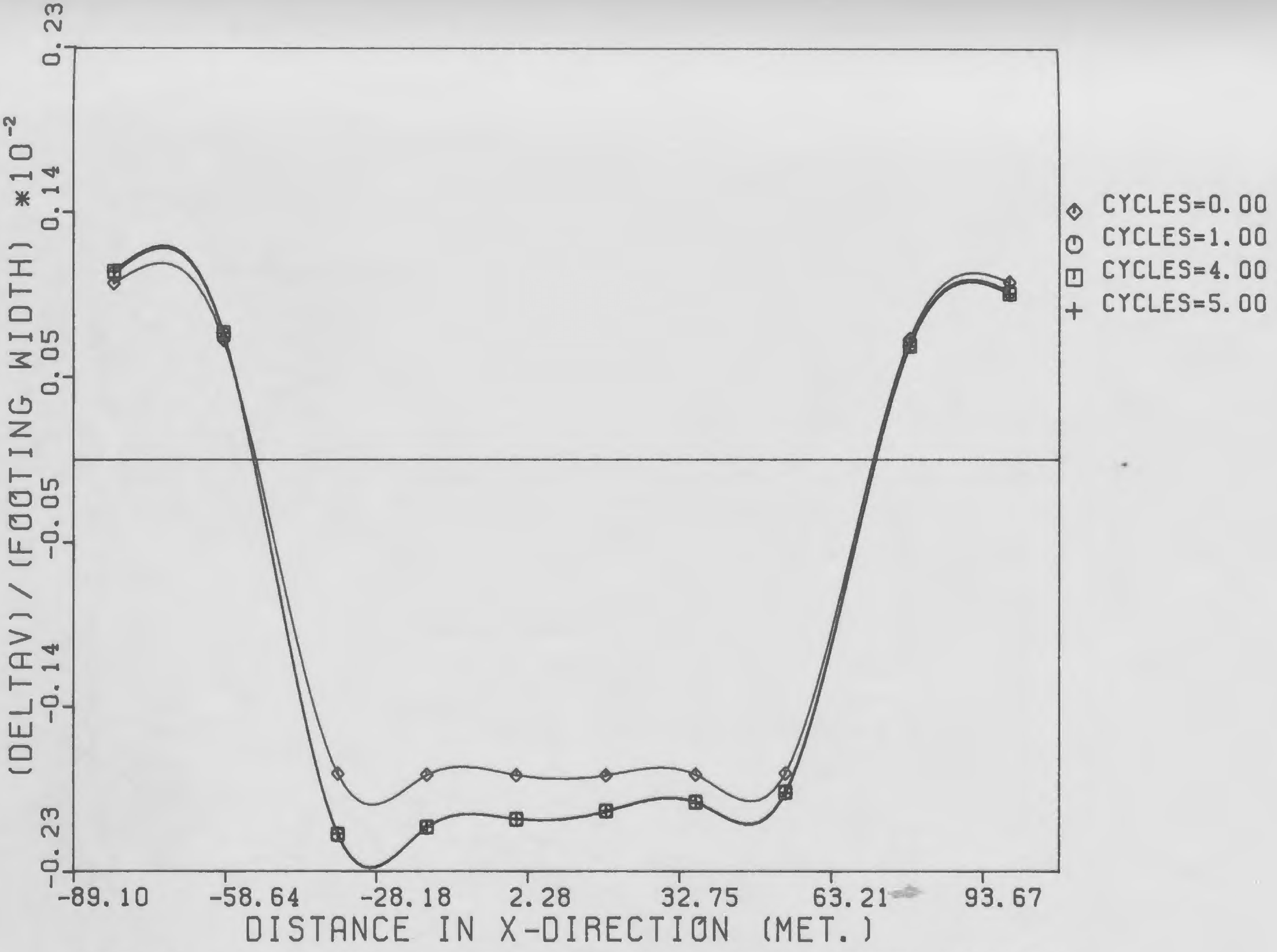


FIG. 6.21 (b) DISTRIBUTION OF VER. DISPLACEMENT
(WITH CAISSON; DYNAMIC ELASTIC ANALYSIS)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC)

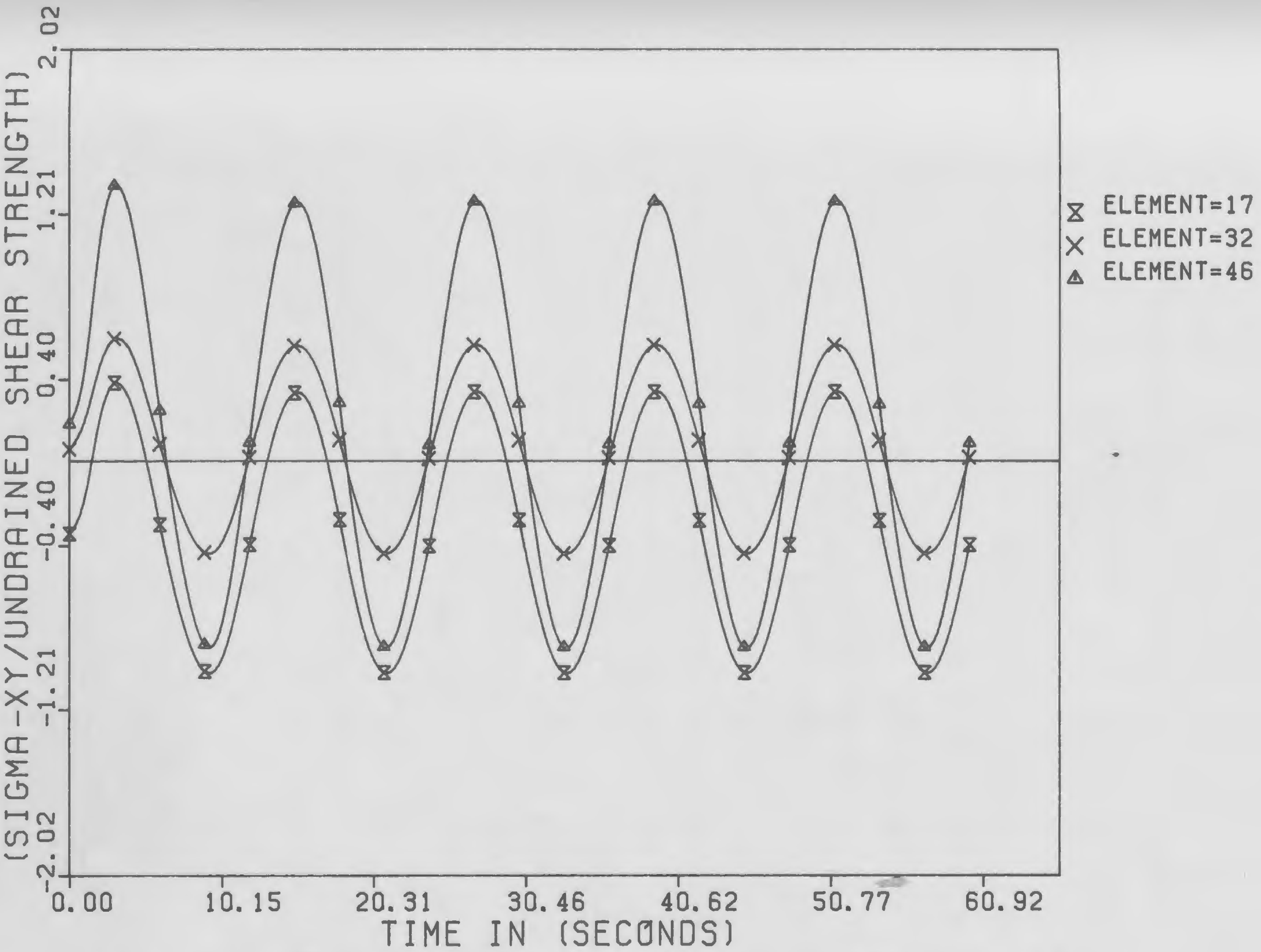


FIG. 6.22 TIME HISTORY PLOT OF SHEAR STRESS
(WITH CAISSON; DYNAMIC ELASTIC ANALYSIS)

1.15, the dynamic shakedown loading domain is obtained as $\frac{F_V}{As_u} = 2.33$ and $\pm \frac{F_H}{As_u} = 0.58$.

Figs. 6.20(a) and (b) depict the time history of elastic displacements in horizontal and vertical directions for selected nodal points. Because of the damping, steady state condition is reached within a very few cycles. The distributions of deformation δ_H and δ_V for various loading cycles are also shown in Figs. 6.21(a) and (b) respectively. Fig. 6.22 shows the shear stress time history for various elements beneath the caisson. The maximum shear stress is observed in element 46 because of the effect of the combined moment and shear loading acting at the right edge of the footing.

The horizontal and vertical loads are applied at top nodes of the caisson in a distributed manner and subsequently the dynamic response analyses are carried out with two types of nonlinear models: a) equivalent linear and b) elastic-plastic.

6.5.2 Equivalent Linear Analysis

The nonlinear effect of the soil due to the cyclic wave loading can be expressed in terms of the soil shear modulus, G, and the damping ratio, D, both dependent on cyclic shear strains. In the equivalent linear analysis, the soil properties are adjusted by an iterative procedure so as to make them compatible with the computed soil strain level. The soil properties used in the analysis are the maximum shear modulus, G_{max}, maximum damping ratio, D_{max}, and undrained shear strength, s_u. For clays, the ratio of G_{max}/s_u, which is constant (Fig. 5.3) is 2300 at low strain level (10^{-4}%) and the maximum damping D_{max} is 29% at 10% of strain (Fig. 5.4). For simplicity the secant shear modulus, G, versus the shear strain, and damping ratio are approximated by a series of straight lines on a semilogarithmic plot. These values are then stored in the programme and used to compute the strain compatible soil modulus and damping ratio during subsequent iterations.

Initial shear modulus is assigned to each element and the natural frequency of the structure-foundation system is first computed. The damping ratio, is selected corresponding to the first frequency of the system and the damping matrix of a typical soil element is formed from Eqn. (5.1). Having assembled the strain dependent damping matrix and the initial set of shear moduli for each soil element, a step-by step direct integration is carried out following Eqn. (5.41) with a time step interval of 0.1429 second. The stress and strain histories obtained from the first analysis is examined for each element and the maximum shear strain is estimated. The new values of shear moduli and damping ratios corresponding to these shear strain amplitudes are estimated from Fig. 5.4 and Table 5.1.

The natural frequency of the soil-foundation system, ω_1, is computed based on new shear moduli and the Rayleigh damping coefficients, B_1 and B_2 are then computed from Eqns. (5.2a) and (5.2b). The global damping matrix is formed again and the response analysis carried out for total number of 420 time steps in order to trace the five cycles of wave loading.

Iteration is continued until the difference between the actual modulus and damping used and the strain compatible modulus and damping values is within 10% or less, and the results obtained from the final iteration are treated as the nonlinear response. Table 6.8 and 6.9 show the converged values of the shear moduli and damping ratios for selected elements beneath the gravity foundation.

TABLE 6.8 - CONVERGENCE OF SOIL SHEAR MODULUS, G, (kPa)

Element Numbers	Iteration Numbers			
	1	2	3	4
2	14651	17680	19425	20293
15	11900	12720	13073	13279
17	7791	7380	7430	7468
27	10800	10730	10712	10702
33	13563	14611	15320	15542
46	7673	7233	7213	7311

TABLE 6.9 - CONVERGENCE OF SOIL DAMPING RATIO'S, D.

Element Numbers	Iteration Numbers			
	1	2	3	4
2	0.1337	0.1235	0.1175	0.1142
15	0.1515	.1441	.1410	0.1392
17	0.1886	0.1965	.1971	0.1915
27	0.1615	0.1621	0.1624	0.1623
33	0.1375	0.1339	0.1319	0.1307
46	0.1896	0.1936	0.1938	0.1930

Figs. 6.23(a) and (b) show the time histories of horizontal and vertical deformations for selected nodes beneath the caisson. Figs. 6.24(a) and (b) depict the distribution of δ_H and δ_V along horizontal directions for different loading cycles. Time history of shear stress is shown in Fig. 6.25 for selected elements beneath the caisson.

6.5.3 Dynamic Elastic-Plastic Analysis

In the nonlinear dynamic analysis, the incremental approach is used and the equilibrium iteration carried out within each time step of increment as described in Sections 5.2.3 and 5.3.2.

Figs. 6.26(a) and (b) show the time histories of nodal displacements in horizontal and vertical directions for selected nodal points beneath the caisson. Figs. 6.27(a) and 6.27(b) depict the distribution of δ_H and δ_V for different load cycles in horizontal directions. It is seen from these figures that the shakedown has taken place beneath the caisson after second loading cycle i.e. no further increment of displacement with respect to cyclic loading. Fig. 6.28 depicts the shear stress time history along the horizontal direction for the elements beneath the caisson structure.

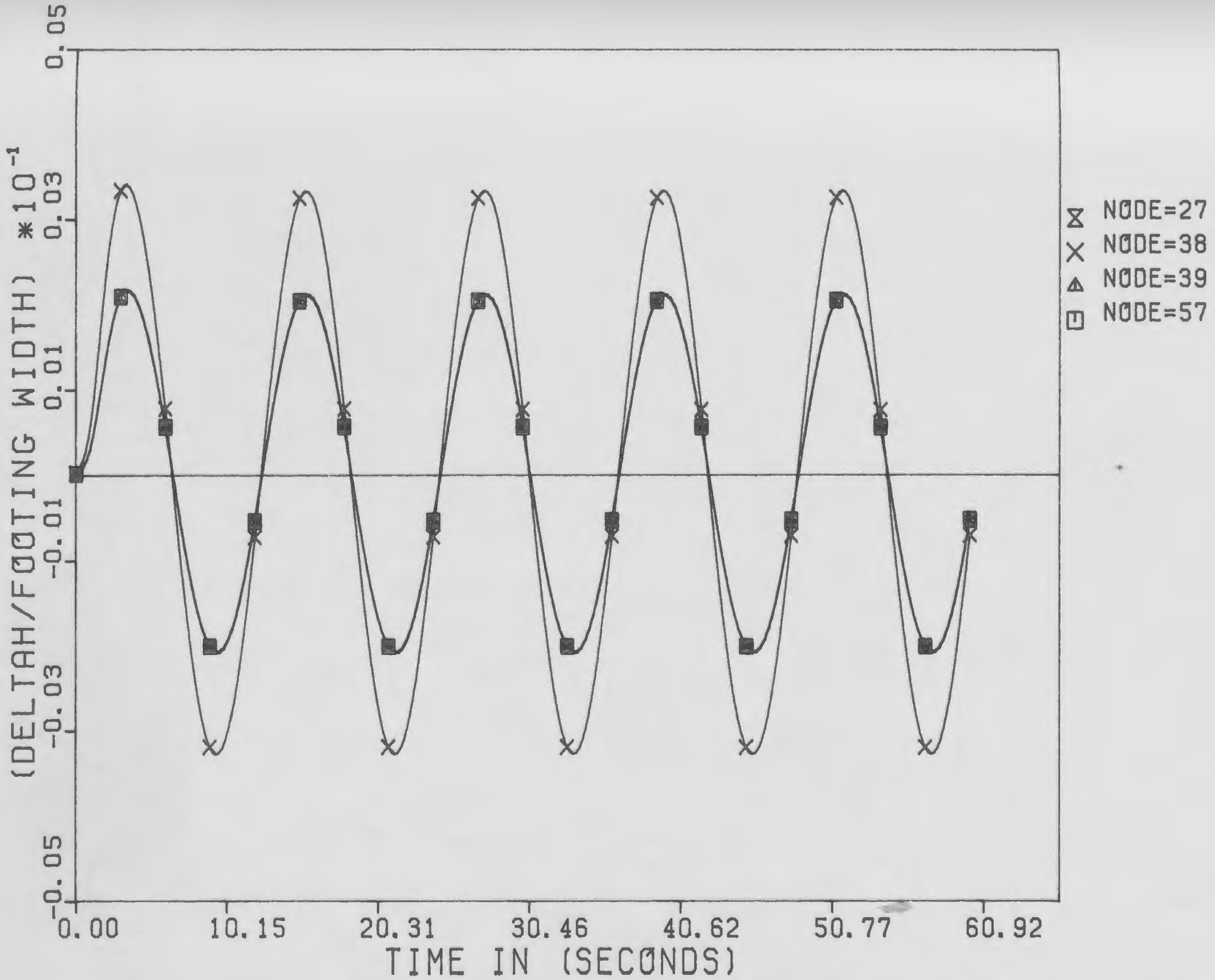


FIG. 6. 23(a) TIME HISTORY PLOT OF HOR. DISPLACEMENT
(WITH CAISSON; EQUIVALENT LINEAR ANALYSIS)

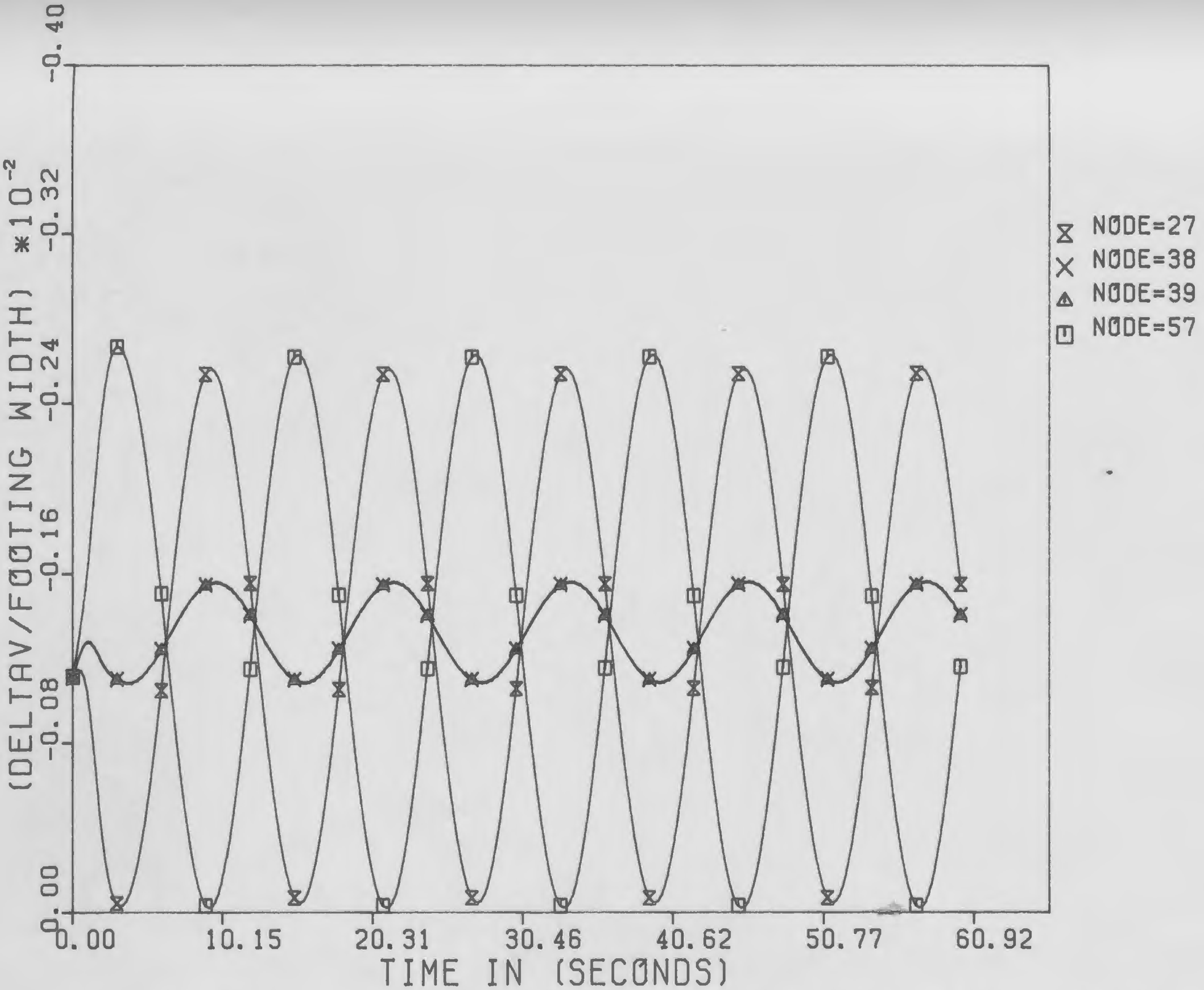


FIG. 6.23(b) TIME HISTORY PLOT OF VER. DISPLACEMENT
(WITH CAISSON; EQUIVALENT LINEAR ANALYSIS)

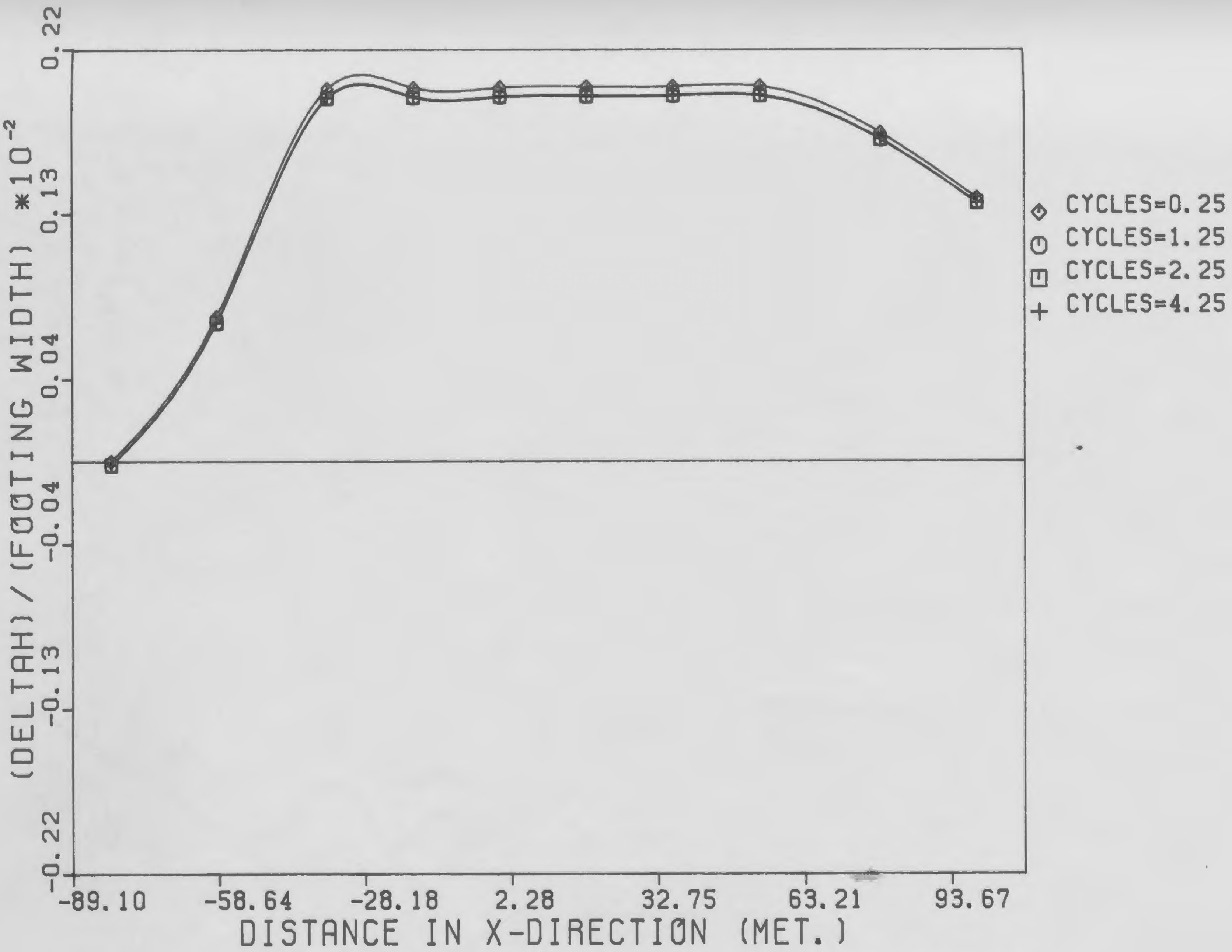


FIG. 6.24 (a) DISTRIBUTION OF HOR. DISPLACEMENT
(WITH CAISSON; EQUIVALENT LINEAR ANALYSIS)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC;)

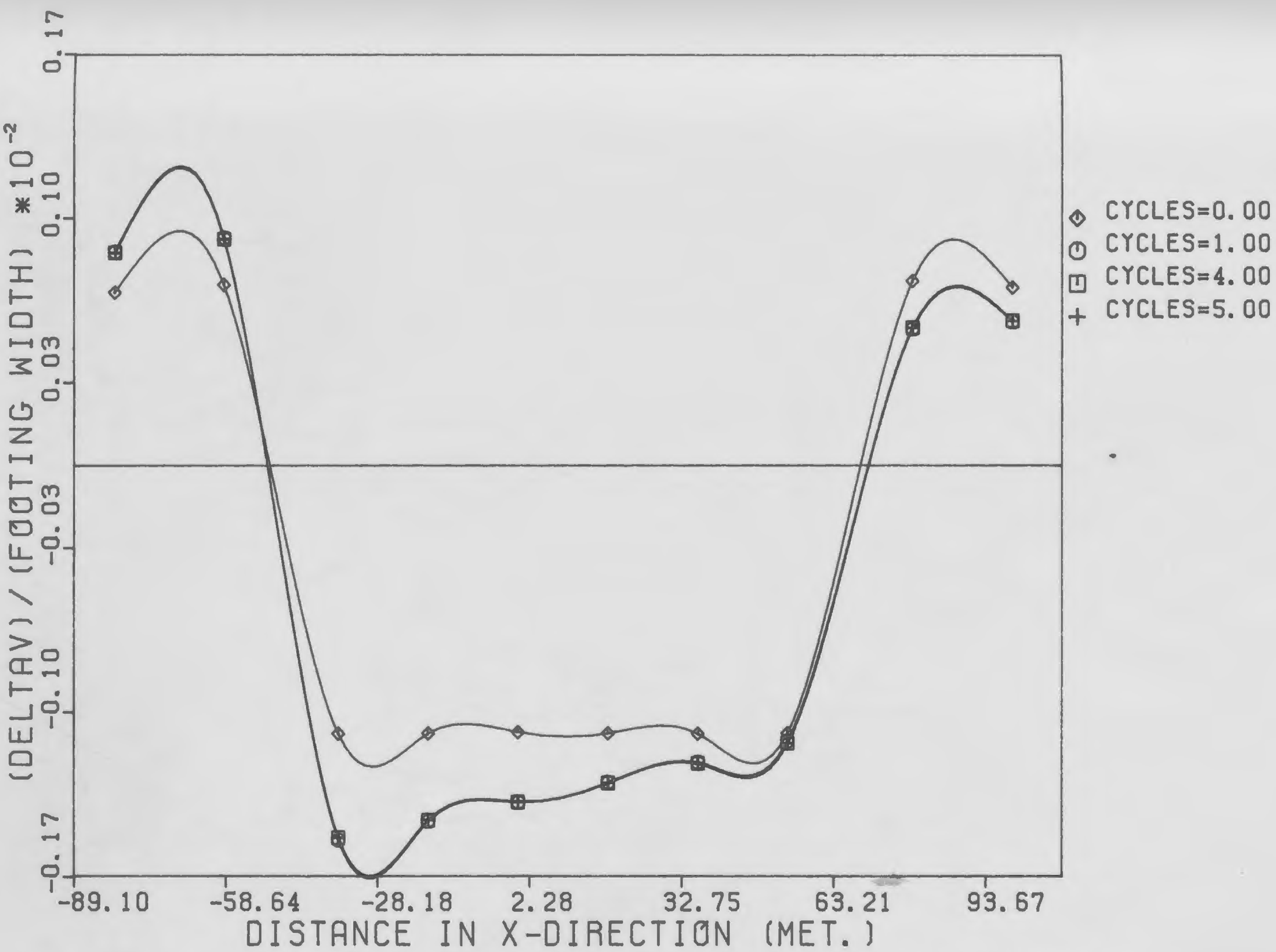


FIG. 6.24(b) DISTRIBUTION OF VER. DISPLACEMENT
(WITH CAISSON; EQUIVALENT LINEAR ANALYSIS)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC;)

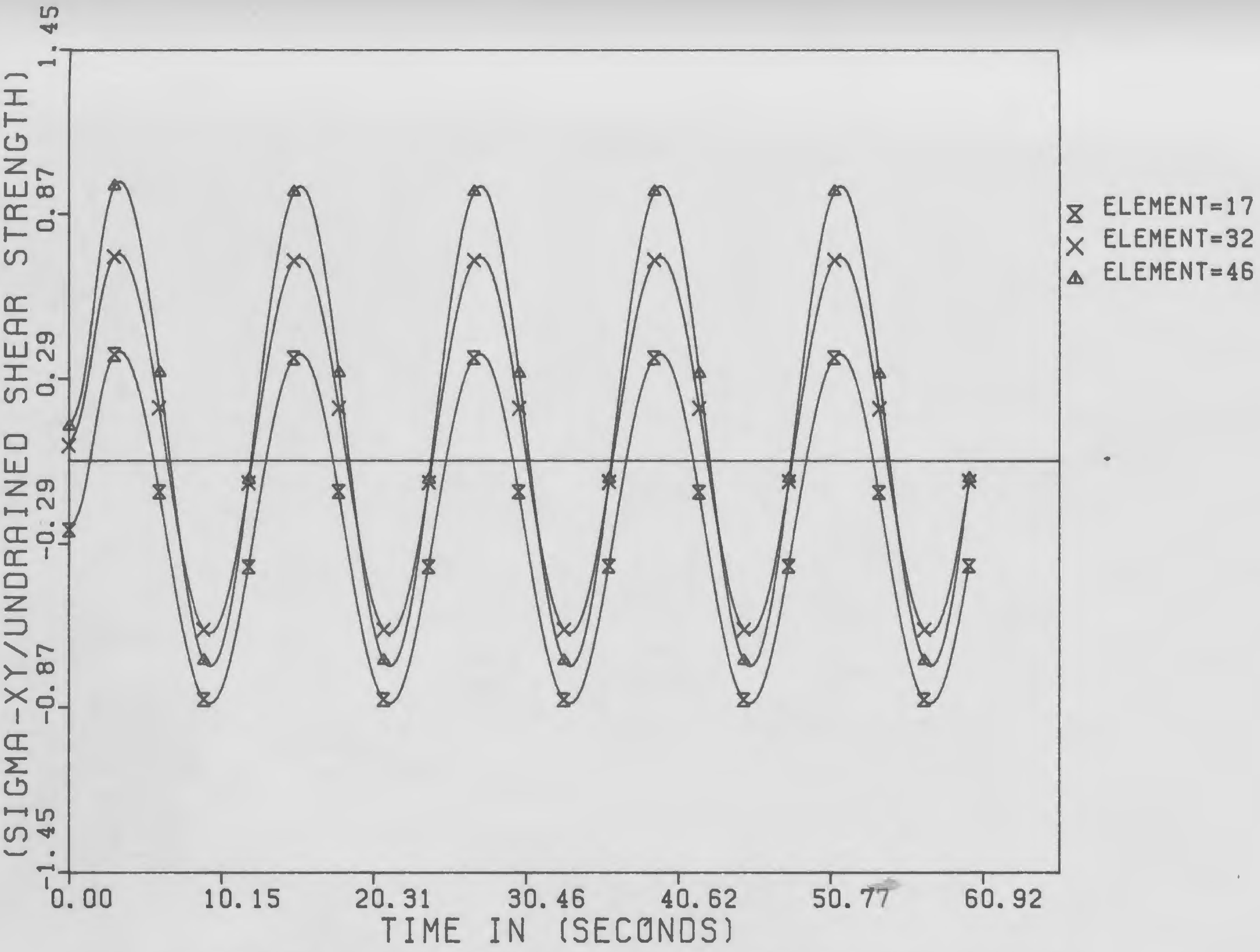


FIG. 6.25 [illegible] TIME HISTORY PLOT OF SHEAR STRESS (WITH CAISSON; EQUIVALENT LINEAR ANALYSIS)

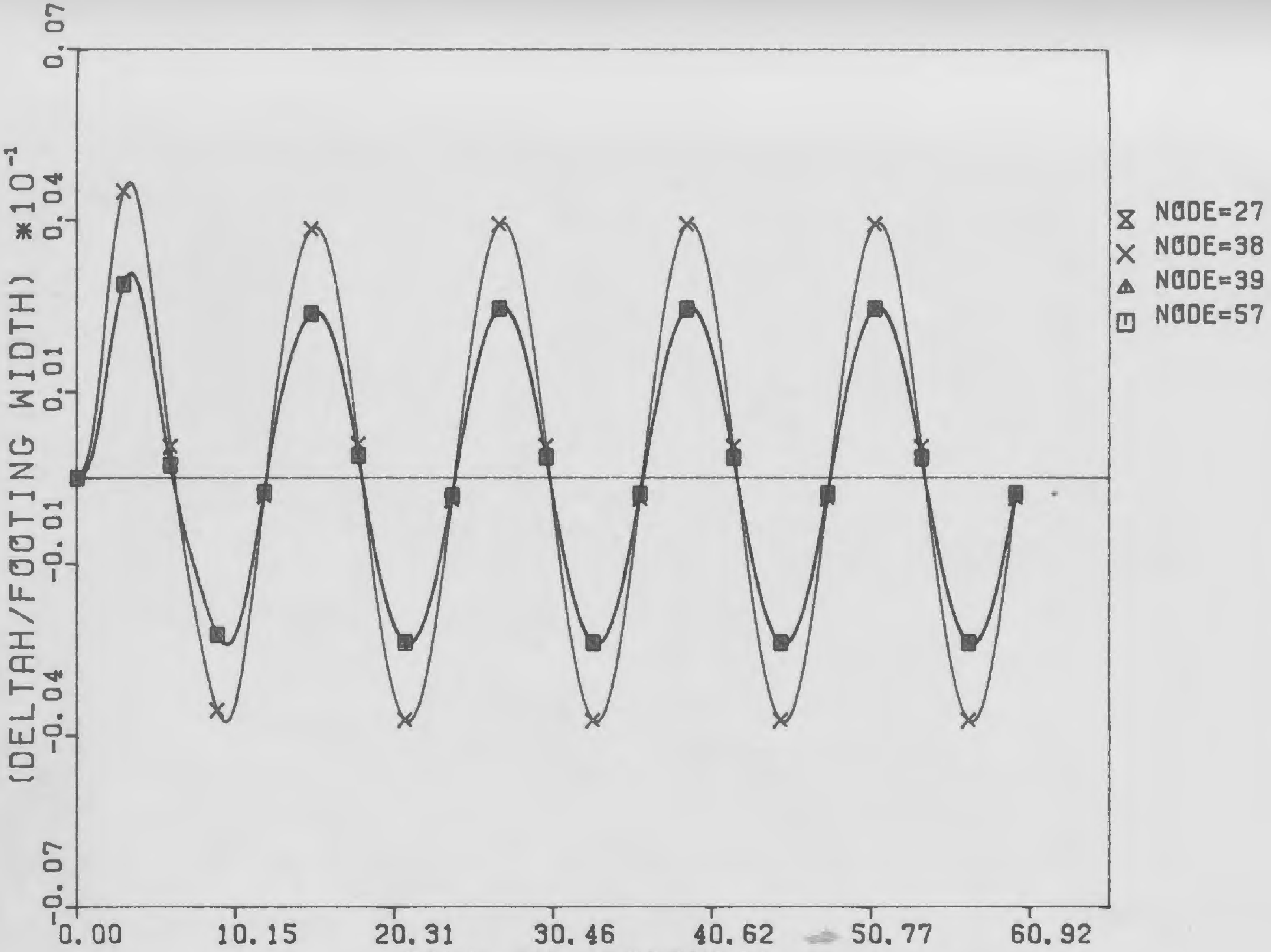


FIG. 6. 26 (a) TIME HISTORY PLOT OF HOR. DISPLACEMENT (WITH CAISSON; DYNAMIC ELASTIC-PLASTIC ANALYSIS)

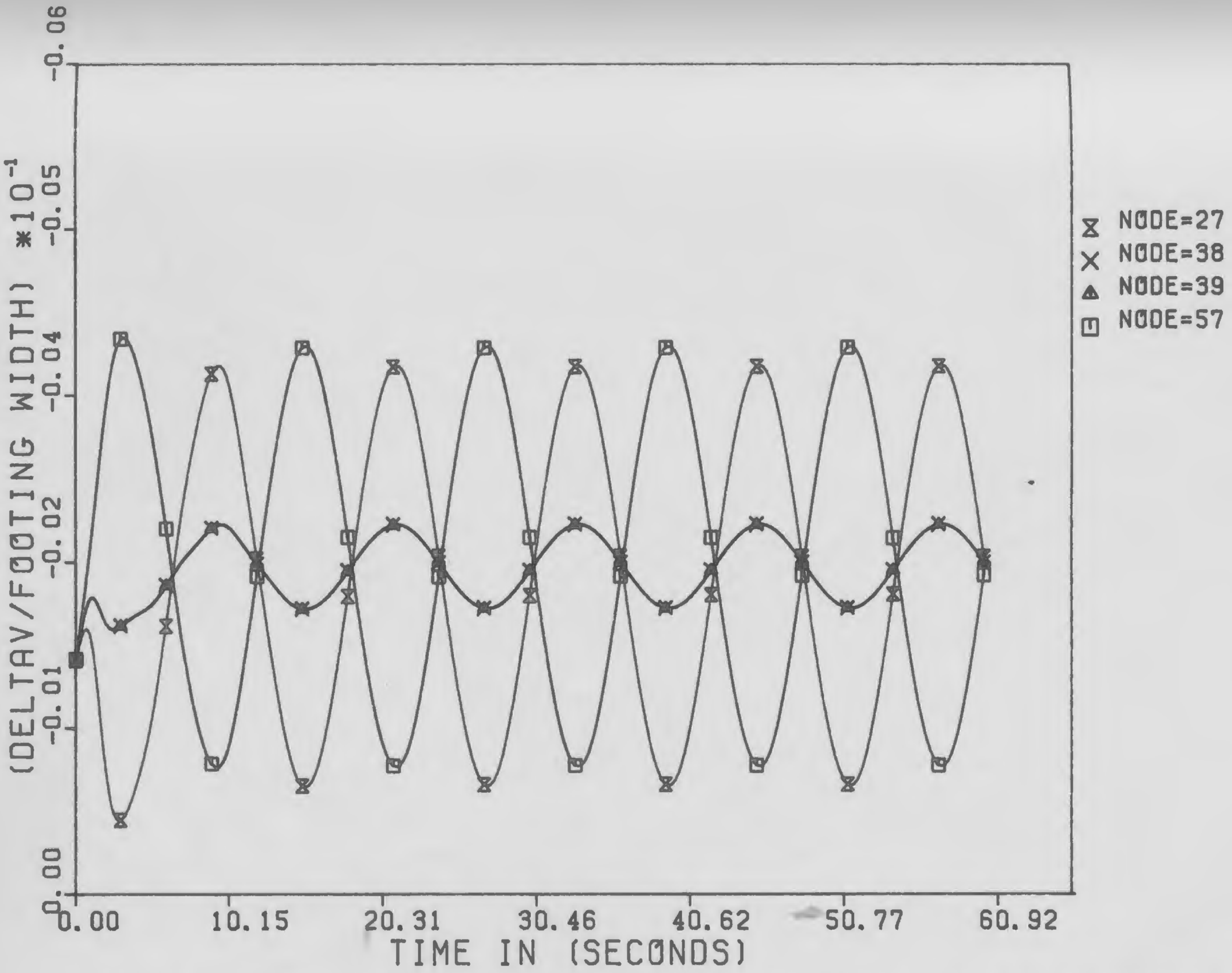


FIG. 6.26(b) TIME HISTORY PLOT OF VER. DISPLACEMENT
(WITH CAISSON; DYNAMIC ELASTIC PLASTIC ANALYSIS)

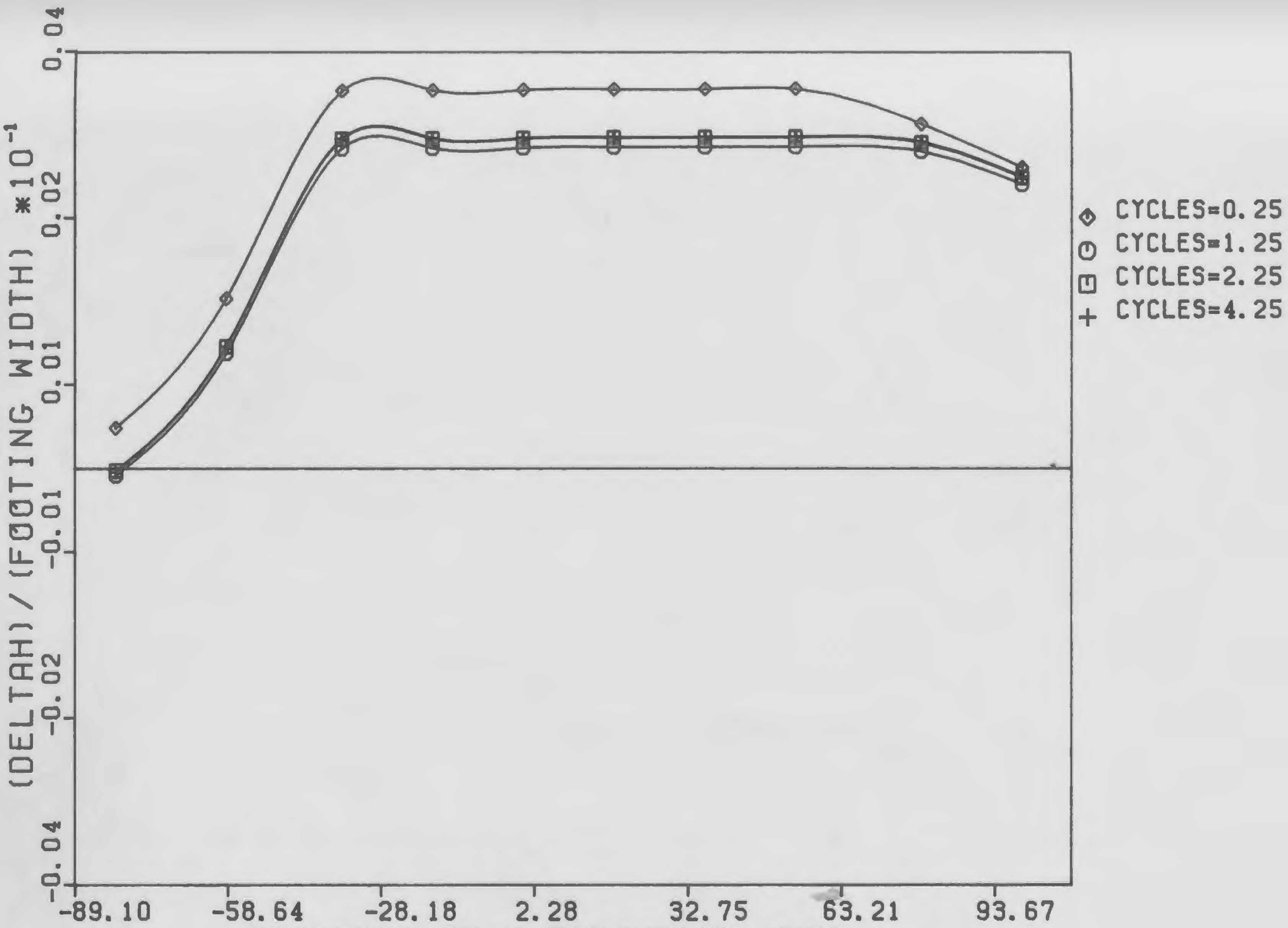


FIG. 6.27(a) DISTRIBUTION OF HOR. DISPLACEMENT
(WITH CAISSON; DYNAMIC PLASTIC ANALYSIS)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC;)

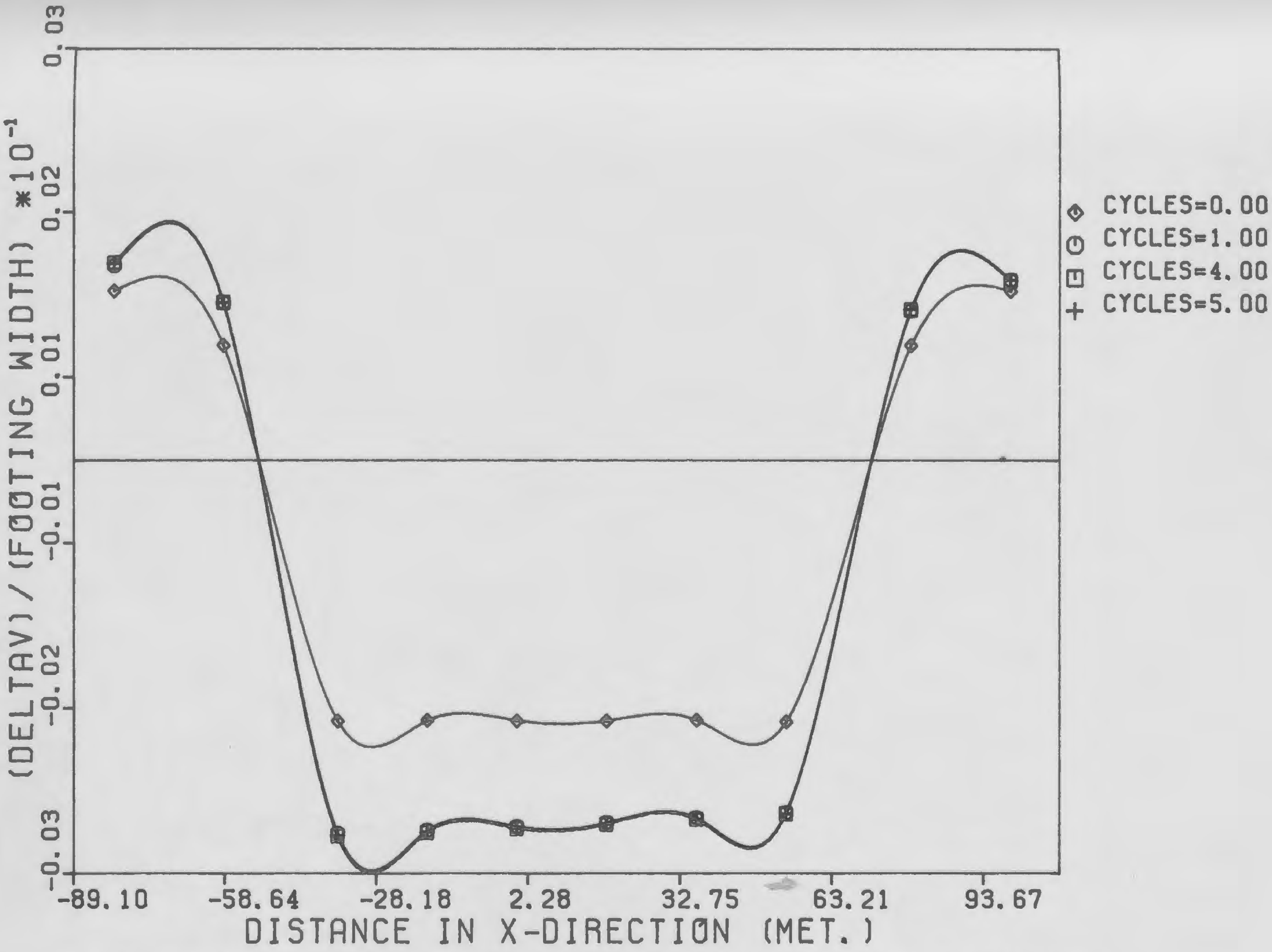


FIG. 6.27(b) DISTRIBUTION OF VER. DISPLACEMENT
(WITH CAISSON; DYNAMIC PLASTIC ANALYSIS)
(NODES 16, 21, 27, 33, 39, 45, 51, 57, 62, 67 ETC;)

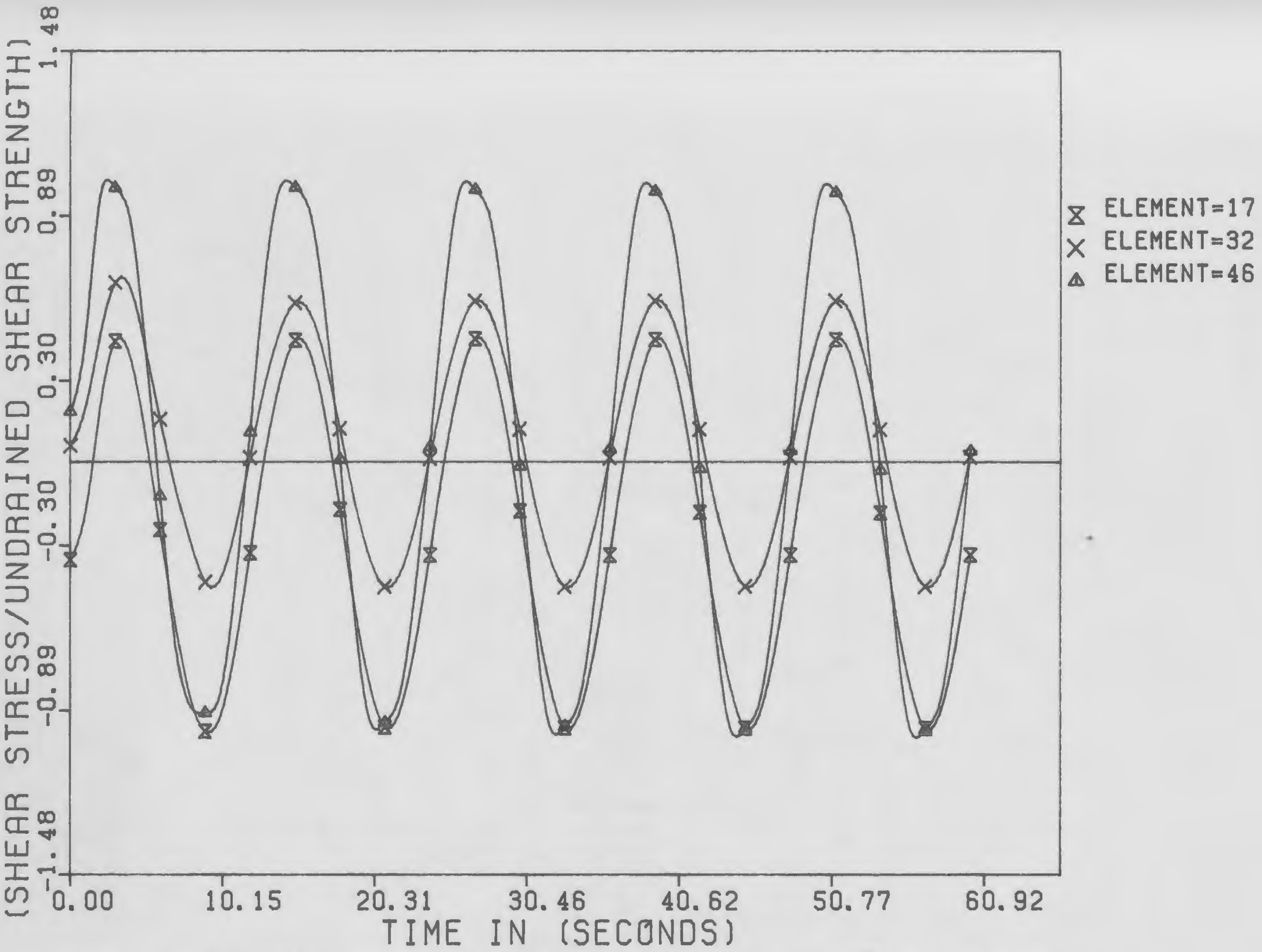


FIG. 6.28 TIME HISTORY PLOT OF SHEAR STRESS
(WITH CAISSON; DYNAMIC PLASTIC ANALYSIS)

6.5.4 Comparison Of Various Analyses

Four types of analysis have been carried out in previous sections. These analyses are: a) static non-linear b) dynamic elastic c) equivalent linear and d) dynamic elastic-plastic. Fig. 6.29 depicts the comparison of time histories of vertical displacement for node 38 for various analyses.

The dynamic elastic-plastic response is almost 8% lower than compared to its static nonlinear counterpart. However, the equivalent linear analysis suppresses the response as much as 40% compared to dynamic elastic-plastic analysis. The difference in the result can be attributed to the fact that in equivalent linear analysis the response is computed with the constant set of strain compatible soil properties for the entire loading history whereas in the true non-linear model, stiffness is updated in each time step based on the revised stress-state in each element. The limitation of the equivalent linear analysis is that it does not allow the computation of the permanent deformation.

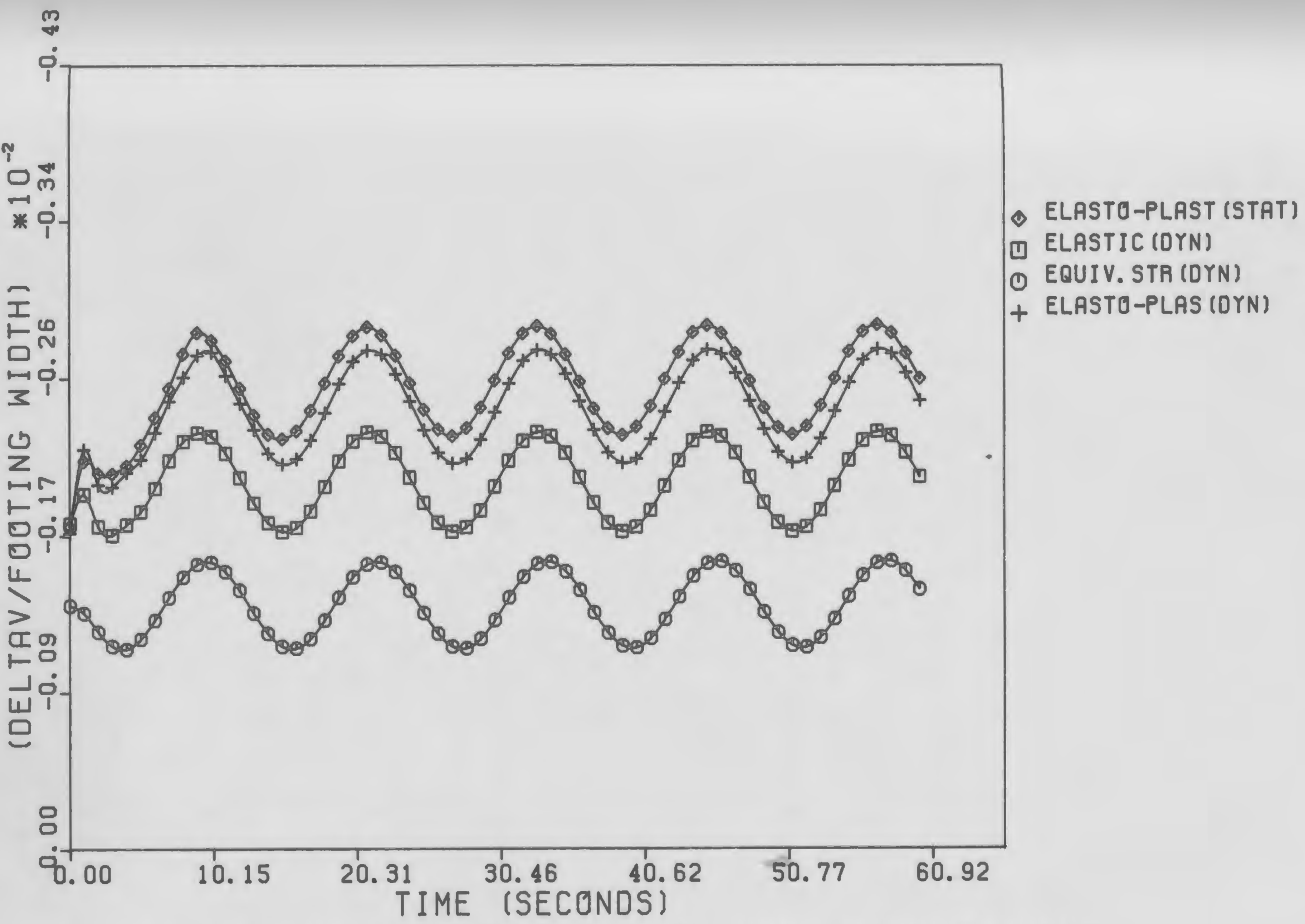


FIG. 6.29 TIME HISRTORY PLOT OF VER. DISPLACEMENT
(COMPARISON OF DIFFERENT SOIL MODELS)
(WITH CAISSON; NODE 38)

CHAPTER VII

CONCLUSIONS

7.1 Introduction

An analytical method is developed for the prediction of the permanent displacement of the foundation of a gravity type offshore platform resting on saturated soils. The wave loading is considered as dynamic and cyclic in nature and the soil is assumed to behave in an undrained manner.

The method is based on the finite element dynamic shakedown formulation of the saturated porous medium using a piecewise linearized convex yield surface. The dynamic shakedown load factor is obtained by first computing the elastodynamic response of the caisson-foundation (saturated) medium for some adjusted initial conditions (e.g. displacement and velocity) and then solving the problem as a linear programming one.

7.2 Conclusions

The displacement based finite element formulation of the saturated porous medium is used to analyze a strip footing problem for drained and undrained conditions. Analyses indicate that the displacement based finite element formulation can quite adequately predict the 'nearly incompressible' undrained deformation as well as the fully drained condition. Results show that the drained deformation is higher compared to its undrained counterpart. This is due to the fact that in undrained deformation volumetric deformaton is almost zero and therefore additional restraint is imposed.

The dual form of the kinematic shakedown problem is used to compute the shakedown load factors for two types of problems. These are a) a thick cylinder under variable internal pressure and b) a strip footing under repeated vertical loading. In the cylinder problem three types of meshes are used and the analysis indicates that for a cylinder with 45 elements, elastic and shakedown limit pressures are in excellent agreement with the theoretical values. For the strip footing problem, vertical pressure at shakedown agrees well with the ultimate bearing capacity of the foundation.

Having computed the shakedown limit load for the strip footing, an incremental analysis is carried out for five cycles of loading. The vertical pressure is assumed to be triangular in shape and varies between zero and pressure at shakedown. Shakedown is observed within 3 cycles of the loading and the permanent displacement is computed as 0.82% of the footing width. The total displacement at shakedown load is 1.58% of the footing width.

The computations of elastic and shakedown limit pressures for a flexibie foundation indicate that elastic and shakedown limit pressures depend on the inclination angle and eccentricity of the load. The elastic and shakedown limit pressures decrease as the eccentricity and the inclination angle increase. For example, the shakedown limit pressure can decrease as much as 54% (for an eccentricity of $e/B = 0.16$ and $\alpha_A = 10^o$) with respect to the case where the loading is purely vertical (i.e. $e/B = 0$ and $\alpha_A = 0.0$). Also, shakedown analyses indicate that vertical pressures under various eccentric loadings are below those predicted by the semi-empirical bearing capacity formula. Therefore, the use of the empirical formula will not produce a conservative estimate of the bearing capacity of the foundation when the loading is cyclic in nature.

The frequencies for the undrained condition are higher than those obtained for drained situation. This is because of the high stiffness associated with the undrained condition.

Dynamic shakedown load factor for the stiff footing problem is only 10% lower than its static counterpart. This is because the amplification of stresses has occurred at frequency ratio of $\omega^* = 0.34$ and therefore the shakedown load factor has further decreased.

In computing the response analyses for three different soil models, it is observed that the equivalent linear anlaysis predicts lower deformation as opposed to the other two types of analyses. However, for the quasistatic analysis of the stiff footing problem, the permanent vertical deformation is within 0.085 percent of the footing width. In all cases, it is seen that shakedown has taken place with respect to the deformations of the foundation.

These findings of this present research lead to the following conclusions.

1. The vertical undrained deformation of a strip footing is almost 44% of its drained counterpart.

2. Shakedown limit pressures for a thick cylinder and a strip footing agree quite well with the theoretical values.

3. Shakedown limit pressures for a flexible footing decrease as the inclination angle and the eccentricity of the load increase.

4. The undrained natural frequency of the soil-foundation is 12% higher than its drained counterpart.

5. Dynamic shakedown limit load for the stiff footing problem is only 10% lower than its quasistatic counterpart.

6. Permanent vertical displacement of the foundation is within 0.08% of the footing width.

7. Equivalent linear model suppresses the foundation displacement as much as 40% than those compared with dynamic elastic-plastic analysis.

7.3 Contributions

1. Development of an analytical method in determining the permanent deformation of the foundation of a gravity base offshore foundation considering soil shakedown.

2. Computations of static and dynamic shakedown load factors for various types of problems e.g. gravity foundation under cyclic inclined eccentric loading conditions.

3. Computations of the response quantities for various types of soil models e.g. equivalent linear and elastic-perfectly plastic.

4. Development of a computer code OPFA (Offshore Platform Foundation Analysis) for carrying out the above computations.

7.4 Recommendations for Future Research

The following recommendations are made to extend the present work.

- Use of nonassociated flow rule so that the work softening criteria can be incorporated in the soil constitutive relationship.

- Geometric nonlinearity to be included in the shakedown analysis.

- Use of workhardening type soil model such as proposed by Prevost (1977) and Mroz et al (1978).

- Efficient algorithm for the solution of linear programming problem related to shakedown analysis.

REFERENCES

Aboustitt, B.L. and Reddy, D.V., 1980. Finite Element Linear Programming Approach to Foundation Shakedown. Proc. Int. Symp. on Soils under Cyclic and Transient Loading (Eds. G. N. Pande and O. C. Zienkiewicz), Swansea, January, pp. 727-738.

Anderson, K.H. 1976. Behaviour of Clay Subjected to Undrained Cyclic Loading. Proc. Int. Conference on Behaviour of Offshore Structures, Trondheim, Norway.

Anderson, K.H., Hansteen, O.E., Hoeg, K., and Prevost, J.H., 1978. Soil Deformations Due to Cyclic Loads on Offshore Structures. Num Meth. in Offshore Eng., pp. 413-452. Chichester: Wiley.

Belytschko, T., 1972. Plane Stress Shakedown Analysis by Finite Elements. Int. J. Mech. Sci, Vol. 14, pp. 619-625.

Biot, M.A., 1941. General Theory of Three dimensional Consolidation. J. of App. Phys. Vol. 12, pp. 155-164.

Biot, M.A., 1955. Theory of Elasticity and Consolidation for a Porous Anisotropic Solid. J. of App. Phys. Vol. 26, pp. 182-185.

Biot, M.A., 1956a. Theory of Deformation of a Porous Viscoelastic Anisotropic Solid, J. of App. Phys. Vol. 27, pp. 457-467.

Biot, M.A., 1956b. General Solution of the Equations of Elasticity and Consolidation for a Porous Material. J. of Appl. Mech. Proc. ASME. Vol. 25, pp. 91-96.

Biot, M.A., 1956c. Theory of Propagation of Elastic Waves in a Fluid Saturated Porous Solid I: Low Frequency Range. J. of Acoust. Soc. of Am., Vol. 28, No. 2, pp. 168-178.

Biot, M.A., 1961. Mechanics of Deformation and Acoustic Propagation in Porous Media. J. of Appl. Phys. Vol. 33, No. 4, pp. 1480-1498.

Bleich, H., 1932. Uber die Bemessung Statisch Unbestimmter Stahltragwerke unter Berucksichtigung des Elastich - Plastischen Verhaltens des Baustoffes; Bauingenieur, 19/20, 261.

Bobby, W., Arockiasamy, M., Haldar, A.K., and Reddy, D.V., 1979. Finite Element Analysis of Pipe-Soil-Wave Interaction. Proc. Second Int. Conf. on Behaviour of Offshore Structures, BOSS '79, London, U.K.

Booker, J.R., Carter, J.P., and Small, J.C., 1976. An Efficient Method of Analysis for the Drained and Undrained Behavior of an Elastic Soil. Int. J. Solids and Struct., Vol. 12, pp. 589-599.

Borkowski, A., and Kleiber, M., 1980. On a Numerical Approach to Shakedown Analysis of Structures. Comp. Meth. in Appl. Mech. and Eng., Vol. 22, pp. 101-119.

Ceradini, G., 1969. Sull'adattamento dei Gorpi elasto-plasticisogetti at azioni denamiche. Giorn. Geniocivile, May, pp. 239-250.

Ceradini, G., 1980. Dynamic Shakedown in Elastic-Plastic Bodies. Proc. ASCE (EMD), Vol. 106, No. EM3, pp. 481-499.

Chen, W.F., 1975. Limit Analysis and Soil Plasticity, Elsevier Scientific Publishing Company, N.Y., U.S.A.

Corradi, L., and Maier, G., 1972. Inadaptation Theorem in the Dynamics of Elastic Work-Hardening Structures. Technical Report ISTC Politechnico, Milano.

Corradi, L., and Zavelani, A., 1974. A Linear Programming Approach to Shakedown of Structures. Comp. Meth. in Appl. Mech. and Eng., Vol. 3, pp. 37-53.

Crochet, M.J., and Naghdi, P.M., 1966. On Constitutive Equations of Flow of Fluid Through an Elastic Solid., Int. J. of Eng. Sci., Vol. 4, pp. 383-401.

Dafalias, Y. F., and Popov, E., 1976. Plastic Internal Variables Formulation of Cyclic Plasticity. J of Appl. Mech., Vol. 98, No. 4, pp. 645-650.

Dereshewicz, H., and Skalak, R., 1963. An Uniqueness in Dynamic Poro-elasticity. Bull. Seism. Soc. of Am., Vol. 53, No. 4, pp. 783-788.

Drucker, D.C. 1950. Some Implications of Work Hardening and Ideal Plasticity. Q. of Appl. Maths, pp. 411-418.

Drucker, D.C., and Prager, W., 1952. Soil Mechanics and Plastic Analysis on Limit Design, Q. Appl. Maths., Vol. 10, pp. 157-165.

Dumas, F., and Lee, K.L., 1979. Cyclic Movements of Offshore Structure on Clay. Proc. ASCE Nat. Conv., Boston, Mass., Session No. 54 and 64, pp. 1-44.

Finn, W.D.L., and Martin, G.R., 1980. Soil as an Anisotropic Kinematic Hardening Solid. Proc. ASCE NAT. Conv. and Expos. Preprint 80-525, Florida, pp. 96-113.

Firoozbakhsh, K., 1976. Mechanics of Deformation of Porous Elastic Media with Microstructure. Int. J. of Solids and Struct., Vol. 12, pp. 649-654.

Ghaboussi, J., and Wilson, E.L., 1972. Variational Formulation of Dynamics of Fluid-Saturated Porous Elastic Solids. Proc. ASCE (EMD), No. EM4, Paper No. 9152, pp. 947-963.

Green, A.E., and Naghdi, P.M., 1965. A Dynamical Theory of Interacting Continua. Int. J. of Eng. Sci., Vol. 3., pp. 231-241.

Green, A.E., and Steel, T.R., 1966. Constitutive Equations for Interacting Continua, Int. J. of Eng. Sci., Vol. 4, pp. 483-500.

Gurtin, M.E., 1964. Variational Principles in Linear Elastodynamics. Arch. Rat. Mech. Anal., Vol. 16, pp. 34-50.

Haldar, A.K., 1977. Dynamic Response of a Framed Offshore Tower to Ice Forces Considering Non-Linear Soil Structure Interaction, M. Eng., Thesis, Memorial University of Newfoundland, Canada.

Haldar, A.K., 1978. Dynamic Water-Soil-Structure Interaction with Particular Reference to Soil Shakedown. Report to School of Grad. Studies, Memorial University of Newfoundland, Canada.

Haldar, A.K., and Reddy, D.V., 1979. Dynamic Finite Element Formulation for a Fluid Saturated Porous Medium. Proc. Seventh Can. Cong. Appl. Mech. Sherbrooke, Canada, Vol. 2.

Haldar, A.K., Reddy, D.V., Arockiasamy, M., and Bobby, W., 1980. Finite Element Nonlinear Seismic Response Analysis of Submarine Pipe-Soil Interaction. Proc. Int. Symp. on Soils under Cyclic and Transient Loading (Ed. Pande and Zienkiewicz), Swansea, January, pp 867-877.

Haldar, A.K., Reddy, D.V., and Arockiasamy, M., 1981. Application of Shakedown Analysis to Foundation Problem. Proc. Conference Past, Present and Future of Geotechnical Engineering, M.I.T., Sept. 24 - 25.

Haldar, A.K., Reddy, D.V., and Arockiasamy, M., 1982a. Shakedown Analysis of Fluid-saturated Foundation Soil. Proc. Fourth Int. Conf. on Num. Meth. in Geomechanics. Edmonton, Canada, May, pp. 965-973.

Haldar, A.K., Reddy, D.V., and Arockiasamy, M., 1982b. Offshore Platform Foundation Shakedown Analysis. Proc. Third Int. Conf. on Behv. of Offshore Struct. (BOSS'82), M.I.T., August, pp. 313-333.

Hill, R., 1950. The Mathematical Theory of Plasticity. Oxford University Press, London, 355 p.

Ho, H.S., 1972. Shakedown in Elasto-Plastic Systems under Dynamic Loading. J. of Appl. Mech., Vol. 39, pp. 416-421.

Hung, N.D., and König, J.A., 1976. A Finite Element Formulation for Shakedown Problems using a Yield Criterion of the Mean. Comp. Meth. in Appl. Mech. and Eng., Vol. 8, pp. 179-192.

Hung, N.D., and Palgen, L., 1979. Shakedown Analysis by Finite Element Methods. Proc. Seventh Can. Cong. Appl. Mech., Sherbrooke, Canada, Vol. 2, pp. 101-102.

Ishihara, K., 1970. Approximate Forms of Wave Equations for Water-Saturated Porous Materials and Related Dynamic Modulus. J. Soils and Found., Vol. 10, No. 4, pp. 10-38.

Jumikis, A.R., 1969. Theoretical Soil Mechanics. Van Nostrand Reinhold Co; N.Y., 432 p.

Kleiber, M., König, J.A., and Sawczuk, A. 1982. Studies on Plastic Structures: Stability, Anisotropic Hardening, Cyclic Loads. Comp. Meth. in Appl. Mech. and Eng. Vol. 33, No. 3, pp. 487-556.

König, J.A., 1969. Shakedown Theory of Plates. Arch. Mech. Stosow. 21, pp. 623-637.

König, J.A., and Maier, G., 1981. Shakedown Analysis of Elastoplastic Structures. A Review of Recent Developments. Nuc. Eng. and Design, Vol. 66, pp. 81-95.

Koiter, W.T., 1960. General Theorems for Elastic-Plastic Solids. Progress in Solid Mechanics, Vol. 1, North Holland Publishers, pp. 167-221. (Ed. by I.N. Sneddon and R. Hill), Amsterdam.

Leckie, F.A., and Penny, R.K., 1969. Shakedown as a Guide to the Design of Pressure Vessels. J. Eng. Ind., Vol. 91, pp. 799-807.

Maier, G., 1969. Shakedown Theory in Perfect Elastoplasticity with Associated And Nonassociated Flow Laws. A Finite Element, Linear Programming Approach. Meccanica, No. 3, Vol. IV, 11 p.

Maier, G., 1970. A Matrix Structural Theory of Piecewise Linear Elastoplasticity with Interacting Yield Planes. Meccanica, Vol. 5, pp. 54-66.

Maier, G., 1972. A shakedown Matrix Theory Allowing for Workhardening and Second Order Geometric Effects. Proc. Int. Symp. Found on Plasticity. (Ed. A. Sawczuk) Warshaw, North Holland, Amsterdam, pp. 417-433.

Maier, G., 1976. Piecewise Linearization of Yield Criteria in Structural Plasticity. Achives of Solid Mechanics, Vol. 1, 239-281.

Marr, W.A., and Christian, J.T., 1981. Permanent Displacement Due to Cyclic Wave Loading. Proc. ASCE (GTD), Vol. 107, No. GT8, pp. 1129-1149.

Martin, J.B., 1975. Plasticity Fundamentals and General Results. M.I.T. Press, Cambridge, Mass.

Melan, E., 1936. Theorie Statisch Unbestimmter Systeme Aus Ideal-Plastisehen Baustoff, Sitz. Ber. AK. Wiss. Wien, IIa, pp. 145-195.

Melan, E., 1938. Zur Plastizitat des raumlichen Kontinuums. Ingenieur - Archiv, Vol. 9, pp. 116-126.

Melian Ayala, G., and Brebbia, C.A., 1972. Solution of Wave Propagation Problem in Saturated Medium. Proc. Int. Conf. of Variational Methods in Engineering, Vol. 1, Sec. 5, pp. 89-103, Southampton.

Meyerhof, G. G., 1963. Some Recent Research in the Bearing Capacity of Foundations. Canadian Geotechnical Journal, Vol 1, No. 1, pp. 16-26.

Mroz, Z., Norris, V.A., and Zienkiewicz, O.C., 1978. An Anisotropic Hardening Model for Soils and its Application to Cyclic Loading. Int. J. Num. and Analyt. Meth. in Geomechanics, Vol. 2, pp. 203-221.

Neal, B.G., 1956. The Plastic Methods of Structural Analysis, Chapman and Hall, London.

Pande, G.N., Abdullah, W.S., and Davies, E.H., 1980. Shakedown of Elasto-Plastic Continuua with Special Reference to Soil-Rock Structures. Proc. Int. Symp. on Soils under Cyclic and Transient Loading. (Eds. G.N. Pande and O.C. Zienkiewicz), Swansea, U.K., January, pp. 715-726.

Pool, J.H., 1976. Geotechnical Aspects of Gravity Structures. J. Ocean Eng., pp. 4-19.

Prager, W., 1956. Shakedown in Elastic, Plastic Media Subjected to Cycles of Load and Temperature. Proc. Symp. Sulla Plast. Nella Scienza delle Costr. Varenna, N. Zanichellied, Bologna, pp. 239-244.

Prevost, J.H., 1977. Mathematical Modelling of Monotonic and Cyclic Undrained Clay Behaviour, Int. J. Num. And Analyt. Meth. in Geomechanics, Vol. 1, No. 2, pp. 195-216.

Roscoe, K.H. and Burland, J.B., 1968. On the Generalized Stress-Strain Behaviour of Wet Clays. In Engineering Plasticity Edited by J. Hayman and F. Leckie, pp. 535-609, Cambridge University Press.

Rowe, P.W., 1975. Displacement and Failure Modes of Model Offshore Gravity Platforms Founded on Clay. Aberdeen Conference, pp. 218.1-218.17.

Sandhu, R.S., and Wilson, E.L., 1969. Finite Element Analysis of Flow in Saturated Porous Elastic Media. Proc. ASCE (EMD), Vol. 95, No. EM3, Paper No. 6615, pp. 641-652.

Seed, H.B., and Idriss, I.M., 1969. Soil Moduli and Damping Factors for Dynamic Response Analysis. EERC Report No. 70-109, Univ. of California, Berkeley, December.

Skempton, A.W., 1941. The = 0 Analysis of Stability and its Theoretical Basis. Proc. Second. Int. Conf. on Soil Mech. and Found. Eng. Vol. 1, pp. 72-78.

Smith, I.M., 1982. *Programming the Finite Element Method with Application to Geomechanics*, John Wiley and Sons, U.K.

Stubbs, S., 1976. Some Aspects of the Design and Construction of Concrete Platform Structures for the North Sea. Offshore Europe, Beaconsfield, England, pp. 38-50.

Symonds, P.S., 1951. Shakedown in Continuous Medium. J. Appl. Mech. pp. 2.

Taga, N., and Togashi, Y., 1975. Dynamic Properties of Fluid Saturated Soil Layers. Proc. IV, Japan Earth. Eng. Symp., pp. 233-240.

Taga, N., and Togashi, Y., 1977. Dynamic Properties of Fluid Saturated Stratified Soil Layer. Proc. Sixth World Conf. on Earthquake Eng., New Delhi, INDIA, Vol. 6, pp. 135-140.

Taylor, B.B., and Matyas, E.L., 1983. Settlement of a Strip Footing on a Confined Clay Layer. Can. Geotechnical Journal, Vol. 20, pp. 535-542.

Van Eekelen, H.A.M., 1980. Fatigue Models for Cyclic Degradation of Soils. Proc. Int. Symp. on Soils under Cyclic Loading. (Eds. G.N. Pande and O.C. Zienkiewicz), pp. 447-450.

Yokoo, Y., Yamagata, K., and Nagoka, H., 1971a. Finite Element Method Applied to Biot's Consolidation Theory. J. Soils and Found., Vol. 11, No. 1, pp. 29-46.

Yokoo, Y., Yamagata, K., and Nagoka, H., 1971b. Variational Principles for Consolidaton. J. Soils and Found., Vol 11, No. 4, pp. 25-35.

Yokoo, Y., Yamagata, K., and Nagoka, H., 1971c. Finite Element Analysis of Consolidation Following Undrained Deformation. J. Soils and Found., Vol. 11, No. 4, pp. 38-58.

Zienkiewicz, O.C., Lewis, R.W., Norris, V.A., and Humpheson, C., 1976. Numerical Analysis for Foundation of Offshore Structures with Special Reference to Progressive Deformation. SPE/DUT - European Spring Meeting, Amsterdam.

Zienkiewicz, O.C., Norris, V.A., Winnicki, L.A., Naylor, D.J., and Lewis, R.W., 1978. A Unified Approach to the Soil Mechanics Problems of Offshore Foundations. Proc. Int. Symp. on Num. Meth. in Offshore Engg., Swansea, U.K., pp. 361-411.

Zienkiewicz, O.C., 1977. The Finite Element Method, McGraw-Hill Book Co. Ltd.

APPENDIX - A

PROOFS OF SHAKEDOWN THEOREMS

A1. PROOF OF MELAN'S THEOREM

For an elastic-perfectly plastic body experiencing simultaneously the action of a system of loads varying slowly in time within prescribed limits, $\sigma^E_{ij}(\bar{x},t)$ and $\varepsilon^E_{ij}(\bar{x},t)$ are assumed as the instantaneous values of soil stresses and strains respectively in the corresponding elastic medium. $\sigma_{ij}(\bar{x},t)$ and $\varepsilon_{ij}(\bar{x},t)$ denote the instantaneous values of the soil stresses and strains in the actual elastic-plastic state.

Therefore, $\sigma^R_{ij}(\bar{x},t)$ and $\varepsilon^R_{ij}(\bar{x},t)$ can be expressed as

$$\sigma^R_{ij}(\bar{x},t) = \sigma_{ij}(\bar{x},t) - \sigma^E_{ij}(\bar{x},t) \qquad \ldots (1)$$

and

$$\varepsilon^R_{ij}(\bar{x},t) = \varepsilon_{ij}(\bar{x},t) - \varepsilon^E_{ij}(\bar{x},t) \qquad \ldots (2)$$

where $\sigma^R_{ij}(\bar{x},t)$ and $\varepsilon^R_{ij}(\bar{x},t)$ are instantaneous residual soil stresses and strains respectively in the body. The elastic strains which are caused by the residual soil stresses are denoted by ε^{ER}_{ij}. Strains, ε^E_{ij}, ε_{ij} are kinematically admissible, which means they satisfy the compatability condition and thus the corresponding displacements satisfy prescribed kinematic boundary conditions shown in Fig. 3.1.

The total soil strains, ε_{ij}, are composed of both elastic, ε^{E}_{ij}, and plastic, ε^{p}_{ij}, parts as:

$$\varepsilon_{ij}(\bar{x},t) = \varepsilon^{E}_{ij}(\bar{x},t) + \varepsilon^{p}_{ij}(\bar{x},t) \qquad \ldots (3)$$

Dividing Eqn. (1) by the appropriate elasticity constants we obtain

$$\varepsilon^{ER}_{ij}(\bar{x},t) = C^{-1}_{ijkl}\,\sigma_{ij}(\bar{x},t) - \varepsilon^{E}_{ij}(\bar{x},t) \qquad \ldots (4)$$

where C^{-1}_{ijkl} =coefficients of elasticity matrix

Adding Eqn. (2) and Eqn. (4) and using Eqn. (3)

$$\varepsilon^{R}_{ij}(\bar{x},t) + C^{-1}_{ijkl}\,\sigma_{ij}(\bar{x},t) = \varepsilon^{ER}_{ij}(\bar{x},t) + \varepsilon^{p}_{ij}(\bar{x},t) + \varepsilon^{E}_{ij}(\bar{x},t) \qquad \ldots (5)$$

Therefore

$$\varepsilon^{R}_{ij}(\bar{x},t) = \varepsilon^{ER}_{ij}(\bar{x},t) + \varepsilon^{p}_{ij}(\bar{x},t) \qquad \ldots (6)$$

and

and

$$\varepsilon_{ij}(\bar{x},t) = \varepsilon^R_{ij}(\bar{x},t) + \varepsilon^E_{ij}(\bar{x},t)$$

$$= \varepsilon^{ER}_{ij}(\bar{x},t) + \varepsilon^p_{ij}(\bar{x},t) + \varepsilon^E_{ij}(\bar{x},t) \quad \ldots (7)$$

The admissible soil plastic strain rate cycle, $\dot{\varepsilon}^p_{ij}$, is defined by its property that the increments of the plastic strains, $\Delta\varepsilon^R_{ij}$, in such a cycle for some time interval, τ, given by

$$\Delta\varepsilon^p_{ij} = \int_0^\tau \dot{\varepsilon}^p_{ij}(t)\,dt \quad \ldots (8)$$

form a kinematically admissible strain distribution.

The strains given by Eqn. (8) may be obtained from Eqn. (3.7a) by prescribing a displacement field, Δu_i, which vanishes on S_1, (Fig. 3.1). The soil plastic strain rate, $\dot{\varepsilon}^p_{ij}$, is related to residual stress rate distribution, $\dot{\sigma}^R_{ij}(\bar{x},t)$. By definition, $\varepsilon^{ER}_{ij}(\bar{x},t)$, corresponds to instantaneous elastic strain distribution caused by the soil residual stress state, $\sigma^R_{ij}(\bar{x},t)$, and defining the, $\dot{u}_i(t)$, as the velocity field the kinematically admissible strain field is obtained from Eqn. (3.5) as

$$\varepsilon'_{ij} = \varepsilon^p_{ij} + \varepsilon^{ER}_{ij} \quad \ldots (9)$$

The displacement increment corresponding to the admissible plastic strain rate cycle is

$$\Delta u_i = \int_0^{\tau} \dot{u}_i \, dt \qquad \ldots (10)$$

The residual soil stress at time $t = \tau$ will be equal to the value of the stress at time $t = 0$ as the plastic strain is kinematically admissible and hence

$$\int_0^{\tau} \dot{\varepsilon}_{ij}^{ER} \, dt = 0 \qquad \ldots (11)$$

According to Melan's theorem if shakedown takes place, a time-independent residual stress field, $\bar{\sigma}_{ij}^{R}(\bar{x})$, will exist. This residual stress field, $\bar{\sigma}_{ij}^{R}(\bar{x})$, can be obtained from the solution of the equilibrium Eqn. (3.1) by satisfying the zero boundary conditions on S_2, (Fig. 3.1) where the traction is prescribed. Defining, $\bar{\varepsilon}_{ij}^{ER}$, as the elastic strain component corresponding to the fictitious residual stresses, $\bar{\sigma}_{ij}^{ER}(\bar{x})$, and admitting that, $\bar{\varepsilon}_{ij}^{ER}$, are not kinematically possible strains, the following two stress fields are defined:

$$(1) \quad \sigma_{ij}^{s} = \sigma_{ij}^{E} + \bar{\sigma}_{ij}^{R} \qquad \ldots (12)$$

where σ_{ij}^{s} = safe soil stress state

and therefore $f_y(\sigma_{ij}^{s}) < 0$ from Eqn. (4.1a)

and

$$(2) \quad \sigma^a_{ij} = \sigma^E_{ij} + \sigma^R_{ij} \qquad \ldots (13)$$

where σ^a_{ij} = admissible soil stress which touches the yield surface and therefore satisfies the yield condition from Eqn. (4.1b) as

$$f_y(\sigma^a_{ij}) \leq o$$

A fictitious positive elastic strain energy, S^e, is defined in terms of the stress difference, $(\sigma^R_{ij} - \bar{\sigma}^R_{ij})$, where, $\bar{\sigma}^R_{ij}$, is a time independent residual stress field satisfying the equilibrium with zero external forces

Therefore

$$S^e = 1/2 \int_V (\sigma^R_{ij} - \sigma^R_{ij}) (\varepsilon^{ER}_{ij} - \bar{\varepsilon}^{ER}_{ij})\, dV \qquad \ldots (14)$$

The stress differences, $(\sigma^R_{ij} - \bar{\sigma}^R_{ij})$, are related to strain differences, $(\varepsilon^{ER}_{ij} - \bar{\varepsilon}^{ER}_{ij})$, by Hooke's law.

Taking the derivative of the strain energy, S^e, with respect to time, we get

$$\dot{S}^e = \frac{d}{dt}(S^e) = \int_V (\sigma^R_{ij} - \bar{\sigma}^R_{ij}) \frac{d}{dt} (\varepsilon^{ER}_{ij} - \bar{\varepsilon}^{ER}_{ij})\, dV \qquad \ldots (15)$$

As the stress, $\bar{\sigma}^R_{ij}$, and the strain, $\bar{\varepsilon}^{ER}_{ij}$, are independent of time, by definition

$$\dot{S}^e = \int_V (\sigma^R_{ij} - \bar{\sigma}^R_{ij})\, \dot{\varepsilon}^{ER}_{ij}\, dV \qquad \ldots (16)$$

From Eqn. (7)

$$\dot{\varepsilon}^{ER}_{ij} = \dot{\varepsilon}_{ij} - \dot{\varepsilon}^E_{ij} - \dot{\varepsilon}^p_{ij}$$

Therefore

$$\dot{S}^e = \int_V (\sigma^R_{ij} - \bar{\sigma}^R_{ij})\, (\dot{\varepsilon}_{ij} - \dot{\varepsilon}^E_{ij} - \dot{\varepsilon}^p_{ij})\, dV \qquad \ldots (17)$$

The distribution of the stress differences, $(\sigma^R_{ij} - \bar{\sigma}^R_{ij})$, is self-equilibrating ie. it satisfies the equilibrium conditions with zero external force and the strain rate difference, $(\dot{\varepsilon}_{ij} - \dot{\varepsilon}^E_{ij})$, is kinematically admissible. Therefore by principle of virtual work, it can be stated that the rate of work of the internal force(s) is equal to the rate of work of the external force(s). As the external forces are zero in this case, while, $u_i - u^E_i = 0$ on part of the surface, S_1, (Fig. 3.1) where displacements are prescribed and using the following relationship

$$\int_V (\sigma^R_{ij} - \bar{\sigma}^R_{ij})\, (\dot{\varepsilon}_{ij} - \dot{\varepsilon}^E_{ij})\, dV = 0 \qquad \ldots (18)$$

Eqn. (17) is further reduced to

$$\dot{S}^e = \int_V (\sigma_{ij}^R - \bar{\sigma}_{ij}^R)\, \dot{\varepsilon}_{ij}^p \, dV \qquad \ldots (19)$$

From Eqns. (12) and (13), Eqn. (19) can be written in the following form

$$\dot{S}^e = - \int_V (\sigma_{ij} - \sigma_{ij}^s)\, \dot{\varepsilon}_{ij}^p \, dV \qquad \ldots (20)$$

The basic idea and proposition of Melan's theroem rest on the assumption that the material (in this case soil medium) obeys Drucker's postulate which is given by Eqn. (4.2a) as

$$(\sigma_{ij} - \sigma_{ij}^s)\, \dot{\varepsilon}_{ij}^p > 0$$

Therefore, $\frac{d}{dt}(S^e) < 0$, as long as, $\dot{\varepsilon}_{ij}^p = 0$; as the elastic strain energy is always positive and can never be negative, and therefore a time denoted by the passage of N cycles, will be reached when plastic flow will cease ie. $\dot{\varepsilon}_{ij}^p = 0$ implying

$$\frac{d}{dt}(S^e) = 0 \qquad \ldots (21)$$

This means the residual stress field will no longer change with respect to time and the soil will subsequently experience only <u>elastic deformaton</u> as the loads are varied cyclically.

A2. PROOF OF DYNAMIC SHAKEDOWN THEOREM

In order to prove the dynamic shakedown theorem as an extension of Melan's quasistatic shakedown theorem, an additional term is added to the Eqn. (14) which is referred to here as the kinetic energy of the system, K^e. The geometry change is assumed to be negligible.

$$TE = \text{Total energy}$$

$$= S^e + K^e$$

$$= \int_V 1/2\,(\sigma_{ij}^{R} - \bar{\sigma}_{ij}^{R})\,(\varepsilon_{ij}^{ER} - \bar{\varepsilon}_{ij}^{ER})\,dV$$

$$+ \int_V 1/2\,m\,(\dot{u}_i - \dot{u}_i^{E})\,(\dot{u}_i - \dot{u}_i^{E})\,dV \qquad \ldots (22)$$

The virtual work equation for the dynamic loading situation can be written in the following rate form as

$$\int_S \hat{T}_i\,\dot{u}_i\,dS + \int_V B_i\,\dot{u}_i\,dV - \int_V m\,\ddot{u}_i\,\dot{u}_i\,dV = \int_V \sigma_{ij}\,\dot{\varepsilon}_{ij}\,dV \qquad \ldots (23)$$

By virtue of Eqn. (23), the drrivative of Eqn. (22) with respect to time is expressed as

$$\begin{aligned}\frac{d}{dt}(T^e) = &\int_V (\sigma^R - \bar{\sigma}^R)\,\frac{d}{dt}(\varepsilon^{ER} - \bar{\varepsilon}^{ER})\,dV \\ &+ \Big[\int_{S_2} (\hat{T}_i - \hat{T}_i^E)(\dot{u}_i - \dot{u}_i^E)\,dS + \int_V (B_i - B_i^E) \\ &(\dot{u}_i - \dot{u}_i^E)\,dV - \int_V (\sigma - \sigma^E)(\dot{\varepsilon} - \dot{\varepsilon}^E)\,dV\Big] \qquad \ldots (24)\end{aligned}$$

As, $\bar{\varepsilon}^{ER}$, is independent of time and the elastic solution, $\dot{u}_i^E$, must satisy the same loading and constraints as given in Eqn. (3.7), both terms inside the bracket are zero. Using Eqn. (7),

$$\begin{aligned}\dot{T}^e &= \int_V (\sigma^R - \bar{\sigma}^R)\,\dot{\varepsilon}^{ER}\,dV - \int_V (\sigma - \sigma^E)(\dot{\varepsilon} - \dot{\varepsilon}^E)\,dV \\ &= \int_V (\sigma^R - \bar{\sigma}^R)(\dot{\varepsilon} - \dot{\varepsilon}^p - \dot{\varepsilon}^E)\,dV \\ &\quad - \int_V (\sigma - \sigma^E)(\dot{\varepsilon} - \dot{\varepsilon}^E)\,dV \qquad \ldots (25)\end{aligned}$$

Substituting $\bar{\sigma}^R = \sigma - \sigma^E$, in Eqn. (25) gives

$$\dot{T}^e = - \int_V (\sigma^R - \bar{\sigma}^R)\,\dot{\varepsilon}^p\,dV - \int_V \bar{\sigma}^R (\dot{\varepsilon} - \dot{\varepsilon}^E)\,dV \qquad \ldots (26)$$

The second term is again zero by the virtual work equation due to the fact that, $\bar{\sigma}^R$, is a time independent residual stress distribution and is always in equilibrium with zero external force.

Therefore

$$\dot{T}^e = - \int_V (\sigma^R - \bar{\sigma}^R) \dot{\varepsilon}^p dV \qquad \ldots (27)$$

From Eqns. (12) and (13), Eqn. (27) reduces to

$$\dot{T}^e = - \int_V (\sigma - \sigma^S) \dot{\varepsilon}^p dV \qquad \ldots (28)$$

$(\sigma - \sigma^S) \dot{\varepsilon}^p > 0$, because of Drucker's stability postulate and therefore, $\dot{T}^e < 0$, as long as, $\dot{\varepsilon}^p > 0$; the total energy is always positive and can never be negative, and therefore a time denoted by the passage of N cycles, will be reached when plastic flow will cease i.e. implying

$$\dot{T}^e = 0 \qquad \ldots (29)$$

This means the residual stress field will no longer change with respect to time under dynamic situation and the soil will subsequently experience only <u>elastic deformation</u> as the loads are varied cyclically.

APPENDIX - B

DERIVATION OF UNDRAINED ELASTIC PLASTIC CONSTITUTIVE MATRIX, EQN. (5.35)

B1. DERIVATION OF EQN. (5.35)

Using Eqn. (3.38h), Eqn. (5.26) and Eqn. (5.31), the term in the denominator in Eqn. (5.35) is expressed for elastic-perfectly plastic material (H = 0) as

$$[L] = \left[a\delta_{ij} + \frac{S_{ij}}{2\sqrt{J_2}}\right]^T \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix} \left[a\delta_{ij} + \frac{S_{ij}}{2\sqrt{J_2}}\right] \qquad \ldots (1)$$

$$= \left[a + \frac{S_x}{2\sqrt{J_2}} \quad a + \frac{S_y}{2\sqrt{J_2}} \quad a + \frac{S_z}{2\sqrt{J_2}} \quad \frac{\tau_{xy}}{\sqrt{J_2}}\right]^T \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} a + \dfrac{S_x}{2\sqrt{J_2}} \\ a + \dfrac{S_y}{2\sqrt{J_2}} \\ a + \dfrac{S_z}{2\sqrt{J_2}} \\ \dfrac{\tau_{xy}}{\sqrt{J_2}} \end{bmatrix}$$

$$= [(\lambda+2\mu) \{(a+\frac{S_x}{2\sqrt{J_2}})^2 + (a+\frac{S_y}{2\sqrt{J_2}})^2 + (a+\frac{S_z}{2\sqrt{J_2}})^2 \}$$

$$+ 2\lambda\{(a+\frac{S_x}{2\sqrt{J_2}})(a+\frac{S_y}{2\sqrt{J_2}}) + (a+\frac{S_x}{2\sqrt{J_2}})(a+\frac{S_z}{2\sqrt{J_2}})$$

$$+ (a+\frac{S_y}{2\sqrt{J_2}})(a+\frac{S_z}{2\sqrt{J_2}})\} + \mu(\frac{\tau_{xy}}{\sqrt{J_2}})^2]$$

$$= [(\lambda+2\mu) \{3a^2+\frac{2a}{2\sqrt{J_2}}(S_x+S_y+S_z) + \frac{S_x^2+S_y^2+S_z^2}{4J_2}\}$$

$$+ 2\lambda \{3a^2+\frac{1}{4J_2}(S_xS_y+S_yS_z+S_zS_x)\} + \mu \frac{\tau_{xy}^2}{J_2}]$$

Since $S_x+S_y+S_z = 0$, one may write

$$L = [(\lambda+2\mu)(3a^2) + \frac{\lambda}{4J_2}\{S_x^2+S_y^2+S_z^2 + 2(S_xS_y+S_yS_z+S_zS_x)\}$$

$$+ 3a^2(2\lambda) + \mu\frac{\tau_{xy}^2}{J_2} + \frac{\mu}{2J_2}(S_x^2+S_y^2+S_z^2)]$$

$$= [3a^2(3\lambda+2\mu) + \mu] \qquad \dots (2)$$

The each term of the numerator in Eqn. (5.35) is derived as follows

Assume
$$r_1 = \frac{\partial f_y}{\partial \sigma_x} = a + \frac{S_x}{2\sqrt{J_2}}$$

$$r_2 = \frac{\partial f_y}{\partial \sigma_y} = a + \frac{S_x}{2\sqrt{J_2}} \qquad \dots (3)$$

$$r_3 = \frac{\partial f_y}{\partial \sigma_z} = a + \frac{S_z}{2\sqrt{J_2}}$$

and
$$r_4 = \frac{\partial f_y}{\partial \tau_{xy}} = \frac{\tau_{xy}}{\sqrt{J_2}}$$

Using the above relationships, Eqn. (5.35) is expressed as

$$[D_{sd}^{Ep}] = [D_{sd}] - \frac{\left[D_{sd}\right]\left\{\frac{\partial f_y}{\partial \sigma}\right\}\left\{\frac{\partial f_y}{\partial \sigma}\right\}^T\left[D_{sd}\right]}{\left[L\right]} \qquad \ldots (4)$$

$$= [D_{sd}]-[L]^{-1}\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}\begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix}\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}$$

$$= [D_{sd}]-[L]^{-1}\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}\begin{bmatrix} r_1^2 & r_1r_2 & r_1r_3 & r_1r_4 \\ r_2r_1 & r_2^2 & r_2r_3 & r_2r_4 \\ r_3r_1 & r_3r_2 & r_3^2 & r_3r_4 \\ r_4r_1 & r_4r_2 & r_4r_3 & r_4^2 \end{bmatrix}\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}$$

$$\ldots (5)$$

Therefore,

$$[D^{Ep}_{sd}] = [D_{sd}]-[L]^{-1}\begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix}\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}$$

$$= [D_{sd}]-[L]^{-1}\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{42} & Y_{44} \end{bmatrix} \qquad \ldots (6)$$

where $Y_{11} = (\lambda+2\mu)X_{11} + \lambda X_{12} + \lambda X_{13}$

$Y_{12} = \lambda X_{11} + (\lambda+2\mu)X_{12} + \lambda X_{13}$

$Y_{13} = \lambda X_{11} + \lambda X_{12} + (\lambda+2\mu)X_{13}$

$Y_{14} = \mu X_{14}$

$Y_{22} = \lambda X_{21} + (\lambda+2\mu)X_{22} + \lambda X_{23}$

$$Y_{23} = \lambda X_{21} + \lambda X_{22} + (\lambda+2\mu)X_{23}$$

$$Y_{24} = \mu X_{24}$$

$$Y_{33} = \lambda X_{31} + \lambda X_{32} + (\lambda+2\mu)X_{33};\ Y_{34} = \mu X_{34}$$

$$Y_{44} = \mu X_{44} \qquad \ldots (7)$$

and

$$X_{11} = [(\lambda+2\mu)r_1^2 + \lambda r_1 r_2 + \lambda r_1 r_3]$$

$$X_{12} = [(\lambda+2\mu)r_1 r_2 + \lambda r_2^2 + \lambda r_2 r_3]$$

$$X_{13} = [(\lambda+2\mu)r_1 r_3 + \lambda r_2 r_3 + \lambda r_3^2]$$

$$X_{14} = [(\lambda+2\mu)r_1 r_4 + \lambda r_2 r_4 + \lambda r_3 r_4]$$

$$X_{21} = [\lambda r_1^2 + (\lambda+2\mu)r_1 r_2 + \lambda r_1 r_3]$$

$$X_{22} = [\lambda r_1 r_2 + (\lambda+2\mu)r_2^2 + \lambda r_2 r_3]$$

$$X_{23} = [\lambda r_1 r_3 + (\lambda+2\mu)r_2 r_3 + \lambda r_3^2]$$

$$X_{24} = [\lambda r_1 r_4 + (\lambda+2\mu)r_2 r_4 + \lambda r_3 r_4]$$

$$X_{31} = [\lambda r_1^2 + \lambda r_2 r_1 + (\lambda+2\mu)r_3 r_1]$$

$$X_{32} = [\lambda r_1 r_2 + \lambda r_2^2 + (\lambda+2\mu)r_3 r_2]$$

$$X_{33} = [\lambda r_1 r_3 + \lambda r_2 r_3 + (\lambda+2\mu) r_3{}^2]$$

$$X_{34} = [\lambda r_1 r_4 + \lambda r_2 r_4 + (\lambda+2\mu) r_3 r_4]$$

$$X_{41} = \mu r_4 r_1$$

$$X_{42} = \mu r_4 r_2$$

$$X_{43} = \mu r_4 r_3$$

$$X_{44} = \mu r_4{}^2 \qquad \ldots (8)$$

Typical term in Eqn. (6) is expressed as follows:

$$Y_{11} = (\lambda+2\mu)\,[(\lambda+2\mu) r_1{}^2 + \lambda r_1 r_2 + \lambda r_1 r_3] + \lambda[(\lambda+2\mu) r_1 r_2$$

$$+ \lambda r_2{}^2 + \lambda r_2 r_3] + \lambda[(\lambda+2\mu) r_1 r_3 + \lambda r_2 r_3 + \lambda r_3{}^2]$$

$$= (\lambda+2\mu)^2 r_1{}^2 + 2\lambda(\lambda+2\mu)\,(r_1 r_2 + r_3 r_1) + \lambda^2 r_2{}^2 + \lambda^2 r_2 r_3$$

$$+ \lambda^2 r_2 r_3 + \lambda^2 r_3{}^2$$

$$= [(\lambda+2\mu) r_1 + \lambda r_2 + \lambda r_3]^2$$

$$= [(\lambda+2\mu)\,(a + \frac{S_x}{2\sqrt{J_2}}) + \lambda\,(a + \frac{S_y}{2\sqrt{J_2}}) + \lambda(a + \frac{S_z}{2\sqrt{J_2}})]$$

$$= [3\lambda a + \frac{\lambda}{2\sqrt{J_2}}\,(S_x+S_y+S_z) + 2\mu a + \frac{2\mu}{2\sqrt{J_2}}\,(S_x)]^2$$

$$= [3\lambda a + 2\mu a + \frac{\mu}{\sqrt{J_2}} S_x]^2$$

$$= [a(3\lambda + 2\mu) + \frac{\mu}{\sqrt{J_2}} S_x]^2 = E_{11}^2 \qquad \ldots (9)$$

$$Y_{22} = \lambda X_{21} + (\lambda+2\mu)X_{22} + \lambda X_{23}$$

$$= \lambda[\lambda r_1^2 + (\lambda+2\mu)r_1 r_2 + \lambda r_1 r_3] + (\lambda+2\mu)\ [\lambda r_1 r_2$$

$$+ (\lambda+2\mu)r_2^2 + \lambda r_2 r_3] + \lambda[\lambda r_1 r_3 + (\lambda+2\mu)r_2 r_3 + \lambda r_3^2]$$

$$= [(\lambda+2\mu)r_2 + \lambda r_1 + \lambda r_3]^2$$

$$= [(\lambda+2\mu)\ (a + \frac{S_y}{2\sqrt{J_2}}) + \lambda(a + \frac{S_x}{2\sqrt{J_2}}) + \lambda(a + \frac{S_z}{2\sqrt{J_2}})]^2$$

$$= [a(3\lambda + 2\mu) + \frac{\mu}{\sqrt{J_2}} S_y]^2 = E_{22}^2 \qquad \ldots (10)$$

$$Y_{14} = [(\lambda+2\mu)r_1 r_4 + \lambda r_2 r_4 + \lambda r_3 r_4]$$

$$= \mu\lambda(r_1 r_4 + r_2 r_4 + r_3 r_4) + 2\mu^2 r_1 r_4$$

$$= \{\mu\lambda[(a + \frac{S_x}{2\sqrt{J_2}}) + (a + \frac{S_y}{2\sqrt{J_2}}) + (a + \frac{S_z}{2\sqrt{J_2}})$$

$$+ 2\mu^2\ (a + \frac{S_x}{2\sqrt{J_2}})\}\ \frac{\tau_{xy}}{\sqrt{J_2}}$$

$$= [3\mu\lambda a + 2\mu^2 a + \mu^2 \frac{S_x}{\sqrt{J_2}}]\ \frac{\tau_{xy}}{\sqrt{J_2}}$$

$$= [a\mu(3\lambda + 2\mu) + \mu^2 \frac{S_x}{\sqrt{J_2'}}] \frac{\tau_{xy}}{\sqrt{J_2'}}$$

$$= [a(3\lambda+2\mu) + \mu \frac{S_x}{\sqrt{J_2'}}] \frac{\mu\tau_{xy}}{\sqrt{J_2'}} = E_{11} E_{12} \qquad \ldots (11)$$

$$Y_{24} = \mu X_{24}$$

$$= \mu[\lambda r_1 r_4 + (\lambda+2\mu) r_2 r_4 + \lambda r_3 r_4]$$

$$= [\lambda\mu[r_1+r_2+r_3] + 2\mu r_2] r_4$$

$$= [a(3\lambda+2\mu) + \mu \frac{S_y}{\sqrt{J_2'}}] \frac{\mu\tau_{xy}}{\sqrt{J_2'}} = E_{22} E_{12} \qquad \ldots (12)$$

$$Y_{33} = \lambda X_{31} + \lambda X_{32} + (\lambda+2\mu) X_{33}$$

$$= \lambda[\lambda r_1^2 + \lambda r_2 r_1 + (\lambda+2\mu) r_1 r_3]$$

$$+ \lambda[\lambda r_1 r_2 + \lambda r_2^2 + (\lambda+2\mu) r_3 r_2]$$

$$+ (\lambda+2\mu) [\lambda r_1 r_3 + \lambda r_2 r_3 + (\lambda+2\mu) r_3^2]$$

$$= 2\lambda(\lambda+2\mu) [r_2 r_3 + r_1 r_3] + (\lambda+2\mu)^2 r_3^2 + \lambda^2 r_2^2 + \lambda^2 r_1^2$$

$$+ 2\lambda^2 [r_1 r_2]$$

$$= [(\lambda+2\mu) r_3 + \lambda r_2 + \lambda r_1]^2$$

$$= [(\lambda+2\mu)\ (a + \frac{S_z}{2\sqrt{J_2'}}) + \lambda(a + \frac{S_y}{2\sqrt{J_2'}}) + \lambda(a + \frac{S_x}{2\sqrt{J_2'}})]^2$$

$$= [a(3\lambda+2\mu) + \frac{\mu}{\sqrt{J_2'}}\ S_z]^2 = E_{33}^2 \qquad \ldots (13)$$

$$Y_{34} = \mu X_{34}$$

$$= \mu[\lambda r_1 r_4 + \lambda r_2 r_4 + (\lambda+2\mu) r_3 r_4]$$

$$= [\lambda\mu[r_1+r_2+r_3] + 2\mu^2 r_3] r_4$$

$$= [a(3\lambda+2\mu) + \mu\ \frac{S_z}{\sqrt{J_2'}}]\ \frac{\mu\tau_{xy}}{\sqrt{J_2'}} = E_{33}\ E_{12} \qquad \ldots (14)$$

$$Y_{44} = \mu X_{44}$$

$$= \mu[\mu {r_4}^2]$$

$$= (\mu r_4)^2 = \left(\mu\ \frac{\tau_{xy}}{\sqrt{J_2'}}\right)^2 = E_{12}^2 \qquad \ldots (15)$$

Similarly the terms Y_{12}, Y_{13} and Y_{23} are derived as follows:

$$Y_{12} = \lambda X_{11} + (\lambda+2\mu) X_{12} + \lambda X_{13}$$

$$= \lambda^2({r_1}^2 + {r_2}^2 + {r_3}^2) + 2\lambda(\lambda+\mu)\ (r_1 r_2 + r_2 r_3 + r_1 r_3)$$

$$+ 2\mu\lambda(r_1^2 + r_2^2) + 4\mu^2 r_1 r_2 + 2\mu\lambda r_1 r_2$$

$$= [\lambda(r_1+r_2+r_3)]^2 + 2\mu\lambda[3a^2 + \frac{1}{4J_2}(S_xS_y + S_yS_z + S_zS_x)]$$

$$+ 2\mu\lambda(r_1 + r_2)^2 + 4\mu^2 r_1 r_2 + 2\mu\lambda r_1 r_2$$

Putting $S_z = -(S_x+S_y)$ one obtains

$$Y_{12} = [\lambda(r_1 r_2+r_3)]^2 + 2\mu\lambda[3a^2 + \frac{1}{4J_2}(-S_x^2-S_y^2-S_xS_y)]$$

$$+ 2\mu\lambda\ [a^2 + \frac{S_x^2}{4J_2} + \frac{aS_x}{\sqrt{J_2}} + a^2 + \frac{S_y^2}{4J_2} + \frac{aS_y}{\sqrt{J_2}}]$$

$$+ 4\mu^2 r_1 r_2 + 2\mu\lambda r_1 r_2$$

$$= (3a\lambda)^2 + 2\mu\lambda(3a^2) + 2\mu\lambda[3a^2 + \frac{3a(S_x+S_y)}{2\sqrt{J_2}}]$$

$$+ 4\mu^2 r_1 r_2$$

$$= (3a\lambda)^2 + 2\mu\lambda[6a^2 + \frac{3a(S_x+S_y)}{2\sqrt{J_2}}] + 4\mu^2\ (a + \frac{S_x}{2\sqrt{J_2}})\ (a + \frac{S_y}{2\sqrt{J_2}})$$

$$= a^2[3\lambda+2\mu]^2 + \frac{\mu a}{\sqrt{J_2}}\ (3\lambda+ 2\mu)\ (S_x+S_y) + \mu^2\ \frac{S_xS_y}{J_2}$$

$$= [a(3\lambda+2\mu) + \frac{\mu S_x}{\sqrt{J_2}}]\ [a(3\lambda+2\mu) + \frac{\mu S_y}{\sqrt{J_2}}] = E_{11}\ E_{12} \quad \ldots\ (16)$$

Following the derivation for Y_{12}, it can be shown

$$Y_{13} = \lambda X_{11} + \lambda X_{12} + (\lambda+2\mu)X_{13}$$

$$= [a(3\lambda+2\mu) + \frac{\mu S_x}{\sqrt{J_2}}]\ [a(3\lambda+2\mu) + \frac{\mu S_z}{\sqrt{J_2}}] = E_{11}\ E_{33} \quad \ldots (17)$$

and finally

$$Y_{23} = \lambda X_{21} + \lambda X_{22} + (\lambda+2\mu)X_{23}$$

$$= [a(3\lambda+2\mu) + \frac{\mu S_y}{\sqrt{J_2}}]\ [a(3\lambda+2\mu) + \frac{\mu S_z}{\sqrt{J_2}}] = E_{22}\ E_{33} \quad \ldots (18)$$

APPENDIX - C

DESCRIPTION OF COMPUTER PROGRAM' OPFA'

The element library at present consists of a four noded isoparametric quadrilateral element (Zienkiewiez, 1971) specified in the plane of analysis (r - z, or y - z plane). The computer programme has features viz dynamic storage, automatic core size adjustment, element and nodal point generation, an equation block solver and a variety of output options.

The following steps describe in general the procedure normally adopted in a two-dimensional soil-structure interaction analysis with the programme OPFA:

Step 1 Read general information such as title, number of elements, number of nodal points etc;

Step 2 Read input data such as nodal point and element data, material properties for constitutive modelling and boundary conditions;

Step 3 Read input data for the forcing function (time history of loading) and transfer these data to create global load vector in time domain;

Step 4 Form total mass matrix;

Step 5 Form total stiffness matrix for either undrained or drained condition;

Step 6 Form total damping matrix which includes the usual damping matrix for the solid part and dissipation matrix for the fluid flow;

Step 7 Solve the system of equations of motion, linear or non-linear in the time domain.

Based on the input information as provided for the constitutive modelling of the soil beneath the foundation, the programme OPFA branches to different subsections.

For a linear elastic analysis, the global stiffness matrix for static or dynamic analysis is formed only once and the information regarding responses are passed to Step 8.

For equivalent linear analysis, the effective shear strains are computed for each element and based on these shear strain amplitudes, strain-compatible soil properties such as shear moduli and damping ratios are determined. These values are compared with the initial properties as used in the analysis and steps 4, 5, 6, 7 are repeated with new properties if required. Once the desired convergence is achieved, response information such as displacements are passed to Step 8.

For an incremental nonlinear analysis, the global stiffness matrix is updated at each time step based on the stress-state of each element. If any element has yielded while using a particular failure model, an elastic-plastic constitutive matrix is assigned to that element. However, elements which have not yielded, are assigned the usual elastic constitutive matrices. During each time or load step increment, an iterative approach in combination with the well known modified Newton-Raphson (MNR) technique is used with the total applied load subdivided into a finite number of load increments. For a dynamic analysis this means the evaluation of the effective load vector at each time step.

For each load/time increment, the incremental deflection is determined using the modified Newton-Raphson technique. If the solution at the start of a load increment is known, the tangent stiffness matrix at that point can be computed and used to obtain an initial linear estimate of the deflection at the end of the increment. However, since the load-deflection or stress-strain curve of the material is nonlinear, the actual load the system can support at this deflection is less than the applied load and thus a load unbalance exists at the end of the first iteration. The unbalanced loads are applied to the system and a revised estimate of the incremental deflection is determined using the same tangent stiffness matrix evaluated at the previous point. Once the desired convergence is achieved, response information such as displacements are passed to Step 8.

In the shakedown analysis an elastic analysis is first carried out for all possible load combinations and initial elastic limit loads are determined for various loading combinations. Next the global compatibility matrix, and constitutive matrix are formed in order to solve the mathematical optimization problem. The result obtained provides the shakedown load factor.

Step 8 Compute the final stress response from the converged displacement using stress-displacement matrix.

The detailed description regarding the internal structure of the programme OPFA and an user's guide with detailed comments on input data are given in the following sections of the Appendix.

The programme OPFA is written in FORTRAN IV language and was originally developed on the IBM-370/158 computer at the Memorial University of Newfoundland, St. John's, Canada.

C.2 OPERATIONAL MODES

Basically the programme OPFA can be used in five different modes of operation. These modes are controlled by a parameter (variable) called 'NDYN'. Depending on the value 'NDYN' as input by the user, the particular type of analysis mode will be selected for the entire problem. These modes are described below:

a) NDYN. EQ.1 - Dynamic analysis with time history of loading function only (e.g. wave loading or any other type of loading)
b) NDYN. EQ.3 - Calculation of eigenvalues and eigenvetors only ie. eigenvalue analysis
c) NDYN. EQ.4 - Static analysis only
d) NDYN. EQ.5 - Transient phenomenon only e.g. consolidation problem

The material modelling is primarily controlled by a parameter called 'KKS'. The following sets of value for 'KKS' can be used in order to handle a specific type of material model. Table C1 presents the various combinations of material modellings (e.g. elastic, equivalent linear, elastic-plastic and piecewise linear) that can be used with different modes of analyses.

a) KKS. EQ.1 - Linear elastic analysis (constant soil properties)
b) KKS. EQ.2 - Equivalent linear soil model with strain compatible soil properties (e.g. variable stiffness and damping)
c) KKS. EQ.3 - Nonlinear soil model with incremental elastic-plastic analysis
d) KKS. EQ.4 - Shakedown analysis with piecewise linearized yield surface for soil; linear programming technique

C.3 SUBPROGRAMMES

The programme OPFA consists of a main programme and 61 subroutines. The flow chart of OPFA is shown in Fig. C1, and the calling sequences of the different subroutines are given in Table C2. This table also describes which subroutines control the 'READ' and/or 'WRITE' operation of the twenty-one (21) logical tape devices used as temporary storage files and the two physical tapes (or mass storage files) used for input and output of data. The main programme and 9 other major subroutines control the basic sequences of operations.

TABLE C1 - MODES OF OPERATIONS IN PROGRAM OPFA

MODE OF OPERATION	DESCRIPTION OF OPERATION	CONSTITUTIVE MODELLING COMBINATION	REMARKS
Mode 1 NDYN = 1	Time history analysis with superstructure loading	- Linear (KKS = 1) - Equivalent linear (KKS = 2) - Incremental analysis (KKS = 3) with nonlinear soil model - Shakedown analysis with piecewise linearized yield surface for soil (KKS = 4)	
Mode 3 NDYN = 3	Frequency analysis; Eigenvalues and Eigenvectors	- Linear soil model (KKS = 1) - Eqivalent linear Soil model (KKS = 2)	
Mode 4 NDYN = 4	Static or quasi - static response analysis only	- Linear (KKS = 1) - Incremental analysis (KKS = 3) with nonlinear soil model - Shakedown analysis with piecewise linearized yield surface for soil (KKS = 4)	
Mode 5 NDYN = 5	Transient static response analysis only e.g. Consolidation problem	- Linear (KKS = 1) - Incremental analysis with nonlinear soil (KKS = 3) model	

FIG. CI — FLOW DIAGRAM FOR OPFA.

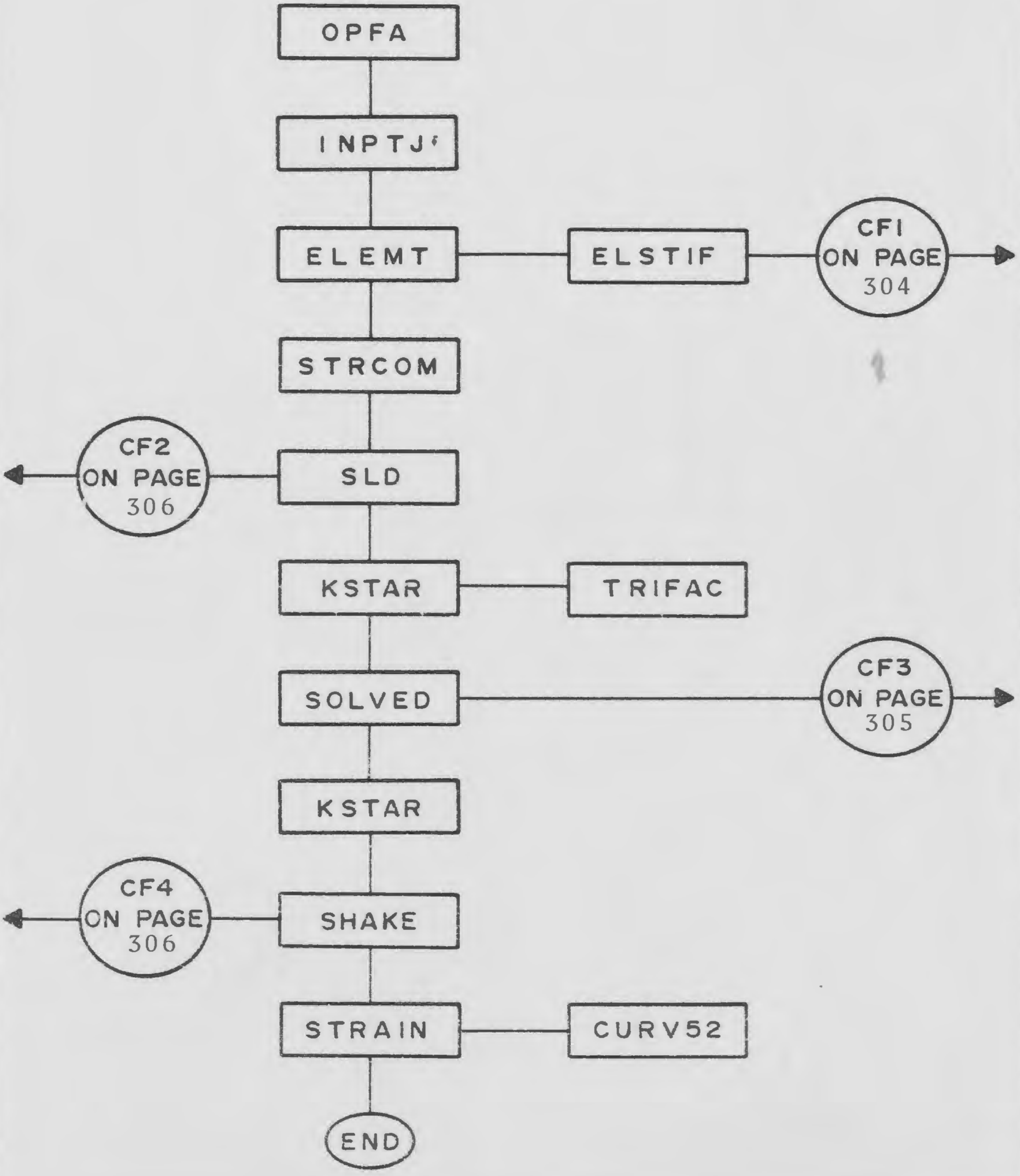


FIG. CI — FLOW DIAGRAM FOR OPFA. (CONT'D.)

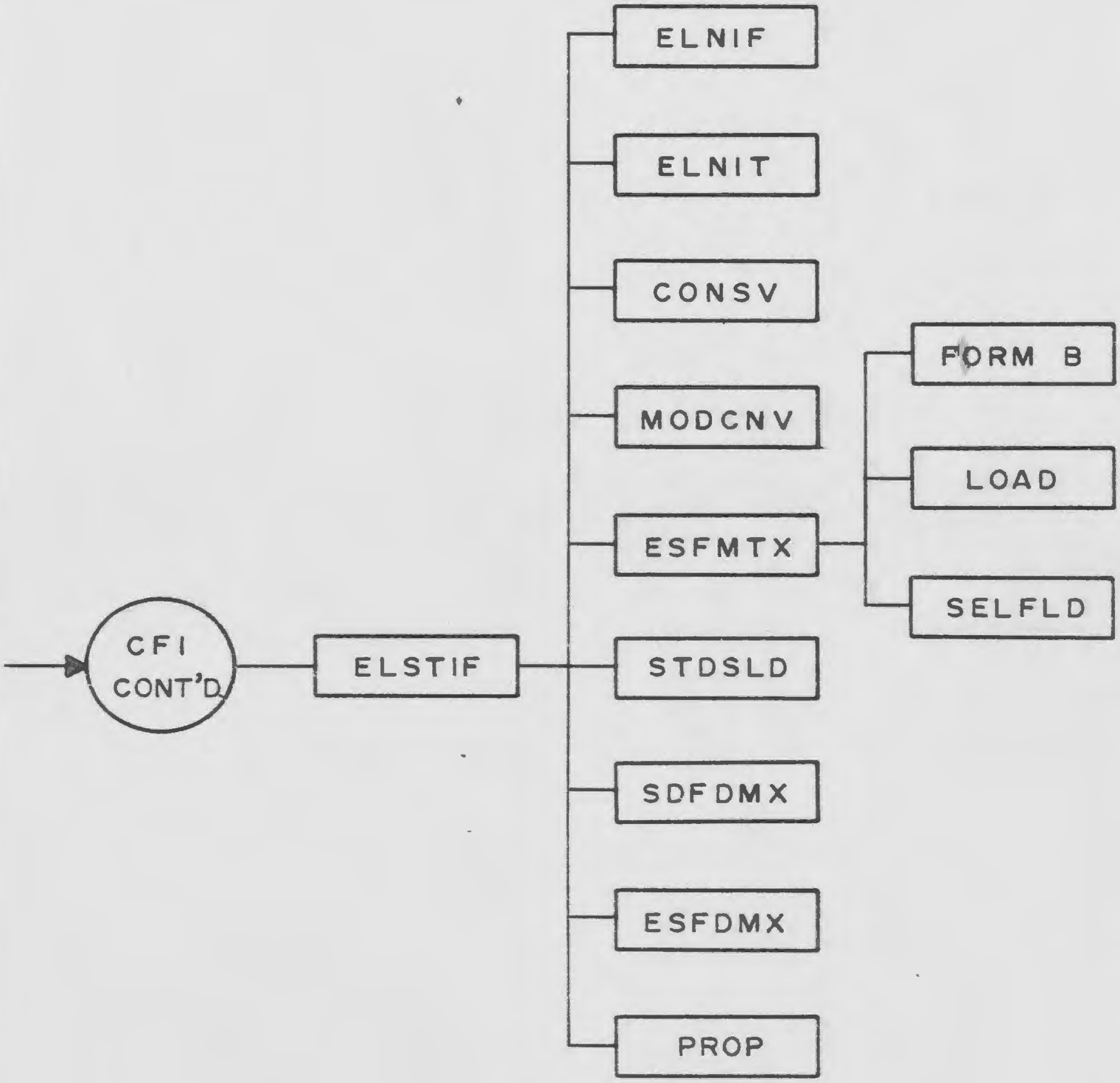


FIG. C1 — FLOW DIAGRAM FOR OPFA. (CONT'D.)

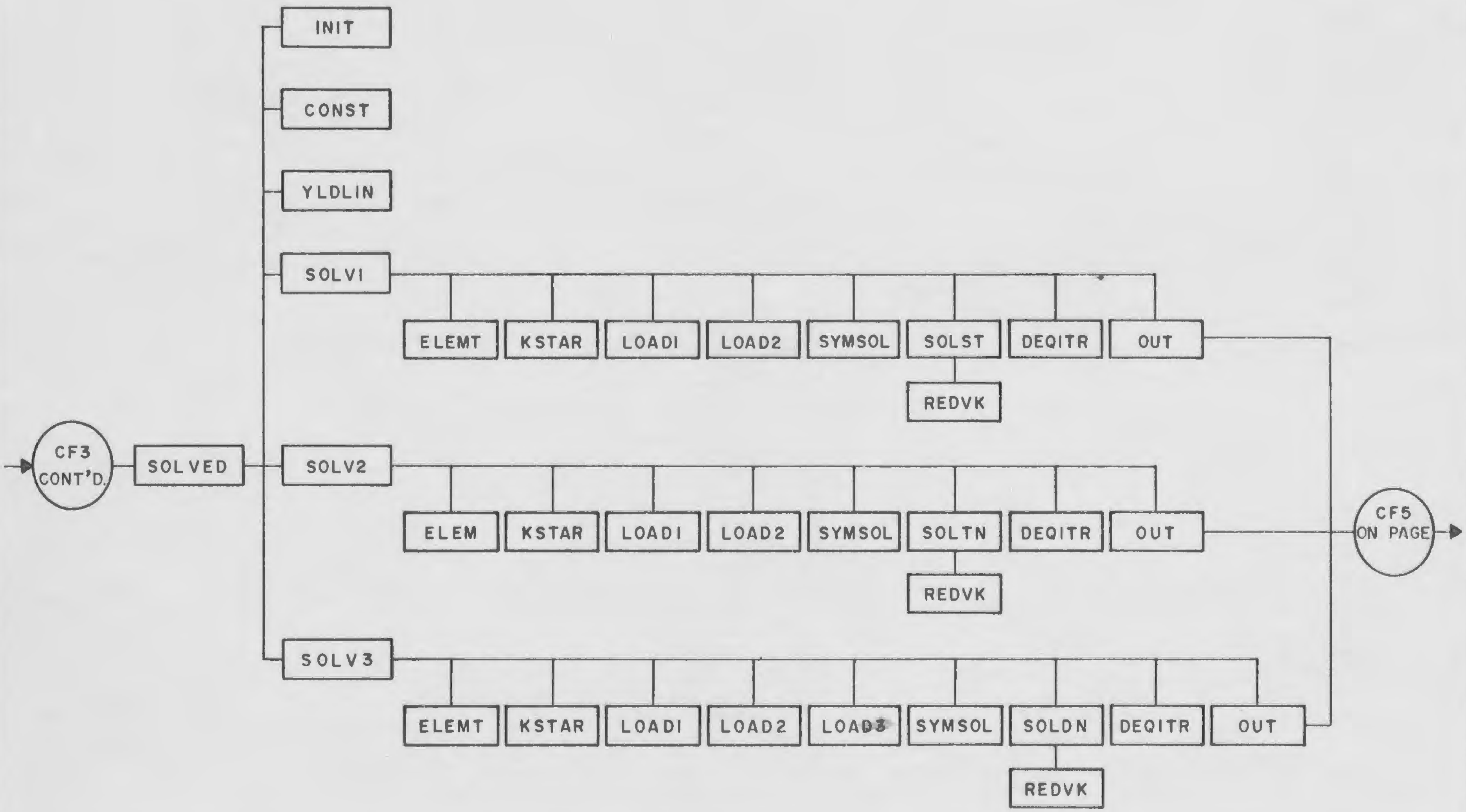


FIG. CI — FLOW DIAGRAM FOR OPFA. (CONT'D.)

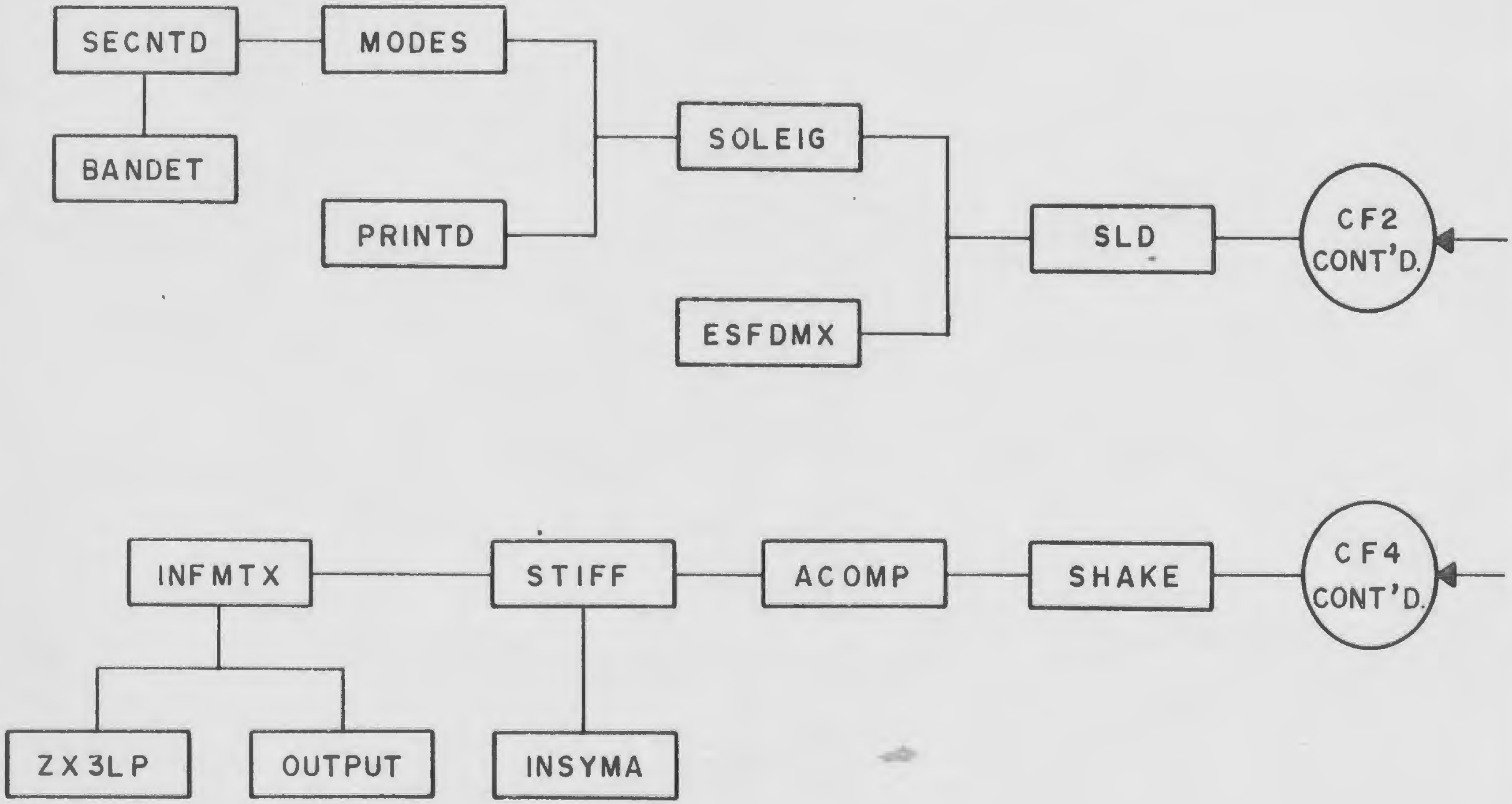


FIG. C1 — FLOW DIAGRAM FOR OPFA. (CONT'D.)

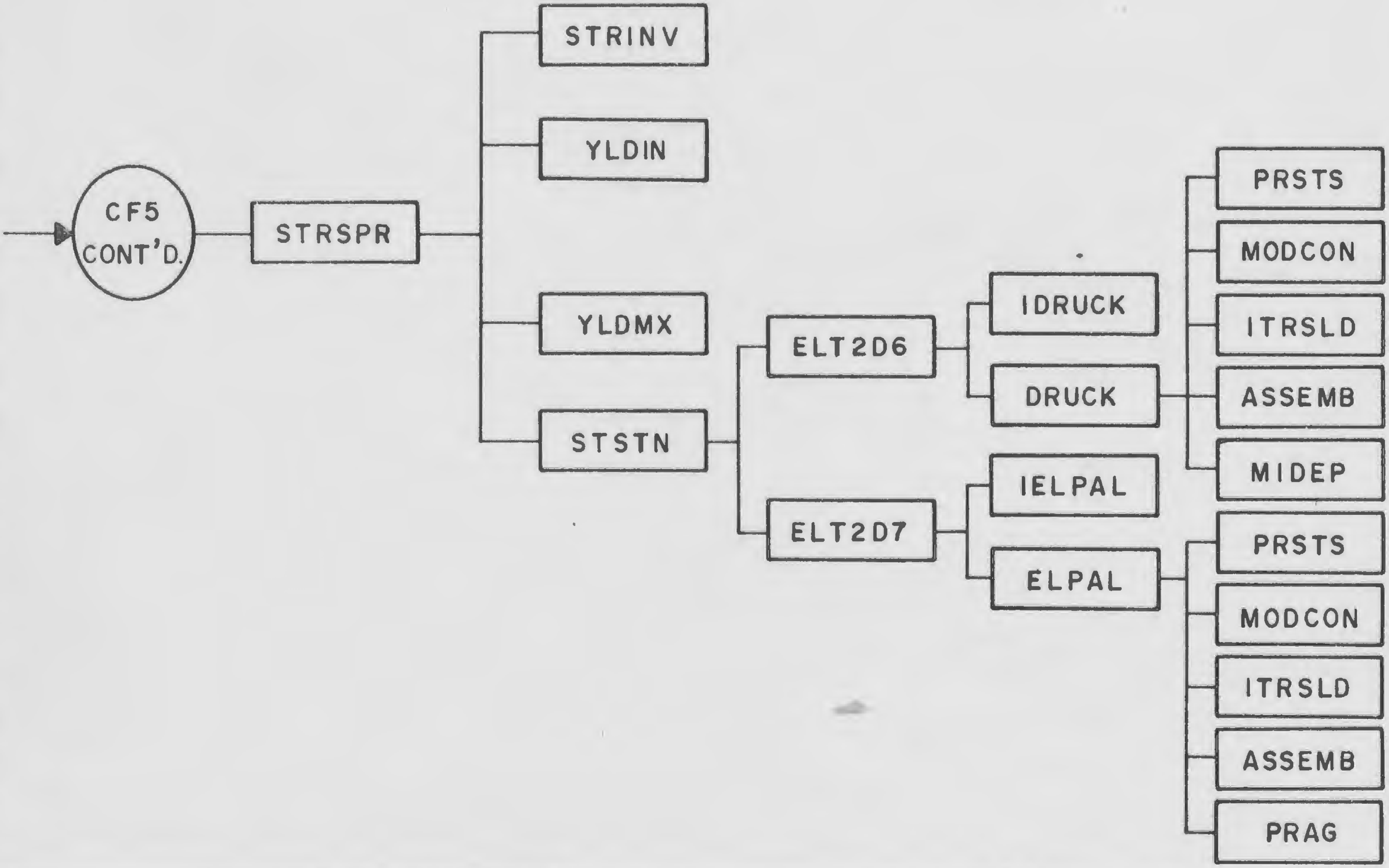


TABLE C2 - CALLING SEQUENCE FOR SUBROUTINES AND TAPES IN PROGRAM OPFA

NAME OF ROUTINE	ROUTINE CALLS	ROUTINE CALLED BY	ROUTINE WRITES ON TAPE	ROUTINE READS FROM TAPE
ELT2D6	IELPAL & ELPAL	STSTN		
ELT2D7	IDRUCK & DRUCK	STSTN		
ELSTIF	ELNIF, ELINT, CONSV, MODCNV, ESFMTX, STDSLD, SDFDMX, ESFDMX, PROP	ELEMT	11, 13	
ELNIF		ELSTIF	14	14
ELINT		ELSTIF		
ESFDMX		ELSTIF		
ESFMTX	FORMB, LOAD MODCNV, SELFLD	ELSTIF		
FORMB		ESFMTX		
EQCHK		SOLV1, SOLV2, & SOLV3		60
IDRUCK		ELT2D7		
IELPAL		ELT2D6		
INFMTX	ZX3LP & OUTPUT	SHAKE		

TABLE C2 - CALLING SEQUENCE FOR SUBROUTINES AND TAPES IN PROGRAM OPFA (CONT'D)

NAME OF ROUTINE	ROUTINE CALLS	ROUTINE CALLED BY	ROUTINE WRITES ON TAPE	ROUTINE READS FROM TAPE
INIT		SOLVED		
INPUTJ		MAIN		
INSYMA		STIFF		
ITRSLD		ELPAL, DRUCK STRSPR		
KSTAR	TRIFAC		4,9,60	13
LOAD		ESFMTX		
LOAD1		SOLV1, SOLV2 & SOLV3		
LOAD2		SOLV1, SOLV2 & SOLV3		
MAIN	(See Section A.3.1)			
MODCNV		ELSTIF		
MODES	SECNTD & BANDET	SOLEIG		10
MIDEP		ELPAL		
PRAGER		DRUCK		

TABLE C2 - CALLING SEQUENCE FOR SUBROUTINES AND TAPES IN PROGRAM OPFA (CONT'D)

NAME OF ROUTINE	ROUTINE CALLS	ROUTINE CALLED BY	ROUTINE WRITES ON TAPE	ROUTINE READS FROM TAPE
PRINTD		SOLEIG		8
PROP		ELSTIF		
PRSTS		ELPAL & DRUCK		
OUT	ASSEMB	SOLV1, SOLV2 & SOLV3	55	55, 57, 18
OUTPUT	CHECK	INFMTX		
REDVK		SOLST, SOLTN, DEQITR, SOLDN		3
MULT		BANDET		
MAXMIN		ELPAL, DRUCK		
MDCNEQ		STRSPR	17	
SDFDMX		ELSTIF	51	
SECNTD	BANDET	MODES	8	9, 4,
SELFLD		ESFMTX		
SHAKE	ACOMP, STIFF, INFMTX	MAIN		

TABLE C2 - CALLING SEQUENCE FOR SUBROUTINES AND TAPES IN PROGRAM <u>OPFA</u> (CONT'D)

NAME OF ROUTINE	ROUTINE CALLS	ROUTINE CALLED BY	ROUTINE WRITES ON TAPE	ROUTINE READS FROM TAPE
SLD	SOLEIG & ESFDMX	MAIN	61, 4, 9, 13	51, 22, 61,
SOLDN	REDVK	SOLV3		
SOLEIG	MODES & PRINTD	SLD	14	14
SOLST	REDVK	SOLV1		
SOLTN	REDVK	SOLV2		
SOLVED	INIT, CONST, YLDLIN, SOLV1, SOLV2 & SOLV3	MAIN		
SOLV1	ELEMT, KSTAR, LOAD1, LOAD2, SYMSOL, SOLST, DEQITR, OUT	SOLVED		
SOLV2	ELEMT, KSTAR, LOAD1, LOAD2, SYMSOL, SOLTN, DEQITR, OUT	SOLVED		
SOLV3	ELEMT, KSTAR, LOAD1, LOAD2, SYMSOL, SOLDN, DEQITR, OUT	SOLVED	12	12
STDSLD	VECTOR	ELSTIF	51	
STIFF	INSYMA	SHAKE		

TABLE C2 - CALLING SEQUENCE FOR SUBROUTINES AND TAPES IN PROGRAM <u>OPFA</u> (CONT'D)

NAME OF ROUTINE	ROUTINE CALLS	ROUTINE CALLED BY	ROUTINE WRITES ON TAPE	ROUTINE READS FROM TAPE
STRAIN	CURV52	MAIN	22	51,22
STRSPR	YLDMX, STSTN, ITRSLD	OUT	18, 57	10, 11, 17, 22, 50
STSTN	ELT2D6 & ELT2D7	STRSPR		
SYMSOL		SOLV1, SOLV2 & SOLV3		
TRIFAC		KSTAR	3	4, 1, 8,
YLDIN		STRSPR		
YLDLIN		SOLVED		
YLDCHK		MAIN		11
YLDMX		STRSPR		
STRINV		STRSPR		
ZX3LP		INFMTX		
VECTOR		STDSLD		

C.3.1 PROGRAM CONTROL

The principal subroutines are detailed in this section and a brief description of the remaining subroutines are given in Table C3.

The programme OPFA calls the main subroutines INPUTJ, ELEMT, STRCOM, SLD, KSTAR, SOLVED, YLDCHK, SHAKE and STRAIN.

The subroutine INPUTJ reads or generates the input data for nodal point coordinates and boundary conditions. For each node the maximum number of allowable displacement boundary conditions that can be prescribed is four i.e. two degrees of freedom (D.O.F.) for solid and two degrees of freedom (D.O.F.) for fluid with respect to solid. These boundary conditions are created by means of a two dimensional array called the nodal connectivity array or ID-array. Based on the active degrees of freedom for each node, the total number of equations for the entire problem is created in this routine. This routine returns with the information on nodal point coordinates, degrees of freedom array (ID-array) and the total number of equations.

The subroutine ELEMT via ELSTIF reads the element information such as nodal connection, material constitutive properties such as bulk modulus, shear modulus, porosity, permeability, fluid bulk modulus, densities of bulk solid and fluid, damping information etc. Based on these properties, this subroutine computes and stores on low speed tapes the following sets of information for each element.

a) element mass matrix (only lumped masses)

b) element stiffness matrix including solid, fluid and the solid-fluid coupling submatrices

c) element damping matrix including the dissipation matrix defined as the function of the permeability of the soil medium

d) element stress-displacement matrix

The subroutine <u>STRCOM</u> reads all material properties information related to the equivalent linear soil model for a dynamic finite element analysis. The equivalent linear soil model is primarily based on strain compatible soil properties and this routine reads these properties such as maximum shear moduli, maximum damping ratios, initial shear moduli, poisson's ratios etc.

The subroutine <u>SLD</u> formulates the solid mass and stiffness matrices either for an equivalent linear analysis or for a shakedown analysis assuming an undrained or drained condition. In the equivalent linear soil model, the fundamental frequency is computed based on the global mass and stiffness matrices. Using this frequency, Rayleigh damping coefficents are evaluated for each element and thus damping matrices for the entire finite element model can be evaluated. For shakedown analysis, this subroutine returns the global assembled stiffness matrix for the two-phase medium either in the undrained or/drained condition as specified.

The subroutine KSTAR assembles all matrices such as mass, stiffness, damping and load vector computed for each element by imposing the prescribed boundary condition. These matrices are read from a low speed storage tape. The solution is normally sought by in-core method and as such these are stored on a low speed storage tape. This routine also returns the decomposed effective assembled stiffness matrix $[\hat{K}]$ using a standard tridiagonal factorization (LDL^T) method. In the incremental nonlinear analysis, the stiffness matrix needs to be updated at every time or load step increment and therefore this routine will be called and updated 'NCYCL' number of times where 'NCYCL' represents the total number of time or load increment steps. However, for other types of analyses, such as elastic, equivalent linear or shakdeown, the global stiffness matrix is constant over the entire time duration of the loading period and therefore needs no updating.

The subroutine <u>SOLVED</u> initializes all variables necessary for the solution of the equations of motion. It also sets up all constants required for the step-by-step integration scheme.

For linear analysis ie. 'KKS = 1', the equations of motion are solved and the stresses and strains are computed from the nodal displacement vector either for a single time step (NDYN = 4) or for the entire time history (NDYN = 1).

For equivalent linear analysis ie. 'KKS = 2', the equations of motion are solved in the time domain and the strain history for each element is determined from the nodal displacement vector.

For incremental nonlinear analysis, ie. 'KKS = 3', the equations of motion are solved in an iterative manner and element strains and stresses are determined from the converged nodal displacement vector obtained from the last iteration.

For shakedown analysis i.e. 'KKS = 4', the linear elastic analysis is performed first in order to determine the initial elastic limit load. This is done fo all possible combination of loadings. The global compatibility matrix is then formed in order to solve the mathematical optimization problem.

C.4 TAPES

The programme OPFA uses a total of 21 tapes, as shown in Table C4. The two physical tapes, #5 and #6 are used to read input data from, and/or write output data on magnetic tapes or mass storage files. The twenty-one (21) logical tape devices i.e. tapes 1 to 4, 8 to 14, 17, 18, 21, 22, 50, 51, 55, 57, 60 and 61 are used for low speed storage of temporary data during the execution mode of the programme. The content of these tapes is described in Table C4. As described previously, Table C2 indicates which subroutines control the 'READ' and/or 'WRITE' operations of all tapes used with the exception of tapes 5, 6 and 7. These three tapes are input and output files which represent a card reader, on-line printer and on-line card punch, respectively.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAM OPFA

NAME OF ROUTINE	DESCRIPTION OF OPERATION
ACOMP	computes global assembled compatibility and constitutive matrices for the shakedown analysis.
ASSEMB	assembles and forms the global residual load vector for incremental nonlinear analysis.
BANDET	extracts the first eigenvalue of the system.
CHECK	computes the global residual force vector for the system.
COMPGT	computes the strain-displacement relationship at each gauss point.
CONST	computes the constants for numerical time step integration.
CONSV	computes the elasticity constants of the constitutive matrix for each element under drained and/or undrained condition.
CURV52	computes the reduction factor for shear modulus and damping ratio for each element based on the maximum shear strain evaluated from the strain history.
DEQITR	performs the equilibrium iteration at each load or time step.
DRUCK	computes incremental strain and stress; checks the DRUCKER-PRAGER yield condition and forms the elastic or elastic-plastic constitutive matrix depending on the stress state of the particular element at the current load and/or time step.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAM OPFA (CONT'D)

NAME OF ROUTINE	DESCRIPTION OF OPERATION
ELEMT	for description see Section C.3.
ELPAL	computes incremental strain and stress; checks the Von Mises yield condition and forms at each time and/or load step, the elastic or elastic-plastic constitutive matrix for each element depending on the stress state of the particular element.
ELINT	initializes all variables related to the formulation of the element matrices.
ELNIF	reads all necessary information related to each element, eg. element nodal connectivity, material properties, etc.
ELSTIF	computes element mass matrices (solid and fluid), stiffness matrices (solid, fluid and solid-fluid coupling) and damping matrices (solid and fluid).
ELT2D6	organizes the storage locations for all variables related to the elastic-plastic analysis based on Von Mises yield condition.
ELT2D7	organizes the storage locations for all variables related to the elastic-plastic analysis based on Drucker-Prager yield condition.
EQCHK	checks the overall equilibrium at each load/time step.
ESFDMX	Forms element matrices with respect to solid-fluid coupling.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAM OPFA (CONT'D)

NAME OF ROUTINE	DESCRIPTION OF OPERATION
ESFMTX	computes element submatrices such as mass, damping and stiffness with respect to solid and fluid part.
IDRUCK	initializes all variables before assigning them to 'DRUCK'.
IELPAL	initialize all variables before assigning them to 'ELPAL'.
INFMTX	sets up the 'tableau' based on dual problem and finds the shakedown load factor by calling a standard linear programming routine (ZX3LP).
INIT	initializes all variables before assigning them to the relevant routines for the solution of the equations of motion.
INPUTJ	reads all nodal point data such as node number, degrees of freedom and coordinates.
INSYMA	inverts a full matrix of size (NEQ X NEQ).
ITRSLD	computes the nodal residual load on each element based on the stresses evaluated at each gauss point.
KSTAR	for description see Section C.3.1.
LOAD	computes residual load vector (nodal) for each element based on the stress states at each gauss point. This routine is required for elastic-plastic analysis.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAMME <u>OPFA</u> (CONT'D)

NAME OF ROUTINE	DESCRIPTION OF OPERATION
LOAD1	reads time histories of surface loading and transfers these loadings to global force vector in time domain.
LOAD2	reads time histories of concentrated loads and transfers these loadings to global force vector in time domain.
MAIN	for description see Section C.3.1.
MDCNEQ	computes the elasticity constants for equivalent linear analysis.
MODCNV	reallocate the elastic-plastic constitutive matrix for subsequent iteration.
MODCON	reallocate the stress vector for elastic-plastic analysis.
MODES	organizes the storage allocations for the eigenvalue analysis.
MIDEP	computes the incremental stress-strain relationship using the Von Mises yield condition.
PRAGER	computes the incremental stress-strain relationship using the Drucker-Prager yield condition.
PRINTD	prints the eigenvalues and eigenvectors.
PROP	reads the material properties such as hardening, yield strength, cohesion, and angle of internal friction.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAMME <u>OPFA</u> (CONT'D)

NAME OF ROUTINE	DESCRIPTION OF OPERATION
PRSTS	stores the stress computed at each gauss point in an array form.
OUT	prints all displacements, velocities, acceleration and also creates a punch file to save these data for subsequent plotting.
OUTPUT	computes permanent inelastic strains and prints them.
REDVK	performs the solution of the equilibrium equations after reading the decomposed stiffness matrix from the tape.
MULT	performs the multiplication of a matrix by a vector.
SDFDMX	computes the element undrained stiffness matrix and the associated solid-fluid coupling matrix.
SECNTD	computes the eigenvalues and eigenvectors of a banded matrix by the determinant search solution.
SHAKE	for description see Section C.3.1.
SLD	for description see Section C.3.1.
SOLDN	solves the dynamic equilibrium equations of motion at a particular load and/or time step.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAMME <u>OPFA</u> (CONT'D)

NAME OF ROUTINE	DESCRIPTION OF OPERATION
SOLEIG	reads the data required for an eigenvalue analysis.
SOLST	solves the static equilibrium equations at a particular time and/or load step.
SOLTN	solves the transient static equilibrium equations at a particular time and/or load step.
SOLVED	for description see Section C.3.1.
SOLV1	solves the static problem.
SOLV2	solves the time dependent quasi-static problem.
SOLV3	solves the dynamic problem.
STDSLD	computes the stress-displacement matrix and condenses the element stiffness matrix for incompatible nodes.
STIFF	converts the banded stiffness matrix to a full matrix before assigning it to 'INSYMA'.
STRAIN	for description see Section C.3.1.
STRCOM	for description see Section C.3.1.
STRSPR	computes element stress, strains from the nodal displacements.

TABLE C3 - BRIEF DESCRIPTION OF SUBROUTINES IN PROGRAMME OPFA (CONT'D)

NAME OF ROUTINE	DESCRIPTION OF OPERATION
STSTN	calls two key routines 'ELT2D6' and 'ELT2D7'.
SYMSOL	solves a set of simultaneous equations.
TRIFAC	decomposes the banded effective stiffness matrix.
YLDIN	initializes the variables before assigning them to routine 'YLDMX'.
YLDLIN	computes gradients of the piecewise linearized yield planes for each material type and writes the coefficients of the gradient matrices on a tape.
YLDCHK	computes the elastic limit load for each load case.
YLDMX	computes the maximum stress responses of each element normalized with respect to each yield plane.
STRINV	computes the normal strain components from stresses for each element.
VECTOR	computes the vector of a matrix.
ZX3LP	solves the linear programming problem using the revised simplex algorithm.

TABLE C4 - DESCRIPTION OF TAPES IN PROGRAMME '<u>OPFA</u>'

TAPE NUMBER	DESCRIPTION OF CONTENT
1	auxiliary tapes used at the time of eigenvalue analysis or at the time of the solution of equations of motion.
2	contains all control information regarding eigenvalue solution.
3	contains all information regarding the factorized decomposed global stiffness matrix for the entire problem.
4	contains the assembled global stiffness matrix in banded form.
8	contains all information regarding eigenvectors in the event an eigenvalue analysis is required; However, the same file is also used in the decomposition of the stiffness matrix for the solution of system of linear or nonlinear equations of motion. In the later case, the tape is primarily used as an auxiliary one.
9	contains the assembled global mass matrix.
10	contains information on given starting iteration vectors for an eighenvalue analysis; However, the same tape unit is also used to store the coefficients of the gradients of the piecewise linearized yield planes if a shakedown analysis is requested.
11	contains information on element stress-displacement matrices for solid

TABLE C4 - DESCRIPTION OF TAPES IN PROGRAMME '<u>OPFA</u>' (CONT'D.)

TAPE NUMBER	DESCRIPTION OF CONTENT
	and fluid parts, element connectivity array, elastic material properties, gauss point coordinates, and volume related to each gauss point.
12	contains information on load vectors in the event an equivalent linear analysis is carried out. In the first iteration the load vectors are formed once and written on this tape for use in subsequent iteration.
13	contains information on element matrices related to connectivity arrays, solid-fluid coupling matrices including solid and fluid stiffness matrices, solid and fluid mass matrices, damping matrices respectively.
14	contains all input data regarding element types, element nodal numbering, material types, element material properties such as Lame's constants, poisson's ratio, etc.
17	contains matrices of coefficients of elastic contants for all elements under drained condition.
18	contains the information on stresses for selected element groups for contour plotting.
21	contains the nodal load vector computed for each element based on the stress state at each gauss point.
22	contains information on maximum shear

TABLE C4 - DESCRIPTION OF TAPES IN PROGRAMME '<u>OPFA</u>' (CONT'D.)

TAPE NUMBER	DESCRIPTION OF CONTENT
	moduli, maximum damping ratios, and shear moduli values used for all elements. This tape is required for an equivalent linear soil model with the dynamic analysis option.
50	contains matrices of coefficients of elastic constants for all elements under the undrained condition. This tape is required in the event a shakedown analysis is performed.
51	contains element connectivity arrays, element stiffness matrices (solid and fluid) and mass matrices (solid and fluid); it also contains element undrained stiffness matrices, fluid stiffness matrices and coupling solid-fluid matrices, etc.
55	contains information on displacement, velocities and acceleration for selected nodal points at selected time steps. These data are later used for plotting purposes.
57	contain information on stresses, and strains on selected elements at specified time-steps. These data are later processed for plotting purposes.
60	contains global stiffness matrix for equilibrium check.
61	contains information regarding element matrices and properties, volume, densities and solid, fluid stiffness matrices including coupling terms; The tape is required in the event KKS = 2 ie. equivalent linear soil model.

C.5 COMMENTS ON PROGRAMME CAPACITY DYNAMIC STORAGE AND RUN TIME

The programme OPFA has been developed on an IBM 370 computer using FORTRAN IV language. During the execution mode of any problem, the core requirement is determined as a function of the following variable as described below:

NUMNP - Total number of nodal points;

NUMEL - Total number of elements;

NUMMAT - Total number of material types;

NCON - Number of parameters required to describe a particular soil model for incremental analysis.

IDWA - Number defining the storage requirements for each element stress-strain information together with the yield condition. (whether elastic or plastic). For KKS = 3, IDWA will vary from 10 to 11 depending on the material model;

NEQ - Number of equations in the finite element model;

MBAND - Half-band width;

NF - Number of frequencies to be evaluated;

NACL - 1;

LL - Number of maximum surface load cards used for all load cases;

NNYP - Number of elements * number of yield planes;

NNSC - Number of elements * number of stress components;

NLC - Number of load cases;

NYP - Number of yield planes;

NSC - Number of stress components.

The length of the dynamic storage array depends on the amount of data to be stored at different stages of the execution and hence will vary during a run. In order to account for this variation and to minimize the dynamic storage space required, the programme has been developed with a special feature. This feature allows the expansion or contractions of the core size required in accordance with the length of the blank common array, n, during the different execution steps.

C.5.1 High Speed Storage Requirements:

The high speed storage requirements of the programme can be changed depending on the size of the problem to be solved. This is done by changing the two

FORTRAN statements at the beginning of the OPFA programme ie.

```
          COMMON D(n)
          MTOT = n
```

The minimum value of n needed is computed as follows except for the shakedown analysis which is controlled by the parameter KKS. EQ.4 as described previously. For the shakedown analysis using the linear programming technique, OPFA requires a larger size of storage allocation even for a very small/moderate size problem. However, for all other types of analysis e.g. linear, equivalent linear and incremental nonlinear analysis, the storage requirement is controlled by the few internal parameters e.g. N1, N5, NK15, N011, NN15, NN17 and N43. The value of N1 is set as 1 at the beginning of the programme and the value of 'n' is determined from the parameter N43 ie. setting n equal to N43. The following key equations describe the determination of the internal parameters as listed above:

```
  N1 = 1
  N5 = N1 + 8 * NUMNP
NK15 = N5 + 11 * NUMMAT + NUMEL + NCON *
       NUMMAT + IDWA * NUMEL
N011 = NK15 + 3 * NEQ * MBAND + 3 * NEQ + 2
       * NEQ * MBAND
```

NN15 = N011 + (3 * NEQ + MBAND - 1) + 9 * NUMMAT

NN17 = NN15 + 1 + NEQ * MAXO (MBAND, NF + 3) + 16 * NEQ + 4 * (NF + 3) + 3 * NF + 2 * NUMNP

N43 = (NN17 or NN15) + 3 * NEQ + 8 * NUMNP (or NEQ) + 12 * NEQ + 2 * NACL + 18 * NUMEL + 10 * NUMNP + 4 * 2 * LL * 2 + 2 * NNYP + NNSC * NNYP + NLC + NYP + NYP * NYP

If the value of 'n' is set less than the required value as determined by N43, an error message is printed and programme execution is terminated indicating the particular value of the internal parameter and the name of the particular subroutine at termination.

APPENDIX - D

INPUT DATA TO 'OPFA'

I. HEADING CARD (10A8)

Notes	Columns	Variable	Description of Input
(1)	1-80	HED(8)	Read heading information to be printed with the output

NOTES/

(1) Begin each new data case with a new heading card.

II. MASTER CONTROL CARD (9I5)

Notes	Columns	Variable	Description of Input
(1)	1-5	NUMNP	Total number of nodal points
(2)	6-10	NUMEL	Total number of elements
(3)	11-15	NDYN	Analysis type code; EQ.1; Forced dynamic response analysis EQ.3; Eigenvalues/vector solutions EQ.4; Static analysis EQ.5; Quasi-static transient analysis
(4)	16-20	IPLNAX	EQ.-1; Plane stress model EQ.0; Plane strain model

II. MASTER CONTROL CARD - CONT'D

Notes	Columns	Variable	Description of Input
(5)	21-25	LL	GE.1; Maximum number of surface load cards for the entire problem
(6)	26-30	NF	Number of frequencies to be found in the eigenvalue analysis
			EQ.0; Static analysis
			GE.1; Dynamic analysis
(7)	31-35	MODEX	EQ.0; No data check
(8)	36-40	NAD	EQ.0; Superstructure not included in the analysis
			EQ.1; Superstructure included in the analysis
(9)	41-45	NGAMA	EQ.0; Body force not included

NOTES/

(1) Total number of nodal points; Nodal points are numbered sequentially from '1' to 'NUMNP'. For comments on nodal points data generation see section VIII.

(2) Total number of elements; Elements are numbered sequentially from '1' to "NUMEL'. For comments on element data generation see section XI.

II. MASTER CONTROL CARD - CONT'D

NOTES/ - CONT'D

(3) If NDYN. EQ.1 the programme performs a forced dynamic response analysis.

If NDYN. EQ.3 the programme performs an eigenvalue analysis only and returns with the eigenvalues and eigenvectors.

If NDYN. EQ.4 the programme performs a static analysis only.

(4) The variable 'IPLNAX' controls the type of two dimensional modelling which has to be selected for a particular problem.

(5) At least one (1) surface load card (LL.GE.1) is required whether or not the user inputs the surface pressure loading. For details see section XXII and XXIII.

(6) In the event of an equivalent linear analysis using 'KKS' = 2, (see next section for reference), NF should be greater than or equal to 1. This is required only when a dynamic equivalent linear analysis with strain compatible soil properties is carried out. However, for NDYN. EQ.3 (eigenvalue analysis only), NF shall also be greater than or equal to 1.

(8) If NAD. EQ.0, no superstructure/foundation is included in the analysis i.e. load is applied directly to the soil. However, by setting NAD.NE.0 a superstructure such as a concrete structure may be included in the analysis.

(9) 'NGAMA' EQ.0 represents soil overburden pressure i.e. body force has not been included in the analysis.

III OUTPUT PRINT & PUNCH CONTROL CARD (3I5)

Notes	Columns	Variable	Description of Input
(1)	1-5	ISTRPR	GE.1; Stress output interval
(2)	6-10	NPRINT	GE.1; Displacement, velocity and acceleration output interval
(3)	11-15	ISTART	GE.1; Stress/strain output interval for stress/strain contour plotting

NOTES/

(1) Every ISTRPR-th time point responses such as stresses, strains at each element will be printed and also saved for plotting purpose.

(2) Every NPRINT-th time point responses such as displacement velocity and acceleration for each nodal point will be printed and saved for subsequent plotting.

(3) Every ISTART-th time point, output data for stress/strain will be saved for contour plotting.

IV MASTER CONTROL CARD (I5, F10.0, I5, F10.0, 5I5)

Notes	Columns	Variable	Description of Input
(1)	1-5	NCYCL	GE.1; Total number of time steps or load steps
(2)	6-15	DT	GT.0; Solution time step for numerical integration,

IV MASTER CONTROL CARD - CONT'D

Notes	Columns	Variable	Description of Input
(3)	16-20	KKS	EQ.1; Elastic analysis
			EQ.2; Equivalent linear analysis with strain compatible soil properties (variable stiffness and damping)
			EQ.3; Incremental non-linear analysis with elastic-plastic soil model
			EQ.4; Shakedown analysis with piecewise linearized yield surface
(4)	21-30	FA	Gravitational constant (value of 'g')
(5)	31-35	NDRN	EQ.0; Undrained condition
			EQ.1; Undrained - drained condition (flow problem)
(6)	36-40	NCON	NCON = 2; Required only if KKS = 3. Otherwise set NCON = 1
(7)	41-45	IDW	IDW = 10 or 11 depending on the parameter 'Model' as described below. Required only if KKS = 3; otherwise set IDW = 1

IV <u>MASTER CONTROL CARD - CONT'D</u>

Notes	Columns	Variable	Description of Input
(8)	46-50	MODEL	EQ.1; Von Mises (IDW = 10) EQ.2; Drucker-Prager (IDW = 11) Required only when KKS = 3; otherwise leave blank
(9)	51-55	NINT	Number of integration points; set NINT = 2

NOTES/

(1) 'NCYCL' controls the toal number of time steps.

(2) The value of 'DT' defines the time step increment. The programme has been written based on constant time steps. However, this may sometimes be disadvantageous when an analysis, 'NDYN. EQ.5 (say, consolidation problem), is carried out because of the large numbers of 'NCYCL' steps.

(3) 'KKS' is one of the key parameters which controls the type of analysis to be performed based on a particular material model as selected.

If KKS. EQ.1 a linear elastic analysis will be performed ie. material properties will remain entirely constant during the analysis.

If KKS. EQ.2, an equivalent linear analysis following the procedure as described by Seed and Idriss (1969) will be carried out. This procedure takes into account the variable damping and shear modulus for each element depending on its strain level.

IV MASTER CONTROL CARD - CONT'D

NOTES/ - CONT'D

If KKS. EQ.3, an incremental nonlinear (elastic-plastic) analysis is performed based on either one of the yield models as selected. The two types of yield models which are at present available are: 1) Von Mises yield criteria with or without isotropic hardening and 2) Drucker-Prager yield criteria with zero hardening (elastic - perfectly plastic).

If KKS. EQ.4, a shakedown analysis is performed using piecewise linearized yield model ie. modelling the quadratic yield function with a suitable number of yield planes. The linear programming technique is then used to evaluate the shakedown load factor.

(4) Input the gravitational constant value for computation of the mass matrix;

(5) The parameter 'NDRN' controls the state of the soil masses under the loading conditions.

If NDRN. EQ.0, this will represent a purely undrained situation i.e. no movement of fluid with respect to solid and as such all degrees of freedom related to fluid are fixed.

VV SELECTIVE INPUT DATA (4E10.3)

Notes	Columns	Variable	Description of Input
(1)	1-10	AREA	GT.0.0; Area of the foundation (Equivalent model of a circular foundation)

V SELECTIVE INPUT DATA (4E10.3) - CONT'D

Notes	Columns	Variable	Description of Input
(2)	11-20	WIDTH	GT.0.0; Width of the footing used in plane strain modelling
(3)	21-20	WEIGHT	GT.0.0; Submerged weight of the platform
(4)	31-40	SHEAR	GT.0.0; Undrained shear strength of the soil beneath the foundation

NOTES/

(1) In this section the equivalent area of the foundation is input.

(2) Width of the eqivalent plane strain footing is input here.

(3) Submerged or buoyant weight of the platform is input here.

(4) Undrained soil shear strength beneath the foundation; As soil shear strength will vary over the depth, the shear strength for the top layer is normally input here.

VI A CONTROL CARD FOR DISPLACEMENT OUTPUT (5X, I5)

Notes	Columns	Variable	Description of Input
(1)	5-10	*JPT	GE.0; Number of nodal points for which plotting is required

VIII NODAL POINT DATA (515, 2F10.0, I5)

Notes	Columns	Variable	Description of Input
(2)	6-10	ID (N,1)	X-translation boundary condition code for solid
	10-15	ID (N,2)	Y-translation boundary condition code for solid
	16-20	ID (N,3)	X-translation boundary condition code for fluid
	21-25	ID (N,4)	Y-translation boundary condition code for fluid
(3)	26-35	R (N)	X - coordinate
(4)	36-45	Y (N)	Y - coordinate
(5)	46-50	KM	Node number increment

NOTES/

(1) Nodal point data must be defined for all (NUMNP) nodes. Nodal point data may be input directly (ie. each node on its own individual card) or the generatin option may be used if applicable (see note 5, below). Nodal point numbering should range from "1" to the total number of nodes 'NUMNP' sequentially.

(2) Boundary condition codes can only have the following sets of assigned values describing the movement of the solid and fluid.

(M = 1, 2, 3 and 4) ID(N,M) = 0; unspecified (free) displacement conponent

VIII NODAL POINT DATA - CONT'D

NOTES/ - CONT'D

ID(N,M) = 1/-1; Deleted displacement component

(3) & (4) Two dimensional coordinate of a node; X or Y in cartesian system.

(5) Nodal point generation parameter; always given on the second card of a sequence.

IX MATERIAL CONTROL CARD (3I5)

Notes	Columns	Variable	Description of Input
(1)	1-5	NUMMAT	GE.1; Number of material property set
(2)	6-10	NELYTP	EQ.0; 4-noded isoparametric element
			EQ.1; 4-noded isoparametric element with incompatible modes
(3)	11-15	NS	Stress output option;
			EQ.0; Centroid and gauss points
			EQ.20; Midside

X MATERIAL PROPERTIES CARD (8E10.3)

For each material identification, two cards must be supplied in order to define each unique set of properties.

X MATERIAL PROPERTIES CARD (8E10.3) - CONT'D

Notes	Columns	Variable	Description of Input
	1-10	ELMDA(N)	Lame's constant - λ (drained)
	11-20	EMD(N)	Lames's constant - μ (drained)
	21-30	EALFA(N)	Solid grain compressibility - α
	31-40	EM(N)	Water bulk modulus - K_w
	41-50	EK(N)	Permeability - k
	51-60	DENB(N)	Bulk density - ρ
	61-70	DENF(N)	Water density - ρ_f
	71-80	FK(N)	Porosity - n

Carry over to next card;

Notes	Columns	Variable	Description of Input
	1-10	DBT1(N)	Rayleigh damping coefficients - B_1
	11-20	DBT2(N)	- do - B_2

NOTES/

XI ELEMENT CARDS (5I3, 5I2, 5E10.3, F5.2)

Notes	Columns	Variable	Description of Input	
(1)	1-3	INEL	Element number	
(2)	4-6	I	Number of nodal point I	
	7-9	J	Number of nodal point J	
	10-12	K	Number of nodal point K	
	13-15	L	Number of nodal point L	
(3)	16-17	IMAT	GE.1;	Material identification number; (i.e. NUMMAT)
(4)	18-19	KN	GE.0;	Element generation code;
(5)	20-21	KJUMP	EQ.0;	
(6)	22-23	IKS	EQ.0;	Free movement of solid
			EQ.1;	Fixed movement
(7)	24-25	IKF	EQ.0;	Free movement of fluid
			EQ.1;	Fixed movement
(9)	76-80	THICK	GE.0;	Element thickness

XII MATERIAL PROPERTY CARD - ELASTIC-PLASTIC ANALYSIS (I5, 2E12.5)

*** THIS INPUT IS REQUIRED IN THE EVENT KKS = 3; OTHERWISE SKIP THE FOLLOWING SECTION. ****

Notes	Columns	Variable	Description of Input
(1)	1-5	N	LE. NUMMAT; Material identification number
(2)	6-18	YIELD(N)/ COH(N)	For MODEL = 1; Read 'YIELD(N)' For MODEL = 2; Read 'COH(N)'
(3)	19-31	HARD(N)/ PHI(N)	For MODEL = 1; Read 'HARD(N)' as hardening parameter For MODEL = 2; Read 'PHI(N)' as angle of internal friction

NOTES/

(2) & (3) Incremental elastic-plastic analysis can be carried out based on two types of material modelling. These are defined as; MODEL = 1; Von Mises yield criteria and MODEL = 2; Drucker-Prager yield criteria. Von Mises material model requires the yield strength and hardening modulus as input parameters. Hardening modulus, E_p, can be expressed as some percentage of the original linear elastic modulus. The actual value can be determined form an uniaxial stress/strain curve. However, for the Drucker-Prager type material model, the analysis can be carried out assuming the material behaves as an elastic-perfectly plastic manner. In this case, the value of E_p will be zero (no hardening) and the only material properties to be input are c' = Cohesion' and 'ϕ_u = angle of internal friction'.

XIII CONTROL CARD FOR NUMBERICAL INTEGRATION

Notes	Columns	Variable	Description of Input
(1)	1-10	TETA	For static analysis = 1.0; However, for dynamic analysis $\theta = 1.0$; constant acceleration method or $\theta = 1.4$; WILSON - θ method
	11-20	BETA	= 0.0;
	21-30	GAMA	= 0.0;
(2)	31-40	RTOL	Tolerance value; Required if KKS = 3; otherwise leave blank
(3)	41-45	NPUNCH	EQ.0; Output do not require saving EQ.1; Output to be saved for subsequent plotting
(4)	46-50	NEQUIB	Normally set equal to zero; However, for KKS = 3; NEQUIB can be set equal to 1
(5)	51-55	ITEMX	Number of iterations specified within each time or load step;

XIII CONTROL CARD FOR NUMBERICAL INTEGRATION - CONT'D

NOTES/

(1) For static analysis 'Θ' is normally chosen as 1.0, ie. equal time step or load step is followed. However, for dynamic analysis 'Θ' could be anywhere between 1.0 to 1.4. For Θ = 1.0, this represents constant acceleration method whereas 'Θ' = 1.4, represents 'WILSON- Θ' method. In the 'WILSON - Θ' method load vector and stiffness matrices are constructed at an extended time step ie. at, $t = t+\tau$, where $\tau = \Theta\Delta t$, and equilibrium equations are solved at '$t+\tau$' time step. The displacement, velocity, and acceleration at time '$t+\Delta t$' are then computed from the corresponding values at time '$t+\Theta\Delta t$'. For all dynamic analysis, time step 'Δt' as defined in sectin IV should be sufficiently small in order to get a stable, meaningful result.

(2) Torerance value required to carry out the incremental elastic-plastic analysis (KKS = 3). If NEQUIB. EQ.0, RTOL is set equal to 0.0. However, when NEQUIB. EQ.1, RTOL is set equal to some small percentage of the load increment. Care and judgement should be exercised in order to avoid poor convergence of the results. The load increment or time increment should be also sufficiently small in order to avoid the divergence of the residual load distribution with respect to the actual incremental load that has been applied on the system.

XIV CONTROL CARD FOR LOAD CASES, LINEARIZED YIELD PLANES AND ITERATION NUMBER

Notes	Columns	Variable	Description of Input
(1)	1-5	NLC	GE.1; Number of load cases
(2)	6-10	NP3	EQ.3; Number of stress components. Required only when KKS = 4; otherwise set as '1'

XIV CONTROL CARD FOR LOAD CASES, LINEARIZED YIELD PLANES AND ITERATION NUMBER - CONT'D

Notes	Columns	Variable	Description of Input
(3)	11-15	NP6	Number of yield planes; varies from 6 to 14. Plane strain model NP6 = 6 whereas plane stress model NP6 = 14; Required only when KKS = 4; otherwise set equal to '1'
(4)	16-20	NUMBER	Total number of iterations required in order to perform an equivalent linear analysis and to obtain strain compatible dynamic soil properties; Required in the event KKS = 2; otherwise set "NUMBER' as 1

XV MATERIAL PROPERTY CARD FOR EQ. LINEAR ANALYSIS (5X, 4E12.5)

*** IF (KKS. NE.2) SKIP THIS SECTION ***

One card is required for each unique set of properties.

Notes	Columns	Variable	Description of Input
(1)	6-18	GMX(N)	Maximum shear modulus evaluated at small strain
(2)	19-31	DPL(N)	Maximum critical damping ratio at small strain
(3)	32-43	GUSE(N)	Shear modulus used in the first iteration

XV MATERIAL PROPERTY CARD FOR EQ. LINEAR ANALYSIS (5X, 4E12.5) - CONT'D

Notes	Columns	Variable	Description of Input
(4)	44-56	PO(N)	Poisson's ratio

NOTES/

(1) Shear modulus at 10^{-4}% strain, see Chapter 5.

(2) Critical damping ratio at small strain (refer Chapter 5 and Table 5.1).

(3) Value of shear modulus used in the first iteration.

XVI CONTROL CARD TO IDENTIFY FOOTING/STRUCTURE ELEMENTS

** IF (KKS. NE.2) SKIP THIS SECTION**

Notes	Columns	Variable	Description of Input
(1)	1-5	NOEL	GE.0; Number of footing elements
(2)	6-80	(NL(I), I = 1,NOEL)	Elements numbered sequentially

NOTES/

(1) Total number of footing or structural elements in the model

(2) Element numbers entered sequentially in order to model the foundation/structure.

XVII CONTROL CARD FOR EIGEN VALUE ANALYSIS (3I5, 2F10.0, I5)

** IF (KKS EQ.2 or NDYN. EQ.3). THIS SECTION IS REQUIRED TO BE INPUT, OTHERWISE SKIP **

Notes	Columns	Variable	Description of Input
(1)	1-5	IFPR	Flag for printing intermediate matrices, norms, etc; calculated during the eigenvalue solution
			EQ.0; Do not print
			EQ.1; Print
(2)	6-10	IFSS	Flag for performing the STRUM SEQUENCE check
			EQ.0; Check to see if eigenvalues were missed
			EQ.1; Pass on the check
(3)	11-15	NITEM	Maximum number of iteration allowed to reach the convergence tolerance;
			EQ.0; default set to '16'
(4)	16-25	RTOL	Convergence tolerance (accuracy) for the highest (NF) request eigenvalue;

XVII CONTROL CARD FOR EIGEN VALUE ANALYSIS (3I5, 2F10.0, I5) - CONT'D

Notes	Columns	Variable	Description of Input
(5)	26-35	COFQ	Cut off frequency (cycles/unit time);
			EQ.0; NF eigenvalues will be extracted
			GT.0; Extract only those values below COFQ
(6)	36-40	NFO	Number of starting iteration vectors to be read from tape 10

XVIII CONTROL CARD FOR SHAKEDOWN ANALYSIS (4I5)

** IF (KKS. NE.4) SKIP THIS SECTION **

Notes	Columns	Variable	Description of Input
(1)	1-5	IPT	EQ.1; Minimize the objective function
(2)	16-20	NPRINT	EQ.0; No intermediate print is required
			EQ.1; Intermediate prints are required

XIX CONTROL CARD FOR MATERIAL PROPERTY; MATERIAL TYPE FOR LINEARIZER MODEL, (I5)

** If one card is required for each set of unique property**

XIX CONTROL CARD FOR MATERIAL PROPERTY; MATERIAL TYPE FOR LINEARIZER MODEL, (I5) - CONT'D

Notes	Columns	Variable	Description of Input
(1)	1-5	ITYPE(N)	EQ.1; None for KKS = 1, 2 & 3
			EQ.0; For KKS = 4 and IPLNAX = 0; Plane strain model

NOTES/

(1) 'ITYPE(N)' is a parameter which controls the type of piecewise linearized yield model to be used in the shakedown analysis.

For plane stress condition ie. IPLNAX = -1, only Von Mises model is available. This model is described by fourteen yield planes.

However, for plane strain condition ie. IPLNAX = 0, three types of piecewise linearized yield model are available at present. These are a) Von Mises, b) Tresca and c) Mohr-Coulomb.

XX CONTROL CARD - MATERIAL PROPERTIES (2E10.3)

Notes	Columns	Variable	Description of Input
(1)	1-10	COH(N)	Cohesion value
(2)	11-20	PHI(N)	Angle of internal friction value

XXI CONTROL CARD FOR SURFACE LOAD DATA

Notes	Columns	Variable	Description of Input
(1)	1-5	ISLC	EQ.0; No concentrated loading is prescribed NE.0; Required concentrated loading as input
(2)	6-80	(NSLCI(K), K = 1, NLC)	Number of surface load cards NSLCI(K) for each 'K' load case. (where K. GE.1 but LE.NLC)

NOTES/

(1) Input data which governs whether a concentrated loading has to be prescribed or not.

(2) Number of surface loading data cards for each load case; 'NSLC' = NSLC(K), K = 1, NLC, where 'NSLC' represents number of surface loading cards for a particular load case and 'NLC' indicates total number of load cases.

XXII CONTROL CARD FOR SCALE FACTORS WITH SURFACE LOADIANG-BULK (8E10.3)

Notes	Columns	Variable	Description of Input
(1)	1-80	(SCALB(I), I = 1, NSLC)	Scale factor for each surface loading card; NSLC represents the number of surface loading cards for one load case.

XXII CONTROL CARD FOR SCALE FACTORS WITH SURFACE LOADIANG-BULK (8E10.3) - CONT'D

NOTES/

(1) The surface pressures for bulk loading (as input in section XXV on page 358) are amplified by the scale factor 'SCALB(I)' for each surface loading data card.

XXIII CONTROL CARD FOR SCALE FACTORS WITH SURFACE LOADING - FLUID (8E10.3)

** IF (NDRN. EQ.0) SKIP THIS SECTION BELOW. **

Notes	Columns	Variable	Description of Input
(1)	1-8	(SCALB(I), I = 1, NSLC)	Description of input scale factor for fluid loading for each surface loading card; 'NSLC' represents the number of surface loading cards for one load case

NOTES/

(1) The surface intensity pressures for fluid loading as input in section XXV on page 358are amplified by this scale factor 'SCALB(I)' for each surface loading data card.

XXIV CONTROL CARD FOR SURFACE LOADING DATA INPUT

A) TIME POINTS (8F10.0)

Notes	Columns	Variable	Description of Input
(1)	1-10	ID(N)	First time point of the data entry; Read the data as a pair of two data sets; TD(1), & TD(2), etc.

B) SURFACE LOAD DATA (2I4, 4E11.4) - BULK LOAD

Notes	Columns	Variable	Description of Input
(1)	1-4	ISC(L)	Node indicating 'I'
(2)	5-8	JSC(L)	Node indicating 'J'
(3)	9-19	SURTRX (N,L,1)	X-direction pressure intensity on node 'I'
(4)	20-30	SURTRX (N,L,2)	X-direction pressure intensity on node 'J'
(5)	31-41	SURTRY (N,L,1)	Y-direction pressure intensity on node 'I'
(6)	42-52	SURTRY (N,L,2)	Y-direction pressure intensity on node 'J'

C) SURFACE LOAD DATA FOR FLUID PRESSURE (2I4, 4E11.4)

** IF (NDRN. EQ.0) SKIP THIS SECTION BELOW. **

Notes	Columns	Variable	Description of Input
	1-4	ISC(L)	Node indicating 'I'

XXIV CONTROL CARD FOR SURFACE LOADING DATA INPUT - CONT'D

C) SURFACE LOAD DATA FOR FUILD PRESSURE - CONT'D

Notes	Columns	Variable	Description of Input
	5-8	JSC(L)	Node indicating 'J'
	9-19	SURPFX (N,L,1)	X-direction surface pressure intensity on node 'I'
	20-30	SURPFX (N,L,2)	X-direction surface pressure intensity on node 'J'
	31-41	SURPFY (N,L,1)	Y-direction surface pressure intensity on node 'I'
	42-52	SURPFY (N,L,2)	Y-direction surface pressure intensity on node 'J'

NOTES/

XXIV (A) (1) Time points for the loading data; Input TD(1) first which represents the starting time point.

(B) (1) Input the associated nodal points, I, on which the surface pressure is prescribed.

(2) Input the associated nodal points, J, on which the surface pressure is prescribed.

(3),(4) (5),(6) Input the surface pressure intensities for bulk load in X and Y directions.

XXIV CONTROL CARD FOR SURFACE LOADING DATA INPUT - CONT'D

NOTES/ - CONT'D

XXIV (C) (3),(4), (5),(6)	Input the surface pressure intensities for fluid loading in X and Y directions. If NDRN.EQ.0 (undrained/drained case), skip this section entirely.

* All surface loading data cards must be input for as many as 'NSLC' times where 'NSLC' represents the number of surface loading cards for a particular load case.

XXV CONTROL CARDS FOR CONCENTRATED LOADING DATA

* THESE CARDS ARE REQUIRED IN THE EVENT ISLC = 0 (SEE SECTION XXII ON PAGE 355)

A) HEADING CARD IDENTIFYING THE LOADING FUNCTION - (10A8)

Notes	Columns	Variable	Description of Input
(1)	1-80	HED(8)	Title of the loading function

B) INPUT DATA DEFINING LOADED POINTS (I5)

Notes	Columns	Variable	Description of Input
(1)	1-5	NUMLP	Total number of loaded points

NOTES/

XV A (1)	Heading card indicating the title of the loading function.
XV B (1)	Total number of nodal points where the concentrated loads are prescribed.

XXVI CONTROL CARDS FOR THE HISTORIES OF LOADING FUNCTION (CONCENTRATED)

* THESE FOLLOWING SETS OF CARDS i.e. CARD GROUPS (A) & (B) SHOULD COVER THE TOTAL DURATION OF TIME (NCYCL*DT) AS DEFINED IN SECTION IV ON PAGE 336;-

XXVI A) TIME POINTS (8F10.0)

Notes	Columns	Variable ,	Description of Input
(1)	1-10	TD(N)	Time points for the loading data ie. TD(1), TC(2)

XXVI B) LOADING DATA (I4, 4E10.3)

Notes	Columns	Variable	Description of Input
(1)	1-5	M	Nodal point where the load is prescribed
(2)	6-15	PDYL(N,1,M)	Bulk loading in X - direction
(3)	16-25	PDYL(N,2,M)	Bulk loading in Y - direction
(4)	26-35	PDYL(N,3,M)	Fluid loading in X - direction
(5)	36-45	PDYL(N,4,M)	Fluid loading in Y - direction

NOTES/

XXVI (A) (1) Time points for the loading data; Input TD(1) first which represents the starting time point.

XXVI CONTROL CARDS FOR THE HISTORIES OF LOADING FUNCTION (CONCENTRATED) - CONT'D

NOTES/ - CONT'D

XXVI (B) (1) Input the associated nodal point, sequentially for which the loads are to be prescribed.

(B) (2) Input the bulk solid and fluid loads

(3) in the X and Y directions for a particluar node; Next repeat the

(4) loading data for as many nodes up to

(5) 'NUMLP' as described in section XXV (B) on page 358. Once the input for loading data is finished, repeat XXVI (A) for time sequences ie. TD(2) and continue the data input until the total time duration (NCYCL*DT) is covered.

FINITE ELEMENT DYNAMIC SHAKEDOWN ANALYSIS OF A GRAVITY TYPE OFFSHORE STRUCTURE-FOUNDATION SYSTEM

VOLUME II

A. K. HALDAR

FINITE ELEMENT DYNAMIC SHAKEDOWN ANALYSIS OF A GRAVITY TYPE OFFSHORE STRUCTURE-FOUNDATION SYSTEM

by

(C) A. K. Haldar, B.E., M.Eng, P.Eng.

A thesis submitted to the School
of Graduate Studies in partial
fulfillment of the requirements
for the degree of Doctor
of Philosophy

VOLUME - II

Faculty of Engineering
Memorial University of Newfoundland
December 1984

St. John's Newfoundland

APPENDIX - E

LISTING OF COMPUTER PROGRAM 'OPFA'

TABLE OF CONTENTS

```
          DATA SET ACOMP      AT LEVEL 039 AS OF 83/01/17
      SUBROUTINE ACOMP(S,C,CT,NEL,NPJ,NPK,AN,NPT)                       00001*33
      IMPLICIT  REAL*8 (A-H,O-Z)                                        00002
                                                                        00003**4
                                                                        00004**4
                                                                        00005**4
      DIMENSION AN(NPJ,1),S(NPJ,1),C(NPJ,1)                             00006*26
      DIMENSION SK(8,8),E(8,8),CC(8,8),GMS(8)                           00007**4
      DIMENSION P(4,8),D(4,4),LM(8),LH(8),SF(8,8)                       00008*24
      DIMENSION H(8,8),NP(4)                                            00009*13
                                                                        00010**9
      DIMENSION XP(16),XM(16),PK(4,8),PN(4,8)                           00011*32
                                                                        00012*32
                                                                        00013**9
                                                                        00014*33
                                                                        00015*33
      DIMENSION CT(NPJ,1)                                               00016*33
                                                                        00017*33
      DIMENSION ST(20,12),T1(5,12),LK(16),PS(16,8),RM(5),ZM(5)          00018*34

      COMMON/BENDY/SG(5,200,4,4),NCOND
                                                                        00019*33
      COMMON/CONTR/DD,NED(8),KADN,NUMNP,NUMSL,NEQ,MBAND,NCYCL,NDYN,IPLNA00020**2
     +X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH ,NEQS,MBANDS                   00021**2
      COMMON/BOBBY/KKS,NUMGT,KC,LOOP,NDRN,NINT,IDWA,NGT                 00022*34
      COMMON/ASIM/MYP,NSC,NSLC,NLC,NNYP,NNSC                            00023*37
                                                                        00024*37
                                                                        00025*35
      DIMENSION FACN(4)                                                 00026*35
                                                                        00027*35
                                                                        00028*35
******INITIALIZATION                                                    00029
                                                                        00030**4
       NDI=1
       IF(NCOND.EQ.1) NDI=4
                                                                        00031**4
                                                                        00032**4
  999 FORMAT(///)                                                       00033
      LQ=NEL*3                                                          00034
      MQ=LQ                                                             00035
                                                                        00036*24
                                                                        00037*24
      DO  700  I=1,NPJ                                                  00038*24
      DO  700  J=1,NPJ                                                  00039*24
  700 S(I,J)=0.0                                                        00040*24
                                                                        00041*24
                                                                        00042*24
      DO  800 I=1,NPJ                                                   00043*24
      DO  820  J=1,NEQS                                                 00044*24
      C(I,J)=0.0                                                        00045*34
      CT(I,J)=0.0                                                       00046*34
  820 CONTINUE                                                          00047*34
                                                                        00048*34
  800 CONTINUE                                                          00049*24
                                                                        00050*24
                                                                        00051*24
      IF(KKS.EQ.4)  GO  TO  990                                         00052*19
                                                                        00053*26
                                                                        00054*26
      DO  970  MJ=1,NPJ                                                 00055*27
```

```
      DO  980  I =1,NPE                                                        00056*19
  980 AG(NJ,I)=0.0                                                             00057*26
  970 CONTINUE                                                                 00058*26
                                                                               00059*26
                                                                               00060*26
  990 CONTINUE                                                                 00061*19
*****************************************************************************00062
                                                                               00063
      JK=3                                                                     00064
      REWIND  50                                                               00065
      REWIND  51                                                               00066
      REWIND  12                                                               00067*16
                                                                               00068*39
        REWIND 17                                                              00069*39
                                                                               00070*18
      REWIND  11                                                               00071*34
                                                                               00072*18
                                                                               00073*18
      *********   TAPE   12  PASSED  FROM  YLDEX    ****************           00074*18
                                                                               00075*18
                                                                               00076*18
                                                                               00077*18
      L=0                                                                      00078
      DO  100  N=1,NPE                                                         00079
         IF(IDEF.NE.0) READ(17) ((D(I,J),J=1,4),I=1,4)                         00080*39
         IF(IDEF.EQ.0) READ(50) ((D(I,J),J=1,4),I=1,4)                         00081*39
                                                                               00082*27
      READ(51)(DM(I),I=1,8),(DP(I),I=1,8),((SK(I,J),J=1,8),I=1,8)              00083*12
     *,((PF(I,J),J=1,8),I=1,4),(GMS(I),I=1,8),VOL,DENS,DENF,F,NP               00084*11
                                                                               00085*21
                                                                               00086*21
      READ(51)((SF(I,J),J=1,8),I=1,8),((R(I,J),J=1,8),I=1,8),((CC(I,J),        00087*16
     *J=1,8),I=1,8),IMAT,KS,FF,ALF,((H(I,J),J=1,8),I=1,8)                      00088*14
     *,(XF(I),I=1,16),(XH(I),I=1,16)                                           00089*32
                                                                               00090*34
      READ(11) ST,TT,PH,GR,ALFA,EM,PM,ZM,PS,RP,IMAT,MER,EMU,YOUNG,POIS,         00091*34
     *VOL,IS,FACT                                                              00092*34
                                                                               00093*34
                                                                               00094*34
 1200  CONTINUE                                                                00095
                                                                               00096

                                                                               00097*34
                                                                               00098*34
      DO  150  II=1,NNI                                                        00099*34
      IF(NCODE.NE.1) GO TO 1300

      CALL COMPUT(PS,P,FACP,IF)                                                00101*34
         WRITE(6,1305)
         WRITE(6,1306) ((P(I,J),J=1,8),I=1,4)
         WRITE(6,1306) FACP(IF)

         WRITE(6,1309) ((D(I,J),J=1,4),I=1,4)

 1309   FORMAT(5X,4E10.3)
                                                                               00100*34

 1300 CONTINUE
```

```
      PRINT OF STRAIN DISPL MATRIX


       WRITE(6,1305)
1305   FORMAT(1H1/5X,'   STRAIN DISPL MATRIX '/)
        WRITE(6,1306) ((PK(I,J),J=1,8),I=1,4)
1306   FORMAT(5X,8F10.3)

       WRITE(6,1309) ((D(I,J),J=1,4),I=1,4)

       WRITE(6,1306) VOL

     DO  200  I=1,4                                              00102
*********CODING  TO  DELETE  FOR  THIRD  ROWS  COMPLETELY        00103
                                                                 00104
     II=I                                                        00105
      IF(I.GT.2) II=I+1                                          00106
     IF(II.GT.4)  GO  TO  200                                    00107
                                                                 00108
                                                                 00109
     L=L+1                                                       00110
      DO  300  J=1,8                                             00111
                                                                 00112*22
     K=NM(J)                                                     00113
     IF(K)  90,90,91                                             00114
                                                                 00115
                                                                 00116
  91 CONTINUE                                                    00117*28
                                                                 00118*33

     IF(NCOND.NE.1) GO TO 1400

     C(L,K)=P(II,J)                                              00119*34
                                                                 00120*34
                                                                 00121*34

     CT(L,K)=P(II,J)*FACN(IP)                                    00122*33
                                                                 00123*34
     GO TO 90

1400 C(L,K)=PK(II,J)
                                                                 00124*33
     CT(L,K)=PK(II,J)*VOL
  90 CONTINUE                                                    00125
 300 CONTINUE                                                    00126
                                                                 00127
                                                                 00128
********STRESS - STRAIN  LAW  ASSEMBLING  FOR  ALL  ELEMENTS ******  00129
                                                                 00130
     DO  400  JN=1,4                                             00131
                                                                 00132*22
     JM=JN                                                       00133
     IF(JN.EQ.3)  GO  TO  400                                    00134
     IF(JN.EQ.4)  JM=JN-1                                        00135
                                                                 00136*34
     IF(NCOND.NE.1) GO TO 1500


                                                                 00137*34
     JJ=JM+(IP-1)*NSC+(N-1)*NSC*NPI                              00138*34
```

```
      GO TO 1600                                                        00139*34
1500 JJ=JM+(N-1)*NSC*NDT
                                                                        00140*34
1314  FORMAT(5X,7I5)
1600  S(L,JJ)=D(II,JM)                                                  00141*25
       WRITE(6,1314) JJ,JM,IE,NSC,NCOND,NDT,N
                                                                        00142*34
 400 CONTINUE                                                           00143
 200 CONTINUE                                                           00144
                                                                        00145
                                                                        00146*34
 150 CONTINUE                                                           00147*34
*************ELEMENT    LOOP     ********                               00148
 100 CONTINUE                                                           00149
                                                                        00150
                                                                        00151
        WRITE(6,1307)
1307   FORMAT(1H1/5X,' GLOBAL STRAIN DISPL MATRIX '/)
                                                                        00152*18
         WRITE(6,1310) ((C(I,J),J=1,NEQS),I=1,NPJ)
        WRITE(6,1307)
         WRITE(6,1310) ((CT(I,J),J=1,NEQS),I=1,NPJ)
        WRITE(6,1307)
         WRITE(6,1311) ((S(I,J),J=1,NPJ),I=1,NPJ)
        WRITE(6,1307)
         WRITE(6,1312) ((AN(I,J),J=1,NPK),I=1,NPJ)
1310    FORMAT(5X,4E10.3)
1311    FORMAT(5X,12E10.3)
1312    FORMAT(5X,24F5.2)                                               00153*18
                                                                        00154*24
                                                                        00155*24
                                                                        00156*18
                                                                        00157*30
      RETURN                                                            00158
      END                                                               00159
```

```
C           PROGRAM  AMAIN
            IMPLICIT REAL*8(A-H,O-Z)
            READ(5,100) KKS
  100       FORMAT(5X,I5)
CC
            IF(KKS.NE.2) GO TO 200
            NCOM=200000
            GO TO 300
  200       NCOM=400000
  300       CONTINUE
C
            CALL MAIN(NCOM)
            STOP
            END
```

```
         DATA SET ASSEMB      AT LEVEL 009 AS OF 83/02/02
       SUBROUTINE ASSEMB(RE,NUMEL,NEQ)                                  00001**6
                                                                        00002**2
    IMPLICIT REAL *8 (A-H,O-Z)                                          00003**2
    DIMENSION RE(1)                                                     00004**2
                                                                        00005**2
    DIMENSION LM(16),XP(16)                                             00006**4
                                                                        00007**4
       DO 100 I=1,NEQ                                                   00008**6
100    RE(I)=0.0                                                        00009**6
                                                                        00010**8
                                                                        00011**3
                                                                        00012**2
    ****  REWIND  TAPE  AND  READ  INFORMATION  **********              00013**2
                                                                        00014**2
       REWIND 21                                                        00015**7
    DO  900 N=1,NUMEL                                                   00016**2
                                                                        00017**2
     READ(21) LM,XP                                                     00018**5

       WRITE(6,160)
160    FORMAT(5X,' *** PRINT IN ASSEMB ***',/)
       WRITE(6,161) (XP(I),I=1,16)
161    FORMAT(5X,8E10.3)
       WRITE(6,162) (LM(I),I=1,16)
162    FORMAT(5X,16I5)
                                                                        00019**5
                                                                        00020**8
                                                                        00021**8
                                                                        00022**2
    DO  360 I=1,16                                                      00023**2
    L=LM(I)                                                             00024**2
400 IF(L)360,360,200                                                    00025**2
200 RE(L)=RE(L)+XP(I)                                                   00026**2
360 CONTINUE                                                            00027**2
900 CONTINUE                                                            00028**2
    RETURN                                                              00029
    END                                                                 00030
```

```
           DATA SET BANDET      AT LEVEL 002 AS OF 81/09/05
       SUBROUTINE BANDET (A,B,V,MAXA,NN,NWA,RA,NSCH,DET,ISCALE,KK)
      IMPLICIT REAL*8(A-H,O-Z)

      CALLED BY:  SECNTD

       COMMON /TAPES/NSTIF,NRED,NL,NR,NT,NMASS
       DIMENSION A(NWA),B(1),V(1),MAXA(1)

       NR=NN-1
       IF (KK-2) 100,700,800

100    TOL=1.0E+04
       RTOL=1.0E-07
 ***        SCALE=2.0D0**200
SCALE = 1.700D+38
       NTF=3
       IS=1
120    REWIND NSTIF
       READ (NSTIF) A
       DO 140 I=1,NN
140    A(I)=A(I)-RA*B(I)
160    IF (NWA.EQ.NN) GO TO 230
       DO 200 N=1,NR
       IH=N+NWA-NN
210    IF (A(IH)) 220,215,220
215    IH=IH-NN
       GO TO 210
220    MAXA(N)=IH
       PIV=A(N)
      IF(PIV) 221,500,221
 500  IS = IS+1
      IF(IS.LE.NTF) GO TO 502
 501  WRITE (6,1000) NTF,RA
      STOP
 502  RA = RA*(1.0-RTOL)
      GO TO 120
221    IL=N+NN
       L=N
       DO 240 I=IL,IH,NN
       L=L+1
       C=A(I)
       IF (C) 225,240,225
225    C=C/PIV
      IF (DABS(C) .LT. TOL) GO TO 235
226    IS=IS+1
       IF (IS.LE.NTF) GO TO 245
      GO TO 501
245    RA=RA*(1.0-RTOL)
       GO TO 120
235    J=L-I
       DO 260 K=1,IH,NN
260    A(K+J)=A(K+J)-C*A(K)
       A(I)=C
240    CONTINUE
200    CONTINUE
230    IF (A(NN).NE.0.0) GO TO 280
      AA=DABS(A(1))
       DO 290 I=2,NR
290   AA=AA+DABS(A(I))
```

```
      A(NN)=-(AA/ER)*1.0E-16

280   NSCH=0
      ISC=0
      DET=1.0
      DO 300 I=1,NN
     IF (DABS(DET) .GT. SCALE) GO TO 320
      DET=DET/SCALE
      ISC=ISC+1
320   DET=DET*A(I)
300   IF (A(I).LT.0.) NSCH=NSCH+1

      IF (ISCALE.LT.1000) GO TO 340
      ISCALE=ISC
      GO TO 900
340   IF (ISC-ISCALE) 350,900,370
350   DET=DET/SCALE
      GO TO 900
370   DET=DET*SCALE
      GO TO 900

700   IL=NN
      DO 400 N=1,NR
      C=V(N)
      V(N)=C/A(N)
      IF (NWA-NN) 410,400,410
410   IL=IL+1
      IH=MAXA(N)
      K=N
      DO 420 I=IL,IH,NN
      K=K+1
420   V(K)=V(K)-C*A(I)
400   CONTINUE
      V(NN)=V(NN)/A(NN)

800   IF (NWA-NN) 430,900,430
430   N=NN
      DO 440 L=2,NN
      N=N-1
      IL=N+NN
      IH=MAXA(N)
      K=N
      DO 460 I=IL,IH,NN
      K=K+1
460   V(N)=V(N)-A(I)*V(K)
440   CONTINUE
 900 CONTINUE
      RETURN

1000 FORMAT (37H0***ERROR   SOLUTION STOP IN *BANDET*, / 12X,
   1           1H(,I3,37H) TRIANGULAR FACTORIZATIONS ATTEMPTED, / 12X,
   2           10HCURRENT SHIFT = ,E20.14 / 1X)

      END
```

```
          DATA SET CHECK       AT LEVEL 003 AS OF 82/06/14
      SUBROUTINE CHECK(CT,SW,SO,NPJ,NEQS)                                00001
      IMPLICIT REAL *8 (A-H,O-Z)                                         00002
      DIMENSION CT(NPJ,1),SO(1),SW(1)                                    00003
      DO  300 I=1,NEQS                                                   00004
  300 SO(I)=0.0                                                          00005
                                                                         00006
                                                                         00007
      DO  100 I=1,NEQS                                                   00008
      DO  200 J=1,NPJ                                                    00009
  200 SO(I)=SO(I)+CT(J,I)*SW(J)                                          00010
  100 CONTINUE                                                           00011
      WRITE(6,1000)                                                      00012**2
 1000 FORMAT(1H1/'    *****    PRINT   OF   RESIDUAL   FORCE   IN   SYSTEM   ')00013**2
      WRITE(6,400) (SO(I),I=1,NEQS)                                      00014**3
  400 FORMAT(5X,6E12.5)                                                  00015**3
      RETURN                                                             00016
      END                                                                00017
```

```
      DATA SET COMPGT      AT LEVEL 001 AS OF 82/06/12
   SUBROUTINE COMPGT(PS,P,FACN,IP)

   IMPLICIT REAL*8(A-H,O-Z)

   DIMENSION PS(16,8),P(4,8),FACN(4)

   DO  100 J=1,4
   DO  100 I=1,8
   JJ=J+(IP-1)*4
   P(J,I)=PS(JJ,I)
100 CONTINUE
   RETURN
   END
```

```
       DATA SET CONST       AT LEVEL 007 AS OF 83/03/02
      SUBROUTINE   CONST
      IMPLICIT  REAL*8  (A-H,O-Z)
      COMMON/CONTRL/DD,HED(8),KADN,NASIM,NUMSB,NKK,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRFR,NPRINT,ISTART,NEQK,MBANDH
     *,NEQS,MBANDS
       COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX
      COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A08
 999 FORMAT(//)
      DT=TETA*DD

      ********END*******************
      ****ADDED   BY  HALDAR  ****

 100 CONTINUE


      DELT1=TETA*DD
      DELT2=DELT1**2
      C1=1./TETA
      C2=1.-C1
      A0=6./DELT2
      A01=3./DELT1
      A02=2.*A01
      A03=DELT1/2.
      A04=A0/TETA
      A05=-A02/TETA
      A06=1.-3./TETA
      A07=DD/2.
      A08=DD**2./6.

2100  FORMAT(5X,10E12.5)
      RETURN
      END
```

```
        DATA SET CONSV       AT LEVEL 036 AS OF 83/01/15
      SUBROUTINE CONSV(MAT,EALFA,ELMDA,EMU,IPLNAX,D,BLK,YOUNG,POIS,NEL,

     *NG,EM,UDD)

      IMPLICIT REAL * 8 (A-H,O-Z)



*****TSIS  SUBROUTINE  CREATES  ELEMENT  CONSTITUTUVE  MATREX  BASED  0
****PROBLEM  TYPE  . PLANE STRAIN,PLANE STRESS AND AXISYMMETRIC  ETC


      DIMENSION  D(4,4),UDD(4,4)
      DIMENSION EALFA(1),ELMDA(1),EMU(1)
   ********   ADD  ADDITIONAL  DIMENSION  FOR  STRAIN COMPTBLE  AND  E


      COMMON/ASHK/C1,C2,C3,AL,ALF,FACC,FACT
      COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT

      COMMON/  NANCY/  CM(12000)
      DIMENSION EM(1)



      *******    INITIALIZE   '   ***   D  MATRIX   ***  '  ************

      DO  5200  I=1,4
      DO  5200  J=1,4
      D(I,J)=0.0
 5200 CONTINUE

      ******   INITIALIZE  UNDRAINED  CONSTITUTIVE  MATRIX   ***********

      DO  360  I=1,4
      DO  360  J=1,4
  360 UDD(I,J)=0.0



  500 CONTINUE
      K=MAT
      AL=EALFA(K)
      ALF=AL*EALFA(K)
      FACC=EALFA(K)
      FACT=ELMDA(K)


      ******  PLANE  STRESS   **************



  600 POIS=(ELMDA(K))/(2.*(ELMDA(K)+EMU(K)))
      YOUN G=EMU(K)*(2.*(1.+POIS))
      CONST=YOUNG/(1.-POIS**2)




      IF(IPLNAX) 600,700,700
```

```
                                                                        00060**6
800 CONTINUE                                                            00061**6
                                                                        00062*12
      ******* CHECK WHAT HAPPEN IF WE PUT C1, C2, ... DIRECTLY 1        00063*12
      ******* D(I,J)            FOR PLANE STRESSS CONDITION AND         00064*12
                                                                        00065*12
                                                                        00066*12
                                                                        00067**3
                                                                        00068**3
    ************ PLANE STRAIN OR AXISYMMETRIC *************             00069**3
700 CONTINUE                                                            00070**2
                                                                        00071*26
                                                                        00072*26
                                                                        00073*27
                                                                        00074*26
530 CONTINUE                                                            00075*26
      C1=2.0*EMU(K)+FACT                                                00076
      C2=FACT                                                           00077
      C3=EMU(K)                                                         00078
      C4=ELMDA(K)+(2.*EMU(K)/3.)                                        00079**7
      BLK=C4                                                            00080**7
                                                                        00081**6
                                                                        00082**6
900 CONTINUE                                                            00083**2
                                                                        00084**3
                                                                        00085**3
   **********    ELASTIC  ISOTROPIC  CONSTITUTIVE  MATRIX   ***********00086**3
                                                                        00087**3
      D(1,1)= C1                                                        00088
      D(1,2)=C2                                                         00089
      D(1,3)=D(1,2)                                                     00090
      D(3,3)=D(1,1)                                                     00091
      D(2,2)=C1                                                         00092
      D(2,3)=D(1,3)                                                     00093
      D(4,4)=C3                                                         00094
                                                                        00095*27
                                                                        00096*27
540 CONTINUE                                                            00097*27
      DO  520  I=1,4                                                    00098
      DO  520  J=1,4                                                    00099
      D(J,I)=D(I,J)                                                     00100
520 CONTINUE                                                            00101
                                                                        00102*12
                                                                        00103*12
      IF(IPLEAX.NE.-1)  GO  TO  400                                     00104*12
                                                                        00105*12
                                                                        00106*12
      ******   CONDENSE  FOR  ZERO  STRESS  CONDITION  IN  Z  DIRECTION 00107*12
                                                                        00108*12
      D(1,1)=D(1,1)-D(3,1)*D(1,3)/D(3,3)                                00109*12
      D(1,2)=D(1,2)-D(3,2)*D(1,3)/D(3,3)                                00110*12
      D(1,4)=D(1,4)-D(3,4)*D(1,3)/D(3,3)                                00111*12
      D(2,2)=D(2,2)-D(3,2)*D(2,3)/D(3,3)                                00112*12
      D(2,4)=D(2,4)-D(3,4)*D(2,3)/D(3,3)                                00113*12
      D(4,4)=D(4,4)-D(3,4)*D(4,3)/D(3,3)                                00114*12
                                                                        00115*12
      DO  650  I=1,4                                                    00116*12
      DO  680  J=1,4                                                    00117*13
680   D(J,I)=D(I,J)                                                     00118*13
      D(I,3)=0.0                                                        00119*12
```

```
650   D(3,1)=0.0                                                        00120*33
                                                                        00121*33
                                                                        00122*12
                                                                        00123*12
400 CONTINUE                                                            00124*12
                                                                        00125*13
      *********   MODIFY   CONSTITUTIVE  MATRICES    *********************00126*13
                                                                        00127*13
      DN=ALF*EM(K)                                                      00128*13
                                                                        00129*14
      UDD(1,1)=D(1,1)+DN                                                00130*14
      UDD(2,2)=D(2,2)+DN                                                00131*14
      UDD(1,2)=D(1,2)+DN                                                00132*14
      UDD(2,1)=UDD(1,2)                                                 00133*14
      UDD(1,3)=UDD(1,2)                                                 00134*14
      UDD(3,3)=UDD(1,1)                                                 00135*14
      UDD(2,3)=UDD(1,3)                                                 00136*14
      UDD(4,4)=D(4,4)                                                   00137*14
      UDD(3,1)=UDD(1,3)                                                 00138*23
      UDD(3,2)=UDD(2,3)                                                 00139*23
                                                                        00140*14
                                                                        00141*14
                                                                        00142*14
      *********   REMOVE  UNDRAINED  POIS  AND  YOUNG  ******************00143*25
                                                                        00144*25
                                                                        00145*14
                                                                        00146*14
        WRITE(50) ((UDD(I,J),J=1,4),I=1,4)                              00147*34
                                                                        00148*14
                                                                        00149*13
                                                                        00150*13
300 WRITE(17) ((D(I,J),J=1,4),I=1,4)                                    00151*36
                                                                        00152*35
                                                                        00153*34
                                                                        00154*12
                                                                        00155*12
                                                                        00156*12
     ******   CONNECT  HERE  FOR  ELASTO PLASTIC  ANALYSIS  FROM  ELPAL 00157*12
                                                                        00158*12
                                                                        00159*14
350 CONTINUE                                                            00160*14
                                                                        00161*17
                                                                        00162*17
250 CONTINUE                                                            00163*17
                                                                        00164*28
                                                                        00165*28
                                                                        00166*28
      RETURN                                                            00167
      END                                                               00168
```

```
      DATA SET CURV52      AT LEVEL 001 AS OF 81/07/23
      SUBROUTINE CURV52(S,GN,DN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GCLAY(11),DCLAY(11)
      DATA  GCLAY/1.00,0.913,0.761,0.565,0.400,0.261,0.152,0.076,0.037,
     1   0.013,0.004/
      DATA DCLAY/2.50,2.50,2.50,3.50,4.75,6.50,9.25,13.8,20.0,26.0,29.0/
      S=DABS(S)
      S=DLOG10(S)*2.+9.
      K=S
      IF(K.GE.1) GO TO 2
      GN=1.
      DN=DCLAY(1)
      RETURN
    2 IF(K.LT.11) GO TO 3
      GN=GCLAY(11)
      DN=DCLAY(11)
      RETURN
    3 GN=GCLAY(K)+(GCLAY(K+1)-GCLAY(K))*(S-K)
      DN=DCLAY(K)+(DCLAY(K+1)-DCLAY(K))*(S-K)
      RETURN
      END
```

```
      DATA SET DEQITR     AT LEVEL 037 AS OF 83/03/02
  SUBROUTINE DEQITR(X0,NDYN,A0,RF,R,ITR,NITR,ITEMX,RTOL,NC,A4,NEQ,    00001
                                                                      00002
 *MBAND,NWA,NEQB,NBLOCK,M1,MAXA,X2,X1,VEL,AC,B,ID,AMAX,PMAX,AM,EK,AN00003**7
                                                                      00004
 *,AN1,EKJ,EM,NUMNP,NUMEL,RNORM,XM,A5,GB,AT,BT,                       00005*16
                                                                      00006*16
 *A3,LC)                                                              00007*16
                                                                      00008
  IMPLICIT REAL *8(A-H,O-Z)                                           00009
                                                                      00010
                                                                      00011
     COMMON/GORA/IFLAG,IGR                                            00012*22
  COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC,ISLC                         00013
                                                                      00014*14
     COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A00015*14
 *08                                                                  00016*14
     COMMON/CONTL/DD,HED(8),RADN,NUMNG,NUMSL,NGG,MBDN,NCYCL,NDGN,     00017*30
 *IPLMAX,ISTRPH,NPRINT,ISTART,NEQH,MBANDH                             00018*30
 *,NEQS,MBANDS                                                        00019*31
                                                                      00020*31
                                                                      00021*30
     COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                      00022*14
                                                                      00023*14
  DIMENSION X0(1),A0(1),RE(1),R(1),A4(NEQ,1),MAXA(1)                  00024
   DIMENSION A3(NEQ,1)                                                00025*16
                                                                      00026
   DIMENSION AT(1),BT(1),GB(1)                                        00027*11
      COMMON/BOBBY/KKB,NUMMAT,KC,LOOP,NORN,NINT,IDWA,NPT              00028*29
                                                                      00029**7
                                                                      00030
  DIMENSION  X1(1),X2(1),VEL(1),AC(1),B(1),ID(NUMNP,1),AMAX(NUMNP,2,00031
 *2),PMAX(NUMEL,1),AM(1),EK(1),AN(NNSC,1),AN1(NYP,1),EKJ(1),DS(4,3) 00032**4
  DIMENSION XM(1),A5(NEQ,1)                                           00033**5
                                                                      00034*12
                                                                      00035*12
     COMMON/HARVEY/UACC(700),UVEL(700),UDISPL(700)                    00036*35
      COMMON/TAPES/NQQ(6),NUMBER,LCOUNT

                                                                      00037*12
      DO 150 I=1,NEQ                                                  00038*35
     UACC(I)=0.0                                                      00039*34
     UVEL(I)=0.0                                                      00040*34
     UDISPL(I)=0.0                                                    00041*34
 150 CONTINUE                                                         00042*34
                                                                      00043*12
      ************   CONVERTION OF INTEGTATION OF CONSTNTS. ******     00044*14
                                                                      00045*14
      ************      A0 =A0                                        00046*14
                        A01=A01                                       00047*14
                        A02=A2                                        00048*14
                        A03=A5                                        00049*14
                        A04=A6                                        00050*14
                        A05=A7                                        00051*14
                        A06=A8                                        00052*14
                        A07=A9                                        00053*14
                        A08=A10                                       00054*14
```

```
        ************                                                 00055*14
                                                                     00056*14
                                                                     00057*14
                                                                     00058
                                                                     00059**3
                                                                     00064**4
                                                                     00065**4
                                                                     00066**4
                                                                     00067**4
     IF(NC.EQ.0) GO TO 31
                                                                     00068**4
     IF(LCOUNT.NE.ISTRPR) GO TO 32
                                                                     00069**4
                                                                     00070**4
  31   CONTINUE

        WRITE(6,152)
 152   FORMAT(1H1/5X,'    PRINT IN DEQITR MASS,STIFFNESS,DAMPING ')
         WRITE (6,153) (XM(I),I=1,NEQ)
 153    FORMAT(5X,12E10.3)
         WRITE(6,154)  ((A5(I,J),J=1,MBAND),I=1,NEQ)
 154    FORMAT(5X,8E10.3)

      WRITE (6,65) NC,ITR                                            00071*16
      WRITE(6,66) (AU(I),I=1,NEQ)                                    00072*16
  65  FORMAT(1H1/5X,'     PRINT IN DEQITR      '/                    00073*16
    *5X,'    NO. OF TIME STEP                      = ',I5/           00074*16
    *5X,'    NO. OF ITERATION                      = ',I5/           00075*16
    *5X,'    DISPL. ARRAY  AU  BEGINING ITERATION    '/)             00076*16
                                                                     00077*16
  66 FORMAT(5X,8E10.3)                                               00078*16
                                                                     00079*16
                                                                     00080*15
                                                                     00081**4
  32  CONTINUE
                                                                     00082**4
                                                                     00083**4
                                                                     00084**4
                                                                     00085*14
        DO 50 ITR=2,NITR
                                                                     00086*14
                                                                     00087*14
                                                                     00088*14
        DO 60 I=1,NEQ                                                00089*12
                                                                     00090*35
        IF(NDYN.EQ.4) GO TO 55                                       00091*35
                                                                     00092*35
        UACC(I)=A0*AU(I)-A02*X1(I)-2.*X2(I)                          00093*12
        UVEL(I)=A01*AU(I)-2.*X1(I)-A03*X2(I)                         00094*12
                                                                     00095*35
                                                                     00096*35
  55    CONTINUE                                                     00097*35
        IF(ITR.NE.2) GO TO 60
        UDISPL(I)=X0(I)+AU(I)                                        00098*12
  60    CONTINUE                                                     00099*12
                                                                     00100*12
          IF(NC.EQ.0) GO TO 52
                                                                     00101*12
         IF(LCOUNT.NE.ISTRPR) GO TO 53
                                                                     00102**0
```

```
 52 CONTINUE
    WRITE(6,67)                                                   00103*16
    WRITE(6,66) (UDISPL(I),I=1,NEQ)                               00104*16
    WRITE(6,66) (UVEL  (I),I=1,NEQ)                               00105*16
    WRITE(6,66) (UACC  (I),I=1,NEQ)                               00106*16
    WRITE(6,66) (X0    (I),I=1,NEQ)                               00107*16
    WRITE(6,66) (X1    (I),I=1,NEQ)                               00108*16
    WRITE(6,66) (X2    (I),I=1,NEQ)                               00109*16
 67 FORMAT(5X,'        (I-1) TH. STEP ITERATION VALUES FOR DISPL.  ') 00110*16
                                                                  00111*16
     WRITE(6,101) NS,NELTYP,ND,NUMGI,MODEL,NCON,IDW,NK015,NK016,NK017
101 FORMAT(5X,10I5)
53  CONTINUE
                                                                  00112*14
                                                                  00061**4
      IGR=ITR                                                     00062*22
                                                                  00063*22
                                                                  00113*14
                                                                  00114*18
       CALL OUT(X1,X0,X2,VEL,AC,A0,B,ID,DS,AMAX,FMAX,NUMNP,NUME   00115*18
                                                                  00116*18
  *L,NC,AM,EK,AN,AN1,EKJ,TM,LC,NEQ,RE,ITR)                        00117*14
                                                                  00118*14
        WRITE(6,66) (UACC(I),I=1,NEQ)
                                                                  00119**8
                                                                  00120**4
                                                                  00121*16
    CALL ASSEMB(FE,NUMEL,NEQ)                                     00122*16
                                                                  00123*16
    REWIND 21                                                     00124*16
         IF(NC.EQ.0) GO TO 51
                                                                  00101*12
        IF(LCOUNT.NE.ISTRPR) GO TO 56
                                                                  00102**8
 51 CONTINUE
                                                                  00125*16
    WRITE(6,68)                                                   00126*16
 68    FORMAT(1H1/5X,'   RESIDUAL LOAD VECTOR AT (I-1) STEP  '/)  00127*18
    WRITE(6,66) (RE(I),I=1,NEQ)                                   00128*16
                                                                  00129*16
 56 CONTINUE
                                                                  00130*16
                                                                  00131**4
                                                                  00132
    ******  CALCULATE OUT OF BALANCE LOADS FOR THIS ITERATION  ****  00133
                                                                  00134
    DO  390  I=1,NEQ                                              00135
390  R(I)=GR(I)-FE(I)                                             00136*14
                                                                  00137*11
    DO 391 I=1,NEQ
391 RE(I)=0.0
     WRITE(6,66) (R(I),I=1,NEQ)

                                                                  00138
     GO TO (1,1,3,3,3),NDYN                                       00139*11
                                                                  00140*11
  1  CONTINUE                                                     00141*11
                                                                  00142*11
     *********   CALCULATE THE EFFECT OF ACCLN,VEL,DISPL AT PREVIOS ST00143*11
                                                                  00144*12
```

```
           WRITE(6,66) (UACC(I),I=1,NEQ)
           WRITE(6,66) (XM(I),I=1,NEQ)

          DO 61 I=1,NEQ                                                  00145*12
          RE(I)= RE(I)+XM(I)*UACC(I)                                     00146*14
   61     CONTINUE                                                       00147*12
            WRITE(6,66) (RE(I),I=1,NEQ)
                                                                         00148*12
           DO 155 I=1,NEQ
  155      R(I)=R(I)-RE(I)

            WRITE(6,66) (R(I),I=1,NEQ)
           DO 156 I=1,NEQ
  156      RE(I)=0.0
           DO 157 I=1,NEQ
  157     RE(I)= RE(I)+A5(I,I)*UVEL(I)                                   00146*14
                                                                         00149*12
              DO 480 I=1,NEQ                                             00150*12
              LQ=I-1                                                     00151*12
              MR=MIN0(MBAND,NEQ-I+1)                                     00152*12
              IF(MR.EQ.1) GO TO 485                                      00153*12
              DO 470 J=2,MR                                              00154*12
              K=LQ+J                                                     00155*12
                                                                         00156*14
          RE(I)=RE(I)+A5(I,J)*UVEL(K)                                    00157*14
          RE(K)=RE(K)+A5(I,J)*UVEL(I)                                    00158*14
                                                                         00159*14
  470         CONTINUE                                                   00160*12
  480         CONTINUE                                                   00161*12
  485         CONTINUE                                                   00162*12
                                                                         00163*12
           WRITE(6,66) (RE(I),I=1,NEQ)
          DO 158 I=1,NEQ
  158     R(I)=R(I)-RE(I)
                                                                         00164*12
           WRITE(6,66) (R(I),I=1,NEQ)
                                                                         00165*11
    3   CONTINUE                                                         00166*11
             IF(NC.EQ.0) GO TO 57
                                                                         00101*12
            IF(LCOUNT.NE.ISTRPR) GO TO 58
                                                                         00102**8
   57 CONTINUE
                                                                         00167*11
      WRITE(6,69)                                                        00168*16
   69 FORMAT(5X,'   UNBALANCED LOAD VECTOR ARRAY      '/)                00169*16
      WRITE (6,66) (R(I),I=1,NEQ)                                        00170*17
                                                                         00171*16
   58 CONTINUE
                                                                         00172*11
                                                                         00173
           IF(ITR.LE.(ITEMX/2+2))   GO TO 201                            00174*18
      ** CALCULATE NORM OF OUT OF BALNCE LOADS AND CHECK FOR DIVERGENCE 00175
                                                                         00176
      RENORM=0.                                                          00177
      DO 410 I=1,NEQ                                                     00178
                                                                         00179*24
  410   RENORM=RENORM+R(I)*R(I)                                          00180*24
```

```
                                                                   00181*24
                                                                   00182*26
       IF(RETORD.GE.RNORM) GO TO 201                               00183*26
                                                                   00184
                                                                   00185*24
                                                                   00186*24
                                                                   00187*14
     WRITE(6,2050) ITR                                             00188
                                                                   00189
     RETURN                                                        00190
                                                                   00191
                                                                   00192
 201 CONTINUE                                                      00193*10
         IF(NC.NE.0) GO TO 200                                     00194**8
                                                                   00195**8
       CALL SYMSOL(1,A3,R,NEQ,MBAND)                               00196*14
       CALL SYMSOL(2,A3,R,NEQ,MBAND)                               00197*14
                                                                   00198*14
                                                                   00199*12
                                                                   00200*12
           GO TO 400                                               00201*12
                                                                   00202*12
 200 CONTINUE                                                      00203**9
                                                                   00204**8
                                                                   00205**4
                                                                   00206*11
       CALL REDVK(A4,R,MAXA,NEQB,NWA,NEQ,NBLOCK,MI,MBAND,NC,KKS)   00207*14
                                                                   00208*11
                                                                   00209*11
 460   CONTINUE                                                    00210*11
                                                                   00211*11
                                                                   00212*11
                                                                   00213**4
                                                                   00214
 400 CONTINUE                                                      00215
                                                                   00216
       ******   UPDATE DISPL FOR CURRENT ITERATION STEP *******     00217*14
        DO 100 I=1,NEQ                                             00218*14
        AO(I)= AO(I)+R(I)                                          00219*14
        UDISPL(I)= XO(I) +AO(I)                                    00220*14
 100    CONTINUE                                                   00221*14
          IF(NC.EQ.0) GO TO 59
                                                                   00101*12
          IF(ICOUNT.NE.IC*KPR) GO TO 19
                                                                   00102**8
  59 CONTINUE
                                                                   00222
     WRITE(6,71)                                                   00223*16
  71 FORMAT(5X,'     INCREMENTAL  R(I)  DISPL. VECTOR FOR THIS ITER. ') 00224*16
     WRITE(6,66) (R(I),I=1,NEQ)                                    00225*16
     WRITE(6,79)                                                   00223*16
  79 FORMAT(5X,'     INCREMENTAL  AO(I) DISPL. VECTOR FOR THIS ITER. ') 00224*16
     WRITE(6,66) (AO(I),I=1,NEQ)                                   00225*16
     WRITE(6,72)                                                   00226*16
  72 FORMAT(5X,'   UPDATED DISPL. COMP.  AO( I)   VECTOR  '/)      00227*16
     WRITE(6,66) (UDISPL(I),I=1,NEQ)                               00228*16
                                                                   00229*16
  19 CONTINUE
                                                                   00230*16
                                                                   00231
```

```
     **** CHECK FOR CONVERGENCE  ************

      DNORM=0.
      DINORM=0.
      DO  420  I=1,NEQ
         DNORM=DNORM+UDISPL(I)*UDISPL(I)

  420  DINORM=DINORM+R(I)*R(I)
      TOL=DNORM*RTOL

         IF(DINORM.LE.TOL) GO TO 430
       WRITE(6,2011) DINORM,DNORM,RTOL,TOL
 2011  FORMAT(5X,4E10.3)

      IF(ITR.LT.ITEMX)  GO  TO  50
      WRITE(6,2010) NC,ITR
      WRITE(6,2020)
      STOP

   50 CONTINUE

  430     CONTINUE

          DO 99 I=1,NEQ
          AU(I)=UDISPL(I)-XU(I)
   99     CONTINUE
           IF(NC.EQ.0) GO TO 81

          IF(LCOUNT.NE.ISTRPR) GO TO 62
   81 CONTINUE

         WRITE(6,73)
   73   FORMAT(1H1/5X,'  * ***** NEW DISPL. INC.  U,I  THIS ITR.  ** '/)
           WRITE(6,74) (AU(I),I=1,NEQ)
   74    FORMAT(5X,8E10.3)
         WRITE(6,75) NC,ITR,DNORM,DINORM
   75    FORMAT(5X,  '               NC             = ',I5/
    *            5X,'               ITR            = ',I5/
    *            5X,'               DNORM          = ',E10.3/
    *            5X,'               DINORM         = ',E10.3/)

   62   CONTINUE
```

```
      GO TO (2,2,4,4,4),MDYN

    4 CONTINUE


         DO 5  I=1,NEQ
    5    X0(I)=UDISPL(I)


      GO TO 8


    2   CONTINUE



         *******   CALCULATE CURRENT CONVERGED ACCLN,VEL DISPL.ETC ***


          DELT=DT
          DO 7 I=1,NEQ
          U0=X0(I)
          U10=X1(I)
          U20=X2(I)
          U=AU(I)
          X2(I)=A04*U+A05*U10+A06*U20
          X1(I)=U10+DELT/2.*(U20+X2(I))
          X0(I)=U0+DELT*U10+A08*(X2(I)+2.*U20)
    7     CONTINUE



    8 CONTINUE






       ********** CONVERT HERE CURRENT INC.DISPL(CONVERGED) TO TOTAL AT


      WRITE(6,2012)
 2012 FORMAT(1H1/5X,'  CONVERGED VALUES OF DISPL,VEL,ACCN IN DEQITR ')
      WRITE(6,2013) (X0(I),I=1,NEQ)
      WRITE(6,2013) (X1(I),I=1,NEQ)
      WRITE(6,2013) (X2(I),I=1,NEQ)
 2013 FORMAT(5X,12E10.3)
     RETURN

2010 FORMAT(//// 37H EQUILIBRIUM ITERATION IN TIME STEP = ,I5//
   *              37H NUMBER OF ITERATIONS              = ,I5  /)
2020 FORMAT(////46H ITERATION LIMIT REACHED  S T O P OF SOLUTION    )
2050 FORMAT(////70H OUT-OF-BALANCE LOADS LARGER THAN INCREMENTAL LOADS
    *AFTER ITERATION = ,I5/)

     END
```

```
           DATA SET DRUCK      AT LEVEL 027 AS OF 83/02/16
          SUBROUTINE DRUCK(PROP,SIG,EPS,ANGLE,COHES,IPEL,PF,PS,FACN,LM, 00001*19
                                                                        00002*19
     *PMAX,NUMEL)                                                       00003*18
      IMPLICIT REAL*8(A-H,O-Z)                                          00004
                                                                        00005
                                                                        00006
 • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •00007
 •                                                                     •00008
 •       IST     NUMBER OF STRESS COMPONENTS                           •00009
 •       ISR     NUMBER OF STRAIN COMPONENTS                           •00010
 •       SIG     STRESSES AT THE END OF THE PREVIOUS UPDATE            •00011
 •       EPS     STRAINS  AT THE END OF THE PREVIOUS UPDATE            •00012
 •       RATIO   PART OF STRAIN INCREMENT TAKEN ELASTICALLY            •00013
 •       DELEPS  INCREMENT IN STRAINS                                  •00014
 •       DELSIG  INCREMENT IN STRESSES, ASSUMING ELASTIC BEHAVIOR      •00015
 •                                                                     •00016
 •       PROP(1) YOUNG S  MODULUS                                      •00017
 •       PROP(2) POISSON S  RATIO                                      •00018
 •       PROP(3) ANGLE OF FRICTION (INPUT UNIT = DEGREES)              •00019
 •       PROP(4) COHESION                                              •00020
 •                                                                     •00021
 •       IPEL    =  1, MATERIAL ELASTIC,                                00022
 •               =  2, MATERIAL PLASTIC                                 00023
 •                                                                     •00024
 • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •00025
                                                                        00026
                                                                        00027
      COMMON/EL/IND,ICOUNT                                              00028**2
      COMMON/VAR/NG,KPRI                                                00029**2
           COMMON/MATMOD/STRESS(9), STRAIN(4),C(4,4),IPT,NEL            00030*18
      COMMON /DRPRAG/ ISR,IST,A1,B1,C1,A2,B2,C2,D2,G,BM,ALFA,YLD        00031
      COMMON /DRPRAG/ A1,B1,C1,A2,B2,C2,D2,G,BM,ALFA,YLD,ISR,IST        00032
      COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017   00033**2
                                                                        00034**2
           COMMON/SORIN/TT(5,12),XP(18),INDNL,ITYP2D                    00035*17
      COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00036*16
      COMMON/CONTL/ DD,HED(8),RADN,NUMNP,NUMSL,NEQ,MBAND,NCYCL,NDYN,IPLN00037**2
     *AX,ISTRPR,NPRINT,ISTART,NEQH,MBANDH                               00038**2
                                                                        00039**2
           COMMON/PRIM/ISTRES,MPRINT,JPRINT,KPRINT,IP                   00040*17
           COMMON/FUBA/ABA,FM,FPSV                                      00041*19
                                                                        00042*19
           COMMON/KALI/JPI,KPL,JPLOT(200),KELMT(200)                    00043*17
                                                                        00044*17
          COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT            00045*19
                                                                        00046*17
                                                                        00047**2
          COMMON/GORA/IFLAG,ITR                                         00048*19
                                                                        00049*19
                                                                        00050**2
           DIMENSION PMAX(NUMEL,1)                                      00051*17
                                                                        00052*17
                                                                        00053
       DIMENSION PROP(NCON),SIG(4),EPS(4)                               00054**2
      DIMENSION TAU(4),DELSIG(4),DELEPS(4),DEPS(4),STATE(2)             00055
                                                                        00056
      EQUIVALENCE  (DELEPS(4),DEPS(4))                                  00057**2
                                                                        00058**2
      DIMENSION  PF(1)                                                  00059**2
```

```
    DIMENSION PS(16,8),FACN(4),LM(16)
          DIMENSION GSTRES(9)
    DATA NGLAST/1000/, STATE/1HE,1HP/
         1
          TIME=IND*DD

    IF (IPT.NE.1) GO TO 110

    IST=4
    IF (ITYP2D.EQ.2) IST=3
    ISR=3
    IF (ITYP2D.EQ.0) ISR=4
    YM=PROP(1)
    PV=PROP(2)
    D2=PV/(PV-1.)
    G =YM/(1.+PV)/2.
    BM=YM/(1.-2.*PV)/3.
    A2=BM+4.*G/3.
    B2=BM-2.*G/3.
    C1=G
    C2=G

          ********   CHANGE FOR UNDRAINED CHARACTERISTIC  *******

    IF (ITYP2D.EQ.2) GO TO 105

    PLANE STRAIN  /  AXISYMMETRIC
    A1=A2
    B1=B2
    GO TO 110

    PLANE STRESS
105 A1=YM/(1.-PV*PV)
    B1=A1*PV

110 ANG=ANGLE
    CEE=COHES
    YM=DCOS(ANG)
    PV=DSIN(ANG)
    YLD=1.732050808*(3.-PV)

    ALFA=2.*PV/YLD
    YLD=6.*CEE*YM/YLD

        1. CALCULATE INCREMENTAL STRAINS

    DO 120 I=1,ISR
120 DELEPS(I) = STRAIN(I) - EPS(I)

        2. CALCULATE THE STRESS INCREMENT,
           ASSUMING ELASTIC BEHAVIOR

    DELSIG(1) = A1*DELEPS(1) + B1*DELEPS(2)
    DELSIG(2) = B1*DELEPS(1) + A1*DELEPS(2)
```

```
     DELSIG(3) = C1*DELEPS(3)
     DELSIG(4) = 0.
     IF (ITYP2D.EQ.2) GO TO 150
     DELSIG(4) = B1 * (DELEPS(1)+DELEPS(2))
     IF (ITYP2D.EQ.1) GO TO 150
     DELSIG(1) = DELSIG(1) + B1*DELEPS(4)
     DELSIG(2) = DELSIG(2) + B1*DELEPS(4)
     DELSIG(4) = DELSIG(4) + A1*DELEPS(4)


        3. CALCULATE TOTAL STRESSES,
           ASSUMING ELASTIC BEHAVIOR

 150 TAU(4) = 0.
     DO 160 I=1,IST
 160 TAU(I) = SIG(I) + DELSIG(I)

        4. CHECK WHETHER *TAU* STATE OF STRESS FALLS
           OUTSIDE THE LOADING SURFACE

     SM = (TAU(1)+TAU(2)+TAU(4))/3.
     SX = TAU(1) - SM
     SY = TAU(2) - SM
     SS = TAU(3)
     SZ = TAU(4) - SM
     SBAR=DSQRT( .5*(SX*SX+SY*SY+SZ*SZ) +SS*SS)

     FT=3.*ALFA*SM + SBAR - YLD

     IF (FT) 170,170,300

     STATE OF STRESS WITHIN LOADING SURFACE  --  ELASTIC BEHAVIOR

 170 IPEL=1
     STRESS(4) = 0.
     DO 180 I=1,IST
 180 STRESS(I) = TAU(I)
     IF (ITYP2D.EQ.2)  STRAIN(4)=EPS(4) + D2*(DELEPS(1)+DELEPS(2))
     GO TO 400


     STATE OF STRESS OUTSIDE LOADING SURFACE  -  PLASTIC BEHAVIOR

 300 IF (IPEL.EQ.1) GO TO 320

     .......WAS PLASTIC

     IPEL=2
     RATIO = 0.
     DO 315 I=1,IST
 315 TAU(I) = SIG(I)
     GO TO 370

     .......WAS ELASTIC

     DETERMINE PART OF STRAIN TAKEN ELASTICLY

 320 IPEL=2

     SM = (SIG(1)+SIG(2)+SIG(4))/3.
```

```
      SX = SIG(1) - SM
      SY = SIG(2) - SM
      SS = SIG(3)
      SZ = SIG(4) - SM

      DM = (DELSIG(1)+DELSIG(2)+DELSIG(4))/3.
      DX = DELSIG(1) - DM
      DY = DELSIG(2) - DM
      DS = DELSIG(3)
      DZ = DELSIG(4) - DM

      A=DX*DX + DY*DY + 2.*DS*DS + DZ*DZ - 18.*( (ALFA*DM)**2 )
      B=SX*DX + SY*DY + 2.*SS*DS + SZ*DZ + ALFA*DM*(6.*YLD-18.*ALFA*SM)
      E=SX*SX + SY*SY + 2.*SS*SS + SZ*SZ +
     1  ALFA*SM*(12.*YLD-18.*ALFA*SM) - 2.*YLD*YLD

      RATIO=(-B + DSQRT(B*B-A*E))/A

      DO 350 I=1,IST
  350 TAU (I) = SIG(I) + RATIO*DELSIG(I)
      IF (ITYP2D.EQ.2)  STRAIN(4)=EPS(4)+RATIO*D2*(DELEPS(1)+DELEPS(2))

      *TAU* NOW CONTAINS (PREVIOUS STRESSES +
      STRESSES DUE TO ELASTIC STRAIN INCREMENTS)


         5. CALCULATE PLASTIC STRESSES

      DETERMINE INCREMENT INTERVAL
  370 M=20.*DSQRT(2.*YLD*ET)/YLD+1.
      IF (M.GT.30) M=30
      XM = (1.-RATIO)/FLOAT(M)


      DO 380 I=1,ISR
  380 DEPS(I) = XM*DELEPS(I)

      .
      .....  CALCULATION OF ELASTOPLASTIC STRESSES  .....  (START)

      DO 600 IM=1,M

      CALL PRAGER (TAU,DEPS,C)

      DO 560 I=1,IST
      DO 560 J=1,ISR
  560 TAU(I) = TAU(I) + C(I,J) * DEPS(J)


      CORRECTION

      DM = (TAU(1)+TAU(2)+TAU(4))/3.
      DX = TAU(1) - DM
      DY = TAU(2) - DM
      DS = TAU(3)
      DZ = TAU(4) - DM

      PERFECTLY PLASTIC MATERIAL

      SBAR=DSQRT(.5*(DX*DX+DY*DY+DZ*DZ)+DS*DS)
```

```
      FI=3.*ALFA*DM + SBAR - YLD                                        00240
      FIA=DABS(FI)/YLD                                                  00241
      IF (FIA.LT.0.005) GO TO 600                                       00242
                                                                        00243
      A=.25 - 9.*(ALFA**4)                                              00244
      B=SBAR - 18.*(ALFA**3)*DM + 6.*(ALFA**2)*YLD                      00245
      E=(SBAR**2) - 9.*(ALFA**2)*(DM**2) + 6.*ALFA*YLD*DM - YLD*YLD     00246
      E=(B-DSQRT(B*B-4.*A*E))/(2.*A)                                    00247
                                                                        00248
      TAU(1)=TAU(1) - E*(ALFA + .5*DX/SBAR)                             00249
      TAU(2)=TAU(2) - E*(ALFA + .5*DY/SBAR)                             00250
      TAU(3)=TAU(3) - E*(       .5*DS/SBAR)                             00251
      IF (IST.EQ.3) GO TO 600                                           00252
      TAU(4)=TAU(4) - E*(ALFA + .5*DZ/SBAR)                             00253
                                                                        00254
 600  CONTINUE                                                          00255
                                                                        00256
      .....  CALCULATION OF ELASTOPLASTIC STRESSES  .....  ( END )      00257
                                                                        00258
      STRESS(4) = 0.                                                    00259
      DO 390 I=1,IST                                                    00260
 390  STRESS(I) = TAU(I)                                                00261
                                                                        00262*14
 400     CONTINUE                                                       00263*19
                                                                        00264*19
                                                                        00265*19
      IF(IFLAG.EQ.0)  GO  TO  610                                       00266*14
                                                                        00267
                                                                        00268
      UPDATING STRESSES,STRAINS,ANGLE,COHESION                          00269
                                                                        00270
          DO 410 I=1,IST                                                00271*19
 410  SIG(I) = STRESS(I)                                                00272
      DO 420 I=1,ISR                                                    00273
 420  EPS(I) = STRAIN(I)                                                00274
      IF (ITYP2D.EQ.2)  EPS(4)=STRAIN(4)                                00275
      ANGLE=ANG                                                         00276
      COHES=CEF                                                         00277
                                                                        00278
 610  CONTINUE                                                          00279*14
                                                                        00280
                                                                        00281
         7. FORM THE MATERIAL LAW                                       00282
                                                                        00283
      IF (IPEL.EQ.1) GO TO 450                                          00284
                                                                        00285
      ELASTO-PLASTIC                                                    00286
      CALL PRAGER (STRESS,DEPS,C)                                       00287
                                                                        00288**3
                                                                        00289**3
                                                                        00290**3
                                                                        00291**3
                                                                        00292*12
      PG=PF(IPT)                                                        00293*12
        CALL PRSTS(STRESS,PG,GSTRES)                                    00294*12
      CALL  MODCON(C,NEL,PG,GSTRES)                                     00295*12
                                                                        00296*12
      IF(IFLAG.NE.0)   GO  TO  620                                      00297*14
                                                                        00298*14
                                                                        00299*14
```

```
            CALL ITRSLD(NEL,IPT,GSTRES,FACN,PS,TT,PG,LM,XP)          00300*17
                                                                      00301*17
                                                                      00302*14
      RETURN                                                          00303*14
                                                                      00304*14
 620  CONTINUE                                                        00305*14
                                                                      00306*14
                                                                      00307*14
                                                                      00308**5
            IF(IND)  820,700,820                                      00309*17
                                                                      00310*17
 820        CONTINUE                                                  00311*17
                                                                      00312*12
                                                                      00313*11
                                                                      00314**5
      IF(ISTRES.EQ.ISTRPR)  GO  TO  700                               00315**5
      RETURN                                                          00316
                                                                      00317
      ELASTIC                                                         00318
 450  DO  460  I=1,IST                                                00319**6
      DO  460  J=1,IST                                                00320**6
 460  C(I,J)=0.                                                       00321
      C(1,1)=A1                                                       00322
      C(2,1)=B1                                                       00323
      C(1,2)=B1                                                       00324
      C(2,2)=A1                                                       00325
      C(3,3)=C1                                                       00326
                                                                      00327**3
                                                                      00328**3
       *******  CHANGE  HERE  FROM  ORIGINAL                          00329**3
      C(1,4)=B1                                                       00330
      C(2,4)=B1                                                       00331
      C(4,1)=B1                                                       00332
      C(4,2)=B1                                                       00333
      C(4,4)=A1                                                       00334
                                                                      00335
                                                                      00336**3
                                                                      00337**3
                                                                      00338**4
                                                                      00339
                                                                      00340*10
 470  CONTINUE                                                        00341*10
                                                                      00342
                                                                      00343**4
                                                                      00344*12
      PG=PF(IPT)                                                      00345*12
        CALL PRSTS(STRESS,PG,GSTRES)                                  00346*12
      CALL  MODCON(C,NEL,PG,GSTRES)                                   00347*12
                                                                      00348*12
      IF(IFLAG.NE.0)  GO  TO  630                                     00349*14
                                                                      00350*14
                                                                      00351*14
                                                                      00352*17
            CALL ITRSLD(NEL,IPT,GSTRES,FACN,PS,TT,PG,LM,XP)          00353*17
                                                                      00354*14
      RETURN                                                          00355*14
                                                                      00356*14
                                                                      00357*14
 630  CONTINUE                                                        00358*14
                                                                      00359*11
```

```
          IF(IND)809,700,809

809       CONTINUE




    IF(ISTRES.EQ.ISTRPR)  GO  TO  700
    RETURN


    P R I N T I N G  O F  S T R E S S E S

700 IF (IPEL.EQ.1) GO TO 800

    DM = (STRESS(1)+STRESS(2)+STRESS(4))/3.
    DX = STRESS(1) - DM
    DY = STRESS(2) - DM
    DS = STRESS(3)
    DZ = STRESS(4) - DM
    SBAR=DSQRT(.5*(DX*DX+DY*DY+DZ*DZ)+DS*DS)
    FT=3.*ALFA*DM + SBAR - YLD


800 CONTINUE
    IF (IPT-1) 810,808,810

802 CONTINUE
808 IF (ITYP2D) 803,805,803
803 WRITE(6,2002)
    GO TO 806
805 WRITE(6,2003)

806 CONTINUE
    WRITE (6,2004) NEL



810       CALL MAXMIN(STRESS,SX,SY,SM,BLKST,SHSTN,PMAX,PG,IPT,TIME,NUME

   * L,SP,SD,SHM)

    IF (ITYP2D) 813,815,813

813 WRITE (6,2005) IPT,STATE(IPEL),(STRESS(I),I=1,3),SX,SY,SM,FT,PF(IP
   *T)

          GO TO 817

815 WRITE (6,2007) IPT,STATE(IPEL),STRESS(4),(STRESS(I),I=1,3),
   1                SX,SY,SM,FT,PF(IPT)



817       CONTINUE


          IF(NEL.EQ.0) GO TO 816
          DN=FELMT(IE)
```

```
           IF(IN.NE.NEL) GO TO 816                                      00420*17
           IF(IPT.NE.1)   GO TO 816                                     00421*17
                                                                        00422*17
           WRITE ON TAPE NO. 57   ********                              00423*17
                                                                        00424*17
                                                                        00425*19
           SM=SP                                                        00426*22
          IF(NPUNCH.NE.0)                                               00427*19
     *WRITE(57)NEL,(STRESS(1),I=1,2),STRESS(4),STRESS(3),(STRAIN(I),I=1,00428*20
     *2),STRAIN(3),STRESS(5),STRESS(6),STRESS(8),SM,SD,PG,SHM,EPSV      00429*20
                                                                        00430*19
                                                                        00431*19
           IP=IP+1                                                      00432*17
                                                                        00433*17
  816      CONTINUE                                                     00434*17
                                                                        00435*17
           ***********    CONTOUR PLOTTING COUNTER **********8          00436*17
                                                                        00437*17
          IF(JPRINT.LT.ISTART) GO TO 819                                00438*17
          IF(IPT.NE.1) GO TO 819                                        00439*17
                                                                        00440*17
          WRITE INFORMATION ON TAPE NO. =18 ****88                      00441*17
                                                                        00442*17
           IF(NPUNCH.NE.0)                                              00443*19
                                                                        00444*20
     *WRITE(18)NEL,STRESS(5),STRESS(6),STRESS(8),SM,SD,                 00445*20
                                                                        00446*20
     *PG,SHM,EPSV                                                       00447*21
  819    CONTINUE                                                       00448*17
                                                                        00449*17
      RETURN                                                            00450
                                                                        00451
                                                                        00452
 2002 FORMAT (90H ELEMENT  STRESS                                       00453
     1                                   ,21X,5HYIELD/99H NUM/IPT   STAT00454
     2E      STRESS-YY     STRESS-ZZ     STRESS-YZ       MAX STRESS    M00455
                                                                        00456**8
     3IN STRESS    ANGLE,9X,8HFUNCTION ,2X,13HPORE PRESSURE / )         00457**8
 2003 FORMAT ( 104H ELEMENT  STRESS                                     00458
     1                                                    ,21X,5HYIELD / 00459
     2 113H NUM/IPT   STATE      STRESS-XX     STRESS-YY     STRESS-ZZ  00460
     3   STRESS-YZ       MAX STRESS    MIN STRESS    ANGLE,9X,8HFUNCTION00461
                                                                        00462**8
     *,2X,13HPORE PRESSURE / )                                          00463**8
 2004 FORMAT (I4)                                                       00464
 2005 FORMAT (5X,I2,2X,A1,6HLASTIC,1X,3E14.6,3X,2E14.6,3X,F6.2,3X,E14.6,00465**8
     *2X,E12.5)                                                         00466**8
 2007 FORMAT (5X,I2,2X,A1,6HLASTIC,1X,4E14.6,3X,2E14.6,3X,F6.2,3X,E14.6,00467**8
     *2X,E12.5)                                                         00468**8
 2100 FORMAT (1H1,21H STOP  DRUCKER-PRAGER )                            00469
                                                                        00470
      END                                                               00471
```

```
C
      SUBROUTINE EIGEN(A,R,N,MV)
      DIMENSION A(1), R(1)
C
      DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,
     *  COSX2,SINCS, RANGE
   11  CONTINUE
C
    5 RANGE = 1.0E-12
      IF (MV-1) 10,25,10
   10 IQ = -N
C
      DO 20 J=1,N
      IQ = IQ + N
      DO 20 I=1,N
      IJ = IQ + I
      R(IJ) = 0.0
      IF (I-J) 20, 15, 20
   15 R(IJ) = 1.0
   20 CONTINUE
C
C     COMPUTE INITIAL AND FINAL NORMS
C
   25 ANORM = 0.0
      DO 35 I=1,N
      DO 35 J=1,N
C
      IF (I-J) 30,35,30
   30 IA = I + (J*J-J) / 2
      ANORM = ANORM + A(IA) * A(IA)
   35 CONTINUE
      IF (ANORM) 165,165,40
   40 ANORM = 1.414 *DSQRT(ANORM)
      ANRMX =  ANORM * RANGE / FLOAT(N)
C
C     INITIALIZE INDICATORS AND COMPUTE THRESHOLD
C
      IND = 0
      THR = ANORM
   45 THR = THR / FLOAT(N)
   50 L = 1
   55 M = L + 1
C
C     COMPUTE SIN & COS
C
   60 MQ = (M*M-M) / 2
      LQ  =(L*L-L)/ 2
      LM = LQ+ MQ
   62 IF(DABS(A(LM))-THR) 130,65,65
   65 IND = 1
      LL = L + LQ
      MM = M + MQ
      X = 0.5* (A(LL)-A(MM))
   68 Y = -A(LM) /DSQRT(A(LM)* A(LM)+X * X)
      IF (X) 70,75,75
   70 Y = -Y
   75 SINX = Y /DSQRT(2.0 * (1.0 +(DSQRT(1.0 - Y*Y))))
      SINX2 = SINX * SINX
   78 COSX =DSQRT (1.0 - SINX2)
      COSX2 = COSX * COSX
```

```
      SINCS = SINX * COSX
C
C     ROTATE L AND M COLUMNS
C
      ILQ = N*(L-1)
      IMQ = N*(M-1)
      DO 125 I=1,N
      IQ = (I*I-I) / 2
      IF(I-L) 80,115,80
 80   IF (I-M) 85,115,90
 85   IM = I+MQ
      GOTO 95
 90   IM = M + IQ
 95   IF(I-L) 100,105,105
100   IL = I + LQ
      GOTO 110
105   IL = L + IQ
110   X = A(IL) * COSX - A(IM) *  SINX
      A(IM) = A(IL) * SINX + A(IM) * COSX
      A(IL) = X
115   IF (MV-1) 120,125,120
120   ILR = ILQ + I
      IMR = IMQ + I
      X = R(ILR) * COSX - R(IMR) * SINX
      R(IMR) = R(ILR) * SINX +R(IMR) * COSX
      R(ILR) = X
125   CONTINUE
      X = 2.0 * A(LM) * SINCS
      Y = A(LL) * COSX2 + A(MM)*SINX2 - X
      X = A(LL) * SINX2 + A(MM) * COSX2 + X
      A(LM) = (A(LL)-A(MM)) * SINCS + A(LM) * (COSX2-SINX2)
      A(LL) = Y
      A(MM) = X
C
C     TESTS FOR COMPLETION
C
C     TEST FOR M =  LAST COLUMN
C
130   IF (M-N) 135,140,135
135   M = M + 1
      GO TO 60
C
C     TEST FOR L = SECOND FROM LAST COLUMN
C
140   IF(L-(N-1)) 145,150,145
145   L = L + 1
      GO TO 55
150   IF(IND-1) 160,155,160
155    IND = 0
      GOTO 50
C
C     COMPARE THRESHOLD WITH FINAL EIGENVECTOR
C
160   IF(THR-ANRMX) 165,165,45
C
C     SORT EIGENVALUES AND EIGENVECTORS
C
165   IQ = -N
      DO 185 I=1,N
      IQ = IQ + N
```

```
        LL = I + (I*I-I)/2
        JQ = N *(I-2)
        DO 185 J=1,N
        JQ = JQ + N
        MM = J +(J*J - J) / 2
C       WRITE(6,169)I,IQ,LL,JQ,J,JQ,MM
        IF (A(LL) - A(MM)) 170,185,185
  170   X = A(LL)
        A(LL) = A(MM)
        A(MM) = X
C       WRITE(6,169)I,IQ,LL,JQ,J,JQ,MM,X,A(LL),A(MM)
  169    FORMAT(5X,7I5,3E16.3)
C
C
        IF (MV - 1)  175,185,175
  175   DO 180 K=1,N
        ILR = IQ + K
        IMR = JQ + K
        X = R(ILR)
        R(ILR) = R(IMR)
  180   R(IMR) = X
  185   CONTINUE
        RETURN
        END
```

```
C          DATA SET ELEMT      AT LEVEL 026 AS OF 83/02/18
      SUBROUTINE ELEMT(NUMMAT)                                          00001**4
      IMPLICIT  REAL*8   (A-H,O-Z)                                      00002
C                                                                       00003
C                                                                       00004
C                                                                       00005
C     SUBROUTINE  ELEMT  CALLS  ELSTIF  WHICH  FORMS  ELEMENT  STIFFN00006
C     MASSES,AND  DAMPING  MATRICES                                     00007
C                                                                       00008
C                                                                       00009
      COMMON/CONTL/DT,HED(6),RADN,NUMNP,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA00010
     *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS                    00011**5
      COMMON/QUAD/CCCC(1500),N1,N2,N3,N4,N5,MTOT,NNN,NK15               00012*20
C                                                                       00013*19
C                                                                       00014*19
      COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00015*24
      COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017   00016*15
C                                                                       00017
C                                                                       00018*25
      COMMON/DUMMY/KKS,NUMDT,NC,LOOP,NDRN,NINT,IDWA,NPT                 00019*25
C                                                                       00020*25
      COMMON/CLOCK/TIM1,TIM2,TIM3                                       00021*23
      COMMON D(1)                                                       00022*23
C                                                                       00023
      REWIND  14                                                        00024*17
C                                                                        00026**7
      IF(NC.GT.0.AND.KKS.EQ.3)  GO  TO  500                             00027**7
      READ(5,1000) NUMMAT,NELTYP,NS                                     00028*11
      WRITE(6,2000) NUMMAT,NELTYP,NS                                    00029*11
      IF(KKS.EQ.3)  WRITE(14) NUMMAT,NELTYP,NS                          00030*11
      GO  TO  600                                                       00031**7
C                                                                       00032**7
C                                                                       00033**7
  500 READ(14) NUMMAT,NELTYP,NS                                         00034*11
  600 CONTINUE                                                          00035**7
C                                                                       00036
C                                                                       00037**7
C                                                                       00038**7
      IF(NS.EQ.0)  NS=4                                                 00039*11
       IF(NELTYP.EQ.0)  ND=8                                            00040*11
      IF(NELTYP.EQ.1)  ND=12                                            00041*11
      IF(NELTYP.EQ.2)  ND=16                                            00042*11

C     WRITE(6,1) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW



      NK6=NUMMAT+N5                                                     00043**7
      NK7=NUMMAT+NK6                                                    00044**7
      NK8=NUMMAT+NK7                                                    00045**7
      NK9=NUMMAT+NK8                                                    00046**7
      NK10=NK9+NUMMAT                                                   00047**7
      NK11=NK10+NUMMAT                                                  00048**7
      NK12=NUMMAT+NK11                                                  00049**7
      NK13=NK12+NUMMAT                                                  00050**7
          NK14=NK13+NUMEL                                               00051*26
C                                                                       00052*12
C       *******   CHANGE  FOR  ELASTO PLASTIC  ANALYSIS  ***************00053*12
C                                                                       00054*12
C                                                                       00055*12
```

```
        NK015=NK14+NUMEL                                                 00056*26
      NK016=NK015+NUMEL                                                  00057*12
      NK017=NK016+NCON*NUMMAT                                            00058*12
      IDWA=IDW*NINT*NINT                                                 00059*19
      NK15=NK017+IDWA*NUMEL                                              00060*19
      NPT=NINT*NINT                                                      00061*19
C                                                                        00062**7
C                                                                        00063**7
      IF(NK15.LE.MTOT)  GO  TO  20                                       00064**7
      NNN=0                                                              00065
      GO  TO  100                                                        00066
   20 CONTINUE                                                           00067
C                                                                        00068**7

C     WRITE(6,1) N1,N2,N3,N4,N5,NK6,NK7,NK8,NK9,NK10,NK11,NK12,NK13,NK14
C    *,NK015,NK016,NK017,NK15,NUMNP,NUMMAT

C                                                                        00069**7
      CALL ELSTIF(D(N1),D(N2),D(N3),D(N4),D(N5),D(NK6) ,D(NK7),D(NK8),  00070*10
C                                                                        00071*11
     * D(NK9),D(NK10),D(NK11),NUMNP,NUMMAT,D(NK12),D(NK13),D(NK14),D(NK000072*12
C                                                                        00073*13
     *15),D(NK016),D(NK017))                                             00074*12
C                                                                        00075*11
    2 FORMAT(5X,2I5,10E10.3/15X,10E10.3/15X,10E10.3/15X,10E10.3/15X,10E
     *10.3/)
C
    1 FORMAT(5X,2I5,22I5)

C     WRITE(6,4)
    4 FORMAT(5X,'   PRINT IN ELEMT  ')
C
C     J=N3-1
C
C
C
C     WRITE(6,2) N2,J,(D(I),I=N2,J)
      J=N4-1
C     WRITE(6,2) N3,J,(D(I),I=N3,J)
      J=N5-1
C
      J=NK6-1
C     WRITE(6,2) N5,J,(D(I),I=N5,J)
      J=NK7-1
C
C     WRITE(6,2) NK6,J,(D(I),I=NK6,J)
C                                                                        00076*21
      J=NK8-1
C      WRITE(6,2) NK7,J,(D(I),I=NK7,J)
      J=NK9-1
C      WRITE(6,2) NK8,J,(D(I),I=NK8,J)
      J=NK10-1
C      WRITE(6,2) NK9,J,(D(I),I=NK9,J)
      J=NK11-1                                                           00077*21
C      WRITE(6,2)NK10,J,(D(I),I=NK10,J)
      J=NK12-1
C      WRITE(6,2)NK11,J,(D(I),I=NK11,J)
      J=NK13-1
C     WRITE(6,2)NK12,J,(D(I),I=NK12,J)
      J=NK14-1
C     WRITE(6,2) NK13,J,(D(I),I=NK13,J)
```

```
      J=NK015-1
C     WRITE(6,2) NK14,J,(D(I),I=NK14,J)
      J=NK016-1
C
C
      J=NK017-1
C     WRITE(6,2) NK016,J,(D(I),I=NK016,J)
C     J=NK15-1
C     WRITE(6,2)NK017,J,(D(I),I=NK017,J)
C
 100  RETURN                                                                00078
C                                                                           00079
1000  FORMAT(3I5)                                                           00080*11
2000  FORMAT(1H1/'                   NUMBER OF MATERIALS = ',I5/            00081*11
     *        '                   CODE FOR ELEMENT TYPE = ',I5/             00082*11
     *  ' EQ.0. 4-NODED RECTANGULAR ELEMENT (LINEAR)   '/                   00083*11
     *  ' EQ.1. 4-NODED RECTANGULAR ELEMENTS WITH INCOMPATIBLE MODES ' /00084*11
     *  ' EQ.2. 8-NODED RECTANGULAR ELEMENTS WITH QUADRATIC   MODESS ' /00085*11
     *        '                   NUMBER OF STRESS OUTPUT = ',I5/           00086*11
     *  ' EQ.0. STRESS OUTPUT RQESTED AT CENTER      ' /                    00087*11
     *  ' EQ. 20 . STRESS OUTPUT REQESTED AT CENTER  PLUS  MIDSIDE  ' /)00088*11
      END                                                                   00089
```

```
        DATA SET ELINT      AT LEVEL 010 AS OF 82/12/02
      SUBROUTINE  ELINT(GMS,NE,X,Y,VOL,TTX0,TTX1,TTX2,NEL)


      IMPLICIT  REAL*8 (A-H,O-Z)



      COMMON/EMB/S(16,16),GM(16),GH(16,16),SS(16,16),XP(16),XN(
     *16),LM(16)
      DIMENSION TT1(12),TT2(12),TTX1(2,12),TTX0(12,2),TTX2(12,12)
      COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),
     *ST(20,12),D(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8)
      COMMON/BFLA /NS,NELTYP ,ND

       COMMON/RANI/RM(5),ZM(5),IPT
      DIMENSION GMS(8),LK(8),LM(8),X(1),Y(1)



       ***********   INITIALIZATION   OF   ELEMENT   MATRICES   ************


      DO  85  I=1,ND
      DO  85  J=1,ND
      SF(I,J)=0.
      H(I,J)=0.
      E(I,J)=0.
      C(I,J)=0.
   85 CONTINUE


       DO  860  I=1,NS
      DO  860 J=  1,ND
      ST(I,J)=0.0
  860 CONTINUE
      DO  90  J=1,8
   90 GMS(J)=0.0
      DO  80  I=1,16
      GM(I)=0.
      XP(I)=0.0
      XN(I)=0.0
      DO  80  J=1,16
      S(I,J)=0.0
      SS(I,J)=0.0
      GH(I,J)=0.0
   80 CONTINUE


      DO  100 I=1,4
        RM(I)=0.0
      ZM(I)=0.0
  100 CONTINUE


************************************************************************
************************************************************************


      CALCULATES  VOLUME  HERE  FOR  MASS  CALCULATION  CHANGE  HERE  FO
      SISTENT  MASS   MATRIVES
```

```
                                                                       00060**6
     *******  IS  IT  ALSO O.K.   FOR  TRIANGLE    ?    ***************00061**6
                                                                       00062**6
     R21=RR(2)-RR(1)                                                   00063
     Z31=ZZ(3)-ZZ(1)                                                   00064
     R31=RR(3)-RR(1)                                                   00065
     Z41=ZZ(4)-ZZ(1)                                                   00066
     Z21=ZZ(2)-ZZ(1)                                                   00067
     R41=RR(4)-RR(1)                                                   00068
     VOL=(R21*Z31 +R31*Z41-R31*Z21-R41*Z31)                            00069**2
     VOL=VOL/2.                                                        00070
     IF(VOL.GT.0.0) GO  TO  81                                         00071
     WRITE(6,2012)  NEL                                                00072
     STOP                                                              00073
  81 CONTINUE                                                          00074
                                                                       00075
                                                                       00076
***********************************************************************00077
                                                                       00078
                                                                       00079
***********************************************************************00080
                                                                       00081
                                                                       00082
                                                                       00083**4
                                                                       00084**4
                                                                       00085**4
     IF(IJUMP.NE.0)  GO  TO  615                                       00086
     DO  8615  I=1,2                                                   00087
     DO  8615 J=1,ND                                                   00088**4
     TTX0(J,I)=0.0                                                     00089
8615 TTX1(I,J)=0.0                                                     00090
     DO  8616  I=1,ND                                                  00091**4
     DO  8616  J=1,ND                                                  00092**4
8616  TTX2(I,J)=0.0                                                    00093
                                                                       00094
 615 CONTINUE                                                          00095**2
     RETURN                                                            00096
2012 FORMAT (//'         ERROR IN  ELEMENT  CARD   = ',I10/)           00097
     END                                                               00098
```

```
      DATA SET ELNIF       AT LEVEL 027 AS OF 82/06/12
  SUBROUTINE ELNIF(FK,DBETA1,DBETA2,NUMMAT,ELMDA,EMU,EALFA,EM,EK,ROS

 *,ROF,MBAND,MBANDH,NEL,XKK,INEL,INP,IMAT,KN,KJUMP,IJUMP,IKS,IKF,

 *PRESS,X,Y,RM,ZM,RR,ZZ,ML,NUMEL,NP,MBANDS,THICK)

  IMPLICIT REAL*8  (A-H,O-Z)



  COMMON/BOBBI/KKS,NUMGT,NC,LOOP,NDRN,NINT,IDWA,NPT
  DIMENSION  INP(4),NP(4),ELMDA(1),EMU(1),EALFA(1),EM(1),EK(1),

 *ROS(1),ROF(1),X(1),Y(1),RR(4),ZZ(4)

  DIMENSION FK(1),DBETA1(1),DBETA2(1)
  DIMENSION PRESS(5)

  DIMENSION RM(5),ZM(5)



  IF(ML.GT.0)   GO  TO  40
  IF(NEL.NE.0)  GO  TO  800
  READ  ELEMENT  INFORMATION   AND  CALL  LOAD  ROUTINE  FOR CALCULA
  LOAD  VECTOR  FOR  STATIC  ANALYSIS





  IF(NC.GT.0)  GO  TO  100
  WRITE(6,2002)




   *******   BRANCH  HERE  FOR ELASTOPLASTIC  ANALYSIS    **********





  DO  20  N=1,NUMMAT

  READ(5,1003) ELMDA(N),EMU(N),EALFA(N),EM(N),EK(N),ROS(N),ROF(N),
 *FK(N),DBETA1(N),DBETA2(N)


  WRITE(6,2003)   ELMDA(N),EMU(N),EALFA(N),EM(N),EK(N),ROS(N),ROF(N)
 *,FK(N),DBETA1(N),DBETA2(N)




100 CONTINUE
 20 CONTINUE
  MBAND=0
  MBANDH=0
  MBANDS=0
  NEL=0
```

```
      XKK=0.577350269189626                                             00060
      IF(NINT.EQ.1) XKK=0.0
      IF(NC.GT.0.AND.KKS.EQ.3)  GO  TO  800                             00061*24
      WRITE(6,2006)                                                     00062
  800 CONTINUE                                                          00063
      IF(NUMEL-NEL) 50,700,30                                           00064
                                                                        00065**9
                                                                        00066**9
                                                                        00067**9
   30 CONTINUE                                                          00068**9
      IF(NC.GT.0.AND.KKS.EQ.3)  GO  TO  200                             00069**9
                                                                        00070**9
      READ(5,1007) INEL,INP,IMAT,KN,KJUMP,IKS,IKF,PRESS,THICK           00071**9
      IF(KKS.EQ.3) WRITE(14)       INEL,INP,IMAT,KN,KJUMP,IKS,IKF,PRESS,00072**9
     *THICK                                                             00072**9
      GO  TO  300                                                       00073**9
  200 READ(14) INEL,INP,IMAT,KN,KJUMP,IKS,IKF,PRESS,THICK               00074**9
  300 CONTINUE                                                          00075**9
                                                                        00076**9
                                                                        00077**9
                                                                        00078**9
      IJUMP=0                                                           00079
   40 NEL=NEL+1                                                         00080
      ML=INEL-NEL                                                       00081
      IF(ML)  50,55,60                                                  00082
   50 WRITE(6,2007)  INEL                                               00083
      STOP                                                              00084
                                                                        00085**9
                                                                        00086**9
                                                                        00087**9
   55 DO  56  I=1,4                                                     00088
   56 NP(I)=INP(I)                                                      00089
      MAT=IMAT                                                          00090
      KS=IKS                                                            00091
      KF=IKF                                                            00092
      GO  TO  62                                                        00093
   60 DO  61  I=1,4                                                     00094
   61 NP(I)=NP(I)+KN                                                    00095
      IF(KJUMP.NE.0)  IJUMP=1                                           00096
   62 CONTINUE                                                          00097*23
      IF(NC.GT.0.AND.KKS.EQ.3)  GO  TO  63                              00098*23
                                                                        00099*23
      WRITE(6,2008)  NEL,NP,MAT,KS,KF,PRESS,THICK                       00100*23
      WRITE(49,1007) NEL,NP,MAT,KN,KJUMP,KS,KF,PRESS,THICK              00100*23
                                                                        00101*23
   63 CONTINUE                                                          00102*23
                                                                        00103**9
                                                                        00104**9
      *******   CHECK  ALSO  FOR  TRIANGLE  CENTROID  CALCULATION   ARBIT00105*14
                                                                        00106*14
      DO  70  I=1,4                                                     00107
      M=NP(I)                                                           00108
      RR(I)=X(M)                                                        00109
   70 ZZ(I)=Y(M)                                                        00110
                                                                        00111*25
                                                                        00112*25
      RM(5)=0.25*(RR(1)+RR(2)+RR(3)+RR(4))                              00113*25
      ZM(5)=0.25*(ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))                              00114*25
                                                                        00115*25
  700 CONTINUE                                                          00116
```

```
      RETURN                                                            00117
                                                                        00118**9
                                                                        00119**9
 1003 FORMAT(8F10.3)                                                    00120**6
 1007 FORMAT(5I3,5I2,5F10.3,F5.2)                                       00121*12
 2002 FORMAT (//'      MATERIAL PROPERTIES'//2X,'MATERIAL'/2X,'NUMBER', 00122
     * 7X,'E',11X,'NU',10X,'N',11X,'M',11X,'K',10X,'ROS',9X,'ROF',5X,'FK00123**6
     *',5X,'DBETA1',5X,'DBETA2'//)                                      00124**7
 2003 FORMAT(5X,10E12.3)                                                00125**6
 2006 FORMAT (1H1/'      ELEMENT DATA'//3X,'ELEMENT',14X,'NODE''NUMBERS',00126
     *25X,'MATERIAL'/3X,'NUMBER',10X,'1',9X,'2',9X,'3',9X,'4',11X,'NUMBE00127
     *R',40X,'PRESSURE LOAD'/)                                          00128
 2007 FORMAT (//'      ELEMENT CARD ERROR, ELEMANT',I5)                 00129
 2008 FORMAT(5X,I5,4I10,5X,I10,2I3,5E10.3,3X,F5.2)                      00130*19
      END                                                               00131
```

```
        DATA SET ELPAL      AT LEVEL 042 AS OF 83/03/05
       SUBROUTINE ELPAL(PROP,SIG,EPS,YIELD,IPEL,PF,PS,FACN,LM,PMAX,   00001*32
                                                                      00002*28
   *NUMEL)                                                            00003*28
                                                                      00004*28
    IMPLICIT REAL*8(A-H,O-Z)                                          00005
                                                                      00006
                                                                      00007
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  00008
.                                                                  .  00009
.      IST     NUMBER OF STRESS COMPONENTS                         .  00010
.      ISR     NUMBER OF STRAIN COMPONENTS                         .  00011
.      SIG     STRESSES AT THE END OF THE PREVIOUS UPDATE          .  00012
.      EPS     STRAINS  AT THE END OF THE PREVIOUS UPDATE          .  00013
.      RATIO   PART OF STRAIN INCREMENT TAKEN ELASTICALLY          .  00014
.      DELEPS  INCREMENT IN STRAINS                                .  00015
.      DELSIG  INCREMENT IN STRESSES, ASSUMING ELASTIC BEHAVIOR    .  00016
.                                                                  .  00017
.      PROP(1) YOUNG S  MODULUS                                    .  00018
.      PROP(2) POISSON S  RATIO                                    .  00019
.      PROP(3) INITIAL YIELD STRESS IN SIMPLE TENSION              .  00020
.      PROP(4) STRAIN HARDENING MODULUS                            .  00021
.                                                                  .  00022
.      IPEL    = 1, MATERIAL ELASTIC                               .  00023
.              = 2, MATERIAL PLASTIC                               .  00024
.                                                                  .  00025
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  00026
                                                                      00027
                                                                      00028
    COMMON/EL/IND,ICOUNT                                              00029**2
    COMMON/VAR/NG,KPRI                                                00030**2
    COMMON /VMISES/ A1,B1,C1,D1,A2,B2,C2,D2,YLD,BM,ISR,IST            00031
        COMMON/MATMOD/STRESS(9), STRAIN(4),C(4,4),IPT,NEL             00032*29
                                                                      00033
    DIMENSION PROP(NCON),SIG(4),EPS(4)                                00034**7
    DIMENSION TAU(4),DELSIG(4),DELEPS(4),DEPS(4),STATE(2)             00035
        COMMON/SORIN/TT(5,12),XP(16),INDNL,ITYP2D                     00036*28
    COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00037*27
                                                                      00038
                                                                      00039*33
      COMMON/DUBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT              00040*33
                                                                      00041*33
                                                                      00042**4
    COMMON/DEDA/PS,NELTIP,ND,NUMGI,MODEL,NCON,IDW,NK015,NK016,NK017   00043**4
                                                                      00044**7
    COMMON/CONTL/DD,HED(8),KADN,NUMNP,NUMSL,NEG,MBAND,NCYCL,NDYN,IPLNA00045**7
   *X,ISTEPR,NPRINT,ISTART,NEGH,MBANDH                                00046**7
        COMMON/PRIM/ISTRES,MPRINT,JPRINT,KPRINT,IP                    00047*28
       COMMON/PUBA/ALFA,EM,EPSV,SHMD                                  00048*42
        COMMON/KALI/JPT,KEL,JPLOT(200),KELMT(200)                     00049*28
        COMMON/CHORDA/WP
                                                                      00050*28
         DIMENSION PMAX(NUMEL,1)                                      00051*28
                                                                      00052**7
    DIMENSION  PF(1)                                                  00053*10
        DIMENSION GSTRES(9)                                           00054*31
    DIMENSION PS(16,8),FACN(4),LM(16)                                 00055*25
                                                                      00056**7
    EQUIVALENCE (DELEPS(4),DEPS(4))                                   00057**5
                                                                      00058
```

```


        COMMON/GORA/IFLAG,ITR
    DATA NGBAST/1000/, STATE/1HE,1HP/



       WP=0.0

          TIME=IND*DD


    IF (IPT.NE.1) GO TO 110

    IST=4
    IF (ITYP2D.EQ.2) IST=3
    ISR=3
    IF (ITYP2D.EQ.0) ISR=4
    YM=PROP(1)
    PV=PROP(2)
    D1=PV/(PV - 1.)
    A2=YM/(1.+PV)
    B2=(1.-PV)/(1.-2.*PV)
    C2=PV/(1.-2.*PV)
    D2=YM*PROP(4)/(YM-PROP(4))
    C1=A2/2.
    BM=YM/(1. - 2.*PV)/3.


          ************    UNDRAINED CHARACTERSTIC *******

    IF (ITYP2D.EQ.2) GO TO 105

    PLANE STRAIN  /  AXISYMMETRIC
    B1=A2*C2
    A1=B1+A2
    GO TO 110

    PLANE STRESS
105 A1=YM/(1.-PV*PV)
    B1=A1*PV

110 YLD = YIELD


       1. CALCULATE INCREMENTAL STRAINS

    DO 120 I=1,ISR
120 DELEPS(I) = STRAIN(I) - EPS(I)


       2. CALCULATE THE STRESS INCREMENT,
          ASSUMING ELASTIC BEHAVIOR

    DELSIG(1) = A1*DELEPS(1) + B1*DELEPS(2)
    DELSIG(2) = B1*DELEPS(1) + A1*DELEPS(2)
    DELSIG(3) = C1*DELEPS(3)
    DELSIG(4) = 0.
    IF (ITYP2D.EQ.2) GO TO 150
```

```
      DELSIG(4) = D1 * (DELEPS(1)+DELEPS(2))
      IF (ITYP2D.EQ.1) GO TO 150
      DELSIG(1) = DELSIG(1) + B1*DELEPS(4)
      DELSIG(2) = DELSIG(2) + B1*DELEPS(4)
      DELSIG(4) = DELSIG(4) + A1*DELEPS(4)


          3. CALCULATE TOTAL STRESSES,
             ASSUMING ELASTIC BEHAVIOR

  150 TAU(4) = 0.
      DO 160 I=1,IST
  160 TAU(I) = SIG(I) + DELSIG(I)

          4. CHECK WHETHER *TAU* STATE OF STRESS FALLS
             OUTSIDE THE LOADING SURFACE

      SM = (TAU(1)+TAU(2)+TAU(4))/3.
      SX = TAU(1) - SM
      SY = TAU(2) - SM
      SS = TAU(3)
      SZ = TAU(4) - SM

      FT =.5*(SX*SX+SY*SY+SZ*SZ) + SS*SS - YLD*YLD/3.

          ***********   TEMPORARY PRINT OUT   *******

          WRITE(6,161)
  161     FORMAT(5X,' *** PRINT IN ELPAL ***',/)
          WRITE(6,162) NC,ITR,IFLAG,NEL,IPT
  162     FORMAT(5X,16I5)
          WRITE(6,163) (STRAIN(I),I=1,4)
          WRITE(6,163) (EPS(I),I=1,4)
          WRITE(6,163) (DELEPS(I),I=1,4)
          WRITE(6,163) (SIG(I),I=1,4)
          WRITE(6,163) (DELSIG(I),I=1,4)
          WRITE(6,163) (TAU(I),I=1,4)
          WRITE(6,163) (STRESS(I),I=1,4)
          WRITE(6,163) SM,SX,SY,SS,SZ,YLD,PF(IPT),FT
  163     FORMAT(5X,8E10.3)





      IF (FT) 170,170,300

      STATE OF STRESS WITHIN LOADING SURFACE  -  ELASTIC BEHAVIOR

  170 IPEL=1
      STRESS(4) = 0.
      DO 180 I=1,IST
  180 STRESS(I) = TAU(I)
      IF (ITYP2D.EQ.2) STRAIN(4)=EPS(4) + D1*(DELEPS(1) + DELEPS(2))
      GO TO 400


      STATE OF STRESS OUTSIDE LOADING SURFACE  -  PLASTIC BEHAVIOR
```

```
300 IF (IPEL.EQ.1) GO TO 320

    .......WAS PLASTIC

    IPEL=2
    RATIO = 0.
    DO 315 I=1,IST
315 TAU(I) = SIG(I)
    GO TO 370

    .......WAS ELASTIC

    DETERMINE PART OF STRAIN TAKEN ELASTICLY

320 IPEL=2

    SM = (SIG(1)+SIG(2)+SIG(4))/3.
    SX = SIG(1) - SM
    SY = SIG(2) - SM
    SS = SIG(3)
    SZ = SIG(4) - SM

    DM = (DELSIG(1)+DELSIG(2)+DELSIG(4))/3.
    DX = DELSIG(1) - DM
    DY = DELSIG(2) - DM
    DS = DELSIG(3)
    DZ = DELSIG(4) - DM

    A = DX*DX + DY*DY + 2.*DS*DS + DZ*DZ
    B = SX*DX + SY*DY + 2.*SS*DS + SZ*DZ
    E= SX*SX + SY*SY + 2.*SS*SS + SZ*SZ - 2.*YLD*YLD/3.

    RATIO=(-B+DSQRT(B*B-A*E))/A

    DO 350 I=1,IST
350 TAU (I) = SIG(I) + RATIO*DELSIG(I)
    IF (ITYP2D.EQ.2) STRAIN(4)=EPS(4) + RATIO*D1*(DELEPS(1)
   1                    + DELEPS(2))

    *TAU* NOW CONTAINS (PREVIOUS STRESSES +
    STRESSES DUE TO ELASTIC STRAIN INCREMENTS)


       5. CALCULATE PLASTIC STRESSES

    DETERMINE INCREMENT INTERVAL
370 M=20.*DSQRT(ET)/YLD+1
    IF (M.GT.30) M=30
    XM = (1. - RATIO)/M


    DO 380 I=1,IST
380 DEPS(I) = XM*DELEPS(I)


    .....  CALCULATION OF ELASTOPLASTIC STRESSES  .....  (START)

    DO 600 IM=1,M

    CALL MIDEP (TAU,DEPS,C)
```

```
      DO 560 I=1,IST
      DO 560 J=1,ISR
  560 TAU(I) = TAU(I) + C(I,J) * DEPS(J)


      CORRECTION

      DM = (TAU(1)+TAU(2)+TAU(4))/3.
      DX = TAU(1) - DM
      DY = TAU(2) - DM
      DS = TAU(3)
      DZ = TAU(4) - DM

      IF (PROP(4).EQ.0.) GO TO 580

      STRAIN-HARDENING MATERIAL  -  UPDATE YLD

      YLD=DSQRT (1.5*(DX*DX+DY*DY+2.*DS*DS+DZ*DZ))
      GO TO 600

      PERFECTLY PLASTIC MATERIAL

  580 FTA=.5*(DX*DX + DY*DY + DZ*DZ) + DS*DS
      FTB=(YLD*YLD)/3.
      FT=FTA - FTB
      IF (FT.EQ.0) GO TO 600
      IF (ITYP2D.EQ.2) GO TO 590

      COEF=-1. +DSQRT(FTB/FTA)
      TAU(1) = TAU(1) + COEF*DX
      TAU(2) = TAU(2) + COEF*DY
      TAU(3) = TAU(3) + COEF*DS
      TAU(4)=TAU(4) + COEF*DZ
      GO TO 600

  590 COEF=DSQRT(FTB/FTA)
      TAU(1)=TAU(1)*COEF
      TAU(2)=TAU(2)*COEF
      TAU(3)=TAU(3)*COEF
      STRAIN(4)=STRAIN(4) + (COEF - 1.)*DM/BM

  600 CONTINUE



      .....  CALCULATION OF ELASTOPLASTIC STRESSES  .....  ( END )

      STRESS(4) = 0.
      DO 390 I=1,IST
  390 STRESS(I) = TAU(I)


  400    CONTINUE

      IF(IFLAG.EQ.0)  GO  TO  610


         6. UPDATING STRESSES, STRAINS, YIELD, NS
```

```
                                                                        00284*32
         DO 410 I=1,IST                                                 00285*32
                                                                        00286*32
410 SIG(I) = STRESS(I)                                                  00287
    DO 420 I=1,ISR                                                      00288
420 EPS(I) = STRAIN(I)                                                  00289
    YIELD = YLD                                                         00290
    IF (ITYP2D.EQ.2) EPS(4)=STRAIN(4)                                   00291
                                                                        00292*24
610 CONTINUE                                                            00293*24
                                                                        00294*10
                                                                        00295*10
                                                                        00296
                                                                        00297
                                                                        00298
       7. FORM THE MATERIAL LAW                                         00299
                                                                        00300
    IF (IPEL.EQ.1) GO TO 450                                            00301
                                                                        00302
    ELASTO-PLASTIC                                                      00303
    CALL MIDEP (TAU,DEPS,C)                                             00304
                                                                        00305**6
                                                                        00306**6
                                                                        00307**6
    ********  REALIGN  THE  CONSTITUTIVE  MATRIX  TO  BE  COMPATIBLE 00308**6
    *******   CONSV  SUBROUTINE.  ALSO  CALCULATE  THE  RESIDUAL LOAD 00309**6
   *****  OF  ELEMENT  LEVEL  AND  WRITE IT ON A TAPE  TO  BE  ASSEMBL00310**6
   ***  SUBROUTINE  KSTAR.                                              00311**6
                                                                        00312**6
                                                                        00313**6
                                                                        00314**6
                                                                        00315*15
                                                                        00316*15
    PG=PF(IPT)                                                          00317*22
                                                                        00318*22
    CALL  PRNIS(STRESS,PG,GSTRES)                                       00319*23
    CALL MODCON(C,NEL,PG,GSTRES)                                        00320*22
                                                                        00321*24
                                                                        00322*24
    IF(NEQUIB.EQ.1.AND.IFLAG.NE.0)  GO  TO  620                                         00323*24
                                                                        00324*24
                                                                        00325*24
         CALL ITRSLD(NEL,IPT,GSTRES,FACN,PS,TT,PG,LM,XP)                00326*28
                                                                        00327*28
     WRITE(6,165)
165 FORMAT(5X,' PRINT IN ELFAL ')
     WRITE(6,166) ((C(I,J),J=1,4),I=1,4)
166 FORMAT(5X,4E10.3)
                                                                        00328*24
       WRITE(6,162) IFLAG,IPEL
       WRITE(6,163) (GSTRES(I),I=1,4),FACN(IPT)
       WRITE(6,163) (XP(I),I=1,16)
       WRITE(6,162) (LM(I),I=1,16)
                                                                        00329*24
      IF(NEQUIB.EQ.0) GO TO 620
    RETURN                                                              00330*24
                                                                        00331*24
620 CONTINUE                                                            00332*24
                                                                        00333*28
         IF(IND)  820,700,820                                           00334*28
```

```
                                                            00335*28
820     CONTINUE                                            00336*28
                                                            00337*28
                                                            00338*24
    IF(ISTRES.EQ.ISTRPR)  GO  TO  700                       00339*12
                                                            00340**6
    RETURN                                                  00341
                                                            00342
    ELASTIC                                                 00343
450 DO  460  I=1,IST                                        00344*13
    DO  460  J=1,IST                                        00345*13
460 C(I,J)=0.                                               00346
    C(1,1)=A1                                               00347
    C(2,1)=B1                                               00348
    C(1,2)=B1                                               00349
    C(2,2)=A1                                               00350
    C(3,3)=C1                                               00351
                                                            00352*10
     CHECK  HERE  FROM  ORIGINAL  ******************        00353*10
440 CONTINUE                                                00354**9
    C(1,4)=B1                                               00355
    C(2,4)=B1                                               00356
    C(4,1)=B1                                               00357
    C(4,2)=B1                                               00358
    C(4,4)=A1                                               00359
430 CONTINUE                                                00360**8
                                                            00361*22
    PG=PF(IPT)                                              00362*22
    WRITE(6,166) A1,B1,C1

                                                            00363*22
    CALL  PRSIS(GSTRESS,PG,GSTRES)                          00364*23
    CALL MODCON(C,NEL,PG,GSTRES)                            00365*22
                                                            00366*26
                                                            00367*26
    IF(NEQUIB.NE.0.AND.IFLAG.NE.0)  GO  TO  630                       00368*25
                                                            00369*25
                                                            00370*25

        CALL ITRSLD(NEL,IPT,GSTRES,FACN,FS,TT,PG,LM,XP)     00371*28
     WRITE(6,165)
165 FORMAT(5X,' PRINT IN ELEAL ')
     WRITE(6,166) ((C(I,J),J=1,4),I=1,4)
166 FORMAT(5X,4E10.3)
                                                            00372*25
       WRITE(6,162) IFLAG,IFFL
       WRITE(6,163) (GSTRES(I),I=1,4),FACN(IPT)
       WRITE(6,163) (XP(I),I=1,16)
       WRITE(6,162) (LM(I),I=1,16)
      IF(NEQUIB.EQ.0) GO TO 630
    RETURN                                                  00373*25
                                                            00374*25
630 CONTINUE                                                00375*25
                                                            00376*22
                                                            00377*13
                                                            00378*12
        IF(IND)  809,700,809                                00379*28
                                                            00380*28
```

```
809       CONTINUE                                                     00381*28
                                                                       00382*28
    IF(ISTRES.EQ.ISTRPR)  GO  TO  700                                  00383*12
                                                                       00384
    RETURN                                                             00385
                                                                       00386
                                                                       00387
    P R I N T I N G   O F   S T R E S S E S                            00388
                                                                       00389
700 IF (IPEL.EQ.1) GO TO 705                                           00390
                                                                       00391
    DM=(STRESS(1) + STRESS(2) + STRESS(4))/3.                          00392
    DX=STRESS(1) - DM                                                  00393
    DY=STRESS(2) - DM                                                  00394
    DS=STRESS(3)                                                       00395
    DZ=STRESS(4) - DM                                                  00396
    FT=.5*(DX*DX + DY*DY + DZ*DZ) + DS*DS - YLD*YLD/3.                 00397
                                                                       00398
705 CONTINUE                                                           00399**8
                                                                       00400*18
        SHM=SHMD                                                       00401*42
    IF(IPT-1)  810,800,810                                             00402*18
                                                                       00403**8
800 CONTINUE                                                           00404*10
                                                                       00405*10
802 CONTINUE                                                           00406*10

    IF(NPUNCH.NE.0) GO TO 810

808 IF (ITYP2D) 803,805,803                                            00407
803 WRITE(6,2002)                                                      00408*22
807 CONTINUE                                                           00409*16
    GO TO 806                                                          00410
805 WRITE(6,2003)                                                      00411*22
                                                                       00412*22
                                                                       00413
806 CONTINUE                                                           00414*10
                                                                       00415*10
                                                                       00416*10
                                                                       00417*10
                                                                       00418*10
    WRITE (6,2004) NEL                                                 00419
810      CALL MAXMIN(STRESS,SX,SY,SM,BLKST,SHSTN,PMAX,PG,IPT,TIME,NUME00420*28
                                                                       00421*35
   * L,SP,SD,SHM)                                                      00422*38

     IF(NPUNCH.NE.0) GO TO 817

    IF (ITYP2D) 813,815,813                                            00423
                                                                       00424
813 WRITE (6,2005) IPT,STATE(IPEL),                                    00425
                                                                       00426*20
   1                 (STRESS(I),I=1,3),SX,SY,WP,FT  ,PF(IPT)           00427*20
                                                                       00428*28
         GO TO 817                                                     00429*28
                                                                       00430*28
                                                                       00431*28
                                                                       00432
815 WRITE (6,2007) IPT,STATE(IPEL),STRESS(4),                          00433
   1                 (STRESS(I),I=1,3),SX,SY,WP,FT  ,PF(IPT)           00434*20
```

```
                                                                          00435*28
                                                                          00436*28
 817      CONTINUE                                                        00437*28
                                                                          00438*28
                                                                          00439*28
          IF(KEL.EQ.0) GO TO 816                                          00440*28
          LN=KELMAT(IP)                                                   00441*28
          IF(LN.NE.NEL) GO TO 816                                         00442*28
          IF(IPT.NE.1)   GO TO 816                                        00443*28
                                                                          00444*28
          WRITE ON TAPE NO. 57   ********                                 00445*28
                                                                          00446*35
                                                                          00447*38
          SM=SP                                                           00448*38
          IF(NPUNCH.NE.0)                                                 00449*35
    *WRITE(57,2032)                                                       00450*36
    *, NEL,(STRESS(I),I=1,2),STRESS(3),(STRAIN(I),I=1,                    00450*36
    *2),STRAIN(3),STRESS(5),STRESS(6),STRESS(8),SM,SD,PG,SHM,EPSV,FT      00451*36
2032 FORMAT(I3,7E11.4/8E10.3/)
                                                                          00452*28
          IP=IP+1                                                         00453*28
                                                                          00454*28
 816      CONTINUE                                                        00455*28
                                                                          00456*28
                                                                          00457*40
                IF(NC) 821,822,821                                        00458*40
                                                                          00459*40
                                                                          00460*40
 821             CONTINUE                                                 00461*40
          ************    CONTOUR PLOTTING COUNTER ***********8           00462*28
                                                                          00463*28
         IF(OPRINT.LT.ISTART) GO TO 819                                   00464*28
                                                                          00465*40
 822             CONTINUE                                                 00466*40
                                                                          00467*40
                                                                          00468*40
         IF(IPT.NE.1) GO TO 819                                           00469*28
                                                                          00470*28
         WRITE INFORMATION ON TAPE NO. =18 ******                         00471*28
                                                                          00472*35
                                                                          00473*35
           IF(NPUNCH.NE.0)                                                00474*35
    *WRITE(18,2033)NEL,STRESS(1),STRESS(2),STRESS(3),STRESS(8),           00475*36
    *PG,SHM,EPSV,STRAIN(3)                                                00476*37
2033 FORMAT(I5,8E9.2,3X)
                                                                          00477*37
                                                                          00478*35
                                                                          00479*28
 819     CONTINUE                                                         00480*28
                                                                          00481*28
      RETURN                                                              00482
                                                                          00483
                                                                          00484
2002 FORMAT (90H ELEMENT  STRESS        STRESS-YY      STRESS-ZZ     STRES00485
     1S-YZ       MAX STRESS     MIN STRESS,21X,5HYIELD/99H NUM/IPT    STAT00486
     2E                                                                   00487
     3              ANGLE,9X,8HFUNCTION ,2X,13HPORE PRESSURE / )          00488*20
2003 FORMAT ( 104H ELEMENT  STRESS        STRESS-XX      STRESS-YY     STR00489
     1ESS-ZZ     STRESS-YZ        MAX STRESS     MIN STRESS,21X,5HYIELD / 00490
     299H NUM/IPT    STATE                                                00491
```

```
     3                              E,9X,8HFUNCTION,                      00492
     *2X,13HPORE PRESSURE / )                                             00493*20
 2005 FORMAT (5X,I2,2X,A1,6HLASTIC,1X,3E14.6,3X,2E14.6,3X,E9.2,1X,E14.6,00494*20
     *1X,E12.5)                                                           00495*20
 2007 FORMAT (5X,I2,2X,A1,6HLASTIC,1X,4E10.3,3X,2E14.6,3X,E9.2,1X,E14.6,00496*20
     *1X,E12.5)                                                           00497*20
 2004 FORMAT (I4/)                                                        00498
                                                                          00499
      END                                                                 00500
```

```
C        DATA SET ELSTIF      AT LEVEL 071 AS OF 83/03/05
      SUBROUTINE ELSTIF (ID,X,Y,IH,ELMDA,EMU,EALFA,EM,EK,ROS,ROF,NUMNP, 00001
C                                                                       00002*30
     *NUMMAT,FK,DBETA1,DBETA2,MATP,PRP,WA)                              00003*30
C                                                                       00004*31
      IMPLICIT  REAL*8   (A-H,O-Z)                                      00005
C                                                                       00006
C                                                                       00007
C     SUBROUTINE  ELSTIF  GENERATES  MASSMATRICES(BOTH  SOLID AND FLUID)00008
C     MATRICES(BOTH  FOR  SOLID AND FLUID),AND STIFFNESS  MATRICES(SOLID00009
C     AND COUPLING  MATRICES), LOAD VECTOR  ELEMENTWISE  FOR  STATIC ANA00010
C     PLANE STRAIN AND AXISYMMETRIC  CASE)                              00011
C                                                                       00012
C                                                                       00013
C                                                                       00014
      COMMON/CONTB/DT,MED(6),RADN,NASIM,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA00015
     *X,ISTRFK,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS                    00016*12
C                                                                       00017*32
      COMMON/EMB/S(16,16),GM(16),GH(16,16),SS(16,16),XP(16),XN(         00018*62
     *16),BM(16)                                                        00019*62
C                                                                       00020*64
      COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00021*65
C                                                                       00022*50
      COMMON/HOBBY/KKS,NUMGAT,NC,LOOP,NORN,NINT,IDWA,NPT                00023*50
      COMMON/ANBN/FA                                                    00024*20
      COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),    00025*27
     *ST(20,12),D(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8),UDD(4,4)      00026*37
     *,PS(16,8),FACN(4)                                                 00027*63
C                                                                       00028*50
C                                                                       00029*29
C                                                                       00030*29
      COMMON/HELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017   00031*29
C                                                                       00032
C                                                                       00033
C                                                                       00034
      COMMON/ABHR/C1,C2,C3,AL,ALF,FACC,FACT                             00035
C                                                                       00036
C                                                                       00037*57
       COMMON/RANI/RM(5),ZM(5),IPT                                      00038*68
C                                                                       00039*57
      COMMON/DIMEL/N43                                                  00040*60
C                                                                       00041*60
      DIMENSION  ID(NUMNP,1),X(1),Y(1),ELMDA(1),EMU(1),EALFA(1),EM(1),  00042
     *EK(1),ROS(1),ROF(1),NP(4),INP(4),TTT(2),SSS(2),IG(8),IL(8),IH(NUMN00043
     *P,1),FK(1),DBETA1(1),DBETA2(1)                                    00044*12
C                                                                       00045
C                                                                       00046
      DIMENSION TT1(12),TT2(12),TTX1(2,12),TTX0(12,2),TTX2(12,12)       00047*27
      DIMENSION GMS(8),LK(8),LH(8),PRESS(5)                             00048*27
C                                                                       00049*29
C                                                                       00050*29
C                                                                       00051*29
      DIMENSION  PRP(NCON,1),WA(IDWA,1)                                 00052*57
C                                                                       00053*57
C                                                                       00054*57
C                                                                       00055*59
      DIMENSION  MATP(1)                                                00056*59
C     <*****************   CALL  ELEMENT  INFORMATION   *************** 00057**2
C                                                                       00058**2
      REWIND 11                                                         00059*12
```

```
      REWIND  13
      REWIND  50
      REWIND  51
C
        REWIND 17
C
   42 FORMAT(5X,I5)
C     **************  TAPE  11  CONTAINS  ELEMENT  INFORMATION  FOR  STRE
C     **************  1AP TAPE  13  CAONTAINS  ELEMENT  MATRICES  FOR  AS
C     *****************  TAPE  50  AND  51  CONTAINS  CONSTITUTIVE  MATRI
C     STRAINN  DISPLAVEMENT  MATRICES  FOR  SHAKEDOWN  ANALYSIS          **
C
C
C
C
      MB=0
      NEL=0
   40 CONTINUE
C
      CALL ELMIF(FK,DBETA1,DBETA2,NUMMAT,ELMDA,EMU,EALFA,EM,EK,ROS,ROF,
C
     *MBAND,MBANDH,NEL,XKK,INEL,INP,IMAT,KN,KJUMP,IJUMP,IKS,IKF,PRESS,X,
C
     *Y,RM,ZM,RR,ZZ,MB,NUMEL,NP,MBANDS,THICK)
C
C
C     ********   ELEMENT   MATRICES   INITIALIZATION
C
C
      CALL ELINT(GMS,NP,X,Y,VOL,TTX0,TTX1,TTX2,NEL)
C
C
C     *********   MODIFY  VOL  FOT  THICKNESS  ***********************
C
C
      VOL=VOL*THICK
C
C
C
      IF(NC.GT.0.AND.KKS.EQ.3)  GO  TO  4200
C
      CALL  CONSV(IMAT,EALFA,ELMDA,EMU,IPLNAX,D,BLK,YOUNG,POIS,NEL,
C
     *NC,EM,DDD)
C
C
C
C
 4200 CONTINUE
C
C************
      MATP(NEL)= IMAT
C
C
      K=IMAT
      DENS=ROS(K)/FA
      DENF=ROF(K)/FA/FK(K)
      PERM=EK(K)
C
      RS=ROS(K)
```

```
      RF=ROF(K)
      COEFS=POIS/(1.-POIS)
      COEFR=1.0
C
C
C*********************************************************************
C
C
C
C*********************************************************************
C     CALL  FORMB  AND  CREATE  ELEMENT  STIFFNESSES  BY  GAUSS  INTEGRA
C     FOR  FOUR  NODED  ISOPARAMETRIC  ELEMENT  **MODIFY  FOR  EIGHT  NO
C
C
C
       CALL  FORMFX(XKK,IPLMAX,IKS,IKF,PERM,TTX0,TTX1,TTX2,NUMNP,THICK,
C
     *NEL,WA,PRESS,FS,RS,RF,COEFS,COEFR,FACN)
C
C
C
      IF(IJUMP.NE.0)  GO  TO  615
C
C
      CALL        STDSLD(NP,LM,IH,LH,NUMNP,NEL,FK,DENS,DENF,GMS,
C
     *VOL,K,F,EM,IMAT,KS,KF,THICK)
C
C
      KS=IKS
      KF=IKF
C
C
C
C
      CALL      SOFDMA(VOL,EM,IMAT,KS,KF,NEL,THICK)
C
C
C*********************************************************************
C
C
C
  615 CONTINUE
      L=0
      DO  620  I=1,4
      IIM=I
      N=NP(IIM)
      DO  620  J=1,4
      L=L+1
      LM(L)=ID(N,J)
  620 CONTINUE
C
C
C
C
      IF(KKS.EQ.2)  GO  TO  775
C     WRITE(6,400)
  400 FORMAT(5X,'  PRINT IN  ELSTIF ')
C     WRITE(6,401) IMAT,DBETA1(IMAT),DBETA2(IMAT)
  401 FORMAT(5X,I5,2E10.3)
```

```
      CALL FSFDMX(VOL,DENS,DEFF,KS,KF,DBETA1,DBETA2,F,NUMNP,NEL,        00176*17
     *ID,IH,IMAT)                                                       00177*12
      WRITE(6,400)
       WRITE(6,401) IMAT,DBETA1(IMAT),DBETA2(IMAT)
                                                                        00178
************************************************************************00179
                                                                        00180
                                                                        00181
                                                                        00182
      ***TAPE 13 **** CONTAINS  INFORMATION  ON  ELEMENT CONNECTIVITY AR00183
      STIFFNESS AND  MASS /DAMPING  MATRICES  AND  LATER  USED  IN  **KS00184
      FOR  ASSEMBLY  PURPOSES                                           00185
                                                                        00186*31
                                                                        00187*31
      ***TAPE11**  STRESS-DISPL MATRIX FOR SOLID,STRAIN -DISPL MATRIX FO00188
      CONNECTIVITY ARRAY,SOLID COMPRESSIBILITY,FLUID COMPRESSIBILITY,AND00189
      LATER  USED  IN  STRESS  SUBROUTINE                               00190
      ******CHANGED  FROM  ORIGINAL  FOR  STATIC  TRANSIENT  PROBLEM****00191
                                                                        00192*31
                                                                        00193*31
                                                                        00194*31
      ******  CALL  LOAD  ROUTINE  TO  CALCULATE  RESIDUAL  STRESSES  I00195*31
      **********  TERMS  OF  INITIAL  STRESSES  OR   NONLINEAR  ANALYSIS00196*31
                                                                        00197*31
      **********    PASS  '  **  WA  ** '  VECTOR  AT  THIS  POINT  FOR 00198*32
     *******ASTIC  ANALYSIS  TO  CALCULATE  THE  RESIDUAL  LOAD  VECTOR 00199*32
                                                                        00200*32
                                                                        00201*32
                                                                        00202*31
                                                                        00203*50
                                                                        00204*31
                                                                        00205*50
                                                                        00206*50
  840 CONTINUE                                                          00207*50
      WRITE(13) LM,S,GM,GH,SS,XP,XN                                     00208*62
                                                                        00209*13
                                                                        00210*13
  725 CONTINUE                                                          00211*13
                                                                        00212*40
      ******   REDEFINE  K  BECAUSE OF ITS USE IN STDSLD      **********00213*40
      K=IMAT                                                            00214*40
                                                                        00215*23
       WRITE(11) ST,IT,PI,LM,EALFA(K),EM(K),RM,ZM,KS,KF,IMAT,BLK,EMU(K),00216*27
     *YOUNG,PHIS,VOL,PS,FACN                                            00217*63
                                                                        00218
                                                                        00219*39
                                                                        00220*41
                                                                        00221
                                                                        00222
************************************************************************00223
                                                                        00224
                                                                        00225
                                                                        00226
                                                                        00227
                                                                        00228
                                                                        00229
                                                                        00230
                                                                        00231*12
                                                                        00232*12
      *******   FIND  BANDWIDTH  FOR  SOLID  PART  ONLY  ************** 00233*12
```

```
      DO  800 I=1,8
      L=LK(I)
      IF(L)  800,800,830
  830 DO  850  J=1,8
      JJM=J
      N=LK(JJM)
      IF(N) 850,850,855
  855 KL=IABS(N-L)
      IF(KL.GT.MBANDS)  MBANDS=KL
  850 CONTINUE
  800 CONTINUE
C
C
C
C     FINDS  BAND  WIDTH
      DO  640  I=1,16
      L=LM(I)
      IF (L) 640,640,630
  630 DO  635  J=1,16
      JJM=J
      N=LM(JJM)
      IF(N) 635,635,632
  632 KL=IABS(N-L)
      IF(KL.GT.MBAND)  MBAND=KL
  635 CONTINUE
  640 CONTINUE
      IF(ML) 50,650,40
  650 IF(NUMEL-NEL) 50,660,40
   50 WRITE(6,2007)  INEL
      STOP
C
  660 MBAND=MBAND+1
      MBANDH=MBAND
      MBANDS=MBANDS+1
        IF(NC.GT.0.AND.KKS.EQ.3) GO TO 10
      WRITE(6,2090) MBAND,MBANDH,MBANDS
C
C
  10     CONTINUE
C
      IF(KKS.NE.3)  GO  TO  700
      IF(NC.NE.0)  GO  TO  70
       DO  80  I=1,NUMMAT
         ENU=(ELMDA(I))/(2.*(ELMDA(I)+EMU(I)))
         PRP(1,I)=EMU(I)*(2.*(1.+ENU))
         PRP(2,I)=ENU
         READ(5,1008) (N,(PRP(J,N),J=3,NCON))
C
   80 CALL  PROP(N,PRP(1,N),NUMMAT)
   70 CONTINUE
C
C
C
C
C
C
C
  700 RETURN
C
C
```

```
C                                                                       00294
 1008     FORMAT(I5,4E12.5)                                             00295*70
C                                                                       00296
 1001 FORMAT (A10)                                                      00297
 2090 FORMAT(1H1//'             BAND WIDTH              = ',I10/        00298*27
     *    '     BAND WIDTH FOR H        = ',I10/                        00299*12
     *    '     BAND WIDTH FOR S        = ',I10)                        00300*12
 2007 FORMAT (//'      ELEMENT CARD ERROR, ELEMANT',I5)                 00301**4
      END                                                               00302
```

```
      DATA SET ELT2D6      AT LEVEL 020 AS OF 83/02/02
      SUBROUTINE ELT2D6(PF,FACN,PS,LM,PMAX,NUMEL)
   IMPLICIT REAL*8(A-H,O-Z)


                    M O D E L   =   6

   E L A S T O P L A S T I C   M O D E L   (VON MISES)

   COMMON/EL/IND,ICOUNT
   COMMON/DIMEL/N43
        COMMON/MATMOD/STRESS(9), STRAIN(4),C(4,4),IPT,NEL
   COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

   COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT


   COMMON D(1)
        DIMENSION PMAX(NUMEL,1)
   DIMENSION IA(400000)
   EQUIVALENCE (D(1),IA(1))
   DIMENSION  PF(1)
   DIMENSION PS(16,8),FACN(4),LM(16)


   FOR ADDRESSES N101,N102,N103,...  SEE SUBROUTINE TODMFE


    WRITE(6,15)
 15 FORMAT(5X,'      PRINT IN ELT2D6 ')
    WRITE(6,20) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
 20 FORMAT(5X,10I5)


   I N I T I A L I Z E  W A   W O R K I N G   A R R A Y
        IF(IND.NE.-1) GO TO 100

   IDW=10
   NN=NK017+(NEL-1)*NPT*IDW
   MATP=IA(2*NK015+NEL-2)
   NM=NK016+(MATP-1)*4
   DO  10  J=1,NPT
 10 CALL  IELPAL(D(NN),D(NN+9+(J-1)*10),D(NM),NPT,J)

    WRITE(6,15)
    WRITE(6,20) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

   RETURN



   F I N D   S T R E S S - S T R A I N   L A W   A N D   S T R E S S


100 CONTINUE
   IDW=10
```

```
      NN=NK017+(NEL-1)*NPI*IDW+(IPT-1)*IDW                              00054**7
      MATP=IA(2*NK015+NEL-2)                                            00055**4
      NM=NK016+(MATP-1)*4                                               00056**4

         WRITE(6,160)
160      FORMAT(5X,' ** PRINT IN ELI2D6 ** ')
         WRITE(6,161) (D(I),I=NM,NM+1)
         WRITE(6,161) (D(I),I=NN,NN+3)
         WRITE(6,161) (D(I),I=NN+4,NN+7)
161      FORMAT(5X,4E10.3)
                                                                        00057**4
           CALL ELPAD(D(NM),D(NN),D(NN+4),D(NN+8),D(NN+9),PF,PS,FACN,LM,00058*17
     *     PMAX,NUMEL)                                                  00059*17
                                                                        00060*17
                                                                        00061
      RETURN                                                            00062
                                                                        00063
                                                                        00064
      END                                                               00065
```

```
     DATA SET ELT2D7     AT LEVEL 026 AS OF 83/01/24
     SUBROUTINE ELT2D7(PF,FACN,PS,LM,PMAX,NUMEL)                      00001*25
IMPLICIT REAL*8(A-H,O-Z)                                              00002
                                                                      00003
M O D E L    =    7                                                   00004
                                                                      00005
E L A S T O P L A S T I C   M O D E L   (DRUCKER-PRAGER)             00006
                                                                      00007
                                                                      00008
COMMON/EL/IND,ICOUNT                                                  00009**6
                                                                      00010**6
COMMON/DIMEL/N43                                                      00011*18
                                                                      00012**6
COMMON/BELA/NG,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017       00013*20
                                                                      00014*20
     COMMON/MATMOD/STRESS(9), STRAIN(4),C(4,4),IPT,NEL                00015*26
                                                                      00016**7
COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT                    00017**7
                                                                      00018*18
                                                                      00019*18
COMMON  D(1)                                                          00020*18
DIMENSION IA(400000)                                                  00021**2
EQUIVALENCE (D(1),IA(1))                                              00022**2
     DIMENSION PMAX(NUMEL,1)                                          00023*25
DIMENSION  PF(1)                                                      00024**7
DIMENSION PS(16,8),FACN(4),LM(16)                                     00025*23
                                                                      00026
FOR ADDRESSES N101,N102,N103,...  SEE SUBROUTINE TODMFE               00027
                                                                      00028
                                                                      00029
MATP=IA(N107 + NEL - 1)                                               00030
                                                                      00031*16
                                                                      00032*16
                                                                      00033*16
                                                                      00034*16
                                                                      00035*16
 MATP=IA(2*NK015+NEL-2)                                               00036*18
                                                                      00037*16
NM=NK016+(MATP-1)*NCON                                                00038*18
                                                                      00039*18
                                                                      00040*11
                                                                      00041*11
     IF(IND.NE.-1) GO TO 100                                          00042*24
                                                                      00043
                                                                      00044
I N I T I A L I Z E  W A   W O R K I N G   A R R A Y                  00045
                                                                      00046
NN=NK017+(NEL-1)*NPT*IDW                                              00047**9
                                                                      00048*13
NG=NN-1                                                               00049*21
                                                                      00050*13
                                                                      00051*13
                                                                      00052*13
                                                                      00053*13
                                                                      00054*13
                                                                      00055
DO  10  J=1,NPT                                                       00056*10
                                                                      00057*11
                                                                      00058*11
                                                                      00059*21
```

```
                                                                      00060*22
      CALL  IDRUCK(D(NN),D(NN+10+(J-1)*11),D(NM),NPT,J)               00061*10
                                                                      00062*10
   10 CONTINUE                                                        00063*10
                                                                      00064
      RETURN                                                          00065
                                                                      00066
                                                                      00067
      F I N D   S T R E S S - S T R A I N   L A W   A N D   S T R E S S 00068
                                                                      00069
                                                                      00070
  100 NS=NK017+((NEL-1)*NPT+(IPT-1))*IDW                              00071**6
                                                                      00072
                                                                      00073*18
                                                                      00074*21
                                                                      00075*22
                                                                      00076*22
                                                                      00077*22
                                                                      00078*18
      CALL DRUCK(D(NM),D(NS),D(NS+4),D(NS+8),D(NS+9),D(NS+10),PF,PS,FACN00079*23
                                                                      00080*25
     *,LM,PMAX,NUMEL)                                                 00081*25
                                                                      00082
      RETURN                                                          00083
                                                                      00084
                                                                      00085
      END                                                             00086
```

```
        DATA SET EQCHK       AT LEVEL 009 AS OF 83/03/05
                SUBROUTINE EQCHK(XM,A5,B,NEQ,MBAND,X0,X1,X2,R,NC,A3,ND00001
   * YN,GB,A0,AS)                                                     00002**6
                                                                      00003
                 IMPLICIT REAL * 8(A-H,O-Z)                           00004
                                                                      00005
       COMMON/BOBBY/KKS

       DIMENSION XM(1),A5(NEQ,1),B(NEQ),X0(1),X1(1),X2(1),R(NEQ)      00006
   *,A3(NEQ,1),A0(1)                                                  00007**2
                                                                      00008
       DIMENSION GB(1)                                                00009**6
       DIMENSION AS(NEQ,1)

             ***********   READ STIFFNESS MATRIX FROM TAPE  ******    00010
                                                                      00011
             REWIND 60                                                00012
              READ(60)  ((A3(I,J),J=1,MBAND),I=1,NEQ)                 00013
                                                                      00014
              WRITE(6,56)
 56         FORMAT(1H1/5X,'  PRINT IN EQCHK ')
              WRITE(6,58)((A3(I,J),J=1,MBAND),I=1,NEQ)
              PRINT 999
              WRITE(6,58)((A5(I,J),J=1,MBAND),I=1,NEQ)
              PRINT 999
              WRITE(6,57) (XM(I),I=1,NEQ)
              PRINT 999
              WRITE(6,57)(X0(I),I=1,NEQ)
              PRINT 999
              WRITE(6,57)(X1(I),I=1,NEQ)
              PRINT 999

999         FORMAT(//)
              WRITE(6,57)(X2(I),I=1,NEQ)
 57         FORMAT(5X,12E10.3)
 58         FORMAT(5X,8E10.3)
                                                                      00015
       DO 50  I=1,NEQ                                                 00016**3
               B(I)=0.0                                               00017
 50            R(I)=0.0                                               00018
                                                                      00019
               DO 51 I=1,NEQ
               DO 51 J=1,NEQ
 51            AS(I,J)=0.0

                                                                      00020
               WRITE(6,70)                                            00021
 70      FORMAT(1H1/5X,'    OVERALL EQUILIBRIUM  CHECK    '/)         00022**8
                                                                      00023**8
               IF(NDYN.EQ.4) GO TO 60                                 00024
                                                                      00025
             COMPUTE INERTIA TERM                                     00026
                                                                      00027
             DO 1 I=1,NEQ                                             00028
  1          B(I)=B(I)+XM(I)*X2(I)                                    00029
                                                                      00030
              WRITE(6,2)                                              00031
  2          FORMAT(    5X,'   PRINT OF INERTIA TERM   ')             00032
             WRITE(6,3) (B(I),I=1,NEQ)                                00033
  3         FORMAT(5X,12E10.3)                                        00034
```

```
C
C        TRANSFER ARRAY
C
         DO 5 I=1,NEQ
  5      R(I)=B(I)

         DO 55 I=1,NEQ
 55       B(I)=0.0
C
C
C
C       COMPUTATION DISSIPATION FORCE
C
C
        DO 4 I=1,NEQ
  4     B(I)=B(I)+A5(I,1)*X1(I)
C
C
         DO 6 I=1,NEQ
         LQ=I-1
         MR=MINO(MBAND,NEQ-I+1)
         IF(MR.EQ.1) GO TO 7
          DO 8 J=2,MR
          M=LQ+J
          B(I)=B(I)+A5(I,J)*X1(M)
          B(M)=B(M)+A5(I,J)*X1(I)
  8      CONTINUE
  6      CONTINUE
  7      CONTINUE .

        WRITE(6,9)
  9     FORMAT(5X,'   PRINT  OF  DAMPING TERM  ')
     WRITE(6,3)(B(I),I=1,NEQ)                                             **4
C
C
       DO 10 I=1,NEQ
 10    R(I)=R(I)+B(I)

 60      CONTINUE
C      COMPUTATION OF STIFFNESS TERM
C
        DO  80 I=1,NEQ
 80      B(I)=0.0
C
C
        DO 12I=1,NEQ
 12     B(I)=B(I)+A3(I,1)*X0(I)
C
C
         DO 13I=1,NEQ
         LQ=I-1
         MR=MINO(MBAND,NEQ-I+1)
         IF(MR.EQ.1) GO TO14
          DO15 J=2,MR
          M=LQ+J
          B(I)=B(I)+A3(I,J)*X0(M)
          B(M)=B(M)+A3(I,J)*X0(I)
 15      CONTINUE
 13      CONTINUE
 14      CONTINUE
```

```

         WRITE(6,16)
 16      FORMAT(5X,'   PRINT  OF  STIFFNESS TERM    '/)
       WRITE(6,3)(B(I),I=1,NEQ)

       ********   TOTAL RESISTING FORCE AT EACH PRINT INTERVAL ****

        DO 100 I=1,NEQ
100    B(I)=B(I)+R(I)

        WRITE(6,200)
200    FORMAT(5X,'   TOTAL  RESISTIVE  FORCE    ')

        WRITE(6,28) NC
        WRITE(6,3) (B(I),I=1,NEQ)


 28     FORMAT(5X,'   TIME STEP INTERVAL     = ',I5/)

       *************   ALSO  CHECK  DEFINITENESS OF  MATRIX FOR COLLAP


      IF(KKS.EQ.3) GO TO 310
       DEF=0.0
       DO 300 I=1,NEQ
       DEF=DEF+XU(I)*B(I)
300    CONTINUE
       WORK=0.0
       DO 700 I=1,NEQ
700    WORK=WORK+GB(I)*XU(I)




       DO 350 I=1,NEQ
       B(I)=1.0
       R(I)=0.0
350   CONTINUE

       DO 355 I=1,NEQ
       R(I)=R(I)+A3(I,1)*AU(I)
355    CONTINUE
       DO 360 I=1,NEQ
       LQ=I-1
       MR=MIN0(MBAND,NEQ-I+1)
       IF(MR.EQ.1) GO TO 365
       DO 370 J=2,MR
       M=LQ+J
       R(I)=R(I)+A3(I,J)*AU(M)
       R(M)=R(M)+A3(I,J)*AU(I)
370     CONTINUE
360    CONTINUE
365    CONTINUE
      SUM=0.0
       DO 375  I=1,NEQ
      SUM=SUM+R(I)*AU(I)
375   CONTINUE

       IF(KKS.NE.3)  GO TO 119
```

```
        IF(SUM.EQ.0.0) GO TO 115
       CALL STIFF1(A3,NEG,MBAND,A5)

      JOBN=10
      CALL EIGRS(A5,NEG,JOBN,R,B,NEG,A0,IER)

      WRITE(6,110) (R(I),I=1,NEG)
      WRITE(6,111) IER
110   FORMAT(12E10.3)
111   FORMAT(5X,I5)
      WORK=R(1)

119    CONTINUE
       IF(WORK.EQ.0.0) GO TO 400

        RATIO=DEF/WORK
115     CONTINUE
       IF(SUM.GT.0.0) GO TO 400
       WRITE(6,500) NC,SUM

         STOP
400     CONTINUE

         WRITE(6,600) NC,DEF,WORK,RATIO,SUM
600       FORMAT(5X,'  *****     TIME STEP  (NC)    = ',I5/
   *                5X,' *****     STIFFNESS  (DEF)   = ',E10.3/
   *                5X,' *****     WORK  TERM (WORK) = ',E10.3/
   *    5X,'          RATIO            = ',E10.3/
   *    5X,'          DEFINITENESS  = ',E10.3/)
500     FORMAT(5X,'       TIME  STEP   NC      = ',I5/
   *    5X,'             VALUE OF MATRIX A3 = ',E10.3/)
310     CONTINUE
          RETURN
          END
```

```
           DATA SET ESFDMX      AT LEVEL 025 AS OF 83/03/05
      SUBROUTINE ESFDMX(VOL,DENS,DENF,KS,KF,DBT1,DBT2,F,NUMNP,NEL,ID,IH,

     *IMAT)

      IMPLICIT REAL*8 (A-H,O-Z)

      DIMENSION  ID(NUMNP,1),X(1),Y(1),ELMDA(1),EMU(1),EALFA(1),EM(1),
     *EK(1),ROS(1),ROF(1),NP(4),INP(4),TTT(2),SSS(2),IS(8),IL(8),IH(NUMN
     *P,1)




      COMMON/CONTL/DT,HED(8),KADN,NASIM,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS
           COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT

      COMMON/EMB/S(16,16),GM(16),GH(16,16),SS(16,16),XP(16),XN(
     *16),LM(16)
      COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),
     *ST(20,12),D(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8),QDD(4,4)

      COMMON/ASHK/C1,C2,C3,AL,ALF,FACC,FACT

      DIMENSION LK(8),LH(8),FK(1),DBT1(1),DBT2(1)



      DATA  IS/1,2,5,6,9,10,13,14/,IL/3,4,7,8,11,12,15,16/


      FORM  ELEMENT  LOCAL  STIFFNESS MASS DAMPING  MATRICES  TAKING  *1
      DEGREES  OF  FREEDOM
      K=IMAT
      WRITE(6,60)K,IMAT,LOOP,DBT1(K),DBT2(K)
   60 FORMAT(5X,3I5,2E10.3)

          VLD=VOL/4.

      DO  626  I=1,8
      NS=IS(I)
      NF=IL(I)
      IF(KS.EQ.-1.AND.KF.EQ.1)  GO  TO  656
          GM(NS)=VLD*DENS
      IF(KS.EQ.1.AND.KF.EQ.-1)  GO  TO  658
  656     GM(NF)=VLD*DENF
  658 CONTINUE


      DO  625  J=1,8
      MS=IS(J)
      MF=IL(J)

     ***************************************************************
      IF(KS.EQ.1.AND.KF.EQ.-1) GO  TO  616
      IF(KS.EQ.0.AND.KF.EQ.0)  GO  TO  616
      IF(KS.EQ.-1.AND.KF.EQ.1)  GO  TO  619
     ***************************************************************
```

```
 616 S(NS,MS)=SF(I,J)

          IF(LOOP.NE.1) GO TO 660

     GM(NS,MS)=DBT1(K)*(GM(MS)-(F**2.)*GM(MF))+DBT2(K)*(SF(I,J)-E(I,J)
    **ALF)

           GO TO 665
 660         CONTINUE

             GM(NS,MS)=DBT1(NEL)*(GM(MS)-(F**2.)*GM(MF))+DBT2(NEL)*(SF(I,
    *J)-E(I,J)*ALF)

 665            CONTINUE
     IF(KS.EQ.1.AND.KF.EQ.-1)  GO  TO  625

 618 S(NS,MF)=C(I,J)
     S(NF,MS)=C(J,I)
 619 GM(NF,MF)=H(I,J)
 617 S(NF,MF)=E(I,J)


       SS(NS,MS)=SF(I,J)

 625 CONTINUE
 626 CONTINUE

**************************************************************************

 999 FORMAT(//)

     IF(NC.GT.0.AND.KKS.EQ.3)  GO  TO  8200
     IF(NEL.NE.0)  GO  TO  8200
     PRINT  999
     WRITE(6,9511)
9511 FORMAT(1H1/'        ******   CONNECTIVITY  ARRAY   ******  '/)
     WRITE(6,220)  (LM(I),I=1,16)
 220 FORMAT(10X,16I5)
     PRINT  999
     WRITE(6,9513)
9513 FORMAT(5X,' ELEMANT  UNDRAINED  STIFFNESS  MATRIX')
     WRITE(6,221)  ((S(I,J),J=1,16),I=1,16)
     PRINT  999
     WRITE(6,9514)
9514 FORMAT(5X,'ELEMENT  MASS  MATRIX')
     WRITE(6,224)  (GM(I),I=1,16)
 224 FORMAT(3 X,16E8.1)
```

```
      PRINT  999
      WRITE(6,9515)
 9515 FORMAT(5X,'ELEMENT  DISSIPIATION  MATRIX')
      WRITE(6,221)((GH (I,J),J=1,16),I=1,16)
  221 FORMAT(3X,16F8.1)
      PRINT  999
      WRITE(6,9517)
 9517 FORMAT(5X,'ELEMENT  UNDRAINED  STIFFNESS MATRIX  ASSUMING  FLUID
     *HAS NO DISPLACEMENT')
      WRITE(6,221)((SS (I,J),J=1,16),I=1,16)
      PRINT  999
      WRITE(6,9518)
 9518 FORMAT(5X,' ELEMENT  INITIAL  STRESS  LOAD  FACTOR  ')
      WRITE(6,221) (XP(I),I=1,16)
      PRINT  999
      WRITE(6,9519)
 9519 FORMAT(5X,' ELEMENT  SELF  WT.  LOAD  VECTOR  ')
      WRITE(6,221) (XN(I),I=1,16)
 8200 CONTINUE
      RETURN
      END
```

```
          DATA SET ESFMTX     AT LEVEL 042 AS OF 83/02/11
      SUBROUTINE ESFMTX(XKK,IPLMAX,IKS,IKF,PERM,TTX0,TTX1,TTX2,NUMNP,   00001**4
                                                                        00002*13
     * THICK,NEL,WA,PRESS,PS,KS,KF,COEFS,COEFK,FACN)                    00003*32
                                                                        00004*13
      IMPLICIT REAL*8(A-H,O-Z)                                          00005
                                                                        00006*13
      COMMON/QUAD/RR(4),ZZ(4),FAC,R,W(8),P(4,12),SF(12,12),H(12,12),    00007*10
     *ST(20,12),D(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8)               00008*10
         COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK01700009*42
                                                                        00010*42
      COMMON/ASHK/C1,C2,C3,AL,ALF,FACC,FACT                             00011
      COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT                00012*22
                                                                        00013*24
         COMMON/RANI/RM(5),ZM(5),IPT                                    00014*40
                                                                        00015*40
                                                                        00016*33
                                                                        00017*35
      COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00018*35
                                                                        00019*31
                                                                        00020*31
         COMMON/GORA/IFLAG,ITR                                          00021*42
                                                                        00022*42
                                                                        00023*31
      DIMENSION X(1),Y(1),ELMDA(1),EMU(1),EALFA(1),EM(1),               00024**5
                                                                        00025*20
                                                                        00026*20
                                                                        00027*20
     *EK(1),ROS(1),ROF(1),NP(4),INP(4),TTT(2),SSS(2),IS(8),IL(8)        00028**5
                                                                        00029*13
                                                                        00030
                                                                        00031
      DIMENSION TT1(12),TT2(12),TTX1(2,12),TTX0(12,2),TTX2(12,12)       00032*10
      DIMENSION LK(8),LH(8)                                             00033**4
                                                                        00034
                                                                        00035
                                                                        00036*20
      DIMENSION  PRESS(5)                                               00037*26
      DIMENSION  WA(IDWA,1)                                             00038*20
      DIMENSION  PS(16,8)                                               00039*28
      DIMENSION  FACN(4)                                                00040*32
                                                                        00041*28
                                                                        00042*28
      DATA SSS/-1.0D0,1.0D0/,TTT/-1.0D0,1.0D0/                          00043*12
************************************************************************00044
                                                                        00045
************************************************************************00046
      CALL  FORMB  AND  CREATE  ELEMENT  STIFFNESSES  BY  GAUSS  INTEGRA00047
      FOR  FOUR  NODED  ISOPARAMETRIC  ELEMENT  **MODIFY  FOR  EIGHT  NO00048
                                                                        00049
                                                                        00050
                                                                        00051*18
                                                                        00052*18
      DO  510  J=1,4                                                    00053**6
      DO  510  I=1,8                                                    00054**6
  510 P1(J,I)=0.0                                                       00055**6
                                                                        00056
      DO  520  I=1,16                                                   00057*36
      DO  520  J=1,8                                                    00058*36
  520 PS(I,J)=0.0                                                       00059*36
```

```


      KS=IKS
      KF=IKF
      DO  500 LR=1,NINT
      SX=SSS(LR)*XKK
      DO  500LZ=1,NINT
      TX=ITT(LZ)*XKK
      CALL FORMB(SX,TX,IPLNAX,THICK)



      IPT=(LR-1)*NINT+LZ

      RM(IPT)=RM(IPT)+Q(1)*RR(1)+Q(2)*RR(2)+Q(3)*RR(3)+Q(4)*RR(4)
      ZM(IPT)=ZM(IPT)+Q(1)*ZZ(1)+Q(2)*ZZ(2)+Q(3)*ZZ(3)+Q(4)*ZZ(4)



      IF(NC.EQ.0)  GO  TO  100
      CALL MODCNV(NEL,IPT,D)


  100 CONTINUE

       WRITE(6,101)
  101 FORMAT(1H1/5X,' PRINT IN ESFMTX.FOR ')
       WRITE(6,102) ((D(I,J),J=1,4),I=1,4)
  102 FORMAT(5X,4E10.3)

      ****************************************************************
      IF(KS.EQ.0.AND.KF.EQ.0)  GO  TO  86
      IF(KS.EQ.1.AND.KF.EQ.-1) GO  TO  86
      IF(KS.EQ.-1.AND.KF.EQ.1)  GO  TO  95
      ****************************************************************


      **********    REPLACE  C1,C2,C3,  ***   BY  D(I,J)    FOR  PLANE  S
      **********   E L A S T O P L A S T I C A N A L Y S I S   WE  NEED
       MATRICES  ANYWAY  ***************************



   86 DO  90  I=1,ND

      *******   CHANGED  FOR  GENERAL  PURPOSES   **********************


      D1=(D(1,1)*P(1,I)+D(1,2)*P(2,I)+D(1,3)*P(3,I)+D(1,4)*P(4,I))*FAC
      D2=(D(2,1)*P(1,I)+D(2,2)*P(2,I)+D(2,3)*P(3,I)+D(2,4)*P(4,I))*FAC
      D3=(D(3,1)*P(1,I)+D(3,2)*P(2,I)+D(3,3)*P(3,I)+D(3,4)*P(4,I))*FAC
      D4=(D(4,1)*P(1,I)+D(4,2)*P(2,I)+D(4,3)*P(3,I)+D(4,4)*P(4,I))*FAC



      DO  90  J=I,ND
      SF(J,I)=SF(J,I)+D1*P(1,J)+D2*P(2,J)+D3*P(3,J)+D4*P(4,J)
   90 SF(I,J)=SF(J,I)
      ****************************************************************

```

```
                                                                        00116*13
                                                                        00117*13
      IF(KS.EQ.1.AND.KF.EQ.-1)  GO  TO  5000                            00118**6
   95 FACK=FAC/PERM                                                     00119
                                                                        00120**9
                                                                        00121**9
                                                                        00122*11
      *******  PORTION  REMOVED  FROM  ORIGINAL  VERSION    ************00123*11
                                                                        00124*11
                                                                        00125*11
                                                                        00126*11
                                                                        00127**9
                                                                        00128**9
                                                                        00129*13
                                                                        00130*13
      IF(NELTYP.EQ.0)  NN=4                                             00131*13
      IF(NELTYP.EQ.1)  NN=6                                             00132*13
                                                                        00133*13
                                                                        00134*13
      IF(NDRN.EQ.0)  GO  TO  5000                                       00135*17
      ****  FORM  FLUID  DISSIPATION  MATRIX    ************************00136*13
                                                                        00137*13
                                                                        00138*13
      II=1                                                              00139
      JJ=2                                                              00140
      L=0                                                               00141
                                                                        00142*14
      ********   SHAPE  FUNCTION    *********                           00143*14
      DO  2100  I=1,NN                                                  00144*14
      DO  2101 J=1,2                                                    00145
      L=L+1                                                             00146
      TT1(L)=0.0                                                        00147
      TT2(L)=0.0                                                        00148
       IF(L.EQ.II)  TT1(L)=Q(I)                                         00149
      IF(L.EQ.JJ)  TT2(L)=Q(I)                                          00150
 2101 CONTINUE                                                          00151
      II=II+2                                                           00152
      JJ=JJ+2                                                           00153
 2100 CONTINUE                                                          00154
                                                                        00155*14
                                                                        00156*14
      DO  2010  J=1,ND                                                  00157*10
 2010 TTX1(I,J)=TT1(J)                                                  00158
      DO  2020  J=1,ND                                                  00159*10
 2020  TTX1(2,J)=TT2(J)                                                 00160
                                                                        00161*14
                                                                        00162*14
      *********   PERMIABILITY  MATRIX  FORMATION    *******************00163*14
                                                                        00164*14
      DO  205  I=1,2                                                    00165
       DO  205  J=1,ND                                                  00166*10
  205  TTX0(J,I)=TTX1(I,J)                                              00167
      DO  206  I=1,ND                                                   00168*10
      DO  206  J=1,ND                                                   00169*10
      DO  207  K=1,2                                                    00170
  207 TTX2(I,J)=TTX2(I,J)+TTX0(I,K)*TTX1(K,J)                           00171
      H(I,J)=H(I,J)+TTX2(I,J)*FACK                                      00172
  206 CONTINUE                                                          00173
                                                                        00174**9
                                                                        00175**9
```

```
5000 CONTINUE                                                           00176**9
************************************************************************00177
                                                                        00178*20
                                                                        00179*20
     ******   CALCULATION  OF  RESIDUAL  LOAD VECTOR  AT  GAUSS  POINT  00180*20
     ****   ELASTOPLASTIC  ANALYSIS                                     00181*20
                                                                        00182*20
                                                                        00183*39
                                                                        00184*20
     CALL LOAD( NEL,PRESS,VOL,WA)                                       00185*20
                                                                        00186*20
                                                                        00187*29
                                                                        00188*29
     DO  700  I=1,ND                                                    00189*29
     DO  700 JJ=1,4                                                     00190*20
     J=JJ+(IPT-1)*4                                                     00191*20
     PS(J,I)=P(JJ,I)                                                    00192*20
 700 CONTINUE                                                           00193*20
                                                                        00194*32
     FACN(IPT)=FAC                                                      00195*32
                                                                        00196*20
                                                                        00197*20
                                                                        00198*20
 600 CONTINUE                                                           00199*26
                                                                        00200*30
                                                                        00201*30
                                                                        00202*31
                                                                        00203*31
     IF(NGAMA.EQ.0)  GO  TO  500                                        00204*31
     **********    CALLL  SELFWT  SUBROUTINE   *************************00205*30
                                                                        00206*32
     IF(NC)  500,550,500                                                00207*32
 550 CONTINUE                                                           00208*32
                                                                        00209*30
                                                                        00210*41
            IF(IFLAG.NE.0)   GO TO 500                                  00211*41
                                                                        00212*41
     CALL SELFLD (NEL,RS,RF,COEFS,COEFR,VOL)                            00213*30
                                                                        00214*30
                                                                        00215*30
 500 CONTINUE                                                           00216**8
                                                                        00217
      WRITE(6,701)
 701 FORMAT('PRINT INESFMTX STRAIN MATRIX VOLUME AT GAUSS POINTS ')
      WRITE(6,702) ((PS(I,J),J=1,8),I=1,16)
      WRITE(6,702) (FACN(L),L=1,4)
 702 FORMAT(5X,8E10.3)
                                                                        00218*16
                                                                        00219*16
                                                                        00220*16
     RETURN                                                             00221
     END                                                                00222
```

```
      DATA SET IDRUCK     AT LEVEL 005 AS OF 82/03/31
    SUBROUTINE IDRUCK(WA,IWA,PROP,NPT,J)
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

    DIMENSION  WA(IDW,1),PROP(NCON)



    SET INITIAL STRESSES AND STRAINS EQUAL TO ZERO

    DO 15 I=1,8
    WA(I,J)=0.0
 15 CONTINUE

    ANGLE=PROP(3)    AND    COHES=PROP(4)

    WA( 9,J)=PROP(3)/57.296
    WA(10,J)=PROP(4)

    SET INITIAL STRESS STATE TO  -ELASTIC-

    IWA=1

 25 CONTINUE


    RETURN

    END
```

```
        DATA SET IELPAL     AT LEVEL 006 AS OF 82/03/31
    SUBROUTINE IFLPAL(WA,IWA,PROP,NPT,J)                                  00001
    IMPLICIT REAL*8(A-H,O-Z)                                              00002
                                                                          00003
    COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017       00004          00004**2
    DIMENSION WA(IDW,1),PROP(NCON)                                        00005**2
                                                                          00006
    SET INITIAL STRESSES AND STRAINS TO ZERO                              00007
    SET INITIAL YIELD POINT TO PROP(3)                                    00008
    SET INITIAL STRESS STATE TO *ELASTIC*                                 00009

       WRITE(6,160)
160    FORMAT(5X,' ***   PRINT IN IELPAL *** ')C
       WRITE(6,161)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
161    FORMAT(5X,10I5)
                                                                          00010
                                                                          00011
    DO  15  I=1,8                                                         00012**2
 15 WA(I,J)=0.0                                                           00013**2
    WA(9,J)=PROP(3)                                                       00014
    IWA=1                                                                 00015
                                                                          00016
 25 CONTINUE                                                              00017**2
                                                                          00018**5
    RETURN                                                                00019
    END                                                                   00020
```

```
         DATA SET INFMTX      AT LEVEL 056 AS OF 82/07/08
      SUBROUTINE INFMTX(S,C,CT,AN,ANIT,AS,SI,SM,SN,SP,SD,SQ,NPJ,NPK,      00001**4
                                                                          00002*28
     *NEQS,A,CE,PSOL,DSOL,RW,IW,AM,EK,KK,IH,COPI,ICOLMS,IDES,ROW,        00003*54
                                                                          00004*55
     *GB,SO,PPSOL,DDSOL,SB,XP)                                            00005*54
                                                                          00006*28
      IMPLICIT REAL*8(A-H,O-Z)                                            00007
                                                                          00008*28
                                                                          00009*28
      DIMENSION S(NPJ,1),C(NPJ,1),CT(NPJ,1),SI(NEQS),AS(NEQS,1),SM(NEQS)00010*43
                                                                          00011*11
     *,SN(1),SP(1),SQ(1),SD(NPJ,1)                                        00012*26
      DIMENSION AN(NPJ,1),ANIT(NPJ,1)                                     00013*26
                                                                          00014
                                                                          00015*13
                                                                          00016*13
      COMMON/ASIM/ NYP,NSC,NSLC,NLC,NNYP,NNSC                             00017*28
      COMMON/CONTL/DD,HED(8),RADN,NUMNP,NUMEL                             00018*28
                                                                          00019*32
      COMMON/TARA/LPC,N,M1,M2,NPT,KPT,NPRNT,M,MM                          00020*56
                                                                          00021*13
      COMMON/BODHO/NDIV,NNOD,JK(20,3)                                     00022*29
                                                                          00023*13
      DIMENSION A(LPC,1),PSOL(1),DSOL(1),RW(LPC),IW(1),AM(1),EK(1)        00024*13
      DIMENSION CE(1)                                                     00025*17
      DIMENSION  PPSOL(1),DDSOL(1),SB(1),IH(NUMNP,1)                      00026*30
      DIMENSION GB(1),SO(1),COPI(NPK,NPK),ICOLMS(1),IDES(1),ROW(1)        00027*54
                                                                          00028*13
                                                                          00029
                                                                          00030*54
      DIMENSION XP(NPK,1)                                                 00031*54
  999 FORMAT(//)                                                          00032**8
                                                                          00033*11
      ******  INITIALIZE                                                  00034**5
      DO 10  K=1,NEQS                                                     00035**5
      SM(K)=0.0                                                           00036*11
   10 SI(K)=0.0                                                           00037**5
      DO  20  J=1,NPJ                                                     00038**5
      SN(J)=0.0                                                           00039**5
      SP(J)=0.0                                                           00040**5
   20 CONTINUE                                                            00041**5
                                                                          00047*19
      MN=(NPK+1)+2                                                        00048*19
      IF(NPT.EQ.1)  MN=NPK+NEQS +1+2                                      00049*32
      IF(NPT.EQ.2)  MN=NPK+NEQS+1+2                                       00050*45
                                                                          00051*45

      IF(KPT.EQ.1) MN=MM
      DO  40  I=1,MN                                                      00052*19
   40 SQ(I)=0.0                                                           00053*19
                                                                          00054*11
      ****  INITIALIZE  ***                                               00055*13
      DO  650  I=1,LPC                                                    00056*13
      DO  660  J=1,N                                                      00057*13
  660 A(I,J)=0.0                                                          00058*13
  650 CONTINUE                                                            00059*13
                                                                          00060*11
                                                                          00061*13
                                                                          00063*32
```

```
                                                                      00064*32
     ********   BRANCH  HERE  FOR  STRESS  OPTIMIZATION W. R. T.  RESI 00065*32
   *  STRESSES     **************************                          00066*33
                                                                      00067*32
      IF(KPT.EQ.2.AND.NPT.EQ.1.OR.KPT.EQ.2.AND.NPT.EQ.2) GO TO 480
      IF(KPT.EQ.1.AND.NPT.EQ.2.OR.KPT.EQ.1.AND.NPT.EQ.1) GO TO 495

      DO  50  J=1,NPJ                                                 00042*27
      DO  30  K=1,NPK                                                 00043**5
      SD(J,K)=0.0                                                     00044*27
   30 CONTINUE                                                        00045*12
   50 CONTINUE                                                        00046*27
                                                                      00070*44
                                                                      00071*44
     ******  COMPUTATION  OF   ' SC '  MATRIX  ************************ 00072**3
      PRINT 999                                                       00073**8
      DO  100  I=1,NPJ                                                00074
      DO  200  K=1,NEQS                                               00075
      SUM=0.0                                                         00076
      DO  300  J=1,NPJ                                                00077
  300 SUM=SUM+S(I,J)*C(J,K)                                           00078*11
  200 SI(K)=SUM                                                       00079
                                                                      00080
                                                                      00081
                                                                      00082
                                                                      00083
                                                                      00084
                                                                      00085
                                                                      00086*11
                                                                      00087*11
     *****  COMPUTE  ' S * C* (K) * -1   MATRIX   (FIRST ROW) *********00088*11
      DO  400 K=1,NEQS                                                00089*11
                                                                      00090*17
                                                                      00091*17
      SUM=0.0                                                         00092*17
      DO  500 L=1,NEQS                                                00093*11
  500 SUM=SUM+SI(L)*AS(L,K)                                           00094*11
      SM(K)=SUM                                                       00095*11
  400 CONTINUE                                                        00096*11
                                                                      00097*11
                                                                      00098*11
     ****  COMPUTE ' S * C *K * -1 CT  '  MATRIX ( FIRST ROW )  ****** 00099*11
      DO  600 J=1,NPJ                                                 00100*11
      SUM=0.0                                                         00101*11
      DO  700 K=1,NEQS                                                00102*11
  700 SUM=SUM+SM(K)*CT(J,K)                                           00103*43
  600 SN(J)=SUM                                                       00104*11
                                                                      00105*11
                                                                      00106*11
     ****  COMPUTE ' C *S K*-1 *CT *S '  MATRIX  ************          00107*11
      DO  800 J=1,NPJ                                                 00108*11
      SUM=0.0                                                         00109*11
      DO  900 L=1,NPJ                                                 00110*11
  900 SUM=SUM+SN(L)*S(L,J)                                            00111*19
  800 SP(J)=SUM                                                       00112*11
     *****  COMPUTE  ' S* C* 5* -1 *CT *S  -  > '  MATRIX ( Z- MATRIX )00113*11
      DO  1000 J=1,NPJ                                                00114*11
 1000 SP(J)=SP(J)-S(I,J)                                              00115*11
                                                                      00116*11
```

```
      DO 1010 J=1,NPJ
       ANIT(1,J)=SP(J)
 1010 CONTINUE

                                                                        00117*11
                                                                        00118*13
      *******   TEST  FOR  SEMIDEFFINITENESS  OF  MATRIX  Z  ***********00119*13
                                                                        00120*13
       SUM=0.0                                                           00121*13
       DO  150  J=1,NPJ                                                  00122*14
  150  SUM=SUM+SP(J)                                                     00123*13
       SO(1)=SUM                                                         00124*13
                                                                        00125*13
                                                                        00126*13
                                                                        00127
                                                                        00128
                                                                        00129
      *******  COMPUTATION  OF  '  Z * N '  MATRIX  ****************    00130**3
      PRINT  999                                                        00131**8
      DO  1900  K=1,NPK                                                 00132
      SUM=0.0                                                           00133
      DO  1700  J=1,NPJ                                                 00134
 1700 SUM=SUM+SP(J)*AN(J,K)                                             00135*26
 1900 SO(1,K)=SUM                                                       00136*26
  100 CONTINUE                                                          00137*11
                                                                        00138*13


       WRITE(6,254)
  254 FORMAT(1H1/5X,'   PRINT OF Z MATRIZ '/)
       WRITE(6,281) ((ANIT(I,J),J=1,NPJ),I=1,NPJ)
  281 FORMAT(5X,12E10.3)

      JOBN=10

      CALL EIGRS(ANIT,NPJ,JOBN,RW,XP,NPK,SO,IER)
      WRITE(6,255)  (RW(I),I=1,NPJ)
      WRITF(6,256) IER
  256 FORMAT(5X,I5)


                                                                        00139*13
       SUM=0.0                                                           00140*13
       DO  160  J=1,NPJ                                                  00141*13
  160  SUM=SUM+SO(J)                                                     00142*13
       IF(SUM.LT.0.0)  GO  TO  180                                       00143*13
       WRITE(6,190)                                                      00144*13
       STOP                                                              00145*13
   15 CONTINUE                                                          00146*41
                                                                        00147*13
                                                                        00148*13

      ********   INPUT  VALUE OF HARD ********
                                                                        00150
      ******  COMPUTATION OF  '  N ** T *Z *N '  NATRIX  ***************00151**3
                                                                        00152*13
  180 CONTINUE                                                          00153*13
```

```
      HARD=0.0
                                                                  00154*11
      DO   4000 I=1,NPK                                           00155*11
      DO   5000 J=1,NPK                                           00156*11
      SUM=0.0                                                     00157*11
      SUM1=0.0

       DO 1200 K=1,NPJ
 1200  SUM1=SUM1+AN(K,I)*AN(K,J)
       SUM1=HARD*SUM1


      DO   1100 K=1,NPJ                                           00158*11
 1100 SUM=SUM+AN(K,I)*SD(K,J)                                     00159*26

      IF(KPT.EQ.1) GO TO 60

      A(I,J+1)= SUM1-SUM                                          00160*22

      GO TO 5000

   60 CONTINUE
      A(I+2,J)=(SUM1-SUM)

 5000 CONTINUE
 4000 CONTINUE                                                    00161*11
                                                                  00162*11
                                                                  00163
                                                                  00164*13
                                                                  00165*13

      DO   220  I=1,NPK                                           00167*13

      IF(KPT.EQ.1) GO TO 222

      DO   230  J=2,N                                             00168*22
  230 COPI(I,J-1)=A(I,J)                                          00169*13

       GO TO 224

  222  CONTINUE

       DO 223 J=1,NPK
  223  COPI(I,J)=A(I+2,J)

  224  CONTINUE

  220 CONTINUE                                                    00170*13

      JOBN=10
      CALL EIGRS(COPI,NPK,JOBN,RW,XP,NPK,SQ,IER)
      WRITE(6,255)   (RW(I),I=1,NPK)
  255 FORMAT(5X,12E10.3)
      WRITE(6,256) IER
```

```
      DO 253 I=1,MM
      DO 253 J=1,MN
253   COPI(I,J)=0.0

        DO 257 I=1,MN
257     SQ(I)=0.0


      TOL=0.100E-32

      DO 242 I=1,NPK
      IF(RW(I)-TOL) 288,289,289
288   CONTINUE
      WRITE(6,250)I,RW(I),TOL                                             00178*19
      STOP                                                                00180*15
289 CONTINUE
242 CONTINUE

                                                                          00175*13
                                                                          00177*15
250 FORMAT(//'  MATRIX  A  IS  NEGATIVE SEMIDEFINITE  '/                  00179*15
   *         '  ELEMENT NO.                          = ',/
   *         '  EIGENVALUE(I)                        = ',E10.3/
   *         '  TOL                                  = ',E10.3/)
290 CONTINUE                                                              00181*15
    WRITE(6,260)                                                          00182*19
260 FORMAT(//'  MATRIX  A  IS  POSITIVE  SEMIDEFINITE  ')                 00183*15
    GO  TO  275                                                           00184*20
295 CONTINUE                                                              00185*15
    WRITE(6,270)                                                          00186*19
270 FORMAT(// '  MATRIX  A   IS  POSITIVE  DEFINITE  ')                   00187*15
                                                                          00188*13
                                                                          00189*13
                                                                          00190*13
275 CONTINUE                                                              00191*19

    IF(KPT.EQ.1.AND.NPT.EQ.0) GO TO 495


    DO  455  I=1,N                                                        00192*13
455 CE(I)=0.0                                                             00193*19
    CE(1)=1.0                                                             00194*19
                                                                          00195*21
                                                                          00196*21
    DO  451 I=1,NPK                                                       00197*13
451 A(I,1)=AM(I)                                                          00198*13
                                                                          00199*23
                                                                          00200*23
    DO  460 I=1,NPK                                                       00201*23
    DO  450  J=2,N                                                        00202*23
450 A(I,J)=-A(I,J)                                                        00203*23
460 CONTINUE                                                              00204*23
                                                                          00205*32
        |                                                                 00206*32
                                                                          00207*34
```

```
      GO  TO    490                                                     00208*34
   **********   LINKED  HERE  THE  BRANCHING  OF  RESIDUAL  STRESSS OP00209*32
    ***************************                                         00210*32
                                                                        00211*32
  480 CONTINUE                                                          00212*33
      NPP=NPJ+1                                                         00213*32
      NPO=NPP+1                                                         00214*32
      NPD=N                                                             00215*48
      DO  457  I=1,NPK                                                  00216*32
      DO  462  J=2,NPP                                                  00217*33
  462 A(I,J)=AN(J-1,I)                                                  00218*33
      DO  456  J=NPO,NPD                                                00219*32
  456 A(I,J)=-AN(J-NPP,I)                                               00220*32
  457 CONTINUE                                                          00221*32
                                                                        00222*48
                                                                        00223*48
                                                                        00224*48
                                                                        00225*48
      NH=NPK+1                                                          00226*32
      NH1=NPK+NEQS                                                      00227*32
      DO  458  I=NH,NH1                                                 00228*32
      DO  459  J=2,NPP                                                  00229*32
  459 A(I,J)=CT(J-1,I-NPK)                                              00230*49
                                                                        00231*34
                                                                        00232*34
      DO  465  J=NPO,NPD                                                00233*32
  465 A(I,J)=-CT(J-NPP,I-NPK)                                           00234*49
                                                                        00235*48
                                                                        00236*34
                                                                        00237*34
  458 CONTINUE                                                          00238*35
      ******   PRINT  OF  MATRICES  BY  PARTS   ********************** 00239*34
                                                                        00240*34
                                                                        00241*36
                                                                        00242*36
      DO  485  I=1,N                                                    00243*36
  485 CE(I)=0.0                                                         00244*36
      CE(1)=1.0                                                         00245*36
                                                                        00246*36
      IF(NPT.EQ.2)  GO  TO  488                                         00247*45
                                                                        00248*44
                                                                        00249*36
      DO  487  I=1,NPK                                                  00250*36
  487 A(I,1)=AM(I)                                                      00251*36
                                                                        00252*34
                                                                        00253*32
                                                                        00254*32
   72 FORMAT(5X,12E10.3)                                                00255*32
                                                                        00256*34
                                                                        00257*34
                                                                        00258*34
  490 CONTINUE                                                          00259*35
      DO  452  I=1,NPK                                                  00260*13
  452 SQ(I)=EK(I)                                                       00261*13
                                                                        00262*44
      GO  TO  495                                                       00263*44
```

```
                                                                      00264*44
  488 CONTINUE                                                        00265*44
                                                                      00266*44
                                                                      00267*44
      NPE=NPK+1                                                       00268*44
      NPG=NPK+NEQS                                                    00269*44
                                                                      00270*44
      DO  489 I=NPE,NPG                                               00271*44
  489 A(I,I)=-GB(I-NPK)                                               00272*54
                                                                      00273*44
      DO  498 I=1,NPK                                                 00274*44
  498 SQ(I)=EK(I)                                                     00275*44
      DO  499 I=NPE,NPG                                               00276*44
  499 SQ(I)=SQ(I-NPK)                                                 00277*50
                                                                      00278*44
                                                                      00279*44
                                                                      00280*44
  495 CONTINUE                                                        00281*44
                                                                      00282*34
                                                                      00283*34
                                                                      00284*51
                                                                      00285*51
      WRITE(6,195)                                                    00286*51
  195 FORMAT(1H1/'  ****   PRINT  OF  VECTOR   MB  *** ')             00287*51
      WRITE(6,196) (GB(I),I=1,NEQS)                                   00288*54
  196 FORMAT(5X,12E10.3)                                              00289*55
      WRITE(6,197)                                                    00290*51
  197 FORMAT(1H1/'  **  PRINT  OF  VECTOR  SQ  ** ')                  00291*51
      WRITE(6,196) (SQ(I),I=1,NEQS)                                   00292*52
      WRITE(6,198)                                                    00293*51
  198 FORMAT(1H1'   ***  PRINT  OF  DIMENSION  FOR  PPTIMIZATION  *** ')00294*51
      WRITE(6,199) NPC,N,M1,M2                                        00295*51
  199 FORMAT(5X,4I5)                                                  00296*51
                                                                      00297*56
                                                                      00298*56
                                                                      00299*56
      IF(NPRNT.EQ.0)  GO  TO  10000                                   00300*56
      WRITE(6,9990)                                                   00301*56
 9990 FORMAT(1H1/'  ***  PRINT  OF   M-VECTOR  ****  ')               00302*56
      WRITE(6,9991) (A(I,I),I=1,NPK)                                  00303*56
 9991 FORMAT(1X,12E10.3)                                              00304*56
      WRITE (6,9992)                                                  00305*56
 9992 FORMAT(1H1/'  ***  PRINT  OF  N-TRANSPOSE  *** ')               00306*56
      WRITE(6,9991) ((A(I,J),J=2,NPP),I=1,NPK)                        00307*56
      WRITE(6,9993)                                                   00308*56
 9993 FORMAT(1H1/'  ****  PRINT  OF -N - TRANSPOSE  *** ')            00309*56
      WRITE(6,9991) ((A(I,J),J=NPU,NPD),I=1,NPK)                      00310*56
      WRITE(6,9994)                                                   00311*56
 9994 FORMAT(1H1/'  ****  PRINT  OF  B - TRANSPOSE  *** ')            00312*56
      WRITE(6,9991) ((A(I,J),J=2,NPP),I=NH,NH1)                       00313*56
      WRITE(6,9995)                                                   00314*56
 9995 FORMAT(1H1/'  ****  PRINT  OF  - C - TRANSPOSE  *** ')          00315*56
      WRITE(6,9991) ((A(I,J),J=NPU,NPD),I=NH,NH1)                     00316*56
      WRITE(6,9996)                                                   00317*56
 9996 FORMAT(1H1/'  ***  PRINT  OF  LOAD  VECTOR  *** ')              00318*56
      WRITE(6,9991) (A(I,I),I=NH,NH1)                                 00319*56
      WRITE(6,9998)                                                   00320*56
 9998 FORMAT(1H1/'  ****  PRINT  OF  K  VECTOR  ** ')                 00321*56
      WRITE(6,9991) (SQ(I),I=1,NPG)                                   00322*56
      WRITE(6,9999)                                                   00323*56
```

```
 9999 FORMAT(1H1/' ****  PRINT  OF  OBJECTIVE  FUNCTION *** ')          00324*56
      WRITE(6,9991) (CE(I),I=1,N)                                       00325*56
```

```
0000 CONTINUE                                                          00326*56
                                                                       00327*56
                                                                       00328*56
     GO TO(80,70),KPT
 70  CONTINUE
                                                                       00329*56
    *************  CALL  OPTIMIZATION  ROUTINE   **************        00330*13
                                                                       00331*13
                                                                       00332*13
    CALL  ZX3LP(A,LPC,SQ,CE,N,M1,M2,SG,PSOL,DSOL,RW,IW,IER)            00333*19
                                                                       00334*31
                                                                       00335*31
                                                                       00336*40
    ********   TEST  FOR  SEMIDEFINITENESS                             00337
                                                                       00338
 190 FORMAT(//'   MATRIX  Z  (INFLUENCE)  IS  NOT  NEG  SEMIDEFINITE ')00339*13
                                                                       00340*21
    PR=0.0
     DO 191 I=1,NPK
 191 PR=PR+DSOL(I)*SQ(I)
     WRITE(6,192)
 192 FORMAT(1H1/'   CHECK  BY  MINIMIZATON  '/)
     WRITE(6,193) PR,(DSOL(I),I=1,NPK)
 193 FORMAT(5X,12E10.3)
                                                                       00341*21
     GO TO  90

 80  CONTINUE

     IF(NPT.EQ.0) GO TO 85

      DO 86 I=1,NPJ
      DO 87 J=1,NPK
 87   A((I+2),(J))=AN(I,J)
      DO 88 K=1,NEGS
      A((I+2),(NPK+K))=-CT(I,K)
      A((I+2),(K+NPK+NEGS))=CT(I,K)
 88   CONTINUE
 86   CONTINUE
 85   CONTINUE

    CALL ZXMIN(CE,A,AN,C,AM,EK,GB,SO,SQ,ICOLMS,IDES,ROW,COPI,PSOL,
   *DSOL,RW,IER,NPK,NPJ,NEGS,NUMEL,NYP,NSC,SD,SN,CT,IH,NUMNP)
     GO TO 95
                                                                       00342*28
                                                                       00343*28
 90  CONTINUE
```

```
      CALL OUTPUT(NUMEL,NYP,NSC,NPJ,NPK,PSOL,SG,AN,SD,SN,CT,SQ,NEQS,IH,
     *NUMNP)

   95 CONTINUE

      IF(KK.EQ.0)  GO  TO  1300
 1201 CONTINUE
      *********   READ  INFORMATION  ABOUT  FOR  WHUICH  BOUNDING  TO  B
      READ(5,1400) NDIV,NNOD
      READ(5,1400)  ((JK(I,J),J=1,3),I=1,NNOD)
 1400 FORMAT(5X,8I5)
      **************   PRINT  ALL  INFORMATION  ABOUT  NODE  TO BE  BOUN

      CALL  BOUND(NEQS,NPJ,NPK,SN,AS,C,SP,S,SD,SG,SQ,CE,PSOL,EK,A,M2,RW,

     *IW,LPC,DSOL,AN,MN,IH,PPSOL,DDSOL,SB,XP)

 1300 CONTINUE
      RETURN
      END
```

```
           DATA SET INIT       AT LEVEL 007 AS OF 83/02/16
      SUBROUTINE INIT(NUMEL,NUMNP,NEQ,PMAX,AMAX,TD, X0,X1,X2,A,B0,A0,VEL00001**2
                                                                        00002**3
     *,B)                                                               00003**2
      IMPLICIT  REAL*8  (A-H,O-Z)                                       00004
                                                                        00005**3
                                                                        00006**3
      COMMON/PRIM/ISTRES,MPRINT,JPRINT,KPRINT                           00007

      COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT
      COMMON/BENOY/SG(5,200,4,4),NCOND,NTENS


          DIMENSION TD(2),PMAX(NUMEL,1),AMAX(NUMNP,12,2),A0(1),B0(1),A(100008**6
     *),                                                                00009**7
                                                                        00010**6
     *VEL(1),B(1),X0(1),X1(1),X2(1),TS(2)                               00011
      COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA00012
     *X,ISTRPR,NPRINT,ISTART,NEQK,MBANDH                                00013
     *,NEQS,MBANDS                                                      00014**4
                                                                        00015**4
       COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX      00016**4
                                                                        00017**3
      REWIND 58
      REWIND 59
                                                                        00018**3
      DT=TETA*DD                                                        00019
      LLL=NDYN                                                          00020
      ISTRES= 0                                                         00021
      MPRINT=0                                                          00022
      JPRINT=0                                                          00023
      KPRINT=0                                                          00024
      TK=NCYCL*DD                                                       00025
                                                                        00026**3
                                                                        00027**3
      DO  401  I=1,NEQ                                                  00028
      X0(I)=0.0                                                         00029
      X1(I)=0.0                                                         00030
      X2(I)=0.0                                                         00031
  401 CONTINUE                                                          00032
                                                                        00033**3
      TS(1)=0.0                                                         00034
      TS(2)=0.0                                                         00035
      TD(1)=0.0                                                         00036
      TD(2)=0.0                                                         00037
                                                                        00038**3
                                                                        00039**3
      DO   5 I=1,NUMEL                                                  00040
      DO   5   J=1,18                                                   00041**3
    5 PMAX(I,J)=0.0                                                     00042
      DO   8   I=1,NUMNP                                                00043
          DO 8 K=1,2                                                    00044**6
          DO 8 J=1,12                                                   00045**5
                                                                        00046**5
                                                                        00047**4
    8 AMAX(I,J,K)=0.0                                                   00048
                                                                        00049**3
                                                                        00050**3
                                                                        00051**3
      DO  10  I=1,NEQ                                                   00052
```

```
     A(I)=0.                                                            00053
     B0(I)=0.                                                           00054
     A0(I)=0.                                                           00055
     VEL(I)=0.                                                          00056
  10 B(I)=0.                                                            00057
                                                                        00058**3
                                                                        00059**3
      **********   ALL  INITIALIZATION  TO  DYNAMIC  ANALYSIS  COMPLETED00060**3
                                                                        00061**3
      IF(KKS.NE.4) GO TO 400
      IF(NDYN.NE.1) GO TO 400
      IF(NTENS.EQ.1) GO TO 400
****** READ HERE PREVIOUS STORED DISPL. VELOCITY COMPONENTS
      READ(58,406) (X0(I),I=1,NEQ)
      READ(59,406) (X1(I),I=1,NEQ)

 405  FORMAT(5X,3I5)
 406  FORMAT(5X,12E10.3)
 400   CONTINUE
     RETURN                                                             00062
     END                                                                00063
```

```
         DATA SET INPUTJ     AT LEVEL 005 AS OF 81/11/03
      SUBROUTINE  INPUTJ (ID,R,Z,IH,NUMNP,NEQ,NEQH,NEQS)
      IMPLICIT  REAL*8   (A-H,O-Z)

      NODAL POINT INPUT AND GENERATION.

******CHANGED  BECAUSE  OF  NEW  ARRAY  CREATED  FOR  SHAKEDOWN  ANALYS
      DIMENSION  R(1),Z(1),ID(NUMNP,4),NID(4),IH(1)
      DIMENSION  R(1),Z(1),ID(NUMNP,4),NID(4),IH(NUMNP,2)
      REWIND 48
      REWIND 49
      ND=3
      WRITE(6,200)
      WRITE(6,204)
      KO=1
      DO  5  I=1,4
    5 NID(I)=0
   11 READ(5,100) N,(ID(N,I),I=1,4),R(N),Z(N),KN
      WRITE(6,203)N,(ID(N,I),I=1,4),R(N),Z(N),KN
      DO  60  I=1,4
      IF(ID(N,I))  61,62,63
   61 NID(I)=-1
      ID(N,I)=1
      GO  TO  60
   62 IF(NID(I).EQ.-1)  ID(N,I)=1
      GO  TO  60
   63 NID(I)=0
   60 CONTINUE
      IF(KO.EQ.1)  GO  TO  12

      CHECK  IF  GENERATION  NEEDED
      IF(KN)  10,10,20
   12 KO=0
   10 CONTINUE
      NUMINT=1
      GO  TO  15

      GENERATION  NEW  NODES

   20 NUMINT=(N-NI)/KN
      DR=(R(N)-R(NI))/NUMINT
      DZ=(Z(N)-Z(NI))/NUMINT
      NUMINT=NUMINT-1
      DO  21  J=1,NUMINT
      NN=NI+J*KN
      R(NN)=R(NN-KN)+DR
      Z(NN)=Z(NN-KN)+DZ

      SET  BOUNDARY  CODES  ...  SAME  AS  FIRST  JOINT  IN  SERIES

      DO  22  JJ=1,4
      IF(ID(NI,JJ)-1)  24,26,25

      GENERATE  NEW  MASTER  NODES
   25 ID(NN,JJ)=ID(NI,JJ)+J*KN
      GO  TO  22
   26 ID(NN,JJ)=ID(NI,JJ)
      GO  TO  22
   24 ID(NN,JJ)=0
   22 CONTINUE
```

```
   21 CONTINUE
   15 N1=N

      CHECK  FOR  LAST  NODAL  POINTS

      IF(NUMNP-N1) 13,13,11
   13 CONTINUE

      PRINT  ALL  NODAL  DATA

      WRITE(6,202)
      WRITE(6,204)
      DO  50  N=1,NUMNP

      WRITE(6,203) N,(ID(N,I),I=1,4),R(N),Z(N)
      WRITE(40,203) N,(ID(N,I),I=1,4),R(N),Z(N)
   50 CONTINUE

********ARRAY  IH(NUMNP,2)  IS  BEING  CREATED  HERE. IT  WILL  BE  THE
********CALLED  IN  ELSTIF ROUTINE  FOR  COMPUTING  THE  LM  ARRAY
      NEQS=0
      DO  750  I=1,NUMNP
      DO  700  J=1,2
       IF(ID(I,J))  680,650,680
  650 NEQS=NEQS+1
       IH(I,J)=NEQS
      GO  TO  700
  680 IH(I,J)=-1
  700 CONTINUE
  750 CONTINUE
***   ARRAY  IH(NUMNP,2)  IS  STORED  IN  NEW  LOCATIONS  (  N4  TO  N5)
*******PRINT  HERE  IH  ARRAY  AND  NEQS
********FINISH  PRINTING
      NEQ=0
      NEQH=0
      DO  75  I=1,NUMNP
      DO  70  J=1,4
      IF(ID(I,J))  68,65,68
   65 NEQ=NEQ+1
      ID(I,J)=NEQ
      GO  TO  70
   68 ID(I,J)=-1
   70 CONTINUE
   75 CONTINUE
      NEQH=NEQ
      WRITE(6,205)  NEQ,NEQH

      RETURN

  100 FORMAT (5I5,2F10.4,I5)
  104 FORMAT (I5,4F10.4)
  200 FORMAT(1H1//26HNODAL POINT DATA AS INPUT//)                        **4
  202 FORMAT(1H1//26HCOMPLETE NODAL POINT DATA  //  )                    **4
  203 FORMAT (I5,4I5,2F10.4,I5)
  204 FORMAT (5H3NODE 3X 10H B.C.CODES  11X
     *  23HNODAL POINT COORDINATES / 7H NUMBER 2X, 1HX,4X,1HY 4X 2HWX 3X
     * 2HWY 11X 1HX 12X 1HY/)
  205 FORMAT(1H1/30H NUMBER OF EQUATIONS .......  I10/                   **5
     *          30H NUMBER OF ROWS OF H            I10)
```

```
        DATA SET INSYMA      AT LEVEL 002 AS OF 81/09/04
      SUBROUTINE  INSYMA(NPD,A)
      IMPLICIT REAL*8(A-H,O-Z)

      INVERSE  SYMMETRIC  MATRIX
      DIMENSION  A(NPD,NPD)
      DO  200  N=1,NPD
      DIAG= A(N,N)
       DO  100  J=1,NPD
  100 A(N,J)=-A(N,J)/DIAG
      DO  150  I=1,NPD
      IF(N.EQ.I)  GO  TO  150
       DO  140  J=1,NPD
      IF(N.EQ.J)  GO  TO  140
  120 CONTINUE
      A(I,J)=A(I,J)+A(I,N)*A(N,J)
  140 CONTINUE
  150 A(I,N)=A(I,N)/DIAG
      A(N,N)=1.0/DIAG
  200 CONTINUE
      RETURN
      END
```

```
        DATA SET ITRSLD      AT LEVEL 010 AS OF 83/02/02
      SUBROUTINE ITRSLD(NEL,L,GSTRES,FACN,PS,TT,PG,LM,XP)     00001**5
                                                              00002**5
                                                              00003**2
    IMPLICIT REAL *8(A-H,O-Z)                                 00004**2
                                                              00005**2
      DIMENSION GSTRES(9),FACN(4),PS(16,8),P(4,8),XP(16),IS(8) 00006**7
                                                              00007**7
   *,IL(8),LM(16),TT(5,12)                                    00008**2
                                                              00009**2
    DATA IS/1,2,5,6,9,10,13,14 /,IL/3,4,7,8,11,12,15,16/      00010**3
                                                              00011**2
                                                              00012**2
                                                              00013**8
    TAU11= GSTRES(1)*FACN(L)                                  00014**2
    TAU22=GSTRES(2)*FACN(L)                                   00015**2
    TAU33=GSTRES(3)*FACN(L)                                   00016**2
    TAU12=GSTRES(4)*FACN(L)                                   00017**2
    TAU=PG*FACN(L)                                            00018**2
                                                              00019**2
                                                              00020**2
       DO 300 I=1,8                                           00021**7
    DO  300 JJ=1,4                                            00022**2
      J=JJ+4*(L-1)                                            00023**6
       P(JJ,I)=PS(J,I)                                        00024**7
                                                              00025**7
300 CONTINUE                                                  00026**2

        WRITE(6,301)
301  FORMAT(1X,'  PRINT OF STRAIN MATRIX AND VOLUME ')
        WRITE(6,302) ((P(I,J),J=1,8),I=1,4)
        WRITE(6,302) (FACN(L))
302     FORMAT(5X,8E10.3)
                                                              00027**2
    DO  310 II=1,8                                            00028**2
    NS=IS(II)                                                 00029**2
    NF=IL(II)                                                 00030**2
    XP(NS)=XP(NS)+P(1,II)*TAU11+P(2,II)*TAU22+P(3,II)*TAU33+P(4,II)* 00031**2
   *TAU12                                                     00032**2
    XP(NF)=XP(NF)+TT(1,II)*TAU                                00033**2
310 CONTINUE                                                  00034**2
                                                              00035**2
    ****   WRITE  INFORMATION ON TAPE FOR ASSEMBLING  ****    00036**2
                                                              00037**2
      WRITE(6,160)
160   FORMAT(5X,' *** PRINT IN ITRSLD ***',/)
      WRITE(6,161) (XP(I),I=1,16)
161   FORMAT(5X,8E10.3)
    RETURN                                                    00038
    END                                                       00039
```

```
      DATA SET KSTAR       AT LEVEL 036 AS OF 83/03/05
      SUBROUTINE KSTAR(IH,A1,A3,XM,NEQ,MBAND,NEQH,MBANDH,A4,A5,A6,MAXA

     * ,NWA,NEQB,NBLOCK,M1,ID,RE,SW)

      IMPLICIT  REAL*8   (A-H,O-Z)


C***TAPE13***  PASS  INFORMATION  ON  ELEMENT  MATRICES************
C***TAPE 12 ***  PASSON  ELEMENT  STATIC  LOAD  VECTOR *********
C***TAPE  4 ***  PASS  ON  EFFECTIVE  STIFFMESS  FOR  DECOMPOSITON
COR CONTAIN  ORIGINAL  STIFFNESS  MATRIX  FOR  STATIC  OR  DYNAMIC(
COF  ANALYSIS)
C***TAPE 9 *** CONTAIN  SYSTEM  MASS  MATRIX  FOR  DYMANIC  ANALYSI



      COMMON/CONTL/DD,HED(8),RADN,NUMNP,NUMEL,NKK,MASIM,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQK,MBANDK,NEQS,MBANDS

      COMMON/EMB/S(16,16),EM(16), H(16,16),SS(16,16),XP(16),XN(
     *16),LM(16)
      COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX


      COMMON/ELPAR/NPAR(14),NELTYP
      COMMON/SOL/ANORM,NBCOCK,NEQC,LL,NF,IDUM,NEIG,NAD,NVV,NFO
      COMMON/EXTRA/MODEX,NT8,N10SV,NT10
           COMMON/TAPES/NQQ(6),NUMBER,LCOUNT




           COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT

      DIMENSION A3(NEQH,1),XM(1),IH(NUMNP,1) ,A1(NEQS,1)
      DIMENSION  A4(NEQ,1),A5(NEQ,1),A6(NEQ,1),MAXA(1),ID(NUMNP,4)
C******
      DIMENSION RE(1),SW(1)
C *****
      DO  11   I=1,NEQ
      DO  11   J=1,MBAND
      A5(I,J)=0.0
      A4(I,J)=0.0
      A6(I,J)=0.0
   11 CONTINUE
      DT1=TETA*DD
      DT2=DT1**2
      A0=6./DT2
      A02=3./DT1

      DO  10  I=1,NEQ
      XM(I)=0.0
      RE(I)=0.0
      SW(I)=0.0
      DO  10  J=1,MBAND
   10 A3(I,J)=0.0

      DT=TETA*DD
      C1=DT
```

```
      C2=1.0/(BETA*DT*DT)                                               00060
      C3=GAMA/(BETA*DT)                                                 00061
      REWIND 13                                                         00062
                                                                        00063
                                                                        00064
      REWIND  9                                                         00065
      REWIND  4                                                         00066
                                                                        00067
      DO  900  N=1,NUMEL                                                00068
      *******CHANED  FROM  ORIGINAL  FOR  STATIC  TRANSIENT  PROBLEM****00069
                                                                        00070*22
                                                                        00071*22
      READ(13) LM,S,EM,H,SS,XP,XN                                       00072*22
                                                                        00073*22
      WRITE(6,39)
   39 FORMAT(5X,'  PRINT IN KSTAR DAMPING MATRIX ')
      WRITE(6,40) ((H(I,J),J=1,16),I=1,16)
   40 FORMAT(2X,16E8.1)
                                                                        00074*22
      DO  360  I=1,16                                                   00075
      L=LM(I)                                                           00076
      GO  TO(400,400,400,500,500),NDYN                                  00077
  400  IF(L)  360,360,200                                               00078
                                                                        00079
                                                                        00080
                                                                        00081
  200      CONTINUE                                                     00082*27
           IF(NDYN.EQ.4.OR.NDYN.EQ.5) GO TO 201                         00083*27
                                                                        00084*27
                                                                        00085*27
            XM(L)=XM(L)+EM(I)                                           00086*27
                                                                        00087*27
  201       CONTINUE                                                    00088*27
   *****   CHECK  FOR  RESIDUAL  LOAD  VECTOR   *******                 00089**9
      RE(L)=RE(L)+XP(I)                                                 00090**9
      SW(L)=SW(L)+XN(I)                                                 00091*19
  500 DO  350  J=1,16                                                   00092
      KL=LM(J)                                                          00093
      IF(KL)  350,350,250                                               00094
  250 IF(KL-L) 350,300,300                                              00095
  300 K=KL-L+1                                                          00096
      A3(L,K)=A3(L,K)+SS(I,J)                                           00097**2
      A4(L,K)=A4(L,K)+S(I,J)                                            00098
      A5(L,K)=A5(L,K)+H(I,J)                                            00099
  350 CONTINUE                                                          00100
  360 CONTINUE                                                          00101
  900 CONTINUE                                                          00102
       WRITE(6,37)
   37 FORMAT(1H1/5X,' PRINT IN KSTAR' )
       WRITE(6,38) (XM(I),I=1,NEQ)
   38 FORMAT(5X,12F10.3)
       WRITE(6,36)((A4(I,J),J=1,MBAND),I=1,NEQ)
        WRITE(6,36)((A5(I,J),J=1,MBAND),I=1,NEQ)

        IF(KKS.NE.3) GO TO 16
         IF(NC.EQ.0) GO TO 16                                           00103*35
                                                                        00104*35
                                                                        00105*34
   15     LCOUNT=LCOUNT+1                                               00107*36
          WRITE(6,18) LCOUNT,ISTRPR
```

```
 18     FORMAT(5X,2I5)
                                                                  00108*34
        IF(LCOUNT.NE.ISTRER) GO TO 20                             00109*34
                                                                  00110*34
                                                                  00112*34
                                                                  00113
 16   CONTINUE

      WRITE(6,34)                                                 00114*15
   34 FORMAT(1H1/'   ***    PRINT  OF  RESIDUAL  LOAD  VECTOR  **** ')  00115*15
      WRITE(6,36)  (RE(I),I=1,NEQ)                                00116*15
      WRITE(6,35)                                                 00117*20
   35 FORMAT(1H1/'   ***   PRINT  OF  SELF LOAD  VECTOR  ****** ' )  00118*20
      WRITE(6,36) (SW(I),I=1,NEQ)                                 00119*20
   36 FORMAT(5X,8E12.5)                                           00120*15
                                                                  00121
                                                                  00122*17
                                                                  00123
                                                                  00124*18
                                                                  00125
                                                                  00126
             *********    CHECK  REWIND LOCATION IN MAIN  ******8  00127*31
                                                                  00128*31
               REWIND 60                                          00129*31
                                                                  00130*31
                                                                  00131*31
 999     FORMAT(//)                                               00132*29
                                                                  00133
                                                                  00134
                                                                  00135*33
                                                                  00136
         WRITE(6,970)                                             00137*26
 970  FORMAT(1H1/'   ******* PRINT OF ASSEMBLED MATRIX IN KSTAR **')  00138*26
                                                                  00139*26
          WRITE(6,971) ((A4(I,J),J=1,MBAND),I=1,NEQ)              00140*26
         PRINT 999                                                00141*29
         WRITE(6,972)                                             00142*29
 972     FORMAT(5X,'      PRINT OF MASS MATRIX  '/)               00143*29
         WRITE(6,971) (XM(I),I=1,NEQ)                             00144*30
         WRITE(6,971) ((A4(I,J),J=1,MBAND),I=1,NEQ)               00140*26
 800     CONTINUE                                                 00145*29
                                                                  00146*26
         WRITE(60) ((A4(I,J),J=1,MBAND),I=1,NEQ)                  00147*28
                                                                  00148*28
                                                                  00149*28
                                                                  00150*34
  20     CONTINUE                                                 00151*34
 971     FORMAT(5X,10E10.3)                                       00152*32
                                                                  00153*29
                                                                  00154*26
                                                                  00155
      WRITE(4) ((A4(I,J),I=1,NEQ),J=1,MBAND)                      00156**
      WRITE(9)  (XM(I),I=1,NEQ)                                   00157
      NEQB=NEQ                                                    00158
      MA=MBAND                                                    00159
      NBLOCK=1                                                    00160
      NEIG=0                                                      00161
      NWA=NEQB*MBAND                                              00162
      NIB=(MBAND-2)/NEQB+1                                        00163
      IF(NIB.GE.NBLOCK)  NIB=NBLOCK-1                             00164
```

```
      MI=NEQB+MBAND-1


      IF(NDYN.NE.5)  GO  TO  966


      DO  965 I=1,NEQ
      IF(A3(I,1).EQ.0.0)  A3(I,1)=1.0E+20
  965 CONTINUE
  966 CONTINUE
      IF(NDYN.EQ.3)  GO  TO  1001
**************************************************************************

      IF(NDYN.EQ.4)  GO  TO  1000
      DO  961  I=1,NEQ




      IF(NDYN.EQ.5)  GO  TO  9610
      A4(I,1)=A4(I,1)+A0*XM(I)+A02*A5(I,1)
      GO  TO  961
 9610 A4(I,1)=A4(I,1)+(1./D11)*A5(I,1)
  961 CONTINUE
      DO  962  J=2,MBAND
      DO  962  I=1,NEQ



      IF(NDYN.EQ.5)  GO  TO  9620
      A4(I,J)=A4(I,J)+A02*A5(I,J)
      GO  TO  962
 9620 A4(I,J)=A4(I,J)+(1./D11)*A5(I,J)
  962 CONTINUE





      INFORMATION  FOR  ENTRRING  IN  TRIFAC   MATRIX  DECOMPOSITION ***
 1000 CONTINUE
      REWIND   4
      WRITE(4)  ((A4(I,J),I=1,NEQ),J=1,MBAND)
       CALL  TRIFAC(A4,A6,MAXA,NEQB,MA,NBLOCK,NWA,NTB,NEQ,MI)
 4000 CONTINUE


 1001 CONTINUE


      RETURN
      END
```

```
        DATA SET LOAD        AT LEVEL 027 AS OF 83/02/16
      SUBROUTINE LOAD(INEL,PRESS,VOL,WA)                                00001**6
                                                                        00002**5
      IMPLICIT  REAL*8(A-H,O-Z)                                         00003
                                                                        00004**5
      COMMON/CONTL/DT,HED(8),RADN,NASIM,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA00005
     *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH                                00006
      COMMON/EMB/ S(16,16),GM(16),GN(16,16),SS(16,16),XP(16),XN(        00007*21
     *16),LM(16)                                                                00008*21
                                                                        00009**2
      *******  ELEMANT  LOAD  VECTOR  EVALUATED  FROM  ELEMENT  CENTROID00010**2
                                                                        00011**2
                                                                        00012
        COMMON/BELA/NU,NELTYP,ND,NUMMAT,MODEL,NCON,IDW,NK015,NK016,NK01700013*27
                                                                        00014**9
                                                                        00015**9
      COMMON/  NANCY/  CM(12000)                                        00016**9
      COMMON/BOBBY/KKS,NUMGAT,NC,LOOP,NDRN,NINT,IDWA,NPT                00017*15
                                                                        00018
                                                                        00019
      COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),    00020**2
     *ST(20,12),D(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8)               00021**2
                                                                        00022*18
                                                                        00023*18
         COMMON/GORA/IFLAG,ITR                                          00024*26
                                                                        00025*23
         COMMON/RANI/RM(5),ZM(5),IPT                                    00026*25
      DIMENSION  IS(8),IL(8)                                            00027**2
                                                                        00028**2
                                                                        00029**2
      DATA IS/1,2,5,6,9,10,13,14/,IL/3,4,7,8,11,12,15,16/               00030**2
                                                                        00031**2
                                                                        00032**2
      DIMENSION  WA(IDWA,1),PRESS(5)                                    00033*15
                                                                        00034**9
                                                                        00035*21
          GO TO (400,400,500,400),KKS                                   00036*23
                                                                        00037*23
 500      IF(IFLAG.EQ.0) GO TO 400                                      00038*23
                                                                        00039*23
                                                                        00040*21
                                                                        00041**9
                                                                        00042*15
                                                                        00043*15
                                                                        00044**9
                                                                        00045*19
                                                                        00046**9
                                                                        00047*22
                                                                        00048*22
      NG=15*(INEL-1)*NPT+(IPT-1)*15                                     00049*22
       DO  200  J=1,5                                                   00050**9
 200  PRESS(J)=CM(NG+10+J)                                              00051**9
                                                                        00052*15
                                                                        00053*15
 400  CONTINUE                                                          00054*16
                                                                        00055*15
                                                                        00056*15
      TAU11=PRESS(1)*FAC                                                00057*15
      TAU22=PRESS(2)*FAC                                                00058*15
      TAU33=PRESS(3)*FAC                                                00059*15
```

```
      TAU12=PRESS(4)*FAC                                                 00060*15
      DO   300   II=1,6                                                  00061*15
                                                                         00062*15
      NS=IS(II)                                                          00063*15
      NF=IL(II)                                                          00064*15
                                                                         00065*15
                                                                         00066*15
      XP(NS)=XP(NS)+P(1,II)*TAU11+P(2,II)*TAU22+P(3,II)*TAU33+P(4,II)    00067*17
     **TAU12                                                             00068*17
                                                                         00069*17
                                                                         00070*15
      XP(NF)=XP(NF)+T1(1,II)*PRESS(5)*FAC                                00071*15
                                                                         00072*15
                                                                         00073*15
  300 CONTINUE                                                           00074*15
                                                                         00075**9
                                                                         00076*12
                                                                         00077*12
  999 FORMAT(//)                                                         00078*12
                                                                         00079*20
                                                                         00080*20
  510 FORMAT(1X,6E12.5)                                                  00081*14
                                                                         00082**9
      RETURN                                                             00083
                                                                         00084*24
                                                                         00085*24
      END                                                                00086
```

```
       DATA SET LOAD1       AT LEVEL 035 AS OF 83/03/05
   SUBROUTINE  LOAD1(NUMNP,NEQ,LC,ISLC,DT,NC,ISC,JSC,SURTRX,SURTRY,

  *X,Y,Q,ID,A01,TR,TM,PDYL,SURPFX,SURPFY)

   IMPLICIT REAL*8(A-H,O-Z)

   COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC

   COMMON/SOL/ANORM,NGBLCK,NEQB,NDC

        COMMON/KRISNA/SCALB(50),SCALF(50),LL

        COMMON/BOBBY/KKS,NUMMAT,NG,LOOP,NDRN,NINT,IDWA,NPT

        COMMON/PETER/AREA,WIDTH,HOR,VERT,ACCG

   DIMENSION  Q(1),X(1),Y(1),TD(2),ISC(1),JSC(1),SURTRX(2,NDC,2),
  *SURTRY(2,NDC,2),ID(NUMNP,1),TS(2),PDYL(2,4,NUMNP)
   DIMENSION  SURPFX(2,NDC,2),SURPFY(2,NDC,2)

    ******   TEMPORARY  STORAGE   **********
   DIMENSION A01(1)
    ISTOP=0
   IF(NSLC.EQ.0)  GO  TO  900

   ***********      SURFACE  LOAD  PRESCRIBED  HERE  **  STATIC  OR
   *****  CHECK  VARIABLE    (1)    *****
   NJ=NUMNP*4
   DO  2  I=1,NJ
 2 Q(I)=0.0

 ***********        CONDITION  CHECK   *******************
   IF(NC.NE.0)  GO  TO  700

READ AND PRINT SURFACE LOADING (TRACTION) CARDS
30 WRITE(6,108)LC
   KL=1
55 CONTINUE

   DO  40  N=KL,2
   READ(5,1012)  TD(N)
   IF(NC.GT.13) GO TO 57
   WRITE(6,2016)  TD(N)
57 CONTINUE
      DO 40 L=1,NSLC

   READ(5,41)ISC(L),JSC(L),SURTRX(N,L,1),SURTRX(N,L,2),SURTRY(N,L,1),
  *SURTRY(N,L,2)
```

```
      SURTRX(N,L,1)=SURTRX(N,L,1)*SCALB(L)                              00053*29
      SURTRX(N,L,2)=SURTRX(N,L,2)*SCALB(L)                              00054*29
      SURTRY(N,L,1)=SURTRY(N,L,1)*SCALB(L)                              00055*29
      SURTRY(N,L,2)=SURTRY(N,L,2)*SCALB(L)                              00056*29
                                                                        00057*29
                                                                        00058*27
      IF(NDRN.EQ.0)  GO  TO  59                                         00059*28
                                                                        00060*27
      ******   READ  FLUID  LOADING  ***************                    00061*27
                                                                        00062*27
      READ(5,41) ISC(L),JSC(L),SURPFX(N,L,1),SURPFX(N,L,2),SURPFY(N,L,1)00063*27
     *,SURPFY(N,L,2)                                                    00064*27
                                                                        00065*29
      SURPFX(N,L,1)=SURPFX(N,L,1)*SCALF(L)                              00066*29
      SURPFX(N,L,2)=SURPFX(N,L,2)*SCALF(L)                              00067*29
      SURPFY(N,L,1)=SURPFY(N,L,1)*SCALF(L)                              00068*29
      SURPFY(N,L,2)=SURPFY(N,L,2)*SCALF(L)                              00069*29
                                                                        00070**8
   59 CONTINUE
                                                                        00071**8
      IF(NC.GT.13) GO TO 58
      WRITE(6,42)                                                       00072
     *ISC(L),JSC(L),SURTRX(N,L,1),SURTRX(N,L,2),SURTRY(N,L,1),          00072
     1 SURTRY(N,L,2),SURPFX(N,L,1),SURPFX(N,L,2),SURPFY(N,L,1),SURPFY(N,00073**3
     *L,2)                                                              00074**3
                                                                        00075**8
   58 CONTINUE
                                                                        00076**8
   40 CONTINUE
                                                                        00077**8
  108 FORMAT(54H1INPUT TABLE 5.. SURFACE LOADING DATA FOR LOADING CASE, 00078
     *I5//)                                                             00079
 2016 FORMAT(                                                           00080
     117X, 43HSURFACE LOAD INTENSITIES AT NODES FOR TIME,F10.7/         00081
     24X,6HNODE I,4X,6HNODE J,10X,2HXI,10X,2HXJ,10X,2HYI,10X,2HYJ)      00082
   41 FORMAT(2I4,4E11.4)                                                00083*27
   42 FORMAT(2I10,8F12.5)                                               00084**3
 CONVERT LINEARLY VARYING SURFACE TRACTIONS TO STATIC EQUIVALENTS,      00085
 AND ADD TO OVERALL LOAD VECTOR R, EQ.(5-61A).                          00086
 *******  SET  UP  LOAD  MATRIX                                         00087
                                                                        00088*12
                                                                        00089*12
                                                                        00090*12
      IF(KL.EQ.2)  GO  TO  117                                          00091
      KL=2                                                              00092
      TM=TD(1)+DT                                                       00093
      L=2                                                               00094
      DTXX=TD(2)-TD(1)                                                  00095
                                                                        00096*12
                                                                        00097*12
  700 CONTINUE                                                          00098
       IF(TM.GT.TK)  GO  TO  120                                        00099
                                                                        00100*35
                                                                        00101*33
                                                                        00102*33
    3    CONTINUE                                                       00103*34
                                                                        00104*34
           WRITE(6,1) DT,NC,TM,TK,TD(2),TD(1)                           00105*33
    1      FORMAT(5X,'           DT           = ',F5.2/                 00106*33
     *           5X,'            NC           = ',I5/                   00107*33
```

```
     *              5X,'           TM           = ',F5.2/                      00108*33
     *              5X,'           TF           = ',F5.2/                      00109*33
     *              5X,'           TD(2)        = ',F5.2/                      00110*33
     *              5X,'           TD(1)        = ',F5.2/)                     00111*33
                                                                               00112*33
                                                                               00113*33
                                                                               00114*35
          IF(TD(2)-TM)  115,120,120                                            00115*35
                                                                               00116*35
  115  TD(1)=TD(2)                                                             00117
                                                                               00118**8
                                                                               00119**8
      DO  116  L=1,NSLC                                                        00120
      SURTRX(1,L,1)=SURTRX(2,L,1)                                              00121
      SURTRX(1,L,2)=SURTRX(2,L,2   )                                           00122
      SURTRY(1,L,1)=SURTRY(2,L,1)                                              00123
      SURTRY(1,L,2)=SURTRY(2,L,2)                                              00124
      SURPFX(1,L,1)=SURPFX(2,L,1)                                              00125**3
      SURPFX(1,L,2)=SURPFX(2,L,2   )                                           00126**3
      SURPFY(1,L,1)=SURPFY(2,L,1)                                              00127**3
      SURPFY(1,L,2)=SURPFY(2,L,2)                                              00128**3
  116 CONTINUE                                                                 00129**5
                                                                               00130**8
                                                                               00131**8
      GO  TO  55                                                               00132
  117 DTXX=TD(2)-TD(1)                                                         00133
      IF(DTXX) 118,115,120                                                     00134
  118 WRITE(6,2010)  TD(2)                                                     00135
      GO  TO  900                                                              00136
 1000 CONTINUE                                                                 00137
  120 DO  130  L=1,NSLC                                                        00138
      I=ISC(L)                                                                 00139
      J=JSC(L)                                                                 00140
      II=4*I                                                                   00141
      JJ=4*J                                                                   00142
      DX=X(J)-X(I)                                                             00143
      DY=Y(J)-Y(I)                                                             00144
      EL=DSQRT(DX*DX+DY*DY)                                                    00145
      IF(TM.GT.TF)  GO  TO  135                                                00146
                                                                               00147**8
                                                                               00148**8
                                                                               00149**8
      TME=TM-TD(1)                                                             00150
      SLOPE1=SURTRX(1,L,1)+((SURTRX(2,L,1)-SURTRX(1,L,1))/DTXX)*TME            00151
      SLOPE2=SURTRX(1,L,2)+((SURTRX(2,L,2)-SURTRX(1,L,2))/DTXX)*TME            00152
      SLOPE3=SURTRY(1,L,1)+((SURTRY(2,L,1)-SURTRY(1,L,1))/DTXX )*TME           00153
      SLOPE4=SURTRY(1,L,2)+((SURTRY(2,L,2)-SURTRY(1,L,2))/DTXX )*TME           00154
      SLOPE5=SURPFX(1,L,1)+((SURPFX(2,L,1)-SURPFX(1,L,1))/DTXX)*TME            00155**3
      SLOPE6=SURPFX(1,L,2)+((SURPFX(2,L,2)-SURPFX(1,L,2))/DTXX)*TME            00156**3
      SLOPE7=SURPFY(1,L,1)+((SURPFY(2,L,1)-SURPFY(1,L,1))/DTXX )*TME           00157**3
      SLOPE8=SURPFY(1,L,2)+((SURPFY(2,L,2)-SURPFY(1,L,2))/DTXX )*TME           00158**3
                                                                               00159**8
                                                                               00160**8
                                                                               00161*32
          HOR=SLOPE1                                                           00162*32
          VERT=SLOPE3                                                          00163*32
      PXI=SLOPE1*E L                                                           00164
      PXJ=SLOPE2*EL                                                            00165
      PYI=SLOPE3*EL                                                            00166
      PYJ=SLOPE4*EL                                                            00167
```

```
      FXI=SLOPE5*EL                                                     00168**3
      FXJ=SLOPE6*EL                                                     00169**3
      FYI=SLOPE7*EL                                                     00170**3
      FYJ=SLOPE8*EL                                                     00171**3
      GO  TO  180                                                       00172
  135 SLOPE1=SURTRX(2,L,1)                                              00173
      SLOPE2=SURTRX(2,L,2)                                              00174
      SLOPE3=SURTRY(2,L,1)                                              00175
      SLOPE4=SURTRY(2,L,2)                                              00176
      SLOPE5=SURPFX(2,L,1)                                              00177**3
      SLOPE6=SURPFX(2,L,2)                                              00178**3
      SLOPE7=SURPFY(2,L,1)                                              00179**3
      SLOPE8=SURPFY(2,L,2)                                              00180**3
                                                                        00181**8
                                                                        00182**8
                                                                        00183**8
      TME=1.0                                                           00184
      PXI=SLOPE1*EL*TME                                                 00185
      PXJ=SLOPE2*EL*TME                                                 00186
      PYI=SLOPE3*EL*TME                                                 00187
      PYJ=SLOPE4*EL*TME                                                 00188
      FXI=SLOPE5*EL*TME                                                 00189**3
      FXJ=SLOPE6*EL*TME                                                 00190**3
      FYI=SLOPE7*EL*TME                                                 00191**3
      FYJ=SLOPE8*EL*TME                                                 00192**3
  180 CONTINUE                                                          00193
                                                                        00194**8
                                                                        00195**8
                                                                        00196**8
                                                                        00197*15
      ************   LINEAR  VARIATION  LOAD  CALCULATION  ON  NODES  **00198**8
                                                                        00199**8
                                                                        00200**8
      Q(II-3)=Q(II-3)+PXI/3.0+PXJ/6.0                                   00201
      Q(JJ-3)=Q(JJ-3)+PXI/6.0+PXJ/3.0                                   00202
                                                                        00203*15
   75 CONTINUE                                                          00204*15
                                                                        00205*15
   76 CONTINUE                                                          00206*15
                                                                        00207*15
                                                                        00208*13
                                                                        00209*13
                                                                        00210*13
***********   'BRANCH  AT  THIS  POINT  FOR  PARABOLIC  SHEAR  LOADING  00211*13
                                                                        00212*13
      Q(II-2)=Q(II-2)+PYI/3.0+FYJ/6.0                                   00213
      Q(JJ-2)=Q (JJ-2)+PYI/6.0+FYJ/3.0                                  00214
      Q(II-1)=Q(II-1)+FXI/3.0+FXJ/6.0                                   00215**3
      Q(JJ-1)=Q(JJ-1)+FXI/6.0+FXJ/3.0                                   00216**3
      Q(II)  =Q(II) + FYI/3.0+FYJ/6.0                                   00217**3
      Q(JJ)  =Q(JJ) + FYI/6.0+FYJ/3.0                                   00218**3
                                                                        00219**8
                                                                        00220**8
                                                                        00221**8
      *******  CALCULATE  ALSO  BY  PARABOLIV  METHOD  *****************00222**8
                                                                        00223**8
                                                                        00224**8
  130 CONTINUE                                                          00225
      DO  600  N=1,NUMNP                                                00226
      JP=3                                                              00227
```

```
      DO  600  J=1,4                                                 00228
      KK=4*M-JP                                                      00229
      PDYL(2,J,M)=Q(KK)                                              00230
      XX=X(I)
      IF(IPLMAX.NE.1) GO TO 601
      PDYL(2,J,M)=Q(KK)*(XX)
  601 CONTINUE

      JP=JP-1                                                        00231
  600 CONTINUE                                                       00232
                                                                     00233**8
                                                                     00234**8
                                                                     00235**8
                                                                     00236**8
      DO  140  I=1,NUMNP                                             00237
      DO  140  J=1,4                                                 00238
      JJ=ID(I,J)                                                     00239
      IF(JJ)  140,140,150                                            00240
  150 AU1(JJ)=PDYL(2,J,1)                                            00241
  140 CONTINUE                                                       00242

                                                                     00243**8
                                                                     00244**8
  900 CONTINUE                                                       00245*24
      RETURN                                                         00246
 1012 FORMAT(8F10.0)                                                 00247
 2010 FORMAT(/'ERROR  IN  BASE ACCEL OR DYN LOAD,TIME    = ',F10.3)  00248
      END                                                            00249
```

```
          DATA SET LOAD2       AT LEVEL 009 AS OF 83/02/11
      SUBROUTINE LOAD2(NUMNP,PDYL,X,NUMLP,ID,DT,TD,NC,TK,TM,AU2)

      IMPLICIT  REAL*8  (A-H,O-Z)



      DIMENSION  PDYL(2,4,1),TD(2),R(1),ID(NUMNP,1)
      COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQK,MBANDH
     *,NEQS,MBANDS

       COMMON/INTEGR/IETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX
       COMMON/PETER/AREA,WIDTH,HOR,VERT,ACCG
      COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC,ISLC


      DIMENSION X(1),AU2(1)
      ***********DYNAMIC  LOAD   MATRIX   FORMULATION  *****************


      DYMAMIC  LOADA


      IF(ISLC.EQ.0)  GO  TO  900
      IF(NC.NE.0)  GO  TO  700

   45 CONTINUE
      DO  50  N=1,2
      DO  50  I=1,NUMNP
      DO  50  J=1,4
   50 PDYL(N,J,I)=0.0
      READ(5,1005)  HED
      READ(5,1006) NUMLP
      WRITE(6,2005) HED ,NUMLP
      IF(NUMLP.EQ.0)  GO  TO  900
      KL=1
   55 DO  60  N=KL,2
      READ(5,1012)  TD(N)
      WRITE(6,2016)  TD(N)
       DO  60  I=1,NUMLP
      READ(5,1007)  M,(PDYL(N,J,M),J=1,4)
   60 WRITE(6,2007) M,(PDYL(N,J,M),J=1,4)
      IF(IPLNAX.NE.1)  GO  TO  80
      DO  70  N=1,2
      DO  70  I=1,NUMNP
        DO  70  J=1,4
      XX=X(I)
   70 PDYL(N,J,I)=PDYL(N,J,I)*XX
   80 CONTINUE
      IF(KL.EQ.2)  GO  TO  117
      KL=2
      L=2
      DTXX=TD(2)-TD(1)
      TM=TD(1)+DT

      SET  UP  LOAD   MATRIX



```

```
      ************CREATE   ACTUAL   LOAD  VECTOR  FOR  DISCRETE  TIME  ST

      FOR  DYNAMIC  LOAD

  700 CONTINUE
      IF(NUMNP.EQ.0)  GO  TO  900
      IF(TM.GT.TK)  GO  TO  120
      IF(TD(2)-TM)  115,120,120
  115 TD(1)=TD(2)
      DO  116 N=1,4
      DO  116 M=1,NUMNP
  116 PDYL(1,N,M)=PDYL(2,N,M)
      GO  TO  55
  117 DTXX=TD(2)-TD(1)
      IF(DTXX)  118,115,120
  118 WRITE(6,2010)  TD(2)
      GO  TO  900
  120 DO  130 I=1,NUMNP
      DO  123 J=1,4
      JJ =ID(1,J)
      IF(TM.GT.TK)  GO  TO  129
      IF(JJ) 123,123,122
  122 SLOPE=(PDYL(2,J,I)-PDYL(1,J,I))/DTXX
        AO2(JJ)=PDYL(1,J,1)+(TM-TD(1))*SLOPE
       GO  TO  123
  129 CONTINUE
      IF(JJ)  123,123,128
  128 AO2(JJ)=PDYL(2,J,I)
  123 CONTINUE
  130 CONTINUE

      WRITE(6,190)
  190 FORMAT(1H1/'     ***  LOAD VECTOR IN  LOAD2  *****  ')
      WRITE(6,195)  NC,(AO2(I),I=1,NEQ)
  195 FORMAT(1X,I2,16E8.1)

  900 CONTINUE

      RETURN
 1005 FORMAT (10A8)
 1006 FORMAT (I5)
 1007 FORMAT(I5,4F10.0)
 1012 FORMAT(8F10.0)
 2005 FORMAT (1H1,/,8A8,/,
     *  ' NUMBER  OF  NODES  SUBJECT  TO DYNAMIC LOAD   = ',I5)
 2007 FORMAT (I5,4F15.4)
 2010 FORMAT(/'ERROR  IN  BASE ACCEL OR DYN LOAD,TIME    = ',F10.3)
 2016 FORMAT (1H1/'      NODAL POINT LOADS AT TIME    = ',F10.3///)
      END
```

```
      DATA SET LOAD3      AT LEVEL 009 AS OF 83/02/11
      SUBROUTINE LOAD3(TS,GACL,DT,NACL,XM,ID,NUMNP,NC,TR,TM,AO3,ACCL)

      IMPLICIT  REAL*8  (A-H,O-Z)

      DIMENSION TS(1), GACL(1),ID(2),AO(1),ID(NUMNP,1),XM(1)

      COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NRK,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQK,MBANDH
     *,NEQS,MBANDS
       COMMON/PETER/AREA,WIDTH,HOR,VERT,ACCG

      DIMENSION AO3(1)


      IF(NC.NE.0)  GO  TO  700
      READ BASE ACCEL.  HISTORY

      READ(5,1004)  HED
      READ(5,1009)  FAC1,FAC2
      READ(5,1008) (TS(I),GACL(I),I=1,NACL)
      TS(1)=0.0
      GACL(1)=0.0
      NTOT=NACL-1
      IN=2
      READ(5,1008)  (GACL(I),I=IN,NACL)


      MIN=1
      DO  251  L=2,NACL
  251 TS(L)=TS(L-MIN)+DD
      WRITE(6,2008)  HED ,FAC1,FAC2
      DO  40  I=1,NACL
      TS(I)=TS(I)*FAC1
   40 GACL(I)=GACL(I)*FAC2
      WRITE(6,2009)  ((TS(N),GACL(N)),N=1,NACL)


      **************END ***************************

  100 CONTINUE
      TM=TS(1)+DT
      L=2
      DTX=TS(L)-TS(L-1)



      PRINT  999
      WRITE(6,2170)  ((A3(I,J),J=1,MBAND),I=1,NEQ)
      PRINT  999
      WRITE(6,2170)  (A4(I ),I=1,NWA)
      PRINT  999
      WRITE(6,2170) ((A5(I,J),J=1,MBAND),I=1,NEQ)
 2170 FORMAT(5X,10E12.5)





      ********CREATE  LOAD  VECTOR  FOR  GROUND  MOTION  DISCRETE  STEP
```

```
     FOR  BASE  MOTION

 700 CONTINUE
 150 IF (TS(L)-TM)  155,155,160
 155 CONTINUE
     IF(L.EQ.NACL)  GO  TO  161
     DTX=TS(L+1)-TS(L)
     L=L+1
     IF(DTX) 158,155,160
 158 WRITE(6,2010)  TS(L-1)
     GO  TO  900
 160 SLOPE=(GACL(L)-GACL(L-1))/DTX
     ACCL=GACL(L-1)+(TM-TS(L-1))*SLOPE
        GO  TO  162
 161  ACCL=GACL(L)
 162  DO  180  I=1,NUMNP
       DO  170 J=1,2
     JJ=ID(1,J)
     IF(JJ)  170,170,165
 165 AU3(JJ)=0.0
     IF(JJ.EQ.ID(1,1)) AU3(JJ)=-ACCL*XM(JJ)
 170 CONTINUE
 180 CONTINUE

       ACCG=ACCL
     WRITE(6,190)
 190 FORMAT(1H1/'     ***  LOAD VECTOR IN  LOAD3  *****  ')
     WRITE(6,195)  NC,(AU3(I),I=1,NEQ)
 195 FORMAT(1X,I2,16E8.1)

 900 CONTINUE
     RETURN
1004 FORMAT (10A8)
1008 FORMAT(8F9.6)
1009 FORMAT(2F8.3)
2008 FORMAT(1H1/10A8/
   *  ' SCALE  FACTOR  FOR  TIME                = ',F10.3/
   *  ' SVALE  FACTOR  FOR  ACCEL                = ',F10.3//)
2009 FORMAT (/5X,'POINT',11X,'TIME',11X,'ACCEL', /(5X,I5,2F15.6))
2009 FORMAT(1H1/6(1X,11HTIME VALUE ,2X,8HFUNCTION), / (6(F5.2,E12.6)))
2010 FORMAT(/'ERROR  IN  BASE ACCEL OR DYN LOAD,TIME     = ',F10.3)
     END
```

```
      DATA SET LODCHK      AT LEVEL 002 AS OF 82/07/14
   SUBROUTINE LODCHK(NDT,NUMEL,NLC,NPJ,NEQS,CT,SN,SU)

   IMPLICIT REAL*8 (A-H,O-Z)

   COMMON/ASIM/NYP,NSC,NSLC,NGC,NNYP,NNSC

   DIMENSION RM(5),ZM(5),SIG(4),STRAIN(4),SU(1),CT(NPJ,1),SN(NPJ)




   REWIND  55

   DO  300 LC=1,NLC

   DO  100 M=1,NUMEL

   DO  200 I=1,NDT

   READ(55) L,RM(L),ZM(L),(SIG(II),II=1,4),(STRAIN(JJ),JJ=1,4),EPSV,
  *EPSVF

   K1=(NSC*I-2)+(M-1)*NSC*NDT
   SN(K1)=SIG(1)
   SN(K1+1)=SIG(2)
   SN(K1+2)=SIG(4)

200 CONTINUE

100 CONTINUE

   CALL CHECK(CT,SN,SU,NPJ,NEQS)

300 CONTINUE
   RETURN
   END
```

```
      DATA SET MAIN       AT LEVEL 103 AS OF 83/02/28
  PROGRAM FSDYN (INPUT,OUTPUT,TAPE3,TAPE4,TAPE5=INPUT,TAPE6=OUTPUT) 00001
  SUBROUTINE MAIN(NCOM)
                                                                    00002
  IMPLICIT  REAL*8   (A-H,O-Z)                                      00003
                                                                    00004*81
      COMMON D(400000)                                              00005102
                                                                    00006*81
                                                                    00007*17
                                                                    00008*17
                                                                    00009*17
  COMMON/EMB/BBBB(1500)                                             00010*17
  COMMON/QUAD/CCCC(1500),N1,N2,N3,N4,N5,MTOT,NNN,NK15               00011*68
  COMMON/CONTL/DT,HED(8),RADN,NUMNP,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA00012
 *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS                    00013**8
  COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00014*78
  COMMON/SOL/ANORM, NBLOCK,NEQB,LL,NF,IDUM,NEIG,NAD,NVV,NFO         00015
  COMMON/EXTRA/MODEX,NT8,N10SV,NT10                                 00016
     COMMON/TAPES/NOQ(6),NUMBER,LCOUNT                              00017101
  COMMON/DIMEB/N43,IPT,IPT1,KK,NPRNT                                00018*90
                                                                    00019*81
  COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017   00020*44
                                                                    00021
        COMMON/BENDY/SG(5,200,4,4),NCOND,NTENS                      00022*99
                                                                    00023*98
         DIMENSION SIG(9)                                           00024*98
         COMMON/GORA/IFLAG                                          00025*98
                                                                    00026*98
                                                                    00027
                                                                    00028
                                                                    00029*77
  COMMON/CLOCK/TIM1,TIM2,TIM3                                       00030*77
                                                                    00031*67
                                                                    00032*67
  COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT                00033*67
  COMMON/ANBN/FA                                                    00034*22
                                                                    00035*46
  *********   ALL  INFORMATION  ABOUT  PLOTTING   *****************00036*46
                                                                    00037*46
   COMMON /KABI/JPT,KEB,JPLOT(200),KEMMT(200)                       00038*94
     COMMON/PETER/AREA,WIDTH,HOR,VERT,ACCG                          00039101
     COMMON/NEVEL/WEIGHT,SHEAR                                      00040101
                                                                    00041*94
                                                                    00042*94
                                                                    00043*77
  ********    FINISH  ALL INFORMATION  REGARDIND  PLOT  ************00044*77
                                                                    00045*77
                                                                    00046*77
                                                                    00047*46
                                                                    00048*46
  DIMENSION  IDUMM(36)                                              00049
  EQUIVALENCE (IDUMM(1),D(1))                                       00050
                                                                    00051*43
                                                                    00052*43
  COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC                            00053
                                                                    00054
                                                                    00055*77
  CALL SETIME                                                       00056*80
                                                                    00057*77
      MTOT=400000                                                   00058103
```

```


      DO  885 I=1,NTOT
  885 D(I)=0.0
      PI=4.0*DATAN(1.0D0)
      RADN=180./PI
      LOOP=1

    5 CONTINUE
       CALL  ERRSET(88,.TRUE.,.FALSE.,.FALSE.,.FALSE.,255)
       CALL  ERRSET(89,.TRUE.,.FALSE.,.FALSE.,.FALSE.,255)
        CALL  ERRSET(208,256,-1,1,1,208)

      READ  CONTROL  DATA



      CALL GETIME(ITIM1)
      TIM1=FLOAT(ITIM1)/1000.

      WRITE(6,1008) ITIM1
      WRITE(6,1003) TIM1


      READ(5,990)  HED
      READ(5,1000) NUMNP,NUMEL,NDYN,IPLNAX,LL,NF,MODEX,NAD,NGAMA
      IF(MODEX.GT.0) MODEX=1
      IF(NUMNP.EQ.0)  STOP
      WRITE(6,2000) HED,NUMNP,NUMEL,NDYN,IPLNAX,LL,NF,MODEX,NAD
       READ(5,1002)  ISTRPR,NPRINT,ISTART
      READ(5,1001) NCYCL,DT,KKS,FA,NDRN,NCON,IDW,MODEL,NINT
      WRITE(6,2001) NCYCL,DT,KKS,FA,NDRN,NGAMA



         READ(5,1020) AREA,WIDTH,WEIGHT,SHEAR
         WRITE(6,1030)AREA,WIDTH,WEIGHT,SHEAR
 1030    FORMAT(1H1///40X,'    PLATFORM AREA       = ',E10.3/
     *                40X,'    PLATFORM WIDTH      = ',E10.3/
     *                40X,'    PLATFORM WEIGHT     = ',E10.3/
     *                40X,'    SOIL SHEAR STRENGTH= ',E10.3/)
 1020    FORMAT(4E10.3)

        *********   INPUT  ALL  DATA  RELATED  PLOTTING  OF  OUTPUT  ****
        READ(5,1008) JPI
        IF(JPI.EQ.0) GO TO 550
        READ(5,1009) (JPLOT(I),I=1,JPI)
  550   READ(5,1011) KEL
        IF(KEL.EQ.0) GO TO 560
        READ(5,1009)(KELMT(I),I=1,KEL)

         WRITE (6,1050)
 1050    FORMAT(1H1/// '  *******    NODE AND ELEMENT DATA  *** '/)
         WRITE(6,1040) JPI,(JPLOT(I),I=1,JPI)
         WRITE(6,1040) KEL,(KELMT(I),I=1,KEL)
 1040    FORMAT(10X,20I5)
  560   CONTINUE



```

```
                                                                     00117*46
      *******  FINISHED  READING  ALL INPUR DATA RELATED  TO  PLOTTING  00118*46
                                                                     00119*46
                                                                     00120*45
                                                                     00121*45
                                                                     00122*45
      CONTINUE                                                       00123
      INPUT  NODAL  POINTS                                           00124
                                                                     00125
      N1=1                                                           00126
      N2=4*NUMNP+N1                                                  00127
      N3=NUMNP+N2                                                    00128
      N4=NUMNP+N3                                                    00129
      IF(N4.LT.MTOT)  GO  TO  810                                    00130*69
      WRITE(6,2006)                                                  00131*69
      WRITE(6,815)                                                   00132*69
  815 FORMAT(1H1/'  ***   VALUE  OF  N4   ****  ')                   00133*69
      WRITE(6,820)  N4                                               00134*69
      STOP                                                           00135*69
  820 FORMAT(2X,I5)                                                  00136*69
                                                                     00137*69
  810 CONTINUE                                                       00138*69
                                                                     00139*69
                                                                     00140
                                                                     00141
                                                                     00142
*********************************************************************00143
      SUBROUTINE  READS  INPUT   DATA   AND   FINDS  THE   TOTAL   EQUAT00144
*********************************************************************00145
                                                                     00146**8
                                                                     00147**8
                                                                     00148*28
      ********  N1       =     ID                                    00149*28
      ********  N2       =     X                                     00150*28
      ********  N3       =     Y                                     00151*28
      ********  N4       =     IH                                    00152*28
      CALL INPUTJ(D(N1),D(N2),D(N3),D(N4),NUMNP,NEQ,NEQH,NEQS)       00153**8
      N5=N4+NUMNP*2                                                  00154
                                                                     00155
                                                                     00156
                                                                     00157
                                                                     00158*77
      CALL GETIME(ITIM2)                                             00159*80
      TIM2=FLOAT(ITIM2)/1000.                                        00160*80
                                                                     00161*81
      WRITE(6,1008) ITIM2                                            00162*81
      WRITE(6,1003) TIM2                                             00163*81
                                                                     00164*81
                                                                     00165*77
                                                                     00166*77
      ***  PRCESSING OF BASIC INPUT DATA   NODAL POINTS   ***        00167*77
                                                                     00168*77
      PINPUT=(TIM2-TIM1)                                             00169*77
                                                                     00170
*********************************************************************00171
      SUBROUTINE  ELEMT  CALLS   ELSTF   AND  PASSES  DIFFERENT  STIFFMA00172
      MASSES,AND  DAMPING  ,LOAD  VECTOR  FOR  STATIC  OR  DYNAMIC  ANAL00173
      AND  WRITES  THOSE   ON  TAPES                                 00174
                                                                     00175
                                                                     00176
```

```

*************************************************************************


         IFLAG=0


      NNN=1
      NC=0


      ****************   TAPE  14  REQUIRED  FOR  ELASTO PLASTIC ANALYAIS

      REWIND 14


      CALL ELEMT(NUMMAT)




      CALL GETIME(ITIM3)
      TIM3=FLOAT(ITIM3)/1000.

      WRITE(6,1008) ITIM3
      WRITE(6,1003) TIM3

      ***  PROCESSING OF BASIC ELEMENT DATA INITIALIZATION
      ***  ELEMENT STIFFNESS,MASS,LOAD VECTOR FORMULATION

      PELEMT=(TIM3-TIM2)

      IF(NNN.EQ.1)  GO  TO  830
      WRITE(6,2006)
      WRITE(6,840)
  840            FORMAT(1H1/'  ***   PROBLEM  IN  ELEMT  SUBROUTINE')
      WRITE(6,820) NK15
      STOP

  830 CONTINUE
        READ(5,1003) TETA,BETA,GAMA,RTOL,NPUNCH,NEQUIB,ITEMX,NACL
        WRITE(6,2008)TETA,BETA,GAMA,RTOL,NPUNCH,NEQUIB,ITEMX
      IF(NACL.NE.0)  WRITE(6,2002)  NACL


          READ(5,505) NLC,NP3,NP6,NUMBER,NCOND,NTENS


      N6=NEQS*MBANDS+NK15
      N7=NEQH*MBANDH+N6
      N07=N7+NEQ
      N007=N07+NEQS
      N8=N007+NEQ
      N09=N8+NEQ*MBAND
      N010=N09+NEQ*MBAND
      N011=N010+NEQ*MBAND
     ******  ALLOCATE  RESIDUAL  LOAD  VECTOR  AT   THIS  POINT   ******
      ******  NP012  IS  THE  STARTING  POINT  FOR  RESIDUAL  LOAD  VECT
      NP012=N011+( NEQ +MBAND-1)
      ND012=NP012+NEQ
```

```
      N012=ND012+(NEQ)                                                00237*73
                                                                      00238*69
                                                                      00239*69
                                                                      00240*69
      IF(N012.LT.MT01)  GO  TO  860                                   00241*70
      WRITE (6,2006)                                                  00242*69
      WRITE(6,850)                                                    00243*69
  850 FORMAT(1H1/'  ****  VALUE  OF  N012  ****')                     00244*70
          WRITE(6,820) N012                                           00245*69
      STOP                                                            00246*69
  860 CONTINUE                                                        00247*70
                                                                      00248
                                                                      00249*17
                                                                      00250*17
                                                                      00251*17
      NN07=N012+NUMMAT                                                00252*17
      NN08=NN07+NUMMAT                                                00253*17
      NN09=NN08+NUMMAT                                                00254*17
      NN10=NN09+NUMMAT                                                00255*17
      NN11=NN10+NUMMAT                                                00256*17
      NN12=NN11+NUMMAT                                                00257*17
      NN13=NN12+NUMMAT                                                00258*17
      NN14=NN13+NUMMAT                                                00259*17
      NN15=NN14+NUMMAT                                                00260*33
                                                                      00261*17
      ND15=NN15+NEQS*NEQS
      IF(KKS.EQ.2) GO TO 835
      ND16=ND15+NEQS*1
      ND17=ND16+NEQS
      ND18=ND17+NEQS*1
      ND19=ND18+NEQS
      GO TO 836
                                                                      00262*17
  835  CONTINUE
      ND16=ND15+NEQS*NEQS
      ND17=ND16+NEQS
      ND18=ND17+NEQS*NEQS
      ND19=ND18+NEQS
  836 CONTINUE


                                                                      00263*17
      **********  NN07        =     GMX                               00264*28
      **********  NN08        =     DPL                               00265*28
      **********  NN09        =     GUSE                              00266*28
      **********  NN10        =     E                                 00267*28
      **********  NN11        =     EN                                00268*28
      ************ NN12       =     PD                                00269*28
      IF(KKS.EQ.2)  CALL  STRCOM(NUMMAT,D(NN07),D(NN08),D(NN09),D(NN10),00270*17
                                                                      00271*41
                                                                      00272*41
     *D(NN11),D(NN12))                                                00273*20
                                                                      00274*78
                                                                      00275*78
      CALL GETIME(ITIM4)                                              00276*80
      TIM4=FLOAT(ITIM4)/1000.                                         00277*80
                                                                      00278*78
                                                                      00279*78
      SUM=TIM4-TIM3                                                   00280*78
```

```
      NDUM=NUMNP
      NUMBER=1

      SUM1=0.
      SUM2=0.
      SUM3=0.
      SUM4=0.

      SUM5=0.

      DO  600  LOOP=1,NUMBER

      LCOUNT=0


      CALL GETIME(ITIM5)
      TIM5=FLOAT(ITIM5)/1000.


      IF(KKS.EQ.2.OR.KKS.EQ.4)
     *   CALL  SED(D(NK15),D(N07),NEQB,MBANDB,NUMNP,NUMEL,

     *KKS,D(N1),D(N4),LOOP,NDUM,D(NN13),D(NN14),NUMMAT,D(NN07)

     *,D(NN08),D(NN09),D(NN10),D(NN11),D(NN12),D(NN15),D(ND15),D(ND16),

     *D(ND17),D(ND18))


      CALL GETIME(ITIM6)
      TIM6=FLOAT(ITIM6)/1000.

      SUM1=SUM1+(TIM6-TIM5)

*************************************************************************
      SUBROUTINE  KSTAR  ASSEMBLE  THE  DIFFERENT  MATRICES  FORMED
      ELSTIF
*************************************************************************


      CALLKSTAR(D(N4),D(NK15),D(N6),D(N7),NEQ,MBAND,NEQH,MBANDH,D(N8),

     *D(N09),D(N010),D(N011),NWA,NEQB,NBLOCK,MI,D(N1),D(NP012),D(ND012))


      CALL GETIME(ITIM7)
      TIM7=FLOAT(ITIM7)/1000.

      **** PROCESSING AND ASSEMBLING DECOMPOSING STIFFNESS MATRIX  ***

      SUM2=SUM2+(TIM7-TIM6)
```

```
                                                                   00338*78
                                                                   00339*77
                                                                   00340*77
                                                                   00341
                                                                   00342
       NN17=N019+1                                                 00343*33
      IF(LOOP.NE.1)  GO  TO  650                                   00344*31
                                                                   00345*69
                                                                   00346*69
                                                                   00347*69
  701 N9=NEQ+NN17                                                  00348*33
      N10=NEQ+N9                                                   00349
      N11=NEQ+N10                                                  00350
      N12=2*4*NUMNP+N11                                              00351
      IF(NDYN.NE.1)  N12=N11                                       00352
      N13=NACL+N12                                                 00353
      N14=NACL+N13                                                 00354
      N15=NEQ+N14                                                  00355
      N16=NEQ+N15                                                  00356
      N17=NEQ+N16                                                  00357
      N017=N17+NEQ                                                 00358
      N18=18*NUMEL+N017                                            00359*17
       N19=24*NUMNP+N18                                            00360*97
                                                                   00361*33
      WRITE(6,508) N1,N2,N3,N4,N5,N6,N7,N07,N007,N8,N09,N010,N011,N012, 00362*33
                                                                   00363*41
                                                                   00364*41
     *NN07,NN08,NN09,NN10,NN11,NN12,NN13,NN14,NN15,NN17,N10,N11,N12,N13,00365*33
                                                                   00366*41
                                                                   00367*41
     *N14,N15,N16,N17,N18,N19                                      00368*33
  508 FORMAT(1H1/'    N1 TO N19 ',/,2(17I6,/))                     00369*43
                                                                   00370*41
                                                                   00371*41
      N20   =2*NEQH+N19                                            00372
       N25=N20+NEQ                                                 00373
        N26=N25+NEQ                                                00374
      N27=N26+NEQ                                                  00375
      N28=N27+NEQ                                                  00376
      N29=N28+NEQ                                                  00377
      N30=N29+NEQ                                                  00378
      N029=N29+NEQ                                                 00379
      N30=N029+NEQ                                                 00380
                                                                   00381*69
                                                                   00382*69
      IF(N30.LT.M101)  GO  TO  875                                 00383*72
      WRITE(6,2006)                                                00384*69
      WRITE(6,870)                                                 00385*69
  870 FORMAT(1H1/'******   VALUE  OF  NT  =***  ')                 00386*69
      WRITE(6,820) N30                                             00387*69
      STOP                                                         00388*69
                                                                   00389*18
                                                                   00390*18
                                                                   00391*18
                                                                   00392*18
 5000 II=N8                                                        00393
      NASIM=N21-1                                                  00394
       DO  1500  I=N20,NASIM                                       00395
      D(I)=D(II)                                                   00396
 1500 II=II+1                                                      00397
```

```
      II=N09                                                            00398
      NAKIM=N22-1                                                       00399
      DO  1501  I=N21,NAKIM                                             00400
      D(I)=D(II)                                                        00401
1501  II=II+1                                                           00402
      II=N010                                                           00403
      NADIM=N23-1                                                       00404
      DO  1502  I=N22,NADIM                                             00405
      D(I)=D(II)                                                        00406
1502   II=II+1                                                          00407
      II=N011                                                           00408
      NAFIM=N24-1                                                       00409
      DO  1503  I=N23,NAFIM                                             00410
      D(I)=D(II)                                                        00411
1503  II=II+1                                                           00412
                                                                        00413
                                                                        00414
                                                                        00415
                                                                        00416
                                                                        00417
 875    CONTINUE                                                        00418100
                                                                        00419
           N21=0                                                        00420103
           N22=0                                                        00421103
           N23=0                                                        00422103
           N24=0                                                        00423103
                                                                        00424
************************************************************************00425
      SUBROUTINE  SOLVED   CALLS   STRESS   AND  FINDS  THE  SOLUTIONS  00426
************************************************************************00427
                                                                        00428
                                                                        00429
 625  CONTINUE                                                          00430*33
      WRITE(6,509)    NLC,NP3,NP6,NUMBER,NCOND,NTENS                    00431*33
                                                                        00432*33
      N1G=NLC-1                                                         00433*57
      NDT=1
      IF(KKS.EQ.4.AND.NCOND.EQ.1) NDT=4
                                                                        00434*33
      NYP=NP6                                                           00435
      NSC=NP3                                                           00436
**********  VARUABLE  DEFINED  FOR  ALLOCATION  *******************      00437
                                                                        00438
      NPJ=NP3*NUMEL                                                     00439
      NPK=NP6*NUMEL                                                     00440
                                                                        00441*82
      NPJ=NPJ*NDT                                                       00442*82
      NPK=NPK*NDT                                                       00443*82
                                                                        00444*82
                                                                        00445*82
      N31=N30+NUMNP*4                                                   00446
      N32=N31+NUMNP                                                     00447
      N33=N32+NUMNP                                                     00448
                                                                        00449*59
                                                                        00450*59
                                                                        00451*59
        *********   TEMPORARY  ALLOCATION  FOR  SURFACE  LOAD  VECTOR   00452*59
                                                                        00453*59
      N34=N33+2*LL*2                                                    00454*73
      N35=N34+2*LL*2                                                    00455*73
```

```
      NNYP=NUMEL*NYP

      NNYP=NNYP*NDT

      N36=N35+NNYP
      N37=N36+NNYP
      NNSC=NUMEL*NSC
      NNSC=NNSC*NDT


      N38=N37+NNSC*NNYP
      N39=N38+NLC
      N40=N39+NYP*NYP
      N41=N40+NYP

       **********   TEMPORARY  ALLOCATION  FOR  SURFACE  PORE  PRESSURE

      N42=N41+2*LL*2
      N43=N42+2*LL*2



      IF(N43.LT.MTOT)  GO TO  880
      WRITE(6,2000)
      WRITE(6,890)
  890 FORMAT(1H1/'  ***  VALUE  OF  N43  *** ')
      WRITE(6,820) N43
      STOP
  880 CONTINUE

      **************   CREATE  HERE  DO  LOOP  FOR  STATIC  SHAKEDOWN  P


      IF(KKS.EQ.4)  READ(5,500) IPT,IPT1,KK,NPRNT

  650 CONTINUE



         ***************   ASSIGN  ALL TAPE NUMBERS FOR OUTPUT *********






       TAPE 21 USED IN  'ASSEMB' AND 'ITRSLD' ROUTINE
       TAPE 55 USED IN  'OUT' ROUTINE TO STORE DISPL.,VEL.,ACCLN.
       TAPE 57 USED IN  'STRSPR' ROUTINE TO STORE STRESSES ,STRAINS


       REWIND 17
       REWIND 18
       REWIND 21
      REWIND  55
      REWIND  57
      REWIND  61
```

```
      DO    8000  IC=1,NBC                                                   00516
                                                                             00517*78
      CALL GETIME(ITIM8)                                                     00518*80
      TIM8=FLOAT(ITIM8)/1000.                                                00519*80
                                                                             00520*78
                                                                             00521*33
                                                                             00522*33
      WRITE(6,7490) N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,N31,         00523
     *N32,N33,N34,N35,N36,N37,N38,N39,N40                                    00524
 7490 FORMAT(1H1/'  N20  TO  N40 ',/,2(10I12,/))                             00525*86
                                                                             00526*18
                                                                             00527*18
                                                                             00528*18
                                                                             00529*18
                                                                             00530*28
                                                                             00531*28
      *******   N007      =     B                                            00532*28
      *******   N9        =     B0                                           00533*28
      *******   N10       =     VEL                                          00534*28
      *******   N11       =     PDYB                                         00535*28
      *******   N12=      =     TS                                           00536*28
      *******   N13       =     GACL                                         00537*28
      *******   N1        =     ID                                           00538*28
      *******   N14       =     K                                            00539*28
      *******   N15       =     A                                            00540*28
      *******   N16       =     A0                                           00541*28
      *******   N17       =     MB                                           00542*28
      *******   N017      =     FMAX                                         00543*28
      *******   N18       =     AMAX                                         00544*28
      *******   N19       =     MB3                                          00545*28
      *******   N8        =     A8                                           00546*28
      *******   N09       =     A5                                           00547*28
      *******   N010      =     A6                                           00548*28
      *******   N011      =     MAXA                                         00549*28
      *******   N20       =     X0                                           00550*28
      *******   N25       =     X1                                           00551*28
      *******   N26 .     =     X2                                           00552*28
      *******   N27       =     AT                                           00553*28
      *******   N28       =     BI                                           00554*28
      *******   N29       =     AC                                           00555*28
      *******   N2        =     X                                            00556*29
      *******   N3        =     Y                                            00557*29
      *******   N30       =     Q                                            00558*28
      *******   N31       =     ISC                                          00559*28
      *******   N31       =     JSC                                          00560*28
      *******   N33       =     SURTRX                                       00561*28
      *******   N34       =     SURTRY                                       00562*28
      *******   N35       =     AM                                           00563*28
      *******   N36       =     EK                                           00564*28
      *******   N37       =     AN                                           00565*28
      *******   N38       =     NSLC1                                        00566*28
      *******   N39       =     AN1                                          00567*28
      *******   N40       =     EKJ                                          00568*28
      *******   N41       =     SURPFX                                       00569*28
      *******   N42       =     SURPFY                                       00570*28
                                                                             00571*33
      *******   NP012           =     RE                                     00572*76
      *******   ND012           =     SW                                     00573*76
                                                                             00574*33
```

```
      WRITE(6,8001) N4,NK15,N6,N7,N007,N9,N10,N11,N12,N13,N1,N14,N15,N16
      WRITE(6,8002)
     *N17,N017,N18,NEQ,NUMNP,NACL,NUMEL,NEQH,N19,N8,N09,N010,N011,N20
      WRITE(6,8003)
     *N25,N26,N27,N28,N29,N029,NWA,NEQB,NBLOCK,M1,N2,N3,N30,N31,N32,N33
      WRITE(6,8004)
     *N34,N35,N36,N37,N38,N39,N40,LC,N41,N42,NP012,ND012
                                                                   00575
8001 FORMAT(1H1/'   LOCATION OF      IH(NUMNP,1) N4    = ',I8/
     *          '   LOCATION OF      A1(NEQS,1)  NK15 = ',I8/
     *          '   LOCATION OF      A3(NEQ,1)   N6    = ',I8/
     *          '   LOCATION OF      XM(NEQ)     N7    = ',I8/
     *          '   LOCATION OF      B(NEQ)      N007 = ',I8/
     *          '   LOCATION OF      BO(NEQ)     N9    = ',I8/
     *          '   LOCATION OF      VEL(NEQ)    N10   = ',I8/
     *          '   LOCATION OF      PDYL(2,4,NUMNP)   = ',I8/
     *          '   LOCATION OF      TS(NACL)    N12   = ',I8/
     *          '   LOCATION OF      GACL(NACL)  N13   = ',I8/
     *          '   LOCATION OF      ID(NUMNP,1) N1    = ',I8/
     *          '   LOCATION OF      R(NEQ)      N14   = ',I8/
     *          '   LOCATION OF      A(NEQ)      N15   = ',I8/
     *          '   LOCATION OF      AU(NEQ)     N16   = ',I8/)
8002  FORMAT(   '   LOCATION OF      GB(NEQ)     N17   = ',I8/
     *          '   LOCATION OF      PMAX(NUMEL,1)     = ',I8/
     *          '   LOCATION OF      AMAX(NUMNP,12,2)  = ',I8/
     *          '                    NEQ               = ',I8/
     *          '                    NUMNP             = ',I8/
     *          '                    NACL              = ',I8/
     *          '                    NUMEL             = ',I8/
     *          '                    NEQH              = ',I8/
     *          '   LOCATION OF      GCB(NEQ)      N19 = ',I8/
     *          '   LOCATION OF      A4(NEQ,1)     N8  = ',I8/
     *          '   LOCATION OF      A5(NEQ,1)     N09 = ',I8/
     *          '   LOCATION OF      A6(NWA)       N010= ',I8/
     *          '   LOCATION OF      MAXA(NEQ)     N011= ',I8/
     *          '   LOCATION OF      X0(NEQ)       N20 = ',I8/)
8003  FORMAT(   '   LOCATION OF      X1(NEQ)       N25 = ',I8/
     *          '   LOCATION OF      X2(NEQ)       N26 = ',I8/
     *          '   LOCATION OF      A1(NEQ)       N27 = ',I8/
     *          '   LOCATION OF      B1(NEQ)       N28 = ',I8/
     *          '   LOCATION OF      AC(NEQ)       N29 = ',I8/
     *          '   LOCATION OF      S0(NEQ)       N029= ',I8/
     *          '                    NWA               = ',I8/
     *          '                    NEQB              = ',I8/
     *          '                    NBLOCK            = ',I8/
     *          '                    M1                = ',I8/
     *          '   LOCATION OF      X(NUMNP)      N2  = ',I8/
     *          '   LOCATION OF      Y(NUMNP)      N3  = ',I8/
     *          '   LOCATION OF      Q(      )     N30 = ',I8/
     *          '   LOCATION OF      ISC(NUMNP)    N31 = ',I8/
     *          '   LOCATION OF      JSC(NUMNP)    N32 = ',I8/
     *          '   LOCATION OF      SURTRX(2,LL,2)N33= ',I8/)
004   FORMAT(   '   LOCATION OF      SURTRY(2,LL,2)N34= ',I8/
     *          '   LOCATION OF      AM(NNYP)        N35 = ',I8/
     *          '   LOCATION OF      EK(NNYP)        N36 = ',I8/
     *          '   LOCATION OF      AN(NNSC*NNYP)   N37 = ',I8/
     *          '   LOCATION OF      NSLC1(NLC)      N38 = ',I8/
     *          '   LOCATION OF      ANI(NYP*NYP)    N39 = ',I8/
     *          '   LOCATION OF      EKJ(NYP)        N40 = ',I8/
```

```
     *            '                 LC                = ',I8/
     *            '  LOCATION OF    SURFFX(2,LL,2)N41 = ',I8/
     *            '  LOCATION OF    SURFFY(2,LL,2)N42 = ',I8/
     *            '  LOCATION OF    RE(NEQ)       NP012= ',I8/
     *            '  LOCATION OF    SW(NEQ)       ND012= ',I8/)
      CALL SOLVED(D(N4),D(NK15),D(N6),D(N7),D(N007),D(N9),D(N10),         00576*42
                                                                          00577*43
     *D(N11),D(N12),D(N13),     D(N1),D(N14),  D(N15),D(N16),D(N17),D(N0100578
                                                                          00579
     *7),D(N18),        NEQ,NUMNP,NACL,NUMEL,NEQH,D(N19),D(N8),D(N09),D(N00580
                                                                          00581
     *010),D(N011),D(N20 ),D(N25),D(N26),D(N27),D(N28),D(N29),D(N029),    00582
                                                                          00583
     *        NWA,NEQB,NBLOCK,MI, D(N2),                                  00584
                                                                          00585
     *D(N3),        D(N30),D(N31),D(N32),D(N33),D(N34),D(N35),D(N36),D(N3700586
                                                                          00587*33
                                                                          00588*33
     *),D(N38),D(N39),D(N40),LC,D(N41),D(N42),D(NP012),D(ND012),D(NN15))00589*73
                                                                          00590*26
                                                                          00591*76
                                                                          00592*76
                                                                          00593*76
       *********    PUT CONDITION FOR GRAVITY STATIC ANALYSIS WITH DYNAM00594*53
      SHAKEDOWN PROBLEM    ******************    CALL  KSTAR  AND  RESET 00595*53
                                                                          00596*53
      GO  TO  (755,755,755,758),KKS                                       00597*56
  758 CONTINUE                                                            00598*56
                                                                          00599*57
                                                                          00600*57
                                                                          00601*57
      IF(LC.NE.N16)  GO  TO  755                                          00602*57
                                                                          00603*57
      GO  TO  (790,790,755,755,755),NDYN                                  00604*57
  790 NDYN=4                                                              00605*57
      NCYCL=1                                                             00606*60
                                                                          00607*57
                                                                          00608*57
                                                                          00609*56
                                                                          00610*56
      CALL KSTAR (D(N4),D(NK15),D(N6),D(N7),NEQ,MBAND,NEQH,MBANDH,D(N8),00611*56
                                                                          00612*73
     *D(N09),D(N010),D(N011),NWA,NEQB,NBLOCK,MI,D(N1),D(NP012),D(ND012))00613*73
                                                                          00614*53
                                                                          00615*26
  755 CONTINUE                                                            00616*56
                                                                          00617*98
                                                                          00618*77
                                                                          00619*77
      CALL GETIME(ITIM9)                                                  00620*80
      TIM9=FLOAT(ITIM9)/1000.                                             00621*80
                                                                          00622*77
                                                                          00623*77
      *****  SOLVING EQN. OF MOTION FOR ALL CASES  *****                  00624*77
                                                                          00625*77
                                                                          00626*77
      SUM3=SUM3+(TIM9-TIM8)                                               00627*78
                                                                          00628*78
                                                                          00629*77
                                                                          00630*77
```

```
                                                                        00631*26
8000 CONTINUE                                                           00632
                                                                        00633
                                                                        00634
                                                                        00635
                                                                        00636
          GO TO (780,760,750,780),KKS                                   00637*99
 760 CONTINUE                                                           00638*26
                                                                        00639*78
      CALL GETIME(ITIM10)                                               00640*80
      TIM10=FLOAT(ITIM10)/1000.                                         00641*80
                                                                        00642*76
                                                                        00643*76
     *************     STRAIN  COMPATIBLE   PROBLEM    *****************00644*76
                                                                        00645*76
                                                                        00646*76
                                                                        00647*26
     CALL STRAIN(D(NO17),NUMEL,NUMMAT,D(NN07),D(NN08),D(NN09),D(NN10),  00648*28
                                                                        00649*46
    *D(NN11),D(NN12))                                                   00650*29
                                                                        00651*26
     CALL GETIME(ITIM11)                                                00652*80
     TIM11=FLOAT(ITIM11)/1000.                                          00653*80
                                                                        00654*78
     SUM4=SUM4+(TIM11-TIM10)                                            00655*78
                                                                        00656*78
                                                                        00657
                                                                        00658
     GO  TO  750                                                        00659*30
     ************     CALL  SHAKE  AT  THIS  STAGE  ************          00660
                                                                        00661*78
                                                                        00662*78
 765 CONTINUE                                                           00663*79
     CALL GETIME(ITIM12)                                                00664*80
     TIM12=FLOAT(ITIM12)/1000.                                          00665*80
                                                                        00666*78
                                                                        00667
 780 CONTINUE                                                           00668*30
                                                                        00669*81
                                                                        00670*77
          CALL YLDCHK(FYLD,IPLNAX,SIG,LC,NUMEL,NPT)                     00671*99
                                                                        00672*99
        IF(KKS.EQ.1) GO TO 750                                          00673*99
                                                                        00674*99
                                                                        00675*77
                                                                        00676*77
 500 FORMAT(4I5)                                                        00677*90
                                                                        00678*26
     WRITE(6,2009) IPT,IPT1,KK                                          00679*78
                                                                        00680*87
                                                                        00681*87
                                                                        00682*26
     CALL SHAKE(NFK,NPJ,IPT,KK,D(N1),D(N4),D(N35),D(N36),D(N37),D(N39)  00683*78
                                                                        00684*78
    *,N43,D(NK15),IPT1,D(N17),D(N029))                                  00685*89
                                                                        00686
     GO  TO  555                                                        00687*43
 768 CONTINUE                                                           00688*79
                                                                        00689*78
                                                                        00690*78
```

```
      CALL GETIME(ITIM13)                                                 00691*80
      TIM13=FLOAT(ITIM13)/1000.                                           00692*80
                                                                          00693*78
      SUM5=SUM5+(TIM13-TIM12)                                             00694*78
                                                                          00695*78
                                                                          00696*78
                                                                          00697
                                                                          00698
  800 WRITE(6,2006)                                                       00699
                                                                          00700
  900  GO  TO  555                                                        00701
                                                                          00702
  750 CONTINUE                                                            00703*26
                                                                          00704*26
                                                                          00705*26
                                                                          00706*31
  600 CONTINUE                                                            00707*31
                                                                          00708*95
  555 CONTINUE                                                            00709
                                                                          00710*77
                                                                          00711*77
      WRITE(6,2010)                                                       00712*77
      WRITE(6,2011) PINPUT,PELEMT,SUM,SUM1,SUM2,SUM3,SUM4,SUM5            00713*78
  505     FORMAT(615)                                                     00714101
  509 FORMAT(1H1/                                                         00715*33
     *  ' NUMBER OF LOAD CASES                         = ',I5/            00716*33
     *  ' NUMBER OF STRESS STATE IN THE BODY           = ',I5/            00717*33
     *  ' NUMBER OF YIELD PLANES                       = ',I5/            00718*33
     *  ' NUMBER OF ITR. FOR KKS =2                    = ',I5/
     *  ' NCOND. FOR KKS =4 ; NCOND. =0;CENTROID       = ',I5/
     *  ' NTENS. FOR KKS=4 ;  NTENS. =0;NO TENSION     = ',I5)
  990 FORMAT(10A8)                                                        00719*55
 1000 FORMAT(9I5)                                                         00720*76
 1001 FORMAT(I5,F10.0,I5,F10.0,5I5)                                       00721*67
 1002 FORMAT(16I5)                                                        00722
 1003     FORMAT(4F10.0,4I5)                                              00723*94
                                                                          00724*46
      *****   ALL  FORMAT  RELATED   TO  PLOTTING   ********************00725*46
 1008 FORMAT(5X,2I5)                                                      00726*46
 1009     FORMAT(16I5)                                                    00727*94
                                                                          00728*94
                                                                          00729*94
 1010 FORMAT(24I5)                                                        00730*46
 1011 FORMAT(5X,2I5)                                                      00731*46
 1012 FORMAT(5X,6I5)                                                      00732*46
 1013 FORMAT(24I5)                                                        00733*46
                                                                          00734*46
      ************   FINISHED  READING  ALL  FORMAT   *****************00735*46
                                                                          00736
 2000 FORMAT(1H1,/,8A8,/,                                                 00737*53
     *  ' NUMBER OF NODAL POINTS                   = ',I5/                00738
     *  ' NUMBER OF ELEMENTS                       = ',I5/                00739
     *  ' CODE FOR DYNAMIC OR STATIC ANALYSIS      = ',I5/                00740
     *  ' EQ.1. LOAD VECTOR (DIRECT ANALYSIS)        ' /                  00741
     *  ' EQ.2. ARTHQUAKE  ANALYSIS (DIRECT INTEGRATION)  ' /             00742
     *  ' EQ.3.  EIGENANALYSIS                       ' /                  00743
     *  ' EQ.4.  STATIC  ANALYSIS  (SOLID)           '   /                00744
     *  ' EQ.5.   TRANSIENT  ANALYSIS  ( SOLID FLUID ) '  /               00745
     *  ' CODE  FOR  PLANE  PROBLEM  ANALYSIY      = ',I5/                00746
     *  ' EQ.-1. PLANE  STRESS   ANALYSIS            ' /                  00747**7
```

```
     *    ' EQ.0.  PLANE  STRAIN   ANALYSIS                   ' /                 00748**7
     *    'EQ.1. AXISYMMETRIC  ANALYSIS                        ' /                 00749**7
     *     '  NUMBR  OF  LOAD  CASES                       = ',I5/                 00750
     *    ' NUMBER  OF  FREQUENCIES                        = ',I5/                 00751
     *    '  SOLUTION   MODE (MODEX)                       = ',I5/                 00752
     *    ' EQ.0.    EXECUTION                              ' /                    00753
     *     ' EQ.1.     DATA  CHECK                          ' /                    00754
     *    ' NUMBER  OF  SUBSPACE                           = ',I5/                 00755
     *    ' ITERATION CYCLES                                          ' /)         00756*75
2001 FORMAT(                                                                       00757
 .   *    ' NUMBER OF CYCLES                               = ',I5/                 00758
     *    ' TIME STEP  INCREMENTS                                  = ',F10.3/      00759*14
     *    ' CODE FOR MATERIAL CONSTITUTIVE PROPERTIES              = ',I5/         00760*14
     *    ' EQ.1.    ELASTIC ANALYSIS (ISOTROPIC)                  ' /             00761*14
     *     ' EQ.2.    STRAIN COMPATIBLE PROPERTIES (EQ. LINEAR) ' /                00762*14
     *    ' EQ.3.    ELASTO PLASTIC ANALYSIS ( ZERO HARDENING ) ' /                00763*14
     *    ' EQ.4.    SHAKEDOEN ANALYSIS (ZERO HARDENING )         ' /              00764*20
     *    ' GRAVITATIONAL ACCL VALUE                         = ',E12.5/            00765*42
     *    ' TYPE OF ANALYSIS TO BE PERFORMED                   = ',I5/             00766*43
     *    ' EQ.0...... UNDRAINED  ANALYSIS    ............  ' /                    00767*43
     *    ' EQ.1.          UNDRAINED  DRAINED  ANALYSIS              ' /           00768*74
     *    '  BODY  FORCES                                  = ' ,I5/                00769*74
     *    ' EQ.0   NOT INCLUDED                                        ' /         00770*74
     *    ' EQ.1         INCLUDED                                      ' /)        00771*74
2002 FORMAT (1H1/                                                                  00772
     *    ' NUMBER  OF  BASE ACCEL POINTS                  = ',I5)                 00773
2006 FORMAT (//' REQUIRED DIMENSION OF ARRAY D  EXCEEDS MTOT')                     00774
2008 FORMAT (1H1/                                                                  00775
     *    ' TETA                            = ', F10.3/                            00776
     *    ' BETA                            = ', F10.3/                            00777
     *    ' GAMA                                 = ',F10.3/                        00778*77
     *    ' TOLERANCE FOR CONVERGANCE            = ',F10.6/                        00779*77
     *    ' PUNCH THE OUTPUT            E        = ',I5/                           00780*77
     *    ' EQ.0.  N3. PUNCH REQUIRED             ' /                              00781*77
     *    ' EQ.1. PUNCHING REQUURED                 ' /                            00782*77
     *    ' EQUILIBRIUM ITERATION NEEDED         = ',I5/                           00783*77
     *    ' EQ.0.   NO                            ' /                              00784*77
     *  ' EQ.1.    YES                            ' /                              00785*77
     *    ' NO.OF ITERATION IN TIME STEP MAX.    = ' ,I5)                          00786*77
                                                                                   00787*77
                                                                                   00788*77
                                                                                   00789*77
2009 FORMAT(1H1/'  TYPE  OF  OPTIMIZATION TO BE DONE  = ',I5/                      00790*62
     *    ' EQ..1 ......  MINIMIZATION  OF   FUNCTION             ' /              00791*43
     *    ' EQ.2.  .......MAXIMIZATION  OF  FUNCTION                ' /            00792*43
     *    ' EQ.3 . ....... MAXIMIZE FOR LIMIT ANALYSIS         ' /                 00793*81
     *    ' TYPE  OF  ANALYSIS                             = ',I5/                 00794*52
     *    ' EQ.0 MAXIMIZATION  W. R. T.  PLASTIC MULTIPLIERS  <     ' /            00795*52
     *    ' EQ.1    MAXIMIZATION  W.R.T.  RESIDUAL STRESS     ' /                  00796*52
     *    ' BOUNDING  TECHNIQUE  TO BE  APPLIED              = ',I5/               00797*43
     *    ' EQ.0......     NONE                               ' /                  00798*43
     *    ' EQ.1......     NEEDED                             ' /)                 00799*43
                                                                                   00800
                                                                                   00801*77
                                                                                   00802*77
2010 FORMAT(1H1/34H S O L U T I O N T I M E   L O G//12X,                          00803*77
     *          11HFOR PROBLEM//1X,12A6////)                                       00804*77
2011 FORMAT(49H INPUT PHASE ....................................F9.2//            00805*77
     *          49H ELEMENT DATA AND STIFFNESS MASS LOAD ...........F9.2//         00806*77
     *          49H INPUT FOR STRAIN COMPT. PROP.  .................F9.2//         00807*78
```

```
     *         49H SOLID STIFFNESS MASS FREQUENCY CAL...............F9.2// 00808*78
     *         49H ASSEMBLING OF STIFFNESS MASS LOAD ETC. ..........F9.2// 00809*77
     *         49H SOLUTION MODES INCLUDING  ALL TYPES .............F9.2// 00810*77
     *         49H STRAIN COMPATIBLE SOLUTION  .....................F9.2// 00811*77
     *         49H SHAKEDOWN SOLUTION ..............................F9.2//)00812*77
                                                                           00813
                                                                           00814
      STOP                                                                 00815
      RETURN
      END                                                                  00816
```

```
         DATA SET MAXMIN      AT LEVEL 012 AS OF 83/03/05
       SUBROUTINE MAXMIN(STRESS,P1,P2,AG,BLKST,SHSTN,PMAX,PP,IPT,TIME,00001**3
                                                                      00002**9
   *NUMEL,SM,SD,SHM)                                                  00003**9
    IMPLICIT REAL*8(A-H,O-Z)                                          00004
                                                                      00005
       COMMON/MATMOD/STRESS(9),STRAIN(4),C(4,4),MPT,NEL               00006**7
       DIMENSION PMAX(NUMEL,1),STRESS(9)                              00007**3
          STRESS(1)=STRESS(1)+PP                                      00008**9
          STRESS(2)=STRESS(2)+PP                                      00009**9
          STRESS(4)=STRESS(4)+PP                                      00010**9
                                                                      00011
                                                                      00012
    CC = (STRESS(1)+STRESS(2)) * 0.5                                  00013
    BB = (STRESS(1)-STRESS(2)) * 0.5                                  00014
    CR = DSQRT(BB**2 +STRESS(3)**2)                                   00015
    P1 =  CC+CR                                                       00016
    P2 =  CC-CR                                                       00017
    AG=45.0                                                           00018
       IF(DABS(BB).LT.1.0D-6) GO TO 1                                 00019**3
                                                                      00020**3
                                                                      00021**3
                                                                      00022
    AG = 28.648*DATAN2(STRESS(3),BB)                                  00023
                                                                      00024
                                                                      00025**3
                                                                      00026**4
  1     CONTINUE                                                      00027**4
                                                                      00028**3
        STRESS(5)=P1                                                  00029**3
        STRESS(6)=P2                                                  00030**3
        STRESS(7)=(P1+P2)/2.                                          00031**3
        STRESS(8)=(P1-P2)/2.                                          00032**3
                                                                      00033**3
        COMPUTE BLKST,SHSTN ETC;                                      00034**3
          SM=(STRESS(1)+STRESS(2)+STRESS(4))/3.                       00035**9
          DX=STRESS(1)-SM                                             00036**9
          DY=STRESS(2)-SM                                             00037**9
          DZ=STRESS(4)-SM                                             00038**9
          DS=STRESS(3)                                                00039**9
          SD=DSQRT(0.5*(DX*DX+DY*DY+DZ*DZ)+DS*DS)                     00040**9
                                                                      00041**9
        IF(STRAIN(3).EQ.0) GO TO 5                                    00042*11
        SHM=(STRESS(3)/STRAIN(3))                                     00043*12
                                                                      00044*12
  5     CONTINUE                                                      00045*11
                                                                      00046**3
        STRESS(9)=AG                                                  00047**3
                                                                      00048**3
  2     CONTINUE                                                      00049**3
        STRESS(4)=SM                                                  00050*10
           STRESS(7)=SD                                               00051**9
                                                                      00052**9
                                                                      00053**3
        DO 3 J=1,8                                                    00054**3
        BM=DABS(STRESS(J))                                            00055**3
        IF(BM-PMAX(NEL,J)) 180,100,100                                00056**3
100    PMAX(NEL,J)=BM                                                 00057**8
                                                                      00058**8
        PMAX(NEL,J+9)=TIME                                            00059**3
```

```
180     CONTINUE
  3     CONTINUE

          PF=DABS(PF)
          IF(PF-PMAX(NEL,9)) 95,200,200
200       PMAX(NEL,9)=PF
        PMAX(NEL,18)=TIME
 95     CONTINUE

      RETURN

      END
```

```
       DATA SET MDCNEQ      AT LEVEL 001 AS OF 83/02/19
  SUBROUTINE MDCNEQ(ENEW,GNEW,POIS,G)

  IMPLICIT REAL *8(A-H,O-Z)

  DIMENSION G(4,4)

       DO 100 I=1,4
       DO 100 J=1,4
100    G(I,J)=0.0

       CONST=(1.-2.*POIS)
       LAMBDA=(2.*POIS*GNEW)/CONST
       C1=LAMBDA+2.*GNEW
       C2=LAMBDA
       C3=GNEW
       G(1,1)=C1
       G(1,2)=C2
       G(1,3)=G(1,2)
       G(3,3)=G(1,1)
       G(2,2)=C1
       G(2,3)=G(1,3)
       G(4,4)=C3


        DO 200 I=1,4
        DO 200 J=1,4
        G(J,I)=G(I,J)
200     CONTINUE

        WRITE(17) ((G(I,J),J=1,4),I=1,4)

        RETURN
        END
```

```
          DATA SET MIDEP       AT LEVEL 009 AS OF 83/02/11
      SUBROUTINE MIDEP (TAU,DEPS,DP)
      IMPLICIT REAL*8(A-H,O-Z)

         FORMS THE ELASTO-PLASTIC MATERIAL MATRIX


      COMMON/EL/IND,ICOUNT
      COMMON /VMISES/A2,B2,C2,D2,A1,B1,C1,D1,YLD,BM,ISR,IST
           COMMON/MATMOD/STRESS(9), STRAIN(4),C(4,4),IPT,NEL
      COMMON/CHORDA/WP
      DIMENSION TAU(4),DEPS(4),DP(16)


           COMMON/SORIN/TT(5,12),XP(16),INDNB,ITYP2D



      SM = (TAU(1)+TAU(2)+TAU(4))/3.
      SX = TAU(1) - SM
      SY = TAU(2) - SM
      SS = TAU(3)
      SZ = TAU(4) - SM
      HPRIME=2.*D1/3.
      BETA=1.5/YLD/YLD/(1.+HPRIME/A1)

      IF (ITYP2D - 1) 10,11,12
   10 WP= SX*DEPS(1) + SY*DEPS(2) + SS*DEPS(3) + SZ*DEPS(4)
      GO TO 15

   11 WP= SX*DEPS(1) + SY*DEPS(2) + SS*DEPS(3)
      GO TO 15

   12 DP1= A1 * (C1 - BETA*SX*SZ)
      DP2= A1 * (C1 - BETA*SY*SZ)
      DP3= A1 * (   - BETA*SS*SZ)
      DP4= A1 * (B1 - BETA*SZ*SZ)

      DEPS(4)= (-DP1*DEPS(1)-DP2*DEPS(2)-DP3*DEPS(3))/DP4
      WP= SX*DEPS(1) + SY*DEPS(2) + SS*DEPS(3) + SZ*DEPS(4)

   15 IF (WP.LT.0.) BETA=0.

      DP ( 1) = A1 * (B1 - BETA*SX*SX)
      DP ( 2) = A1 * (C1 - BETA*SX*SY)
      DP ( 3) = A1 * (   - BETA*SX*SS)
      DP ( 4) = A1 * (C1 - BETA*SX*SZ)
      DP ( 5) = DP (2)
      DP ( 6) = A1 * (B1 - BETA*SY*SY)
      DP ( 7) = A1 * (   - BETA*SY*SS)
      DP ( 8) = A1 * (C1 - BETA*SY*SZ)
      DP ( 9) = DP (3)
      DP (10) = DP (7)
      DP (11) = A1 * (.5 - BETA*SS*SS)
      DP (12) = A1 * (   - BETA*SZ*SS)



      DP (13) = DP (4)
```

```
      DP (14) = DP (8)
      DP (15) = DP(12)
      DP (16) = A1 + (B1 - BETA*SZ*SZ)


        IF(ITYP2D.EQ.0.OR.ITYP2D.EQ.1) RETURN

      PLANE STRESS / MODIFY DP MATRIX

      DO 120 I=1,3
      A=C(I,4)/C(4,4)
      DO 120 J=1,3
      C(I,J)=C(I,J) - C(4,J)*A
  120 C(J,I) = C(I,J)
      IF (WP.LT.0.0) DEPS(4)=D2*(DEPS(1) + DEPS(2))
      STRAIN(4)=STRAIN(4) + DEPS(4)


      RETURN
      END
```

```
      DATA SET MODCNV      AT LEVEL 003 AS OF 82/03/16
      SUBROUTINE MODCNV(NEL,IPT,D)

      IMPLICIT REAL*8(A-H,O-Z)


      DIMENSION  D(4,4)

      COMMON/NANCY/CM(12000)

      COMMON/CONTL/DT,HED(8),RADN,NASIM,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS

C
      COMMON/HOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT


      DO  5200 I=1,4
      DO  5200 J=1,4
      D(I,J)=0.0
 5200 CONTINUE

      NU=(NEL-1)*NPT*15+(IPT-1)*15


      D(1,1)=CM(NU+1)
      D(1,2)=CM(NU+2)
      D(1,3)=CM(NU+3)
      D(1,4)=CM(NU+4)
      D(2,2)=CM(NU+5)
      D(2,3)=CM(NU+6)
      D(2,4)=CM(NU+7)
      D(3,3)=CM(NU+8)
      D(3,4)=CM(NU+9)
      D(4,4)=CM(NU+10)


      DO  520  I=1,4
      DO  520  J=1,4
  520 D(J,I)=D(I,J)

       IF(IPLNAX.NE.-1) GO TO 110
       D(1,1)=D(1,1)-D(3,1)*D(1,3)/D(3,3)
       D(1,2)=D(1,2)-D(3,2)*D(1,3)/D(3,3)
       D(1,4)=D(1,4)-D(3,4)*D(1,3)/D(3,3)
       D(2,2)=D(2,2)-D(3,2)*D(2,3)/D(3,3)
       D(2,4)=D(2,4)-D(3,4)*D(2,3)/D(3,3)
       D(4,4)=D(4,4)-D(3,4)*D(4,3)/D(3,3)
       DO 130 I=1,4
       DO 140 J=1,4
  140  D(J,I)=D(I,J)
       D(1,3)=0.0
       D(3,1)=0.0
  130 CONTINUE

  110  CONTINUE
      WRITE(6,102)
  102 FORMAT(5X,'  PRINT IN MODCNV.FOR')
       WRITE(6,103) ((D(I,J),J=1,4),I=1,4)
  103 FORMAT(5X,4E10.3)
```

```
        DATA SET MODCON      AT LEVEL 020 AS OF 83/01/27
    SUBROUTINE MODCON(C,NEL,PG,GSTRES)                                00001*19
    IMPLICIT REAL*8(A-H,O-Z)                                          00002
                                                                      00003**2
                                                                      00004**2
    COMMON/  NANCY/  CM(12000)                                        00005**2
        COMMON/MATMOD/STRESS(9),STRAIN(4),CC(4,4),IPT                 00006*20
    COMMON/HOBBY/NES,NUMMAT,NC,LOOP,NDRN,NINT,IOWA,NPT                00007*13
                                                                      00008**3
                                                                      00009**3
    DIMENSION C(4,4)                                                  00010**4
                                                                      00011**2
                                                                      00012**2
                                                                      00013*19
                                                                      00014**2
                                                                      00015**2
                                                                      00016*11
                                                                      00017*11
        DIMENSION GSTRES(9)                                           00018*20
                                                                      00019*13
                                                                      00020*13
                                                                      00021*14
    ND=(NEL-1)*NPT*15+(IPT-1)*15                                      00022*14
    CM(ND+1)=C(1,1)                                                   00023**2
    CM(ND+2)=C(1,2)                                                   00024**2
    CM(ND+3)=C(1,4)                                                   00025**2
    CM(ND+4)=C(1,3)                                                   00026**2
    CM(ND+5)=C(2,2)                                                   00027**2
    CM(ND+6)=C(2,4)                                                   00028**2
    CM(ND+7)=C(2,3)                                                   00029**2
    CM(ND+8)=C(4,4)                                                   00030**2
    CM(ND+9)=C(4,3)                                                   00031**2
    CM(ND+10)=C(3,3)                                                  00032**2
                                                                      00033**2
                                                                      00034**2
    ***************    UPDATED  STRESS  VECTOR  FOR  RESIDUAL  LOAD  V00035**2
     CALCULATION   IN   LOAD  *************************************   00036**2
                                                                      00037**2
    CM(ND+11)=GSTRES(1)                                               00038*11
    CM(ND+12)=GSTRES(2)                                               00039*11
    CM(ND+13)=GSTRES(3)                                               00040*11
    CM(ND+14)=GSTRES(4)                                               00041*11
    CM(ND+15)=PG                                                      00042*19
                                                                      00043**2
     WRITE(6,101)
101  FORMAT(5X,' PRINT IN MODCON.FOR ')
     WRITE(6,102)((C(I,J),J=1,4),I=1,4)
102  FORMAT(5X,4E10.3)
     WRITE(6,104) (CM(I),I=1,15)
104  FORMAT(2X,12E10.3)
                                                                      00044*18
                                                                      00045*18
                                                                      00046**2
    RETURN                                                            00047
    END                                                               00048
```

```
          DATA SET MODES      AT LEVEL 007 AS OF 83/02/24
     SUBROUTINE MODES (NEQ,MBAND,NBLOCK,NEQB,NF,MTOT,IFPR,IFSS,RTOL,
    *NITEM,COFQ,NN15,N015)
     IMPLICIT REAL*8(A-H,O-Z)

     CALLS:  SECNTD,SBLOCK,SSPCEB
     CALLED BY:  SOLEIG

   PROGRAM TO COMPUTE SMALLEST EIGENVALUES AND ASSOCIATED VECTORS IN
   THE GENERALIZED EIGENVALUE PROBLEM
                    A*V=RT*B*V  (A POS DEF,B DIAG NONNEG DEF)


     COMMON /SOL/ANORM, IDUM(5),NEIG,NAD,NVV,NFO
         COMMON/TAPES/NSTIF,NRED,NL,NR,NT,NMASS,NUMBER
     COMMON A(1)
     COMMON/SWAMI/ N30,N31,N32,N33


      NSTIF=4
      NMASS=9
      NRED=10
      NL=2
      NR=3
     ********NT  WAS  ORIGINALLY  7  CHANGED  BECAUSE  OF  WAT4 ********
     ********CHECK  THIS  TAPE  NUMBER  AS  AGAINST  OF  TAPE NO.  7***
       NT=8
     ******************************************************************

     PRINT EIGENPROBLEM SUMMARY

     WRITE (6,1000) NEQ,MBAND,NBLOCK,NEQB,NF

      IF (NEIG.GT.0) GO TO 300

     D E T E R M I N A N T   S E A R C H

     IF(NVV.GE.NF) GO TO 110
     WRITE (6,1010) NF,NVV
     STOP
 110 CONTINUE
      NIM=3
      NVM=6
      NC=NF+NIM
      NCA=NEQ*MAX0(MBAND,NC)
     ***************CHANGED  ********************
     N1=NN15+1
     N2=N1+NCA
      N3=N2+NFO
      N4=N3+NEQ
      N5=N4+NFO
      N6=N5+NFO
      N7=N6+NEQ*NVM
      N8=N7+NEQ*NVM
      N9=N8+NC
      N10=N9+NC
      N11=N10+NC
      N12=N11+NC

     N015=N12+NF
```

```
200    CALL  SFCNID(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9
     1),A(N10),A(N11),A(N12),NEQ,MBAND,NF,NC,IFPR,ANORM,COFQ)
       GO TO 600

      S U B S P A C E   I T E R A T I O N

300    NWA=NEQB*MBAND
       NV=2*NF
       IF (NF.GT.8) NV=NF+8
       IF (NAD.NE.0) NV=NAD
       IF (NVV.GE.NV) GO TO 310
       WRITE (6,1010) NV,NVV
       STOP
310    NWV=NV*NEQB
       NTB=(MBAND-2)/NEQB+1
       IF (NTB.GE.NBLOCK) NTB=NBLOCK-1
       NWVV=NWV*(NTB+1)

      CHECK FOR USE OF GIVEN STARTING ITERATION VECTORS

      IF (NFO.LE.0) GO TO 600
      REWIND 10
      READ (10) NEQO,NBLOKO,NEQBO,MBANDO,N10,NFO
      N2=1+NEQBO*NFO
      N3=N2+NEQB*NV


600    RETURN

1000  FORMAT (//// 40H SOLUTION IS SOUGHT FOR FOLLOWING EIGENPROBLEM,//
    1           / 37H NUMBER OF EQUATIONS                  =,I5 //
    2             37H HALF BANDWIDTH OF STIFFNESS MATRIX =,I5 //
    3             37H NUMBER OF EQUATION BLOCKS           =,I5 //
    4             37H NUMBER OF EQUATIONS PER BLOCK       =,I5 //
    5             37H NUMBER OF EIGENVALUES REQUIRED      =,I5 // )
1010 FORMAT (/// 32H0***ERROR   SOLUTION TERMINATED., /
    1          12X,40HNUMBER OF NON-ZERO MASSES REQUIRED      =, I5 /
    2          12X,40HNUMBER OF EXISTING MASSES IN THE MODEL =, I5 )

       END
```

```
        DATA SET MULT        AT LEVEL 001 AS OF 81/07/03
    SUBROUTINE MULT (W,A,V,NN,MA)
    CALLED BY : SECNTD

    IMPLICIT REAL*8(A-H,O-Z)

    DIMENSION A(1),W(1),V(1)

    NM=NN*(MA -1)
    NMA=NN - MA + 1
    DO 20 I=1,NN
    W(I)=0.0
    K=I - 1
    IF (NMA -1) 10,15,15
 10 NM=NM - NN
 15 IL=NM + 1
    DO 20 J=1,IL,NN
    K=K + 1
 20 W(I)=W(I) + A(J)*V(K)

    IF (MA -1) 30,100,30

 30 KK=NN
    DO 40 I=2,MA
    I1=I -1
    KK=KK + NN
    KJ=KK
    DO 40 J=1,I1
    KJ=KJ -NN
 40 W(I)=W(I) + A(KJ + J)*V(J)
    IF (MA.EQ.NN) GO TO 100
    MA1=MA + 1
    IJ=1
    DO 50 I=MA1,NN
    KJ=KK
    IJ=IJ + 1
    I1=I -1
    DO 50 J=IJ,I1
    KJ=KJ - NN
 50 W(I)=W(I) + A(KJ + J)*V(J)

100 RETURN
    END
```

```
      SUBROUTINE NROOT(M,A,B,XL,X)
C
      DOUBLE PRECISION A,B,XL,X,SUMV
      DIMENSION A(1), B(1), XL(1), X(1)
C
      K = 1
      DO.100 J=2,M
      L = M*(J-1)
      DO 100 I=1,J
      L=L+1
      K=K+1
100   B(K)=B(L)
C
C     THE MATRIX B IS A REAL SYMMETRIC MATRIG.
C
      MV=0
      CALL EIGEN (B,X,M,MV)
C
C     FORM RECIPROCALS OF SQUARE ROOT OF EIGENVALUES.  THE RESULTS
C     ARE PREMULTIPLIED BY THE ASSOCIATED EIGENVECTORS.
C
      L=0
      DO 110 J=1,M
      L=L+J
110   XL(J)=1.0/DSQRT(DABS(B(L)))
      K=0
      DO 115 J=1,M
      DO 115 I=1,M
      K=K+1
115   B(K)=X(K)*XL(J)
C
C     FORM (B**(-1/2))PRIME * A * (B**(-1/2))
C
      DO 120 I=1,M
      N2=0
      DO 120 J=1,M
      N1=M*(I-1)
      L=M*(J-1)+I
      X(L)=0.0
      DO 120 K=1,M
      N1=N1+1
      N2=N2+1
120   X(L)=X(L)+B(N1)*A(N2)
      L=0
      DO 130 J=1,M
      DO 130 I=1,J
      N1=I-M
      N2=M*(J-1)
      L=L+1
      A(L)=0.0
      DO 130 K=1,M
      N1=N1+M
      N2=N2+1
130   A(L)=A(L)+X(N1)*B(N2)
C
C     COMPUTE EIGENVALUES AND EIGENVECTORS OF A
C
      CALL EIGEN (A,X,M,MV)
      L = 0
      DO 140 I=1,M
```

```
      L=L+1
140   XL(I)=A(L)
C
C     COMPUTE THE NORMALIZED EIGENVECTORS
C
      DO 150 I=1,M
      N2=0
      DO 150 J=1,M
      N1=I-M
      L=M*(J-1)+1
      A(L)=0.0
      DO 150 K=1,M
      N1=N1+M
      N2=N2+1
150   A(L)=A(L)+B(N1)*X(N2)
      L=0
      K=0
      DO 180 J=1,M
      SUMV=0.0
      DO 170 I=1,M
      L=L+1
170   SUMV=SUMV+A(L)*A(L)
175   SUMV=DSQRT(SUMV)
      DO 180 I=1,M
      K=K+1
180   X(K)=A(K)/SUMV
133   CONTINUE
      RETURN
      END
```

```
      DATA SET OUT        AT LEVEL 060 AS OF 83/03/05
  SUBROUTINE  OUT(X1,X0,X2,VEL,AC,A0,B,ID,DS,AMAX,PMAX,NUMNP,NUMEL,  00001
                                                                     00002*14
 *NC,AM,EK,AN,AN1,END,IP,EC,NEQ,RE,ITR)                              00003*27
                                                                     00004*14
  IMPLICIT  REAL *8(A-H,O-Z)                                         00005
                                                                     00006*14
                                                                     00007**9
                                                                     00008**9
   DIMENSION  X1(1),X0(1),X2(1),VEL(1),AC(1),A0(1),B(1),ID(NUMNP,1),00009
 *DS(4,3),AMAX(NUMNP,12,2),PMAX(NUMEL,1),AM(1),EK(1),AN(NNSC,1)     00010*47
 *,AN1(NYP,1),END(1)                                                 00011*11
                                                                     00012**9
        DIMENSION SIG(9),STRAIN(4)                                   00013*42
                                                                     00014**9
     DIMENSION RE(1)                                                 00015*35
     DIMENSION DISP(12)                                              00016*32
                                                                     00017*32
  COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA00018
 *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQB,MBANDB                     00019**3
                                                                     00020*15
      COMMON/PETER/AREA,WIDTH,HOR,VERT,ACCL                          00021*46
      COMMON/TAPES/NOU(6),NUMBER                                     00022*54
      COMMON/NEVER/WEIGHT,SHEAR                                      00023*54
                                                                     00024*46
      COMMON/BOBBY/KKS,NUMMAT,NG,LOOP,NDRN,NINT,IDWA,NPT             00025*45
                                                                     00026*15

   COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                     00027*15
 *******   PLOTTING   AND   PRINTING   ARRAY   CHECK   ?   **************00028*15
      COMMON/KALI/JPT,KEL,JPLOT(200),KELMT(200)                      00029*31
      COMMON/EL/IND,ICOUNT                                           00030*30
                                                                     00031*30
                                                                     00032*27
                                                                     00033*27
  COMMON/INTEGE/TETA,BETA,GAMA,KTOL,NGAMA,NPUNCH,NEQUIB,ITEMX        00034*28
  COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC                             00035
      COMMON/PRIM/ISTRES,MPRINT,JPRINT,KPRINT,IP                     00036*47
                                                                     00037**9
      COMMON/GORA/IFLAG,IGR                                          00038*46
                                                                     00039**9
       COMMON/BENNY/BG(5,200,4,4),NCOND

                                                                     00040*19
                                                                     00041*13
      COMMON/HARVEY/UACC(700),UVEL(700),UDISPL(700)                  00042*56
                                                                     00043*56
                                                                     00044*22

      WRITE(6,10)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
 10   FORMAT(5X,10I5)
                                                                     00045*22
                                                                     00046*23
                                                                     00047*23
                                                                     00048*23
******    BRANCH   OUT   FOR   WORKING   ARRAY   INITIALIZATION   IN   ELT2D00049*13
      IF(             IND.NE.-1) GO TO 200                           00050*31
                                                                     00051*10
```

```
                                                                       00052*18
          CALL STRSPR(B,TIME,FMAX,NUMEL,NC,AM,EK,AN,ANI,EKJ,LC,ITR,RE)  00053*30
                                                                       00054*30
        WRITE(6,10) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
                                                                       00055*30
                                                                       00056*27
                                                                       00057*27
      RETURN                                                           00058*13
 200 CONTINUE                                                          00059*13
                                                                       00060*13
          IF(KKS.EQ.3.AND.IFLAG.EQ.0) GO TO 592                        00061*38
                                                                       00062*13
5550 DO   591   I=1,NEQ                                                00063
     VEL(I)=X1(I)                                                      00064
     AC(I)=X2(I)                                                       00065
     B(I)=X0(I)                                                        00066
 591 CONTINUE                                                          00067
                                                                       00068
          GO TO 595                                                    00069*38
                                                                       00070*38
                                                                       00071*38
 592      CONTINUE                                                     00072*38
                                                                       00073*38
          DO 594 I=1,NEQ                                               00074*38
           B(I)=UDISPL(I)                                              00075*40
 594     CONTINUE                                                      00076*38
                                                                       00077*38
         WRITE(6,597)
 597     FORMAT(5X,'  PRINT IN OUT ACCLN ')
          WRITE(6,598) (UACC(I),I=1,NEQ)
          WRITE(6,598) (UVEL(I),I=1,NEQ)
          WRITE(6,598) (UDISPL(I),I=1,NEQ)
 598     FORMAT(5X,12E10.3)
                                                                       00078*38
 595     CONTINUE                                                      00079*38
         **************          CHECK THIS ARRAY TRANSFER FOR KKS=3 ****00080*38
                                                                       00081*38
           IF(IFLAG.EQ.0) GO TO 229                                    00082*40
                                                                       00083*38
                                                                       00084*38
                                                                       00085
*****************************************************END  ****************00086
                                                                       00087*12
      DO  228  I=1,NUMNP                                               00088
       DO  227  J=1,2                                                  00089
      JJ=ID(I,J)                                                       00090
       IF(JJ)   227,227,226                                            00091
 226       CONTINUE                                                    00092*40
           IF(NDYN.EQ.2) AC(JJ)=X2(JJ)+ACCL                            00093*40
 227 CONTINUE                                                          00094
 228 CONTINUE                                                          00095
                                                                       00096*38
                                                                       00097*38
 229       CONTINUE                                                    00098*40
                                                                       00099*38
                                                                       00100*59
                                                                       00101*59
  68    FORMAT(5X,'               TIME  STEP (NC)          = ',I5/     00102*47
   *            5X,'               ITERATION NO.(ITR)       = ',I5/     00103*47
   *            5X,'               STRESS PRINT(ISTRPR)     = ',I5/     00104*47
```

```
     *          5X,'               DISPL. PRINT(NPRINT)    = ',I5/
     *          5X,'               CONTOUR PRINT(ISTART)   = ',I5/
     *          5X,'               COUNTER FOR STRESS      = ',I5/
     *          5X,'               COUNTER FOR DISPL.      = ',I5/
     *          5X,'               COUNTER FOR CONTOUR     = ',I5/
     *          5X,'               NODE OR ELEMT PRINT(IP)= ',I5/
     *          5X,'               CONVERGANCE FLAG(IFLAG)= ',I5/)


      TIME=DD*NC

      IF(NC)355,3333,355




  355 CONTINUE

 3333 CONTINUE
      IF(KKS.EQ.3.AND.IFLAG.EQ.0)  GO  TO  245

      IF(KKS.EQ.2.AND.LOOP.NE.NUMBER) GO TO 350

              IF(NC.NE.0.AND.MPRINT.LT.NPRINT)  GO TO  350

      IF(NPUNCH.NE.0) GO TO 351
      WRITE(6,2017) TIME


      WRITE(6,2010) NC,ITR

 2010 FORMAT(//// 37H EQUILIBRIUM ITERATION IN TIME STEP = ,I5//
     *                37H NUMBER OF ITERATIONS             = ,I5  /)
      WRITE(6,2011)

 ***************     DO  LOOP  STARTED  HERE  FOR  ALL  PRINTING  AND
      **********     SAVING  INFORMATION  FOR  ALL  PLOTTING         *****

  351  WRITE(55,991) TIME


  350 CONTINUE





      IP=1
      DO  240  I=1,NUMNP




      DO  9900  IM=1,4
      DO  9900  JM=1,3
 9900 DS(IM,JM)=0.0
      DO  235  J=1,4
      K=ID(I,J)
```

```
      IF(K)235,235,232                                                  00156
  232 DS(J,1)=D(K)                                                      00157
      DS(J,2)=VEL(K)                                                    00158
      DS(J,3)=AC(K)                                                     00159
  235 CONTINUE                                                          00160
                                                                        00161*36
         IF(NC.EQ.0) GO TO 236                                          00162*45
                                                                        00163*36
         IF(MPRINT.LT.NPRINT) GO TO 241                                 00164*36
                                                                        00165*36
                                                                        00166*45
  236    CONTINUE                                                       00167*45
                                                                        00168*36

         IF(NPUNCH.EQ.0)WRITE(6,2012)  I,DS                             00169
                                                                        00170
  241    CONTINUE                                                       00171*37
                                                                        00172*47
               DISP( 1)=DS(1,1)                                         00173*48
               DISP( 2)=DS(2,1)                                         00174*48
               DISP( 3)=DS(3,1)                                         00175*48
               DISP( 4)=DS(4,1)                                         00176*48
               DISP( 5)=DS(1,2)                                         00177*48
               DISP( 6)=DS(2,2)                                         00178*48
               DISP( 7)=DS(3,2)                                         00179*48
               DISP( 8)=DS(4,2)                                         00180*48
               DISP( 9)=DS(1,3)                                         00181*48
               DISP(10)=DS(2,3)                                         00182*48
               DISP(11)=DS(3,3)                                         00183*48
               DISP(12)=DS(4,3)                                         00184*48
                                                                        00185*47
                                                                        00186*47
                                                                        00187*32
       *******   DETERMINE MAXIMUM RESPONSE ******                      00188*32
                                                                        00189*32
        DO 256 J=1,12                                                   00190*32
        BB=DABS(DISP(J))                                                00191*32
        IF(BB-AMAX(I,J,1)) 256,257,257                                  00192*32
  257   AMAX(I,J,1)=BB                                                  00193*32
        AMAX(I,J,2)=TIME                                                00194*32
  256   CONTINUE                                                        00195*32
                                                                        00196*32
                                                                        00197*32
                                                                        00198*45
                                                                        00199*45
          IF(NC.EQ.0) GO TO 237                                         00200*45
          IF(MPRINT.LT.NPRINT) GO TO 240                                00201*45
                                                                        00202*45
  237     CONTINUE                                                      00203*45
                                                                        00204*51
                                                                        00205*51
                                                                        00206*51
               IF(IP.GT.JPT) GO TO 240                                  00207*51
                                                                        00208*51
                                                                        00209*51
        LG=JPLOT(IP)                                                    00210*32
        IF(LG.NE.1) GO TO 240                                           00211*32
                                                                        00212*32
                                                                        00213*50
           IF(KRS.EQ.2.AND.LOOP.NE.NUMBER) GO TO 240                    00214*54
```

```
          ANG=0.0                                                       00215*56
          IF(DISP(1).EQ.0.0.AND.DISP(2).EQ.0.0) GO TO 15                00216*56
                                                                        00217*56
                                                                        00218*50
                                                                        00219*32
          ******   CHECK  ANG  DOBLE PRECISSION ********                00220*46
              ANG=RADN*DATAN2(DISP(1),DISP(2))                          00221*47
  15      CONTINUE                                                      00222*56
                                                                        00223*56
                                                                        00224*47
         IF(NPUNCH.NE.0) WRITE(55,2008)  I,HOR,VERT,DISP(1),DISP(2),ANG,00225*46
     *DISP(5),DISP(9)                                                   00226*46
2008     FORMAT(I3,7E11.4/)
                                                                        00227*46
                                                                        00228*32
                                                                        00229*52
                                                                        00230*59
                                                                        00231*60
                                                                        00232*52
  90        FORMAT(16I5)                                                00233*52
                                                                        00234*52
                                                                        00235*30
                                                                        00236*50
 236    CONTINUE                                                        00237*50
                                                                        00238*50
                                                                        00239
                                                                        00240
                                                                        00241
     IP=IP+1                                                            00242*15
 240 CONTINUE                                                           00243
                                                                        00244*15
      **********   CALL  SUBROUTINE  SAVEPD  AT  THIS  STAGE            00246*15
                                                                        00247*15
                                                                        00249*14
                                                                        00250*14
 242    CONTINUE                                                        00251*36
     **********  STRESSS  OUTPUT  PTINTING   AND  INFORMATION  SAVING  *00252*14
                                                                        00253*14
                                                                        00254*14
                                                                        00255*14
3334 CONTINUE                                                           00256
 245 CONTINUE                                                           00257*16
     NNN=NC                                                             00258
                                                                        00259*15
     ******************   STRESS  OUTPUT   ROUTINE  TO  COMPUTE  STRESSE00260*15
     **********   ELEMENT  NODAL  DISPLACEMENTS     ********************00261*15
      WRITE(6,10)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
                                                                        00262*15
         CALL STRSPR(B,TIME,FMAX,NUMEL,NC,AM,EK,AN,ANI,EKJ,LC,ITR,RE)   00263*30
                                                                        00264*30
                                                                        00265*30
       WRITE(6,10) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
                                                                        00266*27
                                                                        00267*15
                                                                        00268*15
                                                                        00269*17
                                                                        00270*25
```

```
      ******    CHECK    FOR   ISTEPS   COUNTER   ****************                  00271*25
        I                                                                           00272*25
                                                                                    00273*29
                                                                                    00274*29
 250 CONTINUE                                                                       00275
                                                                                    00276
                                                                                    00277*32
 600 CONTINUE                                                                       00278
                                                                                    00279
      IF(NC.EQ.0)  GO  TO  800                                                      00280**8
                                                                                    00281*29
                                                                                    00282*29
              IF(KKS.EQ.3.AND.IFLAG.EQ.0) GO TO 700                                 00283*58
                                                                                    00284*29
                                                                                    00285*29
      TM=TM+DD                                                                      00286
 800 CONTINUE                                                                       00287**8

      IF(MPRINT.LT.NPRINT) GO TO 850

      MPRINT=0

                                                                                    00288*31
                                                                                    00289*31
          GO TO (850,850,850,850),KKS                                               00290*30


 850    CONTINUE


        IF(KKS.EQ.3.AND. NEQUIB.NE.0) GO TO 855

        IF(KKS.EQ.4.AND.NCOND.NE.1) GO TO 855

       CALL ASSEFF(FF,NUMEL,NEQ)                                                    00291*33
                                                                                    00292*33
                                                                                    00293*33
         WRITE(6,857)                                                               00294*34
 857 FORMAT(1H1/5X,'   PRINT OF RESIDUAL LOAD VECTOR IN OUT AT EACH                 00295*39
    + LOAD STEP  ****  '/)                                                          00296*39
                                                                                    00297*39
          WRITE (6,858) (RF(I),I=1,NEQ)                                             00298*34
 858 FORMAT(5X,6E12.5)                                                              00299*34
                                                                                    00300*39
          REWIND 21                                                                 00301*33
                                                                                    00302*30
                                                                                    00303*31
 855      CONTINUE                                                                  00304*33
                                                                                    00305*31
                                                                                    00306*30
                                                                                    00307*36
         NC=NC+1                                                                    00308*36
                                                                                    00309*19
      IF(NC-NCYCL)  700,900,900                                                     00310
                                                                                    00311

 900 CONTINUE
      IF(KKS.EQ.4) GO TO 700
```

```
      WRITE(6,2020) (I,(EMAX(I,J),J=1,18),I=1,NUMEL)               00312**3
                                                                   00313*45

      IF(NPUNCH.NE.0) WRITE(47,2030) (I,(EMAX(I,J),J=1,9),I=1,NUMEL)

2030 FORMAT(I5,8E9.2,3X/E9.2/)

        WRITE(6,2023)                                              00314*49
                                                                   00315*49
                                                                   00316*45
              DO 750 I=1,NUMNP                                     00317*47
              DO 750 K=1,2                                         00318*47
             WRITE(6,2021) I,(AMAX(I,J,K),J=1,12)                  00319*47

             IF(NPUNCH.NE.0) WRITE(46,2031) I,(AMAX(I,J,1),J=1,12)
2031         FORMAT(I3,7E11.4/5E10.3/)

 750          CONTINUE                                             00320*47
                                                                   00321*47
                                                                   00322*50
         IF(KKS.EQ.2.AND.LOOP.NE.NUMBER) GO TO 700                 00323*57
                                                                   00324*57
                                                                   00325*32
                                                                   00326*41
           **********    PUNCH  ON CARD FOR PLOTTING  *********8   00327*45
                                                                   00328*45
           IF(NPUNCH.EQ.0) GO TO 700                               00329*45
                                                                   00330*45
                                                                   00331*45
          NTERM=(NCYCL/NPRINT)+1                                   00332*45
          NSTRP=(NCYCL/ISTRPR)+1                                   00333*45
          NCONT=(NCYCL/ISTART)+1                                   00334*46
                                                                   00335*46
          WRITE(6,13) NTERM,NSTRP,NCONT                            00336*46
                                                                   00337*46
  13       FORMAT(1H1/5X,'    NO OF DISPL. PRINT  REQUEST  = ',I5/ 00338*46
   *       5X,           '    NO. OF STRESS PRINT REQUEST  = ',I5/ 00339*46
   *       5X,           '    NO. OF STRESS CONTR.REQUEST  = ',I5/) 00340*46
                                                                   00341*45
                                                                   00342*41
           IF(NPUNCH.NE.0) GO TO 700

             REWIND 55                                             00343*41
             REWIND 57                                             00344*41
             REWIND 18                                             00345*41
             REWIND 47
             REWIND 46
                                                                   00346*52
            WRITE(6,80)                                            00347*52
  80        FORMAT(1H1/'  *****    PRINT DISPL HISTORY FOR PUNCH **'/)00348*52
                                                                   00349*52
                                                                   00350*41
           DO 710 N=1,NTERM                                        00351*45
                                                                   00352*45
            STEP=(N-1)*NPRINT*DD
           READ(55,991) TIME
           WRITE(6,991)  TIME
 991       FORMAT(1X,' TIME STEP FOR DISP. OUTPUT   = ',F10.4)
```

```
              DO 710 K=1,JPT

        READ(55,2008)  I,HOR,VERT,DISP(1),DISP(2),ANG,DISP(5),DISP(9)
              WRITE(6,9)I,HOR,VERT,DISP(1),DISP(2),ANG,DISP(5),DISP(9)

 710            CONTINUE


2022        FORMAT(1H1/5X,I3,12E10.3)

   7            FORMAT(5X,I3,8E10.3)
   8          FORMAT(I5,8E9.2,3X)
  12          FORMAT(I3,7E11.4/8E10.3/)

   9          FORMAT(I3,7E11.4/)




                WRITE(6,85)
  85            FORMAT(1H1/'     *********  PRINT OF STRESS HISTORY  ***'/)

               DO 715 N=1,NSTEP

               STPS=(N-1)*ISTEPR*DD
             READ(57,992) TIME
             WRITE(6,992) TIME
 992         FORMAT(1X,' TIME STEP FOR STRESS OUTPUT   = ',F10.4/)

          DO 715 K=1,KEL



      READ(57,2032)NE,(SIG(I),I=1,2),SIG(4),(STRAIN(I),I=1,3),
     *      SIG(5),SIG(6),SIG(8),SM,SD,PG,SHM,EPSV,WP

2032    FORMAT(I3,7E11.4/8E10.3/)
      WRITE(6,11)NE,(SIG(I),I=1,2),SIG(4),(STRAIN(I),I=1,3),
     *      SIG(5),SIG(6),SIG(8),SM,SD,PG,SHM,EPSV,WP


 715      CONTINUE

  11  FORMAT(I3,7E11.4/8E10.3/)

                WRITE(6,95)
  95            FORMAT(1H1/'     *********  PRINT OF STRESS CONTOUR  ***'/)

                DO 730 N=1,NCONT
               STPC=(N-1)*ISTART*DD
             READ(18,992)TIME
             WRITE(6,992) TIME
 993         FORMAT(1X,' TIME STEP FOR CONTOUR OUTPUT   = ',F10.4/)
                DO 730 K=1,NUMEL

          READ(18,2033)NE,SIG(1),SIG(2),SIG(4),SIG(8),PG,SHM,EPSV,STRAIN(
     *4)
2033      FORMAT(I5,8E9.2,3X)

          WRITE(6,7)NE,SIG(1),SIG(2),SIG(4),SIG(8),PG,SHM,EPSV,STRAIN(4)
```

```
                                                                        00405*47
                                                                        00406*46
                                                                        00407*46
                                                                        00408*45
 730      CONTINUE                                                      00409*45
                                                                        00410*42

          WRITE(6,98)
  98      FORMAT(1H1/'  ***   PRINT OF MAX. DISPL    '/)
           DO 740 K=1,NUMNP
           KD=K
           READ(46,2031) (KD,(AMAX(K,J,1),J=1,12))
           WRITE(6,2031) (KD,(AMAX(K,J,1),J=1,12))
 740       CONTINUE

            WRITE(6,99)
  99        FORMAT(1H1/'  ***   PRINT OF MAX. DISPL. VEL ACCN. '/)
             DO 745 K=1,NUMEL
             KD=K
             READ(47,2030) (KD,(EMAX(K,J),J=1,9))
             WRITE(6,2030) (KD,(EMAX(K,J),J=1,9))
 745      CONTINUE

                                                                        00411*25
 700     CONTINUE                                                       00412*44
                                                                        00413*44
                                                                        00414*44
                                                                        00415*25
      RETURN                                                            00416
                                                                        00417
                                                                        00418
1009 FORMAT (3I5)                                                       00419
1010 FORMAT(I5,F10.0)                                                   00420
                                                                        00421
2011 FORMAT (                                                           00422
     *    4X,'NODE',17X,'DISPLACEMENTS',30X,'VELOCITY',30X,'ACCLERATION'/00423
     *4X,'NUMBER',5X,'UR',5X,'UZ',5X,'WR',5X,'WZ',8X,'UR',5X,'UZ',5X,   00424
     *  'WR',5X,'WZ',8X,'UR',5X,'UZ',5X,'UZ',5X,'WR'//)                 00425
2012 FORMAT(I5,12E10.3)                                                 00426
2016 FORMAT (1H1/'        NODAL POINT LOADS AT TIME     = ',F10.3///)   00427
2017 FORMAT (1H1/'        RESPONSE  AT  TIME           = ',F10.5//)     00428
2018 FORMAT (1H1/'         INITIAL  CONDITIONS          '//)            00429
2020 FORMAT (1H1/10X,'       MAX. STRESSES                           '/00430
     *'ELEMENT'/'NUMBER',3X,'SIGR',5X,'SIGZ',6X,'SIGT',5X,'SIGRZ',5X,'S 00431
     *IGMAX',4X,'SIGMIN',5X,'MEAN NORMAL',5X,'MAX. DEVIATOR',4X,'FF'/100 00432**3
     *(I6,9E10.3 /6X,9E10.3/))                                          00433**3
2023    FORMAT(1H1/ '    PRINT  OF DISPL.,VEL.,ACCLN., MAXIMA  TIME '/)00434*49
2021    FORMAT(5X,I6,12E10.3/)                                          00435*49
                                                                        00436*49
                                                                        00437*32
      END                                                               00438
```

```
           DATA SET OUTPUT      AT LEVEL 020 AS OF 83/02/24
      SUBROUTINE OUTPUT(NUMEL,NYP,NSC,NPJ,NPR,PSOL,SG,AN,SD,SN,CT,SU,
     *NEGS,IN,NUMNP)

      IMPLICIT REAL * 8(A-H,O-Z)

      DIMENSION PSOL(1),AN(NPJ,1),SD(NPJ,1),EP(3),ANI(14,3),AP(14

     *),SN(NPJ),IN(NUMNP,2)

      DIMENSION ST(20,12),TT(5,12),F(4,8),LM(16),PS(16,6),FACN(4)
      DIMENSION SP(3),Q(2)
      DIMENSION CT(NPJ,1),SU(1),RM(5),ZM(5)
      DIMENSION EF(4)

          COMMON/GRAND/COH(20),PHI(20),ITYPE(20)
          COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX

          COMMON/BENDT/SM(5,200,4,4),NCOND


      COMMON/TARA/LPC,N,M1,M2,NPT,KPT,NPRNT,MD,MM


      COMMON/BOBBY/KKS,NUMGT,NC,LOOP,NDRN,NINT,IDWA,NGT

      REWIND 10
      REWIND 55

      NDT=1
      IF(NCOND.EQ.1) NDT=4



      WRITE(6,500)
  500 FORMAT(1H1/ ' ****  PRINT  OF  SOLUTION VECTOR  FOR PROBLEM  ' )

      WRITE(6,600)  (PSOL(I),I=1,N)
  600 FORMAT(5X,10E12.5)


      PI=4.0*DATAN(1.0D0)
      RADN=180./PI
      RAD2=0.5*RADN
      REWIND  11

      WRITE(6, 400 )

      IF(KPT.EQ.2.AND.NPT.EQ.1.OR.KPT.EQ.2.AND.NPT.EQ.2) GO TO 350

      WRITE(6,700)
      DO  32  I=1,NUMEL
      WRITE(6,609)I


      DO  35  J=1,NYP
```

```
      DO  33 IP=1,NDT                                                   00053*12
                                                                        00054*10
      LX=J+(I-1)*NYP*NDT+(IP-1)*NYP                                     00055*10
                                                                        00056*10
      LP(IP)=LX                                                         00057*12
   33 CONTINUE                                                          00058*12
                                                                        00059*12
      IF(NCOND.NE.1) GO TO 38

      WRITE(6,608)J,PSOL((LP(1)+1)),PSOL((LP(2)+1)),                    00060*12

     *PSOL((LP(3)+1)),PSOL((LP(4)+1))                                   00061*12

      GO TO 35

   38 WRITE(6,608) J,PSOL((LP(1)+1))
   35 CONTINUE                                                          00062*11
                                                                        00063*10
   32 CONTINUE                                                          00064*10
                                                                        00065
                                                                        00066
      WRITE(6,698)                                                      00067
      DO  36  M=1,NUMEL                                                 00068**3
      DO  37 IP=1,NDT                                                   00069*10
                                                                        00070
      ********  RETRANSFER  AN  ARRAY  TO  ANI  ARRAY  *****************00071
      DO  200  J=1,NYP                                                  00072
      LX=J+(M-1)*NYP*NDT+(IP-1)*NYP                                     00073*10
      DO  100  I=1,NSC                                                  00074
      LY=I+(M-1)*NSC*NDT+(IP-1)*NSC                                     00075*10
  100 ANI(J,I)=AN(LY,LX)                                                00076
  200 CONTINUE                                                          00077
                                                                        00078
      DO  34  K=1,NSC                                                   00079
      EP(K)=0.0                                                         00080
      DO  30  J=1,NYP                                                   00081
      AP(J)=ANI(J,K)                                                    00082
      LX=J+(M-1)*NYP*NDT+(IP-1)*NYP                                     00083*10
   30 EP(K)=EP(K)+AP(J)*PSOL(LX+1)                                      00084


       IF(EP(K).NE.0.0) GO TO 34
       LY=K+(M-1)*NSC*NDT+(IP-1)*NSC
       DO 31 N=1,NPJ

       ANIT(N,LY)=0.0
       ANIT(LY,N)=0.0
   31 CONTINUE


   34 CONTINUE
      WRITE(6,699)M,(EP(K),K=1,NSC)                                      00085
          IF(NPUNCH.EQ.0) GO TO 39                                      00086*20
          IF(IP.NE.1)      GO TO 39                                     00087*20
          WRITE(18,40) M,(EP(K),K=1,NSC)                                 00088*20
   39     CONTINUE                                                      00089*20
                                                                        00090*10
   37 CONTINUE                                                          00091*10
   36 CONTINUE
```

```
    ********    PRINT HERE DISPLACEMENT  *******

      IF(KPT.EQ.2.AND.NPT.EQ.0) GO TO 300
 321    FORMAT(2I5)

      WRITE(6,301)
      DO 303 I=1,NUMNP
      DO 302 J=1,2
      U(J)=0.0
      LL=ID(I,J)
      IF(LL) 308,308,305
 305  U(J)=PSOL(LL+1+NPK)
 308  CONTINUE
 302  CONTINUE
      WRITE(6,304)I, (U(K),K=1,2)
      IF(NPUNCH.NE.0) WRITE(55,304) I,U(1),U(2)
 303  CONTINUE
      IF(KPT.EQ.1) GO TO 1000

                                                              00092*10
     K1=(NSC*I-2)+(M-1)*NSC*NGT
     K2=NSC*I+(M-1)*NSC*NGT
     SN(K1)=EP(1)

     SN(K2)=EP(2)

     SN(K3)=EP(3)
                                                              00093*10
                                                              00095
                                                              00096
 300 CONTINUE                                                 00097**5
 301 FORMAT(1H1/'OUTPUT TABLE.  .. NODAL DISPLACEMENTS '//
   *13X,4HNODE,9X,11HU = X-DISP.,9X,11HV = Y-DISP./)
 304 FORMAT(5X,I5,2E10.3)

    WRITE(6,695)                                              00098**3
    DO  265  I=1,NPJ                                          00100
    SUM=0.0                                                   00101
    DO  285  J=1,NPK                                          00102
 285 SUM=SUM+SD(I,J)*PSOL(J+1)                                00103
    SN(I)=SUM                                                 00104
 265 CONTINUE                                                 00105
                                                              00106**5


    GO  TO    370                                             00107**5
                                                              00108**5
                                                              00109**5
 350 DO  360  I=1,NPJ                                         00110**5
 360 SN(I)=PSOL(I+1)-PSOL(I+1+NPJ)                            00111**5
 370 CONTINUE                                                 00112**5
                                                              00113**5
    WRITE(6,612)
                                                              00114**5
    DO  290  M=1,NUMEL                                        00115**3
                                                              00116*10

                                                              00117*10
    DO  295 I=1,NPT                                           00118*11
```

```
                                                                      00119*10
      K1=(NSC*I-2)+(M-1)*NSC*NDI                                      00120*10
      K2=NSC*I+(M-1)*NSC*NDI                                          00121*10
      WRITE(6,699)M,(SN(K),K=K1,K2)                                   00122
          IF(NPUNCH.EQ.0) GO TO 49                                    00123*20
          IF(IP.NE.1)     GO TO 49                                    00124*20
          WRITE(7,40) M,(SN(K),K=K1,K2)                               00125*20
 49       CONTINUE                                                    00126*20
 40       FORMAT(I5,3E10.3)                                           00127*20
                                                                      00128*10
                                                                      00129*10
 295  CONTINUE                                                        00130*10
                                                                      00131*10
 290  CONTINUE                                                        00132
                                                                      00133**8
      CALL CHECK(CT,SN,SO,NPJ,MEQS)                                   00134*12
                                                                      00135**8
       GO TO 1000

      *********   CONVERT  STRESSES  TO  RADIAL  COORDINATES  **********00136**8
                                                                      00137**8
      WRITE(6,695)                                                    00138**8
                                                                      00139**8
                                                                      00140**8
      DO  800 M=1,NUMEL                                               00141*10
                                                                      00142**8
      READ(11) ST,TT,P,BM,ALFA,EM,RM,ZM,KS,KF,IMAT,BLK,EMU,YOUNG,POIS,  00143**8
                                                                      00144**8
     * VOL,PS,FACN                                                    00145**8
                                                                      00146*10
      DO  900 I=1,NDI                                                 00147*10
                                                                      00148**8
                                                                      00149**8
      K1=(NSC*I-2)+(M-1)*NSC*NDI                                      00150*10
      K2=NSC*I+(M-1)*NSC*NDI                                          00151*10
                                                                      00152**8
                                                                      00153**8
      SP(1)=SN(K1)                                                    00154**8
      SP(2)=SN(K1+1)                                                  00155**8
      SP(3)=SN(K1+2)                                                  00156**8
                                                                      00157**8
                                                                      00158**8
      THETA=DATAN(ZM(I)/RM(I))                                        00159*12
                                                                      00160**8
      HARV=DCOS(THETA)                                                00161**8
      DES=DSIN(THETA)                                                 00162**8
      SARTO=DSIN(2.*THETA)                                            00163**8
      SN(K1)=SP(1)*(HARV**2.)+SP(2)*(DES**2.)+SP(3)*(SARTO)           00164**8
      SN(K1+1)=SP(1)*(DES**2.)+SP(2)*(HARV**2.)-SP(3)*(SARTO)         00165**8
      SN(K1+2)= (SP(2)-SP(1))/2.*(SARTO)+SP(3)*(DCOS(2.*THETA))       00166**8
                                                                      00167*17
                                                                      00168*17
                                                                      00169*17
      RZ=DSQRT(RM(I)**2.+ZM(I)**2.)                                   00170*17
      PAL=(2.*CON(IMAT))                                              00171*17
       PAL2=2.*PAL                                                    00172*17
      SN(K1)=SN(K1)/PAL                                               00173*17
      SN(K1+1)=SN(K1+1)/PAL                                           00174*17
      SN(K1+2)=SN(K1+2)/PAL                                           00175*17
```

```
      IF(NCOND.EQ.1) GO TO 901

      RM(II)=RM(I+NGI)
      ZM(I)=ZM(I+NGI)
      RZ=DSQRT(RM(I)**2.+ZM(I)**2.)

  901 WRITE(6,699) M,(SN(K),K=K1,K2),RM(I),ZM(I),RZ

  900 CONTINUE
  800 CONTINUE

 1000 CONTINUE

      WRITE(6,610)PSOL(1)

  400 FORMAT(1H1,/////////////////////////////////
     *55X,' S H A K E D O W N A N A L Y S I S  O U T P U T T A B L E  '/
     *55X,54(1H*))
  700 FORMAT(37H1OUTPUT TABLE 1.. PLASTIC MULTIPLIERS //)
  609 FORMAT(10X,'ELEMENT N3. ',I5///10X,
     *' YIELD PLANE ',2X,' PLASTIC MULTIPLIER ')
  612 FORMAT(1H1/
     *5X,' ELEMENT STRESS       STRESS-XX      STRESS-YY       STRESS-XY'/
     *5X,' NUM '/5X,' IPT '/)

  608 FORMAT(10X,2HX(,I3,4H) = ,4E15.7)
  698 FORMAT(54H1OUTPUT TABLE 2.. PLASTIC STRAINS AT ELEMENT CENTROIDS ,
     *//1X,' ELEMENT ',10X,' EPSILON(X) ',10X,' EPSILON(Y) ',10X,' EPSIL
     *ON(XY) '/)
  699 FORMAT(3X,I5,(3(5X,E15.7)),3F10.3)
  695 FORMAT(1H1/' OUTPUT TABLE 3.. PLASTIC SELF EQUILIBRATING STRESSES
     * AT  THE  ELEMENT CENTROIS ',
     *//1X,' ELEMENT ',10X,' SIGMA(X) ',10X,10X,' SIGMA(Y) ',10X,' SIGMA
     *(XY) '/)

  610 FORMAT(42H1OUTPUT TABLE 4.. SHAKEDOWN LOAD FACTOR IS ,F15.5)

      RETURN
      END
```

```
          DATA SET PRAGER      AT LEVEL 008 AS OF 83/02/11
      SUBROUTINE PRAGER (TAU,DEPS,DP)
      IMPLICIT REAL*8(A-H,O-Z)

         FORMS THE ELASTO-PLASTIC MATERIAL MATRIX
         (FOR THE DRUCKER-PRAGER YIELD CRITERION)

      COMMON/EL/IND,ICOUNT
            COMMON/MATMOD/STRESS(9), STRAIN(4),C(4,4),IPT,NEL
      COMMON /DRPRAG/ A1,B1,C1,A2,B2,C2,D2,G,BM,ALFA,YLD,ISR,IST

            COMMON/SORIN/TT(5,12),XP(16),INDNL,ITYP2D

      DIMENSION TAU(4),DEPS(4),DP(16)

      SM = (TAU(1)+TAU(2)+TAU(4))/3.
      SX = TAU(1) - SM
      SY = TAU(2) - SM
      SS = TAU(3)
      SZ = TAU(4) - SM
      SBAR=DSQRT( .5*(SX*SX+SY*SY+SZ*SZ) +SS*SS )
      QDQ=DSQRT (G +9.*BM*ALFA*ALFA)
      AA=G/SBAR/QDQ
      BB=3.*BM*ALFA/QDQ

      IF (ITYP2D - 1) 10,11,12
   10 DLAMDA= BB * (DEPS(1)+DEPS(2)+DEPS(4)) +
     1         AA * (SX*DEPS(1)+SY*DEPS(2)+SS*DEPS(3)+SZ*DEPS(4))
      GO TO 15

   11 DLAMDA= BB* (DEPS(1)+DEPS(2)) +
     1         AA * (SX*DEPS(1)+SY*DEPS(2)+SS*DEPS(3))
      GO TO 15

   12 SA= AA*SX + BB
      SB= AA*SY + BB
      SC= AA*SS
      SD= AA*SZ + BB

      DP1= B2 - SA*SD
      DP2= B2 - SB*SD
      DP3=     - SC*SD
      DP4= A2 - SD*SD

      DEPS(4)=(-DP1*DEPS(1)-DP2*DEPS(2)-DP3*DEPS(3))/DP4

      DLAMDA= BB * (DEPS(1)+DEPS(2)+DEPS(4)) +
     1         AA * (SX*DEPS(1)+SY*DEPS(2)+SS*DEPS(3)+SZ*DEPS(4))

      IF (DLAMDA.LT.0.) GO TO 16
      SX=SA
      SY=SB
      SS=SC
      SZ=SD
      GO TO 25

   15 IF (DLAMDA.GT.0.) GO TO 20
   16 AA=0.
      BB=0.
```

```
          DATA SET PRINTD     AT LEVEL 003 AS OF 83/02/16
      SUBROUTINE PRINTD(ID,F,B,NEQB,NUMNP,LL,NBLOCK,NEQ,NT,MQ)
      IMPLICIT REAL*8(A-H,O-Z)

      CALLED BY:  SOLEQ,SOLEIG,RESPEC

         DIMENSION B(NEQB,LL)
      DIMENSION ID(NUMNP,2),F(2,LL),DD(6)
      DATA Q11,Q21,Q12,Q22,Q13,Q23/'  LOAD','  CASE','EIGEN-','VECTOR',
     1     ' MODE ','NUMBER'/

      M=NEQ
      NN=NEQB*NBLOCK

      IF (MQ.EQ.2) GO TO 50
      IF (MQ.EQ.3) GO TO 55
      REWIND NT
      Q1=Q11
      Q2=Q21
      GO TO 60
   50 Q1=Q12
      Q2=Q22
      GO TO 60
   55 Q1=Q13
      Q2=Q23
      REWIND NT
      READ (NT)
   60 WRITE (6,2003) Q1,Q2

      N=NUMNP

      DO 500 KK=1,NUMNP

      I=2
      DO  250  II=1,2
      DO 100 L=1,LL
  100 F(I,L)=0.
      IF(M.GT.NN) GO TO 150
      IF (M.EQ.0) GO TO 150
      READ (NT) B
      NN=NN-NEQB
  150 IF(ID(N,I).LT.1)  GO  TO  250
      K=M-NN
      M=M-1

      DO 200 L=1,LL
  200   F(I,L)=B(K,L)
  250 I=I-1

      WRITE(6,2004) N,(L,(F(I,L),I=1,2),L=1,LL)

  500 N=N-1

      RETURN

 2003 FORMAT (1H1,30HN O D E   D I S P L A C E M E N T S / ,
     1          17HR O T A T I O N S,// 3X,4HNODE,2X,A6,2(12X,2HX-,12X,
     2          2HY-,12X,2HZ-), / 7H NUMBER,2X,A6,3(3X,11HTRANSLATION),
     3          3(6X,8HROTATION), / 1X)
 2004     FORMAT(1H0,I6,I8,2E14.5 / (7X,I8,2E14.5) )
```

```
       DATA SET PROP        AT LEVEL 010 AS OF 82/03/16
      SUBROUTINE  PROP(N,PRP,NUMMAT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/DELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017


      COMMON/BOBBY/KKS,NUMGAT,EC,LOOP,NDRN,NINT,IDWA,NPT   '


      DIMENSION PRP(1)




      IF(N.EQ.1)  WRITE(6,2000)  MODEL,NUMMAT,NCON,IDW
     +,NINT



      GO  TO  (1,2),MODEL

      ****************    MODEL  =  1  E L A S T O P L A S T I C ( VON M
    1 WRITE(6,2106) (PRP(I),I=1,NCON)


      GO  TO  3
      ******  MODEL  =  2  E L A S T O P L A S T I C  ( DRUCKER PRAGER )

    2 WRITE(6,2107) (PRP(I),I=1,NCON)

    3 CONTINUE


 2000 FORMAT(1H1/'                  'MATERIAL  DEFINATION = ',I5/
     *  '  EQ.1. E L A S T O P L A S T I C A N A L Y S I S (VON MISSES)'
     */
     *  ' EQ.2. E L A S T O P L A S T I C A N A L Y S I S (DRUCKER PRAGE
     *R)'/
     *           '               NUMBER OF DIFFERENT  MATERIAL  SETS = ',I5/
     *           '               NUMBER  OF  CONSTANTS  SETS  PER MATERIAL
     *  SETS  = ',I5/
     *           '               DIMENSION  OF  STORAGE  ARRAY  FOR  CENTER
     *  STRESSES   =  ',I5/
     *     '                  NUMBER OF INTEGRATION POINT = 'I5//)

 2106 FORMAT(1H ,4X,29HE .............( PRP(1) ).. =, E14.6/,
     *       1H,4X,29HVNU ............( PRP(2) ).. =, E14.6/,
     *       1H,4X,29HVYIELD..........( PRP(3) ).. =, E14.6/,
     *       1H,4X,29HV (HARDEN).....( PRP(4) ).. =, E14.6///)
 2107 FORMAT(1H,4X,29HE .............( PRP(1) ).. =,E14.6/,
     *       1H,4X,29HVNU .............( PRP(2) ).. =, E14.6/,
     *       1H,4X,29HVCOHESION .....( PRP(3) ).. =, E14.6/,
     *       1H,4X,29HV FRICTION ANGLE(PRP(4) ).. =, E14.6///)


      RETURN
      END
```

```
          DATA SET PRSTS      AT LEVEL 014 AS OF 83/02/25
      SUBROUTINE PRSTS(STRESS,PG,GSTRES)                                 00001**9
      IMPLICIT REAL*8(A-H,O-Z)                                           00002
                                                                         00003**5
                                                                         00004*10
                                                                         00005*12
           COMMON/HOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT            00006*12
                                                                         00007*10
                                                                         00008**5
          COMMON/MATMOD/PTRES(9),STRAIN(4),C(4,4),IPT                    00009*11
                                                                         00010*11
                                                                         00011**5
                                                                         00012**5
          DIMENSION GSTRES(9),STRESS(9)                                  00013*10
                                                                         00014**2
                                                                         00015**2
         IF(NDRN.NE.0) GO TO 100                                         00016*14
                                                                         00017**2
                                                                         00018**2
                                                                         00019**2
      GSTRES(1)=STRESS(1)+PG                                             00020**9
      GSTRES(2)=STRESS(2)+PG                                             00021**9
      GSTRES(3)=STRESS(4)+PG                                             00022**9
      GSTRES(4)=STRESS(3)                                                00023**4
                                                                         00024**2
           GO TO 200                                                     00025*13
                                                                         00026*10
 100      CONTINUE                                                       00027*10
                                                                         00028*10
          DO 150 I=1,2                                                   00029*10
 150      GSTRES(I)=STRESS(I)                                            00030*10
                                                                         00031*10
           GSTRES(3)=STRESS(4)                                           00032*10
                                                                         00033*10
          GSTRES(4)=STRESS(3)                                            00034*10
 200      CONTINUE                                                       00035*10
                                                                         00036**2
                                                                         00037**9
      RETURN                                                             00038**3
      END                                                                00039
```

```
        DATA SET REDVR      AT LEVEL 007 AS OF 83/02/24
     SUBROUTINE REDVR(A,VV,MAXA,NEQB,NWA,NEQ,NBLOCK,M1,MA,NCALL,KKS)
    IMPLICIT REAL*8(A-H,O-Z)

    CALLED BY:  SOLSTP

    THIS ROUTINE REDUCES AND BACK-SUBSTITUTES A SINGLE VECTOR STORED
    IN CORE USING A REDUCED MATRIX STORED IN BLOCK FORM.

    DIMENSION        A(NWA),VV(NEQ),MAXA(M1)

     COMMON/TAPES/NSTIF,NRED,NL,NR,IFIL(2),NUMBER

       COMMON/GORA/IFLAG,ITR
     INC=NEQB - 1
       *******   CHECK PASSING ARGUMENT ITR =0 FOR ITERATION ******

    MA1 = MA-1

    PERFORM FORWARD REDUCTION OF THE VECTOR
          GO TO (10,10,15,10),KKS


  10 IF(NBLOCK.EQ.1.AND.NCALL.GT.1) GO TO 22
    GO TO 30
  15 CONTINUE


          IF(ITR.NE.0) GO TO 22

  30  REWIND NRED

     READ (NRED) A,MAXA
  22 ISA = 1
    KSTART = 2
    KEND = NEQB

 500 N = 1
    DO 100 K=KSTART,KEND
    N = N+1
     KL=N + INC
     KU=MAXA(N)
     IF (KU-KL) 100,110,110
 110 KJ = K
     DO 130 KK=KL,KU,INC
     KJ=KJ - 1
130  VV(K)=VV(K) - A(KK)*VV(KJ)
100  CONTINUE

    IF(ISA.EQ.NBLOCK) GO TO 175
    KL = NEQB
    ML = KEND+1
    MR = MIN0(KEND+MA1,NEQ)
    N = NEQB
    DO 140 K=ML,MR
    N = N+1
     KL=KL + NEQB
     KU=MAXA(N)
     IF (KU-KL) 140,150,150
 150 KJ = KEND
```

```
       DO 170 KK=KL,KU,INC
      VV(K) = VV(K) - A(KK)*VV(KJ)
170    KJ=KJ - 1
140    CONTINUE

  175 KST = KSTART-1
      N = 0
      DO 200 K=KST,KEND
      N = N+1
      C = A(N)
       IF (C) 180,200,180
  180 VV(K) = VV(K)/C
200    CONTINUE
  205 IF(ISA.EQ.NBLOCK) GO TO 400
       READ (NRED) A,MAXA
       ISA=ISA+1
      KSTART = KSTART+NEQB
      KEND = MIN0(KEND+NEQB,NEQ)

       GO TO 500

      BACK-SUBSTITUTE REDUCED VECTOR (STORED IN CORE)

  400 IF(ISA.GT.1)
     *BACKSPACE NRED
       ISA=1
      NN = NEQ-(NBLOCK-1)*NEQB
      KEND = NEQ
      GO TO 645

  420 KEND = KEND-NN
      NN = NEQB

       KL=NEQB
      MR = MIN0(NEQ,KEND+MA1)
      ML = KEND+1
      N = NEQB
      DO 600 K=ML,MR
      N = N+1
       KL=KL+NEQB
       KU=MAXA(N)
       IF (KU-KL) 600,610,610
  610 KJ = KEND
       DO 620 KK=KL,KU,INC
       VV(KJ)=VV(KJ) - A(KK)*VV(K)
620    KJ=KJ - 1
600    CONTINUE

  645 N = NN
      K = KEND
      DO 640 L=2,NN
       KL=N + INC
       KU=MAXA(N)
       IF (KU-KL) 655,650,650
  650  KJ=K
       DO 690 KK=KL,KU,INC
       KJ=KJ - 1
690    VV(KJ)=VV(KJ) - A(KK)*VV(K)
  655  N=N - 1
  640 K = K-1
```

```
      DATA SET SOFDMX     AT LEVEL 031 AS OF 83/03/05
      SUBROUTINE SOFDMX(VOL,EM,IMAT,KS,KF,NEL,THICK)

      IMPLICIT REAL*8(A-H,O-Z)


      COMMON/CONTL/DT,HED(8),RADN,NASIM,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS
      COMMON/EMB/ S(16,16),GM(16),GH(16,16),SS(16,16),XP(16)
     *,XN(16),LM(16)
      COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),
     *ST(20,12),B(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8)
         COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
          COMMON/BOBBY/KKS,NUMMAT,NG,LOOP,NORN,NINT,IDWA,NPT
      COMMON/ASHK/C1,C2,C3,AL,ALF,FACC,FACT




      DIMENSION EK(8),EN(8),EM(1)





 **********   CALCULATE  FLUID  STIFFNESS  MATRIX    (  EVALUATE AT CE



      K=IMAT
      FAC=VOL*R
      FACM=FAC*EM(K)
      DO  505     I=1,8
      DO  505     J=1,8
      E(I,J)=E(I,J)+TT(1,I)*TT(1,J)*FACM
      C(I,J)=C(I,J)+E(I,J)*FACC
      SF(I,J)=SF(I,J)+E(I,J)*ALF
  505 CONTINUE
  952 CONTINUE





      ***************************************************************


         IF(NG.GT.0.AND.KKS.EQ.3) GO TO 6200
      IF(NEL.NE.0)  GO  TO  6200
      PRINT  999
  999 FORMAT(//)
      WRITE(6,4)
    4 FORMAT(1H1,/'     *****   UNDRAINED ANALYSIS    *******    '/)
      WRITE(6,225) ((SF(I,J),J=1,8),I=1,8)
       PRINT    999
      WRITE(6,66)
   66 FORMAT(5X,'COUPLING  FLUID-SOLID  MATRIX ')
      WRITE(6,225)     ((C(I,J),J=1,8),I=1,8)
       PRINT    999
      WRITE(6,69)
```

```
  69 FORMAT(5X,'FLUID  MATRIX')                                        00060
     WRITE(6,225)    ((E(I,J),J=1,8),I=1,8)                            00061
 225 FORMAT(10X,8F12.5)                                                00062
     PRINT  999                                                        00063
     WRITE(6,72)                                                       00064
  72 FORMAT(5X,'FLUID DISSIPATION  MATRIX')                            00065
      WRITE(6,225)  ((H(I,J),J=1,8),I=1,8)                             00066
6200 CONTINUE                                                          00067
                                                                       00068
                                                                       00069
                                                                       00070
***********************************************************************00071
                                                                       00072
                                                                       00073*23
 600 CONTINUE                                                          00074*23
                                                                       00075
     WRITE(51)  ((SF(I,J),J=1,8),I=1,8),((E(I,J),J=1,8),I=1,8),((C(I,J)00076**6
    *,J=1,8),I=1,8),IMAT,KS,KF,ALF,((H(I,J),J=1,8),I=1,8)              00077**8
    *,(XP(I),I=1,16),(XN(I),I=1,16)                                    00078*28
                                                                       00079*28
                                                                       00080*28
 500 CONTINUE                                                          00081*15
     RETURN                                                            00082
     END                                                               00083
```

```
      DATA SET SECNTD     AT LEVEL 001 AS OF 81/07/03
       SUBROUTINE  SECNTD (A,B,V,MAXA,N,VV,WW,ROOT,TIM,ERRVL,ERRVR,
     1NITE,M,MA,NROOT,NC,IFPR,ANORM,COFQ)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 TIM1,TIM2,TIM3

      CALLS:  BANDET
      CALLED BY:  MODES

       COMMON /TAPES/NSTIF,NRED,NL,NR,NT,NMASS
      ************CHANGED *****R********************************
       DIMENSION A(N,MA),B(N),V(1),W(1),VV(N,1),WW(N,1),ROOT(1),
     1TIM(1),ERRVL(1),ERRVR(1)
      ************CHANGED *****R********************************
       INTEGER NITE(1),MAXA(1)
      COMMON /EM/ AT(1000),IFIBB(3138)

        COMMON /ASMBLD/NMJIK
      THE FOLLOWING TOLERANCES ARE SET FOR THE IBM 370
       ACTOL=1.0D-04
       RCBTOL=1.D-05
       RTOL=1.0D-10
       RQTOL=1.0D-12
**     SCALE=2.0D0**200
SCALE = 1.7000D+38

       NTF=5
       IITEM=10
       NITEM=60
       NVM=6

       REWIND NT
       REWIND NMASS
       READ (NMASS) B

      ETA = 2.0
       NOV=0
       JR=1
       NSK=0
       NWA=N*MA
       ISC=1000

    FIND LOCATIONS FOR NEGATIVE ELEMENTS IN STARTING ITERATION VECTORS

      REWIND NSTIF
      READ (NSTIF) (A(I,1),I=1,N)
      DO 1 I=1,N
      AA=A(I,1)
      IF (AA.GT.0.) GO TO 1
      WRITE (6,1000) I,AA
      STOP
1     V(I)=B(I)/AA
      DO 2 J=3,NC
      RMAX=0.
      DO 3 I=1,N
      IF (V(I).LT.RMAX) GO TO 3
      RMAX=V(I)
      IMAX=I
3     CONTINUE
      NITE(J)=IMAX
```

```
2     V(IMAX)=0.

  CHECK FOR SINGLE DEGREE-OF-FREEDOM SYSTEM

       IF (N.GT.1) GO TO 5
      IF(B(1).GT.0.) GO TO 7
      WRITE(6,1180)
      STOP
    7 REWIND NSTIF
      READ(NSTIF) A(1,1)
      ROOT(1)=A(1,1)/B(1)
      NSCH=1
      A(1,1)=1.0D0/DSQRT(B(1))
      GO TO 950

    5 CONTINUE
        RA=0.0
        RR=0.0
        CALL BANDET (A,B,V,MAXA,N,NWA,RA,NSCH,DETA,ISC,1)
        FA=DETA
        FR=FA
        DETR=DETA

   CHECK FOR ZERO EIGENVALUE(S)
        WRITE(6,1002)  ANORM
1002 FORMAT(5X,F12.5)
        IF (A(N,1) .GT. ANORM)  GO TO 10
        WRITE (6,1009)
        STOP

   FIND LOWER BOUND ON SMALLEST EIGENVALUE
10      IF (IFPR.EQ.1)
     *  WRITE(6,1010)
        DO 100 I=1,N
100     W(I)=B(I)
        RT=0.0
        IITE=0
        KK=2
110     IITE=IITE+1
        DO 120 I=1,N
120     V(I)=W(I)
        CALL BANDET (A,B,V,MAXA,N,NWA,RA,NSCH,DETA,ISC,KK)
        KK=2
        RQT=0.0
        DO 130 I=1,N
130     RQT=RQT+W(I)*V(I)
        DO 180 I=1,N
180     W(I)=B(I)*V(I)
        RQB=0.0
        DO 140 I=1,N
140     RQB=RQB+W(I)*V(I)
        RQ=RQT/RQB
        IF (IFPR.EQ.1)
     *  WRITE (6,1004) RQ
       BS=DSQRT(RQB)
       TOL=DABS(RQ-RT)/RQ
        IF (TOL.LT.RCBTOL) GO TO 150
        DO 160 I=1,N
160     W(I)=W(I)/BS
        RT=RQ
```

```
      IF (IITF.LT.IITEM) GO TO 110

150   DO 170 I=1,N
170   V(I)=V(I)/RS
     RB=RQ*(1.0D0-DMIN1(1.0D-1,1.0D2*TOL))
      IS=0
230   CALL BANDET (A,B,V,MAXA,N,NWA,RB,NSCH,DETB,ISC,1)
      IF (IFPR.EQ.1)
    * WRITE (6,1020) RB,NSCH
      FB=DETB
      IF (NSCH .EQ. 0) GOTO 300
      IS=IS+1
      IF (IS.LE.NTF) GO TO 240
     WRITE (6,1030) NTF
      STOP
240   RB=RB/(NSCH+1)
      GO TO 230

   ITERN FOR INDIVIDUAL ROOTS
300   IF (IFPR.EQ.1)
    * WRITE (6,1040)
      NITE(JR)=-1
      IF (IFPR.EQ.1)
    * WRITE (6,1050) JR,NITE(JR),RA,DETA,FA,ETA,ISC
      NITE(JR)=0
      IF (IFPR.EQ.1)
    * WRITE (6,1050) JR,NITE(JR),RB,DETB,FB,ETA,ISC

   WE STOP WHEN WE HAVE THE REQUIRED NO OF ROOTS SMALLER THAN RC AND
   NOV=0
310   IF (NSCH.GE.NROOT) GO TO 900
      IF (RB.GT.CHFQ) GO TO 900

03        DIF=FB-FA
      IF (DIF.NE.0.0) GO TO 320
     WRITE (6,1060)
      GO TO 900
320   DEL=FB*(RB-RA)/DIF
      RC=RB-ETA*DEL
      TOL=RCBTOL*RC
     IF (DABS(RC-RB) .GT. TOL) GO TO 330
      IF (IFPR.EQ.1)
    * WRITE (6,1070)
      ROOT(JR)=RB
      GO TO 400

330   CALL BANDET (A,B,V,MAXA,N,NWA,RC,NSCH,DETC,ISC,1)
      FC=DETC
      NITE(JR)=NITE(JR)+1
      IF (JR.EQ.1) GO TO 340
      JJ=JR-1
      DO 350 K=1,JJ
350   FC=FC/(RC-ROOT(K))
340   IF (IFPR.EQ.1)
    * WRITE (6,1050) JR,NITE(JR),RC,DETC,FC,ETA,ISC

   IF WE HAVE MORE SIGNCHANGES THAN EIGENVALUES SMALLER THAN RC WE
   START INV. ITERATION
      NES=0
      IF (JR.EQ.1) GO TO 360
```

```
      DO 360 I=1,JJ
360   IF (ROOT(I).LT.RC) NES=NES+1
380   NOV=NSCH-NES
      IF (NOV.EQ.0) GO TO 370
      IF (IFPR.EQ.1)
    * WRITE (6,1080) NOV
      ROOT(JR)=RC
      IF (NOV.GT.1) NSR=1

      GO TO 400
370   RR=RA
      FR=FA
      DETR=DETA
      RA=RB
      FA=FB
      DETA=DETB
      RB=RC
      FB=FC
      DETB=DETC

   WE RESET ETA IF NECESSARY
      TOL=RB*ACTOL
     IF (DABS(RA-RB) .LT. TOL) ETA=ETA*2.0D0
      IF (NITE(JR).LE.NITEM) GO TO 310
     WRITE (6,1015) JR,NITE(JR)
      GO TO 900

   CHECK FOR STORAGE
400   IF (JR.LE.NC) GO TO 405
      WRITE (6,1090)
      GO TO 900

405   NOR=JR-1
      IF (NOR.GT.NVM) NOR=NVM
      IF (IFPR.EQ.1)
    * WRITE (6,1100) NOR
      IF (JR.EQ.1) GO TO 410
      DO 420 I=1,N
420   V(I)=1.0
      KK=2
     IF (JR.EQ.NC) GO TO 410
     I = NITE(JR+1)
     V(I) = -1.
410   DO 430 I=1,N
430   W(I)=B(I)*V(I)
      IS=0
      GO TO 510

   INVERSE ITERN
440   NITE(JR)=NITE(JR)+1
      DO 450 I=1,N
450   V(I)=W(I)
      CALL BANDFI (A,B,V,MAXA,N,NWA,RC,NSCH,DETC,ISC,KK)
      IF (IS.EQ.1) GO TO 460
      KK=2
      RQT=0.0
      DO 470 I=1,N
470   RQT=RQT+W(I)*V(I)
      DO 475 I=1,N
475   W(I)=B(I)*V(I)
```

```
      RQB=0.0
      DO 480 I=1,N
480   RQB=RQB+W(I)*V(I)
      RQ=RQT/RQB
      RT=ROOT(JR)+RQ
      IF (IFPR.EQ.1)
     * WRITE (6,1110) JR,NITE(JR),RT,RQ
      TOL=RT*RQTOL
     IF (DABS(RT-RTA) .GT. TOL)  GO TO 510
      IS=1
      GO TO 440

510   RTA=RT
     BS=DSQRT(RQB)
      DO 490 I=1,N
490   W(I)=W(I)/BS
      IF (NOR.EQ.0) GO TO 550
      DO 520 K=1,NOR
      AL=0.0
      DO 530 I=1,N
530   AL=AL+VV(I,K)*W(I)
      DO 540 I=1,N
540   W(I)=W(I)-AL*WW(I,K)
520   CONTINUE

550   IF (NITE(JR).LE.NITEM) GO TO 440
     WRITE (6,1015) JR,NITE(JR)
      GO TO 900

460   RQT=0.0
      ERRT=RQB
      DO 570 I=1,N
570   RQT=RQT+V(I)*W(I)
      DO 560 I=1,N
560   W(I)=B(I)*V(I)
      RQB=0.0
      DO 580 I=1,N
580   RQB=RQB+V(I)*W(I)

   OBTAIN A RATHER LARGE ERROR BOUND
      RQ=RQT/RQB
      ROOT(JR)=ROOT(JR)+RQ
     ERR=DSQRT(ERRT/RQB)
      ERRVL(JR)=ROOT(JR)-ERR
      ERRVR(JR)=ROOT(JR)+ERR

     BS=DSQRT(RQB)
      DO 590 I=1,N
      W(I)=W(I)/BS
590   V(I)=V(I)/BS
      JJ=JR
      IF (JJ.LE.NVM) GO TO 610
      WRITE (NT) (VV(J,1),J=1,N)
      DO 600 K=1,N
      DO 600 L=2,NVM
      WW(K,L-1)=WW(K,L)
600   VV(K,L-1)=VV(K,L)
      JJ=NVM
610   DO 620 K=1,N
      WW(K,JJ)=W(K)
```

```
620   VV(K,JJ)=V(K)


   DECIDE STRATEGY FOR ITERN TOWARDS NEXT ROOT
      TOL=RTOL*ROOT(JR)
      IF (NOV.GT.0) GO TO 700
     IF (DABS(ROOT(JR)-RB) .GT. TOL) GO TO 710
      IF (RA.GT.0.0) GO TO 720
      RA=RB/2.
      CALL BANDET (A,B,V,MAXA,N,NWA,RA,NSCH,DETA,ISC,1)
      FA=DETA
720   RB=RA
      FB=FA
      DETB=DETA
      RA=RR
      FA=FR
      DETA=DETR
      GO TO 710

700   IF (ROOT(JR).GT.RC) NSK=1
      IF (NSK.EQ.1) GO TO 730
     IF (DABS(RC-ROOT(JR)) .LT. TOL) GO TO 740
     IF (DABS(ROOT(JR)-RB) .LT. TOL) GO TO 750
      RA=RB
      FA=FB
      DETA=DETB
750   RB=RC
      FB=FC
      DETB=DETC
      GO TO 710
 740 IF (DABS(ROOT(JR)-RB) .GT. TOL) GO TO 710
      IF (RA.GT.0.0) GO TO 760
      RA=RB/2.
      CALL BANDET (A,B,V,MAXA,N,NWA,RA,NSCH,DETA,ISC,1)
      FA=DETA
760   RB=RA
      FB=FA
      DETB=DETA
      RA=RR
      FA=FR
      DETA=DETR
710   FA=FA/(RA-ROOT(JR))
      FB=FB/(RB-ROOT(JR))
      JR=JR+1
     ETA=2.0
      GOTO 300

730   IF (RA.GT.0.0) GO TO 780
      RA=RB/2.
      CALL BANDET (A,B,V,MAXA,N,NWA,RA,NSCH,DETA,ISC,1)
      FA=DETA
780  IF (DABS(ROOT(JR)-RB).GT.TOL) GO TO 770
      RB=RA
      FB=FA
      DETB=DETA
      RA=RR
      FA=FR
      DETA=DETR
770   FA=FA/(RA-ROOT(JR))
      FB=FB/(RB-ROOT(JR))
```

```
      FR=FR/(FR-ROOT(JR))
         IF (ROOT(JR).LE.RC) NOV=NOV-1
      JR=JR+1
      NITE(JR)=0
      ROOT(JR)=RC
      IF (NOV.GT.0) GO TO 400
      NSR=0
     ETA=2.0
      GOTO 300

900   NROOT=JR-1
     IF(NROOT.GT.0) GO TO 902
     WRITE (6,1180)
     STOP
 902 CONTINUE
      IF (IFPR.EQ.0) GO TO 905
      WRITE (6,1140)
      WRITE (6,1006) (NITE(J),J=1,NROOT)
      WRITE (6,1150)
      WRITE (6,1008) (TIM(J),J=1,NROOT)
      WRITE (6,1160)
      WRITE (6,1004) (ERRVL(J),J=1,NROOT)
      WRITE (6,1004) (ERRVR(J),J=1,NROOT)

   READ EIGENVECTORS INTO CORE
905   IF (NROOT.LE.NVM) GO TO 906
      NDIF=NROOT - NVM
      REWIND NT
      DO 904 L=1,NDIF
      READ (NT) (A(I,L),I=1,N)
904   CONTINUE
      GO TO 908
906   NDIF=0
908   NROOT=NROOT - NDIF
      DO 912 L=1,NROOT
      DO 912 I=1,N
912   A(I,L+NDIF)=VV(I,L)

   ARRANGE EIGENVALUES AND VECTORS IN ASCENDING ORDER
      IF (JR.EQ.2) GO TO 950
      JR=JR-2
910   IS=0
      DO 920 I=1,JR
      IF (ROOT(I+1).GE.ROOT(I)) GO TO 920
      IS=IS+1
      RT=ROOT(I+1)
      ROOT(I+1)=ROOT(I)
      ROOT(I)=RT
      DO 930 K=1,N
      RT=A(K,I+1)
      A(K,I+1)=A(K,I)
930   A(K,I)=RT
920   CONTINUE
      IF (IS.GT.0) GO TO 910

950   WRITE (6,1170)
      NROOT=NSCH
      WRITE (6,1004) (ROOT(J),J=1,NROOT)

     CALCULATE PHYSCIAL ERROR NORMS
```

```
      REWIND NT
      DO 955 L=1,NROOT
  955 WRITE (NT) (A(K,L),K=1,N)
      REWIND NSTIF
      READ (NSTIF) (A(I,1),I=1,NWA)
      REWIND NT
      DO 960 L=1,NROOT
      RT = ROOT(L)
      READ (NT) (V(I),I=1,N)
      CALL MULT (W,A,V,N,MA)
      VNORM=0.
      DO 958 I=1,N
  958 VNORM = VNORM + W(I)*W(I)
      DO 966 I=1,N
  966 W(I) = W(I) -RT*B(I)*V(I)
      WNORM = 0.0
      DO 968 I=1,N
  968 WNORM = WNORM + W(I)*W(I)
      VNORM =DSQRT(VNORM)
      WNORM =DSQRT(WNORM)
      ERRVL(L) = WNORM/VNORM
  960 CONTINUE
      REWIND NT
      DO 969 L=1,NROOT
  969 READ (NT) (A(K,L),K=1,N)

      WRITE (6,1196)
      WRITE ( 6,1004) (ERRVL(J),J=1,NROOT)

       REWIND NT
       DO 970 I=1,NROOT
  970 ROOT(I)=DSQRT(ROOT(I))
       WRITE (NT) (ROOT(I),I=1,NROOT)
       NWA=N*NROOT
      WRITE (NT) (A(I,1),I=1,NWA)
      PI2=8.0D0*DATAN(1.0D0)
       DO 980 I=1,NROOT
  980  AT(I)=PI2/ROOT(I)

       RETURN

 1000 FORMAT (44H ***ERROR    NEG OR ZERO DIAGONAL ELEMENT A(,I4,4H) = ,
    1          E11.4,21HBEFORE DECOMPOSITION          )
 1004  FORMAT (1H0,6E20.12)
 1006  FORMAT (1H0,6I20)
 1008  FORMAT (1H0,6F20.2)
 1009 FORMAT (43H0***ERROR   SOLUTION TERMINATED IN *SECNTD*, /
    1          12X,25HRIGID BODY MODE(S) FOUND., / 1X)
 1010 FORMAT (51H1INVERSE ITERATION GIVES FOLLOWING APPROXIMATION TO,
    1          18H LOWEST EIGENVALUE, 1X)
 1015 FORMAT (41H0***ERROR   PRE-MATURE EXIT FROM *SECNTD*, / 12X,
    1          37HITERATION ABANDONED FOR ROOT NUMBER =, I4 / 12X,
    2          37HNUMBER OF ITERATIONS PERFORMED      =, I4 / 1X)
 1020  FORMAT (5H0RC = E20.12,7H NSCH = I4)
 1030 FORMAT (30H0***ERROR   SOLUTION STOP IN *SECANTD*, / 12X, 1H(,
    1          I3,48H) FACTORIZATIONS PERFORMED IN AN ATTEMPT TO FIND,
    2          32H LOWER BOUND ON FIRST EIGENVALUE, / 12X,
    3          16HCHECK THE MODEL., / 1X)
 1040  FORMAT (1H1,4X,4HROOT,4X,4HNITE,18X,2HRC,15X,12HDET (A-RC*B),15X,
```

```
     /2HFC,13X,3HETA,4X,3HISC)
1050  FORMAT (1H0,4X,I4,4X,I4,8X,3E22.14,F7.2,I6)
1060  FORMAT (42H0THE DEFLATED POLYNOMIAL HAS NO MORE ROOTS   )
1070  FORMAT (29H0(RC-RB) IS SMALLER THAN TOL )
1080  FORMAT (16H0WE JUMPED OVER I4,16H UNKNOWN ROOT(S)   )
1090 FORMAT (41H0***ERROR   PRE-MATURE EXIT FROM *SECNTD*,
    1           34H CAUSED BY EITHER OF THE FOLLOWING, / 12X,
    2           22H(1) BAD MODEL DATA, OR, / 12X,
    3           52H(2) ROOT CLUSTER (I.E., NEAR EQUAL OR REPEATED EIGEN,
    4           36HVALUES) ENCOUNTERED AT CURRENT SHIFT, / 16X,
    5           25HCAUSING STORAGE OVER-FLOW, 1X)
1100  FORMAT (1H0,34X,4HROOT,18X,2HRG,18X,4HNOR=,I2)
1110  FORMAT (1H0,4X,I4,4X,I4,8X,2E22.14)
1140  FORMAT (42H0NO OF ITERATIONS FOR EACH EIGENVALUE ARE   /)
1150  FORMAT (30H0TIME USED FOR EACH EIGENVALUE /)
1160  FORMAT (43H0FOLLOWING ARE ERROR BOUNDS ON EIGENVALUES )
1170  FORMAT (///  40H WE SOLVED FOR THE FOLLOWING EIGENVALUES     )
1180 FORMAT (37H0***ERROR    SOLUTION STOP IN *SECNTD*, / 12X,
    1           23HNO EIGENVALUES COMPUTED, / 1X)
1190 FORMAT (/// 40H THE FOLLOWING ARE PHYSICAL ERROR BOUNDS,
    1                  20H ON THE EIGENPAIRS       )

      END
```

```
      DATA SET SHAKE       AT LEVEL 059 AS OF 83/02/11
  SUBROUTINE SHAKE(NPK,NPJ,KPT,KK,ID,IH,AM,EK,AN,ANI,N43,AI,NPT,

 *GB,SO)


  IMPLICIT REAL*8  (A-H,O-Z)
  COMMON D(1)



  COMMON/CONTR/DD,HED(8),KADN,NUMNP,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA
 *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH ,NEQS,MBANDS
  COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC
  COMMON/TARA/LPC,N,M1,M2,NGT,NGF,NPRNT,M,MM


  COMMON/DIMEL/M43,IPT,IPT1,MK,MPRNT
  COMMON/QUAD/CCCC(1500),N1,N2,N3,N4,N5,MTOT,NNN,NK15


  DIMENSION ID(NUMNP,1),IH(NUMNP,1),AM(1),EK(1),AN(NPJ,1),ANI(NYP,1)

 *,AI(NEQS,1)
 *,GB(1),SO(1)





  N44=N43+NPJ*NPJ
  N45=N44+NPJ*NEQS
  N045=N45+NPJ*NEQS
  N046=N045+NPJ*NPJ
  IF(IPT.EQ.1.AND.IPT1.EQ.1.OR.IPT.EQ.1.AND.IPT1.EQ.2)N046=N045+NPJ
  NGT=NPT
  NGF=KPT
  NPRNT=MPRNT




  IF(KPT.EQ.0)  GO  TO  3

1 CONTINUE



2 CONTINUE
  ************     CALL  SHAKEMN AT  THIS  STAGE  *************


3 CONTINUE



  N047=N046+NEQS*NEQS
  N050= N047+NEQS
  N051=N050+NEQS
  N052=N051+NPJ
```

```
      N053=N052+NPJ                                                00061*21
      N054=N053+NPJ+NPK
      IF(IPT.EQ.1.AND.IPT1.EQ.1.OR.IPT.EQ.1.AND.IPT1.EQ.2)N054=N053+NPJ


      IF(KPT.EQ.1) GO TO 4

      *****   ALLOCATION  FOR  ZX3LP = MAXIMIZATION  *********

      MN=NPK+1+2                                                   00062*28
                                                                   00063*33
      IF(NPT.EQ.1)  MN=(NPK+NEQS)+1+2                              00064*33
      IF(NPT.EQ.2) MN=(NPK+NEQS)+1+2                               00065*43
                                                                   00066*28
                                                                   00067*28
      N055=N054+MN                                                 00069*29
                                                                   00070*27
                                                                   00071*27
                                                                   00072*27
      M1=NPK                                                       00073*27
      M2=0                                                         00074*27
      IF(NPT.EQ.1)  M2=NEQS                                        00075*33
      IF(NPT.EQ.2) M2=NEQS                                         00076*43
      LPC=(M1+M2+2)+1                                              00077*28
                                                                   00078*28
      MM=M1+M2+1
                                                                   00079*28
      N=NPK+1                                                      00080*27
                                                                   00081*33
                                                                   00082*33
       IF(NPT.EQ.1.OR.NPT.EQ.2) N=(1+2*NPJ)                        00083*45
                                                                   00084*43
      N056=N055+LPC+N                                              00085*27
      N52=N056+N+2                                                 00086*33
      N53=N52+MAX0(N,M1+M2+1)                                      00087*48
      N54=N53+LPC                                                  00088*27
      N55=N54+LPC+LPC+3*M1+2*M2+4                                  00089*27
      N56=N55+2*M2+3*M1+4                                          00090*27
                                                                   00091*48
      *****  DUMMY  DIMENSION FOR COPI,ICOLMS,IDES,&ROW ******

      NM56=N56+(NPK*NPK)
      N057=NM56+1
      N058=N057+1
      N059=N058+1


      GO TO 6

      *******   ALLOCTION  FOR ZXMIN= MINIMIZATION  ********

    4 CONTINUE

      M1=0
      M2=NPK+1
      IF(NPT.EQ.1.OR.NPT.EQ.2) M2=NPJ+1
      LPC=M2+2
```

```
      M=M2+1
      MM=M+1
      N=NPK+2*NEQS

      N055=N054+(MM)
      N056=N055+(MM*N)
      N52=N056+N+1
      N53=N52+MM
      N54=N53+MM
      N55=N54+N
      N56=N55+1
      NM56=N56+(MM*MM)
      N057=NM56+LPC
      N058=N057+LPC
      N059=N058+LPC

      ******    FINISH   ******


    6 CONTINUE


      CHECK   FOR   BOUND    ***********
      N57=N059+MAX0(N,M1+M2+1)

      IF(KK.EQ.0) N57=N059+1
      N58=N57+LPC

      IF(KK.EQ.0) N58=N57+1
      N59=N58+NPK

      IF(KK.EQ.0)  N59=N58+1
      N60=N59+NPK*N
      IF(KK.EQ.0)  N60=N59+NPK*1
      WRITE(6,108)N43,N44,N45,N045

      WRITE(6,108)N046,N047,N050,N051,N052,N053,N054,N055
      WRITE(6,108)N056,N52,N53,N54,N55,N56,NM56,N057,N058
      WRITE(6,108)N059,N57,N58,N59,N60

      IF(N60.GT.MTOT)  GO  TO  900



      CALL ACOMP(D(N43),D(N44),D(N45),NUMEL,NPJ,NPK,AN,NPT)






      IF(NPT.EQ.0)  GO  TO  800

      IF(NPT.EQ.2.OR.NPT.EQ.1)  GO  TO  110


      CALL STIFF(A1,NEQS,MBANDS,D(N046))


  110 CONTINUE
```

```
      LO=NPJ                                                     00130*11
      MO=NPK                                                     00131*11
      PRINT 109                                                  00132*17
      PRINT 109                                                  00133*17
  108 FORMAT(' ',10I12)                                          00134*49
```

```
9     FORMAT('  ****  RETURN FROM STIFF    *******')                           00135*17
                                                                                00136*11
                                                                                00137*11
                                                                                00138*29
      *************     N43        =         S                                  00139*29
      *************     N44        =         C                                  00140*29
      *************     N45        =         CT                                 00141*29
      *************     N045       =         ANIT                               00142*29
      *************     N046       =         AS                                 00143*29
      *************     N047       =         SI                                 00144*29
      *************     N050       =         SM                                 00145*29
      *************     N051       =         SN                                 00146*29
      *************     N051       =         SP                                 00147*29
      *************     N053       =         SD                                 00148*29
      *************     N054       =         SQ                                 00149*29
      *************     N055       =         A                                  00150*29
      *************     N056                 CE                                 00151*29
      *************     N52        =         PSOL                               00152*29
      *************     N53        =         DSOL                               00153*29
      *************     N54        =         RW                                 00154*29
      *************     N55        =         IW                                 00155*29
                                                                                00156*29
                                                                                00157*29
                                                                                00158*11
      CALL  INFMTX(D(N43),D(N44),D(N45),AN,D(N045),D(N046),D(N047),D(N0500159*24
                                                                                00160*29
                                                                                00161*29
     *0),D(N051),D(N052),D(N053),D(N054),NPO,NPR,NEGS,D(N055),D(N056),          00162*27
                                                                                00163*29
                                                                                00164*29
     *D(N52),D(N53),D(N54),D(N55),AM,EK,KR,IH,D(N56),D(NM56),D(N057)            00165*41
                                                                                00166*41
     *,D(N058),GB,SD,D(N059),D(N57),D(N58),D(N59))                              00167*52
                                                                                00168*11
                                                                                00169*11
                                                                                00170*11
      GO  TO  900                                                               00171*21
800   CONTINUE                                                                  00172*22
                                                                                00173
                                                                                00174
      ********   ALLOCATION  FOR   SHAKE  *****************                     00175
                                                                                00176**6
                                                                                00177**6
                                                                                00178*27
                                                                                00179*11
                                                                                00180*11
                                                                                00181*11
                                                                                00182
900   CONTINUE                                                                  00183
      RETURN                                                                    00184
      END                                                                       00185
```

```
       DATA SET SDD          AT LEVEL 049 AS OF 83/02/24
   SUBROUTINE SDD(AI,XMS,NEQS,MBANDS,NUMNP,NUMEL,KKS,ID,IH,LOOP,NDUM,00001*11
                                                                    00002*31
  *DBT1,DBT2,NUMMAT,GMX,DPL,GOSE,ES,EN,PO,AS,X,ROOT,VROOT,ROOT1)    00003*25
                                                                    00004*31
   IMPLICIT REAL*8(A-H,O-Z)                                         00005*19
   COMMON D(1)                                                      00006*20
                                                                    00007*31
                                                                    00008**6
                                                                    00009**6
                                                                    00010**6
   COMMON/EMB/S(16,16),GM(16),GH(16,16),SS(16,16),XP(16),XN(        00011*42
  *16),EM(16)                                                       00012*42
   COMMON/CONTL/ID,HED(8),NADN,NUMKP,NUMSL,NEQ,MBAND,NCYCL,NDYN,IPLNA00013**5
  *X,ISTRER,NPRINT,ISTART,NEQH,MBANDH                               00014*16
  *,NEQG,MBANDG                                                     00015*45
   COMMON/SOL/ANORM,NBLOCK,NEQB,LL,NF,IDUM,NEIG,NAD,NVV,NFO         00016**4
   COMMON/ASOL/W1                                                   00017**4
   COMMON/ELPAR/NPAR(14),NELTYP                                     00018*11
   COMMON/EXTRA/MODEX,NT6,NT6SV,NT10                                00019*11
       COMMON/TAPES/NGO(6),NUMBER                                   00020*49
   COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),   00021*28
  *ST(20,12),DG(4,4),C(12,12),E(12,12),TT(5,12),P1(4,8)             00022*30
      COMMON/BUBBT/KGS,NUNAT,NC,LOOP,NDRN,NINT,IDWA,NPT             00023*44
                                                                    00024*11
                                                                    00025*11
                                                                    00026*11
                                                                    00027*11
    DIMENSION EK(8),LH(8),SK(8,8),       GMS(8),NP(4)               00028*14
   DIMENSION XMS(1),AI(NEQS,1),DBETA1(1),DBETA2(1),ID(NUMNP,1),IH(NUM00029**2
  *NP,1),DPL(1),GMX(1),GOSE(1),DBT1(1),DBT2(1),ES(1),EN(1),PO(1)    00030*11
                                                                    00031**6

    DIMENSION AS(NEQS,1),X(NEQS,1),ROOT(NEQS),VROOT(NEQS,1),
  *ROOT1(NEQS)
                                                                    00032**6
                                                                    00033**6
   REWIND 22                                                        00034*11
   REWIND  61                                                       00035*11
   REWIND  51                                                       00036
                                                                    00037*32
                                                                    00038*32
   DO  300  I=1,NEQS                                                00039
   XMS(I)=0.0                                                       00040
   DO  300 J=1,MBANDS                                               00041**2
300 AI(I,J)=0.0                                                     00042
                                                                    00043*14
                                                                    00044*14
   KK=1                                                             00045*41
     WRITE(6,451)
451   FORMAT(1H1/5X,'  PRINT IN SDD ')

                                                                    00046*14
   DO  100  N=1,NUMEL                                               00047
   READ(51)(EK(I),I=1,8),(LH(I),I=1,8),((SK(I,J),J=1,8),I=1,8)      00048*18
  *,((P1(I,J),J=1,8),I=1,4),(GMS(I),I=1,8),VOL,DENS,DENF,F,NP       00049*34
                                                                    00050*22
                                                                    00051*22
                                                                    00052*32
   *****  CHECK  HERE  FOR  SHAKEDOWN  UNDER  UNDRAINED   CONDITION  00053*31
```

```
     *****   SHOUD  WE  USE  SK  OR  SF  *****  CHANGE  IN  CONSV  FOR
     ****  ED  CONDITION  BY  ADJUSTING  THE  LAME ' S  CONSTANTS  ****


     READ(51) ((SF(I,J),J=1,8),I=1,8),((E(I,J),J=1,8),I=1,8),((C(I,J),J
    *=1,8),I=1,8),IMAT,KS,KF,ALF,((H(I,J),J=1,8),I=1,8)
    *,(XP(I),I=1,16),(XN(I),I=1,16)


     IF(KKS.NE.2)  GO  TO  6000

     KG=IMAT

          IF(LOOP.EQ.1)  GO TO 450

          READ(22) ENEW,GNEW,DAMP,POIS

          AK=ENEW/EO(IMAT)


 450 CONTINUE

              IF(LOOP.EQ.1) AK=1.0

        WRITE(6,452) N,AK
 452    FORMAT(5X,I5,E10.3)



     DO  400  I=1,8
     DO  400  J=1,8
 400 SK(I,J)=SK(I,J)*AK
     DO  600 I=1,8
     DO  600  J=1,8
 600 SF(I,J)=SK(I,J)+ALF*E(I,J)
     WRITE(61) VOL,DENS,DENF,F,SF,E,C,IMAT,KS,KF,ALF,H,NP
    *,XP,XN
6000 CONTINUE



 550 CONTINUE

     DO  360  I=1,8
     LP=LK(I)
     IF(LP) 360,360,290
 290 XMS(LP)=XMS(LP)+GMS(I)
 201 DO  350 J=1,8
     KL=LK(J)
     IF(KL)  350,350,250
 250 IF(KL-LP) 350,390,390
 390 KQ=KL-LP+1
     IF((KKS.EQ.4.AND.NDRN.EQ.0).OR.(KKS.EQ.2.AND.NDRN.EQ.0))
    *  GO  TO  365
     AI(LP,KQ)=AI(LP,KQ)+SK(I,J)
     GO  TO  350
```

```
                                                                          00112*32
                                                                          00113*32
    ***   S H A K E D O W N UNDER U N D R A I N E D   CONDITION      *****00114*32
365 AI(LP,KQ)=AI(LP,KQ)+SF(I,J)                                           00115*32
                                                                          00116*32
350 CONTINUE                                                              00117
360 CONTINUE                                                              00118
                                                                          00119*25
                                                                          00120*41
    KK=KG+1                                                               00121*41
                                                                          00122*25
100 CONTINUE                                                              00123
    IF(KKS.EQ.4)  GO  TO  9000                                            00128
                                                                          00129*11
                                                                          00130*11
                                                                          00131*11
    REWIND  9                                                             00132
    REWIND  4                                                             00133
    NPAR(1)=1                                                             00134*11
    NELTYP=1                                                              00135*11
    NEQB=NEQS                                                             00136
    NBLOCK=1                                                              00137
    MM=1                                                                  00138
    NDEG=0                                                                00139
    NVV=0                                                                 00140
    ANORM=0.0                                                             00141
    DO  710  L=1,NEQS                                                     00142
    IF(AI(L,1).NE.0)  NDEG=NDEG+1                                         00143
    IF(XMS(L).NE.0)  NVV=NVV+1                                            00144
    IF(AI(L,1).EQ.0) AI(L,1)=1.0E+20                                      00145
710 CONTINUE                                                              00146
     WRITE(4) ((AI(I,J),I=1,NEQB),J=1,MBANDS)                             00147**3
     WRITE(9)  (XMS(I),I=1,NEQS)                                          00148
    IF(NDEG.GT.0)  GO  TO  730                                            00149
    WRITE(6,1010)                                                         00150
    STOP                                                                  00151
730 ANORM=(ANORM/NDEG)*1.0E-20                                            00152*44
1010 FORMAT(51H0STRUCTURE WITH NO DEGREES OF FREEDOM  CHECK DATA    )     00153
                                                                          00154
                                                                          00155
                                                                          00156
     CALL SOLEIG(LOOP,IH,NDDM,NN15,NN17)                                   00157*25
      WRITE(6,105)                                                          00124*39
105   FORMAT(1H1/'   ***  PRINT  OF  ASSEMBLED  UNDRAINED  MATRIX ** ') 00125*39
      WRITE(6,106)  ((AI(I,J),J=1,MBANDS),I=1,NEQS)                         00126*39
      WRITE(6,106) (XMS(I),I=1,NEQS)
106   FORMAT(5X,6E12.5)                                                    00127*40
                                                                          00158
     CALL STIFF1(AI,NEQS,MBANDS,AS)
     DO 200 I=1,NEQS
     DO 200 J=1,NEQS
     IF(I.EQ.J) X(I,J)=XMS(I)
     IF(I.NE.J) X(I,J)=0.0
200  CONTINUE
     WRITE(6,101)                                                          00124*39
101  FORMAT(1H1/'   ***  PRINT  OF  ASSEMBLED  UNDRAINED  MATRIX ** ') 00125*39
      WRITE(6,102)  ((AS(I,J),J=1,NEQS),I=1,NEQS)                           00126*39
      WRITE(6,209) ((X(I,J),J=1,NEQS),I=1,NEQS)
102  FORMAT(5X,8E12.5)                                                     00127*40
     CALL WROOT(NEQS,AS,X,ROOT,VROOT)
```

```
       WRITE(6,209)((AS(I,J),J=1,NEQS),I=1,NEQS)
       WRITE(6,209) (ROOT(N),N=1,NEQS)
 209  FORMAT(5X,8E10.3)
      DO 210 N=1,NEQS
      ROOT1(N)=ROOT(N)
 210  CONTINUE
      DO 211 N=1,NEQS
      NK=NEQS+1
 211  ROOT(N)=ROOT1(NK-N)
                                                                00159
       DO 212 N=1,NEQS
 212   ROOT(N)=(DSQRT(ROOT(N)))
       WRITE(6,209) (ROOT(N),N=1,NEQS)

       IF(NDIM.EQ.3) GO TO 9000


       W1=ROOT(1)


     REWIND  61                                                 00160*11
                                                                00161*47
                                                                00162*47
              IF(LOOP.NE.1) GO TO 850                           00163*47
                                                                00164*47
     DO  800  M=1,NUMMAT                                        00165
     DBT1(M)=DPL(M)*W1                                          00166*11
     DBT2(M)=DPL(M)/W1                                          00167*11
 800 CONTINUE                                                   00168
                                                                00169*47
              GO TO 855                                         00170*47
                                                                00171*47
                                                                00172*47
 850          CONTINUE                                          00173*47
                                                                00174*47
                                                                00175*47
              REWIND 22                                         00176*47
                                                                00177*47
                                                                00178*11
                                                                00179*11
                                                                00180*11
              DO 860 M=1,NUMEL                                  00181*47
              READ(22) ENEW,GNEW,DAMP,POIS                      00182*47
              DBT1(M)=DAMP*W1                                   00183*47
              DBT2(M)=DAMP/W1                                   00184*47
 860          CONTINUE                                          00185*47
                                                                00186*47
 855          CONTINUE                                          00187*47
                                                                00188*48
                                                                00189*48
          REWIND 13                                             00190*48
                                                                00191*48
                                                                00192*48
     DO  500  M=1,NUMEL                                         00193*21
                                                                00194*25
                                                                00195*25
     NEL=M                                                      00196*25
     READ(61)  VOL,DENS,DENF,F,SF,E,C,IMAT,KS,KF,ALF,H,NP       00197*14
    *,XP,XN                                                     00198*43
```

```
                                                                        00199*14
                                                                        00200*14
                                                                        00201*14
      L=0                                                               00202*14
      DO  620 I=1,4                                                     00203*14
      IIM=I                                                             00204*14
      N=NP(IIM)                                                         00205*14
      DO  620 J=1,4                                                     00206*14
      L=L+1                                                             00207*14
      LM(L)=ID(N,J)                                                     00208*14
  620 CONTINUE                                                          00209*14
                                                                        00210*14
                                                                        00211*14
      CALL MSFDMX(VOL,DENS,DFNF,KS,KF,DBT1,DBT2,F,NUMNP,NEL,ID,IH,IMAT) 00212*26
                                                                        00213*34
       *******   SHOULD  WE  CALL   LOAD  ROUTINE     *********         00214*34
                                                                        00215*34
                                                                        00216*42
*******    REFORMULATE  FOR  SELF  WT  *********************************00217*42
                                                                        00218*42
                                                                        00219*42
      WRITE(13)  LM,S,GM,GH,SS,XP,XN                                    00220*42
                                                                        00221*42
                                                                        00222*42
  500 CONTINUE                                                          00223
 9000 CONTINUE                                                          00224
                                                                        00225*31
                                                                        00226*31
      RETURN                                                            00227
      END                                                               00228
```

```
          DATA SET SOLDN        AT LEVEL 010 AS OF 83/02/16
      SUBROUTINE    SOLDN (AT,BT,X0,A5,NEQ,X1,X2,A4,NWA,AB,MAXA,NEQB,

     *NBLOCK,M1,MBAND,NC ,XM)

      IMPLICIT  REAL *8(A-H,O-Z)

         COMMON/GORA/IFLAG

      COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A08
      COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,NBAND,NCYCL,NDYN,IPLNA
     *X,INTRPR,NPRINT,ISTART,NEQR,MBANDR
     *,NEQS,MBANDS
           COMMON/BOBBY/KKS,NUMMAT,NG,LOOP,NDRN,NINT,IDWA,NPT
           COMMON/INTEGE/TETA,BETA,GAMA,RIOL,NGAMA,NPUNCH,NEQUIB,ITEMX




      DIMENSION  AT(1),X0(1),X1(1),X2(1),BT(1),A5(NEQ,1),A4(NWA),AB(1) ,
     *XM(1),MAXA(1)
      *********************END********************************************


      *********DIVERT  FROM  THIS  STEP   CODED  BY  HALDAR ************

      SOLVE  FOR  DISPLACEMENTS ,VELOCITIESAND ACCELERATIONS



      DO  4500 I=1,NEQ




     *****  FOR  INELASTIC  ANALYSIS  ************************************

           GO TO (510,510,500,510),KKS

 510       CONTINUE

      AT(I)=A0*X0(I)+A02*X1(I)+2.*X2(I)
      BT(I)=A01*X0(I)+2.*X1(I)+0.5*DELT1*X2(I)
      GO  TO  4500


 500  AT(I)=A02*X1(I)+2.*X2(I)
      BT(I)=2.*X1(I)+A03*X2(I)
4500  CONTINUE


      DO  450  I=1,NEQ




             GO TO (610,610,600,610),KKS

 610         CONTINUE

```

```
      AU(I)=AU(I)+XM(I)*(A0*X0(I)+A02*X1(I)+2.*X2(I))+A5(I,I)*(A01*X0(I)00060
     *+2.*X1(I)+0.5*DELT1*X2(I))                                        00061
                                                                        00062**4
      GO  TO  450                                                       00063**5
                                                                        00064**4
      *******  FOR  INELASTIC  ANALYSIS  ONLY  *************************00065**4
                                                                        00066**4
  600 AU(I)=AU(I)+XM(I)*AT(I)+A5(I,I)*BT(I)                             00067**4
                                                                        00068**4
                                                                        00069**4
  450 CONTINUE                                                          00070
                                                                        00071**4
                                                                        00072**4
      DO  451  N=1,NEQ                                                  00073
      LQ=N-1                                                            00074
      MR=MIN0(MBAND,NEQ-N+1)                                            00075
      IF(MR.EQ.1)   GO  TO  3600                                        00076
      DO  3200  J=2,MR                                                  00077
      M=LQ+J                                                            00078
                                                                        00079
                                                                        00080
                                                                        00081
                                                                        00082**4
                                                                        00083**4
                                                                        00084**4
            GO TO (710,710,700,710),KKS                                 00085**8
                                                                        00086**8
  710       CONTINUE                                                    00087**8
                                                                        00088
      AU(N)=AU(N)+A5(N,J)*(A01*X0(M)+2*X1(M)+0.5*DELT1*X2(M))           00089
      AU(M)=AU(M)+A5(N,J)*(A01*X0(N)+2.*X1(N)+0.5*DELT1*X2(N))          00090
                                                                        00091**5
                                                                        00092**5
      GO  TO  3200                                                      00093**5
  700 AU(N)=AU(N)+A5(N,J)*BT(M)                                         00094**4
      AU(M)=AU(M)+A5(N,J)*BT(N)                                         00095**4
                                                                        00096**4
                                                                        00097**4
 3200 CONTINUE                                                          00098
  451 CONTINUE                                                          00099
 3600 CONTINUE                                                          00100
                                                                        00101**5
                 WRITE(6,7000) NG                                        00102*10
 7000            FORMAT(1H1/'           PRINT   IN SUBDN        '/      00103*10
     *                   5X,'           TIME STEP(NG)      = ',I5/)     00104*10
                                                                        00105*10
                 WRITE(6,7020)                                           00106*10
 7020            FORMAT(5X,'             MODIFIED LOAD VECTOR AU(I)   '/) 00107*10
                                                                        00108*10
                   WRITE(6,7030) (AU(I),I=1,NEQ)                         00109*10
                   WRITE(6,7030) (AT(I),I=1,NEQ)
                   WRITE(6,7030) (BT(I),I=1,NEQ)
 7030            FORMAT(5X,12E10.3)                                     00110*10
                                                                        00111**5
************************************************************************00112
                                                                        00113
*******SPECIAL  FORM  FOR  STATIC   ANALYSIS  CONSISTENT  WITH  KSTAR**00114
                                                                        00115
      ******  PRINT  CERTAIN  QUANTITIES  ***********                   00116
                                                                        00117**5
```

```
      CALL REDVK(A1,A0,FAXA,NEQD,NWA,NEQ,NBLOCK,M1,NBAND,NC,KKS)


************************************************************************


        IF(NFQUIR.NE.0) GO TO 5540


    DELT=DD
    DO  590  I=1,NEQ
    U0=X0(I)
    U10=X1(I)
    U20=X2(I)
     U=A0(I)
        GO TO (810,810,800,810),KKS

810     CONTINUE
    U1=A01*U-BT(I)
    U2=A0*U-AT(I)
    X2(I)=C2*U20+C1*U2

        GO TO 820

    ***********  FOR  INELASTIC  ANALYSIS  ONLY  MODIFY  X2(I)  *******
    *******   U  DENOTES  INCREMENTAL  DISPLACEMENT  BRTWEEN ' T+DT ' 
    ********  ' T '  TIME  STEP  **************************************

830     CONTINUE

800     X2(I)=A04*U+A05*U10+A06*U20



820     CONTINUE
    X1(I)=U10+DELT/2.*(U20+X2(I))
    X0(I)=U0+DELT*U10+A08*(X2(I)+2.*U20)
590 CONTINUE

5540 CONTINUE

            WRITE(6,7040)
7040        FORMAT(1H1//5X, '    BEFORE RETURN IN SOLDN  '/
   *                5X,'             A0(I)          '/
   *                5X,'             X0(I)          '/
   *                5X,'             X1(I)          '/
   *                5X,'             X2(I)          '/)
            WRITE(6,7030) (A0(I),I=1,NEQ)
            WRITE(6,7030) (X0(I),I=1,NEQ)
            WRITE(6,7030) (X1(I),I=1,NEQ)
            WRITE(6,7030) (X2(I),I=1,NEQ)

    RETURN
    END
```

```
         DATA SET SOLFIG      AT LEVEL 013 AS OF 83/02/16
      SUBROUTINE SOLFIG(LOOP,IH,NDUM,NN15,NN17)
      IMPLICIT REAL*8(A-H,O-Z)

      CALLS:  MODES,PRINTD
      CALLED BY:  MAIN

      SOLUTION OF THE EIGENVALUE PROBLEM

      COMMON A(1)
      COMMON/ELPAR/NPAR(14),NELTYP
      COMMON/CONTRL/DT,HED(8),RADN,NUMNP,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS
         COMMON/QUAD/CCCC(1500),N1,N2,N3,N4,N5,MTOT,NNN
      COMMON /SOL/ANORM,NBLOCK,NEQB,LL,NF,IDUM,NEIG,NAD,NVV,NFO
      COMMON /EM/ A1(1000),IFILL1(3138)
      COMMON /EXTRA/ MODEX,NT8,IFILL2(14)
      COMMON/ASOL/W1

         DIMENSION IH(NUMNP,1)
      REAL*4 TT(3)
           DIMENSION C(100)
       *********CHANGED *****************
        N1=8

      READ CONTROL CARD

      REWIND  14
      WRITE (6,1003)
      IF(LOOP.NE.1)  GO  TO  101
      READ (5,100) IFPR,IFSS,NITEM,RTOL,COFQ,NFO
      WRITE(14)     IFPR,IFSS,NITEM,RTOL,COFQ,NFO
      GO  TO  102
  101 READ(14)      IFPR,IFSS,NITEM,RTOL,COFQ,NFO

  102 IF  (IFPR.GT.0) IFPR=1
      IF  (IFSS.GT.0) IFSS=1
      IF  (NITEM.EQ.0) NITEM=16
      IF  (RTOL.EQ.0.) RTOL=1.E-05
      IF  (COFQ.EQ.0.) COFQ=1.E00

      IF (NEIG.GT.0) GO TO 10

      WRITE (6,1001)
      GO TO 15
   10 WRITE (6,1002)
   15 WRITE (6,1000) IFPR,IFSS,NITEM,RTOL,COFQ,NFO
20    IF (MODEX.EQ.1) RETURN
      TPI=8.0D0*DATAN(1.0D0)
      COFQ=COFQ*TPI
      COFQ=COFQ*COFQ

      CALL SOLUTION ROUTINE

      CALL MODES(NEQS,MBANDS,NBLOCK,NEQB,NF,MTOT,IFPR,IFSS,RTOL,NITEM,
     *COFQ,NN15,NU15)


      WRITE CONTROL INFORMATION ON TAPE  -- FOR RESTART OPTION
```

```
      NC=2
      REWIND NC
      WRITE (NC) NEW,NBLOCK,NEQB,MBAND,NT,NF,(AT(I),I=1,NF)

      PRINT OF EIGENVALUES (OMEGA) AND EIGENVECTORS

      REWIND NT
      READ (NT) (C(I),I=1,NF)
      K=NF+1
      DO 30 I=1,NF
      K=K-1
      KK=(K-1)*3+1
      C(KK)=C(K)
      C(KK+1)=C(K)/TPI
30    C(KK+2)=TPI/C(K)
      IF (NEIG.GT.0) GO TO 25
      WRITE (6,1009)
      DO 41 I=1,NF
      K1=3*I-2
      K2=3*I
 41   WRITE (6,1020) I,(C(J),J=K1,K2)
      GO TO 35
  25  WRITE (6,1010)
      DO 40 I=1,NF
      K1=3*I-2
      K2=3*I
  40  WRITE (6,1020) I,(C(J),J=K1,K2),AT(NF+I)

      **********CHANGED  ********************************
  35  CONTINUE
      W1=C(2)
        NN16=NN15+2*NF
          NN17=NN16+NEQB*NF


      WRITE (6,1030)
      REWIND   14
      READ(14) IFPR,IFSS,NITEM,RTOL,COFQ,NFO
      WRITE(6,98) ((ID(I,J),J=1,2),I=1,NUMNP)
  98  FORMAT(5X,2I5)
      CALL PRINTD(ID,A(NN15),A(NN16),NEQB,NUMNP,NF,NBLOCK,NEQS,NT,2)

      COMPUTE TIME LOG


 100  FORMAT (3I5,2F10.0,I5)

1000  FORMAT (1H0 // 20H CONTROL INFORMATION, //
     1          5X,31HFLAG FOR ADDITIONAL PRINTING  =, I5 /
     2          7X,14HEQ.0, SUPPRESS, /
     3          7X,11HEQ.1, PRINT,      //
     4          5X,31HSTURM SEQUENCE CHECK FLAG (*) =, I5 /
     5          7X,19HEQ.0, PERFORM CHECK, /
     6          7X,10HEQ.1, PASS,          //
     7          5X,31HMAXIMUM ITERATION CYCLES (*)  =, I5 //
     8          5X,31HCONVERGENCE TOLERANCE (*)     =, E14.4 //
     9          5X,31HCUT-OFF FREQUENCY (CPS)       =, E14.4 //
     *          5X,31HNUMBER OF STARTING ITERATION   ,        /
     *          5X,31HVECTORS TO BE READ FROM        ,        /
     *          5X,31HTAPE10 (*)                    =, I5     ///
```

```
    A          5X,27H(*)  APPLICABLE TO SUBSPACE, /
    B          5X,29H     ITERATION SOLUTIONS ONLY, 1X)
1001 FORMAT (44H0DETERMINANT SEARCH SOLUTION IS CARRIED OUT  )
1002 FORMAT (44H0SUBSPACE ITERATION SOLUTION IS CARRIED OUT  )
1003 FORMAT (1H1,//41H E I G E N V A L U E   A N A L Y S I S   //)
1009 FORMAT (1H1,20HPRINT OF FREQUENCIES  //
    1           23H MODE        CIRCULAR      /
    2           49H NUMBER      FREQUENCY     FREQUENCY     PERIOD        /
    3           49H            (RAD/SEC)  (CYCLES/SEC)    (SEC)          )
1010 FORMAT (1H1,20HPRINT OF FREQUENCIES  //
    1           23H MODE        CIRCULAR     /
    2           56H NUMBER      FREQUENCY     FREQUENCY     PERIOD     TOL
    3ERANCE     /
    4           49H            (RAD/SEC)  (CYCLES/SEC)    (SEC)          )
1020 FORMAT (1H0,I4,6X,4(E10.4,2X))
1030 FORMAT (/// 22H PRINT OF EIGENVECTORS, // 1X)

     RETURN
     END
```

```
          DATA SET SOLST      AT LEVEL 011 AS OF 83/02/02
      SUBROUTINE    SOLST(A4,AU,NEQ,MBAND,XO,NC,NWA,NEQB,NBLOCK,M1,MAXA) 00001
                                                                         00002**3
      IMPLICIT  REAL*8  (A-H,O-Z)                                        00003
                                                                         00004**3
             COMMON/GORA/IFLAG,ITR                                       00005**9
             COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB       00006**9
                                                                         00007**9
         COMMON/BOBBY/KKS,NUMMAT,NG,LOOP,NDRN,NINT,IDWA,NPT              00008*11
      DIMENSION  XO(1),A4(NWA), MAXA(1),AU(1)                            00009
*******************************SPECIAL    FOR  STATIC  ANALYSIS**********00010
                                                                         00011**3
                                                                         00012**3
         CALL REDVK(A4,AU,MAXA,NEQB,NWA,NEQ,NBLOCK,M1,MBAND,NC,KKS)      00013**7
                                                                         00014**7
                                                                         00015**6
                                                                         00016**8
         GO TO (1,1,300,1),KKS                                           00017**8
                                                                         00018**8
                                                                         00019**8
   1   CONTINUE                                                          00020**8
                                                                         00021**8
        DO 100 I=1,NEQ                                                   00022**8
 100    XO(I)=AU(I)                                                      00023**8
                                                                         00024**8
        GO TO 900                                                        00025**8
                                                                         00026**8
                                                                         00027**8
                                                                         00028**6
                                                                         00029**7
                                                                         00030**6
 300 CONTINUE                                                            00031**6
                                                                         00032**9
             IF(NEQUIB.NE.0) GO TO 900                                   00033**9
                                                                         00034**9
      DO  200  I=1,NEQ                                                   00035**4
      UU=XO(I)                                                           00036**4
      U=AU(I)                                                            00037**5
      XO(I)=UU+U                                                         00038**4
 200  CONTINUE                                                           00039**4
 900  CONTINUE                                                           00040
                                                                         00041**3
      RETURN                                                             00042
      END                                                                00043
```

```
          DATA SET SOLTN       AT LEVEL 005 AS OF 83/02/11
      SUBROUTINE    SOLTN(AT,BT,X0,A5,NEQ,X1,X2,A4,NWA,AU,MAXA,NEQB,

     *NBLOCK,M1,MBAND,NC)

      IMPLICIT  REAL  *8 (A-H,O-Z)

      COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSG,NKK,NBAND,NCYCL,NDYN,IPLNA
     *X,ISTRPR,NPRINT,ISTART,NEQR,MBANDH
          COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB
          COMMON/BOBBY/KKS,NUMMAT,NG,LOOP,NDRN,NINT,IDWA,NPT

      DIMENSION  AT(1),BT(1),X0(1),A5(NEQ,1),X1(1),X2(1),A4(NWA),AU(1),
     *MAXA(1)


      DT=TETA*DD
      DO  4500  I=1,NEQ
 4501 AT(I)=0.0
      BT(I)=(1./DT)*X0(I)
 4500 CONTINUE
      DO  459  I=1,NEW
  459 AU(I)=AU(I)+A5(I,1)*BT(I)
      DO  451  N=1,NEQ
      LQ=N-1
      MR=MIN0(MBAND,NEQ-N+1)
      IF(MR.EQ.1)  GO  TO  3600
      DO  3200  J=2,MR
      M=LQ+J
 3201 AU(N)=AU(N)+A5(N,J)*BT(M)
      AU(M)=AU(M)+A5(N,J)*BT(N)
 3200 CONTINUE
  451 CONTINUE
 3600 CONTINUE
 9000 CONTINUE
 8000 CONTINUE
          CALL REDVK(A4,AU,MAXA,NEQB,NWA,NEQ,NBLOCK,M1,MBAND,NC,KKS)

 5540  DO  5900  I=1,NEQ
      U0=X0(I)
      U10=X1(I)
      U=AU(I)
      X1(I)=(1./DT)*(U-U0)
      X0(I)=U
 5900 CONTINUE


      RETURN
      END
```

```
      DATA SET SOLVI       AT LEVEL 051 AS OF 83/03/05
 SUBROUTINE  SOLVI (IH,A1,A3,XM,B,BO,VEL,PDYL,IS,GACL,ID,R,A,AU,
*GD,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEQH,GCB,A4,A5,A6,MAXA,X0,X1,X2,
*AT,BT,AC, SU,NWA,NEQB,NBLOCK,M1,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,
*AN,NSLCI,AN1,EKJ,LC,SURPFX,SURPFY,RE,SW,AS)
 IMPLICIT  REAL*8   (A-H,O-Z)
C

 COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA
*X,ISTRPR,NPRINT,ISTART,NEQR,MBANDH,NEQS,MBANDS
 COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC,ISLC

 COMMON/SOL/ANORM,NGLOCK,NEQD,NDC

*******    COMMON BLOCK INSERTED FOR CHECKING    ****
 COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
******                                           ****

 COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX
       COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT
 COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A08
      COMMON/PRTM/ISTRES,MPRINT,JPRINT,KPRINT,IP
      COMMON/GORA/IFLAG,ITR

 COMMON/DIMEL/N43,IP1,IPT1,KK

      COMMON/EB/IND,ICOUNT



 DIMENSION A1(NEQS,1),GCB(1),A3(NEQ,1),XM(1),B(1),BO(1),VEL(1),
*PDYL(2,4,1),IS(1),GACL(1),TD(2),HEDG(8),ID(NUMNP,1),R(1),A(1),
*AU(1),GB(1),PMAX(NUMEL,1),AMAX(NUMNP,12,2),DS(4,3),IH(NUMNP,1)

 DIMENSION AS(NEQS,1)

 DIMENSION  A4(NWA),A5(NEQ,1),A6(NWA),MAXA(1),X0(1),X1(1),X2(1),AT(
*1),BT(1),AC(1),SU(1),X(1),Y(1),Q(1),ISC(1),JSC(1),SURTRX(2,NDC,2),
*SURTRY(2,NDC,2),AN(NNSC,1),AM(1),EK(1),AN1(NYP,1),EKJ(NYP),NSLCI(1
*),SURPFX(2,NDC,2),SURPFY(2,NDC,2)

 DIMENSION  RE(1)
 DIMENSION  SW(1)

    COMMON/TAPES/NSTIF,NRED,NL,NR,IFIL(2),NUMBER,LCOUNT


  *********   TEMPORARY  STORAGE  FOR  LOAD  VECTOR  ADDITION *****


    IND=-1
    IF(KKS.EQ.3.AND.IND.EQ.-1) GO TO 500
400 CONTINUE
```

```
     NC=0
        DT=0.0
       IND=NC
     TK=NCYCL*DD

         TM=0.0

**********    BRANCH  FOR  WORK/NG ARRAY  INITIALIZATION  IN  ELT2D6  **

     ******   INITIALIZE  GLOBAL  LOAD  VECTOR  *******
 700  DO  1050  I=1,NEQ
1050  A0(I)=0.0

      ***************************

       WRITE(6,100)
 100   FORMAT(5X,'     PRINT OF BELA IN SOLVI BEFORE ELEMT       ')
        WRITE(6,105) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
 105   FORMAT(5X,10I5)

      ********************

     ********   E L A S T O P L A S T I C   A N A L Y S I S   S T A R T

     IF(KKS.NE.3)  GO   TO  600
     IF(NC.EQ.0)   GO   TO  600

     CALL  ELEMT(NUMMAT)

     CALL  KSTAR(IH,A1,A3,XM,NEQ,MBAND,NEQH,MBANDH,A4,A5,A6,MAXA,
    *NWA,NEQB,NBLOCK,MI,ID,KE,BW)

     ****************           /

      WRITE(6,105)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
 600 CONTINUE

      ******   ASSEMBLE  LOAD  VECTOR***

     DO  9000 I=1,NEQ
9000 SO(I)=0.0

     CALL      LOAD1(NUMNP,NEQ,LC,ISLC,DT,NC,ISC,JSC,SURTRX,SURTRY,X,Y,Q0
    *,ID,SO,TK,TM,PDTL,SURPFX,SURPFY)

     DO  1000 I=1,NEQ
1000 R(I)=0.0
```

```
    CALL  LOAD2(NUMNP,FDYL,X,NUMLP,ID,DT,ID,NC,IK,R)                   00103**6
                                                                       00104
                                                                       00105*37
  1      CONTINUE                                                      00106*37
                                                                       00107*37
         DO 5 I=1,NEQ                                                  00108*37
  5      AO(I)=SO(I)+R(I)                                              00109*37
                                                                       00110*37
                                                                       00111*37
  2      CONTINUE                                                      00112*37
                                                                       00113*43
                                                                       00114*27
                                                                       00115*27
                                                                       00116*37
  6      CONTINUE                                                      00117*37
                                                                       00118*37
                                                                       00119*38
                                                                       00120*38
        GO TO (51,51,51,52),KKS                                        00121*38
                                                                       00122*38
 52     CONTINUE                                                       00123*38
                                                                       00124*38
                                                                       00125*38
    ********   ONLY  FOR  LIMIT  ANALYSIS  PROBLEM   ****************00126*27
                                                                       00127*27
         IF(IPT1.EQ.2.AND.LC.EQ.NLC) GO TO 55                          00128*39
         IF(IPT1.EQ.1.AND.LC.EQ.NLC) GO TO 55                          00128*39
                                                                       00129*16
                                                                       00130*20
                                                                       00131*38
 51     CONTINUE                                                       00132*38
                                                                       00133*20
 **************    SAVE  LOAD  VECTOR  FOR  EACH  CYCLE  BEFORE  ITER00134*20
                                                                       00135*20
                                                                       00136*20
    DO  49  I=1,NEQ                                                    00137*20
                                                                       00138*28
 49 GB(I)=AO(I)                                                        00139*30
                                                                       00140*22
                                                                       00141*39
         IF(KKS.NE.3) GO TO 55                                         00142*39
                                                                       00143*39
    *******  CALCULATE NORM OF THE ACTUAL INCREMENTAL LOAD APPLIED ***00144*22
                                                                       00145*22
                                                                       00146*25
                                                                       00147*25
    RNORM=0.                                                           00148*25
    DO  50  I=1,NEQ                                                    00149*25
    RNORM=RNORM+GB(I)*GB(I)                                            00150*30
 50 CONTINUE                                                           00151*25
                                                                       00152*25
                                                                       00153*25
         DO 800 I=1,NEQ                                                00154*43
800      AO(I)=SO(I)+R(I)-RE(I)                                        00155*43
                                                                       00156*22
                                                                       00157*20
 55      CONTINUE                                                      00158*39
                                                                       00159*20
                                                                       00160*50
        IF(NC.EQ.0) GO TO 10                                           00161*50
```

```
                                                                        00162*50
      LCOUNT=LCOUNT+1
      WRITE(6,105) LCOUNT
                                                                        00163*51
                                                                        00165*51
                                                                        00167*50
          IF(LCOUNT.NE.ISTRFR) GO TO 20                                 00168*50
                                                                        00169*50
                                                                        00170*50
 10   CONTINUE
                                                                        00171*50
                                                                        00172*16
      WRITE(6,45)                                                       00173*16
 45   FORMAT(1H1/'   ***  PRINT  OF  ASSEMBLED  LOAD  VECTOR IN SOLV1 ') 00174*16
      WRITE(6,46) (AO(I),I=1,NEQ)                                       00175*16
 46   FORMAT(5X,9E12.5)                                                 00176*16
                                                                        00177**6
 20       CONTINUE                                                      00178*50
                                                                        00179*35
           IF(NC.NE.0) GO TO 42                                         00180*43
            DO 43 I=1,NEQ                                               00181*43
 43          AO(I)=AO(I)-SW(I)                                          00182*43
 42         CONTINUE                                                    00183*43
                                                                        00184*43
                                                                        00185**6
      ********   CHECK  OVERALL  LOGIC  FOR  ALL  MATERIAL  MODEL     **00186*11
                                                                        00187*11
          IF(NC.EQ.0) GO TO 41                                          00188*44
                                                                        00189**6
      ISTRES=ISTRES+1                                                   00190
      MPRINT=MPRINT+1                                                   00191
      JPRINT=JPRINT+1                                                   00192
      KPRINT=KPRINT+1                                                   00193
                                                                        00194*20
 41        CONTINUE                                                     00195*44
      ******   PROCEED  FOR  ITERATION  *********************************00196*20
                                                                        00197*21
      *******   SET  ITERATION  NUMBER    ********                      00198*21
                                                                        00199*25
                                                                        00200*25
      *************          IFLAG. EQ. 0  MEANS  NOT  CONVERGED  VALUES00201*25
      *************          IFLAG. EQ. 1  MEANS       CONVERGED  VALUES00202*25
                                                                        00203*25
      IFLAG=0                                                           00204*21
      NITR=ITERX                                                        00205*39
                                                                        00206*39
                                                                        00207*23
      ITR=0                                                             00208*23
                                                                        00209*32

       WRITE(6,105)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                        00210*32
          IF(NC.NE.0) GO TO 440                                         00211*32
          CALL SYMSOL (1,A3,AO,NEQ,MBAND)                               00212*32
          CALL SYMSOL(2,A3,AO,NEQ,MBAND)                                00213*32
                                                                        00214*39
                                                                        00215*39
           IF(KRS.EQ.3)   GO TO 451                                     00216*39
```

```
      DO 450 I=1,NEQ
450   X0(I)=A0(I)

       GO TO 420
451     CONTINUE

        IF(NEQUIB.NE.0) GO TO 453
        DO 452 I=1,NEQ
452      X0(I)=X0(I)+A0(I)

453      CONTINUE

      GO TO 420
440   CONTINUE


   CALL SOLSI(A4,A0,NEQ,MBAND,X0,NC,NWA,NEQB,NBLOCK,MI,MAXA)

420      CONTINUE
    WRITE (6,105) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017


   IF(NEQUIB.EQ.0)  GO  TO  430


    *********     COUNTER  FOR  DUNAMIC  EQUILIBRIUM   **************


   CALL       DEQITR(X0,NDYN,A0,RE,R,ITR,NITR,ITEMX,RTOL,NC,A4,NEQ,
  *MBAND,NWA,NEQB,NBLOCK,MI,MAXA,X2,X1,VEL,AC,B,ID,AMAX,PMAX,AM,EK,AN
  *,AN1,EKJ,TM,NUMNP,NUMEL,RNORM,XM,A5,GB,AT,B1,
  * A3,LC)


430 CONTINUE

   IFLAG=1
```

```
 500 CONTINUE
     CALL        OUT(X1,X0,X2,VEL,AC,AO,B,ID,DS,AMAX,PMAX,NUMNP,NUMEL,
    *NC,AM,EK,AN,AN1,EKJ,TM,LC,NEQ,RE,ITR)
        IF(IND.EQ.-1) GO TO 400
          IF(NC.EQ.0) DT=TETA*DD
        IF(NC.EQ.0) TM=TM+DT
       IF(NC.EQ.0) GO TO 15
       IF(LCOUNT.NE.ISTRPR) GO TO 30
       LCOUNT=0
  15   CONTINUE
         CALL EQCHK(XM,A5,B,NEQ,MBAND,X0,X1,X2,R,NC,A3,NDYN,GB,AO,AS)
  30   CONTINUE
         NC=NC+1
         ************      EQULIBRIUM CHECK FOR EVERY LOAD STEP ****
         IF(NC-NCYCL)700,700,900
 900 CONTINUE
      RETURN
2010 FORMAT(//// 37H EQULIBRIUM ITERATION IN TIME STEP = ,I5//
   *             37H NUMBER OF ITERATIONS                = ,I5  /)
2020 FORMAT(////46H ITERATION LIMIT REACHED  S T O P OF SOLUTION    )
2050 FORMAT(////70H OUT-OF-BALANCE LOADS LARGER THAN INCREMENTAL LOADS
    *AFTER ITERATION = ,I5/)
     END
```

```
      DATA SET SOLV2       AT LEVEL 032 AS OF 83/03/05
 SUBROUTINE  SOLV2 (IH,A1,A3,XM,B,B0,VEL,PDYL,TS,GACL,TD,R,A,A0,      00001
                                                                      00002*11
*GB,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEQH,GCB,A4,A5,A6,MAXA,X0,X1,X2,00003*20
                                                                      00004*11
*AT,BT,AC, SO,NWA,NEQB,NBLOCK,MI,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,00005
                                                                      00006*11
*AN,NSLCI,ANI,EKJ,LC,SURFFX,SURFFY,RE,SW,AS)                          00007*18
                                                                      00008*11
 IMPLICIT  REAL*8   (A-H,O-Z)                                         00009
                                                                      00010**3
                                                                      00011**3
                                                                      00012**3
 COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA00013
*X,ISTRPR,NPRINT,ISTART,NEQK,MBANDH,NEQS,MBANDS                       00014**3
       COMMON/INTEGE/IETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX    00015*21
      COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT              00016*27
      COMMON/GORA/IFLAG,ITR                                           00017*27
                                                                      00018*27
 COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A08 00019
 COMMON/SOL/ANORM,NBLOCK,NEQB,NDC

                                                                      00020*14
 COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC,ISLC                          00021*14
                                                                      00022**3
  COMMON/EL/IND,ICOUNT                                                00023*22
                                                                      00024*22
                                                                      00025**3
 DIMENSION A1(NEQS,1),GCB(1),A3(NEQ,1),XM(1),B(1),B0(1),VEL(1),       00026*20
*PDYL(2,4,1),TS(1),GACL(1),TD(2),HEDG(8),ID(NUMNP,1),R(1),A(1),       00027
*A0(1),GB(1),PMAX(NUMEL,1),AMAX(NUMNP,12,2),DS(4,3),IH(NUMNP,1)       00028*29
                                                                      00029**3

 DIMENSION AS(NEQS,1)                                                                    00030**3
 DIMENSION  A4(NWA),A5(NEQ,1),A6(NWA),MAXA(1),X0(1),X1(1),X2(1),AT(00031
*1),BT(1),AC(1),SO(1),X(1),Y(1),Q(1),ISC(1),JSC(1),SURTRX(2,NDC,2),00032
*SURTRY(2,NDC,2),AN(NNSC,1),AM(1),EK(1),ANI(NYP,1),EKJ(NYP),NSLCI(100033**5
*),SURFFX(2,NDC,2),SURFFY(2,NDC,2)                                    00034**2
                                                                      00035**3
 DIMENSION  RE(1)                                                     00036*12
                                                                      00037**3
 DIMENSION  SW(1)                                                     00038*18
    COMMON/TAPES/NSTIF,NRED,NL,NR,IFIL(2),NUMBER,LCOUNT               00039*30
     COMMON/PRIM/ISTRES,MPRINT,JPRINT,KPRINT,IP                       00040*29
                                                                      00041
 ************     TEMPORARY  STORAGE    ********                      00042
       IND=-1                                                         00043*21
       IF(KKS.EQ.3.AND.IND.EQ.-1) GO TO 600                           00044*21
400    CONTINUE                                                       00045*21
                                                                      00046
 NC=0                                                                 00047**2
 TK=NCYCL*DD                                                          00048
       DT=0.0                                                         00049*21
       IND=NC                                                         00050*21
                                                                      00051*15
                                                                      00052*15
    TM=0.0                                                            00053*30
                                                                      00054*15
                                                                      00055
 **********   INITIALIZE  GLOBAL  LOAD  VECTOR   **********           00056
```

```
 700  DO  1050  I=1,NEQ
1050  AO(I)=0.0




        IF(KKS.NE.3) GO TO 860
        IF(NC.EQ.0) GO TO 860
 861 CONTINUE

        CALL ELEMT(NUMMAT)
        CALL KSTAR(IH,A1,A3,XM,NEQ,MBAND,NEQH,MBANDH,A4,A5,A6,MAXA,NWA

    *,NEQB,NBLOCK,MI,ID,RE,SW)

 860     CONTINUE



     DO  9000 I=1,NEQ
9000 SO(I)=0.0
      ******   ASSEMBLE  LOAD  VECTOR***

     CALL      LOAD1(NUMNP,NEQ,LC,ISLC,DT,NC,ISC,JSC,SURTRX,SURTRY,X,Y,Q

    *,ID,SO,TK,TM,PDYL,SURPFX,SURPFY)


     DO  1000 I=1,NEQ
1000 R(I)=0.0
     CALL  LOAD2(NUMNP,PDYL,X,NUMEP,ID,DT,TD,NC,TK,R)




   1         CONTINUE
     DO  800  I=1,NEQ
 800 AO(I)=SO(I)+R(I)



   2      CONTINUE







        DO 49 I=1,NEQ
  49    GB(I)=AO(I)
         IF(KKS.NE.3) GO TO 3

        RNORM=0.0
        DO 50 I=1,NEQ
  50    RNORM=RNORM+GB(I)*GB(I)
```

```
         **********     RESIDUAL LOAD VECTOR CORRECTION ON CONVERGED ST00117*27
                                                                        00118*27
             DO 850 I=1,NEQ                                             00119*27
850            AO(I)=AO(I)-RE(I)                                        00120*27
                                                                        00121*27
                                                                        00122*27
  3           CONTINUE                                                  00123*27
        IF(NC.EQ.0) GO TO 10                                            00124*31
                                                                        00125*31
                                                                        00127*32
                                                                        00129*31
        IF(LCOUNT.NE.ISTRPR) GO TO 20                                   00130*31
                                                                        00131*31
                                                                        00132*31
 10 CONTINUE
                                                                        00133*31
                                                                        00134**5
                                                                        00135*15
                                                                        00136*15
    WRITE(6,42)                                                         00137*15
 42 FORMAT(1H1/'   ***  PRINT  OF  ASSEMBLED  LOAD  VECTOR   IN SOLV2 ')00138*15
                                                                        00139*15
    WRITE(6,46) (AO(I),I=1,NEQ)                                         00140*15
 20      CONTINUE                                                       00141*31
                                                                        00142*27
             IF(NC.NE.0) GO TO 43                                       00143*27
              DO 44 I=1,NEQ                                             00144*27
 44          AO(I)=AO(I)-SW(I)                                          00145*27
 43          CONTINUE                                                   00146*27
                                                                        00147*15
                                                                        00148*16
 46 FORMAT(5X,10E12.5)                                                  00149*16
    *******    E L A S T O P L A S T I C   P A R T N O T C O M P L E T E00150*13
                                                                        00151*27
             IF(NC.NE.0)  GO TO 45                                      00152*27
                                                                        00153*13
    ISTRES=ISTRES+1                                                     00154
    MPRINT=MPRINT+1                                                     00155
    JPRINT=JPRINT+1                                                     00156
    KPRINT=KPRINT+1                                                     00157
 45          CONTINUE                                                   00158*27
                                                                        00159*27
                                                                        00160*27
             IFLAG=0                                                    00161*27
             NITR=ITEMX                                                 00162*27
             ITR=0                                                      00163*27
                                                                        00164*27
                                                                        00165**5
                                                                        00166**5
                                                                        00167**5
    IF(NC.NE.0)  GO  TO  500                                            00168**2
    CALL SYMSOL(1,A3,AO,NEQ,MBAND)                                      00169**2
    CALL SYMSOL(2,A3,AO,NEQ,MBAND)                                      00170**2
                                                                        00171*17
            IF(KKS.EQ.3) GO TO 451                                      00172*27
                                                                        00173*27
                                                                        00174*17
         *********   KKS.EQ.3   AO(I) =  INCREMENTAL *****              00175*26
         ********    KKS.NE.3)  AO(I) =  TOTAL       *****              00176*26
```

```
      DO  47  I=1,NEQ
   47 X0(I)=A0(I)
            GO TO 650

  451       CONTINUE

           IF(NEQUIB.NE.0) GO TO 601
            DO 452 I=1,NEQ
  452         X0(I)=X0(I)+A0(I)

             GO TO 650


  601 CONTINUE



C*******    SOLVE  THE  SIMULYTENEOUS  EQUATION  FOR  DISPLACEMENT  VEC
  500 CONTINUE
      CALL          SOLTN(AT,BT,X0,A5,NEQ,X1,X2,A4,NWA,A0,MAXA,NEQB,

     *NBLOCK,MI,MBAND,NC)



  650          CONTINUE

         IF(NEQUIB.EQ.0) GO TO 600
         CALL DEQITR(X0,NDYN,A0,RE,R,ITR,NITR,ITEMX,RTOL,NC,A4,NEQ,

     *MBAND,NWA,NEQB,NBLOCK,MI,MAXA,X2,X1,VEL,AC,B,ID,AMAX,PMAX,AM,EK,AN



     *,AN1,EKJ,IM,NUMNP,NUMEL,RNORM,XM,A5,GB,AT,BT,A3,LC)



  600 CONTINUE
      IFLAG=1

      CALL         OUT(X1,X0,X2,VEL,AC,A0,B,ID,DS,AMAX,PMAX,NUMNP,NUMEL,

     *NC,AM,EK,AN,AN1,EKJ,IM,LC,NEQ,RE,ITR)

       ******  CHECK  FOR  TRANSIENT  PROBLEM   TIME  STEP  CONSIDER 0 T

       IF(IND.EQ.-1) GO TO 400
```

```
          IF(NC.EQ.0) DT=TETA*DD                                  00237*28
    IF(NC.EQ.0) TM=TM+DT                                          00238*21
                                                                  00239*21

    IF(NC.EQ.0) GO TO 35

      IF(LCOUNT.NE.ISTRPR) GO TO 30                               00240*31
                                                                  00241*31
      LCOUNT=0                                                    00242*31
                                                                  00243*21
 35   CONTINUE
                                                                  00244*27
        CALL EQCHK(XM,A5,B,NEQ,MBAND,X0,X1,X2,R,NC,A3,NDYN,GB,A0,AS)00245*28
                                                                  00246*27
                                                                  00247*27
                                                                  00248**5
 30   CONTINUE                                                    00249*31
                                                                  00250*26
        NC=NC+1                                                   00251*26
                                                                  00252*26
                                                                  00253**5
                                                                  00254*19
                                                                  00255*19
   IF(NC-NCYCL)  700,700,900                                      00256*19
                                                                  00257*19
900 CONTINUE                                                      00258
    RETURN                                                        00259
    END                                                           00260
```

```
       DATA SET SOLV3       AT LEVEL 028 AS OF 83/03/05
   SUBROUTINE  SOLV3 (IH,A1,A3,XM,B,BO,VEL,PDYL,TS,GACL,ID,R,A,AO,

  *GB,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEQH,GCB,A4,A5,A6,MAXA,X0,X1,X2,

  *AT,BT,AC, SO,NWA,NEQB,NBLOCK,MI,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,

  *AN,NSLCI,ANI,EKJ,LC,SURPFX,SURPFY,RE,SW,AS)

   IMPLICIT  REAL*8   (A-H,O-Z)

   COMMON/CONTL/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA
  *X,ISTRPR,NPRINT,ISTART,NEQK,MBANDH,NEQS,MBANDS
        COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX
       COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT
       COMMON/GORA/IFLAG,ITR
   COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A08

   COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC,ISLC

   COMMON/SOL/ANORM,NGLOCK,NEQD,NDC


   DIMENSION A1(NEQS,1),GCB(1),A3(NEQ,1),XM(1),B(1),BO(1),VEL(1),
  *PDYL(2,4,1),TS(1),GACL(1),TD(2),HEDG(8),ID(NUMNP,1),R(1),A(1),
  *AO(1),GB(1),PMAX(NUMEL,1),AMAX(NUMNP,12,2),DS(4,3),IH(NUMNP,1)

    DIMENSION AS(NEQS,1)

   DIMENSION  A4(NWA),A5(NEQ,1),A6(NWA),MAXA(1),X0(1),X1(1),X2(1),AT(
  *1),BT(1),AC(1),SO(1),X(1),Y(1),Q(1),ISC(1),JSC(1),SURTRX(2,NDC,2),
  *SURTRY(2,NDC,2),AN(NNSC,1),AM(1),EK(1),ANI(NYP,1),EKJ(NYP),NSLCI(1
  *),SURPFX(2,NDC,2),SURPFY(2,NDC,2)


   DIMENSION  RE(1)
   DIMENSION SW(1)
      COMMON/TAPES/NSTIF,NRED,NL,NR,IFIL(2),NUMBER,LCOUNT
       COMMON/PRTM/ISTRES,MPRINT,JPRINT,KPRINT,IP

   COMMON/EL/IND,ICOUNT

   COMMON/BENGY/SG(5,200,4,4),NCOND,NTENS

     REWIND 58
     REWIND 59


        IND=-1
        IF(KKS.EQ.3.AND.IND.EQ.-1) GO TO 200
455  CONTINUE


    *******  TEMPORARY  STORAGE  *********
   TK=NCYCL*DD
      NC=0
       DT=0.0

      TM=0.0
```

```
         IND=NC                                                         00054*20
                                                                        00055*20
                                                                        00056*10
                                                                        00057*10
      ******  BRANCH  FOR  WORKING  ARRAY  INITIALIZATION  IN ELT2D6   *00058*10
                                                                        00059*10
                                                                        00060*10
                                                                        00061**5
      REWIND  12                                                        00062**5
                                                                        00063*20
                                                                        00064*20
 700  DO  1050  I=1,NEQ                                                 00065
1050  AO(I)=0.0                                                         00066
                                                                        00067**5
       IF(NC.EQ.0) GO TO 701
       LCOUNT=LCOUNT+1
 701  CONTINUE
      GO  TO  (400,300,400,400),KKS                                     00068**5
 300  IF(LOOP.EQ.1)  GO  TO  400                                        00069**5
      READ(12) (AO(I),I=1,NEQ)                                          00070**5
      GO  TO  500                                                       00071**5
                                                                        00072
                                                                        00073
       ******  ASSEMBLE  LOAD  VECTOR***                                00074
                                                                        00075
 400 CONTINUE                                                           00076**5
                                                                        00077*10
                                                                        00078*10
                                                                        00079*10
      ******  E L A S T O P L A S T I C ANALYSIS  BRANCHING  ***********00080*10
                                                                        00081*10
      IF(KKS.NE.3)  GO  TO  850                                         00082*10
      IF(NC.EQ.0)  GO  TO  850                                          00083*13
                                                                        00084*10
      CALL  ELEMT(NUMMAT)                                               00085*10
                                                                        00086*11
      CALL        KSTAR(IH,A1,A3,XM,NEQ,MBAND,NEQH,MBANDH,A4,A5,A6,MAXA 00087*11
                                                                        00088*11
     *,NWA,NEQB,NBLOCK,M1,ID,NE,SW)                                     00089*15
                                                                        00090*10
                                                                        00091*10
      *******  RETURN  RESIDUAL  STRESS  VECTOR  FOR  SUBSEQUENT CALCULA00092*10
                                                                        00093*10
                                                                        00094*10
 850 CONTINUE                                                           00095**8
                                                                        00096**5
                                                                        00097**5
      IF(NDYN.EQ.2)  GO  TO  601                                        00098
                                                                        00099*13
                                                                        00100*13
      DO  880  I=1,NEQ                                                  00101*13
 880 SO(I)=0.0                                                          00102*13
                                                                        00103*20
                                                                        00104*20
      CALL      LOADI(NUMNP,NEQ,LC,ISLC,DT,NC,ISC,JSC,SURTRX,SURTRY,X,Y,000105
```

```
                                                                        00106**6
                                                                        00107**6
    *,ID,SU,IK,TM,PDYL,SURPFX,SURPFY)                                   00108**6
                                                                        00109
                                                                        00110
                                                                        00111*10
                                                                        00112*10
                                                                        00113*10
      *******   MODIFIED  LOAD  VECTOR  CALCULATION  BOTH ELASTIC OR INE00114*10
                                                                        00115*10
                                                                        00116*10
      DO  9000  I=1,NEQ                                                 00117
9000  R(I)=0.0                                                          00118**6
                                                                        00119**6
      CALL LOAD2(NUMNP,PDYL,X,NUMLP,ID,DT,TD,NC,TK,TM,R)                00120**6
                                                                        00121
      DO  8000  I=1,NEQ                                                 00122
8000  GB(I)=0.0                                                         00123*16
                                                                        00124*20
                                                                        00125*20
       GO  TO  600                                                      00126
                                                                        00127*12
                                                                        00128*12
    ******   EARTHQUAKE  ANALYSIS  *************************************00129*12
                                                                        00130*13
                                                                        00131*13
                                                                        00132*12
 601  CONTINUE                                                          00133
      CALL LOAD3(IS,GACL,DT,NACL,XM,ID,NUMNP,NC,TK,TM,GB,ACCL)          00134*16
 600  CONTINUE                                                          00135
                                                                        00136
      *********   TOTAL  LOAD  VECTOR  ASSEMBLED   *********************00137*12
                                                                        00138*12
      DO  800  I=1,NEQ                                                  00139
 800  AU(I)=SU(I)+R(I)+GB(I)                                            00140*16
                                                                        00141*11
       WRITE(6,46) (SU(I),I=1,NEQ)
       WRITE(6,46) (AU(I),I=1,NEQ)
                                                                        00142*22
                                                                        00143*22
      IF(KKS.NE.2)  GO  TO  500                                         00144**5
      IF(LOOP.NE.1)  GO  TO  500                                        00145**5
      WRITE(12)  (AU(I),I=1,NEQ)                                        00146**5
 500  CONTINUE                                                          00147**5
                                                                        00148**5
        DO 49 I=1,NEQ                                                   00149*20
  49    GB(I)=AU(I)                                                     00150*21
                                                                        00151*25
                                                                        00152*25
          IF(KKS.NE.3)  GO TO 1                                         00153*25
                                                                        00154*25
        RNORM=0.0                                                       00155*20
        DO 50 I=1,NEQ                                                   00156*20
  50    RNORM=RNORM+GB(I)*GB(I)                                         00157*21
                                                                        00158*21
                                                                        00159*23
   2           CONTINUE                                                 00160*23
                                                                        00161*23
            DO 860 I=1,NEQ                                              00162*23
 860        AU(I)=AU(I)-RE(I)                                           00163*23
```

```


 1         CONTINUE

       IF(NC.EQ.0) GO TO 10


       IF(LCOUNT.NE.ISTRPR) GO TO 20

10     CONTINUE


       WRITE(6,45)
45     FORMAT(1H1/'   ASSEMBLED LOAD VECTOR IN SOLV3 ***** '/)

       WRITE(6,46) (GB(I),I=1,NEQ)
46     FORMAT(5X,11E10.3)

20      CONTINUE

          IF(NC.NE.0) GO TO 48
           DO 43 I=1,NEQ
43         AO(I)=AO(I)-SW(I)
48        CONTINUE



          IF(NC.EQ.0) GO TO 42
    ********   CHECK  OVERALL  LOGIC  FOR  ALL  MATERIAL  MODEL      **


    ISTRES=ISTRES+1
    MPRINT=MPRINT+1
    JPRINT=JPRINT+1
    KPRINT=KPRINT+1

42         CONTINUE
      IFLAG=0
         NITR=ITEMA

      ITR=0


       IF(NC.NE.0) GO TO 440
       CALL SYMSOL (1,A3,AO,NEQ,MBAND)
       CALL SYMSOL(2,A3,AO,NEQ,MBAND)
          IF(KKS.EQ.3) GO TO 451

        ******  KKS.EQ.3  AO(I) =   INCREMENTAL   *******
        *****   KKS.NE.3  AO(I) =   TOTAL          *****


       DO 450 I=1,NEQ
450 XO(I)=AO(I)+XO(I)


          GO TO 420

451        CONTINUE
```

```
           IF(NEQUIB.NE.0) GO TO 453
            DO 452 I=1,NEQ
452         X0(I)=X0(I)+A0(I)
453         CONTINUE

       GO TO 420
440    CONTINUE


        I
C*******   SOLVE  THE  SIMULYTENEOUS  EQUATION  FOR  DISPLACEMENT  VEC

     CALL          SOLDN(AT,BT,X0,A5,NEQ,X1,X2,A4,NWA,A0,MAXA,NEQB,

   *NBLOCK,MI,MBAND,NC ,XM)

420     CONTINUE
        IF(NEQUIB.EQ.0) GO TO 200

          CALL DEQITR(X0,NDYN,A0,RE,R,ITR,NITR,ITEMX,RTOL,NC,A4,NEQ,

   *MBAND,NWA,NEQB,NBLOCK,MI,MAXA,X2,X1,VEL,AC,B,ID,AMAX,PMAX,AM,EK,AN

   *,ANI,EKJ,TM,NUMNP,NUMEL,RNORM,XM,A5,GB,AT,BT,A3,LC)



  ******       INPUT  DISPLACEMENT  TO  ' OUT '  SUBROUTINE  TO  CALCUL
 *******     STRESS  VECTOR  AND  RESPONSES       ***************

200 CONTINUE

    IFLAG=1

    CALL          OUT(X1,X0,X2,VEL,AC,A0,B,ID,DS,AMAX,PMAX,NUMNP,NUMEL,

   *NC,AM,EK,AN,ANI,EKJ,TM,LC,NEQ,RE,ITR)



      IF(IND.EQ.-1) GO TO 455


            IF(NC.EQ.0) DT=TETA*DD
         IF(NC.EQ.0) TM=TM+DT

        IF(NC.EQ.0) GO TO 35

        IF(LCOUNT.NE.ISTRPR) GO TO 30

         LCOUNT=0

 35        CONTINUE
```

```
      CALL FQCHK(XM,A5,B,NEQ,MBAND,X0,X1,X2,K,NC,A3,NDYN,GB,A0,AS)   00278*24
                                                                      00279*23
 30     CONTINUE                                                      00280*27
                                                                      00281**5
        NC=NC+1                                                       00282*22
                                                                      00283*22
                                                                      00284**5
         IF(NC-NCYCL)700,700,900                                      00285*25
900 CONTINUE                                                          00286
901 FORMAT(5X,3I5)
    IF(KKS.NE.4) GO TO 990
    IF(NDYN.NE.1) GO TO 990
    IF(NTENS.NE.1) GO TO 990
    ******  WRITE HERE LAST DISPL. AND VELOCITY COMPONENTS FOR REPEAT RUN
    WRITE(58,902) (X0(I),I=1,NEQ)
    WRITE(59,902) (X1(I),I=1,NEQ)


902  FORMAT(5X,12E10.3)
990 CONTINUE
    RETURN                                                            00287
    END                                                               00288
```

```
        DATA SET SOLVED      AT LEVEL 035 AS OF 83/02/28
  SUBROUTINE  SOLVED(IH,A1,A3,XM,B,BO,VEL,PDYL,TS,GACL,ID,R,A,AO,

 *GB,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEGH,GCB,A4,A5,A6,MAXA,X0,X1,X2,

 *AT,BT,AC, SO,NWA,NEQB,NBLOCK,MI,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,

 *AN,NSLCI,ANI,EKJ,LC,SURPFX,SURPFY,RE,SW,AS)

  IMPLICIT  REAL*8   (A-H,O-Z)

  COMMON/CONTR/DD,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL,NDYN,IPLNA
 *X,ISTRPR,NPRINT,ISTART,NEQK,MBANDH,NEQS,MBANDS
      COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT

  COMMON/ACNST/DELT1,DELT2,C1,C2,A0,A01,A02,A03,A04,A05,A06,A07,A08
  COMMON/INTEGE /TETA,BETA,GAMA


      COMMON/PRTM/ISTRES,MPRINT,JPRINT,KPRINT,IP

      COMMON/SOL/ANORM,NGLOCK,NEDB,NDC



  DIMENSION A1(NEQS,1),GCB(1),A3(NEQ,1),XM(1),B(1),BO(1),VEL(1),
 *PDYL(2,4,1),TS(1),GACL(1),TD(2),HEDG(8),ID(NUMNP,1),R(1),A(1),
 *AO(1),GB(1),PMAX(NUMEL,1),AMAX(NUMNP,12,2),DS(4,3),IH(NUMNP,1)


  DIMENSION AS(NEQS,1)

  DIMENSION  A4(NWA),A5(NEQ,1),A6(NWA),MAXA(1),X0(1),X1(1),X2(1),AT(
 *1),BT(1),AC(1),SO(1),X(1),Y(1),Q(1),ISC(1),JSC(1),SURTRX(2,NDC,2),
 *SURTRY(2,NDC,2),AN(NNSC,1),AM(1),EK(1),ANI(NYP,1),EKJ(NYP),NSLCI(1
 *),SURPFX(2,NDC,2),SURPFY(2,NDC,2)


  COMMON/ASIM/NYP,NSC,NSLC,NLC,NNYP,NNSC,ISLC
  COMMON /TAPES/ NSTIF,NRED,NL,NR,IFILL(2)

        COMMON/GRAND/COH(20),PHI(20),ITYPE(20)

  DIMENSION  RE(1)
  DIMENSION SW(1)


    *******   TEMPRARY  ALLOCATION  FOR  SURFACE  LOAD  VECTOR
     COMMON/KRISNA/SCALB(50),SCALF(50),LL




  CALL        INIT(NUMEL,NUMNP,NEQ,PMAX,AMAX,TD, X0,X1,X2,A,BO,AO,VEL
 *,B)



  CALL  CONST

```

```
      LLL=NDYN                                                          00055
                                                                        00056
              IF(LOOP.NE.1) GO TO 195                                   00057*32
              IF(KKS.EQ.4.AND.LC.GT.1) GO TO 195                        00058*32
                                                                        00059*32
                                                                        00060
                                                                        00061
*********READ  HERE   LINEARIZED  YIELD  SURFACE   **************       00062
                                                                        00063*24
           DO 160 N=1,NUMMAT                                            00064*25
                                                                        00065*25
               READ(5,150) ITYPE(N)                                     00066*32
                                                                        00067*25
 160       READ(5,170) COH(N),PHI(N)                                    00068*24
 150       FORMAT(I5)                                                   00069*24
 170       FORMAT(2E10.3)                                               00070*24
                                                                        00071*24
                                                                        00072*24
                                                                        00073*24
 180 FORMAT(1H1/'****** TYPE OF LINEARIZED YIELD MODEL PROPOSED = ',I5/00074*24
    *   ' EQ. -1      ............... NONE .....................   ' / 00075*24
    *   ' EQ.  0      ............... GIUDA METHOD ..........      ' / 00076*24
    *   ' EQ.  1      ..............  ANDERHEGGEN MODEL .....      '/  00077*24
    *   ' EQ.  2      ..............  MOHR COLUMB MODEL  .....      '/) 00078*24
                                                                        00079*24
                                                                        00080*33
              WRITE(6,190) ((N,ITYPE(N),COH(N),PHI(N)),N=1,NUMMAT)      00081*32
                                                                        00082*31
                                                                        00083*30
 192   CONTINUE                                                         00084*24
                                                                        00085*24
 190   FORMAT(/5X,'   MATERIAL NUMBER   ......              = ',I5   /00086*24
    *         '                    ITYPE(N)          = ',I5/           00087*32
    *   ' COHESION ....COH(N)            ......          = ',E10.3/00088*24
    *   ' ANGLE OF INTERNAL FRICTION     .....          = ',E10.3/)00089*24
                                                                        00090*24
                                                                        00091*24
 195 CONTINUE                                                           00092*24
                                                                        00093
                                                                        00094
                                                                        00095**9
                                                                        00096**9
     GO  TO  (550,550,550,580) ,KKS                                     00097**3
 580 CONTINUE                                                           00098**3
                                                                        00099*16
                                                                        00100*16
     IF(LC.GT.1)  GO  TO  800                                           00101**3
                                                                        00102*16
                                                                        00103*16
     CALL  YLDLIN(LC,NYP,NSC,ANI,EKJ)                                   00104**3
                                                                        00105*16
                                                                        00106*16
 550 CONTINUE                                                           00107**3
                                                                        00108*10
                                                                        00109*10
                                                                        00110*12
                                                                        00111*12
     ******  FOR  ELASTO PLASTIC  ANALYSIS  READ  MATERIAL  PROPERTIES  00112*12
                                                                        00113*12
                                                                        00114
```

```

      IF(NDYN.EQ.2)  GO  TO  8000
      IF(LOOP.NE.1)  GO  TO  8000


          READ(5,505) ISLC,(NSLC1(K),K=1,NLC)


 505 FORMAT(16I5)
 506 FORMAT(1H1//' INPUT TABLE 1B, MULTIPLE LOADING CASES ' //
    15X,'LOADING',10X,'NUMBER OF SURFACE'/
    25X,'CASE NO.',10X,'LOAD CARDS' /)
 507 FORMAT(5X,I5,19H      .      .       ,I5,I5)

  ISLC = 0 PROPORTIONAL CONCENCRATED LOADS
  ISLC = 1 VARIABLE REPEATED SURFACE TRACTIONS,
           AND PRORORTIONAL CONCENCRATED LOADS
  ISLC = 2 VARIABLE REPEATED SURFACE TRACTIONS,
            VARIABLE REPEATED CONCENCRATED LOADS,
           AND PRORORTIONAL CONCENCRATED LOADS




 800 CONTINUE


      NSLC=NSLC1(LC)
      WRITE(6,506)
      WRITE(6,507) LC,NSLC,ISLC
8000 CONTINUE





 500 CONTINUE


          IF(LOOP.NE.1)  GO TO 50

      LL=NSLC


      IF(LL.EQ.0)  GO  TO  49


      READ(5,45) (SCALB(I),I=1,NSLC)
          IF(NDRN. EQ.0) GO TO 50



  51       CONTINUE




      READ(5,45) (SCALF(I),I=1,NSLC)
  50 CONTINUE
```

```
 45 FORMAT(8E10.3)

    WRITE(6,46)
 46 FORMAT(1H1/'  ***  SCALE FACTOR FOR BULK LOAD *** ')
    WRITE(6,45) (SCALB(I),I=1,NSLC)

    IF(NDRN.EQ.0)  GO  TO  49

    WRITE(6,48)
 48 FORMAT(1H1/'  ** SURFACE FLUID LOAD SCALE FACTOR *** ')
    WRITE(6,45) (SCALF(I),I=1,NSLC)

 49 CONTINUE

     GO  TO  (300,300,400,100,200),NDYN

    *****  NDYN.  EQ  .  1  OR  2.  DYNAMIC  EXTERNAL  LOAD  ANALYSIS
 300 CONTINUE

        CALL   SOLV3 (IH,A1,A3,XM,B,BO,VEL,PDYL,TS,GACL,ID,R,A,AO,
   *GB,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEQH,GCB,A4,A5,A6,MAXA,X0,X1,X2,
   *AT,BT,AC, SO,NWA,NEQB,NBLOCK,MI,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,
   *AN,NSLC1,AN1,EKJ,LC,SURPFX,SURPFY,RE,SW,AS)
    GO  TO  700

******  NDYN  .  EQ.  3  FREQUENCY   ANALYSIS   **************
 400 CONTINUE
    GO  TO  700
****************  NDYN .EQ.  4   STATIC  SOLID  ONLY ****************
 100  CONTINUE

        CALL   SOLV1 (IH,A1,A3,XM,B,BO,VEL,PDYL,TS,GACL,ID,R,A,AO,
   *GB,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEQH,GCB,A4,A5,A6,MAXA,X0,X1,X2,
   *AT,BT,AC, SO,NWA,NEQB,NBLOCK,MI,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,
   *AN,NSLC1,AN1,EKJ,LC,SURPFX,SURPFY,RE,SW,AS)
    GO  TO  700


********  NDYN .EQ . 5   SOLID - FLUID  TRANSIENT  ANALYSIS  ONLY  ****
 200 CONTINUE
```

```
      CALL    SOLV2 (IH,A1,A3,XM,B,BO,VEL,PDYL,TS,GACL,ID,R,A,AO,

     *GB,PMAX,AMAX,NEQ,NUMNP,NACL,NUMEL,NEQH,GCB,A4,A5,A6,MAXA,X0,X1,X2,

     *AT,BT,AC, SO,NWA,NEQB,NBLOCK,M1,X,Y,Q,ISC,JSC,SURTRX,SURTRY,AM,EK,

     *AN,NSLC1,AN1,EKJ,LC,SURPFX,SURPFY,RE,SW,AS)


      GO  TO  700


  700 RETURN
      END
```

```
         DATA SET STDSLD     AT LEVEL 051 AS OF 83/02/11
    SUBROUTINE STDSLD(NP,LK,IH,LH,NUMNP,NEL,FK,DENS,DENF,GMS,         00001*39
                                                                      00002*39
                                                                      00003*39
   *VOL,K,F,EM,IMAT,KS,KF,THICK)                                      00004*26
    IMPLICIT  REAL*8(A-H,O-Z)                                         00005
                                                                      00006*23
                                                                      00007*23
    COMMON/QUAD/RR(4),ZZ(4),FAC,R,Q(6),P(4,12),SF(12,12),H(12,12),    00008*39
   *ST(20,12),D(4,4),C(12,12),E(12,12),TT(5,12),PI(4,8),UDD(4,4)      00009*25
    COMMON/BELA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017   00010*39
                                                                      00011*19
                                                                      00012*48
        COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT            00013*48
     COMMON/ASHK/F1,F2,F3,AL,ALF,FACC,FACT                            00014*25
    COMMON/CONTL/DT,HED(8),KADN,NASIM,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPNAX00015*25
   *,ISTRPR,NPRINT,ISTART,NEQH,MBANDH,NEQS,MBANDS                     00016*51
                                                                      00017*25
    *******   DIMENSION AND DATA FOR ROTATION OF  STRESSES  **********00018*25
                                                                      00019*25
    DIMENSION  IVECT(4),JVECT(4) ,V(4)                                00020*25
    DATA  IVECT/4,2,1,3/,JVECT/1,3,2,4/                               00021*25
                                                                      00022*19
     DIMENSION NP(4),LK(8),LH(8),IH(NUMNP,1)                          00023*39
                                                                      00024*19
                                                                      00025*19
    DIMENSION  SSS(5),TTT(5)                                          00026*19
    DATA  SSS/0.,-1.,1.,0.,0./,TTT/0.,0.,0.,-1.,1./                   00027*19
    DIMENSION GMS(8),FK(1) ,SK(8,8)                                   00028**5
                                                                      00029
    DIMENSION  EM(1)                                                  00030*30
                                                                      00031
                                                                      00032*47
    DIMENSION  SP(12,12)                                              00033*47
                                                                      00034
                                                                      00035
                                                                      00036*19
    ******   CALCULATE  STRESS  DISPLACEMENT  MATRIX  FOR  ELEMENT    00037*19
                                                                      00038*19

       WRITE(6,160)
160    FORMAT(5X,' ** PRINT IN STDSLD **')
       WRITE(6,161) NK015,NK016,NK017
161    FORMAT(5X,3I5)
                                                                      00039*25
    ********   TOTAL  STIFFNESS  OF  ELEMENT  CONSIDERING  UNDRAINED  00040*25
                                                                      00041*47
    DO  455  I=1,12                                                   00042*47
    DO  455  J=1,12                                                   00043*47
455 SP(I,J)=0.0                                                       00044*47
                                                                      00045*25
                                                                      00046*25
                                                                      00047*25
350 CONTINUE                                                          00048*25
                                                                      00049*19
                                                                      00050*23
    ******  TRIANGLE  STIFFNESS  CONDITION  AND  STRESS  OUTPUT  REQUI00051*23
                                                                      00052*23
                                                                      00053*23
    IF( (NP(3).EQ.NP(4)).AND.(NS.EQ.20))  NS= 16                      00054*23
```

```
                                                                        00055*23
      LL=NS/4                                                           00056*19
                                                                        00057*39
                                                                        00058*39
      DO  530  L=1,LL                                                   00059*36
                                                                        00060*36
      CALL FORMB(SSS(L),TIT(L),IPNAX,THICK)                             00061*36
                                                                        00062*19
                                                                        00063*33
                                                                        00064*33
      *******  FLUID  DISPLACEMENT  FUNCTION  ARE CALCULATED  ONLY FOR 00065*33
                                                                        00066*33
      IF(L.NE.1)  GO  TO  600                                           00067*33
                                                                        00068*33
                                                                        00069*33
                                                                        00070*33
      DO  900 I=1,4                                                     00071*19
      DO  900  J=1,8                                                    00072*19
  900 PI(I,J)=P(I,J)                                                    00073*19
                                                                        00074*19
                                                                        00075*19
      *******  CALCULATE  FLUID  STRAIN  -  DISPLACEMENT   *************00076*19
  500 CONTINUE                                                          00077*19
      DO  600  JJ=1,ND                                                  00078*19
      TT(L,JJ)=P(1,JJ)+P(2,JJ)+P(3,JJ)                                  00079*19
  600 CONTINUE                                                          00080*19
                                                                        00081*19
                                                                        00082*19
      *******  CALCULATE  SOLID  STRESS  DISPLACEMENT  MATRIX  *********00083*19
                                                                        00084*19
      DO  530  II=1,4                                                   00085*19
      I=II+4*(L-1)                                                      00086*19
      DO  530  J=1,ND                                                   00087*19
      ST(I,J)=0.0                                                       00088*19
      DO  530  K=1,4                                                    00089*19
                                                                        00090*25
                                                                        00091*47
      ******   DELETE  TEMPORARILY  FOR  STRESS  MATRIX  CONDENSATION  *00092*47
                                                                        00093*47
                                                                        00094*47
      ST(I,J)=ST(I,J)+D(II,K)*P(K,J)                                    00095*25
                                                                        00096*25
  530 CONTINUE                                                          00097*25
      *******   END  OF  STRESS  DISPLACEMENT  MATRIX   ****************00098*19
                                                                        00099*19
                                                                        00100*23
                                                                        00101*23
      *****   CONDITION  FOR  TRIAGULAR  ELEMENT  OT  NON - INCOMPATIBL00102*23
                                                                        00103*23
      IF(NELTYP.EQ.0)  GO  TO  400                                      00104*23
      IF(NP(3).EQ.NP(4))  GO  TO  400                                   00105*23
                                                                        00106*33
                                                                        00107*33
                                                                        00108*33
      ******  CONDENSATION  OF  SF,  H  ,  ST,  TT  FROM  12 *12  TO  8 00109*23
      ********   STATIC  CONDENSATION  OF  INCOMPATIBLE  NODES  ********00110*19
                                                                        00111*19
                                                                        00112*19
      DO  550  NN=1,4                                                   00113*19
      L=12-NN                                                           00114*19
```

```
      K=L+1
      DO  550  I=1,L
      GP=SF(I,K)/SF(K,K)

      IF(NDRN.EQ.0)  GO  TO    560


      CD=H(I,K)/H(K,K)
  560 CONTINUE
      DO  540  J=1,NS
      GG=1.0
  540 ST(J,I)=ST(J,I)-GP*ST(J,K)

      ******    CHECK  CRTICALLY   ***********************
      *****     FLUID  VOLUMETRIC  STRAIN  COMPONENT  CONDENSATION  *****



      DO  550  J=1,L
      SF(I,J)=SF(I,J)-GP*SF(K,J)


      IF(NDRN.EQ.0)  GO  TO  550

      H(I,J)=H(I,J)-CD*H(K,J)
  550 CONTINUE

*******      END  OF  CONDESATION   ************************************



     *******  ROTATE  AT  THIS  POINT  STRESS-DISPLACEMENT  MATRIX  ACCO
     ********     TO   S A P - 4  PROBLEM                  ***************








  400 CONTINUE



      NSET=LL-1
      IF(NSET.LE.0)  GO  TO  730
      DO  720  L=1,NSET
      IV=IVECT(L)
      JV=JVECT(L)
      CALL VECTOR(V,RR(IV),ZZ(IV),0.0D0,RR(JV),ZZ(JV),0.0D0)
      S2=V(1)*V(1)
        C2=V(2)*V(2)
      SC=-V(1)*V(2)
      I1=4*L +1
      I2=I1+1
      I4=I1+3
      DO  710  J=1,8
      B1=ST(I1,J)
```

```
      B2=ST(I2,J)                                                       00175*25
      B4=ST(I4,J)                                                       00176*25
      B5=2.0*SC*B4                                                      00177*25
      ST(I1,J)=C2*B1+S2*B2+B5                                           00178*45
      ST(I2,J)=S2*B1+C2*B2-B5                                           00179*25
  710 ST(I4,J)=SC*(B2-B1)+(C2-S2)*B4                                    00180*25
  720 CONTINUE                                                          00181*25
  730 CONTI NUE                                                         00182*25
                                                                        00183*25
                                                                        00184*25
                                                                        00185*23
                                                                        00186*23
      VLD=VOL/4.                                                        00187*35
      M=0                                                               00188
        DO  690  I=1,4                                                  00189
      IIM=I                                                             00190
      N=NP(IIM)                                                         00191
      DO  690  J1=1,2                                                   00192
      M=M+1                                                             00193
      LK(M)=IH(N,J1)                                                    00194
  690 CONTINUE                                                          00195
                                                                        00196*50
              GO TO (700,800,700,800),KRS                               00197*50
                                                                        00198*50
  800         CONTINUE                                                  00199*50
                                                                        00200*50
      DO  300  I=1,8                                                    00201**6
      GMS(I)=VLD*(DENS-(FK(K)**2.)*DENF)                                00202*35
      DO  300 J=1,8                                                     00203**6
  300 SK(I,J)=SF(I,J)                                                   00204**6
                                                                        00205
                                                                        00206
                                                                        00207
  700 CONTINUE                                                          00208*15
                                                                        00209
      K=IMAT                                                            00210*37
      F=FK(K)                                                           00211**9
                                                                        00212*14
                                                                        00213*14
                                                                        00214*31
                                                                        00215*31
                                                                        00216
      WRITE(51) (LK(I),I=1,8),(LH(I),I=1,8),((SK(I,J),J=1,8),I=1,8),    00217**5
     *((PI(I,J),J=1,8),I=1,4),(GMS(I),I=1,8),VOL,DENS,DENF,F,NP         00218*12
                                                                        00219*13
                                                                        00220*13
*****THIS   COMPLETES   FOR   WRITING   ONE   SET   OF   INFORMATION   FOR   ONE00221
                                                                        00222
  250 CONTINUE                                                          00223*32
                                                                        00224
  850 CONTINUE                                                          00225*19
      RETURN                                                            00226
      END                                                               00227
```

```
        DATA SET STIFF      AT LEVEL 012 AS OF 82/01/05
      SUBROUTINE STIFF(A1,NEQS,MBANDS,AS)                               00001**9
      IMPLICIT  REAL*8(A-H,O-Z)                                         00002
                                                                        00003
                                                                        00004
                                                                        00005
      DIMENSION A1(NEQS,1),AS(NEQS,NEQS)                                00006**9
                                                                        00007**2
                                                                        00008**2
                                                                        00009**2
                                                                        00010
  999 FORMAT(//)                                                        00011**3
  800 FORMAT(5X,8E12.5)                                                 00012**3
      DO  550  I=1,NEQS                                                 00013
      IF(I.NE.1)  GO  TO  530                                           00014
      DO  590  J=1,MBANDS                                               00015**4
  590 AS(I,J)=A1(I,J)                                                   00016**2
      GO  TO  550                                                       00017
  530  JJ=I                                                             00018
      DO 560 J=1,MBANDS                                                 00019**4
      IF(JJ.GT.NEQS)  GO  TO  550                                       00020
      AS(I,JJ)=A1(I,J)                                                  00021**3
  560 JJ=JJ+1                                                           00022
  550 CONTINUE                                                          00023
      DO  5930  I=1,NEQS                                                00024
      DO  5930  J=1,NEQS                                                00025
      AS(J,I)=AS(I,J)                                                   00026
  930 CONTINUE                                                          00027
                                                                        00028
       WRITE(6,5931)
  931 FORMAT(1H1/5X,'   PRINT OF STIFFNESS MATRIX ')
       WRITE(6,5932) ((AS(I,J),J=1,NEQS),I=1,NEQS)
  932 FORMAT(5X,4E10.3)
                                                                        00029
                                                                        00030
                                                                        00031
                                                                        00032
      CALL INSYMA(NEQS,AS)                                              00033**9
                                                                        00034*11
       WRITE(6,5931)
       WRITE(6,5932) ((AS(I,J),J=1,NEQS),I=1,NEQS)
                                                                        00035*11
  500  CONTINUE                                                         00036
                                                                        00037
      RETURN                                                            00038
      END                                                               00039
```

```
      DATA SET STIFF       AT LEVEL 012 AS OF 82/01/05
      SUBROUTINE STIFF1(A1,NEQS,MBANDS,AS)                           00001**9
      IMPLICIT  REAL*8(A-H,O-Z)                                      00002
                                                                     00003
                                                                     00004
                                                                     00005
      DIMENSION A1(NEQS,1),AS(NEQS,NEQS)                             00006**9
                                                                     00007**2
      DO 551 I=1,NEQS
      DO 551 J=1,NEQS
  551 AS(I,J)=0.0

                                                                     00008**2
                                                                     00009**2
                                                                     00010
  999 FORMAT(//)                                                     00011**3
 5800 FORMAT(5X,8E12.5)                                              00012**3
      DO  550  I=1,NEQS                                              00013
      IF(I.NE.1)  GO  TO  530                                        00014
      DO  590  J=1,MBANDS                                            00015**4
  590 AS(I,J)=A1(I,J)                                                00016**2
      GO  TO  550                                                    00017
  530  JJ=I                                                          00018
      DO 560 J=1,MBANDS                                              00019**4
      IF(JJ.GT.NEQS)  GO  TO  550                                    00020
      AS(I,JJ)=A1(I,J)                                               00021**3
  560 JJ=JJ+1                                                        00022
  550 CONTINUE                                                       00023
      DO  5930  I=1,NEQS                                             00024
      DO  5930  J=1,NEQS                                             00025
      AS(J,I)=AS(I,J)                                                00026
 5930 CONTINUE                                                       00027
                                                                     00028
                                                                     00029
                                                                     00030
                                                                     00031
                                                                     00032
                                                                     00034*11
                                                                     00035*11
 8500  CONTINUE                                                      00036
                                                                     00037
      RETURN                                                         00038
      END                                                            00039
```

```
        DATA SET STRAIN      AT LEVEL 015 AS OF 83/02/24
    SUBROUTINE STRAIN(PMAX,NUMEL,NUMMAT,GMX,DPL,GUSE,E,EN,PO)
    IMPLICIT REAL*8(A-H,O-Z)
       COMMON/FOOT/NOEL,NL(20)
       COMMON/SOL/ANORM,NBLOCK,NEQB,LL,NF,IDUM,NEIG,NAD,NVV,NFO
       COMMON/BOBBY/KKS,NUGMAT,NC,LOOP,NDRN,NINT,IDWA,NPT


       DIMENSION EITR(300),GITR(300),POIS(300),DAMP(300)



    DIMENSION PMAX(NUMEL,1),GMX(1),DPL(1),GUSE(1),EN(1),E(1),PO(1)

    DIMENSION LM(8),LH(8),SK(8,8),P(4,8),GMS(8),SF(8,8),EG(8,8),CC(8,8
   *),H(8,8),NP(4)

    DIMENSION XP(16),XN(16)

    REWIND  22
    REWIND  51

        KK=1

       WRITE(6,21) LOOP



        KG=1
    DO  80  M=1,NUMEL

    READ(51) (LM(I),I=1,8),(LH(I),I=1,8),((SK(I,J),J=1,8),I=1,8),
   *((P(I,J),J=1,8),I=1,4),(GMS(I),I=1,8),VOL,DENS,DENF,F,NP


    READ(51)((SF(I,J),J=1,8),I=1,8),((EG(I,J),J=1,8),I=1,8),((CC(I,J),
   *J=1,8),I=1,8),IMAT,KS,KF,ALF,((H(I,J),J=1,8),I=1,8)
   *,(XP(I),I=1,16),(XN(I),I=1,16)

    K=IMAT


        IF(LOOP.EQ.1) GO TO 90

        READ(22) ENEW,GNEW,DNEW,PSN


 90     CONTINUE

 81      CONTINUE

         IF(NAD.EQ.0) GO TO 300

         IF(M.EQ.NL(KG)) GO TO 100

300      CONTINUE



        IF(LOOP.NE.1) GO TO 95
```

```
                                                                      00060**9
                                                                      00061**9
           GNEW=GUSE(K)                                               00062*10
           DNEW=DPL(K)                                                00063*10
     AMAX1    =PMAX(M,8)/GUSE(K)                                      00064**2
                                                                      00065**9
           GO TO 96                                                   00066**9
                                                                      00067**9
                                                                      00068**9
  95        CONTINUE                                                  00069**9
           AMAX1=PMAX(M,8)/GNEW                                       00070**9
                                                                      00071**9
  96        CONTINUE                                                  00072**9
                                                                      00073**9
     AMAX1=100.*AMAX1                                                 00074**2
     AJ=0.65*AMAX1                                                    00075**2
     CALL  CURV52(AJ,GN,DN)                                           00076**2
     GN=GN*GMX(K)                                                     00077**2
     DN=DN/100.                                                       00078**2
                                                                      00079**8
                                                                      00080**8
            GO TO 200                                                 00081**8
 100        CONTINUE                                                  00082**8
           GN=GUSE(K)                                                 00083**8
           DN=DPL(K)                                                  00084**8
           GNEW=GN                                                    00085*10
           DNEW=DN                                                    00086*10
                                                                      00087*10
                                                                      00088**8
                                                                      00089**8
           KG=KG+1                                                    00090**8
                                                                      00091**8
 200       CONTINUE                                                   00092**8
                                                                      00093**9
             DE=100.*(GNEW-GN)/GN                                     00094**9
             DZ=100.*(DNEW-DN)/DN                                     00095**9
             EITR(M)=GN*2.*(1.+PU(K))                                 00096**9
                                                                      00097**8
                                                                      00098**8
                                                                      00099*10
         WRITE(6,20) M,GNEW,GN,DNEW,DN,DE,DZ                          00100*13
                                                                      00101*10
                                                                      00102*10
                                                                      00103*10
                                                                      00104**9
              GITR(M)=GN                                              00105**9
              DAMP(M)=DN                                              00106**9
              POIS(M)=PU(K)                                           00107**9
                                                                      00108**9
                                                                      00109**7
           KK=K+1                                                     00110**7
                                                                      00111**7
  80 CONTINUE                                                         00112**2
          REWIND 22                                                   00113*12
                                                                      00114*12
                                                                      00115**2
         DO 99 M=1,NUMEL                                              00116*12
              WRITE(22) EITR(M),GITR(M),DAMP(M),POIS(M)               00117*10
  99          CONTINUE                                                00118*10
                                                                      00119**4
```

```
                                                                     00120**4
                                                                     00121**4
 65 FORMAT(//'  ELEMENT NO., USED SHEAR MODULUS, NEWLY CALCULATED SHEA00122
   1R MODULUS, PERCENTAGE DIFFERENCE  '//,2X,I5,6X,E17.9,6X,E17.9,10X,00123
   2E15.9//)                                                         00124
666 FORMAT(//'  ELEMENT NO., USED DAMPING VALUE,NEWLY CALCULATED VALUE00125
   1, PERCENTAGE DIFFERENCE  '//,2X,I5,6X,E17.6,6X,E17.6,10X, E15.9//)00126
 70 FORMAT('  TEMPERATURE          E(N)          E(S)          E(T)     NU(NS00127
   1)    NU(ST)    NU(ST)        G(NS)    '/10X, 4E12.5,3F10.4,E12.5//)  00128
                                                                     00129*13
 20      FORMAT(5X,I5,6E12.5)                                        00130*13
                                                                     00131*13
 21       FORMAT(1H1/ 40X,'********** ITERATION  NO.          = ',I5//  00132*14
   *                 40X,'             STRAIN COMP.SOIL PROPERTIES   '// 00133*14
   *5X,'ELEMT.',4X,'GUSE',4X,'GNEW',4X,'DUSE',4X,'DNEW',4X,'G-DIFF',4X00134*14
   *,'D-DIFF'/                                                       00135*14
   *5X,' NO. ',35X,'(PERCENT)',3X,'(PERCENT)'/)                       00136*14
    RETURN                                                           00137
    END                                                              00138
```

```
          DATA SET STRCOM      AT LEVEL 008 AS OF 83/02/18
      SUBROUTINE STRCOM(NUMMAT,GMX,DPL,GUSE,E,EN,PO)                    00001
      IMPLICIT REAL*8(A-H,O-Z)                                          00002
                                                                        00003**6
         COMMON/FOOT/NOEL,NL(20)                                        00004**6
         COMMON/SOL/ANORM,NBLOCK,NEQB,LL,NF,IDUM,NEIG,NAD,NVV,NFU       00005**6
                                                                        00006**6
      DIMENSION GMX(1),DPL(1),GUSE(1),E(1),EN(1),PO(1)                  00007
      WRITE(6,101)
  101 FORMAT(1H1/5X,' PRINT IN STRCOM '  )
      REWIND  22                                                        00008
      DO  30  N=1,NUMMAT                                                00009
      READ(5,100) GMX(N),DPL(N),GUSE(N),PO(N)                           00010**2
      WRITE(6,102)GMX(N),DPL(N),GUSE(N),PO(N)
  102 FORMAT(5X,4E12.5)
  100 FORMAT(5X,4E12.5)                                                 00011
   30 CONTINUE                                                          00012
                                                                        00013**6
          IF(NAD.EQ.0)  GO TO 70                                        00014**6
                                                                        00015**6
           NAD=0  NO CONCRETE FOOTING                                   00016**6
                                                                        00017**6
         *********     READ HERE  CONCRETE FOOTING ELEMENT  ****        00018**6
                                                                        00019**6
          READ(5,60) NOEL,(NL(I),I=1,NOEL)                              00020**6
   60     FORMAT(16I5)                                                  00021**6
          WRITE(6,65) NOEL,(NL(I),I=1,NOEL)                             00022**6
   65     FORMAT(1H1/5X,'      NO. OF FOOTING ELEMENT  = ',I5/          00023**6
     *                   5X,'      FOOTING ELEMENT         = ', 6I5/)   00024**6
                                                                        00025**6
   70      CONTINUE                                                     00026**6
      DO  40  N=1,NUMMAT                                                00027
          E(N)=GUSE(N)*(2.*(1.+PO(N)))                                  00028**8
                                                                        00029**8
      EN(N)=E(N)                                                        00030
   40 CONTINUE                                                          00031
   50 CONTINUE                                                          00032
      RETURN                                                            00033
      END                                                               00034
```

```
        DATA SET STRINV     AT LEVEL 010 AS OF 83/01/15
       SUBROUTINE STRINV(SIG,EPS,BLK,EMU,YOUNG,POIS,G,KKS,IPT,PF)

      IMPLICIT REAL*8 (A-H,O-Z)

      DIMENSION SIG(4),EPS(4)

         DIMENSION PF(5)
      DIMENSION G(4,4)
      *********   CALCULATE  ELASTIC  STRAIN  BASED  ON  CURRENT  DISPLAC



  100 CONTINUE
      DO  300 I=1,4
      DO  300 J=1,4
  300 SIG(I)=SIG(I)+G(I,J)*EPS(J)

        DO 200 I=1,3
  200 SIG(I)=SIG(I)+PF(IPT)


      RETURN
      END
```

```
       DATA SET STRSPR     AT LEVEL 134 AS OF 83/03/05
   SUBROUTINE STRSPR(B,TIME,PMAX,NUMEL,NNN,AM,EK,AN,AN1,EKJ,LC,ITR,00001*94
 *RE)                                                               00002*94
                                                                    00003*94
                                                                    00004*33
                                                                    00005*33
  IMPLICIT  REAL*8   (A-H,O-Z)                                      00006
                                                                    00007
                                                                    00008108
    COMMON  D(1)                                                     00009111
                                                                    00010108
  COMMON/CONTL/DD,HED(8),KADN,NUMNP,NUMSL,NEQ,MBAND,NCYCL,NDYN,IPLNA00011
 *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH                                00012
 *,NEQS,MBANDS                                                      00013118
                                                                    00014*71
                                                                    00015*71
  DIMENSION ST(20,12),LM(16),SIG(9)                                 00016*76
 *,PS(16,8)                                                         00017*56
 *,FACN(4)                                                          00018*71
                                                                    00019
  DIMENSION G(4,4)                                                  00020*82
                                                                    00021
                                                                    00022
  COMMON/BELA/MS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017   00023*32
  COMMON/BOBBY/KKS,NUMMAT,MC,LOOP,NDRN,NINT,IDWA,NPT                00024*56
       COMMON/MATMOD/STRESS(9),STRAIN(4),C(4,4),IPT,NEL             00025108
                                                                    00026108
                                                                    00027107
      COMMON/SORIN/T1(5,12),XP(16),INDNL,ITYP2D                     00028104
  COMMON/INTEGE/TETA,BETA,GAMA,RTOL,NGAMA,NPUNCH,NEQUIB,ITEMX       00029*77
                                                                    00030*32
  *******    BRING  NBC  HERE  VIA  COMMON   ***********************00031*44
                                                                    00032*32
  COMMON/EL/IND,ICOUNT                                              00033*35
                                                                    00034*32
     COMMON/GORA/IFLAG,IGR                                          00035115
    COMMON/TAPES/NQQ(6),NUMBER                                      00036130
                                                                    00037130
                                                                    00038115
   *******   TAKE IT TO ELASTO PLASTIC ROUTINE TO UNDRAINED ****    00039*96
     COMMON/POBA/ALFA,EM,EPSV,SHFD                                  00040134
                                                                    00041134
                                                                    00042*96
   DIMENSION TO STORE SRESS RESPONSE FOR ELASTIC LIMIT LOAD         00043*96
                                                                    00044*96
                                                                    00045*96
     COMMON/BENOY/SG(5,200,4,4),NCOND                               00046117
                                                                    00047*96
                                                                    00048*96
                                                                    00049*32
                                                                    00050*32
  *********   PASS  THIS  VARIABLE  FROM  ELSTIF  IN  DYMAMIC  ALLOC00051*32
                                                                    00052*32
                                                                    00053*32
                                                                    00054108
  DIMENSION PE(5),P(4,8),ST(20,12),SIG(9),PS(16,8),FACN(4),G(4,4),  00055*30
 *LM(16)
  DIMENSION B(1),PMAX(NUMEL,1)                                      00056
                                                                    00057
```

```
      DIMENSION KP(1),GSTRES(5)                                        00058*94
                                                                       00059*94
   ******************************  COMMON  STATEMENT   FOR  SHAK00060
   *************    ANALYSIS  INTRODUCED  AT  THIS  STAGE  ************00061
                                                                       00062
                                                                       00063
        COMMON/PRIN/ISTRES,IPRINT,JPRINT,KPRINT,IP                     00064104
   *********   AND  INFORMATION   ABOUT  PLOTTING   ******************00065*34
                                                                       00066*34
      COMMON/KALI/JFI,KFL,JPLOT(200),KEBPT(200)                        00067*94
                                                                       00068*94
                                                                       00069*94
                                                                       00070*34
                                                                       00071
   COMMON/ASIF/NYP,NSC,NSLC,NLC,NNYP,NNSC                              00072
                                                                       00073*90
                                                                       00074*90
       COMMON/GRAND/COH(20),PHI(20),ITYPE(20)                          00075124
      COMMON/PEPPE/AREA,WIDTH,HOR,VERT,ACCG                            00076118
      DIMENSION AMJ(14),ANIJ(3),AMAX(14,100,4),AM(1),EK(1)             00077130
                                                                       00078130
   DIMENSION AN(NNSC,1),EKJ(1),ANI(NYP,1)                              00079*79
   DIMENSION KM(5),ZM(5)                                               00080*79
                                                                       00081
                                                                       00082*69
                                                                       00083*69
   DIMENSION STATE(1),IPP(5)                                           00084*69
                                                                       00085*69
                                                                       00086*69
                                                                       00087*69
   DATA IPP/3HU-D,3HL-I,3HU-R,3HI-J,3HR-L/,STATE/1HE/                  00088*69
                                                                       00089*69
                                                                       00090*69
    WRITE(6,164)
164 FORMAT(5X,'    PRINT  IN STFSPR  AT  BEGINING   ')

    WRITE(6,163) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDV,NK015,NK016,NK017
    WRITE(6,163) KKS,LC

    WP=0.0

                                                                       00091
      IF(KKS.EQ.3.AND.IDD.EQ.-1) GO TO 86                              00092*94
                                                                       00093*94
                                                                       00094
   NES=NUMEL*NSC                                                       00095**6
   NEY=NUMEL*NYP                                                       00096**6
   NDT=1
   IF(KKS.EQ.4.AND.NCOND.EQ.1) NDT=4

                                                                       00097*79
   NES=NES*NDT                                                         00098*79
   NEY=NEY*NDT                                                         00099*79
                                                                       00100*79
   REWIND  11                                                          00101
                                                                       00102*30
                                                                       00103*30
                                                                       00104*30
   GO  TO  (3,3,3,4),KKS                                               00105**2
 4 CONTINUE                                                            00106**3
```

```
                                                                        00107*44
                                                                        00108*44
      ******   FOR  DYNAMIC  LOADING  STRESS  VECTOR  MAXIMIZATION   ***00109*44
                                                                        00110*44
                                                                        00111*44
     IF(LC.GT.1.OR.NC.GT.1)  GO  TO  3                                  00112*46
                                                                        00113*17
     CALL YLDIN(NSC,NYP,NUMEL,NES,NEY,AM,EN,AN,AMAX)                    00114**2
                                                                        00115*44
                                                                        00116*44
                                                                        00117
   3 CONTINUE                                                           00118
     RAD2=0.5*RADN                                                      00119
     PI2=2.0 *DATAN(1.0D0)                                              00120
                                                                        00121*39
                                                                        00122*69
     IFEL=1                                                             00123*69
                                                                        00124104
                                                                        00125127
        NC=NNN                                                          00126127
                                                                        00127127
                                                                        00128127
                                                                        00129104
        IF(KKS.EQ.3.AND.IFLAG.EQ.0) GO TO 87                            00130114
                                                                        00131134
         IF(NC.NE.0.AND.ISTRES.NE.ISTRPR) GO TO 87                      00132134
                                                                        00133134
        IF(NPUNCH.NE.0) GO TO 2097
                                                                        00134104
                                                                        00135104
     WRITE(6,2020) TIME                                                 00136*69
     WRITE(6,2015) NC,ITR                                               00137*76
                                                                        00138*39
2097 CONTINUE
     IF(KKS.EQ.2.AND.LOOP.NE.NUMBER) GO TO 87


     WRITE(57,2099) TIME
     WRITE(16,2099) TIME
2099 FORMAT(1X,' TIME STEP FOR STRESS OUTPUT   =',F10.4)
                                                                        00139114
  87    IND=NNN                                                         00140114
  86 CONTINUE                                                           00141*39
                                                                        00142113
                                                                        00143114
                                                                        00144114
                                                                        00145113
     IK=0                                                               00146*13
       IP=1                                                             00147*95
     REWIND 10                                                          00148*19
                                                                        00149*82
     REWIND   50                                                        00150*82
                                                                        00151*82
        REWIND 17                                                       00152*98
        REWIND 22                                                       00153128
                                                                        00154128
     KK=1                                                               00155*19
                                                                        00156*13
                                                                        00157*13
                                                                        00158*27
```

```
                                                                        00159126
                                                                        00160126
                                                                        00161126
                                                                        00162126
                                                                        00163*27
                                                                        00164*13
      *********  TAPE  11  PASSED  FRPM  ELSTIF  ROUTINE                00165*13
      ************TAPE  13  PASSED  FROM  ELSTIF  ROUTINE               00166*13
      **********  TAPE  12  CREATED  IN  THIS  ROURINE                  00167*13
      ********* TAPE  17 CONTAINS INFORMATION ON DRAINED STRESS-STRAIN M00168*98
      MATRIX                                                            00169*98
                                                                        00170*13
                                                                        00171*13
      DO  300  NEL=1,NUMEL                                              00172
                                                                        00173*30
                                                                        00174113
                                                                        00175113
                                                                        00176*30
          IF(KKS.EQ.3.AND.IND.EQ.-1) GO TO 75                           00177*94
                                                                        00178*94
                                                                        00179*30
      ********   READ  ELEMENT  INFORMATION  AS  CREATED  IN  ELSTIF  **00180*30
                                                                        00181*30
                                                                        00182*30
      READ(11) ST,TT,P ,LM,ALFA, EM,RM,ZM,KS,KF,IMAT,BLK,EMU,YOUNG,     00183*31
     *POIS,VOL,PS                                                       00184*56
     *,FACN                                                             00185*71
                                                                        00186*30
           SUMD=EMU                                                     00187134
                                                                        00188134
        WRITE(6,163)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
                                                                        00189*49
                                                                        00190128
          IF(KKS.NE.2) GO TO 345                                        00191128
                                                                        00192128
          IF(LOOP.EQ.1) GO TO 345                                       00193128
                                                                        00194128
           IF(NC.GT.0) GO TO 344                                        00195128
                                                                        00196128
           READ(22) ENEW,GNEW,DAMP,POIS                                 00197128
                                                                        00198128
                                                                        00199128
           CALL MDCNEW(ENEW,GNEW,POIS,G)                                00200128
                                                                        00201128
           GO TO 340                                                    00202128
                                                                        00203128
344        CONTINUE                                                     00204128
                                                                        00205128
                                                                        00206129
         IF(NEL.NE.1) GO TO 345                                         00207129
                                                                        00208129
                                                                        00209128
           REWIND 17                                                    00210128
                                                                        00211128
                                                                        00212128
345        CONTINUE                                                     00213128
                                                                        00214128
                                                                        00215128
                                                                        00216*86
                                                                        00217*86
```

```
                                                                        00218*87
       READ(17) ((G(I,J),J=1,4),I=1,4)                                  00219*98
                                                                        00220*86
                                                                        00221*86
                                                                        00222*86
340 CONTINUE                                                            00223*87
    *******  PUT   A   LOOP  FOR  CALCULATION  OF  STRESS  OUTPUT  *****00224*30
                                                                        00225*30
    PRINT  999                                                          00226*47
                                                                        00227*30
        DO 301 I=1,16                                                   00228*99
301     XP(I)=0.0                                                       00229*99
                                                                        00230*56
                                                                        00231*56
                                                                        00232*56
                                                                        00233*56
        LL=NPT                                                          00234101

       IF(KKS.EQ.4.AND.NCOND.NE.1) LL=NS/4
                                                                        00235*56
    DO  95  L=1,LL                                                      00236*30
                                                                        00237*30
    FYLD=0.0                                                            00238*90
                                                                        00239*30
                                                                        00240*30
    PF(L)=0.0                                                           00241*30
                                                                        00242*56
    IPT=0                                                               00243*56
                                                                        00244*64
                                                                        00245*64
     WRITE(6,163)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                        00246*64
                                                                        00247*66
                                                                        00248*66
    DO  30  I=1,4                                                       00249
    SIG(I)=0.0                                                          00250*19
 30 STRAIN(I)=0.0                                                       00251*32
    BB0=0.0                                                             00252*10
    BBT=0.                                                              00253
    M=0                                                                 00254
    BB1=0.0                                                             00255*19
    BB2=0.0                                                             00256*19
    BB3=0.0                                                             00257*19
    BB4=0.0                                                             00258*19
         SW=0.0                                                         00259117
         SP=0.0                                                         00260117
      EPSV=0.0                                                          00261*95
    EPSVF=0.0                                                           00262*33
                                                                        00263*19
                                                                        00264*19
*****       COMPUTATION  OF  SOLID  STRAIN  CALCULATION                 00265*26
                                                                        00266*26
                                                                        00267*26
    DO  60  I=1,4                                                       00268
    MM=I+1                                                              00269
    MN=MM-1                                                             00270
                                                                        00271123
         DO 41 K=MN,MM                                                  00272123
                                                                        00273121
```

```
      M=M+1                                                             00274
      ******************************************************************00275
      IF(KS.EQ.0.AND.KF.EQ.0)  GO  TO  32                               00276
      IF(KS.EQ.1.AND.KF.EQ.-1) GO  TO  32                               00277
      IF(KS.EQ.-1.AND.KF.EQ.1)  GO  TO  41                              00278
      ******************************************************************00279
   32 N=LM(M)                                                           00280
      IF(N) 41,41,35                                                    00281
   35 CONTINUE                                                          00282*30
                                                                        00283*19
                                                                        00284*19
                                                                        00285*30
      *****  ELASTIC  EFFECTIVE  STRESS  CALCULATION  ******************00286*30
                                                                        00287*35
      *******    IF(NDRN.NE.0)  ...  ST(1,J)   EFFECTIVE STRESS DISPL.  00288*39
                                                                        00289*39
     *********      IF(NDRN.EQ.0)      ST(1,J) ...  TOTAL STRESS DISPLACE00290*35
                                                                        00291*35
                                                                        00292*30
                                                                        00293*30
      DO  40  JJ=1,4                                                    00294*30
      J=JJ+4*(L-1)                                                      00295*30
                                                                        00296*56
           GO TO (55,55,55,56),KKS                                      00297*94
                                                                        00298*94
   56  CONTINUE
       IF(NCOND.EQ.1) GO TO 55
                                                                        00299*79
      SIG(JJ)=SIG(JJ)+ST(J,K)*B(N)                                      00300*56
      GO  TO  40                                                        00301*56
   55 STRAIN(JJ)=STRAIN(JJ)+PS(J,K)*B(N)                                00302*56
   40 CONTINUE                                                          00303*56
           EPSV=EPSV+TT(1,K)*B(N)                                       00304*94
                                                                        00305*19
                                                                        00306*19
   41 CONTINUE                                                          00307
                                                                        00308*19
                                                                        00309*19
                                                                        00310*19
      *****  CALCULATION  OF  FLUID  VOMETRIC  STRAIN  *****************00311*19
                                                                        00312*33
                                                                        00313*33
                                                                        00314*33
      DO  45  K=MN,MM                                                   00315
      M=M+1                                                             00316
      N=LM(M)                                                           00317
      IF(N)  45,45,42                                                   00318
                                                                        00319*38
                                                                        00320*38
      *******   FLUID  VOLUMETRIC  STRAINS  ARE CALCULATED  W.R.T. ONLY 00321*38
      ********   USE  ONLY TT(1,J),J=1,ND   ****************            00322*38
                                                                        00323*38
   42 EPSVF=EPSVF+TT(1,K)*B(N)                                          00324*38
                                                                        00325*19
                                                                        00326*19
                                                                        00327*26
   45 CONTINUE                                                          00328
   60 CONTINUE                                                          00329
                                                                        00330*56
                                                                        00331*56
```

```
      PF(L)=PO+(ALFA*PFSV+PFSVF)                                         00332*95
IF(KKS.EQ.4.AND.NCOND.NE.1) GO TO 57


       WRITE(6,161)
 161   FORMAT(/' ** PRINT IN STRSPR ** ')
       WRITE(6,162) (STRAIN(I),I=1,4),PF(L)
       WRITE(6,163) NC,ITR,IFLAG,NEL,L
       WRITE(6,162) ((G(I,J),J=1,4),I=1,4)
                                                                         00333*95
                                                                         00334132
       WRITE(6,163) NS,NELTYP,ND,NUMGI,MODEL,NCON,IDW,NK015,NK016,NK017

       WRITE(6,163) KKS,LC,NEL,L
                                                                         00335132
                                                                         00336*56
     IF(KKS.EQ.3)  GO  TO  88                                            00337*57
     ******  CALCULATE SOLID STRAIN INVERTING STRESSES  ****************00338*39
                                                                         00339*39
                                                                         00340*39
      CALL STRINV(SIG,STRAIN,BLK,EMU,YOUNG,PO,G,KKS,IPT,PF)              00341*95
                                                                         00342*49
                                                                         00343*49

      GO TO 59
                                                                         00344*39
                                                                         00345*39
                                                                         00346*30
                                                                         00347*56
  88 CONTINUE                                                            00348*57
     ********  CALCULATE  FLUID  PRESSURE  ****************************00349*30
                                                                         00350*30
                                                                         00351*30
      WRITE(6,163) NS,NELTYP,ND,NUMGI,MODEL,NCON,IDW,NK015,NK016,NK017
                                                                         00352*94
  57  CONTINUE

      DO 58 I=1,3
  58  SIG(I)=SIG(I)+PF(L)
                                                                         00353*94
  59  CONTINUE

                                                                         00354*95
                                                                         00355*93
          IF(NETYP.NE.4) GO TO 65                                        00356102
                                                                         00357102
                                                                         00358*96
       DO 66 I=1,4                                                       00359*96
       SG(NC,NEL,L,I)=SIG(I)                                             00360*96
  66   CONTINUE                                                          00361*96
       ********   COMPLETION OF STRESS RESPONSE FOR ELATIC LIMIT LOAD 00362*96
                                                                         00363*96
                                                                         00364*96
                                                                         00365*19
                                                                         00366*19
                                                                         00367*27
                                                                         00368*27
```

```
 65 CONTINUE                                                            00369*27
    BB0=EPSV                                                            00370*51
    BBT=EPSVF                                                           00371*52
                                                                        00372*19
    IF(KKS.NE.3)  GO  TO  160                                           00373*19
                                                                        00374*19
                                                                        00375*19
     *********  ELASTO PLASTIC  ANALYSIS  TO BE PERFORMED  ***********00376*30
                                                                        00377*32
    IF(IPLNAX.EQ.-1) ITYP2D=IPLNAX+3                                    00378*32
    IF(IPLNAX.EQ.0)  ITYP2D=  IPLNAX+1                                  00379*32
    IF(IPLNAX.EQ.1)  ITYP2D=IPLNAX-1                                    00380*32
                                                                        00381*32
      ITYP2D=2 ; PLANE STRESS                                           00382*95
      ITYP2D=1 ; PLANE STRAIN                                           00383*95
      ITYP2D=0 ; AXISYMMETRIC                                           00384*95

     WRITE(6,163) NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                        00385*95
     *********    MODIFY  ALSO  STRAIN  VECTOR  TO  SUIT  NONSAP  TOUT00386*32
                                                                        00387*33
    GSTN=STRAIN(3)                                                      00388*33
    FSTN=STRAIN(4)                                                      00389*33
    STRAIN(3)=FSTN                                                      00390*33
    STRAIN(4)=GSTN                                                      00391*33
                                                                        00392*33
    *******  NOW  THE  STRAIN  VECTOR  ARE  ACCORDING  TO  NONSAP    *00393*33
                                                                        00394*33
                                                                        00395*32
                                                                        00396*32

     WRITE(6,163)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                        00397*30
                                                                        00398*33
                                                                        00399*33
    *****  CHECK   PASSING  ARGUMENT  CRITICALLY   ******************00400*33
    *****  SPECIALLY ASSIGNED NK016,NK017                               00401*63
                                                                        00402*63
                                                                        00403*35
                                                                        00404*35
     NK016=164
     NK017=166
                                                                        00405124
                                                                        00406124
                                                                        00407124
 66    FORMAT(5X,'               TIME  STEP (NC)         = ',I5/        00408124
     *          5X,'               ITERATION NO.(ITR)      = ',I5/        00409124
     *          5X,'               STRESS PRINT(ISTRPR)    = ',I5/        00410124
     *          5X,'               DISPL. PRINT(NPRINT)    = ',I5/        00411124
     *          5X,'               CONTOUR PRINT(ISTART)   = ',I5/        00412124
     *          5X,'               COUNTER FOR STRESS      = ',I5/        00413124
     *          5X,'               COUNTER FOR DISPL.      = ',I5/        00414124
     *          5X,'               COUNTER FOR CONTOUR     = ',I5/        00415124
     *          5X,'               NODE OR ELEMT PRINT(IP)= ',I5/         00416124
     *          5X,'               CONVERGANCE FLAG(IFLAG)= ',I5/)        00417124
                                                                        00418124
 75 CONTINUE                                                            00419*35
                                                                        00420106
```

```
      WRITE(6,163)IND,NNN,NC,KKS,NFL,IPT
      WRITE(6,163)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017
      WRITE(6,162) (D(I),I=NK015,NK016-1)
      WRITE(6,162) (D(I),I=NK016,NK017-1)
                                                                        00421*71
         CALL STSTN(IDW,PF,PS,FACN,LM,PMAX,                             00422108
   * NUMEL)                                                             00425104
                                                                        00423108
                                                                        00424110
                                                                        00426104
     WRITE(6,163)NS,NELTYP,ND,NUMGT,MODEL,NCON,IDW,NK015,NK016,NK017

                                                                        00427*53
                                                                        00428*32
                                                                        00429*32
                                                                        00430*30
                                                                        00431*30
                                                                        00432*94
       GO TO 95                                                         00433*94
                                                                        00434*30
    ****  EXCLUDING  ELASTO PLASTIC ANALYSIS  FOLLOW  THESE  STEPS    *00435*30
                                                                        00436*30
                                                                        00437*19
                                                                        00438*19
160 CONTINUE                                                            00439*19
                                                                        00440*19
                                                                        00441*19
     *************   PRINCIPAL  STRESS  COMPUTATION  *****************00442*19
                                                                        00443*19
                                                                        00444*19
    CC=(SIG(1)+SIG(2))/2.0                                              00445
    BB=(SIG(1)-SIG(2))/2.0                                              00446
    CR=DSQRT(BB**2+SIG(4)**2)                                           00447
    SIG(5)=CC+CR                                                        00448
    SIG(6) =CC-CR                                                       00449
    SIG(7)=(SIG(5)+SIG(6))/2.                                           00450**7
    SIG(8)=(SIG(5)-SIG(6))/2.                                           00451**8
                                                                        00452118
                                                                        00453118
    IF(SIG(4).GT.10E-15.OR.BB.GT.10E-15)  GO  TO  80                    00454
    SIG(9)=100.0                                                        00455**7
    GO  TO  90                                                          00456
 80 SIG(9)=RAD2*DATAN2(SIG(4),BB)                                       00457**7
 90 CONTINUE                                                            00458
                                                                        00459
                                                                        00460
                                                                        00461*32
       COMPUTATION OF J1  &  J2  FOR EACH TIME & FOR EACH SELECTED EL00462118
                                                                        00463118
       SM=(SIG(1)+SIG(2)+SIG(3))/3.                                     00464118
       SX=SIG(1)-SM                                                     00465118
       SY=SIG(2)-SM                                                     00466118
       SZ=SIG(3)-SM                                                     00467118
       SXY=SIG(4)                                                       00468118
       SD=DSQRT(0.5*(SX*SX+SY*SY+SZ*SZ)+SXY*SXY)                        00469132
                                                                        00470131
       IF(STRAIN(4).EQ.0.0) GO TO 14                                    00471131
                                                                        00472131
        SHM=(SIG(4)/STRAIN(4))                                          00473134
                                                                        00474134
```

```
                                                                        00475118
  14      CONTINUE                                                      00476131
                                                                        00477118
             ************  COMPUTE HERE STRESS COMPONENT  P  &  Q  **** 00478117
                                                                        00479117
                                                                        00480*32
      GO TO (12,12,12,13),KKS                                           00481**2
                                                                        00482*33
                                                                        00483*33
  13  CONTINUE                                                          00484**2
                                                                        00485*80
C                                                                       00486*80
      IF(L.GT.1)  GO  TO  800                                           00487*81
                                                                        00488*80
      KG=IMAT                                                           00489*19
       WRITE(6,799) NUMMAT,IMAT,KG,KK
 799  FORMAT(5X,4I5)
      IF(KK.NE.KG)  GO  TO  800                                         00490*19
      READ(10) ((AN1(J,I),I=1,NSC),J=1,NYP),(EKJ(J),J=1,NYP)            00491*19
  800 CONTINUE                                                          00492*19
                                                                        00493*81
                                                                        00494*81
                                                                        00495*81
      CALL YLDMX(NEL,NYP,NSC,AN1,AMU,BC,NBC,AM,EK,AN,NES,AMAX,EKJ,SIG,IK00496*13
     *,NEY,IPT,VOL)                                                     00497*79
                                                                   )    00498*33
                                                                        00499*33
  12  CONTINUE                                                          00500**3
                                                                        00501*30

                                                                        00592*95
        PG=PF(IPT)                                                      00593*95
                                                                        00590*96
                                                                        00591*95
        IF(KKS.EQ.4.AND.NCOND.NE.1) GO TO 52
                                                                        00589*96
          CALL ITRSHD(NEL,L,SIG,FACN,PS,PT,PG,LM,XP)                    00594100
                                                                        00595100
  52      CONTINUE                                                      
                                                                        00502*30
                                                                        00503*95
                                                                        00504*95
      *********  BRANCHING  HERE  FOR  STRESS  OUTPUT  FOR  DIFFERENT   00505*30
                                                                        00506*30
                                                                        00507*90
                                                                        00508*90
                                                                        00509*30
                                                                        00510*69
                                                                        00511*69
                                                                        00512*69
                                                                        00513*69
                                                                        00514*94
                                                                        00515*94
                                                                        00516*94
                                                                        00517*94
                                                                        00518*94
      IF(NC)808,805,808                                                 00519*69
                                                                        00520*97
                                                                        00521*97
  808 IF(ISTRES.NE.ISTREP)  GO  TO  810                                 00522*69
```

```
                                                        00523*84
                                                        00524*84
805 CONTINUE                                            00525*95
                                                        00526*97
                                                        00527*95
                                                        00528*95
    IF(L.NE.1) GO TO 820                                00529*95
                                                        00530*95
    IF(NPUNCH.NE.0) GO TO 820
                                                        00531*84
                                                        00532*84
    WRITE(6,2003)                                       00533*84
    WRITE(6,2004) NEL                                   00534*69
                                                        00535*69
                                                        00536*84
                                                        00537*84
820 CONTINUE                                            00538*84

     IF(NCOND.EQ.1) GO TO 821

     RM(L)=RM(L+NPT)
     ZM(L)=ZM(L+NPT)
821  CONTINUE

     IF(NPUNCH.EQ.0)
   *WRITE(6,2007) L,RM(L),ZM(L),SIG,PF(IPT),FYLD        00539*90
       IF(KKS.NE.2) GO TO 811                           00540118
      IF(LOOP.NE.NUMBER) GO TO 810                      00541130
811    CONTINUE                                         00542118
                                                        00543118
                                                        00544128
       IF(NPUNCH.EQ.0)  GO TO 810                       00545128
                                                        00546128
                                                        00547*90
                                                        00548*97
                                                        00549128
                                                        00550128
        IF(IP.GT.KEL) GO TO 810                         00551128
                                                        00552128
                                                        00553128
        LN=KELMT(IP)                                    00554*97
        IF(LN.NE.NEL) GO TO 812                         00555132
                                                        00556*97
        IF(L.NE.1) GO TO 814                            00557*97
                                                        00558102
                                                        00559102
                                                        00560128
                                                        00561*97
                                                        00562*93
                                                        00563129
                                                        00564129
                                                        00565129
```

```
                                                                       00566131
          IF(STRAIN(4).EQ.0.0) SD=0.0                                  00567131
                                                                       00568131
                                                                       00569118
     WRITE(57,203)NEL,(SIG(I),I=1,2),SIG(4),(STRAIN(I),I=1,2),STRAIN(4)00570119
    *,SIG(5),SIG(6),SIG(8),SM,SD,PP(IPT),SNO,EFSV,FYLD                              00571119
 203   FORMAT(I3,7E11.4/6F10.3/)
                                                                       00572119
                                                                       00573119
 814     CONTINUE
                                                                       00574119
                                                                       00575118
                                                                       00576*97
         IP=IP+1                                                       00577102
                                                                       00578*97
            WRITE OTHER INFORMATION ON TAPE FOR PLOTTING               00579*97
                                                                       00580*97
             ALSO CALCULATE THE MAXIMUM STRESS STRAIN AND PORE PRESSURE 00581*97
                                                                       00582*97
                                                                       00583*93
                                                                       00584*93
 812     CONTINUE
                                                                       00585*70
                                                                       00587*95
                                                                       00588*96
 810      CONTINUE
                                                                       00596*99
                                                                       00597124
           SIG(3)=SM                                                   00598118
           SIG(7)=SD                                                   00599118
                                                                       00600102
         *********   COMPUTE MAXIMA OF STRESSES AT EACH TIME STEP ***** 00601102
                                                                       00602102
          DO 180 J=1,8                                                 00603102
         BB=DABS(SIG(J))                                               00604103
          IF(BB-PMAX(NEL,J)) 180,100,100                               00605102
 100      PMAX(NEL,J)=BB                                               00606102
          PMAX(NEL,J+9)=TIME                                           00607102
 180      CONTINUE                                                     00608102
          *********  REDEFINE PORE PRESSURE ********                   00609102
                                                                       00610102
                                                                       00611102
        PP=DABS(PG)                                                    00612130
        IF(PP-PMAX(NEL,9)) 850,200,200                                 00613130
 200      PMAX(NEL,9)=PP                                               00614102
          PMAX(NEL,18)=TIME                                            00615102
                                                                       00616102
                                                                       00617130
 850    CONTINUE                                                       00618130
                                                                       00619130
                IF(IKS.NE.2) GO TO 813                                 00620124
        IF(LOOP.NE.NUMBER) GO TO 95                                    00621130
                                                                       00622130
 813            CONTINUE                                               00623124
                                                                       00624128
             IF(KC) 806,809,806                                        00625128
                                                                       00626128
                                                                       00627128
```

```
806         CONTINUE                                                    00628128
                                                                        00629124
                                                                        00630102
       **************   STRESS STRAIN CONTOUR DATA  *******             00631104
                                                                        00632104
      IF(JPRINT.LT.ISTART) GO TO 95                                     00633104
                                                                        00634128
809        CONTINUE                                                     00635128
                                                                        00636128
      IF(L.NE.1)   GO TO 95                                             00637104
                                                                        00638104
      *******   WRITE IT ON TAPE 18   *****                             00639104
                                                                        00640104
           IF(NPUNCH.NE.0)                                              00641118
                                                                        00642118
                                                                        00643119
                                                                        00644121
     *WRITE(18,204) NEL,SIG(1),SIG(2),SIG(4),SIG(8),PG,SHM,EPSV,STRAIN(400645132
     *)
204 FORMAT(I5,8E9.2,3X)
                                                                        00646132
                                                                        00647122
                                                                        00648*95
                                                                        00649*95
                                                                        00650*69
                                                                        00651*69
 95 CONTINUE                                                            00652*35
                                                                        00653*58
                                                                        00654*75
    ******      WRITE  ALL INFORMATION FOR OUTPUT ON TAPE **************00655*75
                                                                        00656*75
    *******     SELECT  OPTION  FOR  STRESS  OUTPUT    *****************00657*75
                                                                        00658*75
                                                                        00659*75
                                                                        00660*75
      IF(KFS.NE.4) GO TO 350                                            00661*95
                                                                        00662*75
                                                                        00663*75
    KK=KG+1                                                             00664*75
                                                                        00665*75
                                                                        00666*75
    IF(KK.GT.NUMMAT) REWIND 10
    IF(KK.GT.NUMMAT) KK=1
                                                                        00667*75
                                                                        00668*58
350 CONTINUE                                                            00669*95
                                                                        00670102
                                                                        00671116
       IF(IND.EQ.-1) GO TO 300                                          00672116
                                                                        00673116
       IF(NEGOIB.NE.0.AND.IFLAG.NE.0) GO TO 300                         00674104
                                                                        00675115
                                                                        00676115
                                                                        00677115
                                                                        00678132
       WRITE(6,162) (XP(I),I=1,16)
       WRITE(6,163) (LM(I),I=1,16)
162    FORMAT(5X,8E10.3)
163    FORMAT(5X,16I5)
```

```
            IF(KKS.EQ.4.AND.NCOND.NE.1) GO TO 300
                                                                        00679115
                                                                        00680*99
             WRITE (21) LM,XF                                           00681*99
            WRITE(6,162) (XP(I),I=1,16)
            WRITE(6,163) (LM(I),I=1,16)
                                                                        00682*70
                                                                        00683*70
  300 CONTINUE                                                          00684*70
                                                                        00685131
                IF(IFLAG.EQ.0)  GO TO 311                               00686131
                                                                        00687131
            IF(JPRINT.LT.ISTART) GO TO 311                              00688104
            JPRINT=0                                                    00689104
                                                                        00690105
  311    CONTINUE                                                       00691105
                                                                        00692*71
                 IF(IFLAG.EQ.0)  GO TO 310                              00693131
                                                                        00694131
        IF(ISTRES.LT.ISTRPR)  GO  TO  310                               00695*71
        ISTRES=0                                                        00696*71
  310 CONTINUE                                                          00697*71
                                                                        00698*69
                                                                        00699*69
        IF (KKS.NE.4)  GO  TO  900                                      00700*21
                                                                        00701*13
        IF(LC.NE.NLC)  GO  TO  900                                      00702*44
                                                                        00703*24
                                                                        00704*24
                                                                        00705*24
  999 FORMAT(//)                                                        00706
      PRINT  999                                                        00707
      WRITE(6,6000)                                                     00708
 6000 FORMAT(1H1/'     ****  MAX. LOAD VECTOR  NORMALIZED FOR  ALL  LOAD  00709*44
     *  CASES    ****  ')                                               00710*44
      WRITE(6,6001)  (AM(J),J=1,NNYP)                                   00711
      PRINT  999                                                        00712
      WRITE(6,6002)                                                     00713
 6002 FORMAT(1H1/'  ****  K  VECTOR  ******  ')                         00714*44
      WRITE(6,6001)  (FK(J),J=1,NNYP)                                   00715
 6001 FORMAT(5X,12E10.3)                                                00716
                                                                        00717*19
                                                                        00718*19
  900 CONTINUE                                                          00719*21
                                                                        00720*75
                                                                        00721*75
                                                                        00723*76
                                                                        00724*76
 2015 FORMAT(//// 37H EQUILIBRIUM ITERATION IN TIME STEP = ,I5//        00725*76
     *              37H NUMBER OF ITERATIONS              = ,I5  /)     00726*76
 2020 FORMAT(1H1/'        STRESSES AND FLOWS AT TIME    = ',F10.3//)    00727*36
                                                                        00728
 2001 FORMAT( 1H1/'        STRESSES AND FLOWS AT  TIME    = ',F10.3///'  00729
     * ELEMENT'/' NUMBER',3X,'SIGR',5X,'SIGZ',6X,'SIGT',5X,'SIGRZ',5X,'S00730
     *IGMAX',4X,'SIGMIN',5X,'MEAN NORMAL',5X,'MAX.DEVIATOR',5X,'ANGLE', 00731**7
     *5X,'PPT',4X,'PPC',4X,'PPB'//)                                     00732*19
 2005 FORMAT(I6,12E10.3)                                                00733**7
 2002 FORMAT (1H1/'********* STRAINS AND VOLUMES AT TIME *****= ',F10.3/00734**7
     *//' ELEMENT'/' NUMBER',3X,'EPSVS',5X'EPSVF',6X,'BULK-STRAIN',6X,'S00735**7
```

```
     *SEAR-STRAIN',10X,'RM',10X,'ZM'//)                                  00736**8
                                                                         00737*69
                                                                         00738*69
 2004 FORMAT(14/)                                                        00739*69
                                                                         00740
                                                                         00741*31
                                                                         00742*31
 2003 FORMAT(1X,/,'NELEMENT  X-CORD Y-CORD  STRESS-X  STRESS-Y  STRESS-Z00743*89
     *  STRESS-XY  MAX-STRESS  MIN STRESS  NORMAL  SHEAR  ANGLE  PORE',/00744*89
     *,1X,'NUM/IPT ', 91X,'MAX. ',1X,' MAX. ',1X,'PRES. ',2X,' FYLD ' ) 00745*90
 2007 FORMAT(2X,I5,2F6.2,(6(1X,E10.3)),5E9.2)                            00746*92
                                                                         00747*85
      RETURN                                                             00748*85
      END                                                                00749
```

```
          DATA SET STSTN       AT LEVEL 012 AS OF 83/02/11
       SUBROUTINE STSTN(IDW,PF,PS,FACN,LM,PMAX,NUMEL)                 00001**9
    IMPLICIT REAL*8(A-H,O-Z)                                          00002
                                                                      00003
                                                                      00004
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   00005
 .                                                                .   00006
 .                                                                .   00007
 .  S U B R O U T I N E                                           .   00008
 .                                                                .   00009
 .     TO FIND STRESS STRAIN LAW AND STRESSES FOR                 .   00010
 .     NONLINEAR MATERIAL MODELS                                  .   00011
 .                                                                .   00012
 .                                                                .   00013
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   00014
                                                                      00015
                                                                      00016
    COMMON/EL/IND,ICOUNT                                              00017**2
    COMMON/VAR/NG,KPRI                                                00018**2
    COMMON/BFLA/NS,NELTYP,ND,NUMGT,MODEL,NCON,IGW,NK015,NK016,NK017   00019**5
                                                                      00020*11
         COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT           00021*11
                                                                      00022*11
         COMMON/SORIN/TT(5,12),XP(16),INDNL,ITYP2D                    00023*12
       COMMON/MATMOD/STRESS(9),STRAIN(4),C(4,4),IPT,NEL               00024**9
                                                                      00025**9
       DIMENSION PMAX(NUMEL,1)                                        00026**9
                                                                      00027**9
                                                                      00028**9
                                                                      00029
         DIMENSION PROP(NCON),WA(IDWA,1)                              00030*11
                                                                      00031*11
                                                                      00032*11
                                                                      00033*10
    DIMENSION PS(16,8),FACN(4),LM(16)                                 00034**8
                                                                      00035
                                                                      00036
    DIMENSION  PF(1)                                                  00037**7
     IST=3                                                             00038
     IF (ITYP2D.EQ.0) IST=4                                            00039
     IP=NCON/4                                                         00040

       WRITE(6,160)
 160   FORMAT(5X,' *** PRINT IN STSTN **** ')
       WRITE(6,161)NS,NELTYP,ND,NUMGT,MODEL,NCON,IGW,NK015,NK016,NK017
 161   FORMAT(5X,10I5)
                                                                      00041
                                                                      00042
    D E F I N I T I O N   O F   S T R A I N                           00043
                                                                      00044
                                                                      00045
    LINEAR STRAIN TERMS                                               00046
                                                                      00047
                                                                      00048
                                                                      00049
                                                                      00050
    C A L C U L A T I O N   O F   S T R E S S - S T R A I N           00051
                 M A T R I X   A N D   S T R E S S E S                00052
      I                                                               00053
                                                                      00054
```

```
 80 GO TO (1,2),MODEL

.... MODEL = 1    L I N E A R   I S O T R O P I C

.... MODEL = 6    E L A S T O P L A S T I C (VON MISES YIELD CRITERION)

  1     CONTINUE

        CALL ELT2D6(PF,FACN,PS,LM,PMAX,NUMEL)
         WRITE(6,160)
      WRITE(6,161) NS,NELTYP,ND,NUMGT,MODEL,NCON,IGW,NK015,NK016,NK017
     RETURN

.... MODEL = 7    E L A S T O P L A S T I C (DRUCKER-PRAGER)

  2      CONTINUE
        CALL ELT2D7(PF,FACN,PS,LM,PMAX,NUMEL)
     RETURN

     END
```

```
         DATA SET SYMSOL      AT LEVEL 001 AS OF 81/04/28
      SUBROUTINESYMSOL (KKK,A,B,NN,MM)
      IMPLICIT REAL * 8 (A-H,O-Z)


      DIMENSION A(NN,1),B(1)

    1 FORMAT (16I5)
       GO TO (5,275),KKK
    5 CONTINUE
    5 DO 270 N = 1, NN
      DO 270 N = 1, NN
      DO 260 L = 2, MM
      IF(A(N,1).EQ.0)  WRITE(6,1000)  N,L,J,K
 1000 FORMAT(5X,4I5)
      C = A(N,L)/A(N,1)
      I = N+L-1
      IF(NN-I) 260,240,240
  240 J = 0
      DO 250 K = L, MM
      J = J + 1
  250 A(I,J) = A(I,J)-C*A(N,K)
  260 A(N,L) = C
  270 CONTINUE
      RETURN

  275 DO 290 N = 1, NN
      Y = B(N)
      DO 280 L = 2, MM
      I = N+L-1
      IF(NN-I) 290,280,280
  280 B(I) = B(I) - A(N,L)*Y
  290 B(N) = Y    /A(N,1)

      N = NN
  300 N = N - 1
      IF(N) 350,500,350
  350 DO 400 K = 2, MM
      L = N+K-1
      IF(NN-L) 400,370,370
  370 B(N) = B(N) - A(N,K)*B(L)
  400 CONTINUE
      GO TO 300
  500 RETURN
      END
```

```
     DATA SET TRIFAC      AT LEVEL 003 AS OF 83/02/24
  SUBROUTINE TRIFAC (A,B,MAXA,NEQB,MA,NBLOCK,NWA,NTB,NEQ,MI)
 IMPLICIT REAL*8(A-H,O-Z)

 CALLED BY:  STEP

 THIS ROUTINE DECOMPOSES THE SYSTEM MATRIX IN BLOCKS

 DIMENSION       A(NWA),B(NWA),MAXA(MI)

  COMMON/TAPES/NSTIF,NRED,NL,NR,IFIL(2),NUMBER

  MA2=MA - 2
 IF(MA2.EQ.0) MA2 = 1
  INC=NEQB - 1

 SET TAPE ASSIGNMENTS

 NSTIF = 4
 NRED  = 3
 NL    = 1
   NR=8

  N1=NL
  N2=NR
  REWIND NSTIF
  REWIND NRED
  REWIND N1
  REWIND N2

  MAIN LOOP OVER ALL BLOCKS
  DO 600 NJ=1,NBLOCK
  IF (NJ.NE.1) GO TO 10
  READ (NSTIF) A
  GO TO 100
10    IF (NTB.EQ.1) GO TO 100
  REWIND N1
  REWIND N2
  READ (N1) A

  FIND COLUMN HEIGHTS
100   KU=1
  KM=MIN0(MA,NEQB)
  MAXA(1)=1
  DO 110 N=2,MI
  IF (N-MA) 120,120,130
120   KU=KU + NEQB
  KK=KU
 MM = MIN0(N,KM)
  GO TO 140
130   KU=KU + 1
  KK=KU
  IF (N-NEQB) 140,140,136
136   MM=MM - 1
140   DO 160 K=1,MM
  IF (A(KK)) 110,160,110
160   KK=KK - INC
110   MAXA(N)=KK
 IF(A(1)) 172,174,176
 174 KK = (NJ-1)*NEQB +1
```

```
     IF(KK.GT.NEQ) GO TO 590
     WRITE (6,1000) KK
     STOP
 172 KK = (NJ-1)*NEQB +1
     WRITE (6,1010) KK

      FACTORIZE LEADING BLOCK
 176  DO 200 N=2,NEQB
      NH=MAXA(N)
      IF (NH-N) 200,200,210
210   KL=N + INC
      KU=NH
      K=N
      D=0.
      DO 220 KK=KL,KU,INC
      K=K - 1
      C=A(KK)/A(K)
      D=D + C*A(KK)
220   A(KK)=C
      A(N)=A(N) - D

      IF (A(N)) 222,224,230
224   KK=(NJ-1)*NEQB + N
      IF (KK.GT.NEQ) GO TO 590
     WRITE (6,1000) KK
      STOP
 222 KK = (NJ-1)*NEQB +N
     WRITE (6,1010) KK

230   IC=NEQB
      DO 240 J=1,MA2
      MJ=MAXA(N+J) - IC
      IF (MJ-N) 240,240,280
280   KU=MIN0(MJ,NH)
      KN=N + IC
      C=0.
      DO 300 KK=KL,KU,INC
300   C=C + A(KK)*A(KK+IC)
      A(KN)=A(KN) - C
240   IC=IC + NEQB

200   CONTINUE

      CARRY OVER INTO TRAILING BLOCKS
320   DO 400 NK=1,NTB
      IF ((NK+NJ).GT.NBLOCK) GO TO 400
      NI=N1
      IF ((NJ.EQ.1).OR.(NK.EQ.NTB)) NI=NSTIF
      READ (NI) B
      ML=NK*NEQB + 1
      MR=MIN0((NK+1)*NEQB,MI)
     MD = MI-ML
      KL=NEQB + (NK-1)*NEQB*NEQB
      N=1

      DO 500 M=ML,MR
      NH=MAXA(M)
      KL=KL + NEQB
     IF(NH-KL) 505,510,510
510   KU=NH
```

```
      DATA SET VECTOR      AT LEVEL 003 AS OF 81/11/19
   SUBROUTINE VECTOR(V,XI,YI,ZI,XJ,YJ,ZJ)                          00001
   IMPLICIT REAL * 8 (A-H,O-Z)                                     00002**2
   DIMENSION  V(4)                                                 00003**3
   X=XJ-XI                                                         00004
   Y=YJ-YI                                                         00005
   Z=ZJ-ZI                                                         00006
   V(4)=DSQRT(X*X+Y*Y+Z*Z)                                         00007
   V(3)=Z/V(4)                                                     00008
   V(2)=Y/V(4)                                                     00009
   V(1)=X/V(4)                                                     00010
   RETURN                                                          00011
   END                                                             00012
```

```
        DATA SET YLDCHK      AT LEVEL 016 AS OF 83/03/02

       SUBROUTINE YLDCHK(FYLD,IPLMAX,SIG,LG,NUMEL,NPT)                00001*15
    IMPLICIT REAL * 8(A-H,O-Z)                                        00002
                                                                      00003
    DIMENSION SIG(9)                                                  00004
                                                                      00005
        DIMENSION ALPA(5,200,4),SF(20,12),LM(16),PS(16,8),FACN(4)     00006**6
        DIMENSION RM(5),ZM(5)                                         00007*15
                                                                      00008**6
      COMMON/BEMOY/SG(5,200,4,4),NCOND                                00009*13
                                                                      00010*13
        COMMON/GRAND/CON(20),PHI(20),ITYPE(20)                        00011*15
                                                                      00012*13
                                                                      00013*12
       COMMON/ASIM/NYP,NSC,NSDC,NLC,NNYP,NNSC                         00014*12
                                                                      00015*12
       DIMENSION P(4,8),IT(5,12)                                      00016*12
                                                                      00017*12
                                                                      00018
      NDT=1
      IF(NCOND.EQ.1) NDT=4
                                                                      00019
       WRITE(6,900) IPLMAX,LG,NUMEL,NPT,NLC                           00020*16
 900   FORMAT(16I5)                                                   00021*16
                                                                      00022**6
                                                                      00023*12
        DO 100 LC=1,NLC                                               00024*12

        DO 320 N=1,NUMEL                                              00025*12


        DO 320 L=1,NDT                                                00026**6

 320    ALPA(LC,N,L)=0.0

        REWIND 11                                                     00027**6
                                                                      00028**6
       DO 200 N=1,NUMEL                                               00029**7
                                                                      00030**6
        READ(11)  SF,IT,P,LM,ALFA,E4,RM,ZM,KS,KF,IMAT,BLK,EMU,YOUNG,P00031**6
   *OIS,VOL,PS,FACN                                                   00032**6
                                                                      00033**6
                                                                      00034**6
         DO 300 L=1,NDT                                               00035**6
                                                                      00036**6
         DO 400 I=1,4                                                 00037**6
 400     SIG(I)=SG(LC,N,L,I)                                          00038**6
                                                                      00039**6
                                                                      00040**6
          IF(IPLMAX) 600,700,700                                      00041**6
                                                                      00042**6
                                                                      00043**6
 600      SIGX2=SIG(1)**2                                             00044**6
          SIGY2=SIG(2)**2                                             00045**6
          SIGXY=SIG(1)*SIG(2)                                         00046**6
          SIGXY2=3.*(SIG(4)**2)                                       00047**6
          Z=(SIGX2+SIGY2-SIGXY+SIGXY2)                                00048**6
          YLD=CON(IMAT)                                               00049*18
```

```
          YLD=YLD**2                                               00050*18
          IF(Z.EQ.0.0) GO TO 751
          ALPA(LC,N,L)=DSQRT(YLD/Z)                                00051*18
751       CONTINUE
          WRITE(6,1000) LC,N,L,SIGX2,SIGY2,SIGXY,Z,YLD,COH(IMAT),  00052*18
   *      ALPA(LC,N,L)                                             00053*18
          GO TO 300                                                00054**6
                                                                   00055**6
700       CONTINUE                                                          00056**6

          IF(ITYPE(IMAT).EQ.0) GO TO 702
          IF(ITYPE(IMAT).EQ.1) GO TO 702
          IF(ITYPE(IMAT).EQ.2) GO TO 703

703       SIGX=SIG(1)                                              00057**6
          SIGY=SIG(2)                                              00058**6
          SIGXY=SIG(4)                                             00059**6
          YP=(SIGX+SIGY)
          YP=YP*(DSIN(PHI(IMAT)))
          YN=(SIGX-SIGY)                                           00060**6
          YN=YN**2                                                 00061**6
          SXY=4.*((SIGXY)**2)                                      00062**6
          TERM=(YN+SXY)                                            00063**6
          YLD=(2.*(COH(IMAT))*(DCOS(PHI(IMAT)))+YP)                00064**6

          FLD=DSQRT(TERM)                                          00065**6

          GO TO 704

702       DM=(SIG(1)+SIG(2)+SIG(3))/3.
          DX=SIG(1)-DM
          DY=SIG(2)-DM
          DZ=SIG(3)-DM
          DS=SIG(4)
          FLD=0.5*(DX*DX+DY*DY+DZ*DZ)+DS*DS
          A=3.*(COH(IMAT)**2)*(DCOS(PHI(IMAT))**2)
          B=3.+(DSIN(PHI(IMAT))**2)
          C=(DSIN(PHI(IMAT))**2)
          AL=(C/B)*(1./3.)
          AL=DSQRT(AL)

          GK=A/B
          YLD=DSQRT(GK)
          FLD=DSQRT(FLD)+AL*DM
704        IF(FLD.EQ.0.0) GO TO 750
          ALPA(LC,N,L)=(YLD/FLD)                                   00066**6
                                                                   00067**6
750       CONTINUE
                                                                   00068*16
        WRITE(6,1000) LC,N,L,SIG(1),SIG(2),SIG(3),COH(IMAT),PHI(IMAT) 00069*16
   *,YLD,FLD,ALPA(LC,N,L)                                          00070*16
                                                                   00071*16
                                                                   00072*16
1000 FORMAT(5X,3I5,8E10.3)                                         00073*16
                                                                   00074**6
300        CONTINUE                                                00075**6
200        CONTINUE                                                00076**6
                                                                   00077**6
                                                                   00078**6
```

```
           DATA SET YLDIN      AT LEVEL 010 AS OF 82/06/12
      SUBROUTINEYLDIN(NSC,NYP,NUMEL,NES,NEY,AM,EK,AN,AMAX)              00001
                                                                        00002**8
      IMPLICIT REAL*8(A-H,O-Z)                                          00003
                                                                        00004**8
C                                                                       00005*10
      COMMON/BOBBY/KKS,NUMGT,NC,LOOP,NORN,NINT,IDWA,NPT                 00006*10
      DIMENSION AM(1),EK(1),AN(NES,1),AMAX(NYP,1,1)                     00007*10
                                                                        00008**8

      COMMON/BENDY/SG(5,200,4,4),NCOND

      NDT=1
      IF(NCOND.EQ.1) NDT=4
                                                                        00009**8
           DO 2 I=1,NEY                                                 00010
           AM(I)=0.0                                                    00011
    2 EK(I)=0.0                                                         00012**3
      DO   4  I=1,NES                                                   00013**4
      DO   3  J=1,NEY                                                   00014**3
    3 AN(I,J)=0.0                                                       00015**7
    4 CONTINUE                                                          00016**4
   15 FORMAT(5X,12E10.3)                                                00017**6
           DO 7 M=1,NUMEL                                               00018
                                                                        00019*10
      DO   7  I=1,NDT                                                   00020*10
                                                                        00021*10
           DO 7 J=1,NYP                                                 00022
                                                                        00023*10
                                                                        00024*10
      AMAX(J,M,I)=0.0                                                   00025*10
                                                                        00026*10
    7 CONTINUE                                                          00027*10
                                                                        00028
      RETURN                                                            00029
      END                                                               00030
```

```
      DATA SET YLDLIN      AT LEVEL 027 AS OF 83/02/28
   SUBROUTINE YLDLIN(LC,NYP,NSC,ANI,EKJ)                           00001**6
                                                                   00002**4
   IMPLICIT REAL*8(A-H,O-Z)                                        00003
                                                                   00004**4
                                                                   00005**4
   DIMENSION ANI(NYP,1),EKJ(NYP)                                   00006
   COMMON/CONTR/DD,HED(8),RADN,NUMNP,NUMEL,NEQ,MBAND,NCYCL,NDYN,IPLNA00007**2
  *X,ISTRPR,NPRINT,ISTART,NEQH,MBANDH                              00008**2
  *,NEQS,MBANDS                                                    00009*24
     COMMON/BOBBY/KKS,NUMMAT,NC,LOOP,NDRN,NINT,IDWA,NPT            00010*24
                                                                   00011*24
                                                                   00012**2
       COMMON/GRAND/COH(20),PHI(20),ITYP(20)                       00013*26
                                                                   00014*26
       COMMON/BENOY/SG(5,200,4,4),NCOND,NTENS
                                                                   00015*25
                                                                   00016*21
      COMMON/GORA/IFLAG,ITR                                        00017*24
                                                                   00018*24
                                                                   00019**2
                                                                   00020**2
   DIMENSION  ANG(14,14)                                           00021*14
                                                                   00022**2
       WRITE(6,301) NUMMAT,COH(1),PHI(1),ITYP(1)                   00023*27
 301   FORMAT(5X,I5,2E10.3,I5)                                     00024*27
                                                                   00025*16
                                                                   00026*16
                                                                   00027*16
   PI=4.0*DATAN(1.0D0)                                             00028*16
   RADN=PI/180.                                                    00029*16
 500 FORMAT(I5)                                                    00030**5
 400 FORMAT(1H1/'   TYPE OF LINEARIZATION MODEL  PROPOSED    = ',I5/ 00031**5
   *  ' EQ.0. .....  GIODA METHOD .....                      ' /   00032**5
   *  ' EQ.1 ..........  ANDERHEGGEN  METHOD            ' /        00033*16
   *  ' EQ.2 .............. MOHR COULOMB                ' /)       00034*16
                                                                   00035**2
 200 FORMAT(5X,2E10.3)                                             00036**2
                                                                   00037*11
                                                                   00038**2
                                                                   00039**2
   REWIND 10                                                       00040**2
                                                                   00041**2
                                                                   00042**2
   DO  300 N=1,NUMMAT                                              00043**2
                                                                   00044**2
 *******   INITIALIZE  ************                                00045**2
                                                                   00046**2
   DO  700  J=1,NYP                                                00047**2
   DO  700  I=1,NYP                                                00048*12
   ANI(J,I)=0.0                                                    00049**7
   ANG(J,I)=0.0                                                    00050**7
 700 CONTINUE                                                      00051**7
          WRITE(6,400) ITYP(N)                                     00052*26
           ITYPE=ITYP(N)                                           00053*26
   WRITE(6,200)  COH(N),PHI(N)                                     00054*12
                                                                   00055**2
   PHI(N)=PHI(N)*RADN
                                                                   00056**8
   WRITE(6,200)  COH(N),PHI(N)                                     00054*12
```

```
                                                                        00057**8
      IF(IPLMAX)  900,650,655                                           00058**8
  650 CONTINUE                                                          00059**8
      IF(ITYPE.NE.0)  GO  TO  800                                       00060**6
                                                                        00061**2
                                                                        00062**2
      AGM=(1.+DSIN(PHI(N)))/(1.-DSIN(PHI(N)))                           00063**3
      AA=DSQRT(1.+AGM**2.)                                              00064**2
      BB=(2.*COH(N)*DCOS(PHI(N)))/(1.-DSIN(PHI(N)))                     00065**3
      ANI(1,1)= 1./AA                                                   00066**2
      ANI(1,2)=-AGM/AA                                                  00067**2
      ANI(2,1)=-AGM/AA                                                  00068**2
      ANI(2,2)=1./AA                                                    00069**2
       ANI(3,2)=ANI(2,2)                                                00070**2
      ANI(3,3)=ANI(2,1)                                                 00071**2
      ANI(4,2)=ANI(2,1)                                                 00072**2
      ANI(4,3)=ANI(1,1)                                                 00073**2
      ANI(5,1)=ANI(1,1)                                                 00074**2
      ANI(5,3)=ANI(1,2)                                                 00075**2
      ANI(6,1)=ANI(1,2)                                                 00076**2
      ANI(6,3)=ANI(1,1)                                                 00077**2
                                                                        00078**2
                                                                        00079**2
      DO  600 J=1,NYP                                                   00080**2
  600 ENJ(J)=BB/AA                                                      00081**2
                                                                        00082**5
                                                                        00083**5
      GO  TO  900                                                       00084**5
                                                                        00085**5
  800 CONTINUE                                                          00086**6
      GO  TO  (1,2),ITYPE                                               00087*16
                                                                        00088**5
                                                                        00089**5
      ********   ANDERHEGGEN   MODEL  FOR  LINEARIZATION MODEL     *****00090**5
                                                                        00091*16
    1 CONTINUE                                                          00092*16
                                                                        00093**5
      S=DSIN(PHI(N))                                                    00094**5
      C=DCOS(PHI(N))                                                    00095**5
      CC=1./C                                                           00096**5
      ANI(1,1)=1.-S                                                     00097**5
      ANI(1,2)=-S                                                       00098*15
      ANI(1,3)=-1.-S                                                    00099**5
      ANI(1,4)=-1.-S                                                    00100**5
      ANI(1,5)=-S                                                       00101**5
      ANI(1,6)=1.-S                                                     00102**5
      ANI(2,1)=-1.-S                                                    00103**5
      ANI(2,2)=-S                                                       00104**5
      ANI(2,3)=1.-S                                                     00105**5
      ANI(2,4)=1.-S                                                     00106**5
      ANI(2,5)=-S                                                       00107**5
      ANI(2,6)=-1.-S                                                    00108**5
      ANI(3,1)=2/1.73                                                   00109**5
      ANI(3,2)=E/1.73                                                   00110**5
      ANI(3,3)=ANI(3,1)                                                 00111**5
      ANI(3,4)=-ANI(3,1)                                                00112**5
      ANI(3,5)=-ANI(3,2)                                                00113**5
      ANI(3,6)=-ANI(3,1)                                                00114**5
                                                                        00115**7
                                                                        00116**7
```

```
      WRITE(6,750)                                                      00117*10
  750 FORMAT(1H1/' *****   PRINT  OF  MATRICES  IN  ITDRIN  *** ' /)    00118*10
      WRITE(6,751)          ((ANI(I,J),J=1,NYP),I=1,NSC)                00119*10
  751 FORMAT(5X,3F17.3)                                                 00120*10
      DO  1200  J=1,NYP                                                 00121**7
      DO  1200  I=1,NSC                                                 00122**7
 1200 ANG(J,I)=ANI(I,J)*CC                                              00123**7
      DO  1000  J=1,NYP                                                 00124**5
      DO  1000  I=1,NSC                                                 00125**5
 1000 ANI(J,I)=ANG(J,I)                                                 00126**7
                                                                        00127*10
                                                                        00128*10
      WRITE(6,752)  ((ANG(J,I),I=1,NSC),J=1,NYP)                        00129*10
      WRITE(6,752)  ((ANI(J,I),I=1,NSC),J=1,NYP)                        00130*10
  752 FORMAT(5X,3F12.3)                                                 00131*10
      DO  1100  J=1,NYP                                                 00132**5
 1100 FKJ(J)=2.*CON(M)                                                  00133**6
                                                                        00134*20
      GO  TO  1500                                                      00135*20
                                                                        00136**5
                                                                        00137*16
    2 CONTINUE                                                          00138*16
      X=(2.*PI)/NYP                                                     00139*16
      Y=PI/NYP                                                          00140*16
                                                                        00141*16
                                                                        00142*16
      DO  3  KP=1,NYP                                                   00143*16
      ANI(KP,1)=DCOS(X*KP)-(DSIN(PHI(N))*DCOS(Y))                       00144*16
      ANI(KP,2)=DCOS(X*KP)+(DSIN(PHI(N))*DCOS(Y))                       00145*16
                                                                        00146*18
      ANI(KP,2)=-ANI(KP,2)                                              00147*18
                                                                        00148*18
      ANI(KP,3)=2.*(DSIN(X*KP))                                         00149*16
      FKJ(KP)=2.*CON(N)*(DCOS(PHI(N)))*(DCOS(Y))                        00150*16
    3 CONTINUE                                                          00151*16
                                                                        00152*16
                                                                        00153*16
                                                                        00154**5
  900 CONTINUE                                                          00155**5
                                                                        00156**8
                                                                        00157**8
      IF(IPLMAX.EQ.0.OR.IPLMAX.EQ.1)  GO  TO  1500                      00158**8
                                                                        00159**8
      ANI(1,1)=1.0                                                      00160**8
      ANI(1,3)=1.0                                                      00161**8
      ANI(2,1)=1.0                                                      00162**8
      ANI(2,3)=-1.0                                                     00163**8
      ANI(3,1)=-1.0                                                     00164**8
      ANI(3,3)=1.0                                                      00165**8
      ANI(4,1)=-1.0                                                     00166**8
      ANI(4,3)=-1.0                                                     00167**8
      ANI(5,2)=1.0                                                      00168**8
      ANI(5,3)=1.0                                                      00169**8
      ANI(6,2)=1.0                                                      00170**8
      ANI(6,3)=-1.0                                                     00171**8
      ANI(7,2)=-1.0                                                     00172**8
      ANI(7,3)=1.0                                                      00173**8
      ANI(8,2)=-1.0                                                     00174**8
      ANI(8,3)=-1.0                                                     00175**8
      ANI(9,1)=1.0                                                      00176**8
```

```
      ANI(9,2)=-1.0                                                     00177**8
      ANI(9,3)=1.0                                                      00178**8
      ANI(10,1)=1.0                                                     00179**8
      ANI(10,2)=-1.0                                                    00180**8
      ANI(10,3)=-1.0                                                    00181**8
      ANI(11,1)=-1.0                                                    00182**8
      ANI(11,2)=1.0                                                     00183**8
      ANI(11,3)=1.0                                                     00184**8
       ANI(12,1)=-1.0                                                   00185**8
      ANI(12,2)=1.0                                                     00186**8
      ANI(12,3)=-1.0                                                    00187**8
      ANI(13,3)=1.0                                                     00188**8
      ANI(14,3)=-1.0                                                    00189**8
      ND=NYP-2                                                          00190**9
      WRITE(6,1700)  IPLNAX,CON(N)                                      00191*10
 1700 FORMAT(5X,I5,E10.3)                                               00192*10
                                                                        00193**9
                                                                        00194**9
      DO  1600 J=1,ND                                                   00195**9
 1600 EKJ(J)=CON(N)                                                     00196**8
                                                                        00197**9
      EKJ(NYP-1)=CON(N)/1.73                                            00198*19
      EKJ(NYP)=EKJ(NYP-1)                                               00199**9

      GO TO 1500


  655 CONTINUE
      *******  AXISYMMETRIC  PROBLEM  ******

                                                                        00200**8
 1500 CONTINUE                                                          00201**8
      WRITE(10) ((ANI(J,I),I=1,NSC),J=1,NYP),(EKJ(J),J=1,NYP)           00202**2
                                                                        00203**2
                                                                        00204**2
      WRITE(6,502)                                                      00205
      DO  504  J=1,NYP                                                  00206
  504 WRITE(6,503)J,(ANI(J,I),I=1,NSC),EKJ(J)                           00207
                                                                        00208**4
                                                                        00209**4
  501 FORMAT(4E10.3)                                                    00210
  502 FORMAT(//43H0INPUT TABLE 2B. LINEARIZED YIELD FUNCTION  //5X,     00211
     16H YIELD,5X,27HDIRECTION COSIN OF THE UNIT,5X,13HDISTANCE FROM /  00212
     25X,6H PLANE,19X,13HNORMAL VECTOR,6X,12HSTRESS ORIGN/)             00213
  503 FORMAT(5X,I6,2X,3F10.3,8X,F10.3)                                  00214
  300 CONTINUE                                                          00215**2
                                                                        00216**2
                                                                        00217**2
                                                                        00218**2
      RETURN                                                            00219
      END                                                               00220
```

```
        DATA SET YLDMX      AT LEVEL 020 AS OF 83/02/11
   SUBROUTINE YLDMX(NEL,NYP,NSC,ANI,AMJ,LC,NLC,AM,EK,AN,NES,AMAX,      00001**3
                                                                       00002*11
  *EKU,SIG,IK,NPT,IPT,VOL)                                             00003*19
                                                                       00004*10
   IMPLICIT REAL*8 (A-H,O-Z)                                           00005
                                                                       00006*10
                                                                       00007*10
                                                                       00008*18
   COMMON/BOBBY/KKS,NUMGT,NC,LOOP,NDRN,NINT,IDWA,NPT                   00009*18
   COMMON/CONTL/ND,HED(8),RADN,NASIM,NUMSL,NKK,MBAND,NCYCL             00010*16
  *,NDYN,IPLMAX,ISTRPR,NPRINT,ISTART,NEGH,MBANDH,NEGS,MBANDS           00011*20
                                                                       00012*20
    COMMON/BENOY/SG(5,200,4,4),NCOND
    COMMON/DIMEL/N43,KPT,KPT1,KK,NPRNT

                                                                       00013*10
   DIMENSION ANI(NYP,1),AMJ(14),AM(1),EK(1),AN(NES,1)                  00014*16
   DIMENSION ANIJ(3),AMAX(NYP,1,1)                                     00015*18
   DIMENSION EKU(1),SIG(9)                                             00016**4
                                                                       00017
    ************     N * TRANSPOSED * SIGMA  CALCULATION  ***********00018
COMPUTE THE MAXIMUM WITH RESPECT TO THE TIME OF THE STRESS COMPONTANT00019

NORMAL TO EACH YIELD PLANE FOR EACH ELEMENT.                           00020
                                                                       00021
     NDT=1
   IF(NCOND.EQ.1) NDT=4
   M=NEL                                                               00022
      DO14 J=1,NYP                                                     00023
        AMJ(J)=0.0                                                     00024
      DO 4 I=1,NSC                                                     00025
        ANIJ(I)=ANI(J,I)                                               00026
   IF(I.EQ.NSC)  GO  TO  5                                             00027
   AMJ(J)=AMJ(J)+ANIJ(I)*SIG(I)                                        00028
   GO  TO  4                                                           00029
 5 AMJ(J)=AMJ(J)+ANIJ(I)*SIG(I+1)                                      00030
 4 CONTINUE                                                            00031
                                                                       00032**8
   ******************   IMPOSE  ARTIFICIAL  CONDITION  SO  THAT  STRE00033*12
     ARE  ALAWAYS   IN  COMPRESSETION   *****************************00034*12
                                                                       00035*12
                                                                       00036*12
                                                                       00037*12
                                                                       00038**8
    ***********    MODIFY  FOR  VERTICAL  LOADING  LOAD  CASE  NO.  300039**8
                                                                       00040**8
                                                                       00041*17
   IF(LC.EQ.NLC)  GO  TO  14                                           00042*17
                                                                       00043*12
   ******   SET  ABSOLUTE  VALUES  FOR  MAXIMA  CHECKING   *********00044*12
                                                                       00045*12
   AND=AMJ(J)                                                          00046*14
                                                                       00047*18
   ANG=AMAX(J,M,IPT)                                                   00048*18
                                                                       00049*18
   IF(AND.LT.ANG)  GO  TO  14                                          00050*12
                                                                       00051*18
   AMAX(J,M,IPT)=AMJ(J)                                                00052*18
                                                                       00053*18
```

```
14      CONTINUE                                                           00054
                                                                           00055
                                                                           00056
 ASSEMBLE THE MAXIMUM STRESS COMPONANT IN ONE VECTOR AM ,THE NORMAL       00057
 AND THE NORMALTY MATRIX AN.                                               00058
 DISTANCE FROM STRESS ORIGN TO THE YIELD PLANES IN ONE VECTOR EK ,         00059
                                                                           00060
    IF(LC.LT.NLC.AND.NC.LT.NCYCL)  GO  TO  11                              00061*16
       DO 6 J=1,NYP                                                        00062
                                                                           00063*18
    LX=J+(M-1)*NYP*NDI+(IPT-1)*NYP                                         00064*18
                                                                           00065*18
    AM(LX)=AMAX(J,M,IPT)                                                   00066*18
                                                                           00067**8

    IF(KPT.EQ.1) GO TO 15

                                                                           00068**8
     *************   MODIFY  K- VECTOR  FOR  VERTICAL  LOADING  ******00069**8
                                                                           00070**8
    EK(LX)=(EKJ(J)-AMJ(J))                                                 00071*15

    GO TO 6

 15 CONTINUE

    EK(LX)=EKJ(J)

  6 CONTINUE                                                               00072**5
                                                                           00073**8
                                                                           00074**8
                                                                           00075**8
    DO  200 J=1,NYP                                                        00076**9
                                                                           00077*18
    LX=J+(M-1)*NYP*NDI+(IPT-1)*NYP                                         00078*18
    DO  100  I=1,NSC                                                       00079**9
    LY=I+(M-1)*NSC*NDI+(IPT-1)*NSC                                         00080*18
100 AN(LY,LX)=ANI(J,I)                                                     00081**9
200 CONTINUE                                                               00082**9
 11     CONTINUE                                                           00083
                                                                           00084
    RETURN                                                                 00085
    END                                                                    00086
```

```
      SUBROUTINE ZXMIN(CE,A,AN,C,AM,EK,GB,SU,SQ,ICOLMS,IDES,ROW,COPI,
     *PSOL,DSOL,G,IER,NPK,NPJ,NEQS,NUMEL,NYP,NSC,SD,SN,CT,IH,NUMNP)
C
C
C      *********  DIMENSION  *******
C
C
C
       IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/TARA/LPC,N,M1,M2,NPT,KPT,NPRNT,M,MM
C
      DIMENSION CE(1),A(MM,1),AN(NPJ,1),C(NPJ,1),AM(1),EK(1),GB(1),SU(1)
     *,SQ(1),ICOLMS(1),IDES(1),ROW(1),COPI(MM,1),PSOL(1),DSOL(1),G(1)
C
       DIMENSION SD(NPJ,1),SN(1),CT(NPJ,1),IH(NUMNP,1)
C
C
C
   95  CONTINUE
       WRITE(6,96) LPC,M1,M2,N,NPT,KPT,NPRNT,M,MM,NPJ,NPK,NEQS
   96  FORMAT(16I5)
C
       IF(NPT.EQ.2) GO TO 14
C
       DO 10 J=1,NPK
   10  A(LPC,J)=AM(J)
C
       NO=NPK+1
       DO 11 J=NO,N
   11  A(LPC,J)=0.D0
C
C
C      DO 33 I=1,NEQS
C  33  SU(I)=0.0
C
       GO TO 18
C
C
   14  CONTINUE
       JJ=NPK+1
       JJ1=NPK+NEQS
C
       DO 19 J=1,NPK
   19  A(LPC,J)=0.D0
       DO 20 J=JJ,JJ1
   20  A(LPC,J)=GB(J-NPK)
       JJ2=NPK+NEQS+1
       DO 21 J=JJ2,N
   21  A(LPC,J)=-GB(J-JJ1)
C
C
   18  CONTINUE
C
C
       DO 1 I=1,NPK
       A(2,I)=+EK(I)
    1  CONTINUE
C
```

```
C      WRITE(6,200) (A(2,I),I=1,NPK)
      N1=NPK+1
      N2=NPK+NEQS
C
      DO 2 I=N1,N2
    2 A(2,I)=-SQ(I-NPK)
      N3=NPK+NEQS+1
      N4=N
      DO 3 I=N3,N
    3 A(2,I)=+SQ(I-N2)
C
C
      DO 4 I=1,N
      SUM=0.0
      DO 5 J=3,LPC
    5 SUM=SUM+A(J,I)
      G(I)=-(A(2,I)+SUM)
    4 CONTINUE
      WRITE(6,34) (SQ(I),I=1,NEQS)
   34 FORMAT(12E10.3)
C
      II=1
C
      DO 6 I=1,N
    6 A(II,I)=G(I)
C
C
      DO 8 I=1,LPC
    8 SQ(I)=0.0
      SQ(LPC)=1.D0
C
C
      ICOLMS(II)=2
      ICOLMS(II+1)=1
      DO 9 I=3,LPC
    9 ICOLMS(I)=1
C
C
      ROW(II)=-1.D0
      ROW(II+1)=1.D0
C
C
     DO 13 I=3,LPC
   13 ROW(I)=0.D0
C
C
     KG=1
     ITMAX=5*N
     IR=M+1
C
C
     DO 56 I=1,MM
     DO 12 J=1,MM
     COPI(I,J)=0.0
   12 CONTINUE
   56 CONTINUE
     DO 17 I=1,MM
     COPI(I,I)=1.D0
   17 CONTINUE
```

```
C
      COPI(I1,I1+1)=1.D0
C
C
      IDES(I1)=N+2
      IDES(I1+1)=N+1
      DO 140 I=3,LPC
  140 IDES(I)=N+1
C
C
C       WRITE(6,100)
  100 FORMAT(1H1/'  PRINT  ')
C       WRITE(6,200) ((A(I,J),J=1,12),I=1,15)
C       WRITE(6,999)
C       WRITE(6,200) ((A(I,J),J=13,24),I=1,15)
C       WRITE(6,999)
C       WRITE(6,210) ((A(I,J),J=25,32),I=1,15)
C       WRITE(6,999)
C       WRITE(6,210) ((A(I,J),J=33,40),I=1,15)
C
  200 FORMAT(5X,12E10.3)
  210 FORMAT(5X,8E10.3)
      WRITE(6,100)
C
C        WRITE(6,200) (SQ(I),I=1,LPC)
C
C        WRITE(6,999)
C
  999  FORMAT(//)
C        WRITE(6,150) (ICOLMS(I),I=1,LPC)
C 150    FORMAT(16I5)
C
C        WRITE(6,999)
C        WRITE(6,200) (ROW(I),I=1,LPC)
C
C        WRITE(6,999)
C        WRITE(6,200) ((COPI(I,J),J=1,12),I=1,MM)
C
C        WRITE(6,999)
C        WRITE(6,150) (IDES(I),I=1,LPC)
C
C
C
        IPHASE=1
C
C
      CALL ZX0LP(IPHASE,A,SQ,ICOLMS,ROW,KG,M,N,ITMAX,LPC,MM,COPI,IDES,
     *PSOL,DSOL,IER)
C
      WRITE(6,91) IER,ITMAX
   91 FORMAT(5X,2I5)
C
      IF(IER.NE.0) GO TO 16
C
      DO 15 I=2,LPC
   15 IF(IDES(I).GT.(N+1)) GO TO 29
C
C
C       WRITE(6,999)
C       WRITE(6,150) (IDES(I),I=1,LPC)
```

```
C
C
C      WRITE(6,999)
C      WRITE(6,150) (ICOLMS(I),I=1,LPC)
C
      LPC=1
      DO 22 I=1,LPC
   22 ICOLMS(I)=ICOLMS(I)-ISIGN(1,ICOLMS(I))
C
       ROW(II)=1.D0
       M=M-1
C
       IPHASE=2
C
       I=1
       L=2
C
       CALL ZX0LP(IPHASE,A(L,1),SQ(L),ICOLMS,ROW,KG,M,N,ITMAX,LPC,MM,
     *COPT(L,1),IDES(L),PSOL,DSOL,IER)
C
      WRITE(6,91) IER,ITMAX
C
        DO 23 I=1,N
   23   G(I)=0.0
C
       M=M2+1
       DO 24 I=1,M
       IF(IDES(I+1).EQ.(N+1))  Z=-PSOL(I)
       IF(IDES(I+1).LE.N) G(IDES(I+1))=PSOL(I)
   24  CONTINUE
C
C
        DO 25 I=1,NEQS
        G(I+NPK)=G(I+NPK)-G(I+NPK+NEQS)
   25   CONTINUE
C
        CE(1)=Z
C
         DO 26 I=1,NPK
   26   CE(I+1)=G(I)
C
        DO 27 I=1,NEQS
   27  CE(I+NPK+1)=G(I+NPK)
C
        NG=NPK+NEQS+1
C        WRITE(6,52)
C  52   FORMAT(1H1/' PRINT OF ZAMIN')
C        WRITE(6,53) ((CE(I),I=1,NG)
C  53   FORMAT(5X,12E10.3)
C
C
       CALL OUTPUT(NOMEL,FTP,ESC,NPJ,NPK,CE,Z,AN,SD,SN,CT,SU,NEQS,IH,
     * NUMNP)
       GO TO 55
C
   29 CONTINUE
C
      WRITE(6,50)
C
```

```
  50  FORMAT(1H1/' PHASE1 DID NOR REMOVE ALL ARTIFICIAL VARIABLES ')
C
  16  CONTINUE
C
       WRITE(6,49)
  49  FORMAT(1H1/'  PHASE ERROR SET')
C
  55  CONTINUE
C
      RETURN
      END
```