DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

NUMERICAL ANALYSIS OF SOIL-PILE INTERACTION USING FINITE ELEMENT METHOD

by
Devrim ERDOĞAN

December, 2005
İZMİR

NUMERICAL ANALYSIS OF SOIL-PILE INTERACTION USING FINITE ELEMENT METHOD

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in
Civil Engineering, Geotechnics Program

by

Devrim ERDOĞAN

December, 2005
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **NUMERICAL ANALYSIS OF SOIL-PILE INTERACTION USING FINITE ELEMENT METHOD** completed by **DEVRİM ERDOĞAN** under supervision of **ASST. PROF. DR. GÜRKAN ÖZDEN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Asst. Prof. Dr. Gürkan ÖZDEN

Supervisor

Prof. Dr. Arif Şengün KAYALAR

Thesis Committee Member

Prof. Dr. Necdet TÜRK

Thesis Committee Member

Assoc. Prof. Dr. Haydar LİVATYALI

Examining Committee Member

Asst. Prof. Dr. Selim ALTUN

Examining Committee Member

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I am grateful to Assist. Prof. Dr. Gürkan ÖZDEN, the supervisor of this Dissertation, for his great support and valuable contributions all through my graduate study. He is a dedicated researcher and a teacher who advised me with patience through the hard and frustrating stages of this research.

I would also like to thank Prof. Dr. Arif Şengün Kayalar and Prof. Dr. Necdet TÜRK for taking time to serve in my thesis committee and valuable contributions.

I would like to thank Assoc.Prof. Dr. Recep Yılmaz, Prof. Orhan Yüksel and Prof.Dr.Hikmet Aydın in Civil Engineering Department of Ege University for the support they gave me during my stay at Wayne State University.

I would like to give my special thanks to Prof. Dr. Takaaki Kagawa, for his great contributions to the progress of this research during my stay as a researcher at Wayne State University. I feel very lucky to have worked with such a dedicated researcher who introduced me the different research areas in numerical modeling and hence helped me improve my version of dealing with engineering problems.

I would also like to thank my friend Mehmet Kuruoğlu for his great support in the most critical days of the writing process of this thesis.

Finally, I'm so grateful to my mother Ülgen OSKAY for her continuous support and understanding in the most critical and frustrating times of my graduate study since she clearly understands the meaning of this period due to her also being in academic life. For me, Ph.D is also the success of the people living with or closer to the person doing the Ph.D, so I have been one of the luckiest for having someone supportive in the most frustrating times of this period.

Devrim **ERDOĞAN**

NUMERICAL ANALYSIS OF SOIL – PILE INTERACTION USING FINITE ELEMENT METHOD

ABSTRACT

A nonlinear finite element algorithm has been developed in this study. The motivation behind this attempt was the investigation of seismic inertial soil-pile-structure interaction effects on the response of the surrounding soil to the laterally deforming pile. Emphasis has been given to the understanding of excess pore pressure development since this parameter governs the mechanism by determining effective stress field around the pile. A state of the art strain space plasticity constitutive model has been employed in the finite element program. It should be noted that both contractive and dilative soil behavior are accounted for in the utilized model. Besides, the model involves numerical robustness features enabling computation of the soil response near failure line where the soil attains small strength and stiffness. The finite element discretization and solution of dynamic governing differential equations were made using well established FEM solution procedures. The developed program has proved to be applicable to dynamic soil response applications although some convergence problems, which could be overcome by further studies, were met in the analysis of SPSI problems. The algorithm can also be used in the study of pure soil dynamics problems in addition to the soil-pile-structure interaction cases.

Keywords: finite element method, soil plasticity, soil-pile-structure interaction

ZEMİN-KAZIK ETKİLEŞİMİ PROBLEMİNİN SONLU ELEMANLAR YÖNTEMİ İLE SAYISAL ANALİZİ

ÖZ

Bu çalışmada doğrusal olmayan bir sonlu elemanlar algoritması geliştirilmiştir. Bu gayretin arkasındaki motivasyon zemin-kazık-yapı etkileşiminin sismik yükler altında yatay yönde deforme olan bir kazığın etrafındaki zemin üzerindeki etkilerinin araştırılmasıdır. Aşırı boşluk suyu basıncının araştırlımasına özel önem verilmiştir çünkü bu parametre, kazık etrafındaki zemin davranışını belirleyen efektif gerilme dağılımını kontrol etmektedir. Sonlu elemanlar programında mevcut bilgi düzeyini yansıtan birim deformasyon uzayı tabanlı bir plastisite modeli kullanılmıştır. Kullanılan bu modelin zeminde sıkışma ve genleşme davranışlarını modelliyebildiği göz önüne alınmalıdır. Ayrıca modelde zemin göçme zarfına yakınken ortaya çıkan sayısal hesaplama zorluklarını aşmaya yarayan özelikler bulunmaktadır. Bilindiği gibi göçmeye yakınlaşıldığında zeminin dayanım ve rijitliği oldukça düşmektedir. Problemi tanımlayan diferansiyel denklemlerin dinamik sonlu elemanlar çözümlemeleri bilinen ve oturmuş yöntemler kullanılarak yapılmıştır. Zemin-yapı-etkileşimi probleminin analizinde bir takım yakınsama problemleri yaşanmasına rağmen, geliştirilen algoritma dinamik zemin davranışı problemlerinin incelenmesinde yararlı olduğunu ortaya koymuştur. Bu algoritma etkileşim problemlerinin yanısıra yalnızca dinamik zemin davranışından oluşan konuların araştırılması için de kullanılabilir.

Anahtar kelimeler: sonlu elemanlar yöntemi, zemin plastisitesi, zemin-kazık-yapı etkileşimi

CONTENTS

Page

APPENDICES

CHAPTER ONE
INTRODUCTION

1.1 Seismic Soil-Pile-Structure Interaction

The development of analysis methods for seismic-soil-pile-structure interaction has been primarily driven by offshore oil production and the nuclear power industries. The research studies, which started with the needs of these sectors, continued in a more dedicated manner following observed damaged pile foundations due to liquefaction problems in large scale earthquakes such as 1964 Alaskan Earthquake and 1995 Kobe Earthquake (Mizuno, 1987; Hamada and O'Rourke, 1992; Ross et al., 1973; Seed and Idriss, 1967; Kagawa and Kraft, 1981). The degree of damages to pile foundations in these earthquakes surprised many engineers since the civil engineers believed that piles that were designed properly against axial loading would also perform well during earthquakes. This belief may have come from the misconception that the movements of foundation soils during earthquakes are not so large and that piles move essentially with foundation soils without receiving excessive soil resistance or loading (Kagawa and Kraft, 1980).

Seismic soil-structure interaction for structures supported by shallow foundations has been studied since 1970's. For this type of soil-structure interaction (i.e. structures with shallow foundations), the superstructure is excited by the seismic waves at the foundation level. However, in case of soil-pile-structure interaction, seismic waves excite the pile at all levels until they arrive at the foundation pad. This means, in SSPSI, the response of the superstructure depends on how piles respond to soil movements both on the foundation level and also along the pile length (Kagawa, 1983). Hence, it can be stated that seismic soil-pile-structure interaction (SSPSI) problem involves two mechanisms: kinematic interaction and inertial interaction.

The kinematic interaction corresponds to the interplay between soil and pile due to the deformations of the soil solely. It represents seismic effects on piles in the absence of the superstructure. During seismic excitation, stiffer structural foundation cannot conform to the distortions of the soil and incompatibility between the pile and

soil arises inevitably resulting in the generation of bending and shear forces, which reach to their maximum at sections where the stiffness contrast between the pile and the soil profile is highest. The geometry of the foundation and the travel path of the waves across the soil-structure interface for ideally rigid structures control the degree of kinematic interaction. For example, kinematic effects are not pronounced for the foundations that are located on the ground surface with zero embedment depth and subjected to vertically propagating shear waves.

The inertial interaction, on the other hand, is due to the super structural dynamic loads being generated as a result of pile head level seismic excitation. During inertial interaction, relative motions of the foundation with respect to the surrounding soil generate the inertial interaction. An idealized soil-pile-superstructure system under seismic excitation is given in Figure 1.1 where m1, mf and k1 stand for the structural mass, foundation mass and equivalent stiffness of the structure.

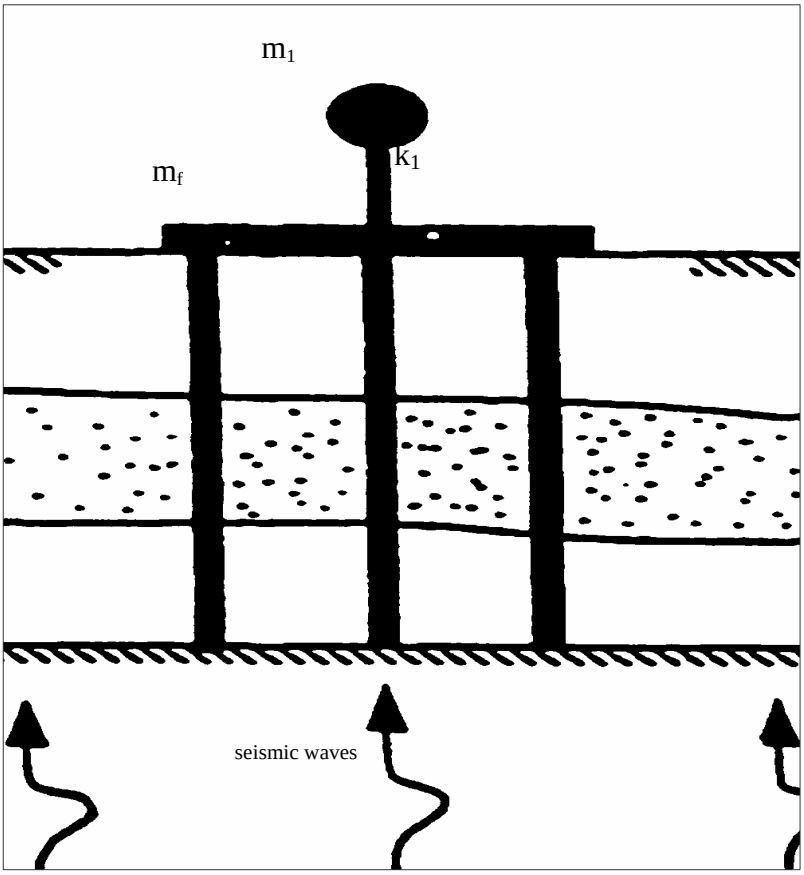


Figure 1.1 Soil-pile-structure system under seismic excitation

1.2 Piles in Liquefied Soils

It is a known fact that pile foundations perform better than shallow foundations under seismic loads in soils with low bearing capacity. However, it is possible to find a lot of cases in the literature in which the pile foundations failed due to the poor understanding of kinematic and inertial effects. Mizuno (1987) reported cases encountered in the earthquakes between 1923-1983 in Japan. Some of the cases involved damaged piles in liquefied soils. Accordingly, basic sources of damage in pile foundation in liquefied soils can be reported as follows:

a) Forces transferred from the superstructure (inertial effects)
b) Kinematic forces acting on pile foundations due to shear deformation of soils under seismic excitation
c) Lateral spreading of soils occurring while the pile foundation is in passive condition

Under seismic loading conditions, the effective stress state of the soil around the pile with which the stiffness of the soil is closely related has a major effect on the magnitudes of inertial and kinematic forces. Liquefaction of saturated cohesionless soils is a major phenomenon frequently encountered in previous large-scale earthquakes. During liquefaction, effective stress in the soil around the pile reduces down to zero due to the increase in pore water pressure leading to the decrease in soil strength and stiffness. This takes place, however, in loose to medium dense sands, which tend to contract under dynamic loading. Saturated dense sands, on the other hand, dilate when sheared resulting in negative excess pore water pressure generation. Therefore, the difference between saturated loose to medium dense and dense cohesionless soils is that, in dense sands, effective stress does not stay at zero level even if it reduces down to this value at a discrete time during the earthquake, and the soil does not liquefy. Such soils, however, exhibit stiffness degradation and produce plastic strains, which might be called as cyclic mobility instead of liquefaction.

Loose to medium dense sands cause loss of lateral support to dynamically loaded piles when they liquefy during earthquakes or they load the pile in passive mode as they flow and demonstrate permanent deformations at the ground surface. In some cases both mechanisms would take place at the same time. Damages near ground surface are caused by either inertial or lateral spreading forces (Figure 1.2) whereas kinematical interaction effects generate damages well below the surface since inertial forces are in general damped out at such depths (Figure 1.3). Negative influence of loss of lateral soil support to pile foundation may also be noticed in Figure 1.3.

Figure 1.2 Pile damage near ground surface (Kobe Earthquake, 1995)

It may be stated that realistic design of pile foundations in cohesionless soils against seismic loads requires definition of effective stress state of the soil around the pile since soil deformation and strength characteristics both depend on effective stress. This, in turn, necessitates determination of excess pore water pressure due to kinematic and inertial interaction effects, which is a complex problem of soil-pile-structure interaction area.

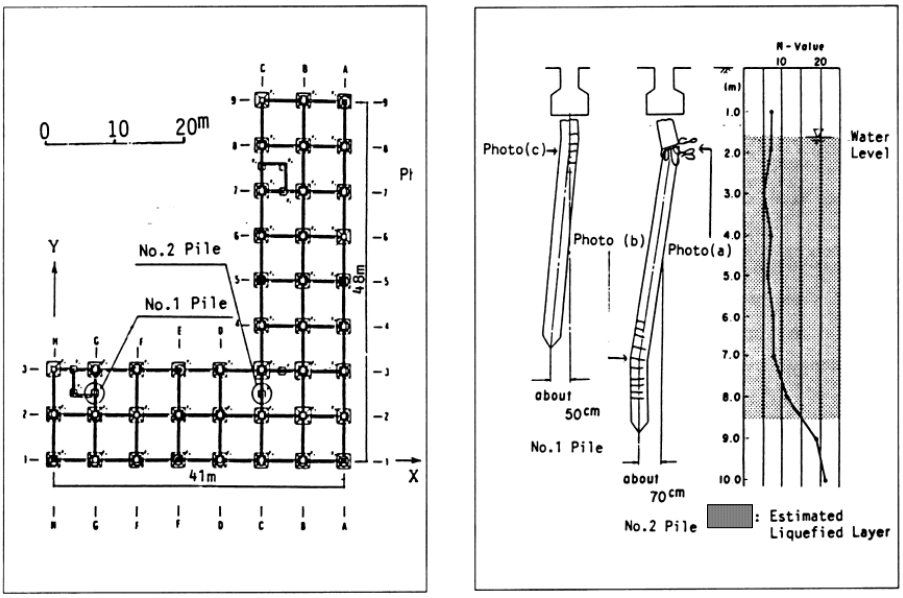


Figure 1.3 Effect of loss of lateral soil support on pile response (Hamada, 1992)

1.3 Goal of the Study

It has been shown in past studies that excess pore water pressure around a laterally deforming pile is under the control of kinematical and inertial interplay between soil and pile (Kagawa, 1981; Ashour and Norris, 1998). Kinematical effects on excess pore water pressure field in the immediate neighborhood of the pile, however, are negligible compared with the free field soil response. Therefore, it is reasonable to pay more attention to the influence of inertial interaction on excess pore water pressure generation and distribution within a zone of 16 to 20 pile radii in diameter. Numerical and experimental studies (Ozden and Kagawa, 2000) demonstrated that inertial mechanism alone has been capable of producing pore water pressure that would control soil resistance to the laterally displacing pile. Analytical and numerical models proposed to handle this mechanism do not cover many aspects of the inertially generated excess pore water pressure such as influence of cross sectional shape of the pile, loading frequency and group interaction. The problem demands both experimental and numerical work from the researchers.

This dissertation, however, aims to study influence of inertial interaction on excess pore water generation using a preferred constitutive model, which has been

coded into an FEM algorithm and advanced enough to simulate both contractive and dilative soil behavior.

The main goal has been to develop a specific finite element code that would enable study of the several aspects of this phenomenon such as extent of the zone where excess pore water pressure generation is significant, influence of pile cross sectional shape and loading frequency as well as the wave form, and group interaction for a single pile undergoing undrained cyclic loading in saturated cohesionless soil.

1.4 Organization of the Dissertation

This dissertation consists of eight chapters in addition to the introduction. The statement of the problem is made in the second chapter where the essence of the numerical model has been introduced to the reader. The third chapter covers governing differential equations for dynamic behavior of the porous medium (i.e. cohesionless soil). The following chapter is about the finite element discretization of the governing equations. In the fifth chapter, verification of the finite element code with several examples is presented. The sixth chapter is devoted to the explanation of the soil constitutive model. General characteristics of the finite element code developed during this study with the results of numerical experiments and their discussions are presented in the seventh, which is followed by the final chapter of conclusions and recommendations. The print outs of the FEM code are given in the Appendix B.

CHAPTER TWO
STATEMENT OF THE PROBLEM

2.1 Statement of the Problem

In the previous chapter, it has been stated that, both the liquefaction and cyclic mobility mechanisms in saturated cohesionless soils occur due to the change in effective stress of the soil which is caused by the increase or decrease of pore water pressure under seismic loading conditions. As a result of the changes in effective stresses, strength and stiffness of the soil are altered as well. So, this means, the resistance to deformation at any point in the soil deposit is a function of effective stress, which in turn depends on the contemporaneous rates of generation and dissipation of pore water pressure. Both mechanisms, sometimes in sequence, take place under dynamic loading. Liquefaction and cyclic mobility of saturated sands causes poor performance of pile foundations under inertial and kinematic effects. In soil-pile interaction, the controlling parameter is the effective stress state of the soil in the vicinity of the pile which changes according to the generation or dissipation of pore water pressure. Therefore, pore pressure development mechanism is the key factor on pile response in saturated soils under dynamic loads.

It is expected that the effective stress state of the soil near the pile is different from that of the free field. This is natural because soil-pile interaction effects which are inertial and kinematic forces are effective in the close neighborhood of the pile. So, pore pressure build up occurs mostly due to these inertial and kinematic effects near the pile. Mostly, inertial loads are effective up to 15-pile diameter depth along the pile and kinematic loads are effective at deeper depths. It is expected that pile response is governed by the inertial excess pore water pressure build-up since pile deformations at deeper elevations are not large enough to alter free field excess pore pressure field. Consequently, degradation of soil-pile springs at those depths is governed by pore pressure accumulated in the free field.

FEM modeling of both soil and the pile under seismic loading is an expensive way of estimating pile response since this approach requires utilization of three-dimensional solid finite elements (Trochanis, 1988). Among approximate methods such as boundary element (Sanchez, 1983; Kaynia and Kausel, 1982), and plain strain (Novak, 1974) solutions, nonlinear soil response is preferably modeled using accordingly established soil-pile springs (Kagawa and Kraft, 1980; Wang and Reese, 1998; Ashour and Norris, 1998). Modeling of soil response using nonlinear springs may be considered as an extension of the well-known Winkler foundation approach to dynamic method is more feasible and enables finer modeling of the pile foundation and superstructure using finite elements.

The researcher, however, has to employ one of the more elaborate methods that are briefly mentioned above in order to gain insight regarding influence of inertial interaction effects so that a procedure would be established to determine nonlinear soil-pile springs, which are defined in such a manner that the model of the soil resistance to the laterally deforming pile is coupled with an appropriate pore pressure generation-dissipation model (Kagawa and Kraft, 1981; Özden, 1999; Finn et al., 1977). It can be said without too much hesitation that preferred method of numerical analysis for the investigation of dynamic soil-pile-structure interaction (SPSI) is the nonlinear dynamic finite element method.

2.2 Scope of the Dissertation

In order to comply with the above-mentioned requirements of the research on inertial interaction influence on excess pore water pressure development in the immediate neighborhood of the pile of a dynamic SPSI problem, a dynamic nonlinear finite element code that would operate in the time domain has been planned. The focus of the research has been on the investigation of the stiffness and strength change in saturated cohesionless soil around the pile due to the changes in effective stress state occurring as a result of pore pressure generation originating from the dynamic displacement of the pile. It has been assumed that fully undrained

conditions are valid within the soil body (i.e. dissipation and redistribution has not been taken into consideration).

The finite element program considers a plain strain rigid section of a pile subject to dynamic lateral head loading. This kind of modeling consideration is reasonable since at a effective overburden stress the section of the pile may be considered as rigid with respect to the soil if the problem is solved for displacement boundary conditions assigned along the perimeter of the pile section. Such assumptions were previously made during model tests (Ozden, 1999) and numerical analysis studies (Iai, 2002).

For instance the geometry and loading considerations of the problem to be investigated by the FEM algorithm are similar to the design considerations of the model pile section (MPS) testing system of Ozden (1999). Therefore, numerical simulation and verification of the MPS testing system data would be possible with the development of the algorithm. A brief introduction of the design considerations and the capabilities of the MPS testing system would give more information about the boundary conditions for the FEM algorithm (Ozden, 1999):

a) A short rigid pile placed in a test chamber filled with saturated loose sand, was used in MPS tests. For practical purposes, a small portion of the laterally deforming pile at a particular depth may be considered as infinitely rigid and vertical. So, it was reasonable to represent that portion with a relatively short and stiff vertical model pile section.
b) The rigid pile is moved with respect to the sand specimen in the test chamber to simulate pile-head loading conditions. This pile head dynamic excitation in the form of displacement is applied at different sinusoidal and triangular waveforms having different frequencies and displacement amplitudes. Hence, MPS test can be called as a displacement controlled test. The MPS system does not allow the generation of free-field strains inside the test specimen and is only suitable for the investigation of pile-head loading conditions and inertial effects.

c) The simulation of field effective stress levels can be achieved with the MPS testing system.

d) The MPS testing system was designed in such a way that the excess pore water pressure development at and near the pile section surface was detected with the transducers placed along the vertical length of the pile and on the horizontal level at a certain distance from the pile. Additionally, simultaneous recording of soil resistance-pile displacement data can be made during the tests.

e) The goal of the development of the model pile section (MPS) testing system was to study soil-pile interaction in loose submerged cohesionless soils. With this system, effects of excess pore water pressure build up on the load-deformation behavior of laterally loaded pile can be investigated. Soil-pile interactions, loading rate and effective overburden stress have a great effect on the development of excess pore water pressure. Additionally, effects of soil damping on load-deformation behavior can be studied with the production of real time cyclic hysteresis loops.

The purpose of developing a dynamic nonlinear finite element code is to analyze inertial soil-pile interaction for the above-mentioned boundary conditions with an emphasis on excess pore water pressure development. Proposed FEM model can also be illustrated by the help of Figure 2.1. The plan view of the meshed area is shown in Figure 2.2. Boundary conditions of the system are specified as in the following:

a) Nodes lying on the outermost circle are constrained against displacements along x and y-axes.

b) The rigid pile section behavior is represented through application of sinusoidal displacement boundary conditions to the global X direction degree of freedoms of the nodes on the perimeter of the pile while the global Y direction degree of freedoms of these nodes are constrained against movements in this direction. That is, the rigid pile section makes cyclic displacements in the global X direction. If the sign of sinusoidal prescribed displacements is positive, this means the pile moves in the positive global X direction to the right generating compressive strains (having

negative sign) on the right hand and extension strains (having positive sign) on the lefthand soil medium.

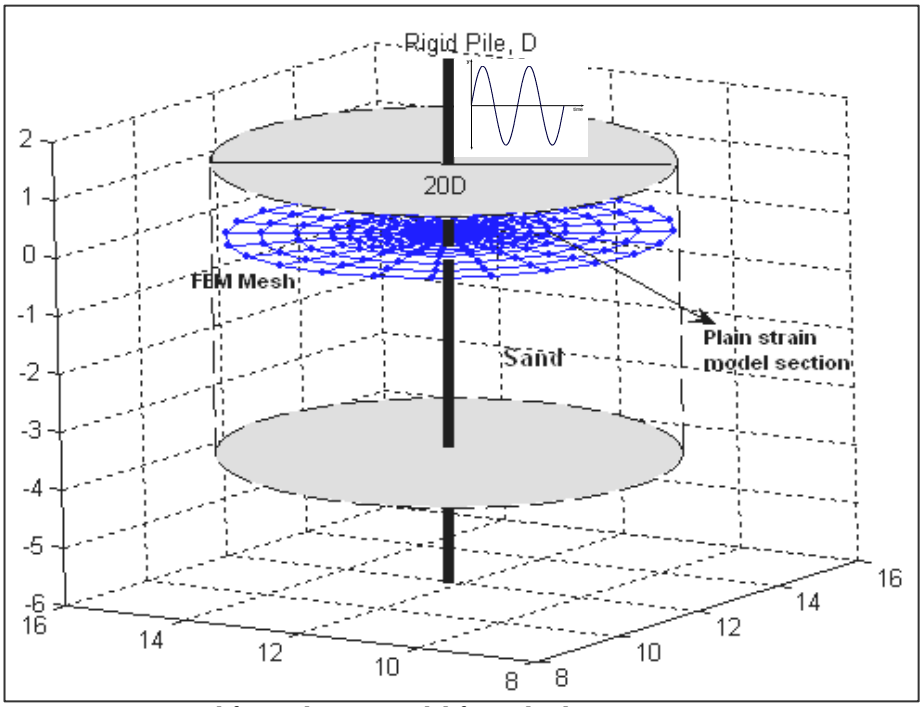


Figure 2.1 Proposed finite element model for soil-pile interaction

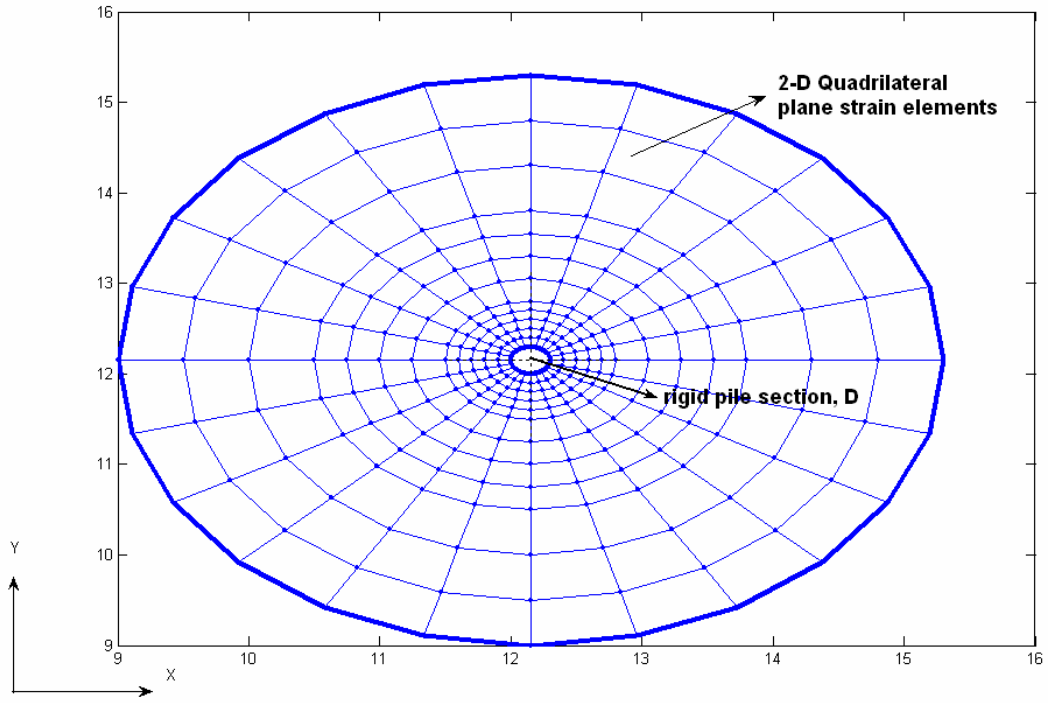


Figure 2.2 FEM mesh of a plain strain section

c) Initial effective stress state of the soil medium is represented by K0 conditions which is equal to vertical effective stresses at the depth of exploration multiplied by lateral earth pressure coefficient at rest (K0=1-sin∅).

d) Constraint 1: When the rigid pile section moves, the soil medium in front of the pile attains passive state of stress while, the soil medium in the back is in the active state of stress. It is a known fact that failure due to active lateral stresses occur earlier than the failure due to passive lateral stresses. In other words, greater amount of shear stress should be mobilized for failure due to passive lateral stresses than the amount needed for the failure due to active lateral stresses. Hence, it is natural for the shear strength to be greater for the passive case than the active case as shown in Figure 3. When failure occurs in an element in the active part, the stiffness parameters of this element stay the same until it is loaded in the passive state. Physically, failure of the element in the active part means, the sand medium follows the pile.

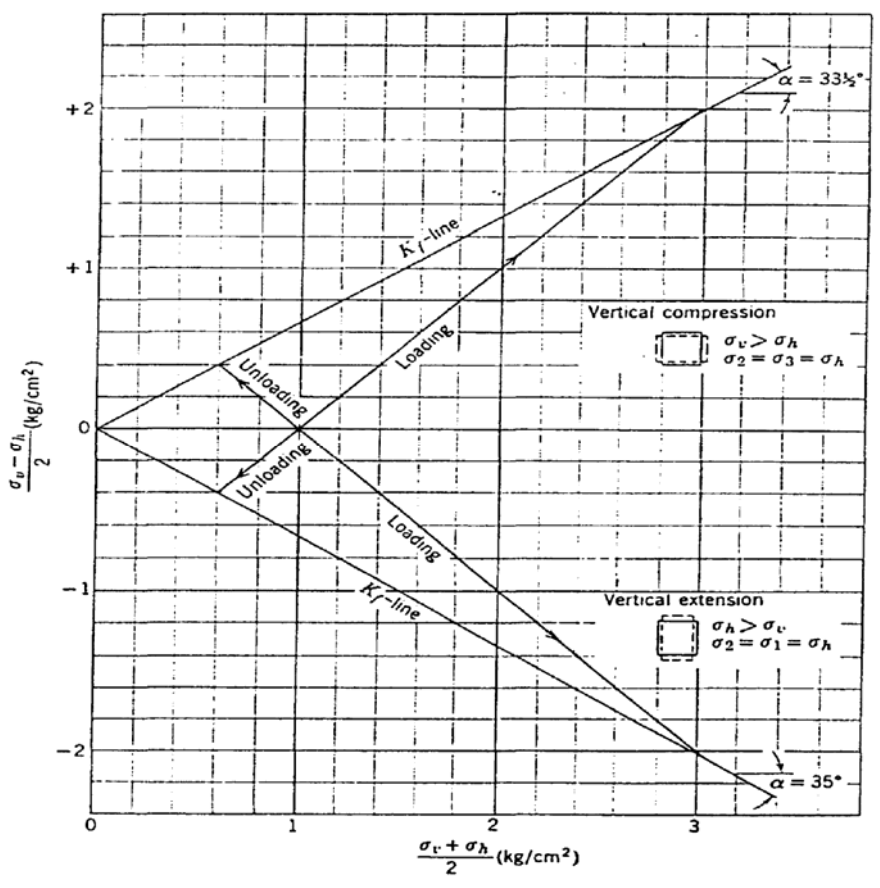


Figure 2.3 Stress paths for triaxial tests (Lambe and Whitman, 1969)

e) Constraint 2: In addition, in sands, pore water pressure should not be less than -1 atm. If the pore pressure falls below this limit, the pore water will cavitate. When it cavitates, the sand medium will no longer remain at constant volume even though it is undrained.

The 2D strain space plasticity constitutive model (Iai et al., 1990 and 1992) to be employed in the FEM algorithm allows use of two dimensional plane strain elements. The version utilized in this study is appropriate for undrained loading conditions. Iai soil constitutive model enables determination of the mean effective stress state of the sand medium through shear work correlations. This procedure also allows computation of pore water pressure accumulation. Once the change in effective stress state is obtained, strength parameters, stiffness and stress state of the system can be updated for the next time step.

CHAPTER THREE

GOVERNING DIFFERENTIAL EQUATIONS FOR DYNAMIC POROUS MEDIUM

3.1 Introduction

All researchers dealing with soil mechanics accept that the concept of effective stress is essential for the interpreting of real soil behaviour. However, practical engineering calculations are traditionally based on total stress approaches. When the researchers like Zienkiewicz, started applying numerical procedures in soil mechanics and soil dynamics, they have seen that a realistic prediction of the behaviour of the soil masses could only be achieved if the total stress approaches were abandoned. However, engineers continue to use the methods based on total stress for earthquake response analysis, often introduced with the linear approximations which can sometimes lead to erroneous results. Hence the correct approach was to consider the coupled interaction of the soil skeleton and the pore fluid parts. Especially, the liquefaction and cyclic mobility in saturated sands under seismic loading conditions can only be explained by considering this two-phase action and a sophisticated computation procedure. So, simple limit methods applied in statics are no longer useful.

Zienkiewicz et al (1999) stated that, since 1975, large number of researchers have studied on establishing the foundations of numerical predictions and approximations based on concepts introduced by Biot (1941). In the following sections brief documentation of these relations are introduced.

3.2 Coupled Systems

In practice, it is too common to see two or more physical systems interacting with each other. In this case, solution of one system cannot be independent of the simultaneous solution of the other system. Such systems are known as coupled systems. The coupling may be weak or strong depending on the degree of interaction.

The definition of coupled systems may be generalized to include a wide range of problems and their numerical discretisation as:

Coupled systems and formulations are those applicable to multiple domains and dependent variables which usually (but not always) describe different physical phenomena and in which

a) neither domain can be solved while seperated from each other;
b) neither set of dependent variables can be explicitly eliminated at the differential equation level (Zienkiewicz and Taylor, 1991).

In coupled systems, the solution of any single system is a well-posed problem and is possible when the variables corresponding to the other system are prescribed.
Coupled systems can be classified in two categories:

Class 1: This class contains problems in which coupling occurs on domain interfaces via the boundary conditions imposed there. Generally the domainsdescribe different physical situations but it is possible to considercoupling between domains that are physically similar in which different discretisation processes have been used. Fluid-structure interaction problems are good examples for this class of coupled systems.

Class 2: This class contains problems in which the various domains overlap (totally or partially). Here the coupling occurs through the differential governing equations describing different physical phenomena. Examples for this category may be listed as:

a) Metal extrusion where the plastic flow is strongly coupled with the temperature field while at the same time the latter is influenced by the heat generated in the plastic flow.

b) Earthquake response of a dam in which seepage flow and pressures interact with the dynamic behaviour of the soil skeleton (Zienkiewicz and Taylor, 1991).

3.3 Derivation of the Governing Differential Equations for the Saturated Porous Media

Current interest in geomechanics is focused on transient phenomena occuring in earthquakes, wave loading and consolidation. For all of these, the coupling between the deformation of the solid skeleton of the soil or rock and the motion of the pore fluid is of primary importance.

The basic equations of motion for porous media were first established for quasi-static phenomena in 1941 by Biot who then extended them to dynamics. Later on Biot's original equations were rederived with some minor modifications and even some rederivations were obtained. For example, mixture theory introduced in 1957 by Trusdell and improved by many others provided a new basis for such coupled equations. However, no rational change was achieved compared to the ones obtained by Biot.

Later on, owing to increasing interest in non-linear applications, a generalised incremental form was derived by Zienkiewicz et al, Prevost and others in which large strain and nonlinear material behavior are included (Zienkiewicz and Shiomi,1984).

In the following sections, stress and strain concepts as defined for soils will be given and then the basic differential equations governing the statics and dynamics of a deforming porous medium will be presented. As stated in the preceding paragraphs, all the governing equations root from Biot's theory. The basic assumptions of this theory regarding the formulations which will be presented for the porous medium can be stated as follows:

- The solid phase is a porous skeleton of particles surrounded by one or more fluids.

- The shear stresses in the fluid phase are small . Fluid phase applies an all-round pressure on the solid phase.
- Small strain theory is considered and so Darcy's law is assumed valid in terms of absolute fluid velocity.

Another important point to be stated is that, the governing equations presented in the following sections are derived at the macroscobic level.(Lewis,Schrefler,1987). The microscobic level of expressing the phenomena is out of the targets of this thesis.

3.3.1 Stresses and Strains In a Soil Mass

Figure 3.1 shows a two dimensional element of soil of which the sides are parallel with the coordinate axes. This element may be considered as homogeneous because the soil element is taken to be large enough compared to the size of the pores. In addition, the same soil element can be taken as infinitesimal because it is small enough compared to the macroscopic phenomena considered (Biot, 1940).

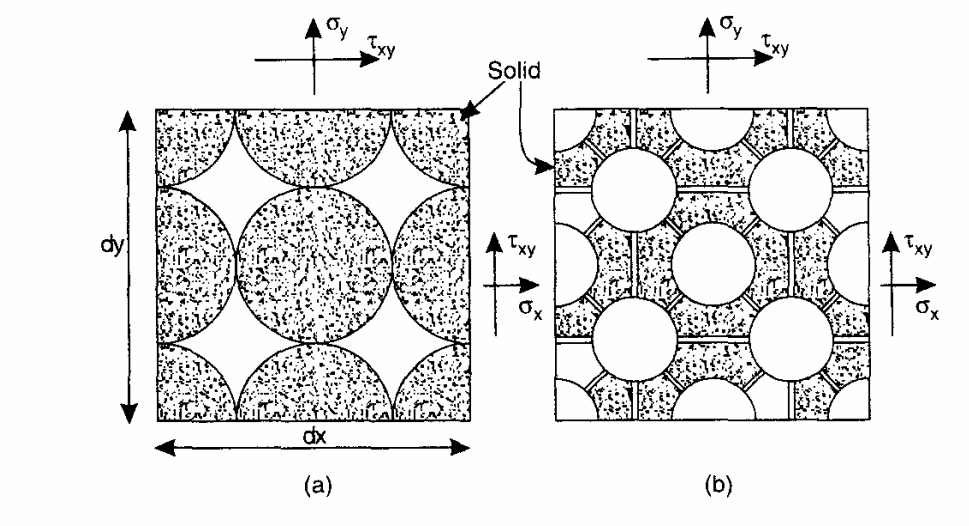


Figure 3.1 Various idealised structures of fluid saturated porous solids
a) a granular material b) a perforated solid with interconnecting voids.
(Zienkiewicsz, Chan, Pastor, Schrefler, Shiomi, 1999)

The average stress state of the two dimensional soil element under any kind of loading condition, either static or dynamic, is represented by the total stress vector . When the soil element is subject to a total stress of σ, some part of the total stress

will be carried by the soil skeleton (porous skeleton) itself and the rest of it will be carried by the pore water present in the pores of the soil skeleton. Contribution of stress carried by the soil skeleton is called the effective stress (σ′) which governs the major deformation of the porous skeleton. This deformation may be linear or nonlinear and the failure states can only be adequately expressed in terms of such effective stresses. The rest of the total stress is carried by the pore water pressure (p). Effective stress relation is expressed by (3.1). In this relation, negative sign is introduced in front of the pore pressure part because pressures are generally defined as being positive in compression, while it is convenient to define stress components as positive in tension.

$$\begin{aligned} \sigma &= \sigma' - \mathrm{p} \\ \sigma &= \sigma' - mp \end{aligned} \tag{3.1}$$

in vector form;

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \sigma_x' \\ \sigma_y' \\ \tau_{xy} \end{Bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} p \tag{3.2}$$

where;

$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}$: total stress vector

$\sigma' = \begin{Bmatrix} \sigma_x' \\ \sigma_y' \\ \tau_{xy} \end{Bmatrix}$: effective stress vector representing the stress carried by the solid skeleton.

$\mathrm{p} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} p$: represents the part of the total stress vector carried by pore water.

$[m] = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$: represents Kronecker delta. (m is equal to unity for the normal stress components and zero for the shear stress components. This is understandable due to the fact that pore water can only transmit normal stresses).

In the theory of water-saturated porous media, equations of motions are first established by considering stresses and forces which act on the solid portion and fluid portion of the two-phase media. However, these stresses and forces are different in definition from the effective stresses σ'_x, σ'_y, σ'_z, and the pore water pressure p, commonly used in the discipline of soil mechanics (as mentioned in the preceding paragraph). It is essential to show the relationships between them. The pore water pressure acting on the solid portion is (1-n)p and that on the fluid portion is np, where n is the porosity of soil. Therefore following equations can be written between total and effective stresses. These relationships are summarised in Table 3.1.

$$\begin{aligned} \sigma_x &= \sigma_x' + (1-n)p \\ \sigma_y &= \sigma_y' + (1-n)p \\ \sigma_z &= \sigma_z' + (1-n)p \\ \mathrm{P} &= np \\ \sigma_x^* &= \sigma_x + \mathrm{P} \\ \sigma_y^* &= \sigma_y + \mathrm{P} \\ \sigma_z^* &= \sigma_z + \mathrm{P} \end{aligned} \qquad (3.3)$$

Table 3.1. Relationships Between Two Stress Systems (Ishihara and Towhata, 1982)

	Effective Stress	Pore Water Pressure	Total
Stresses acting on the solid phase	σ′	(1-n)p	σ
Stresses acting on the fluid phase	0	np	P
Total	σ′	p	σ*

It is further possible to seperate the effective stress vector into deviatoric and volumetric components as follows:

$$\begin{Bmatrix} \sigma'_x \\ \sigma'_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \sigma'_{xd} \\ \sigma'_{yd} \\ \tau_{xy} \end{Bmatrix} + \begin{Bmatrix} \frac{\sigma'_x + \sigma'_y}{2} \\ \frac{\sigma'_x + \sigma'_y}{2} \\ 0 \end{Bmatrix} \qquad (3.4)$$

in which,

$\begin{Bmatrix} \sigma'_{xd} \\ \sigma'_{yd} \\ \tau_{xy} \end{Bmatrix}$:deviatoric stress component of the effective stress vector

$\begin{Bmatrix} \frac{\sigma'_x + \sigma'_y}{2} \\ \frac{\sigma'_x + \sigma'_y}{2} \\ 0 \end{Bmatrix}$:volumetric stress component of the effective stress vector.

$\frac{\sigma'_x + \sigma'_y}{2}$: mean effective stress

If the displacements of a two dimensional soil element are u and v, in the x and y directions respectively, then the total strains of the soil element are defined as follows:

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} \\ \varepsilon_y &= \frac{\partial v}{\partial y} \\ \gamma_{xy} &= \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \end{aligned} \qquad (3.5)$$

The total and incremental strain vectors are expressed as:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad , \qquad \{d\varepsilon\} = \begin{Bmatrix} d\varepsilon_x \\ d\varepsilon_y \\ d\gamma_{xy} \end{Bmatrix} \qquad (3.6)$$

Here, $\{\varepsilon\}$: total strain vector.

$\{d\varepsilon\}$: total strain vector (in incremental form).

ε_x, ε_y : strains in x and y directions.

γ_{xy} : shear strain in the x-y plane.

The total strain vector can be expressed in incremental form as follows:

$$\{d\varepsilon\} = \{d\varepsilon_{\sigma'}\} + \{d\varepsilon_c\} + \{d\varepsilon_0\} \tag{3.7}$$

Here,

$\{d\varepsilon\}$: incremental total strain vector

$\{d\varepsilon_{\sigma'}\}$: incremental strain vector component dependent on effective stresses

$\{d\varepsilon_c\}$: incremental strain vector component due to creep effects

$\{d\varepsilon_0\}$: incremental strain vector component due to autogeneous volumetric strains

Autogeneous volumetric strains reflect the compaction of grain configuration due to cyclic loading in liquefaction phenomena. This volumetric strain vector has elastic and plastic parts too. Changes occuring in autogeneous volumetric strains reflect themselves in pore pressure changes. The most important problem is how to quantify this kind of strain. The solution is to follow the laboratory stress paths and connect them to the variables of the real problem.

$$\{d\varepsilon_{\sigma'}\} = \{d\varepsilon\} - \{d\varepsilon_c\} - \{d\varepsilon_0\} \tag{3.8}$$

$$\{d\sigma'\} = [D]\{d\varepsilon_{\sigma'}\} \tag{3.9}$$

$$\{d\sigma'\} = [D\ ](\{d\varepsilon\} - \{d\varepsilon_c\} - \{d\varepsilon_0\}) \tag{3.10}$$

D : tangent stres-strain matrix

Excluding the strain contribution for creep;

$$\{d\sigma'\} = [D](\{d\varepsilon\} - \{d\varepsilon_0\}) \tag{3.11}$$

is obtained. (3.8)-(3.11) states that increment in effective stresses are completely related with the changes in effective stress dependent strain vector (Zienkiewicz et al.,1999; Zienkiewicz et al.,1991).

3.3.2 Biot's Formulations for the Static and Dynamic Behaviour of Fully Saturated Porous Medium with a Single Pore Fluid (Water)

According to Biot's theory, the formulations governing the behaviour of the soil-pore fluid system can be stated as follows:

1. Overall equilibrium or momentum balance relation for the soil-fluid mixture.
2. Equation for the momentum balance of the fluid.
3. Equation for the mass balance of flow (flow conservation equation)

These governing equations with the addition of appropriate constitutive relations and the boundary conditions related to the nature of the problem ,define the behaviour of the soil-fluid system under both static and dynamic loading conditions. Biot formulations are defined in different forms for both saturated and partially saturated soil conditions and even,including more than one fluid, however, only the relations dealing with fully saturated behaviour with a single pore fluid (water) will be included in the context of this thesis.

The variables of this system are the displacements of the solid matrix (u), the velocity of the fluid flow (w) and the pressure of the fluid (p). So, the boundary conditions of the whole system can be defined in terms of these variables. In the derivation of the following governing differential equations, control volume of soil +pore fluid (dx.dy.dz) is assumed.

3.3.2.1 Overall equilibrium or momentum balance relation for the soil-fluid mixture

Overall equilibrium must exist between the total stress-tensor gradients, body forces and inertia forces. This relation is defined in tensor (3.12) and matrix (3.13) forms as follows:

$$\sigma_{ij,j} - \rho \ddot{u}_i - \underline{\rho_f [\dot{w}_i + w_j w_{i,j}]} + \rho b_i = 0 \qquad (3.12)$$

$$S^T \sigma - \rho \ddot{u} - \underline{\rho_f [\dot{w} + w \nabla^T w]} + \rho b = 0 \qquad (3.13)$$

In these equations, the underlined terms represent the fluid acceleration relative to the solid and convective terms of this acceleration. This acceleration is usually small and can be omitted.

Here (for two dimensional case);

$$[S] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \text{ : (3 x 2) matrix operator}$$

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \text{ : total stress vector for the soil-pore fluid system}$$

$$\{b\} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \text{ : vector of gravity accelerations (body force per unit mass)}$$

$$\begin{Bmatrix} \ddot{u} \end{Bmatrix} = \begin{Bmatrix} \ddot{u} \\ \ddot{v} \end{Bmatrix} \text{ : acceleration vector due to total displacements of the system}$$

$\left\{w\right\} = \begin{Bmatrix} w_x \\ w_y \end{Bmatrix}$: average (Darcy) velocity of the percolating water

$\left\{\dot{w}\right\} = \begin{Bmatrix} \dot{w}_x \\ \dot{w}_y \end{Bmatrix}$: acceleration vector due to the relative fluid displacements

$$\rho = n\rho_f + (1-n)\rho_s \tag{3.14}$$

ρ : density of the whole soil-fluid mixture

ρ_f : density of the fluid

ρ_s : density of the solid particles

3.3.2.2 Equation for the Momentum Balance of the Fluid (Equation of Fluid Flow)

The following governing equation is related with the momentum balance of the fluid. In the derivation of this equation, fluid phase is assumed to move with the solid phase and also the same control volume dx.dy.dz is used.

In quasi-static flow through a porous medium, the resistive forces due to the fluid viscosity balance the pressure gradient as follows:

$$-p_{,i} = R_i \text{ (tensoral form)} \tag{3.15}$$

$$-\nabla p = R \text{ (matrix form)} \tag{3.16}$$

Here;

R: viscous drag forces

P : pressure of the fluid (water)

Introducing Darcy's seepage law (3.17) and (3.18) into (3.15) or (3.16) in tensoral and matrix forms ;

$$k_{ij}R_j = w_i \text{ (tensoral form)} \tag{3.17}$$

$$kR = w \text{ (matrix form)} \tag{3.18}$$

and (3.19) is obtained for the quasi-static case.

$$-\nabla p = k^{-1} w \tag{3.19}$$

where;

k: permeability of the soil medium (length^{3}time/mass), but in these equations, its definition is different from the one in classical soil mechanics as:

$$k = \frac{k'}{\rho_f' g'} \tag{3.20}$$

k' : permeability in the soil mechanics convention (length/time)

ρ_f' : fluid density

g' : gravitational acceleration at which the permeability is measured.

Taking into consideration the contribution due to dynamic effects, (3.15) for tensoral form and (3.16) for matrix form;

$$-p_{,i} - R_i - \rho_f \ddot{u}_i - \underline{\rho_f [\dot{w}_i + w_j w_{i,j}]/n} + \rho_f b_i = 0 \tag{3.21}$$

$$-\nabla p - R - \rho_f \ddot{u} - \underline{\rho_f [\dot{w} + w \nabla^T w]/n} + \rho_f b = 0 \tag{3.22}$$

In these equations, the underlined terms represent the convective fluid acceleration and can be omitted as they are small.

3.3.2.3 Equation for the mass balance of flow (flow conservation equation)

Equation for the 'Mass balance of flow' states that the divergence of the flow velocity must be equal to the rate of decrease of the pore space and the fluid expansion rate. In other words, the flow divergence $w_{i,i}$ occuring in time dt is balanced by the storage in the pores of a unit volume of soil in that time increment. The components of this balance are:

- *the increased volume due to a change in strain:*

$$\delta_{ij} d\varepsilon_{ij} = d\varepsilon_{ii} = \mathrm{m}^{\mathrm{T}} \mathrm{d}\varepsilon \tag{3.23}$$

- *additional volume stored by compression of void fluid due to fluid pressure increase:*

$$\frac{ndp}{K_f} \tag{3.24}$$

- *additional volume stored by the compression of grains by the fluid pressure increase:*

$$\frac{(1-n)dp}{K_s} \tag{3.25}$$

- *the change in volume of the solid phase due to a change in the intergranular effective contact stress:*

$$\left(\sigma_{ij}^{'} = \sigma_{ij} + \delta_{ij} p\right): -\frac{1}{3}\delta_{ij}\frac{d\sigma_{ij}^{'}}{K_s} = -\frac{K_T}{K_s}\left(d\varepsilon_{ii} + \frac{dp}{K_s}\right) \tag{3.26}$$

In these equations,

K_f : Bulk modulus of the pore fluid

K_s : Bulk modulus of the solid phase

K_T : Average bulk modulus of the solid skeleton

Therefore, the flow conservation equation can be written as:

$$w_{i,i} + \dot{\varepsilon}_{ii} + \frac{n\dot{p}}{K_f} + \frac{(1-n)\dot{p}}{K_s} - \frac{K_T}{K_s}\left(\dot{\varepsilon}_{ii} + \frac{\dot{p}}{K_s}\right) + n\frac{\dot{\rho}_f}{\rho_f} + \dot{s}_0 = 0 \tag{3.27}$$

$$w_{i,i} + \alpha\dot{\varepsilon}_{ii} + \frac{\dot{p}}{Q} + n\frac{\dot{\rho}_f}{\rho_f} + \dot{s}_0 = 0 \tag{3.28}$$

$$\nabla^T \mathrm{w} + \alpha \mathrm{m}\dot{\varepsilon} + \frac{\dot{p}}{Q} + n\frac{\dot{\rho}_f}{\rho_f} + \dot{s}_0 = 0 \tag{3.29}$$

$$\frac{1}{Q} \equiv \frac{n}{K_f} + \frac{\alpha - n}{K_s} \cong \frac{n}{K_f} + \frac{1-n}{K_s} \tag{3.30}$$

Therefore, full set of relations defining the behaviour of saturated porous medium can be stated once more as follows:

1. Biot's formulations governing the behaviour of saturated porous medium
2. Appropriate constitutive relation defining the behaviour
3. Boundary conditions defined according to the nature of the problem. These boundary conditions are imposed on the variables u (displacement), p(pore pressure), w(velocity of the fluid flow).

(Zienkiewicz et al.,1999; Zienkiewicz et al.,1991)

3.3.2.4 Some Approximations to the Full Biot Theory

It is reasonable to use the full set of Biot's equations (3.13, 3.22 and 3.27) in numerical solution process because this set gives the exact results. However, the full set is suitable for explicit time integration procedures. In implicit time integration procedures, large equations systems arise and the size of these equations can be decreased by omitting the underlined terms which include the variable w (velocity of the fluid) (Zienkiewicz et al.,1999).

Full set of equations (including all the variables which are u, w and p):

$$S^T\sigma - \rho\ddot{u} - \underline{\rho_f[\dot{w} + w\nabla^T w]} + \rho b = 0$$

$$-\nabla p - R - \rho_f \ddot{u} - \underline{\rho_f[\dot{w} + w\nabla^T w]/n} + \rho_f b = 0 \qquad (3.31)$$

$$w_{i,i} + \dot{\varepsilon}_{ii} + \frac{n\dot{p}}{K_f} + \frac{(1-n)\dot{p}}{K_s} - \frac{K_T}{K_s}\left(\dot{\varepsilon}_{ii} + \frac{\dot{p}}{K_s}\right) + n\frac{\dot{\rho}_f}{\rho_f} + \dot{s}_0 = 0$$

u-p Approximation (for dynamics of lower frequencies, exact for consolidation):

Elimination of w can be achieved in the following manner:

1. Omitting the terms related with w in (3.13);

$$S^T\sigma - \rho\ddot{u} + \rho b = 0 \qquad (3.32a)$$

2. By coupling (3.22) and (3.27) , in addition using (3.18) and omitting the density changes;

$$\nabla^T \mathrm{k}(-\nabla p - \rho_f \ddot{\mathrm{u}} + \rho_f \mathrm{b}) + \alpha \mathrm{m}\dot{\varepsilon} + \frac{\dot{p}}{Q} + \dot{s} = 0 \qquad (3.32b)$$

In the discretisation process, contribution of the solid acceleration term is neglected in (3.32b) resulting in (3.32c), because it is revealed by Leung (Leung,1984) that inclusion of this term makes the system non-symmetric. Later on, the effects of this omission was investigated by Chan (Chan, 1988) and was found to be insignificant. The same researcher (Chan, 1995) included it in the force term in the FEM code SWANDYNE-II (Zienkiewicz et al.,1999).

$$\nabla^T \mathrm{k}(-\nabla p + \rho_f \mathrm{b}) + \alpha \mathrm{m}\dot{\varepsilon} + \frac{\dot{p}}{Q} + \dot{s} = 0 \qquad (3.32c)$$

u-p approximation is suitable for medium-speed phenomena. The inaccuracies of the u-p version are pronounced only in high-frequency, short-duration phenomena (Zienkiewicz et al.,1999).

.

Approximation for the Consolidation Phenomena (very slow phenomena):

If in the u-p approximation formulations (3.32a and 3.32b) , all the acceleration terms are neglected then the resulting system corresponds to the well-known, quasi-static, consolidation equation frequently used in soil mechanics (Zienkiewicz, et al., 1980).

$$S^T \sigma + \rho b = 0 \qquad (3.33a)$$

$$\nabla^T \mathrm{k}(-\nabla p + \rho_f \mathrm{b}) + \alpha \mathrm{m}\dot{\varepsilon} + \frac{\dot{p}}{Q} + \dot{s} = 0 \qquad (3.33b)$$

Undrained Approximation (w=0):

Another approximation to the full dynamic equation occurs if no relative movement of the fluid is permitted (fluid displacement=0 which means w=0 and also k=0) . The solution can be obtained either using the u-p variable system or by reduction to the variable u (solid displacement) alone providing that K_f (bulk modulus of the pore fluid) is finite. Such formulation is of a penalty type when numerical analysis is used. (Zienkiewicz, et al., 1980; Zienkiewicz. and Bettess, 1982; Zienkiewicz, et al., 1988). (3.34b) is obtained from (3.28) by assuming w=0.

$$S^T \sigma - \rho \ddot{u} + \rho b = 0 \qquad (3.34a)$$

$$dp = -Q\alpha d\varepsilon_{ii} \qquad (3.34b)$$

This formulation needs the tangent stress-strain matrix (D) to be modified as:

$$\overline{D}_{ijkl} = D_{ijkl} + \alpha^2 Q \delta_{ik} \delta_{jl} \qquad (3.34c)$$

in matrix form,

$$\overline{D} = D + \alpha^2 \mathrm{m} Q \mathrm{m}^T$$

Detailed information about the derivation and the numerical solution procedures of the full Biot equations and its approximations can be found in the literature. (Zienkiewicz et al.,1999; Zienkiewicz et al.,1991; Zienkiewicz et al.,1982; Zienkiewicz and Bettess,1982; Zienkiewicz, et al., 1988; Zienkiewicz and Shiomi, 1984; Zienkiewicz, et al., 1980) Literature in the area of saturated porous medium formulations and their numerical solution processes is so wide and new theories are being added to the preceding ones.

CHAPTER FOUR

FINITE ELEMENT DISCRETISATION OF THE GOVERNING EQUATIONS

4.1. Introduction

An exact theoretical solution for any engineering problem requires that equilibrium, compatibility, material behavior and boundary conditions (both force and displacement) must all be satisfied. Taking into account this requirement, current methods of analysis being used in the solution of geotechnical engineering problems can be grouped into the following categories as presented in Table 4.1. These categories simply are closed form, simple methods (limit equilibrium, stress field and limit analysis) and numerical methods (beam-spring approaches and full numerical analysis) (Potts and Zdravkovic,1999).

Table 4.1 Basic solution requirements satisfied by various methods (Potts, and Zdravkovic, 1999)

METHOD OF ANALYSIS		SOLUTION REQUIREMENTS				
		Equilibrium	Compatibility	Constitutive behaviour	Boundary conditions	
					Force	Disp
Closed form		S	S	Linear elastic	S	S
Limit equilibrium		S	NS	Rigid with a failure criterion	S	NS
Stress field		S	NS	Rigid with a failure criterion	S	NS
Limit analysis	Lower bound	S	NS	Ideal plasticity with associated flow rule	S	NS
	Upper bound	NS	S		NS	S
Beam-Spring approaches		S	S	Soil modelled by springs or elastic interaction factors	S	S
Full Numerical analysis		S	S	Any	S	S

S - Satisfied; NS - Not Satisfied

According to Table 4.1, it is not possible to obtain closed form analytical solutions incorporating realistic soil constitutive models which satisfy all four fundamental

requirements. The analytical solution methods (limit equilibrium, stress field methods and limit analysis methods) fail to satisfy at least one of the fundamental requirements. These approaches only give information on stability, they do not provide information on movements or structural forces under working load conditions. Simple numerical methods such as the beam-spring approach, can provide information on local stability and on wall movements and structural forces under working load conditions. Therefore, they are an improvement over the simpler analytical methods. However, they do not provide information on overall stability or on movements in the adjacent soil and the effects on adjacent structures. On the other hand, full numerical analysis (approaches based on finite difference and finite element methods) can provide information on all design requirements, so they provide the ultimate method of analysis. Especially the capability of full numerical methods to incoorporate any soil constitutive model is a very important feature of them. However, they require large amounts of computing resources and an experienced operator, but on the other hand they can solve incredibly difficult problems.

The finite element method is cosidered as a powerful numerical tool for the analysis of geotechnical problems. When dynamic loading conditions like earthquake, wave or traffic loads, etc are included, it becomes essential to consider (take into account) the complex nonlinear dynamic soil behavior. Additionally, soil-structure interaction effects contribute to the nonlinear behavior especially under dynamic loading conditions. As a matter of fact (honestly saying), most problems in geotechnical engineering include soil-structure interaction effects. The finite element method can incorporate complex soil constitutive models and boundary conditions that realistically simulate field conditions. Their ability to reflect field conditions essentially depends on the ability of the constitutive model to represent real soil behaviour and the correctness of the boundary conditions imposed (Potts and Zdravkovic, 1999).

4.2 Basic Steps of the Finite Element Method

The finite element method is well documented in the literature (Bathe, 1996; Huges, 2000; Zienkiewicz, 1991; Pots and Zdravkovic, 1999). Therefore, brief overview of the basic concepts will be presented as applied in the dynamic nonlinear soil-pile interaction code developed for this research. These steps are briefly listed as:

a) Differential Equations Governing the Problem (Element Equations)
b) Element Discretisation
c) Primary Variable Approximation
d) Construction of Element Matrices
e) Construction of Global Matrices
f) Solution Techniques for Nonlinear Analysis including the application of complex soil constitutive models
g) Determination of Suitable Time Integration Algorithm
h) Application of Boundary Conditions
i) Solution of Global Equations

The finite element method has a strong mathematical basis. However, when one goes deep into the method, it is interesting to see that this mathematical basis actually depends on a lot of approximations. Each step briefly discussed above, is an approximation itself. There are a lot of algorithms related with these steps in the literature.

Performance of finite element codes emerges as one of the problems when dynamic nonlinear analysis is performed. Especially the integration of nonlinear soil constitutive models requires a lot of iterations which affect the speed and performance of the FEM code. The remedy for this category of problems lies in having the knowledge of the theory behind these algorithms. Additionally, selection of the programming language and the efficient programming techniques is of great

importance. For example, parallel programming techniques both in the software and hardware basis are widely used.

4.3 **Finite Element Formulations (For Structural Problems)**

The finite element method models a continuous system as a discrete numerical system. The actual system is idealised as an assemblage of elements. To calculate the response of a body under external forces, the governing equations of equilibrium need to be established. An approach to express the equilibrium of a structural body is to use the principle of virtual displacements which asserts that for any small virtual displacement imposed on the body, the total internal virtual work must equal the total external virtual work.

The governing differential equations of a dynamic saturated porous medium were given in the previous chapter of this thesis. They are somehow different from the governing equations for structural bodies obtained by the application of the virtual work principle. The differences come from the additional body forces and inertial terms due to the motion of the fluid in the pores of the porous medium and the need for a governing equation representing the flow of the fluid. However, when discretised, they all reduce to the well known equation of motion, which is obtained by the application of the virtual work principle with minor changes in force terms.

The form of Iai's strain space plasticity soil constitutive model used in this research allows only the use of undrained loading conditions. Hence, undrained approximation of Biot's equations which are defined by Zienkiewicz were used in analysis. As there is no flow of water relative to the soil skeleton in undrained condition which means that the soil skeleton and the pore fluid move together, the only necessary governing equation is the equation of motion for overall equilibrium. In this case, the effect of the undrained condition is hidden in the stress-strain matrix.

Therefore, at this point, it will be helpful to present the finite element discretisation of the governing equation for a body obtained by the application of

virtual work principle. If Galerkin Method is applied on the governing equations for a dynamic porous medium, the discretised equation of motion will be obtained in the same format.

Application of the virtual work principle to obtain the equation of motion of a structural body and the discretisation process proceeds together after which the element matrices are obtained. Hence, selection of the type of the finite element which will be used in the analysis should be made. Here, all the formulations will be obtained for 2D quadrilateral element since they will be used in the construction of the FEM code.

4.3.1 2-D Quadrilateral Element

The 20·diameter wide mesh around the pile will be represented by four noded 2D quadrilateral elements which belong to the isoparametric family of elements. This axisymmetric mesh will be constructed along the depth of the pile at specified station points. This process will enable the FEM code to reflect the approximate three dimensional behavior of the model. Local and global coordinates for a two dimensional quadrilateral element is shown in Figure 4.1.

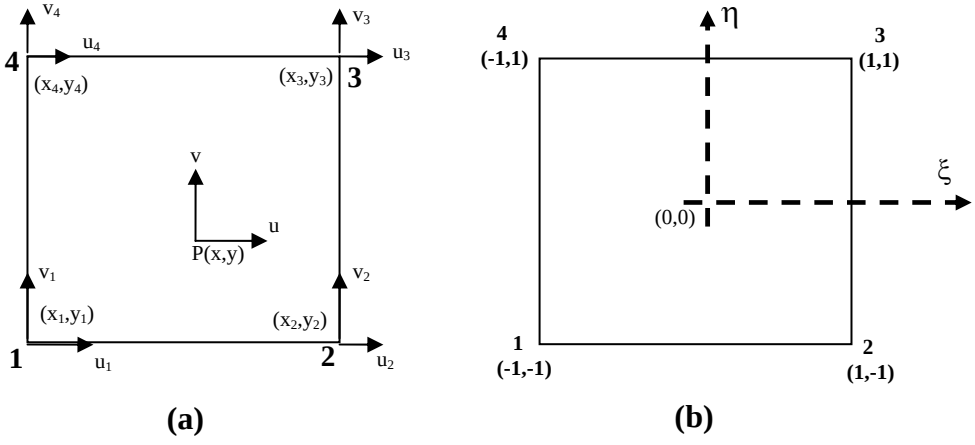


Figure 4.1 Four-noded quadrilateral continuum element(a) global coordinates and the displacement field, (b) local coordinates

In the above figure, x-y represents the global coordinate system and ξ-η represents the local coordinate system. The local nodes are numbered as 1,2,3 and 4 in a counterclockwise fashion as shown. (x_i,y_i) and (ξ_i,η_i) are the coordinates of each

node in the global and local coordinate systems respectively. The geometry and the displacement field are represented by the same shape functions which are Lagrange shape functions. Due to this fact, this group of elements belong to the isoparametric family of elements. Shape functions for each node are as follows:

$$N_i = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i)$$

Node 1, ξ_1= -1, η_1= -1 $\quad N_1 = \frac{1}{4}(1-\xi)(1-\eta)$

Node 2, ξ_2= 1,η_2= -1 $\quad N_2 = \frac{1}{4}(1+\xi)(1-\eta)$ (4.1)

Node 3, ξ_3= 1,η_3= 1 $\quad N_3 = \frac{1}{4}(1+\xi)(1+\eta)$

Node 4, ξ_4= 1,η_4= 1 $\quad N_4 = \frac{1}{4}(1-\xi)(1+\eta)$

The shape functions for each node are defined in local coordinates. It is possible to define the coordinates and displacement of any node inside the element by using these shape functions as follows:

4.3.1.1 Coordinates of any point inside the element

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 \\ y &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 \end{aligned} \quad (4.2)$$

In matrix form;

$$\{\mathbf{x}\} = [\mathbf{N}]\{\bar{\mathbf{x}}\} \quad (4.3)$$

$\{\mathbf{x}\} = \begin{Bmatrix} x \\ y \end{Bmatrix}$ vector of x and y coordinates of any node inside the element

$[\mathbf{N}] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}$ Matrix of shape functions

$$\left\{\bar{\mathbf{x}}\right\} = \begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{Bmatrix}$$ vector of nodal coordinates of the quadrialteral element

General coordinate definition can be expressed as:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{Bmatrix} \tag{4.4}$$

4.3.1.2 Displacement of any point inside the element

The displacements within each element are expressed as a function of nodal displacements through interpolation functions. Hence, using the same shape functions as in geometry interpolation, displacement field inside the element can be expressed as follows:

$$\begin{aligned} u &= N_1 u_1 + N_2 u_2 + N_3 u_3 + N_4 u_4 \\ v &= N_1 v_1 + N_2 v_2 + N_3 v_3 + N_4 v_4 \end{aligned} \tag{4.5}$$

$$\{\mathbf{u}\} = [\mathbf{N}]\left\{\bar{\mathbf{u}}\right\} \tag{4.6}$$

$\{\mathbf{u}\} = \begin{Bmatrix} u \\ v \end{Bmatrix}$ vector of displacements of any node inside the element

$$\left\{\bar{\mathbf{u}}\right\} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}$$ vector of nodal displacements of the quadrialteral element

General displacement field can be defined as:

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \tag{4.7}$$

4.3.1.3 Element Strains

Strain at a point within an element is a function of the derivative of the displacement. Displacement and strain distributions within an element are affected only by the displacements at the nodes of that element. The strain vector within an element can be expressed in terms of nodal displacements as:

$$\{\boldsymbol{\varepsilon}\} = [\mathbf{B}]\left\{\bar{\mathbf{u}}\right\} \tag{4.8}$$

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \qquad \varepsilon_x = \frac{\partial u}{\partial x} \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u1 \\ v1 \\ u2 \\ v2 \\ u3 \\ v3 \\ u4 \\ v4 \end{Bmatrix}$$

$$= \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix} \begin{Bmatrix} u1 \\ v1 \\ u2 \\ v2 \\ u3 \\ v3 \\ u4 \\ v4 \end{Bmatrix} \tag{4.9}$$

Displacements and strains inside an element can be expressed in incremental form which is essential for nonlinear analysis as follows:

$$d\left\{\mathbf{u}\right\} = [\mathbf{N}]d\left\{\bar{\mathbf{u}}\right\}$$
$$d\left\{\boldsymbol{\varepsilon}\right\} = [\mathbf{B}]d\left\{\bar{\mathbf{u}}\right\} \tag{4.10}$$

4.3.1.4 Element Stresses

Element total stres vector is expressed as:

$$\{\boldsymbol{\sigma}\} = [\mathbf{D}](\{\boldsymbol{\varepsilon}\} - \{\varepsilon_0\}) + \{\sigma_0\} \quad \text{for linear elastic behaviour} \tag{4.11}$$

$$\{\boldsymbol{\sigma}\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}$$

In the above,

- $\{\varepsilon_0\}$ represents the initial strains that the material within the element boundaries may be subjected to such as due to temperature changes, shrinkage.
- $\{\sigma_0\}$ represents the initial residual stresses that the body has been subject to.
- $[\mathbf{D}]$ elastic stress-strain matrix. In nonlinear behaviour this matrix changes at the end of each time step of dynamic loading.

The relation between stresses and strains can be expressed in incremental form for nonlinear behaviour as $d\{\sigma\} = [\mathbf{D}] d\{\varepsilon\}$.

4.3.1.5 Virtual Work Principle to Obtain the Element Matrices

To make the nodal forces statically equivalent to the actual boundary stresses and distributed loads, the simplest procedure is to impose an arbitrary (virtual) nodal displacement and to equate the external and internal work done by the various forces and stresses during that displacement. Let such a virtual displacement be ;

$$d\left\{\bar{u}\right\} = d\begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \quad \text{at the nodes} \tag{4.12}$$

and this virtual displacement at the nodes results in virtual disp and virtual strains within the element as follows:

$$d\left\{\mathbf{u}\right\} = [N] d\left\{\bar{u}\right\} \qquad \text{virtual displacements inside the element} \tag{4.13}$$

$d\{\boldsymbol{\varepsilon}\} = [B] d\left\{\bar{u}\right\}$ virtual strains inside the element

External Work done by the nodal forces = Internal Work done by internal stresses and distributed loads $\Rightarrow$

External Work done by the nodal forces= $d\left\{\bar{u}\right\}^T \{f\}$

$$= d\begin{bmatrix} u_1 & v_1 & u_2 & v_2 & u_3 & v_3 & u_4 & v_4 \end{bmatrix} \begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \\ f_{3x} \\ f_{3y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} \qquad (4.14)$$

Internal Work done by per unit volume by stresses and distributed forces =

$$d\{\boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\} - d\{\mathbf{u}\}^T \{\mathbf{b}\} - d\{\mathbf{u}\}^T \{\mathbf{t}\} \qquad (4.15)$$

Explanation for matrix operations:

$$d\{\boldsymbol{\varepsilon}\}^T = ([B] d\left\{\bar{u}\right\})^T = d\left\{\bar{u}\right\}^T [B]^T \qquad (4.16)$$

$$d\left\{\mathbf{u}\right\}^T = ([N] d\left\{\bar{u}\right\})^T = d\left\{\bar{u}\right\}^T [N]^T$$

$$= d\left\{\bar{u}\right\}^T [B]^T \{\boldsymbol{\sigma}\} - d\left\{\bar{u}\right\}^T [N]^T \{\mathbf{b}\} -$$

$$d\left\{\bar{u}\right\}^T [N]^T \{\mathbf{t}\} = d\left\{\bar{u}\right\}^T ([B]^T \{\boldsymbol{\sigma}\} - [N]^T \{\mathbf{b}\} - [N]^T \{\mathbf{t}\}) \qquad (4.17)$$

Equating the external work with the total internal work obtained by integrating over the volume of the element V:

$$d\left\{\bar{u}\right\}^T\{f\} = d\left\{\bar{u}\right\}^T\left(\int_V [B]^T\{\boldsymbol{\sigma}\}dV - \int_V [N]^T\{\mathbf{b}\}dV - \int_S [N]^T\{\mathbf{t}\}dS\right) \tag{4.18}$$

$$\{f\} = \int_V [B]^T\{\boldsymbol{\sigma}\}dV - \int_V [N]^T\{\mathbf{b}\}dV - \int_S [N]^T\{\mathbf{t}\}dS \tag{4.19}$$

$$\{f\} = \int_V [B]^T([\mathbf{D}](\{\boldsymbol{\varepsilon}\} - \{\varepsilon_0\}) + \{\sigma_0\})dV - \int_V [N]^T\left(\left\{\bar{b}\right\} - \rho\frac{\partial^2}{\partial t^2}\{\mathbf{u}\} - \mu\frac{\partial}{\partial t}\{\mathbf{u}\}\right)dV - \int_S [N]^T\{\mathbf{t}\}dS \tag{4.20}$$

$$\begin{aligned}&\{f\} = \int_V [B]^T[D]\{\boldsymbol{\varepsilon}\}dV - \int_V [B]^T\{\varepsilon_0\}dV + \int_V [B]^T\{\sigma_0\}dV - .....\\ &\int_V [N]^T\left\{\bar{b}\right\}dV + \int_V [N]^T\rho\frac{\partial^2}{\partial t^2}\{\mathbf{u}\}dV + \int_V [N]^T\mu\frac{\partial}{\partial t}\{\mathbf{u}\}dV - \int_S [N]^T\{\mathbf{t}\}dS\end{aligned} \tag{4.21}$$

$$\begin{aligned}&\int_V [B]^T[D]\{\boldsymbol{\varepsilon}\}dV + \int_V [N]^T\rho\frac{\partial^2}{\partial t^2}\{\mathbf{u}\}dV + \int_V [N]^T\mu\frac{\partial}{\partial t}\{\mathbf{u}\}dV = ...\\ &\{f\} + \int_V [B]^T\{\varepsilon_0\}dV - \int_V [B]^T\{\sigma_0\}dV + \int_V [N]^T\left\{\bar{b}\right\}dV + \int_S [N]^T\{\mathbf{t}\}dS\end{aligned} \tag{4.22}$$

$$\begin{aligned}&\int_V [B]^T[D][B]\left\{\bar{u}\right\}dV + \int_V [N]^T\rho\frac{\partial^2}{\partial t^2}[N]\left\{\bar{u}\right\}dV + \int_V [N]^T\mu\frac{\partial}{\partial t}[N]\left\{\bar{u}\right\}dV = ...\\ &\{f\} + \int_V [B]^T\{\varepsilon_0\}dV - \int_V [B]^T\{\sigma_0\}dV + \int_V [N]^T\left\{\bar{b}\right\}dV + \int_S [N]^T\{\mathbf{t}\}dS\end{aligned} \tag{4.23}$$

$$\begin{aligned}&\left(\int_V [B]^T[D][B]dV\right)\left\{\bar{u}\right\} + \left(\int_V [N]^T\rho[N]dV\right)\frac{\partial^2}{\partial t^2}\left\{\bar{u}\right\} + \left(\int_V [N]^T\mu[N]dV\right)\frac{\partial}{\partial t}\left\{\bar{u}\right\} = ...\\ &\{f\} + \int_V [B]^T\{\varepsilon_0\}dV - \int_V [B]^T\{\sigma_0\}dV + \int_V [N]^T\left\{\bar{b}\right\}dV + \int_S [N]^T\{\mathbf{t}\}dS\end{aligned} \tag{4.24}$$

$$
\begin{aligned}
[\mathbf{K}] &= \left(\int_V [B]^T [D][B] dV \right) \\
[\mathbf{C}] &= \left(\int_V [N]^T \mu [N] dV \right) \\
[\mathbf{M}] &= \left(\int_V [N]^T \rho [N] dV \right) \\
[\mathbf{F}] &= \{f\} + \int_V [B]^T \{\varepsilon_0\} dV - \int_V [B]^T \{\sigma_0\} dV + \int_V [N]^T \left\{ \bar{b} \right\} dV + \int_S [N]^T \{\mathbf{t}\} dS \\
[\mathbf{M}] \left\{ \ddot{u} \right\} &+ [\mathbf{C}] \left\{ \dot{u} \right\} + [\mathbf{K}] \{u\} = [\mathbf{F}]
\end{aligned}
\tag{4.25}
$$

4.4 Newmark's Time Integration Algorithm

This method proposed by Newmark is one of the mostpopular one-step methods and the recurrance formulas are as follows:

$$
\begin{aligned}
\{u_{t+1}\} &= \{u_t\} + \Delta t \left\{ \dot{u}_t \right\} + \left[\left(\frac{1}{2} - \beta_2 \right) \left\{ \ddot{u}_t \right\} + \beta_2 \left\{ \ddot{u}_{t+1} \right\} \right] \Delta t^2 : \\
\left\{ \dot{u}_{t+1} \right\} &= \left\{ \dot{u}_t \right\} + \left[(1 - \beta_1) \left\{ \ddot{u}_t \right\} + \beta_1 \left\{ \ddot{u}_{t+1} \right\} \right] \Delta t
\end{aligned}
\tag{4.26}
$$

Here,

$\{u_t\}, \left\{ \dot{u}_t \right\}, \left\{ \ddot{u}_t \right\}$: displacement, velocity and acceleration vectors at time t.

$\{u_{t+1}\}, \left\{ \dot{u}_{t+1} \right\}, \left\{ \ddot{u}_{t+1} \right\}$: displacement, velocity and acceleration vectors at time t+1.

Δt : time increment

β_1, β_2 : Newmark time integration parameters

$2\beta_2 >= \beta_1 = 0.5$ the method is unconditionally stable with second order accuracy.

$\beta_2 = 0.25$ and $\beta_1 = 0.5$ no numerical damping is introduced.

$\beta_1 > 0.5$ numerical damping is introduced but the accuracy is of first order.

In order to obtain a stable solution, the parameter β_2 should be selected by:

$$\beta_2 = \frac{1}{4}\left(\beta_1 + \frac{1}{2}\right)^2 \tag{4.27}$$

Other variations of this method are presented in Table 4.2:

Table 4.2 Parameters for Variations of the Newmark Method

β_1	β_2	**Method**
1/2	1/6	Linear Acceleration
1/2	¼	Average Acceleration, Trapezoidal Rule

$\beta_2 > 0.5$, $\beta 1 > 0.5$ is required for the unconditional stability (Zienkiewicz and Taylor, 1985; Chan, 1988). Usually $\beta_1 = 0.6$ and $\beta_2 = 0.3025$ are used for undrained analysis.

For an efficient solution process, elements of the equation of motion can be rearranged according to the known and unknown degrees of freedom :

$$\begin{bmatrix} M_{aa} & M_{ab} \\ M_{ba} & M_{bb} \end{bmatrix} \begin{Bmatrix} \ddot{u}_a \\ \ddot{u}_b \end{Bmatrix}_{t+1} + \begin{bmatrix} C_{aa} & C_{ab} \\ C_{ba} & C_{bb} \end{bmatrix} \begin{Bmatrix} \dot{u}_a \\ \dot{u}_b \end{Bmatrix}_{t+1} + \begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix} \begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{t+1} = \begin{Bmatrix} R_a \\ R_b \end{Bmatrix}_{t+1} \tag{4.28}$$

$$[M_{aa}]\{\ddot{u}_a\}_{t+1} + [C_{aa}]\{\dot{u}_a\}_{t+1} + [K_{aa}]\{u_a\}_{t+1} = \{R_a\}_{t+1} - [M_{ab}]\{\ddot{u}_b\}_{t+1} - [C_{ab}]\{\dot{u}_b\}_{t+1} - [K_{ab}]\{u_b\}_{t+1}$$

$$[M_{ba}]\{\ddot{u}_a\}_{t+1} + [M_{bb}]\{\ddot{u}_b\}_{t+1} + [C_{ba}]\{\dot{u}_a\}_{t+1} + [C_{bb}]\{\dot{u}_b\}_{t+1} + [K_{ba}]\{u_a\}_{t+1} + [K_{bb}]\{u_b\}_{t+1} = \{R_b\}_{t+1} \quad (4.29)$$

Here,

a: represents unknown degrees of freedom

b: represents known degrees of freedom

M_{aa}, C_{aa}, K_{aa}: Rearranged global mass, damping and stiffness matrices according to (unknown x unknown) degrees of freedom, respectively.

M_{bb}, C_{bb}, K_{bb}: Rearranged global mass, damping and stiffness matrices according to (known x known) degrees of freedom, respectively.

M_{ab}, C_{ab}, K_{ab}: Rearranged global mass, damping and stiffness matrices according to (unknown x known) degrees of freedom, respectively.

M_{ba}, C_{ba}, K_{ba}: Rearranged global mass, damping and stiffness matrices according to (known x unknown) degrees of freedom, respectively.

$\begin{Bmatrix} R_a \\ R_b \end{Bmatrix}_{t+1}$: Rearranged form of the global force vector. R_a represents the global force vector corresponding to the unknown degrees of freedom and R_b represents the global force vector corresponding to the known degrees of freedom (degrees of freedom representing prescribed displacements and support reactions.

$\begin{Bmatrix} \ddot{u}_a \\ \ddot{u}_b \end{Bmatrix}_{t+1}$: Rearranged form of the acceleration vector .

$\begin{Bmatrix} \dot{u}_a \\ \dot{u}_b \end{Bmatrix}_{t+1}$: Rearranged form of the velocity vector.

$\left\{ \begin{matrix} u_a \\ u_b \end{matrix} \right\}_{t+1}$: Rearranged form of the displacement vector.

In case of linear finite element analysis, (4.26) can be rearranged in the following forms:

1.st form (total formulation) :

$$\left\{ \ddot{u}_{t+1} \right\} = \frac{1}{\beta_2 \Delta t^2} \left\{ u_{t+1} \right\} - \frac{1}{\beta_2 \Delta t^2} \left\{ u_t \right\} - \frac{1}{\beta_2 \Delta t} \left\{ \dot{u}_t \right\} - \frac{0.5 - \beta_2}{\beta_2} \left\{ \ddot{u}_t \right\} \tag{4.30}$$

$$\left\{ \dot{u}_{t+1} \right\} = \left(1 - \frac{\beta_1}{\beta_2} \right) \left\{ \dot{u}_t \right\} - \frac{\beta_1}{\beta_2 \Delta t} \left\{ u_t \right\} + \frac{\beta_1}{\beta_2 \Delta t} \left\{ u_{t+1} \right\} + \left[(1 - \beta_1) \Delta t - \frac{\beta_1 \Delta t (0.5 - \beta_2)}{\beta_2} \right] \left\{ \ddot{u}_t \right\}$$

By using the equation of motion and (4.28) ,(4.29) and (4.30) ;

$$\begin{aligned} &\left(\frac{1}{\beta_2 \Delta t^2} [M_{aa}] + \frac{\beta_1}{\beta_2 \Delta t} [C_{aa}] + [K_{aa}] \right) \left\{ u_a \right\}_{t+1} = \left\{ R_a \right\}_{t+1} - [M_{ab}] \left\{ \ddot{u}_b \right\}_{t+1} - [C_{ab}] \left\{ \dot{u}_b \right\}_{t+1} - [K_{ab}] \left\{ u_b \right\}_{t+1} \\ &- \left(- \frac{1}{\beta_2 \Delta t^2} [M_{aa}] - \frac{\beta_1}{\beta_2 \Delta t} [C_{aa}] \right) \left\{ u_a \right\}_t - \left(- \frac{1}{\beta_2 \Delta t} [M_{aa}] + \left(1 - \frac{\beta_1}{\beta_2} \right) [C_{aa}] \right) \left\{ \dot{u}_a \right\}_t \\ &- \left(- \frac{0.5 - \beta_2}{\beta_2} [M_{aa}] + \left((1 - \beta_1) \Delta t - \frac{\beta_1 \Delta t (0.5 - \beta_2)}{\beta_2} \right) [C_{aa}] \right) \left\{ \ddot{u}_a \right\}_t \end{aligned} \tag{4.31}$$

$\left\{ u_a \right\}_{t+1}$ is solved from (4.31) and then replaced in (4.30) for $\left\{ \ddot{u}_a \right\}_{t+1}$ and $\left\{ \dot{u}_a \right\}_{t+1}$ and finally, by using (4.29) , support reactions and the forces corresponding to the prescribed degrees of freedom are evaluated.

2nd. Form (incremental formulation):

$$\left\{ u_{t+1} \right\} = \left\{ u_t \right\} + \left\{ \Delta u \right\}$$

$$\left\{ \ddot{u}_{t+1} \right\} = \frac{1}{\beta_2 \Delta t^2} \left\{ \Delta u \right\} - \frac{1}{\beta_2 \Delta t} \left\{ \dot{u}_t \right\} - \frac{0.5 - \beta_2}{\beta_2} \left\{ \ddot{u}_t \right\} \tag{4.32}$$

$$\left\{\dot{u}_{t+1}\right\}=\left(1-\frac{\beta_1}{\beta_2}\right)\left\{\dot{u}_t\right\}+\frac{\beta_1}{\beta_2\Delta t}\{\Delta u\}+\left[(1-\beta_1)\Delta t-\frac{\beta_1(0.5-\beta_2)\Delta t}{\beta_2}\right]\left\{\ddot{u}_t\right\}$$

$$=\left(1-\frac{\beta_1}{\beta_2}\right)\left\{\dot{u}_t\right\}+\frac{\beta_1}{\beta_2\Delta t}\{\Delta u\}+\Delta t\left(1-\frac{\beta_1}{2\beta_2}\right)\left\{\ddot{u}_t\right\}$$

Substituting (4.32) in (4.28);

$$\left(\frac{1}{\beta_2\Delta t^2}[M_{aa}]+\frac{\beta_1}{\beta_2\Delta t}[C_{aa}]+[K_{aa}]\right)\left\{\Delta u_a\right\}=\{R_a\}_{t+1}-[M_{ab}]\left\{\ddot{u}_b\right\}_{t+1}-[C_{ab}]\left\{\dot{u}_b\right\}_{t+1}-[K_{ab}]\left\{u_b\right\}_{t+1}$$
$$-[K_{aa}]\left\{u_a\right\}_t+\left(\frac{1}{\beta_2\Delta t}[M_{aa}]-\left(1-\frac{\beta_1}{\beta_2}\right)[C_{aa}]\right)\left\{\dot{u}_a\right\}_t$$
$$+\left(\frac{0.5-\beta_2}{\beta_2}[M_{aa}]-\left((1-\beta_1)\Delta t-\frac{\beta_1\Delta t(0.5-\beta_2)}{\beta_2}\right)[C_{aa}]\right)\left\{\ddot{u}_a\right\}_t \tag{4.33}$$

$\left\{\Delta u_a\right\}$ is solved from (4.33) and then replaced in (4.32) for $\left\{\ddot{u}_a\right\}_{t+1}$ and $\left\{\dot{u}_a\right\}_{t+1}$ and finally, by using (4.29) , support reactions and the forces corresponding to the prescribed degrees of freedom are evaluated.

4.5 Solution of Nonlinear Finite Element Equations

The philosophy of the solution of step-by-step nonlinear finite element equations is presented in Figure 4.2. According to this figure, iteration on displacement increments is performed until convergence criteria is reached after which the real displacement increment for that time step is obtained finally. This iteration is based on the minimisation of the residuals of the system.

According to Newton-Raphson method, the incremental formulations are rearranged in a way that iterations are possible within a time step in order to reach the final total increment value (total displacement increment) for that time step. For this reason, the the overall nature of the following formulations incremental+iterative. Equations (4.34), (4.35) and (4.36) represent the relations

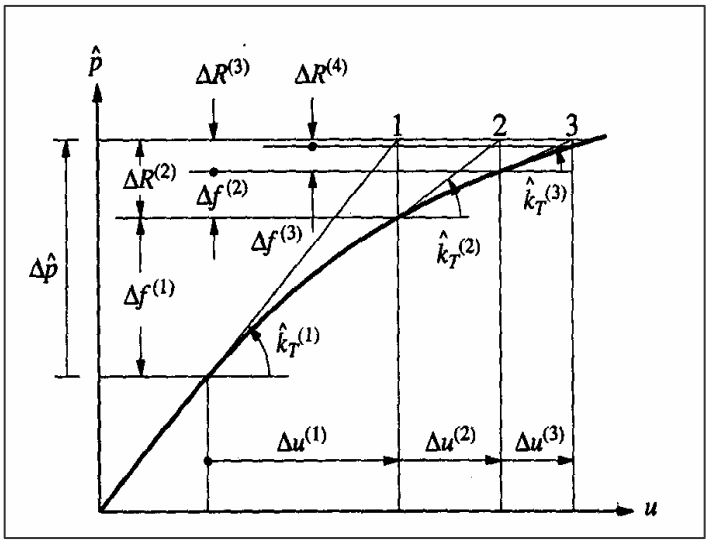


Figure 4.2 Iteration within a time step for nonlinear systems
Newton Raphson Iterations (Chopra, 1995)

between the total displacement increment for the time step, the total displacement at the end of the time step, the total displacement increment at the end of an iteration within the time step and the total displacement at the end of an iteration within the time step.

$$\begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t+1)}^{(k)} = \begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t+1)}^{(k-1)} + \begin{Bmatrix} \delta u_a \\ \delta u_b \end{Bmatrix}_{(t+1)}^{(k)} \tag{4.34}$$

Here,

$\begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t+1)}^{(k)}$: displacement vector at the end of an iteration within the time step.

$\begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t+1)}^{(k-1)}$: displacement vector at the end of the previous iteration within the time step.

$\begin{Bmatrix} \delta u_a \\ \delta u_b \end{Bmatrix}_{(t+1)}^{(k)}$: displacement increment vector corresponding to an iteration within the time step.

$$\begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t+1)} = \begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t)} + \sum_{j=1}^{k} \begin{Bmatrix} \delta u_a \\ \delta u_b \end{Bmatrix}_{(t+1)}^{(j)} = \begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t)} + \begin{Bmatrix} \Delta u_a \\ \Delta u_b \end{Bmatrix}_{(t+1)} \tag{4.35}$$

$$\begin{Bmatrix}\Delta u_a\\ \Delta u_b\end{Bmatrix}_{(t+1)} = \sum_{j=1}^{k}\begin{Bmatrix}\delta u_a\\ \delta u_b\end{Bmatrix}_{(t+1)}^{(j)} \tag{4.36}$$

Here,

$\begin{Bmatrix}u_a\\ u_b\end{Bmatrix}_{(t+1)}$: displacement vector at the end of the time step

$\begin{Bmatrix}u_a\\ u_b\end{Bmatrix}_{(t)}$: displacement vector at the end of the previous time step

$\begin{Bmatrix}\Delta u_a\\ \Delta u_b\end{Bmatrix}_{(t+1)} = \sum_{j=1}^{k}\begin{Bmatrix}\delta u_a\\ \delta u_b\end{Bmatrix}_{(t+1)}^{(j)}$: total displacement increment vector for the time step.

Equation of motion as defined in (4.28) is rearranged for the iterative process as shown in (4.37) and (4.38);

$$\begin{bmatrix}M_{aa} & M_{ab}\\ M_{ba} & M_{bb}\end{bmatrix}\begin{Bmatrix}\ddot{u}_a\\ \ddot{u}_b\end{Bmatrix}_{(t+1)}^{(k)} + \begin{bmatrix}C_{aa} & C_{ab}\\ C_{ba} & C_{bb}\end{bmatrix}\begin{Bmatrix}\dot{u}_a\\ \dot{u}_b\end{Bmatrix}_{(t+1)}^{(k)} + \begin{bmatrix}K_{aa} & K_{ab}\\ K_{ba} & K_{bb}\end{bmatrix}\begin{Bmatrix}u_a\\ u_b\end{Bmatrix}_{(t+1)}^{(k)} = \begin{Bmatrix}R_a\\ R_b\end{Bmatrix}_{(t+1)} \tag{4.37}$$

$$\begin{bmatrix}M_{aa} & M_{ab}\\ M_{ba} & M_{bb}\end{bmatrix}\left(\begin{Bmatrix}\ddot{u}_a\\ \ddot{u}_b\end{Bmatrix}_{(t+1)}^{(k-1)} + \begin{Bmatrix}\ddot{\delta u}_a\\ \ddot{\delta u}_b\end{Bmatrix}_{(t+1)}^{(k)}\right) + \begin{bmatrix}C_{aa} & C_{ab}\\ C_{ba} & C_{bb}\end{bmatrix}\left(\begin{Bmatrix}\dot{u}_a\\ \dot{u}_b\end{Bmatrix}_{(t+1)}^{(k-1)} + \begin{Bmatrix}\dot{\delta u}_a\\ \dot{\delta u}_b\end{Bmatrix}_{(t+1)}^{(k)}\right) + \begin{bmatrix}K_{aa} & K_{ab}\\ K_{ba} & K_{bb}\end{bmatrix}\left(\begin{Bmatrix}u_a\\ u_b\end{Bmatrix}_{(t+1)}^{(k-1)} + \begin{Bmatrix}\delta u_a\\ \delta u_b\end{Bmatrix}_{(t+1)}^{(k)}\right) = \begin{Bmatrix}R_a\\ R_b\end{Bmatrix}_{(t+1)} \tag{4.38}$$

Rearranging (4.38) results in (4.39) and (4.40):

$$\begin{aligned}&\begin{bmatrix}M_{aa} & M_{ab}\\ M_{ba} & M_{bb}\end{bmatrix}\begin{Bmatrix}\ddot{\delta u}_a\\ \ddot{\delta u}_b\end{Bmatrix}_{(t+1)}^{(k)} + \begin{bmatrix}C_{aa} & C_{ab}\\ C_{ba} & C_{bb}\end{bmatrix}\begin{Bmatrix}\dot{\delta u}_a\\ \dot{\delta u}_b\end{Bmatrix}_{(t+1)}^{(k)} + \begin{bmatrix}K_{aa} & K_{ab}\\ K_{ba} & K_{bb}\end{bmatrix}\begin{Bmatrix}\delta u_a\\ \delta u_b\end{Bmatrix}_{(t+1)}^{(k)} = \begin{Bmatrix}R_a\\ R_b\end{Bmatrix}_{(t+1)}\\ &-\begin{bmatrix}M_{aa} & M_{ab}\\ M_{ba} & M_{bb}\end{bmatrix}\begin{Bmatrix}\ddot{u}_a\\ \ddot{u}_b\end{Bmatrix}_{(t+1)}^{(k-1)} - \begin{bmatrix}C_{aa} & C_{ab}\\ C_{ba} & C_{bb}\end{bmatrix}\begin{Bmatrix}\dot{u}_a\\ \dot{u}_b\end{Bmatrix}_{(t+1)}^{(k-1)} - \begin{bmatrix}K_{aa} & K_{ab}\\ K_{ba} & K_{bb}\end{bmatrix}\begin{Bmatrix}u_a\\ u_b\end{Bmatrix}_{(t+1)}^{(k-1)}\end{aligned} \tag{4.39}$$

$$
\begin{aligned}
&[M_{aa}]\left\{\ddot{u}_a\right\}_{(t+1)}^{(k-1)}+[M_{ab}]\left\{\ddot{u}_b\right\}_{(t+1)}^{(k-1)}+[M_{aa}]\left\{\ddot{\delta u}_a\right\}_{(t+1)}^{(k)}+[M_{ab}]\left\{\ddot{\delta u}_b\right\}_{(t+1)}^{(k)}+\\
&[C_{aa}]\left\{\dot{u}_a\right\}_{(t+1)}^{(k-1)}+[C_{ab}]\left\{\dot{u}_b\right\}_{(t+1)}^{(k-1)}+[C_{aa}]\left\{\dot{\delta u}_a\right\}_{(t+1)}^{(k)}+[C_{ab}]\left\{\dot{\delta u}_b\right\}_{(t+1)}^{(k)}+\\
&[K_{aa}]\left\{u_a\right\}_{(t+1)}^{(k-1)}+[K_{ab}]\left\{u_b\right\}_{(t+1)}^{(k-1)}+[K_{aa}]\left\{\delta u_a\right\}_{(t+1)}^{(k)}+[K_{ab}]\left\{\delta u_b\right\}_{(t+1)}^{(k)}=\left\{R_a\right\}_{(t+1)}\\
&---------------------------------\\
&[M_{ba}]\left\{\ddot{u}_a\right\}_{(t+1)}^{(k-1)}+[M_{bb}]\left\{\ddot{u}_b\right\}_{(t+1)}^{(k-1)}+[M_{ba}]\left\{\ddot{\delta u}_a\right\}_{(t+1)}^{(k)}+[M_{bb}]\left\{\ddot{\delta u}_b\right\}_{(t+1)}^{(k)}+\\
&[C_{ba}]\left\{\dot{u}_a\right\}_{(t+1)}^{(k-1)}+[C_{bb}]\left\{\dot{u}_b\right\}_{(t+1)}^{(k-1)}+[C_{ba}]\left\{\dot{\delta u}_a\right\}_{(t+1)}^{(k)}+[C_{bb}]\left\{\dot{\delta u}_b\right\}_{(t+1)}^{(k)}+\\
&[K_{ba}]\left\{u_a\right\}_{(t+1)}^{(k-1)}+[K_{bb}]\left\{u_b\right\}_{(t+1)}^{(k-1)}+[K_{ba}]\left\{\delta u_a\right\}_{(t+1)}^{(k)}+[K_{bb}]\left\{\delta u_b\right\}_{(t+1)}^{(k)}=\left\{R_b\right\}_{(t+1)}
\end{aligned}
\tag{4.40}
$$

Introducing (4.41) into (4.40), equation set (4.42) can be obtained;

$$
\begin{aligned}
\left\{\ddot{u}_a\right\}_{(t+1)}^{(k)}&=\left\{\ddot{u}_a\right\}_{(t+1)}^{(k-1)}+\left\{\delta\ddot{u}_a\right\}_{(t+1)}^{(k)}\\
\left\{\dot{u}_a\right\}_{(t+1)}^{(k)}&=\left\{\dot{u}_a\right\}_{(t+1)}^{(k-1)}+\left\{\delta\dot{u}_a\right\}_{(t+1)}^{(k)}\\
\left\{u_a\right\}_{(t+1)}^{(k)}&=\left\{u_a\right\}_{(t+1)}^{(k-1)}+\left\{\delta u_a\right\}_{(t+1)}^{(k)}
\end{aligned}
\tag{4.41}
$$

$$
\begin{aligned}
\left\{\delta\ddot{u}_a\right\}_{(t+1)}^{(k)}&=\frac{1}{\beta_2\Delta t^2}\left\{\delta u_a\right\}_{(t+1)}^{(k)}\\
\left\{\delta\dot{u}_a\right\}_{(t+1)}^{(k)}&=\frac{\beta_1}{\beta_2\Delta t}\left\{\delta u_a\right\}_{(t+1)}^{(k)}
\end{aligned}
$$

$$
\begin{aligned}
&[M_{aa}]\left\{\ddot{\delta u}_a\right\}_{(t+1)}^{(k)} + [C_{aa}]\left\{\dot{\delta u}_a\right\}_{(t+1)}^{(k)} + [K_{aa}]\left\{\delta u_a\right\}_{(t+1)}^{(k)} = \{R_a\}_{(t+1)} \\
&-[M_{aa}]\left\{\ddot{u}_a\right\}_{(t+1)}^{(k-1)} - [C_{aa}]\left\{\dot{u}_a\right\}_{(t+1)}^{(k-1)} - [K_{aa}]\left\{u_a\right\}_{(t+1)}^{(k-1)} \\
&-[M_{ab}]\left\{\ddot{u}_b\right\}_{(t+1)}^{(k)} - [C_{ab}]\left\{\dot{u}_b\right\}_{(t+1)}^{(k)} - [K_{ab}]\left\{u_b\right\}_{(t+1)}^{(k)}
\end{aligned}
$$

$$
\begin{aligned}
&[M_{ba}]\left\{\ddot{\delta u}_a\right\}_{(t+1)}^{(k)} + [C_{ba}]\left\{\dot{\delta u}_a\right\}_{(t+1)}^{(k)} + [K_{ba}]\left\{\delta u_a\right\}_{(t+1)}^{(k)} = \{R_b\}_{(t+1)} \\
&-[M_{ba}]\left\{\ddot{u}_a\right\}_{(t+1)}^{(k-1)} - [C_{ba}]\left\{\dot{u}_a\right\}_{(t+1)}^{(k-1)} - [K_{ba}]\left\{u_a\right\}_{(t+1)}^{(k-1)} \\
&-[M_{bb}]\left\{\ddot{u}_b\right\}_{(t+1)}^{(k)} - [C_{bb}]\left\{\dot{u}_b\right\}_{(t+1)}^{(k)} - [K_{bb}]\left\{u_b\right\}_{(t+1)}^{(k)}
\end{aligned}
\tag{4.42}
$$

And finally, introducing (4.43) into (4.42) results in (4.44);

$$
\begin{aligned}
\left\{\ddot{u}_b\right\}_{(t+1)}^{(k)} &= \left\{\ddot{u}_b\right\}_{(t+1)} \\
\left\{\dot{u}_b\right\}_{(t+1)}^{(k)} &= \left\{\dot{u}_b\right\}_{(t+1)} \\
\left\{u_b\right\}_{(t+1)}^{(k)} &= \left\{u_b\right\}_{(t+1)}
\end{aligned}
\tag{4.43}
$$

Residuals and Reaction Forces:

$$
\{\Psi\}_{(t+1)}^{(k)} = \begin{bmatrix} M_{aa} & M_{ab} \\ M_{ba} & M_{bb} \end{bmatrix} \begin{Bmatrix} \ddot{u}_a \\ \ddot{u}_b \end{Bmatrix}_{(t+1)}^{(k)} + \begin{bmatrix} C_{aa} & C_{ab} \\ C_{ba} & C_{bb} \end{bmatrix} \begin{Bmatrix} \dot{u}_a \\ \dot{u}_b \end{Bmatrix}_{(t+1)}^{(k)} + \begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix} \begin{Bmatrix} u_a \\ u_b \end{Bmatrix}_{(t+1)}^{(k)} - \begin{Bmatrix} R_a \\ R_b \end{Bmatrix}_{(t+1)}
\tag{4.44}
$$

4.6 Finite Element Discretisation and Solution of the Governing Differential Equations (u-p Approximation) for Saturated Dynamic Porous Medium

In chapter 3, it was explained briefly under which circumstances, the full Biot equations can be approximated to u-p formulations. In the following sections, finite element discretisation and the solution of u-p formulations under dynamic loading conditions will be presented.

4.6.1 Finite Element Discretisation in Spatial Coordinates

Biot's governing differential equations for the dynamic porous medium was approximated in u-p form as shown in (4.45) and (4.46)

$$S^T \sigma - \rho \ddot{u} + \rho b = 0 \tag{4.45}$$

$$\nabla^T k(-\nabla p - \rho_f \ddot{u} + \rho_f b) + \alpha m \dot{\varepsilon} + \frac{\dot{p}}{Q} + \dot{s}_0 = 0 \tag{4.46}$$

In these formulations, the basic variables are the displacement of the soil-pore fluid medium(u) and the pressure of the fluid (p). Hence, variables u and p should be approximated by suitable shape functions at the beginning of the finite element discretisation process (discretisation of the displacement and the pore pressure fields).

$$\mathrm{u} = \sum_{k=1}^{n} N_k^u \bar{u}_k \tag{4.47}$$

$$\mathrm{p} = \sum_{k=1}^{m} N_k^p \bar{p}_k \tag{4.48}$$

In case of two dimensional quadrilateral elements, each node will have two degrees of freedom for displacement variable and one degree of freedom for pore pressure variable. For the discretisation of displacement and pore pressure fields, the same shape functions are used (4.49). This is not necessary, but especially in incompressible,undrained problems, the finite element equations are solvable only if $n_u >= n_p$, which means the parameters of the function defining the displacement field

should be greater or equal to the parameters of the function defining the pore pressure field (Zienkiewicz and Taylor,1991). Equations (4.47) and (4.48) can be expressed in matrix form as shown in (4.50) and (4.51).

$$N_i = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i)$$

Node 1, ξ_1= -1, η_1= -1 $\quad N_1^u = N_1^p = \frac{1}{4}(1-\xi)(1-\eta)$

Node 2, ξ_2= 1,η_2= -1 $\quad N_2^u = N_2^p = \frac{1}{4}(1+\xi)(1-\eta)$ (4.49)

Node 3, ξ_3= 1,η_3= 1 $\quad N_3^u = N_3^p = \frac{1}{4}(1+\xi)(1+\eta)$

Node 4, ξ_4= 1,η_4= 1 $\quad N_4^u = N_4^p = \frac{1}{4}(1-\xi)(1+\eta)$

$$\mathrm{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \mathrm{N}^u \bar{\mathrm{u}} = \begin{bmatrix} N_1^u & 0 & N_2^u & 0 & N_3^u & 0 & N_4^u & 0 \\ 0 & N_1^u & 0 & N_2^u & 0 & N_3^u & 0 & N_4^u \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ \bar{u}_2 \\ \bar{v}_2 \\ \bar{u}_3 \\ \bar{v}_3 \\ \bar{u}_4 \\ \bar{v}_4 \end{Bmatrix} \tag{4.50}$$

$$\mathrm{p} = \{p\} = \mathrm{N}^p \bar{p} = \begin{bmatrix} N_1^p & N_2^p & N_3^p & N_4^p \end{bmatrix} \begin{Bmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \bar{p}_3 \\ \bar{p}_4 \end{Bmatrix} \tag{4.51}$$

where;

$N_1^u, N_2^u, N_3^u, N_4^u$: displacement shape functions in terms of local coordinates.

$N_1^p, N_2^p, N_3^p, N_4^p$: pore pressure shape functions in terms of local coordinates.

$\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4$: displacement degrees of freedom in the X global direction

$\bar{v}_1, \bar{v}_2, \bar{v}_3, \bar{v}_4$: displacement degrees of freedom in the Y global direction

$\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4$: pore pressure degrees of freedom

$\begin{Bmatrix} u \\ v \end{Bmatrix}, \{p\}$: represents the displacement and the pore pressure at any node inside the element (4.52).

$$\begin{aligned} u &= N_1^u \bar{u}_1 + N_2^u \bar{u}_2 + N_3^u \bar{u}_3 + N_4^u \bar{u}_4 \\ v &= N_1^u \bar{v}_1 + N_2^u \bar{v}_2 + N_3^u \bar{v}_3 + N_4^u \bar{v}_4 \\ p &= N_1^p \bar{p}_1 + N_2^p \bar{p}_2 + N_3^p \bar{p}_3 + N_4^p \bar{p}_4 \end{aligned} \tag{4.52}$$

Applying the Galerkin procedure(method of weighted residuals) for the discretisation of (4.45) and (4.46) (Zienkiewicz and Taylor,1991; Zienkiewicz and Taylor,1989), Eqns. (4.53) and (4.54) are obtained. Galerkin procedure is applied on (4.45), by premultiplying it by N_u^T and then integrated by parts, resulting in (4.53) and finally (4.54).

$$\left[\int_\Omega (\mathrm{N}^u)^T \rho \mathrm{N}^u d\Omega\right]\ddot{\bar{\mathrm{u}}} + \left[\int_\Omega (\mathrm{N}^u)^T \mu \mathrm{N}^u d\Omega\right]\dot{\bar{\mathrm{u}}} + \int_\Omega \mathrm{B}^T \sigma d\Omega = \int_\Omega (\mathrm{N}^u)^T \rho \mathrm{b} d\Omega + \int_\Gamma (\mathrm{N}^u)^T \bar{t} d\Gamma \tag{4.53}$$

$$\mathrm{M}\ddot{\bar{\mathrm{u}}} + \mathrm{C}\dot{\bar{\mathrm{u}}} + \int_\Omega \mathrm{B}^T \sigma d\Omega = f^{(1)} \tag{4.54}$$

Where;

$\mathrm{M} = \int_\Omega (\mathrm{N}^u)^T \rho \mathrm{N}^u d\Omega$:mass matrix

$\mathrm{C} = \int_\Omega (\mathrm{N}^u)^T \mu \mathrm{N}^u d\Omega$:damping matrix

$$\mathrm{B} = \mathrm{SN}^{\mathrm{u}} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1^{\,u} & 0 & N_2^{\,u} & 0 & N_3^{\,u} & 0 & N_4^{\,u} & 0 \\ 0 & N_1^{\,u} & 0 & N_2^{\,u} & 0 & N_3^{\,u} & 0 & N_4^{\,u} \end{bmatrix}$$

$\int_{\Omega} \mathrm{B}^T \sigma d\Omega$:force vector due to internal stresses

$f^{(1)} = \int_{\Omega} (\mathrm{N}^u)^T \rho \mathrm{b} d\Omega + \int_{\Gamma} (\mathrm{N}^u)^T \bar{t} d\Gamma$:force term

$\int_{\Omega} (\mathrm{N}^u)^T \rho \mathrm{b} d\Omega$: force term due to body forces

$\int_{\Omega} (\mathrm{N}^u)^T \bar{t} d\Gamma$:force term due to prescribed boundary tractions

Furthermore, the term involving the stress can be split into two parts. Remembering the relation for effective stress:

$$\sigma = \sigma^{'} - \chi_w \mathrm{m} p \tag{4.55}$$

$$\mathrm{m} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

in matrix form,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \sigma_x^{'} \\ \sigma_y^{'} \\ \tau_{xy} \end{Bmatrix} - \chi_w \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} p \tag{4.56}$$

$$\int_{\Omega} \mathrm{B}^T \sigma d\Omega \equiv \int_{\Omega} \mathrm{B}^T \sigma^{'} d\Omega - \int_{\Omega} \mathrm{B}^T \chi_{\mathrm{w}} \mathrm{m} p d\Omega \tag{4.57}$$

Introducing (4.57) into (4.54), (4.58) is obtained;

$$\mathrm{M}\ddot{\mathrm{u}} + \mathrm{C}\dot{\mathrm{u}} + \int_{\Omega} B^T \sigma^{'} d\Omega - \left(\int_{\Omega} \mathrm{B}^T \chi_{\mathrm{w}} \mathrm{m} \mathrm{N}^p d\Omega \right) \bar{p} = f^{(1)} \tag{4.58}$$

Where;

$\int_{\Omega} \mathrm{B}^{\mathrm{T}} \sigma' d\Omega$: force vector due to effective stresses (internal)

If linear analysis is performed, then (4.58) becomes:

$$\mathrm{M}\ddot{\bar{u}} + \mathrm{C}\dot{\bar{u}} + \mathrm{K}\bar{u} - \mathrm{Q}\bar{p} = f^{(1)}$$

where;

$\mathrm{K} = \left(\int_{\Omega} \mathrm{B}^{T} D \mathrm{B} d\Omega \right)$:stiffness matrix, $\mathrm{K}\bar{u}$: force vector due to effective stresses

$\mathrm{Q} = \left(\int_{\Omega} \mathrm{B}^{T} \chi_{\mathrm{w}} \mathrm{m} \mathrm{N}^{p} d\Omega \right)$:coupling matrix

D : stress-strain matrix (constitutive matrix)

Applying the Galerkin procedure to (4.46) by premultiplied by $\mathrm{N_p}^{\mathrm{T}}$ and then integrating by parts gives (4.59) and finally results in (4.60).

$$\left(\int_{\Omega} \mathrm{N}^{\mathrm{p}T} \chi_{\mathrm{w}} \mathrm{m} \mathrm{B} d\Omega \right) \dot{\bar{\mathrm{u}}} + \left(\int_{\Omega} \left(\nabla \mathrm{N}^{\mathrm{p}} \right)^{T} k \nabla \mathrm{N}^{\mathrm{p}} d\Omega \right) \bar{p} + \left(\int_{\Omega} (\mathrm{N}^{\mathrm{p}})^{T} \frac{1}{Q^{*}} \mathrm{N}^{\mathrm{p}} d\Omega \right) \dot{\bar{\mathrm{p}}} - \left(- \int_{\Omega} \left(\mathrm{N}^{\mathrm{p}} \right)^{T} \nabla^{T} (\mathrm{k} S_{w} \rho_{f} \mathrm{b}) \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{w}}} \left(\mathrm{N}^{\mathrm{p}} \right)^{T} \bar{q} d\Gamma \right) = 0 \tag{4.59}$$

$$\mathrm{Q}^{\mathrm{T}} \dot{\bar{\mathrm{u}}} + \mathrm{H}\bar{\mathrm{p}} - \bar{\mathrm{S}}\dot{\bar{p}} = f^{(2)} \tag{4.60}$$

where;

$\mathrm{Q}^{T} = \left(\int_{\Omega} \mathrm{B}^{\mathrm{T}} \chi_{\mathrm{w}} \mathrm{m} \mathrm{N}^{\mathrm{p}} d\Omega \right)^{T} = \int_{\Omega} \mathrm{N}^{\mathrm{p}T} \chi_{\mathrm{w}} \mathrm{m} \mathrm{B} d\Omega$: transpose of the coupling matrix

$\mathrm{H} = \int_{\Omega} \left(\nabla \mathrm{N}^{\mathrm{p}} \right)^{T} k \nabla \mathrm{N}^{\mathrm{p}} d\Omega$: Permeability matrix

$\bar{\mathrm{S}} = \int_{\Omega} (\mathrm{N}^{\mathrm{p}})^{T} \frac{1}{Q^{*}} \mathrm{N}^{\mathrm{p}} d\Omega$: Compressibility matrix

$f^{(2)} = -\int_{\Omega} \left(\mathrm{N}^{\mathrm{p}} \right)^{T} \nabla^{T} (\mathrm{k} S_{w} \rho_{f} \mathrm{b}) \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{w}}} \left(\mathrm{N}^{\mathrm{p}} \right)^{T} \bar{q} d\Gamma$: force vector

$$\frac{1}{Q^*} \equiv C_s + \frac{nS_w}{K_f} + \frac{(\alpha - n)S_w}{K_s} \tag{4.61}$$

C_s, S_w, C_w, k depend on p.

S_w: parameter for saturation

K_f : Bulk modulus of the pore fluid

K_s : Bulk modulus of the solid phase

K_T : Average bulk modulus of the solid skeleton

n: porosity

$\alpha \approx 1$

Coupled equation set (4.58) and (4.60) can also be rearranged in the following form for linear material behaviour;

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u} \\ \ddot{p} \end{Bmatrix} + \begin{bmatrix} C & 0 \\ Q^T & S \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{p} \end{Bmatrix} + \begin{bmatrix} K & -Q \\ 0 & H \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} = \begin{Bmatrix} f^1 \\ f^2 \end{Bmatrix} \tag{4.62}$$

(Zienkiewicz et al, 1999, Zienkiewicz and Taylor,1991, Zienkiewicz et al.,1991;Zienkiewicz and Taylor, 1985)

4.6.2 Discretisation of the Coupled System in the Time Domain

Time discretisation of the coupled systems is usually achieved by using either multi-step methods or the single-step methods. However, multistep methods (Wood,1990) are usually inconvenient because they are not self statrting and a procedure needs to be established to handle this part. On the other hand, single step methods handle each step seperately and no problem occurs in the algorithm for restart. Two families of single-step methods are widely used in the literature for the solution of the coupled systems which are;

1. *SSpj Algorithm* (Single-Step p th. order scheme for j th. order differential equation) This algorithm was introduced by Zienkiewicz et al and then extended by Wood (Zienkiewicz et al.,1999).

2. *GNpj Algorithm*. This algorithm was first established by Newmark and then renamed as Beta-m Method by Katona and finally GNpj algorithm by Katona and Zienkiewicz (Zienkiewicz et al.,1999).

Both of these methods have similar stability characteristics. SSpj algorithm needs no initial conditions, on the other hand, for the GNpj algorithm, all quantities are defined at a discrete time station, so the transfer of these quantities between the two equations (coupled system) is easier to handle. In the context of this thesis, GNpj algorithm will be introduced briefly. GN11 algorithm (4.63) will be used for the pore pressure part and the GN22 algorithm (4.64) will be used for the displacement part. GN22 algorithm is exactly the same as Newmark time integration algorithm (4.26) but expressed in the form of acceleration increments.

$$\begin{aligned} \dot{\overline{p}}_{t+1} &= \dot{\overline{p}}_t + \Delta \dot{\overline{p}}_t \\ \overline{p}_{t+1} &= \overline{p}_t + \dot{\overline{p}}_t \, \Delta t + \overline{\beta}_1 \Delta \dot{\overline{p}}_t \, \Delta t \end{aligned} \tag{4.63}$$

$$\begin{aligned} \ddot{\overline{u}}_{t+1} &= \ddot{\overline{u}}_t + \Delta \ddot{\overline{u}}_t \\ \dot{\overline{u}}_{t+1} &= \dot{\overline{u}}_t + \ddot{\overline{u}}_t \, \Delta t + \beta_1 \Delta \ddot{\overline{u}}_t \, \Delta t \\ \overline{u}_{t+1} &= \overline{u}_t + \dot{\overline{u}}_t \, \Delta t + \frac{1}{2} \ddot{\overline{u}}_t \, \Delta t^2 + \frac{1}{2} \beta_2 \Delta \ddot{\overline{u}}_t \, \Delta t^2 \end{aligned} \tag{4.64}$$

For the unconditional stability of the recurrance scheme, time integration parameters ($\beta_1, \beta_2, \overline{\beta}_1$) are usually chosen in the following manner:

$$\beta_2 > \beta_1 \geq \frac{1}{2} \quad \text{and} \quad \overline{\beta}_1 \geq \frac{1}{2} \tag{4.65}$$

if $\beta_2 = \beta_1 = \frac{1}{2}$ and $\overline{\beta}_1 \geq \frac{1}{2}$ numerical oscillation may occur if no physical damping is present.

if $\begin{matrix} \beta_2 = 0.605 \;\; \beta_1 = 0.6 \;\; \overline{\beta}_1 = 0.6 \\ \beta_2 = 0.515 \;\; \beta_1 = 0.51 \;\; \overline{\beta}_1 = 0.51 \end{matrix}$ some numerical damping is introduced.

If the coupled equation system (4.58 and 4.60) is written for time (t+1), equation set (4.66) is obtained:

$$\begin{aligned} &M\ddot{\bar{u}}_{t+1}+C\dot{\bar{u}}_{t+1}+K\bar{u}_{t+1}-Q\bar{p}_{t+1}=f^{(1)}{}_{t+1} \\ &Q^{T}\dot{\bar{u}}_{t+1}+H\bar{p}_{t+1}-\bar{S}\dot{\bar{p}}_{t+1}=f^{(2)}{}_{t+1} \end{aligned} \tag{4.66}$$

Replacing the GN11 (4.63) and GN22 (4.64) algorithms in (4.66), (4.67) is obtained for the linear material behaviour:

$$\begin{bmatrix} M+C\beta_1\Delta t+\frac{1}{2}K_T\beta_2\Delta t^2 & -Q\overline{\beta_1}\Delta t \\ Q^T\beta_1\Delta t & S+H\beta_1\Delta t \end{bmatrix}\begin{Bmatrix} \Delta\ddot{\bar{u}}_t \\ \Delta\dot{\bar{p}}_t \end{Bmatrix}=\begin{Bmatrix} F^1_{t+1} \\ F^2_{t+1} \end{Bmatrix} \tag{4.67}$$

In order for the left hand side matrix to be symmetric, the second row should be multiplied by -1. Hence as $\beta_1=\overline{\beta_1}$, there will be no symmetry problem. If this modification is applied to (4.67) , the resulting system will be (4.68);

$$\begin{bmatrix} M+C\beta_1\Delta t+\frac{1}{2}K_T\beta_2\Delta t^2 & -Q\overline{\beta_1}\Delta t \\ -Q^T\beta_1\Delta t & -S-H\beta_1\Delta t \end{bmatrix}\begin{Bmatrix} \Delta\ddot{\bar{u}}_t \\ \Delta\dot{\bar{p}}_t \end{Bmatrix}=\begin{Bmatrix} F^1_{t+1} \\ F^2_{t+1} \end{Bmatrix} \tag{4.68}$$

$$\begin{aligned} F^1_{t+1} &= f^{1}{}_{t+1}-M\ddot{\bar{u}}_t+Q\bar{p}_t+Q\dot{\bar{p}}_t\,\Delta t \\ F^2_{t+1} &= -f^{2}{}_{t+1}+S\dot{\bar{p}}_t+H\dot{\bar{p}}_t\,\Delta t+H\bar{p}_t+Q^T\ddot{\bar{u}}_t\,\Delta t+Q^T\dot{\bar{u}}_t \end{aligned} \tag{4.69}$$

$$K_T=\int_{\Omega}B^T D_T B\,d\Omega \tag{4.70}$$

4.6.3 Solution of the Coupled Equation System for the Nonlinear Material Behaviour

For nonlinear material behaviour, the coupled equation system is written in the following form in which the effective stress contribution of the internal stress term in the first equation is left in integral form;

$$\mathrm{M}\ddot{\bar{\mathrm{u}}} + \mathrm{C}\dot{\bar{\mathrm{u}}} + \int_{\Omega} \mathrm{B}^{T} \sigma' d\Omega - \mathrm{Q}\bar{\mathrm{p}} = f^{(1)} \tag{4.71}$$

$$\mathrm{Q}^{\mathrm{T}}\dot{\bar{\mathrm{u}}} + \mathrm{H}\bar{\mathrm{p}} - \bar{\mathrm{S}}\dot{\bar{p}} = f^{(2)} \tag{4.72}$$

Writing (4.71) and (4.72) for time (t+1);

$$\mathrm{M}\ddot{\bar{\mathrm{u}}}_{t+1} + \mathrm{C}\dot{\bar{\mathrm{u}}}_{t+1} + (\int_{\Omega} \mathrm{B}^{T} \sigma'_{t+1} d\Omega) - \mathrm{Q}\bar{\mathrm{p}}_{t+1} = f^{(1)}{}_{t+1} \tag{4.73}$$

$$\mathrm{Q}^{\mathrm{T}}\dot{\bar{\mathrm{u}}}_{t+1} + \mathrm{H}\bar{\mathrm{p}}_{t+1} - \bar{\mathrm{S}}\dot{\bar{p}}_{t+1} = f^{(2)}{}_{t+1} \tag{4.74}$$

Replacing the GN11 and GN22 algorithms (4.73) ;

$$\begin{aligned} &\mathrm{M}\Delta\ddot{\bar{\mathrm{u}}}_{t} + \int_{\Omega} \mathrm{B}^{\mathrm{T}} \sigma'_{t+1} d\Omega - \mathrm{Q}\overline{\beta_1}\Delta\dot{\bar{\mathrm{p}}}_{t}\,\Delta t = \mathrm{f}^{1}{}_{t+1} - \mathrm{M}\ddot{\bar{\mathrm{u}}}_{t} + \mathrm{Q}\bar{\mathrm{p}}_{t} + \mathrm{Q}\dot{\bar{\mathrm{p}}}_{t}\,\Delta t \\ &\mathrm{F}^{1}_{t+1} = \mathrm{f}^{1}{}_{t+1} - \mathrm{M}\ddot{\bar{\mathrm{u}}}_{t} + \mathrm{Q}\bar{\mathrm{p}}_{t} + \mathrm{Q}\dot{\bar{\mathrm{p}}}_{t}\,\Delta t \\ &\mathrm{M}\Delta\ddot{\bar{\mathrm{u}}}_{t} + \int_{\Omega} \mathrm{B}^{\mathrm{T}} \sigma'_{t+1} d\Omega - \mathrm{Q}\overline{\beta_1}\Delta\dot{\bar{\mathrm{p}}}_{t}\,\Delta t - \mathrm{F}^{1}_{t+1} = 0 \end{aligned} \tag{4.75}$$

Internal effective stress term $\int_{\Omega} \mathrm{B}^{\mathrm{T}} \sigma'_{t+1} d\Omega$ can be expressed as :

$$P(\bar{\mathrm{u}}_{t+1}) = \int_{\Omega} \mathrm{B}^{\mathrm{T}} \sigma'_{t+1} d\Omega = \int_{\Omega} \mathrm{B}^{\mathrm{T}} \Delta\sigma'_{t} d\Omega + P(\bar{\mathrm{u}}_{t}) \tag{4.76}$$

Hence,

$$\mathrm{M}\Delta\ddot{\bar{\mathrm{u}}}_{t} + P(\bar{\mathrm{u}}_{t+1}) - \mathrm{Q}\overline{\beta_1}\Delta\dot{\bar{\mathrm{p}}}_{t}\,\Delta t - \mathrm{F}^{1}_{t+1} = 0 \tag{4.77}$$

Replacing the GN11 and GN22 algorithms in (4.74) and multiplying by -1 for symmetry ;

$$-\mathrm{Q}^{\mathrm{T}}\beta_1\Delta\ddot{\mathrm{u}}_t\,\Delta t-\mathrm{H}\overline{\beta_1}\Delta\dot{\mathrm{p}}_t\,\Delta t-\mathrm{S}\Delta\dot{\mathrm{p}}_t=-\mathrm{f}^{2}{}_{t+1}+\mathrm{S}\dot{\mathrm{p}}_t+\mathrm{H}\dot{\mathrm{p}}_t\,\Delta t+\mathrm{H}\bar{\mathrm{p}}_t+\mathrm{Q}^{\mathrm{T}}\ddot{\mathrm{u}}_t\,\Delta t+\mathrm{Q}^{\mathrm{T}}\dot{\mathrm{u}}_t$$

$$\mathrm{F}^{2}_{t+1}=-\mathrm{f}^{2}{}_{t+1}+\mathrm{S}\dot{\mathrm{p}}_t+\mathrm{H}\dot{\mathrm{p}}_t\,\Delta t+\mathrm{H}\bar{\mathrm{p}}_t+\mathrm{Q}^{\mathrm{T}}\ddot{\mathrm{u}}_t\,\Delta t+\mathrm{Q}^{\mathrm{T}}\dot{\mathrm{u}}_t$$

$$-\mathrm{Q}^{\mathrm{T}}\beta_1\Delta\ddot{\mathrm{u}}_t\,\Delta t-\mathrm{H}\overline{\beta_1}\Delta\dot{\mathrm{p}}_t\,\Delta t-\mathrm{S}\Delta\dot{\mathrm{p}}_t-\mathrm{F}^{2}_{t+1}=0$$

(4.78)

For the nonlinear solution process , residuals of the system will be expressed as in (4.79) and then some form of Newton-Raphson process will be applied as shown in (4.80):

$$\Psi^{1}_{t+1}=\mathrm{M}\Delta\ddot{\mathrm{u}}_t+P(\bar{\mathrm{u}}_{t+1})-\mathrm{Q}\overline{\beta_1}\Delta\dot{\mathrm{p}}_t\,\Delta t-\mathrm{F}^{1}_{t+1}=0 \tag{4.79}$$

$$\Psi^{2}_{t+1}=-\mathrm{Q}^{\mathrm{T}}\beta_1\Delta\ddot{\mathrm{u}}_t\,\Delta t-\mathrm{H}\overline{\beta_1}\Delta\dot{\mathrm{p}}_t\,\Delta t-\mathrm{S}\Delta\dot{\mathrm{p}}_t-\mathrm{F}^{2}_{t+1}=0$$

$$J\begin{Bmatrix}\delta\Delta\ddot{u}_t\\ \delta\Delta\dot{p}_t\end{Bmatrix}^{i+1}=-\begin{Bmatrix}\Psi^{(1)}_{t+1}\\ \Psi^{(2)}_{t+1}\end{Bmatrix}^{i}$$

$$\begin{Bmatrix}\Delta\ddot{u}_t\\ \Delta\dot{p}_t\end{Bmatrix}^{i+1}=\begin{Bmatrix}\Delta\ddot{u}_t\\ \Delta\dot{p}_t\end{Bmatrix}^{i}+\begin{Bmatrix}\delta\Delta\ddot{u}_t\\ \delta\Delta\dot{p}_t\end{Bmatrix}^{i+1} \tag{4.80}$$

The jacobien matrix is given by:

$$J=\begin{bmatrix}\dfrac{\partial\Psi^{(1)}_{t+1}}{\partial\Delta\ddot{u}_t} & \dfrac{\partial\Psi^{(1)}_{t+1}}{\partial\Delta\dot{p}_t}\\ \dfrac{\partial\Psi^{(2)}_{t+1}}{\partial\Delta\ddot{u}_t} & \dfrac{\partial\Psi^{(2)}_{t+1}}{\partial\Delta\dot{p}_t}\end{bmatrix}=\begin{bmatrix}\mathrm{M}+\dfrac{1}{2}\mathrm{K}_{\mathrm{T}}\beta_2\Delta t^2 & -\mathrm{Q}\overline{\beta_1}\Delta t\\ -\mathrm{Q}^{\mathrm{T}}\beta_1\Delta t & -\mathrm{S}+\mathrm{H}\beta_1\Delta t\end{bmatrix} \tag{4.81}$$

$$\mathrm{K}_{\mathrm{T}}=\int_{\Omega}\mathrm{B}^{\mathrm{T}}\mathrm{DB}\mathrm{d}\Omega$$

(Zienkiewicz and Taylor, 1991; Zienkiewicz et al., 1999; Zienkiewicz and Taylor, 1985)

CHAPTER 5
VERIFICATION OF THE FINITE ELEMENT PROGRAM

5.1 Introduction

The nonlinear dynamic finite element program constructed for the purpose of analysing the nonlinear dynamic soil pile interaction problem has been verified for the following cases selected from geotechnical engineering.

1. Finite element analysis of the deformations in a linear elastic soil medium under self weight. (Static, Linear Elastic).
2. Finite element analysis of the deformations in a linear elastic soil medium due to a rigid foundation constructed on the surface of the soil layer. Rigid foundation is represented by applying a uniform displacement at the nodes on which the rigid foundation is placed. The analysis is performed for both the drained and undrained conditions. (Static, Linear Elastic)
3. Finite element analysis of the deformations in a linear elastic soil medium due to sinusoidal displacements applied at the specified points on surface of the soil layer (Dynamic, Linear Elastic).
4. Dynamic finite element analysis of a circular linear elastic soil medium around a section of a linear elastic rigid pile with sinusoidal prescribed displacements applied on the degrees of freedom of the nodes just on the boundary of the pile with the soil medium.

5.2 Verification Example 1:

In the following verification example, the deformation behaviour of a linear elastic soil medium under gravitational forces will be analysed with the finite element method. Naturally, this will be a static finite element analysis, hence, this example will enable the verification of the basic steps like construction of the element and global matrices, their assemblage, solution of the equation for equilibrium and calculation of displacements, strains and stresses for a linear elastic soil medium.

It is possible to determine the initial stress state of the soil medium under gravitational forces by applying the self weight of the soil medium as a force vector.However, this type of analysis doesn't give the real displacements under self weight.

The finite element mesh shown in Figure 5.1. is constructed by four noded two dimensional plane strain elements which behave linear elastically according to Hooke's Law. The dimensions of the mesh is 30m. in the x direction and 10 m. in the y direction with the dimensions of each element being 2m.x1m.. The nodes lying on the bottom line of the mesh is constrained both in the x and y directions and the ones on the right and left sides of the mesh are constrained only in the x direction.Total number of elements and that of nodes are 150 and 176, respectively. For verification example 1, the finite element mesh should be considered without taking the external excitation into account. Necessary material properties are as follows:

Unit weight: 20 KN/m^3
Saturated unit weight : 21 KN/m^3
Unit weight of water: 9.81 KN/m^3
Elasticity modulus: 50000KN/m^2
Poisson's ratio: 0.35

Analysis of the vertical deformations of a linear elastic soil medium under its self weight is a static problem, therefore parts of the force vector due to mass and damping of the system is excluded from the overall finite element equilibrium equation as (5.1). This equation is solved for displacements afterwhich strains and stresses are obtained by using the stain displacement (5.2) and stress-strain (5.3) (constitutive) relations. Flow chart of the FEM program for the verification example 1 is given in Figure 5.2.

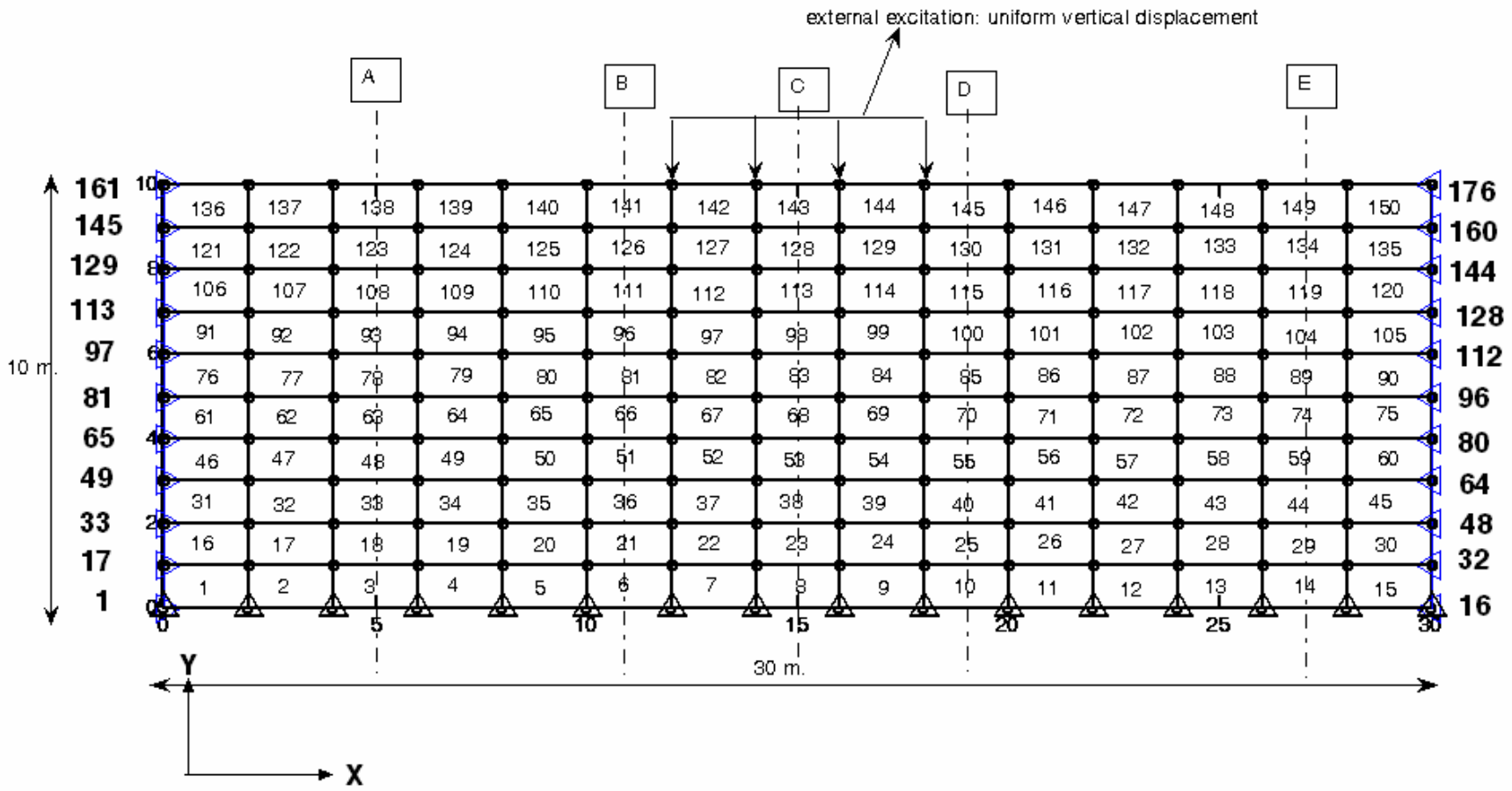


Figure 5.1. Finite Element Mesh for Verification Examples 1 and 2

$$\begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix} \begin{Bmatrix} u_a \\ u_b \end{Bmatrix} = \begin{Bmatrix} F_a \\ F_b + R_b \end{Bmatrix} \tag{5.1}$$

Here,

$\begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix}$: Stiffness matrix rearranged for the solution process according to known and unknown degrees of freedom

$\begin{Bmatrix} u_a \\ u_b \end{Bmatrix}$: Displacement matrix rearranged for the solution process according to known and unknown degrees of freedom

$\begin{Bmatrix} F_a \\ F_b + R_b \end{Bmatrix}$: Force vector including the contribution from gravity forces and the support reactions. This vector is also rearranged for the solution process according to known and unknown degrees of freedom

$$\{\varepsilon\} = [B]\{u\} \tag{5.2}$$

$$\{\sigma\} = [D]\{\varepsilon\} \tag{5.3}$$

Here,

$\{\varepsilon\}$: strain vector

$\{\sigma\}$: stress vector

$[B]$: strain-displacement matrix

$[D]$: stress-strain (constitutive) matrix.

For linear elastic plane strain case, constitutive matrix is given as:

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \dfrac{1-2\nu}{2} \end{bmatrix} \tag{5.4}$$

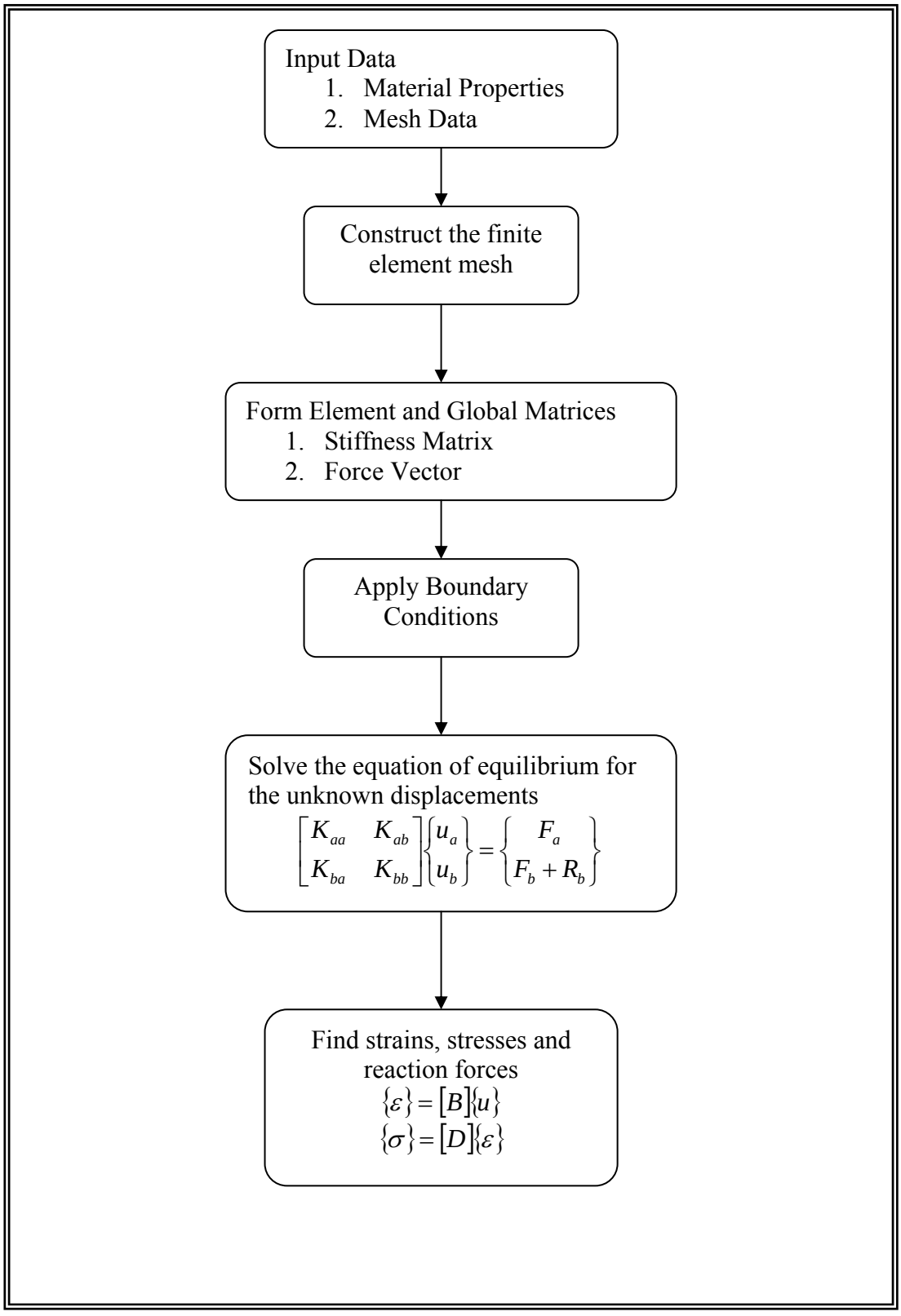
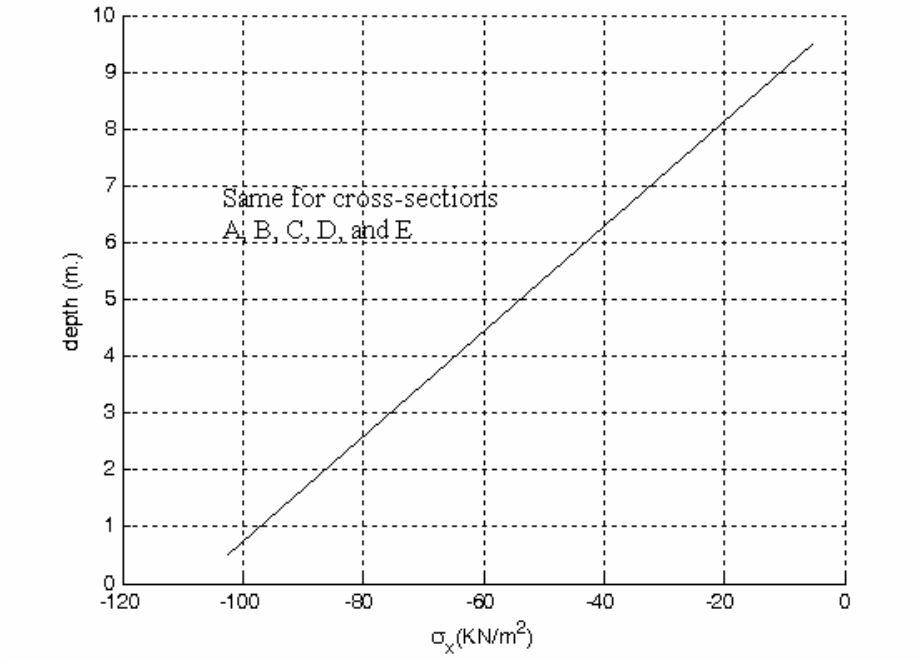


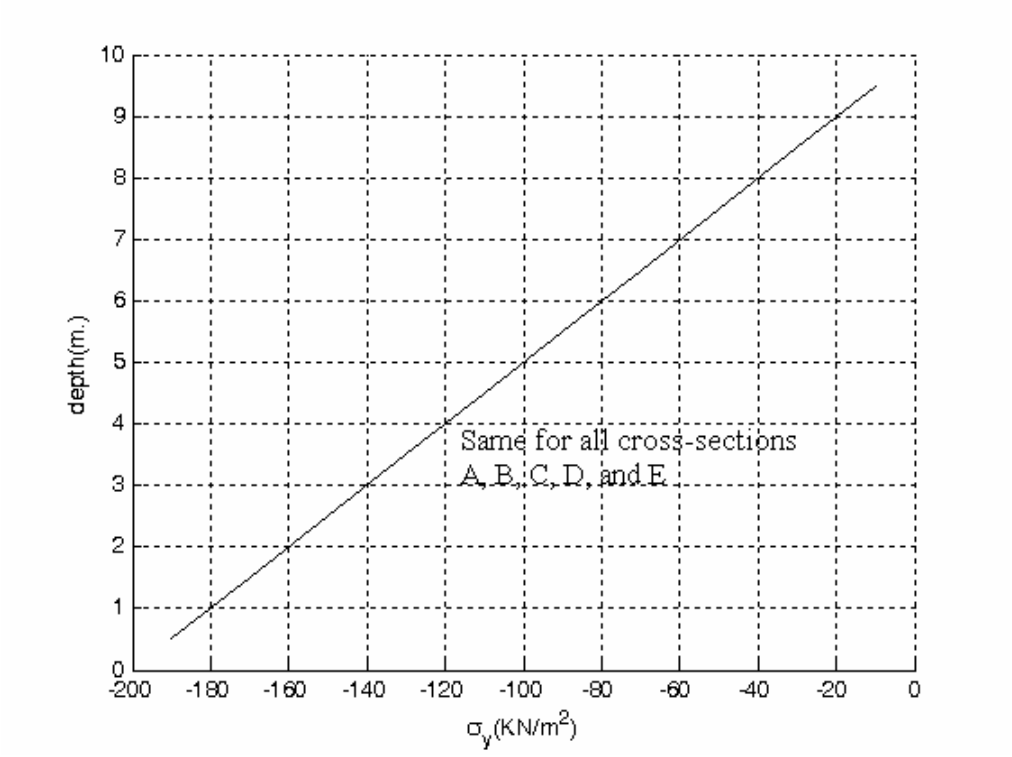
Figure 5.2 Flow chart for Verification Problem 1.

Figures 5.3 and 5.4 present the results of the finite element analysis. According to Figures 5.3 (a) and (b) vertical (σ_y)and horizontal stresses (σ_x) increase with depth for all the cross-sections (A, B, C, D, and E). Horizontal stresses are calculated approximately as equal to half of the vertical stresses at the depth of analysis. In addition, shear stresses are calculated close to zero as shown in Figure 5.3 (c).

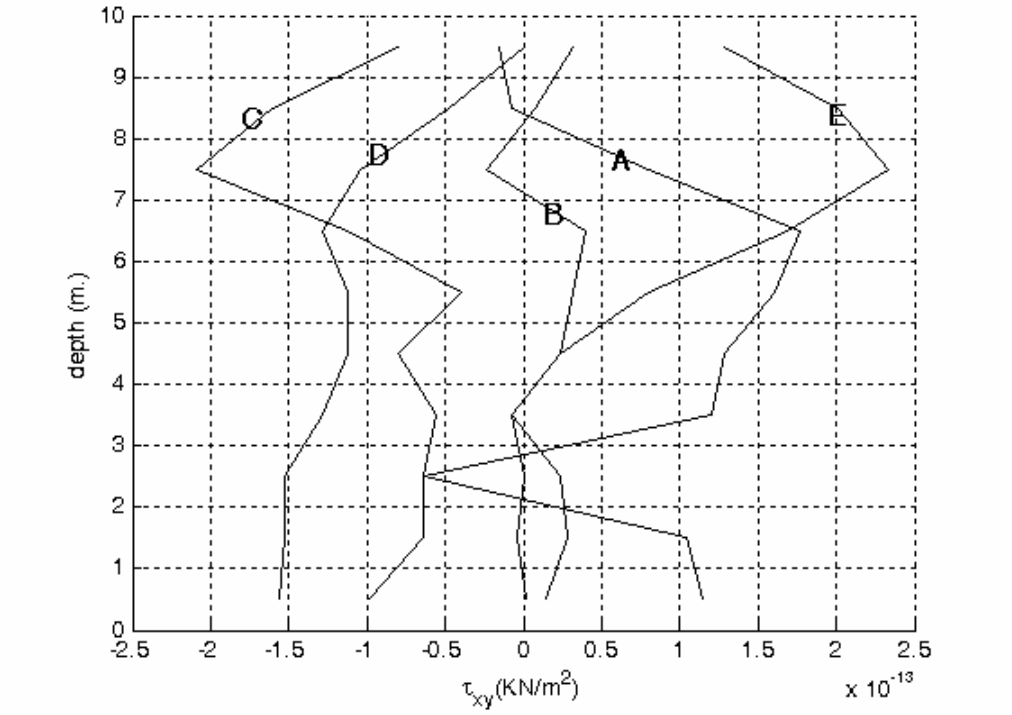
FEM analysis performed for this example has the capability to reflect the stresses under gravitational forces close to the ones in reality. However, when the deformations are considered, different type procedures called K0 procedure and gravity turn-on procedure (Burland,J.B.,1978) should be followed. Hence, the initial stress and strain system can be calculated closer to reality with the help of these procedures. Under gravitational forces, the real vertical displacement behaviour should be like plot(2) as shown in Figure 5.5.



(a)

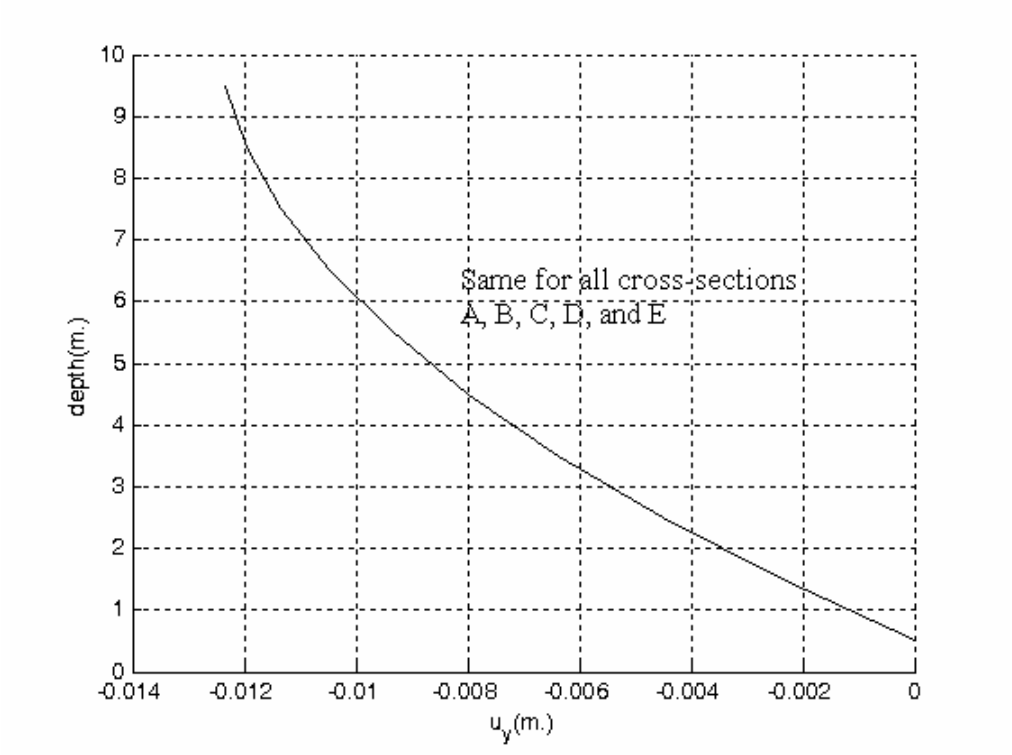


(b)



(c)

Figure 5.3 σ_x-depth (a), σ_y-depth (b), τ_{xy}-depth (c), for Cross-sections A, B, C, D, and E



Şekil 5.4 Change of vertical displacements (u_y) with the depth of the soil medium for cross-sections A, B, C, D, and E

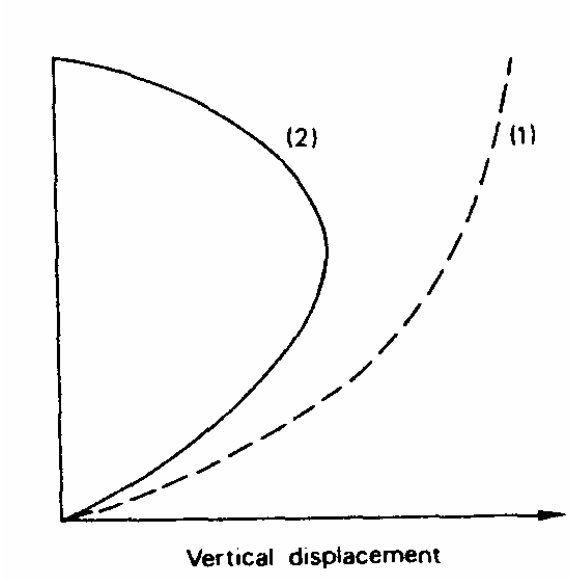


Figure 5.5 Vertical displacement behaviour under gravitational forces (Burland, 1978)

5.3. Verification Example 2:

The following verification example simulates the deformation and stress-strain behaviour of a linear elastic soil medium when a rigid foundation having a width of 6m. is constructed at the top of a soil layer and makes a uniform settlement of 50 mm. The finite element mesh of the problem (Figure 5.1) and the material properties are the same as in verification problem 1. The exaggerated deformed mesh of the problem for 2m. settlement is shown in Figure 5.6.

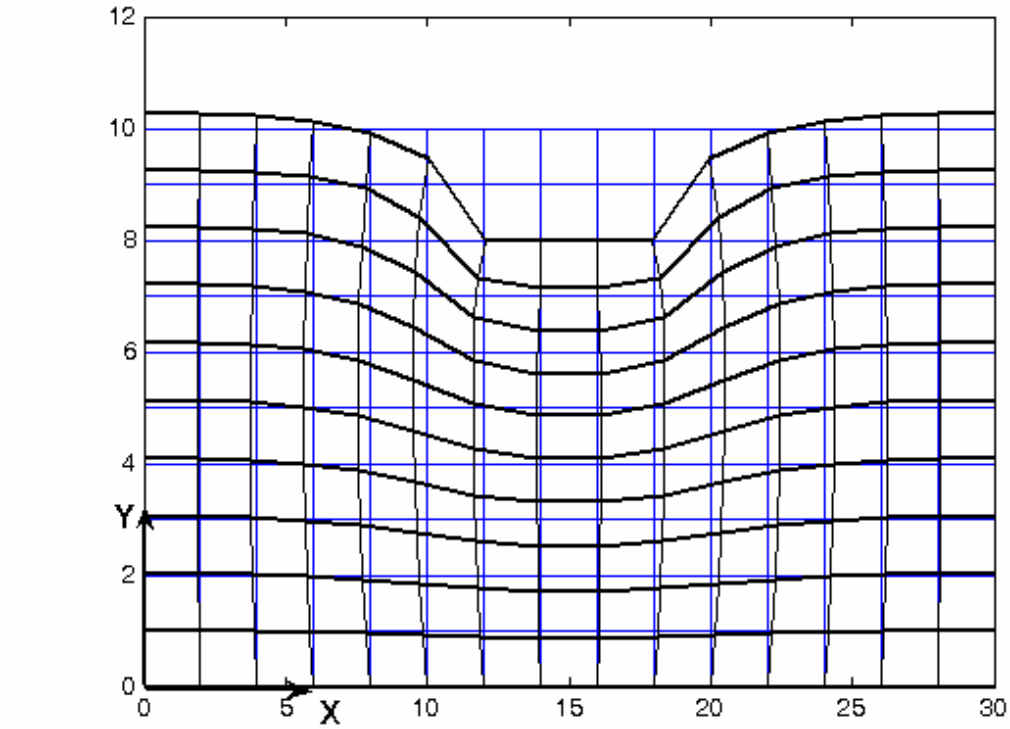
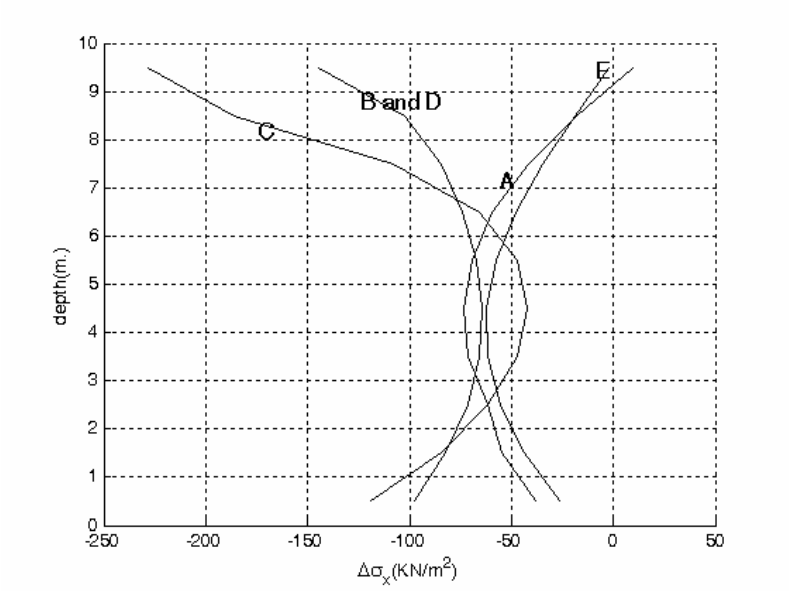


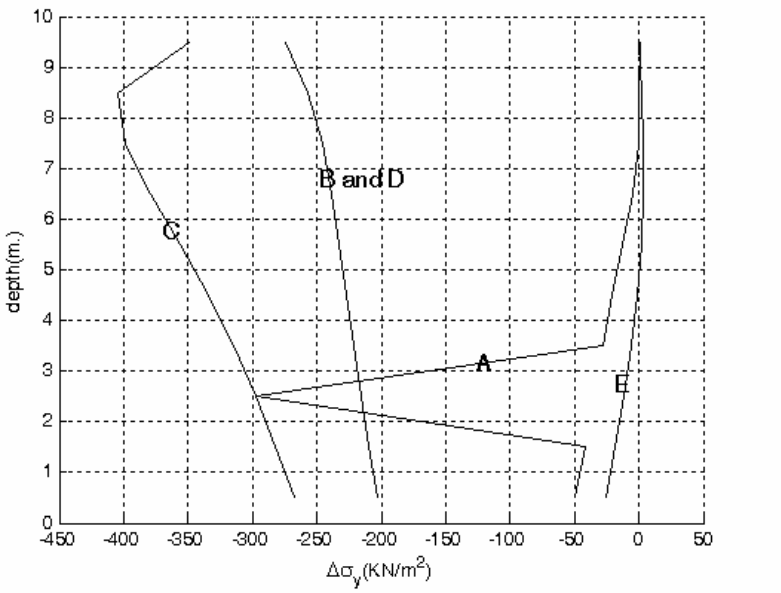
Figure 5.6. Exaggerated Deformed Mesh for a Rigid Foundation (B=6m.) Making a Settlement of 2m.

Finite element analysis for this problem is exactly the same as that for the verification problem 1. So the solution process of the equilibrium equation for displacements, stresses and strains follow the same flow chart as in Figure 5.2. Figures 5.7, 5.8, and 5.9 show the variation of stresses, strains and displacements respectively, for the cross-sections A, B, C, D and E. These curves are obtained by considering the values at the middle of each element. The general trend in these

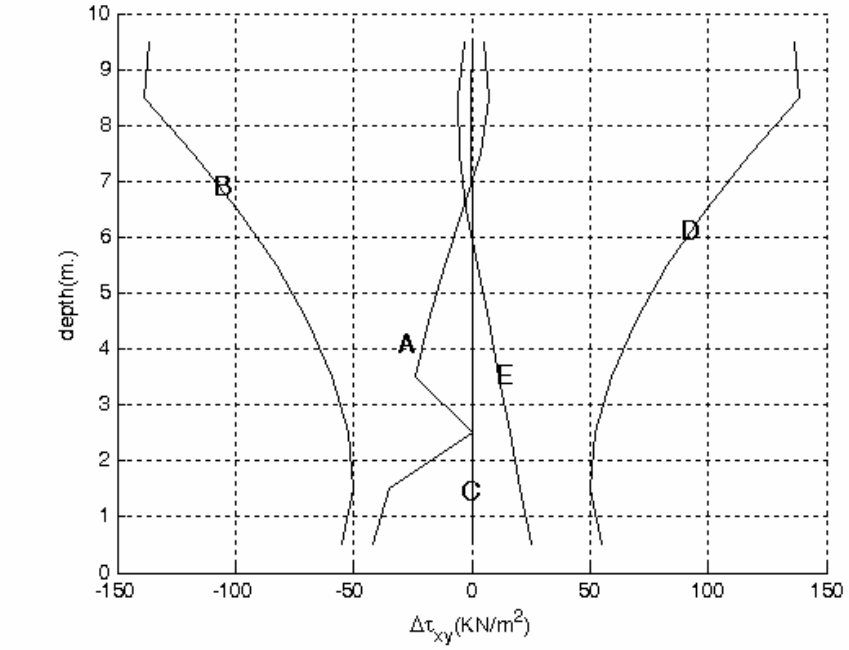
figures is cross-sections B, D which pass through the edges of the rigid foundation and A, E which are taken from the left and right sides of the mesh show the same trend. Vertical stresses (Figure 5.7 (b)) and strains (Figure 5.8 (b)) decrease with depth reaching their maximum values at the cross section C passing through the center of the rigid foundation. Similarly, vertical displacements (Figure 5.9 (b)) decrease with depth having their maximum values at the top layer for each cross-section.



(a)

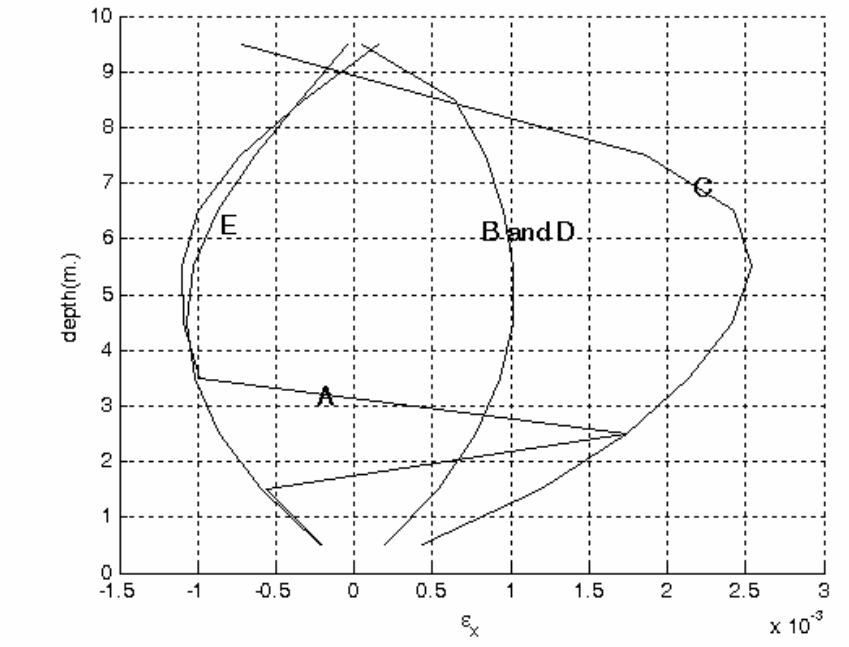


(b)

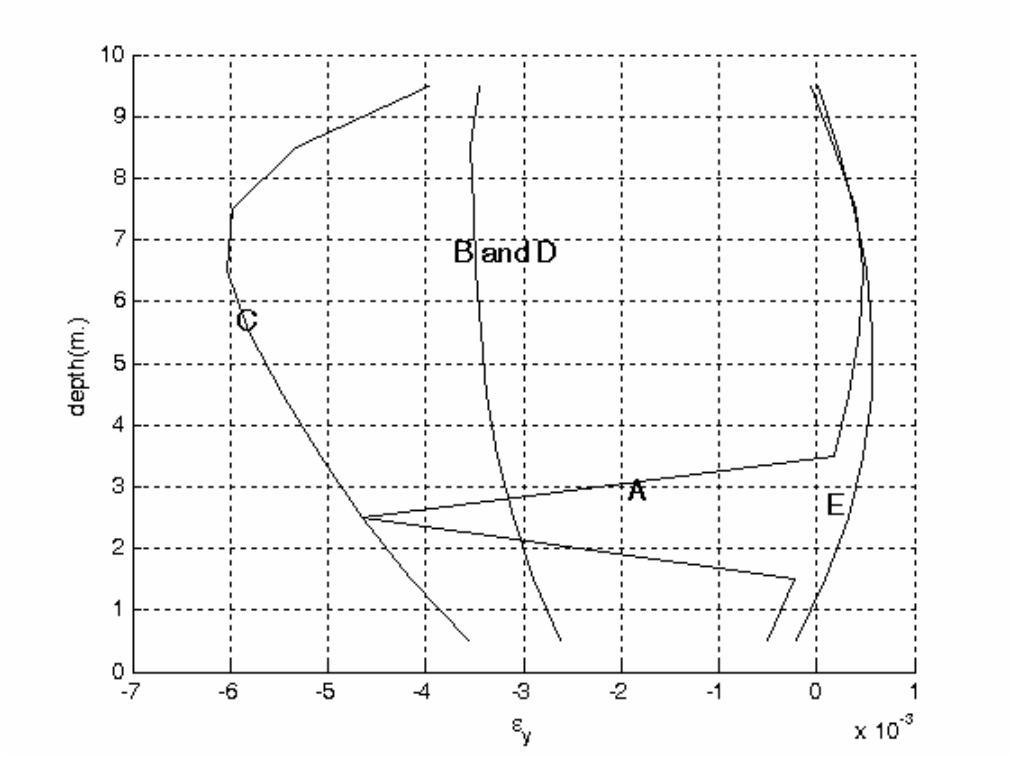


(c)

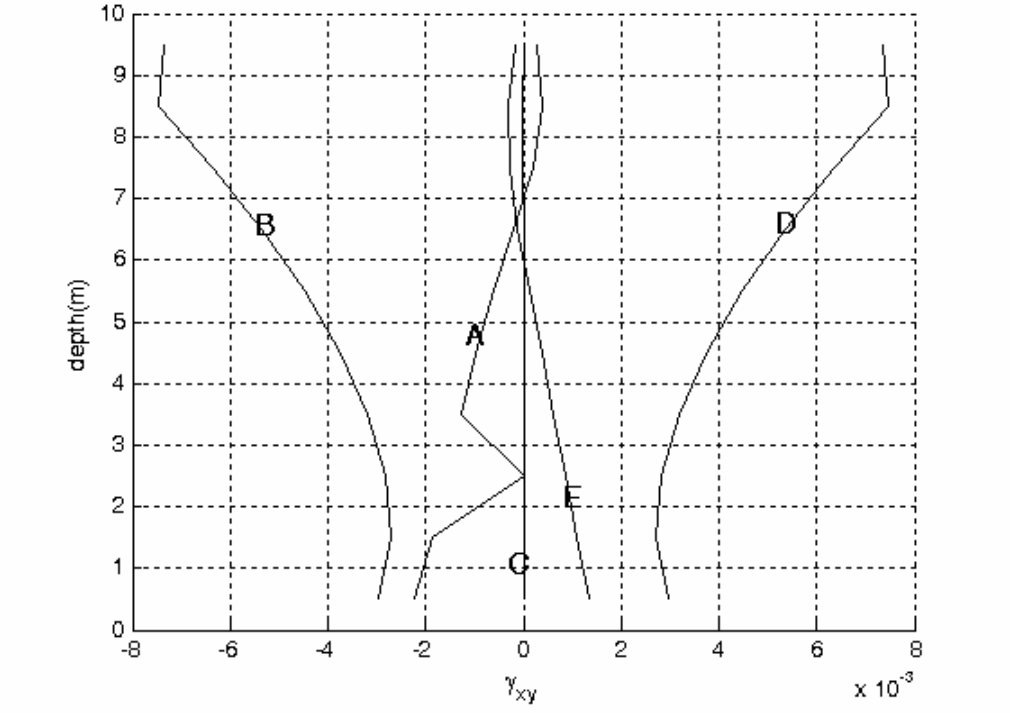
Figure 5.7. Variation of σ_x, σ_y, τ_{xy} with depth for cross-sections A, B, C, D,E.



(a)

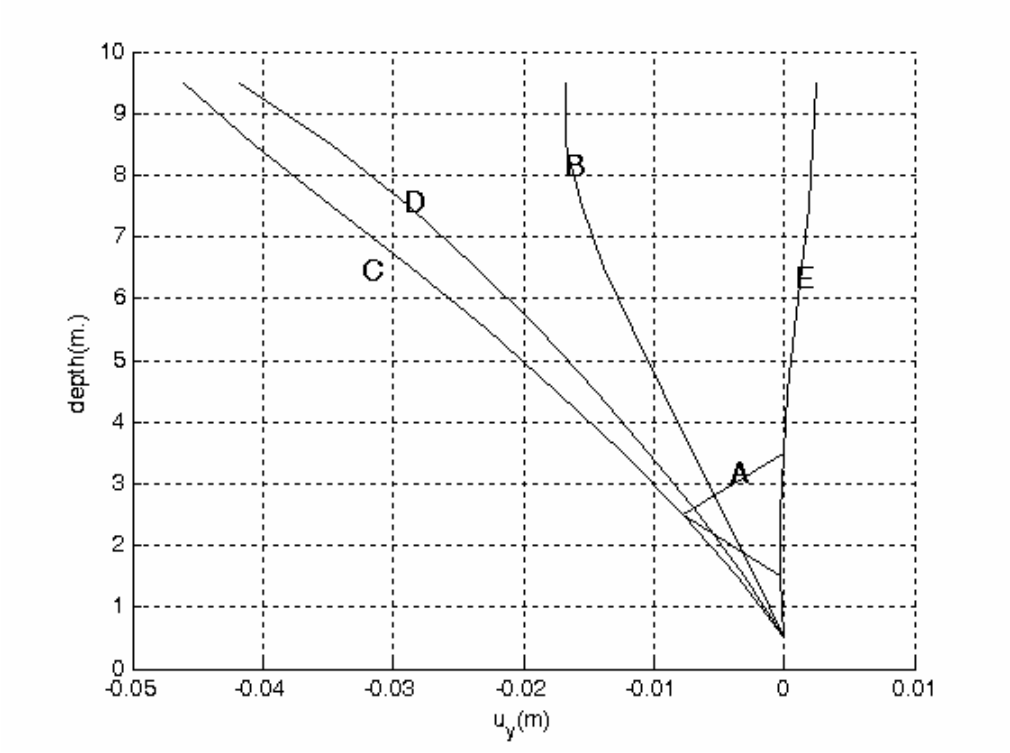


(b)

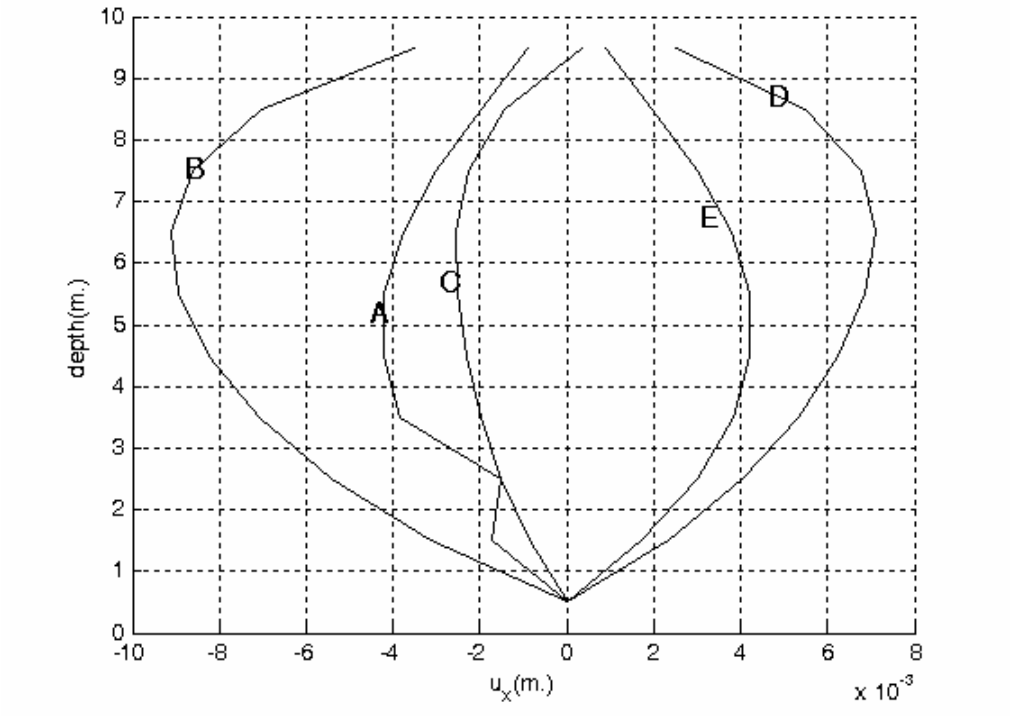


(c)

Figure 5.8 Variation of ε_x, ε_y, γ_{xy} with depth for cross-sections A, B, C, D,E.



(a)



(b)

Figure 5.9 Variation of vertical (u_y) and horizontal displacements (u_x) with depth for cross-sections A, B, C, D,E.

5.4. Verification Example 3:

The following verification example simulates the vertical sinusoidal displacements of a rigid foundation on a linear elastic soil medium. This example naturally doesn't represent the actual soil behaviour under such a condition because it doesn't involve the appropriate soil constitutive model , hence it will only verify that the incremental form of the equation of motion with the time integration algorithm incorporated in it, works properly.

Figure 5.10 shows the finite element mesh of the problem. The soil medium is 30 m. and 10 m. in length in the global X and Y directions respectively. As was the case in the previous example, the nodes along the bottom length of the soil medium is constrained in the x and y directions and the ones along the very right and left vertical sides are constrained only in the x direction. Four noded two dimensional plane strain elements with sides 1m. in length in both directions are used in the construction of the finite element mesh. The nodes and the elements are numbered in the same manner as in Figure 5.1. The properties of the linear elastic soil medium and that of the vertical dynamic excitation are given as follows:

Material properties of the soil medium:
Unit weight: 20 KN/m^3
Saturated unit weight : 21 KN/m^3
Unit weight of water: 9.81 KN/m^3
Elasticity modulus: 50000KN/m^2
Poisson's ratio: 0.35

Properties of the dynamic excitation:
Prescribed displacements, velocities and accelerations are calculated according to:
$d(t)=d_0\sin(wt)$, $v(t)=d_0 w\cos(wt)$, $a(t)= -d_0 w^2\sin(wt)$
d0: amplitude of the sinusoidal displacements, w: frequency, t :time , d(t): displacement, v(t):velocity, a(t):acceleration
d0 =0.0008763m. , w= 1Hz.=6.283 rad/sn.

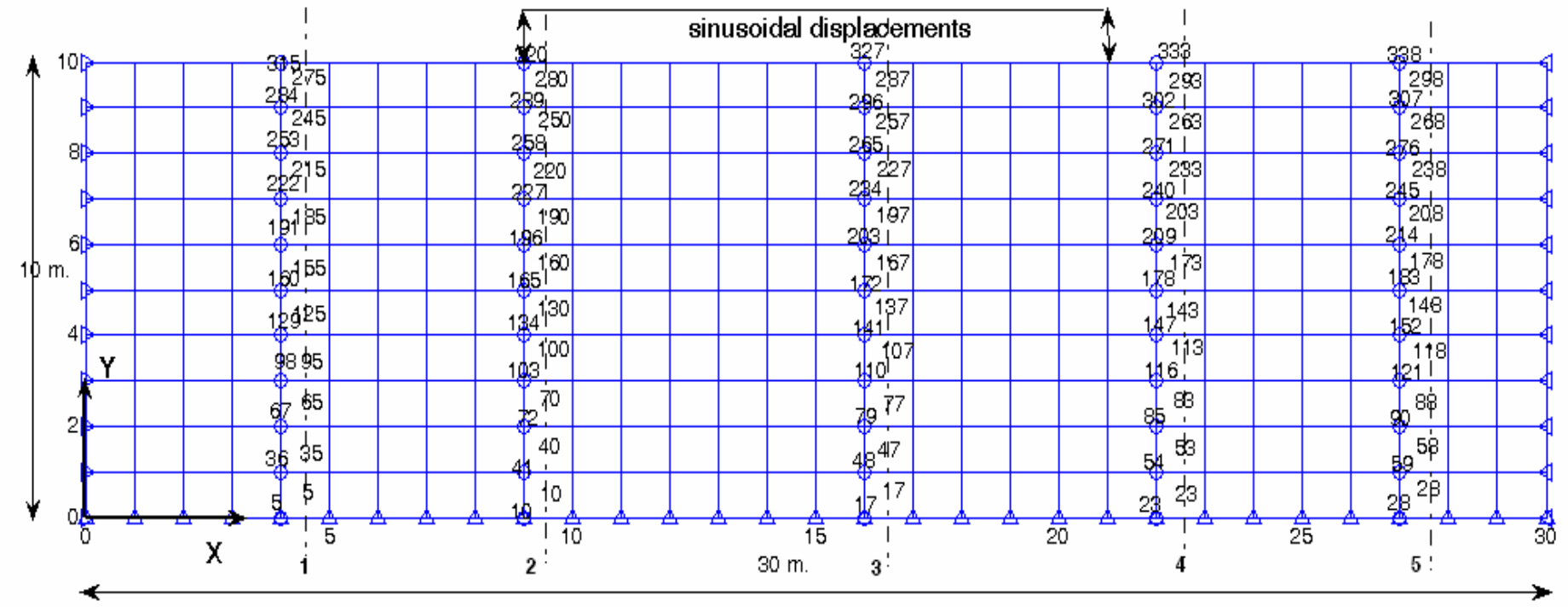
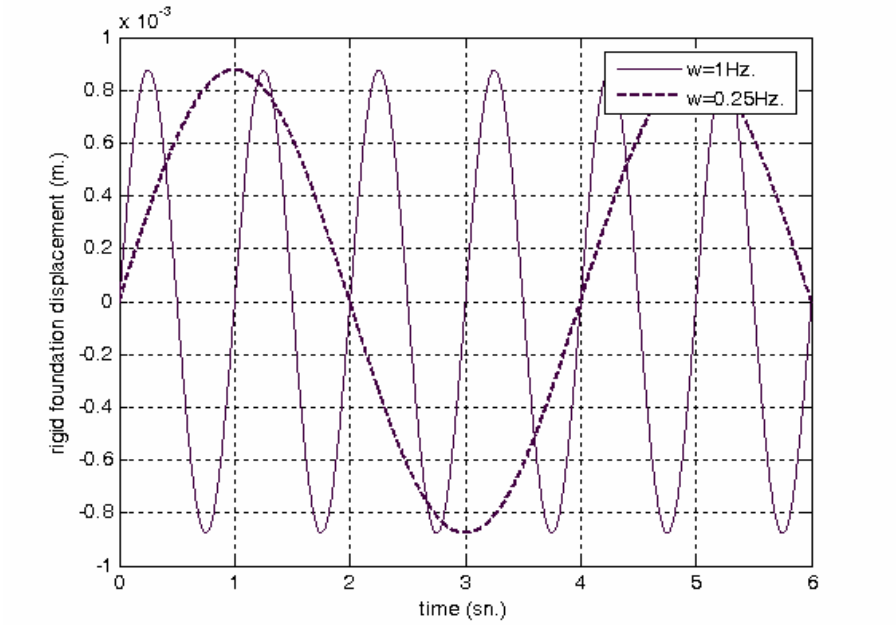
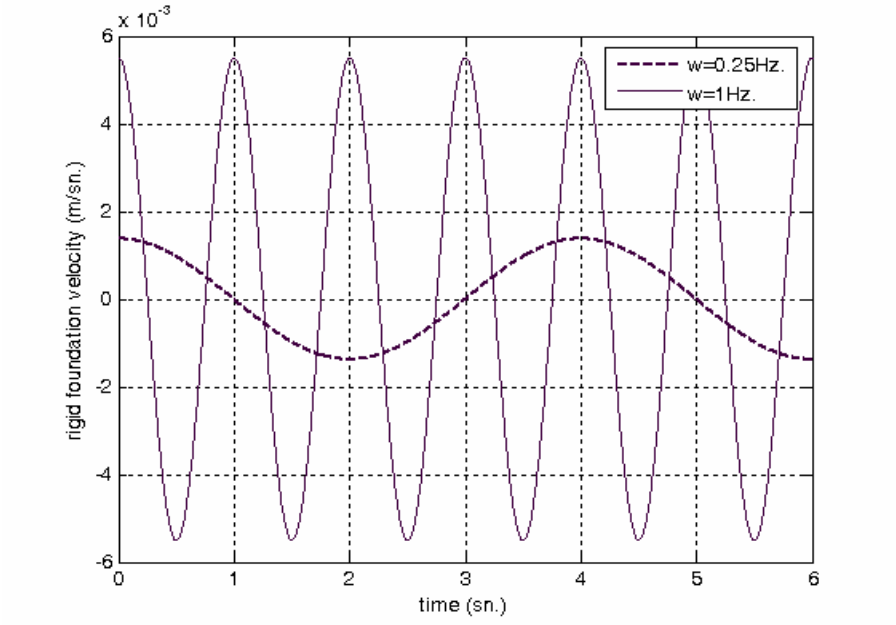


Figure 5.10 Finite element mesh for verification problem 3

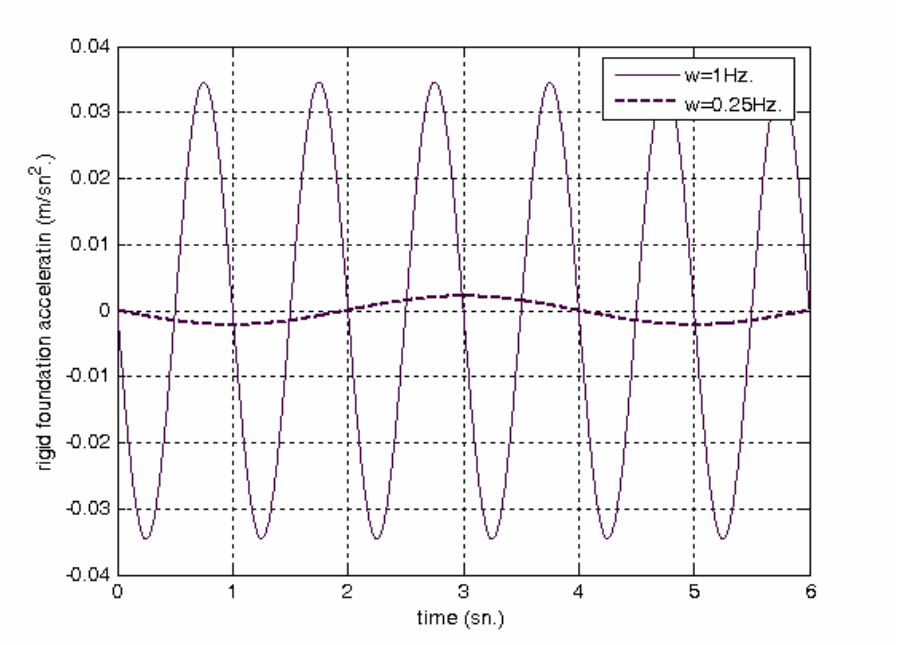
Rigid foundation displacements, velocities and accelerations are presented as a function of time in Figures 5.11 (a), (b) and (c), respectively. In addition, Figures 5.12 (a) and (b), show the deformed finite element mesh under exaggerated cyclic excitation



(a)

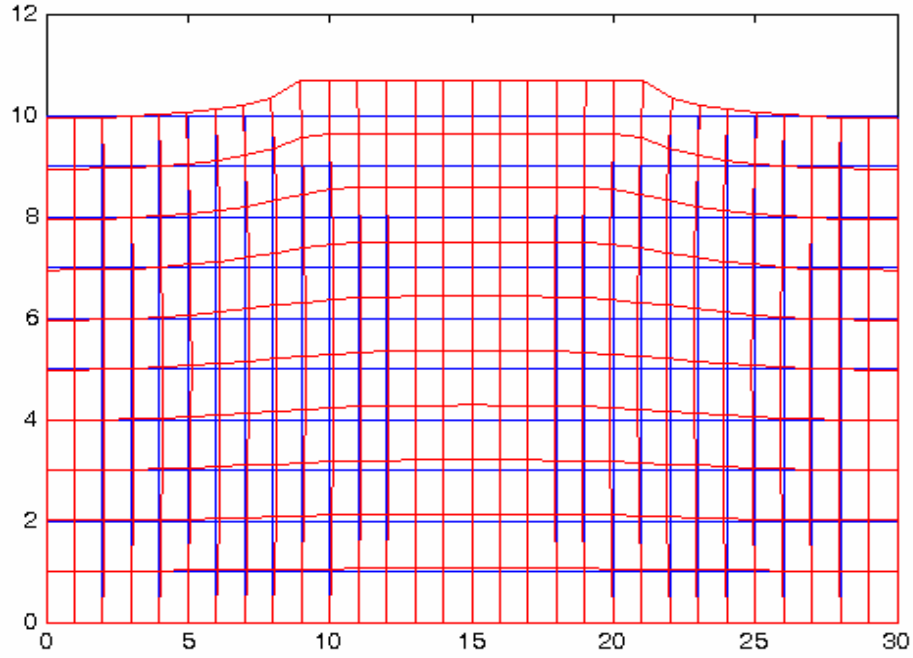


(b)

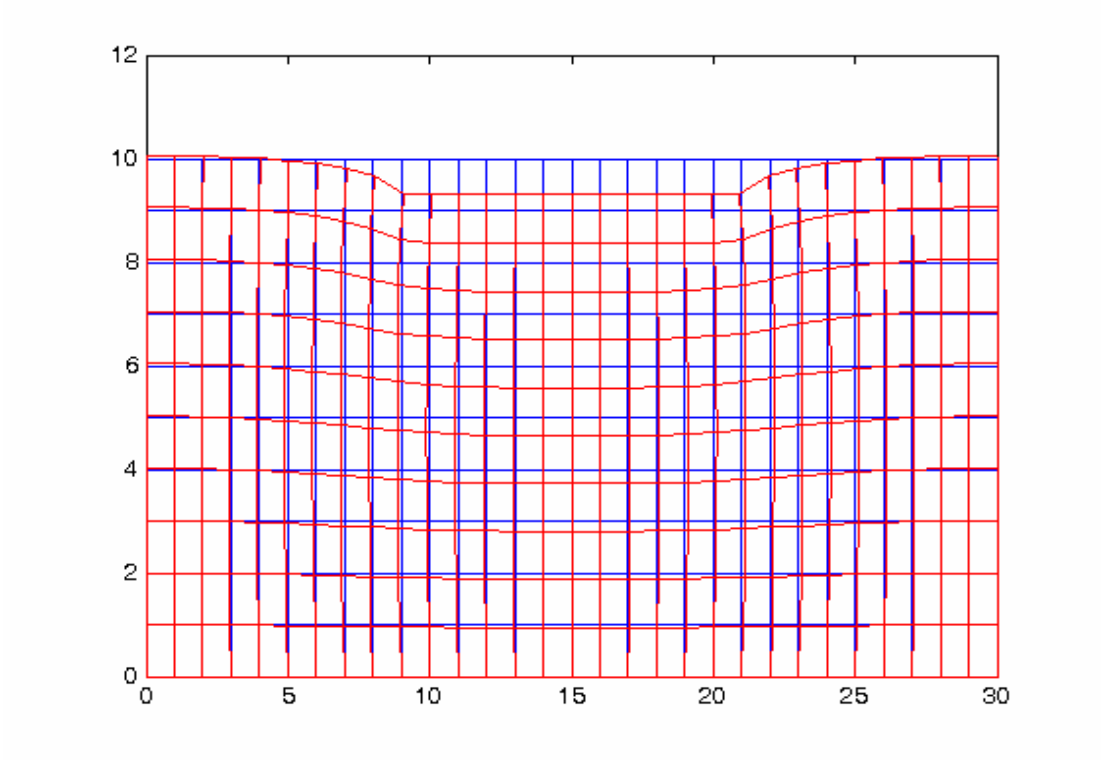


(c)

Figure 5.11 Prescribed sinusoidal displacements (a), velocities (b) and accelerations (c)



(a)



(b)

Figure 5.12 Deformed Finite Element Meshes in Exaggerated Form, (a) Unloading (b) Loading

The flow chart the verification problem 3 is given in Figure 5.13. Calculated displacements, velocities and accelerations in the Y global direction for the second nodes of the elements lying on the cross-sections 1, 2, 3, 4 and 5 are presented in Figures 5.14, 5.15 and 5.16.

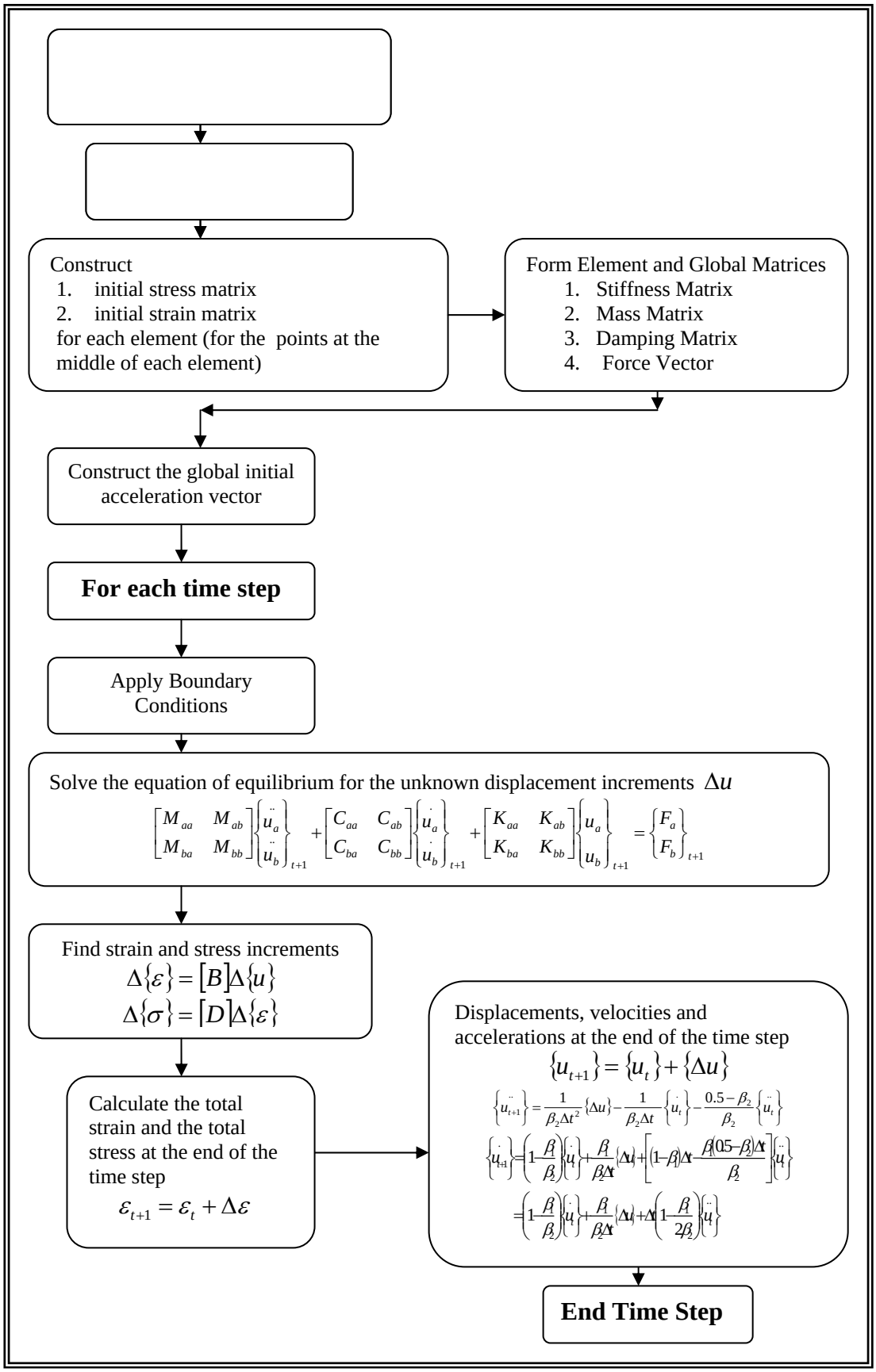
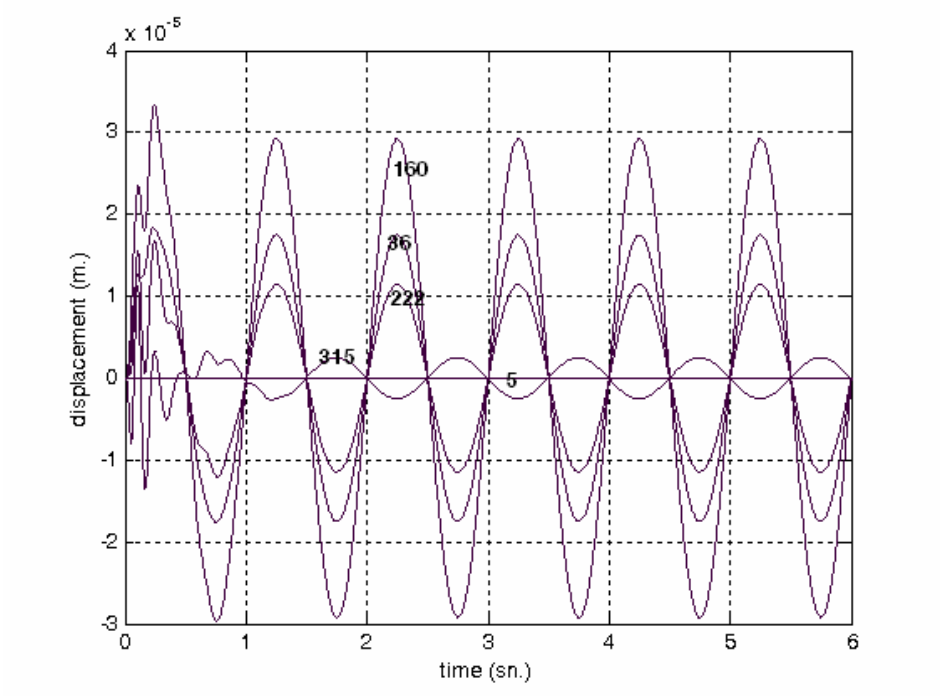
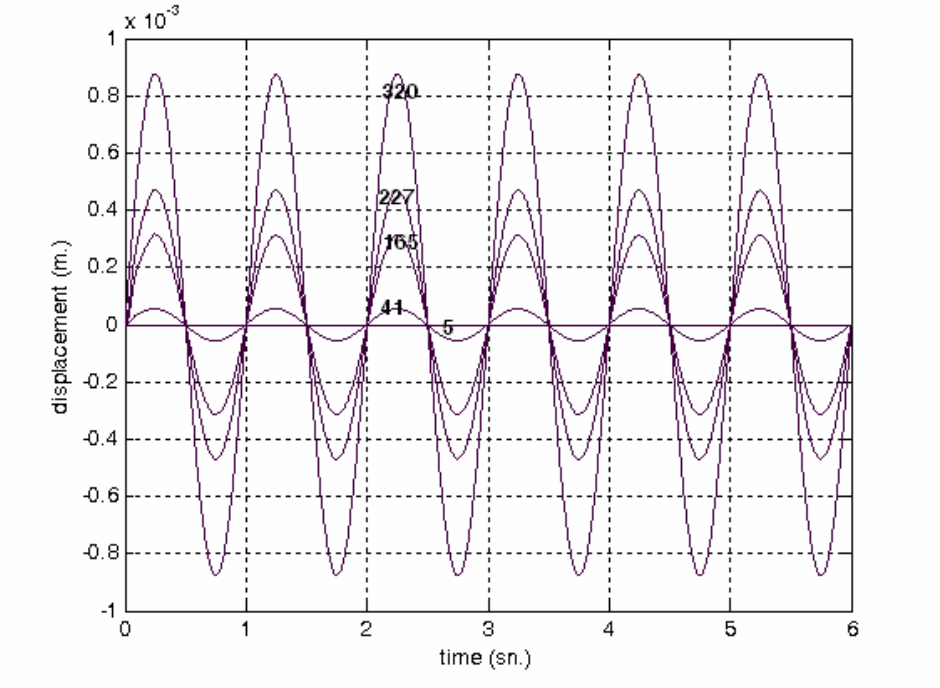


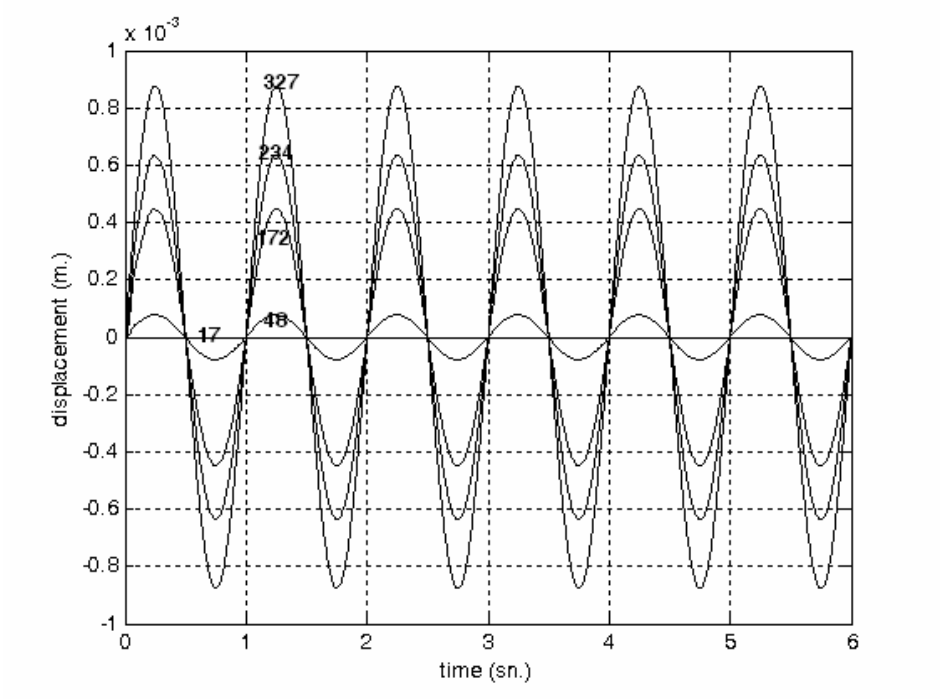
Figure 5.13 Flow chart for the verification problem 3



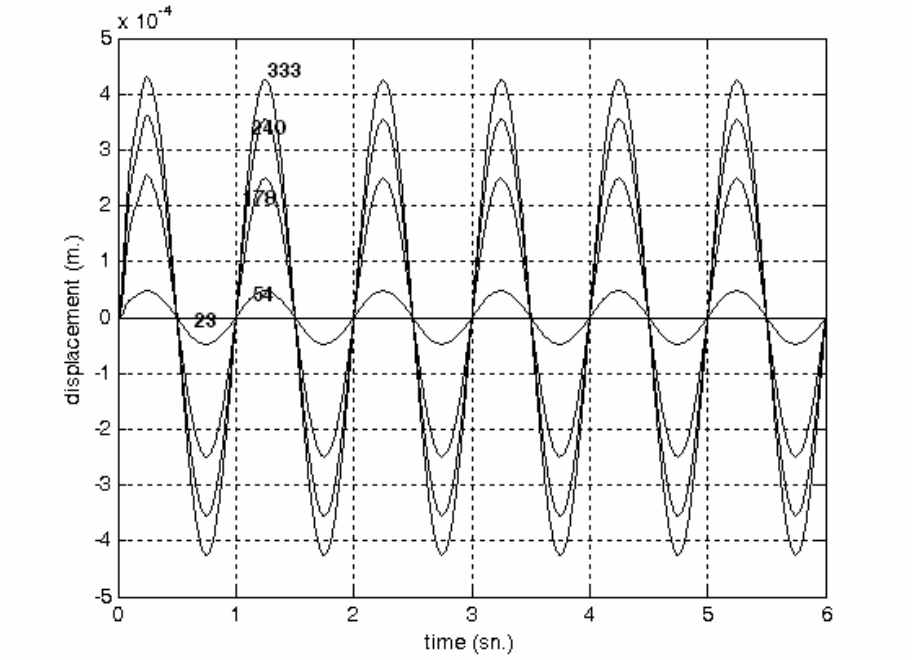
(a) cross-section 1



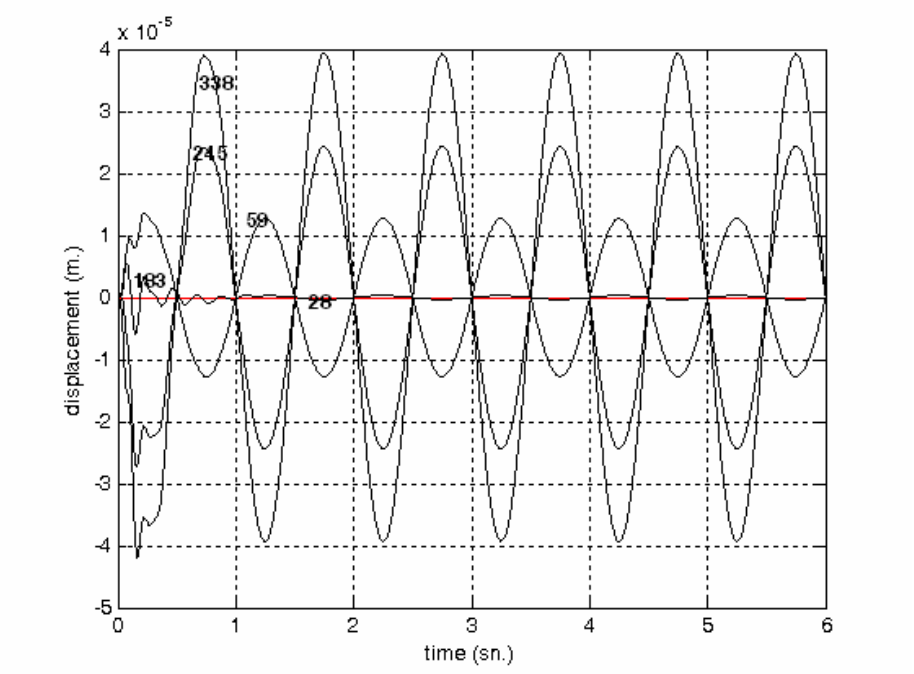
(b) cross-section 2



(c) cross-section 3

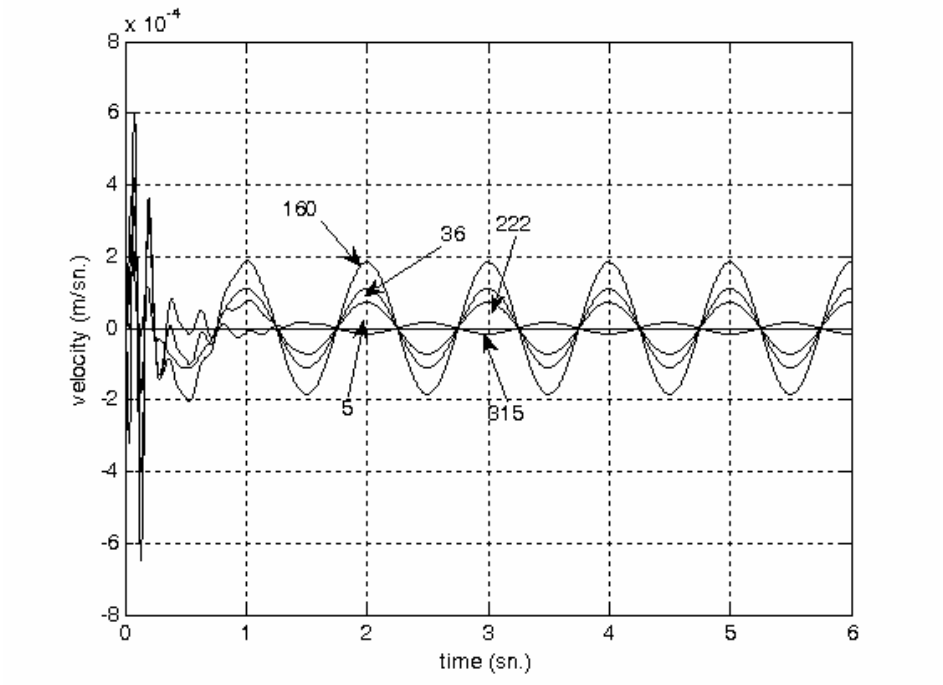


(d) cross-section 4

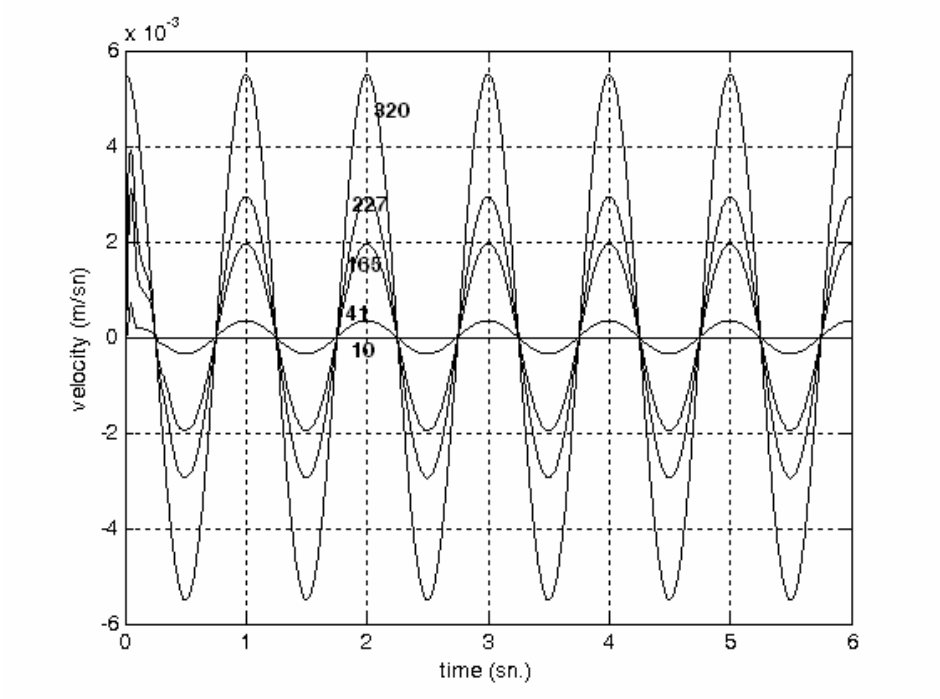


(e) cross-section 5

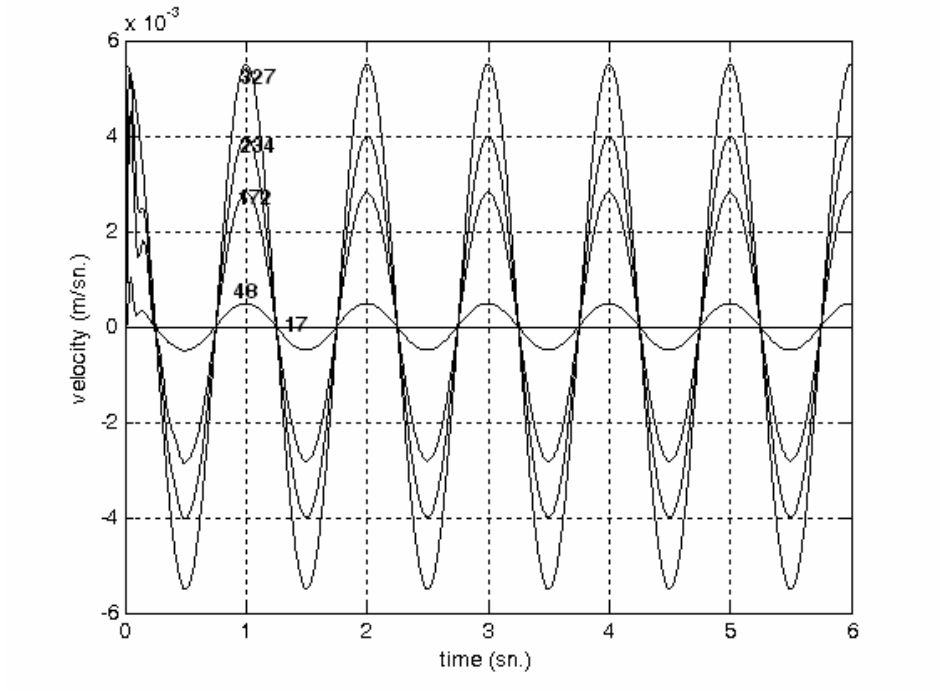
Figure 5.14 Calculated displacements in the Y global direction for the second nodes of the elements lying on the selected cross -sections



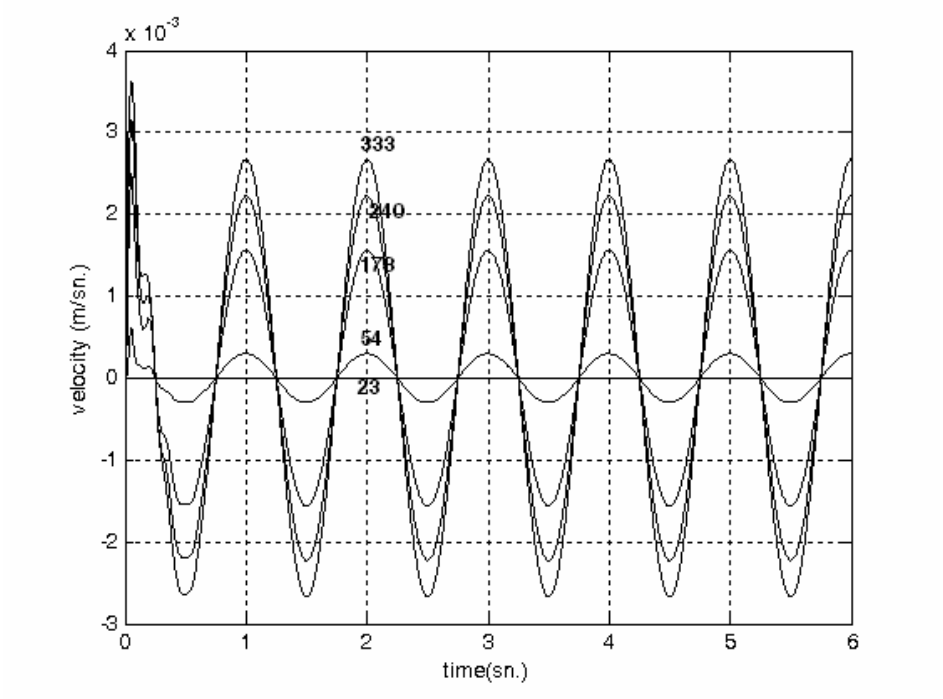
(a) cross-section 1



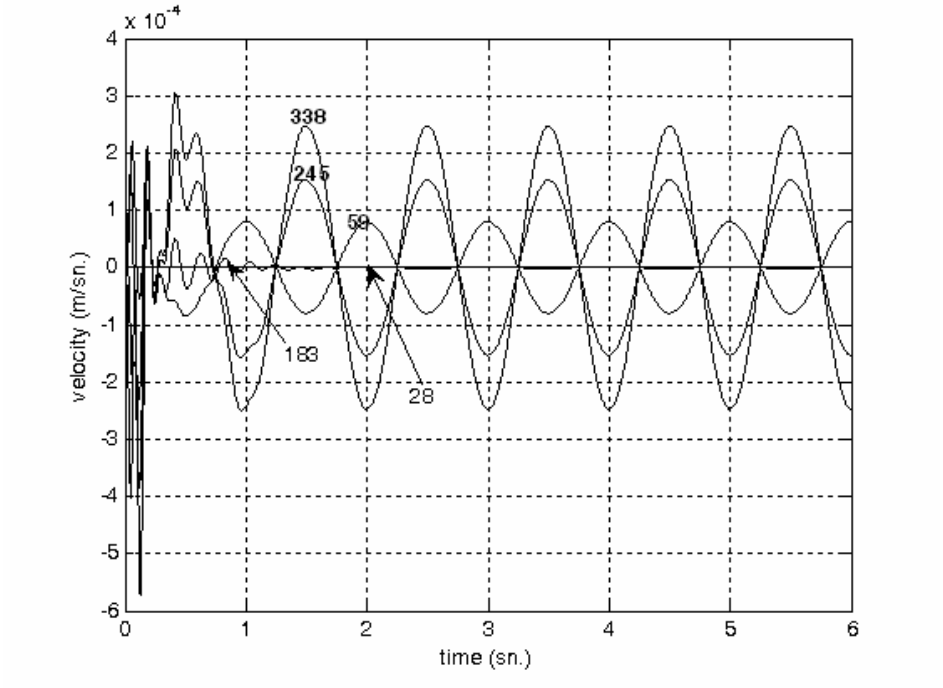
(b) cross-section 2



(c) cross-section 3

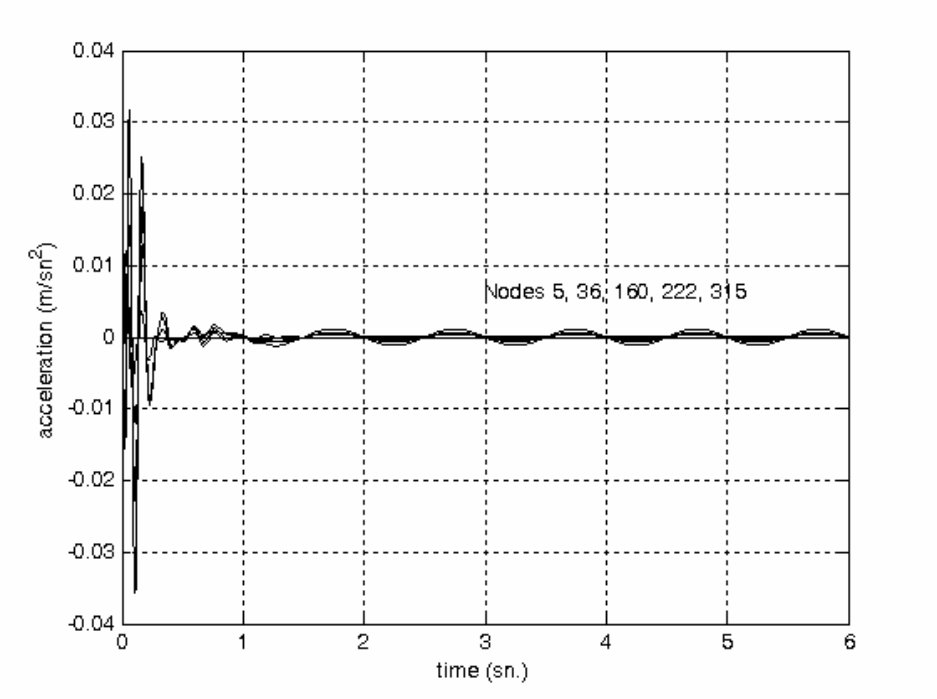


(d) cross-section 4

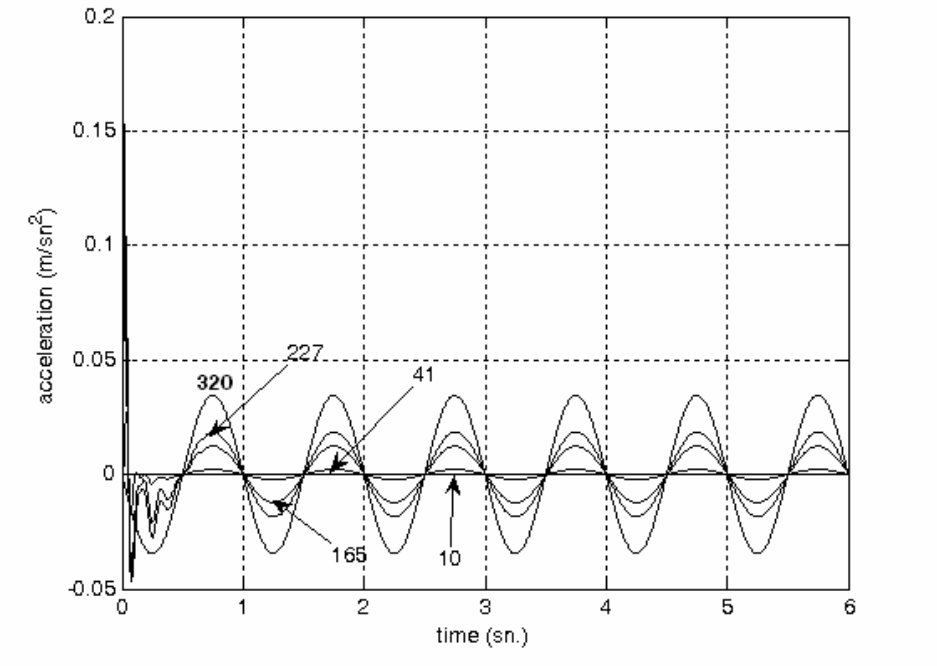


(e) cross-section 5

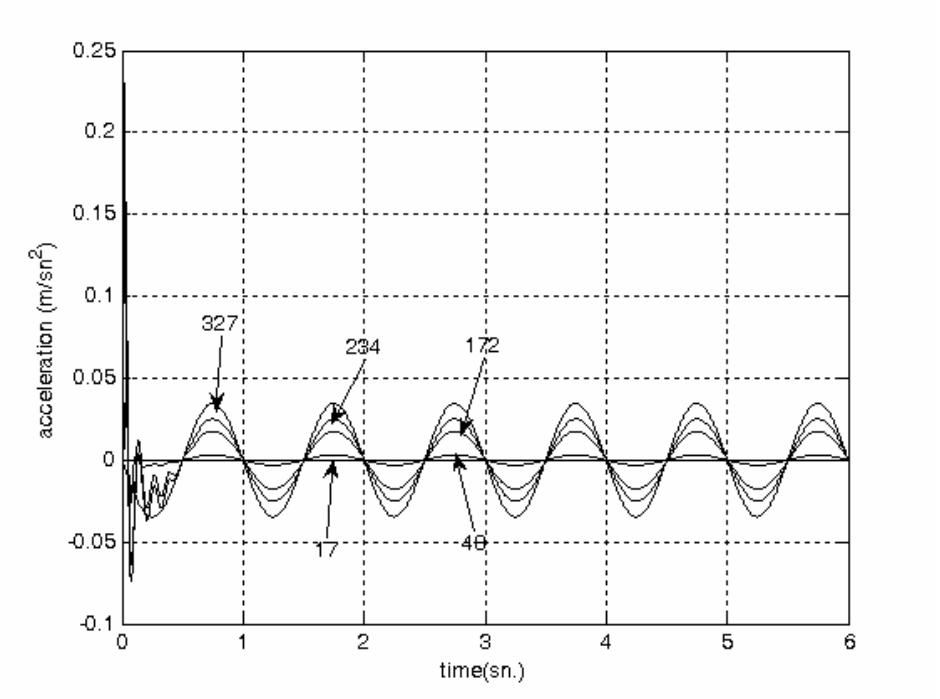
Figure 5.15 Calculated velocities in the Y global direction for the second nodes of the elements lying on the selected cross-sections



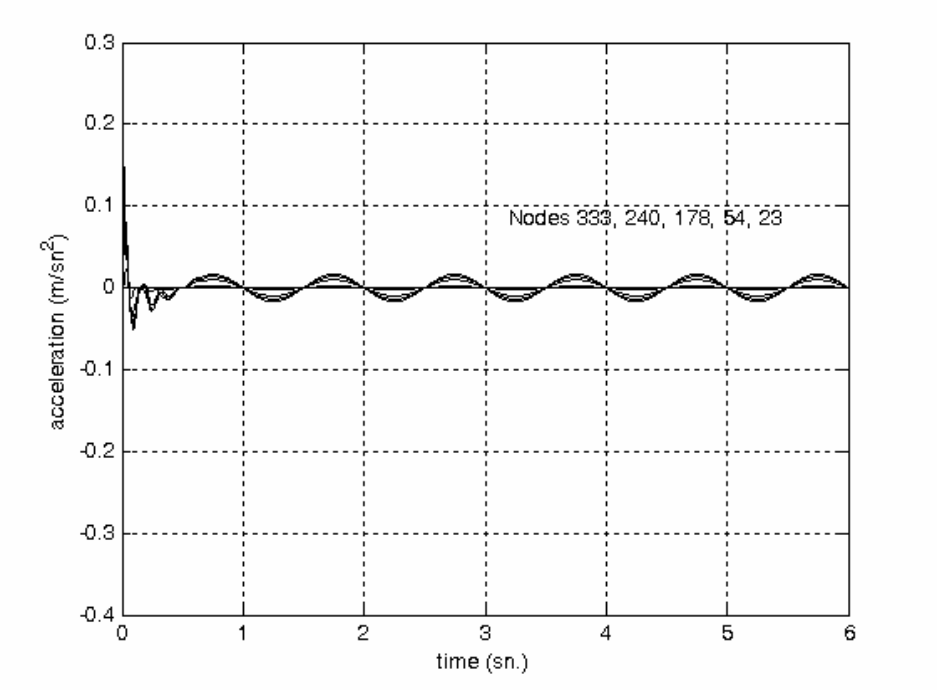
(a) cross-section 1



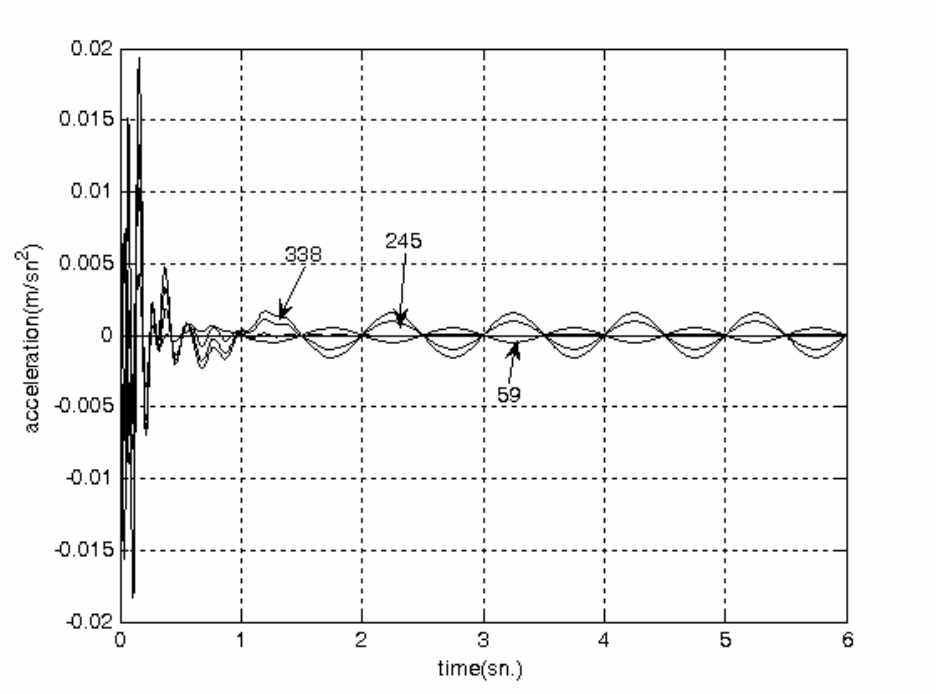
(b) cross-section 2



(c) cross-section 3



(d) cross-section 4



(e) cross-section 5

Figure 5.16 Calculated accelerations in the Y global direction for the second nodes of the elements lying on the selected cross-sections

5.5. Verification Example 4:

In the following verification example, dynamic finite element analysis of a linear elastic circular soil medium around a section of a linear elastic rigid pile is performed. Hence, the basic parts of the nonlinear dynamic FEM code for the soil-pile interaction problem will have been verified excluding the constitutive model part.

5.5.1. Construction of the Finite Element Mesh

The finite element mesh shown in Figure 5.17 is constructed by four noded two dimensional plane strain elements which behave linear elastically according to Hooke's Law. The diameter of the pile is 0.3 m. and a circular area having a diameter of approximately (20 x pile diameter) around the pile is meshed by using two dimensional elements. There are 12 bands of elements around the pile. The angle

between the radial lines is taken as15° and each radial line is divided into sections in order to construct a finer mesh in the region closer to the pile. Bands 1 through 5 cover the distance of 0.5 m. from the pile and then come the bands 6 through 9 covering a distance of 1 m. starting from band 5 and finally, bands 10 through 12 are constructed covering a distance of 1.5m. starting from band 9.

The nodes lying on the outside boundary of the finite element mesh is constrained both in the x and y directions. The nodes lying on the boundary of the pile with the soil medium are constrained in the global y direction, however in the x direction, prescribed sinusoidal displacements are applied. Numbering scheme for the nodes and the elements is given in Figure 5.18. In addition, prescribed sinusoidal displacement, velocity and acceleration values for the x degree of freedom of the nodes on the pile boundary are given in Figures 5.19, 5.20 and 5.21, respectively. Prescribed displacements, velocities and accelerations are applied at frequencies of 0.25Hz. and 1Hz. Necessary material properties are taken as:
Unit weight: 20 KN/m^3, Saturated unit weight : 21 KN/m^3, Unit weight of water: 9.81 KN/m^3, Elasticity modulus: 50000KN/m^2, Poisson's ratio: 0.35

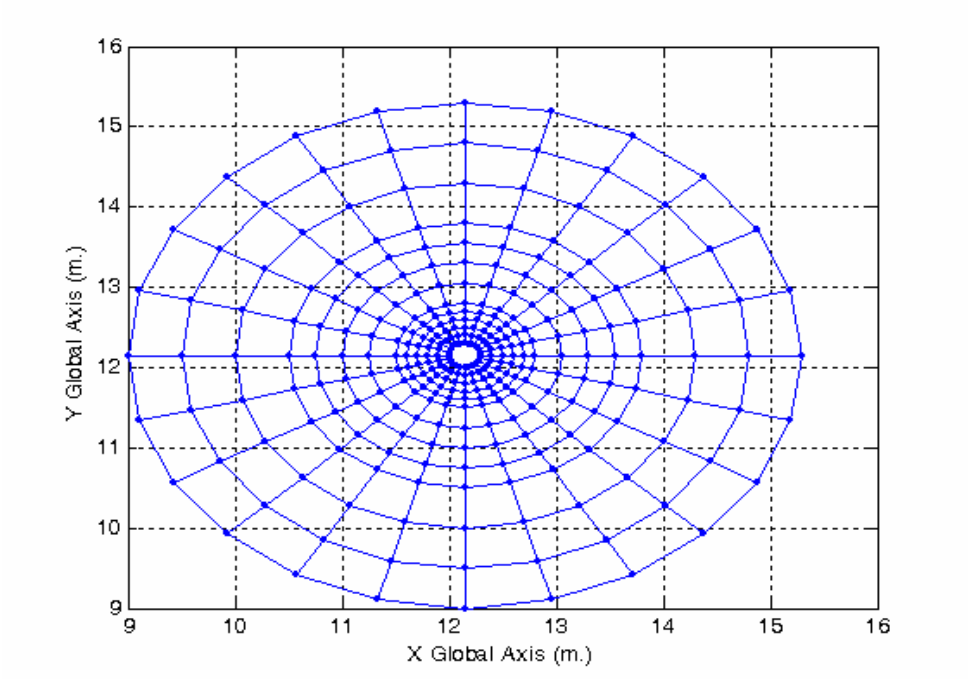


Figure 5.17 Undeformed Finite Element Mesh

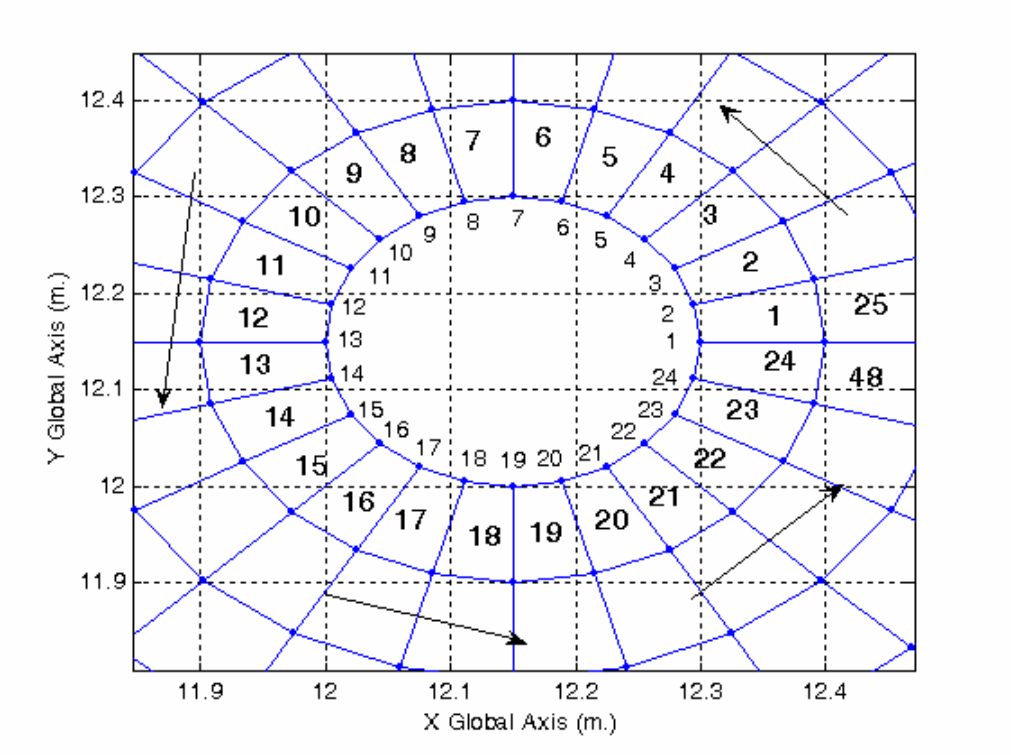


Figure 5.18 Numbering Scheme for the Nodes and the Elements

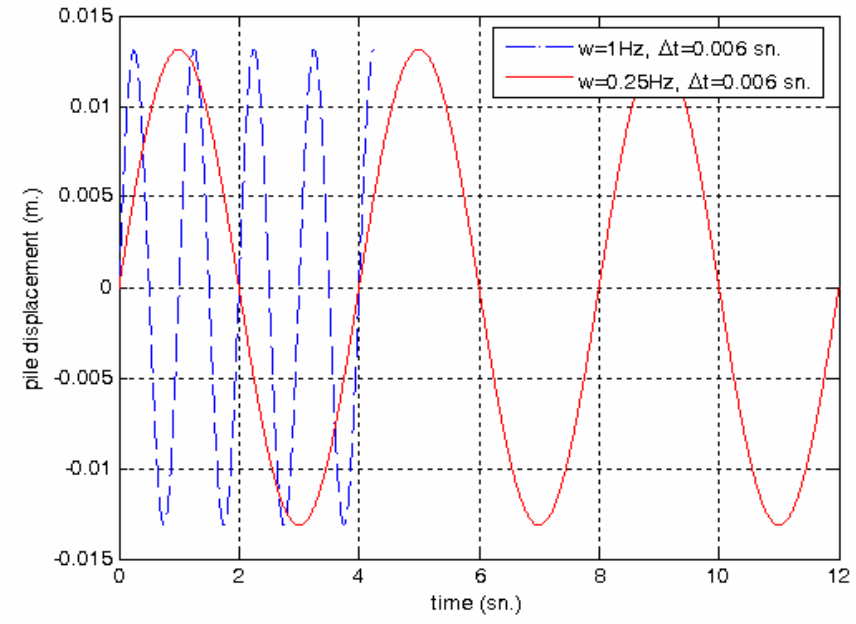


Figure 5.19 Prescribed Pile Displacements for w=1Hz and w=0.25Hz.

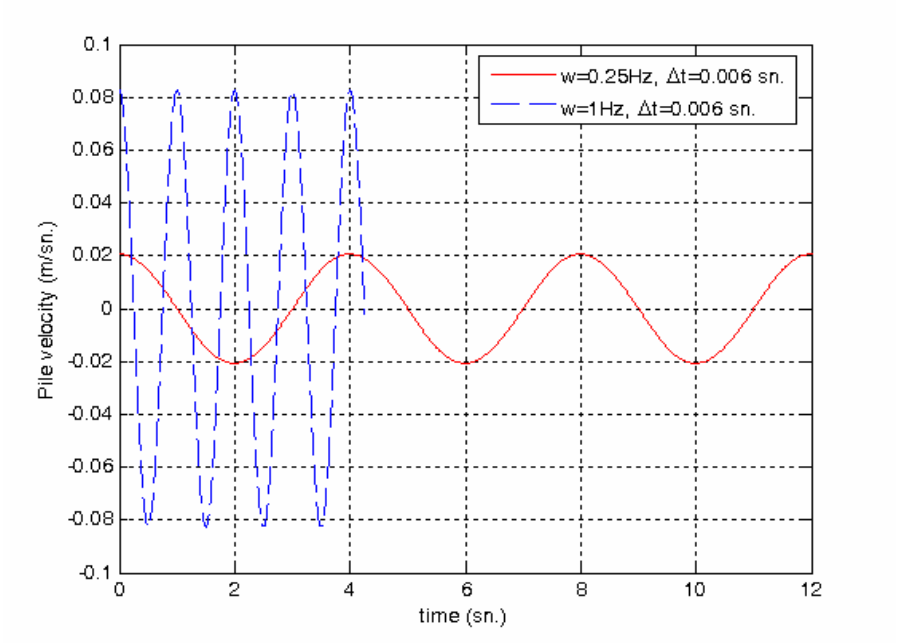


Figure 5.20 Prescribed Pile Velocity for w=1Hz. and w=0.25Hz.

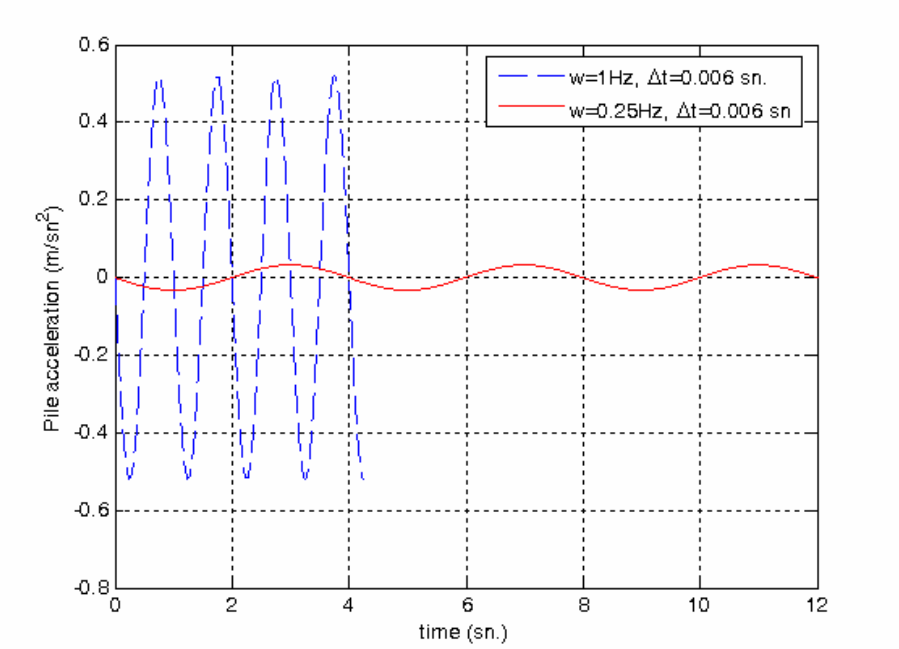


Figure 5.21 Prescribed Pile Accelerations for w=1Hz. and w=0.25Hz.

5.5.2 Analysis Results

For the rigid pile section problem, dynamic linear analysis is an important stage before starting the dynamic nonlinear analysis in which an advanced soil constitutive model is used for simulating the soil behaviour under dynamic loading conditions. Analysis of the linear case is performed by taking stripes of elements in different directions around the rigid pile section as shown in Figure 5.22. This type of analysis will be instructive to observe the change in behaviour with distance all around the rigid pile section.

Setting of the boundary conditions is an important stage of FEM analysis. In this problem, sinusoidal displacements are applied at the x degree of freedom of the nodes on the boundary of the pile section with the sand medium. When the pile moves to the right in the global X direction, the sand medium is compressed in front of the pile and extended in the back due to the applied prescribed sinusoidal displacements. In real life, the sand medium in back of the pile would present the same behaviour by following the pile and tensile stresses would occur in this case too. Hence, the application of the boundary conditions is sensible. However, it is a known fact that soils can't carry tensile stresses. Then, it is natural to think that during the application of cyclic excitation (i.e sinusoidal prescribed displacements), if the calculated σ_x exceeds the tensile strength of the sand, failure will occur in back of the pile. In this case, failure in the back of the pile should occur faster than the failure in the front. However, this type of failure can occur when a gap occurs between the pile and the soil medium like in clays. In case of sands, no gap occurs since the sand medium will follow the pile (Pinto, P., 1998). Hence there will always be some confining pressure around the sand elements. In this case, failure is expected to occur in the back of the pile due to extension shear which is defined by the failure line in the constitutive law.

The results of the analysis for 0.25 Hz. are presented in the following figures. The variation of σ_x with time, σ_y with time and τxy with time for the eight stripes (i.e

stripes 1 through 8) of elements are given in Figures 5.23 through 5.30, Figures 5.31 through 5.38 and Figures 5.39 through 5.46, respectively.

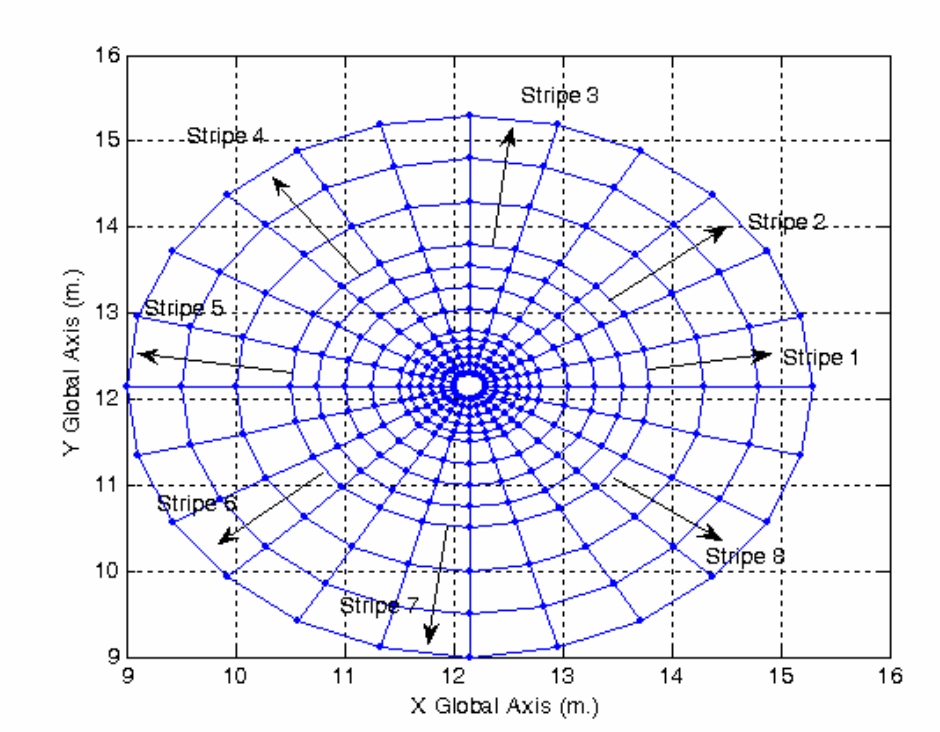


Figure 5.22 Stripes of Elements Selected for Analysis

In the following paragraphs, general characteristics of the calculated behaviour starting from t=0 until the end of initial loading will be investigated. In such a case, the pile starts to move in the positive X direction and compresses the soil elements on the right until unloading starts. The behavior can be explained by dividing the mesh into four regions:

First region (stripes 1, 2 and 3)

While moving from stripe 1 towards stripe 3, σ_x and σ_y gradually decrease while the shear stresses τ_{xy} gradually increase. At stripe 3, min σ_x and σ_y and max. τ_{xy} occur which is the expected behaviour. In addition, the sign of σ_x is negative expressing compressive stress state in the global X direction and the sign of σ_y is calculated as positive representing tensile stres state in the global Y direction.

Second region (stripes 4 and 5)

In this region, σ_x and σ_y change sign and they gradually increase while τ_{xy} gradually decreases preserving the same sign. At stripe 5, τ_{xy} reaches its minimum value while σ_x and σ_y reach their maximum values again.

Third region (stripes 6 and 7)

In this region, σ_x and σ_y gradually decrease again preserving the same sign they have in the second region while τ_{xy} gradually increases, and at the same time, changes its sign. τ_{xy} once more reaches its maximum value at stripe 7.

Fourth region (stripe 8)

In this region, σ_x becomes compressive while σ_y becomes tensile agan. At the same time τ_{xy} gradually decreases.

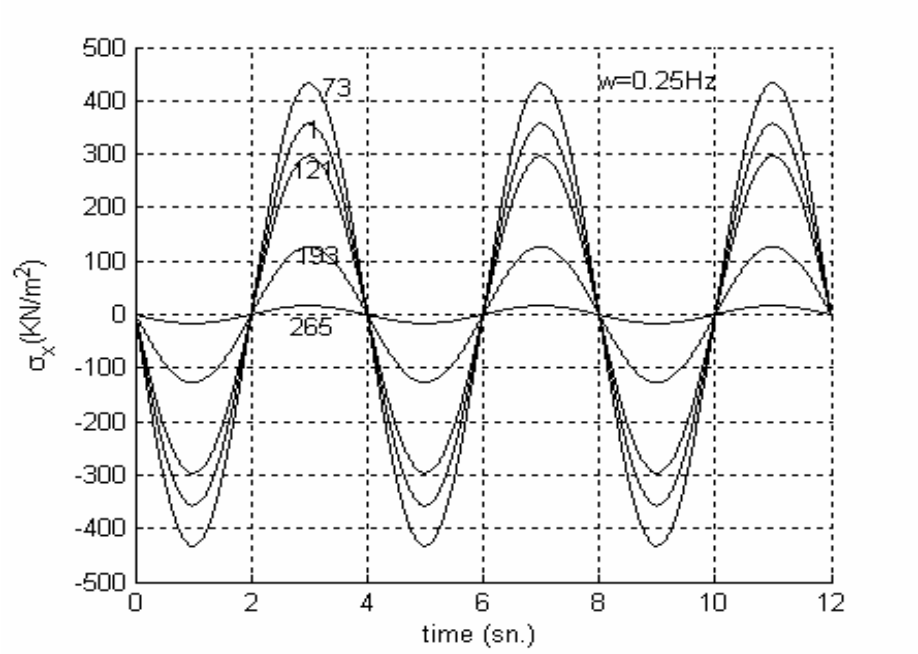


Figure 5.23 Variation of σ_x with time for stripe 1 (elements 1,73, 121, 193, 265)

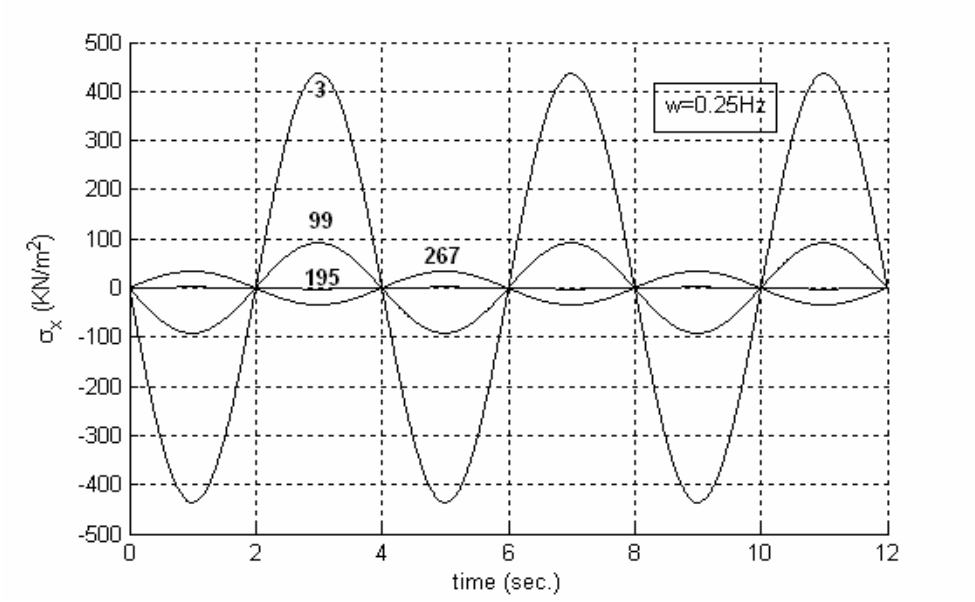


Figure 5.24 Variation of σ_x with time for stripe 2 (elements 3, 99, 195, 267)

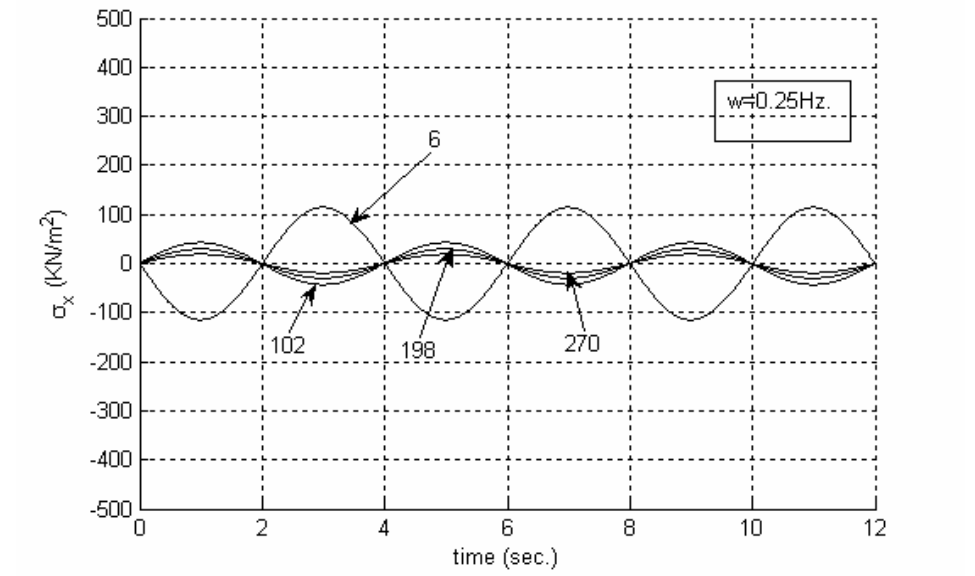


Figure 5.25 Variation of σ_x with time for stripe 3 (elements 6, 102, 198, 270)

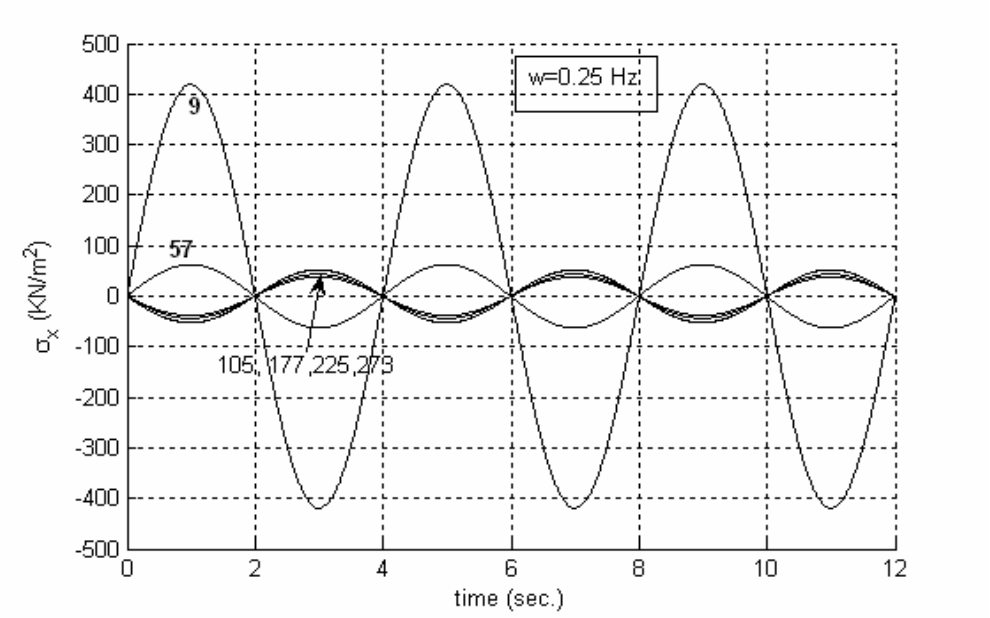


Figure 5.26 Variation of σ_x with time for stripe 4 (elements 9, 57, 105, 177, 225, 273)

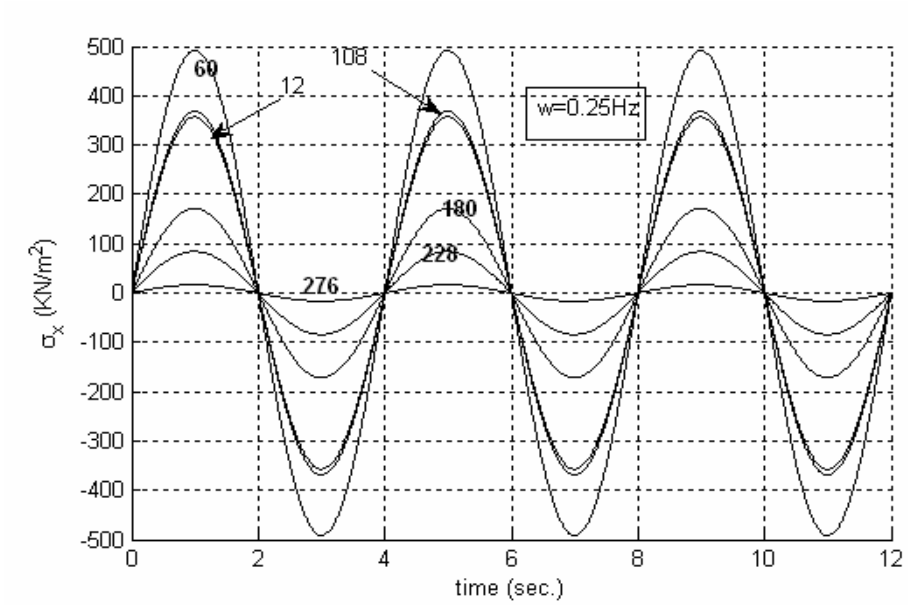


Figure 5.27 Variation of σ_x with time for stripe 5 (elements 12, 60, 108, 180, 228, 276)

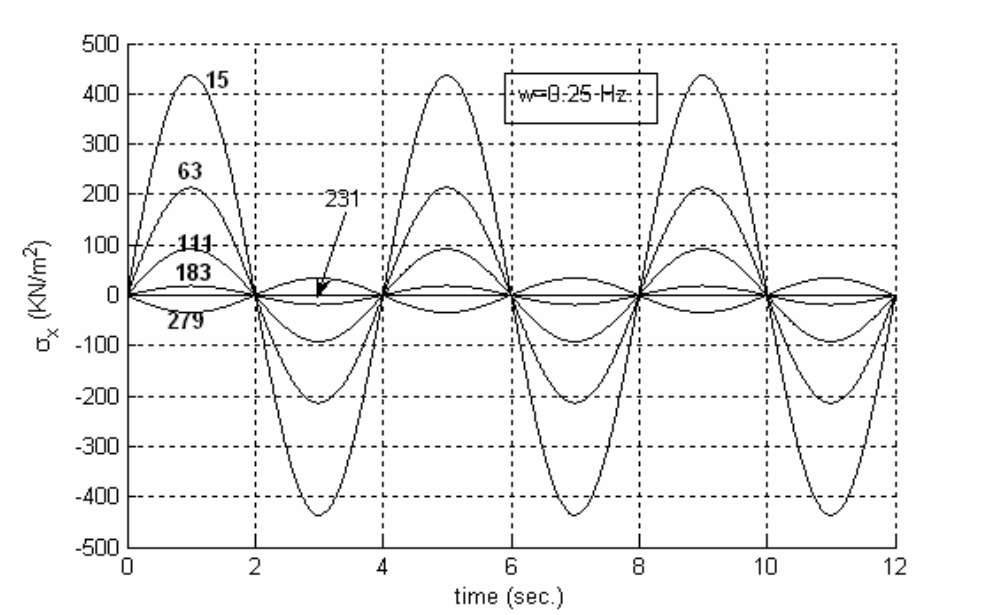


Figure 5.28 Variation of σ_x with time for stripe 6 (elements 15, 63, 111, 183, 231, 279)

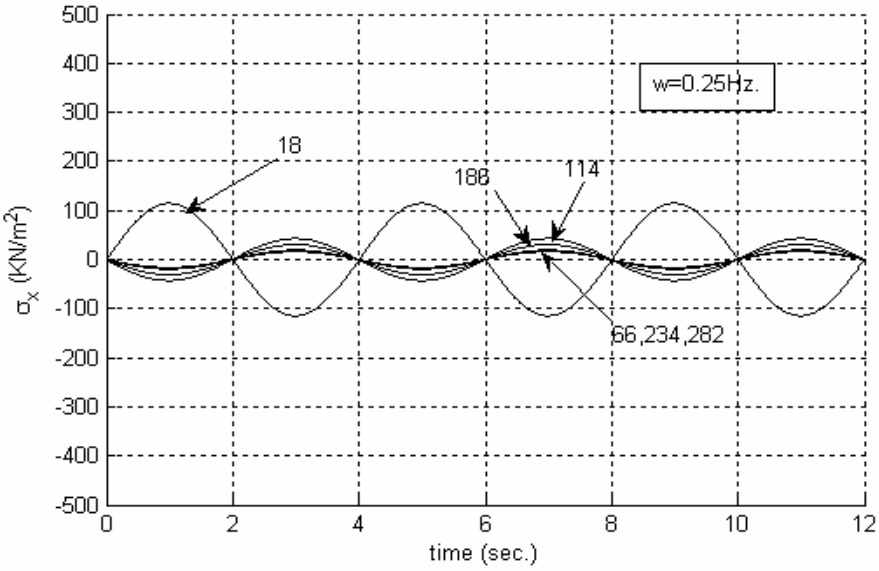


Figure 5.29 Variation of σ_x with time for stripe 7 (elements 18, 66, 114, 188, 234, 282)

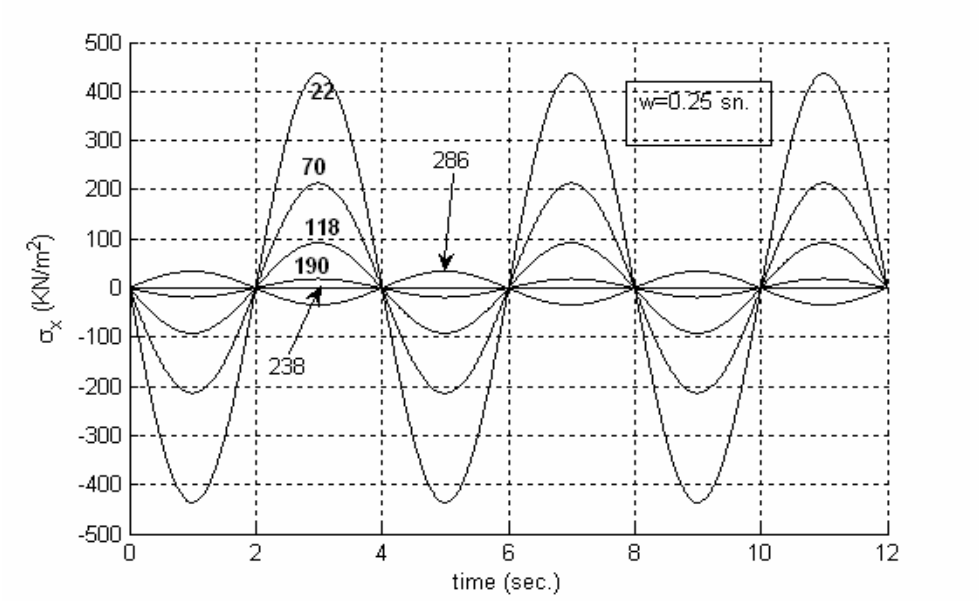


Figure 5.30 Variation of σ_x with time for stripe 8 (elements 22, 70, 118, 190, 238, 286)

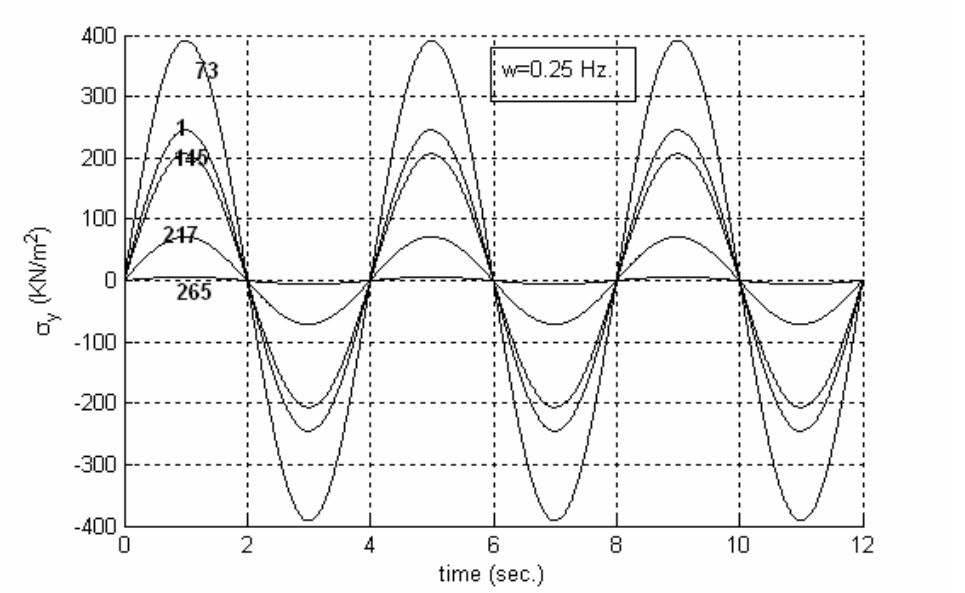


Figure 5.31 Variation of σ_y with time for stripe 1 (elements 1, 73, 145, 217, 265)

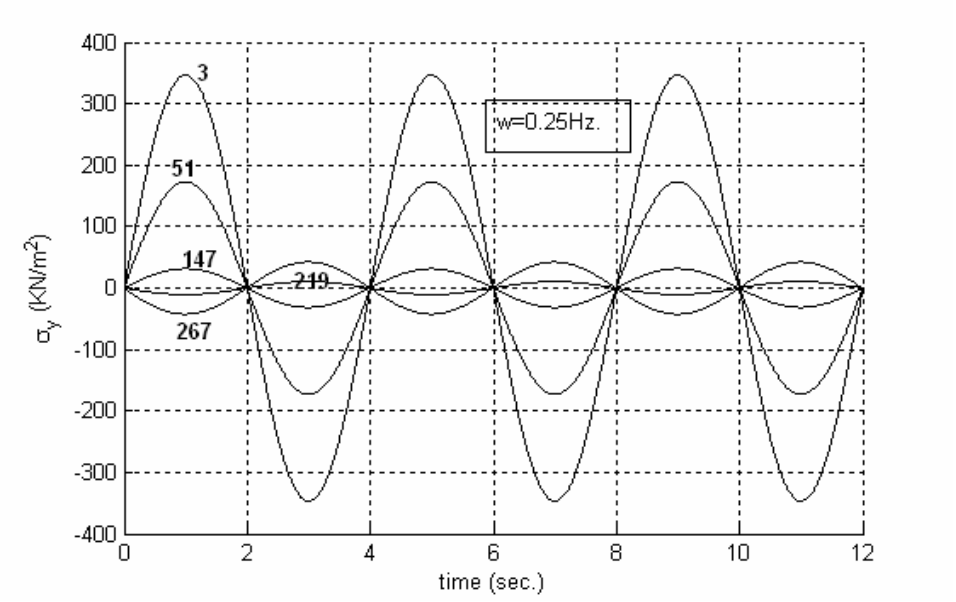


Figure 5.32 Variation of σ_y with time for stripe 2 (elements 3, 51, 147, 219, 267)

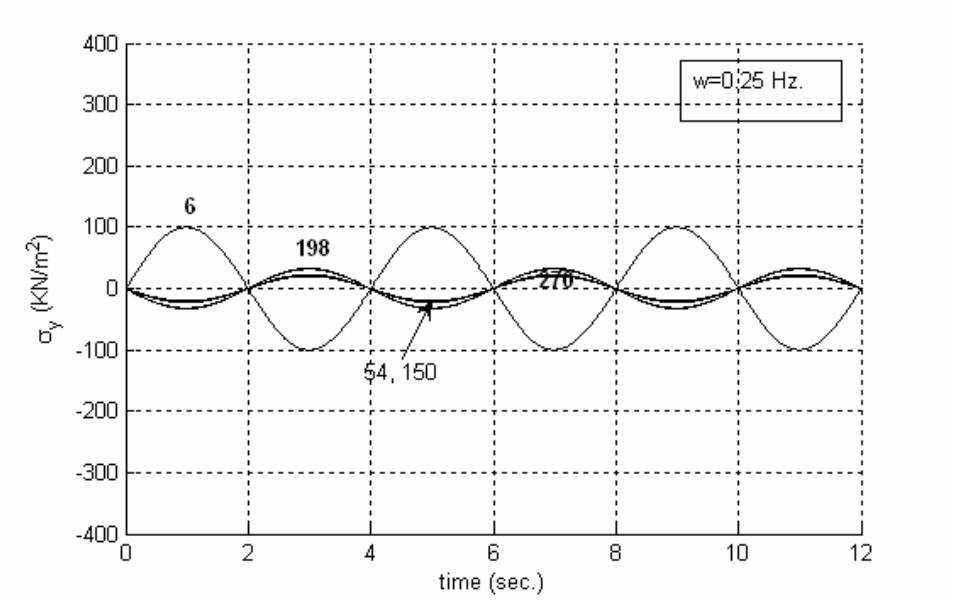


Figure 5.33 Variation of σ_y with time for stripe 3 (elements 6, 54, 150, 198, 276)

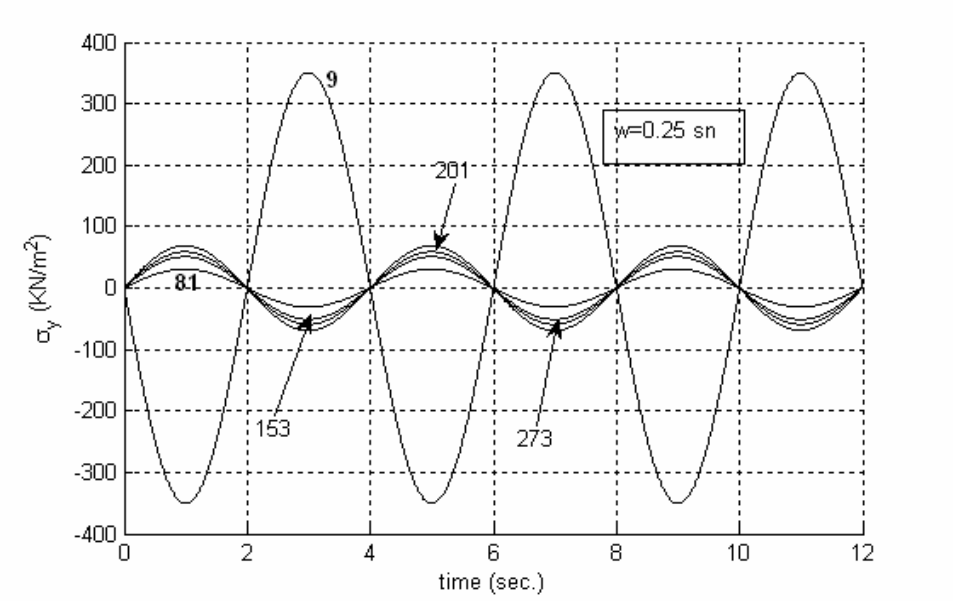


Figure 5.34 Variation of σ_y with time for stripe 4 (elements 9, 81, 153, 201, 273)

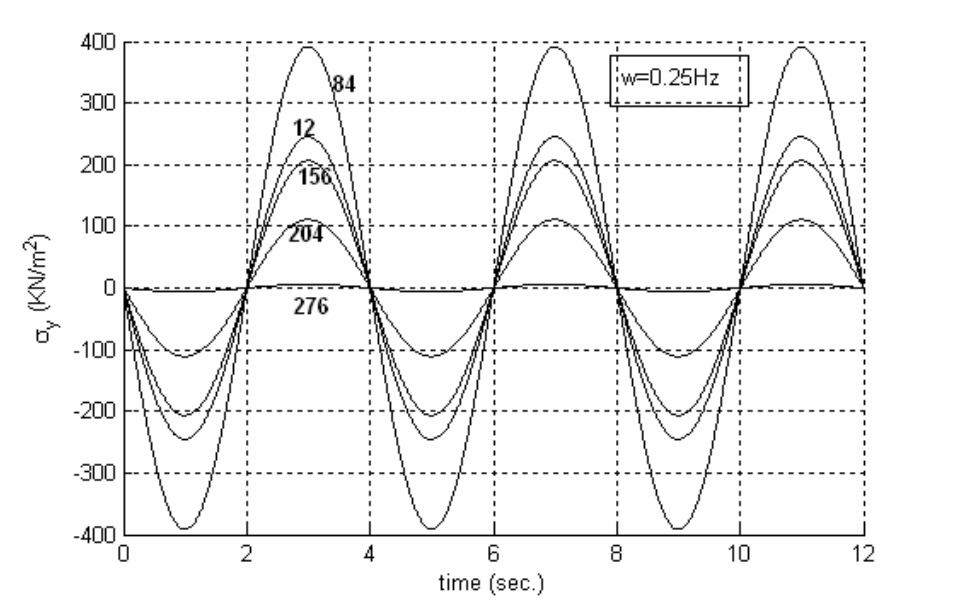


Figure 5.35 Variation of σ_y with time for stripe 5 (elements 12, 84, 156, 204, 276)

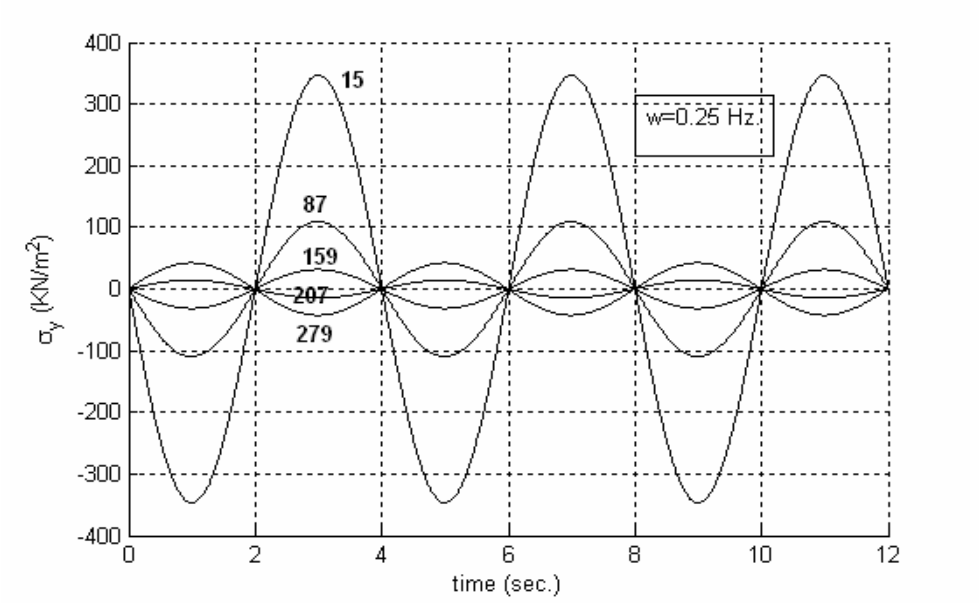


Figure 5.36 Variation of σ_y with time for stripe 6 (elements 15, 87, 159 207, 279)

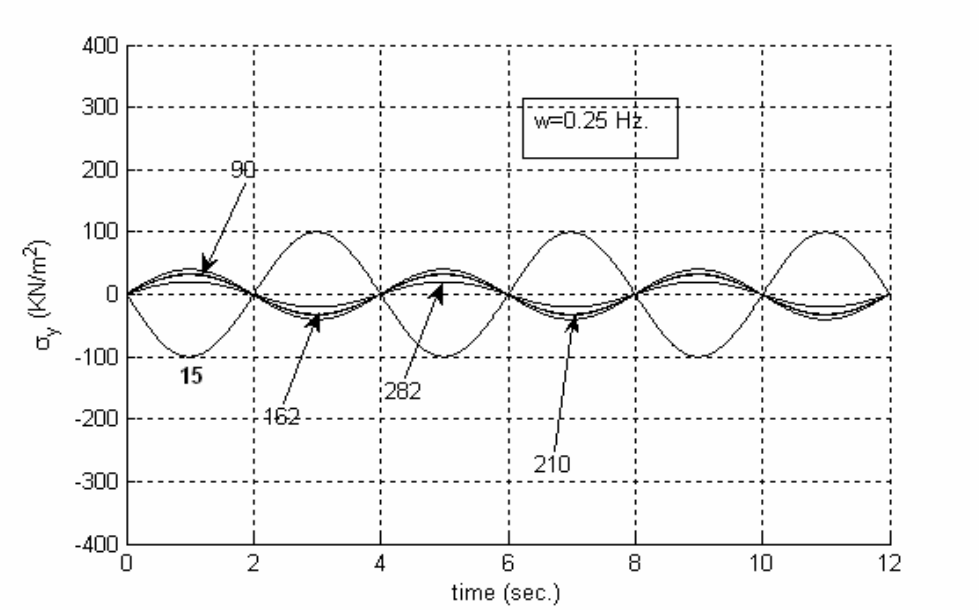


Figure 5.37 Variation of σ_y with time for stripe 7 (elements 18, 90, 162, 210, 282)

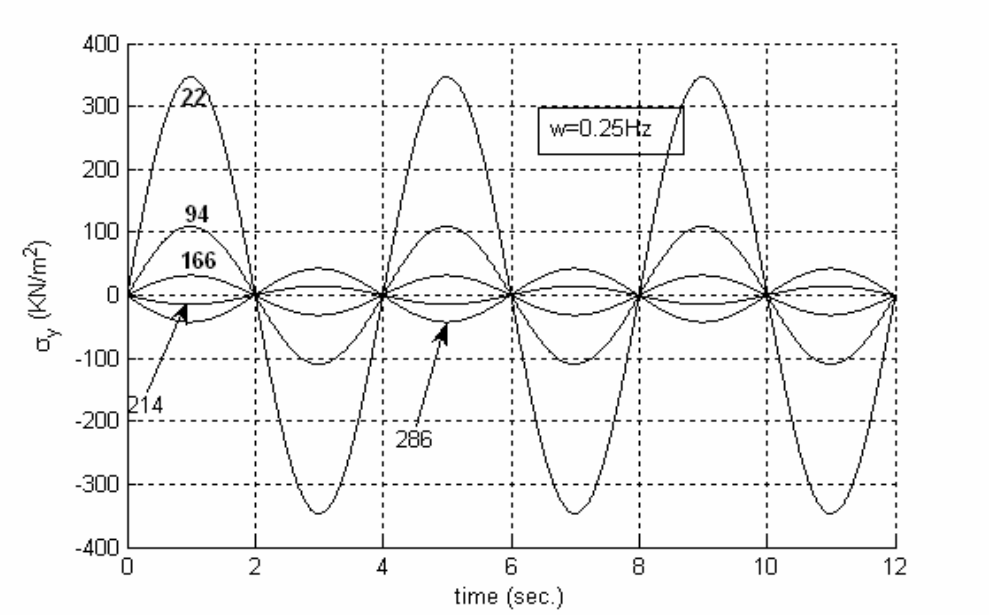


Figure 5.38 Variation of σ_y with time for stripe 8 (elements 22, 94, 166, 214, 286)

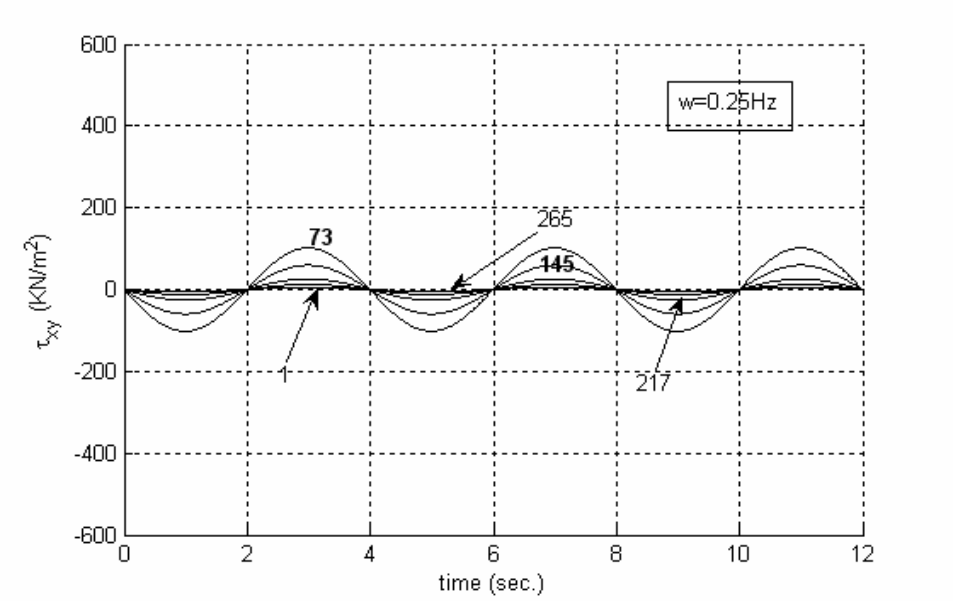


Figure 5.39 Variation of τ_{xy} with time for stripe 1 (elements 1, 73, 145, 217, 265)

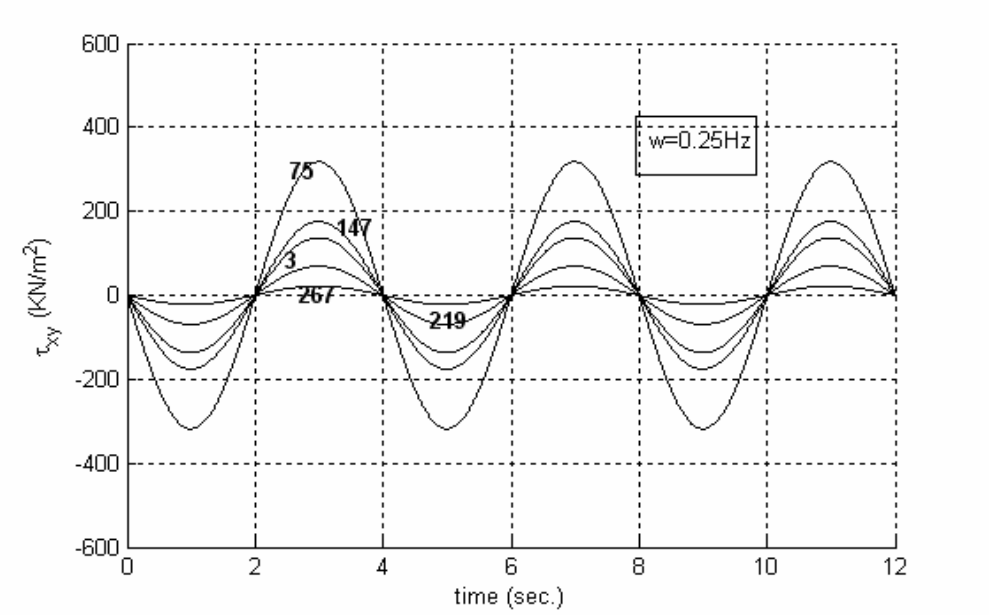


Figure 5.40 Variation of τ_{xy} with time for stripe 2 (elements 3, 75, 147, 219, 267)

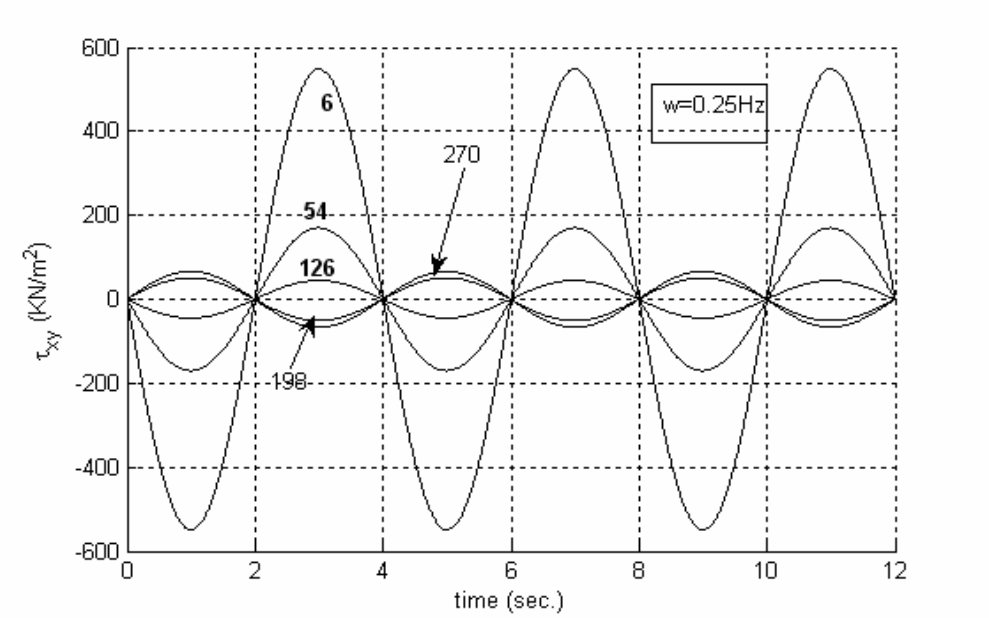


Figure 5.41 Variation of τ_{xy} with time for stripe 3 (elements 6, 54, 126, 198, 270)

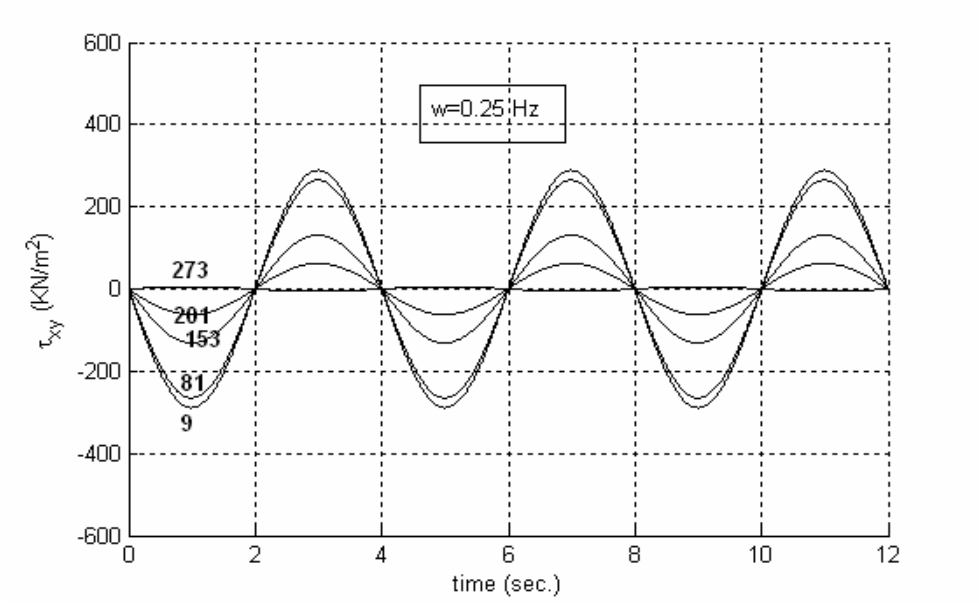


Figure 5.42 Variation of τ_{xy} with time for stripe 4 (elements 9, 81, 153, 201, 273)

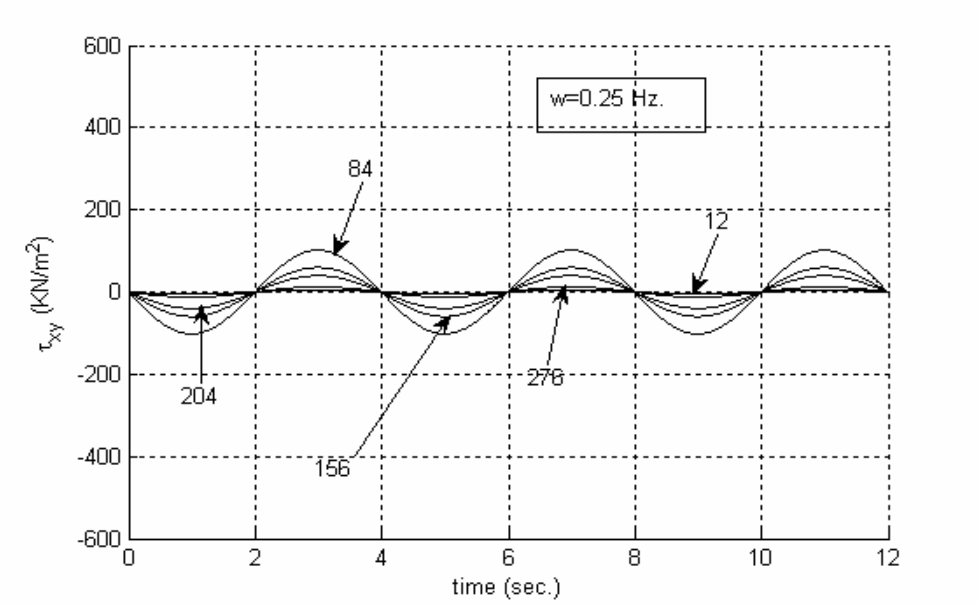


Figure 5.43 Variation of τ_{xy} with time for stripe 5 (elements 12, 84, 156, 204, 276)

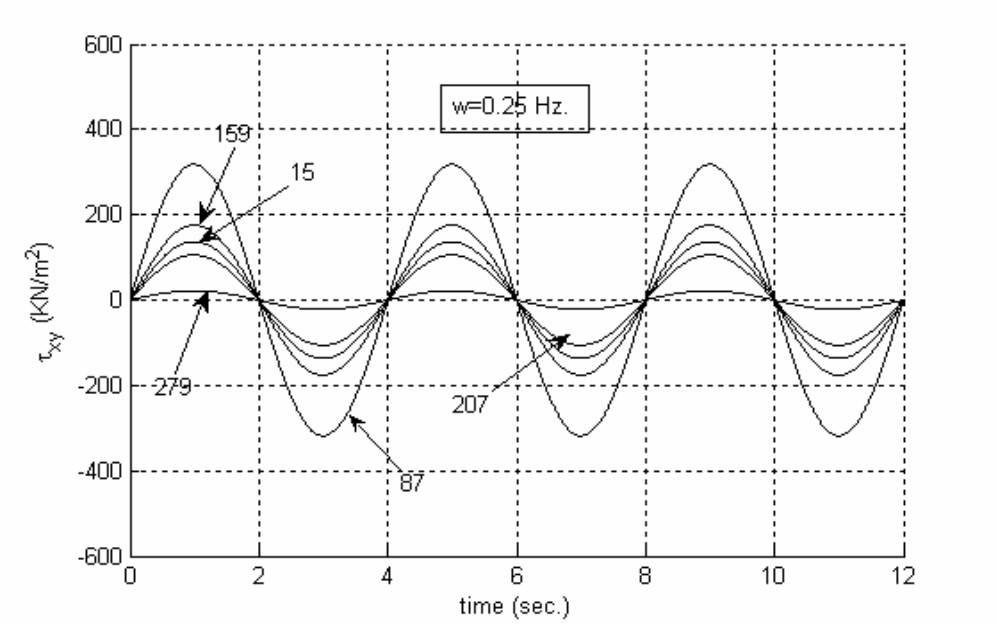


Figure 5.44 Variation of τ_{xy} with time for stripe 6 (elements 15, 87, 159, 207, 279)

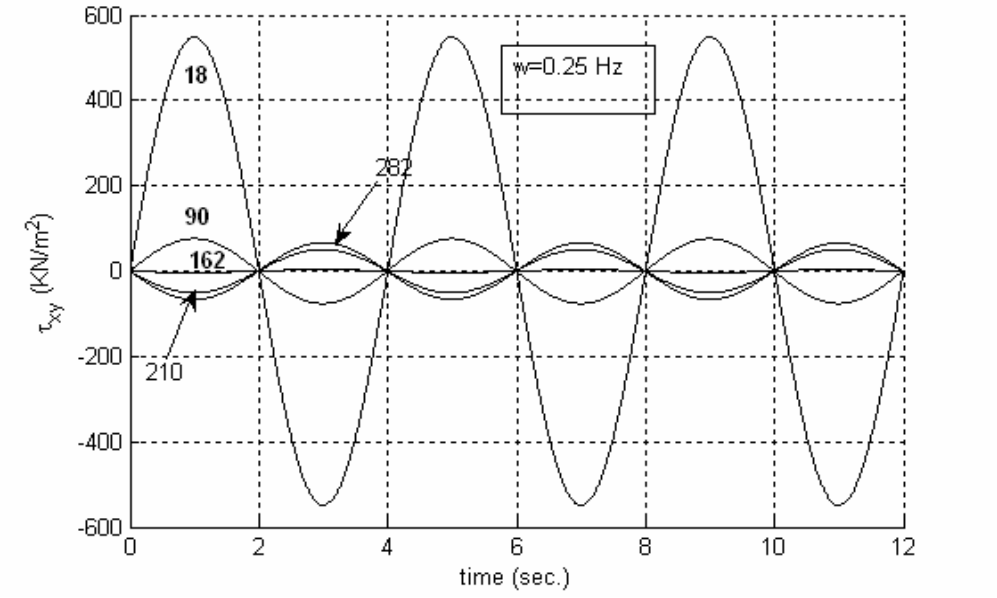


Figure 5.45 Variation of τ_{xy} with time for stripe 7 (elements 18, 90, 162, 210, 282)

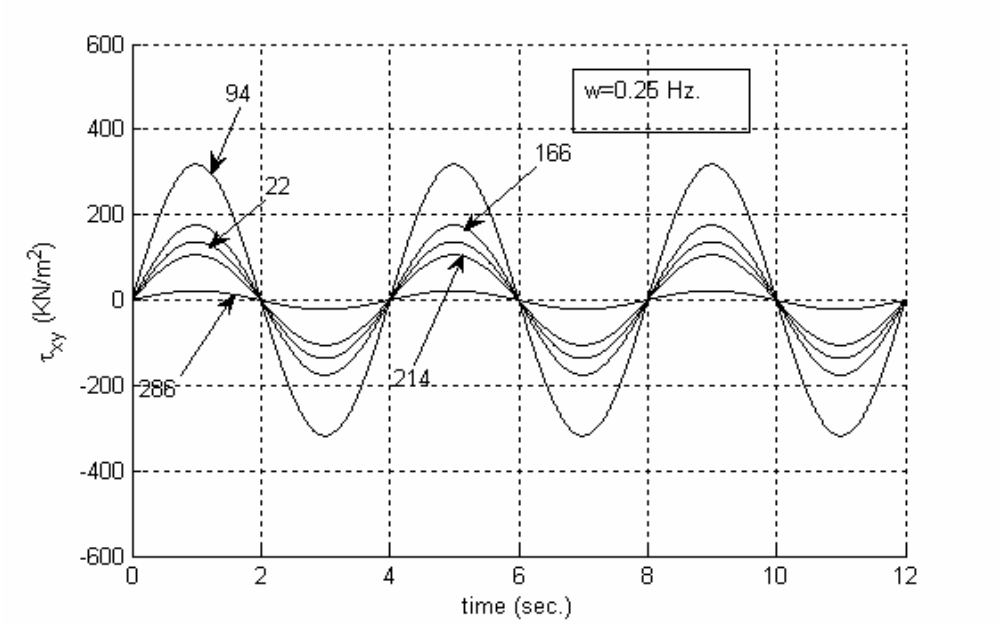
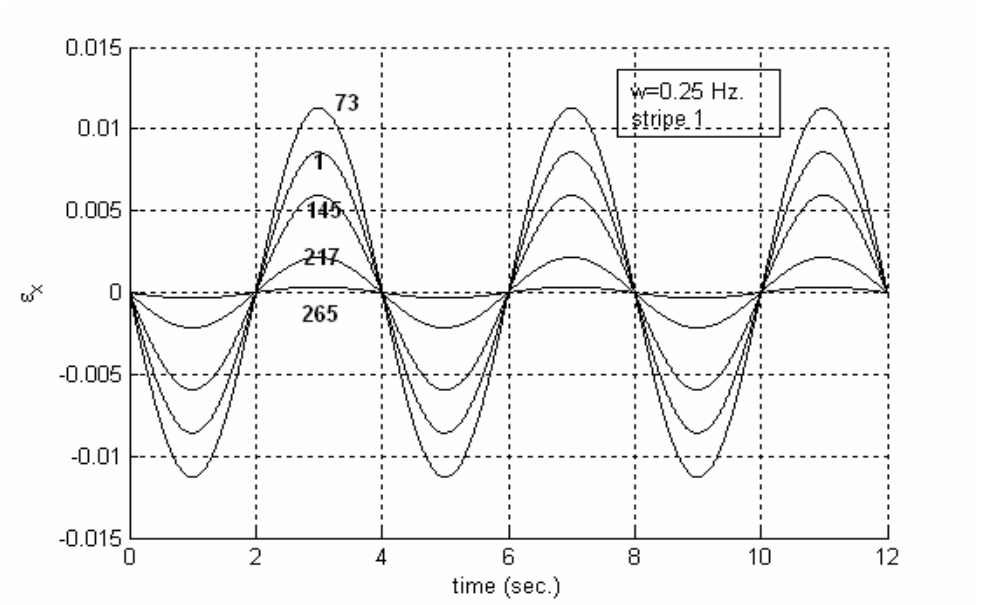
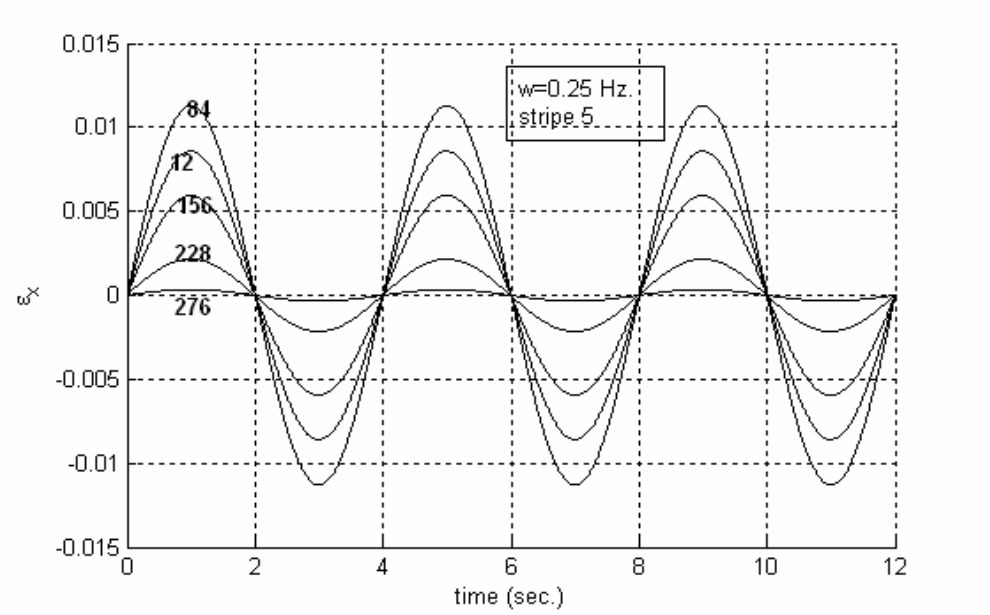


Figure 5.46 Variation of τ_{xy} with time for stripe 8 (elements 22, 94, 166, 214, 286)

Figure 5.47 demonstrate the the variation of strain in the x-direction for the elements on stripes 1 and 5 emphasizing that the signs of the strains are also compatible with the expected, compressive strains being negative and the extension strains being positive.



(a)



(b)

Figure 5.47 Variation of εx with time for (a) stripe 1, (b) stripe 5

Figures 5.48 and 5.49 demonstrate the variation of σx and τxy with the distance from the pile for each stripe, respectively. σx and τxy data represent the 170. time step which corresponds approximately to the end of the initial loading. According to these figures, normal and shear stresses gradually decrease while moving away from the pile which is the expected behavior. However, some elements approximately in the 0.2 m. distance from the pile on stripes 1 and 5 present a different behaviour which may be due to stress accumulation. Figure 5.50 representing the deformed mesh under an exaggerated loading condition, demonstrate all the behavior mentioned.

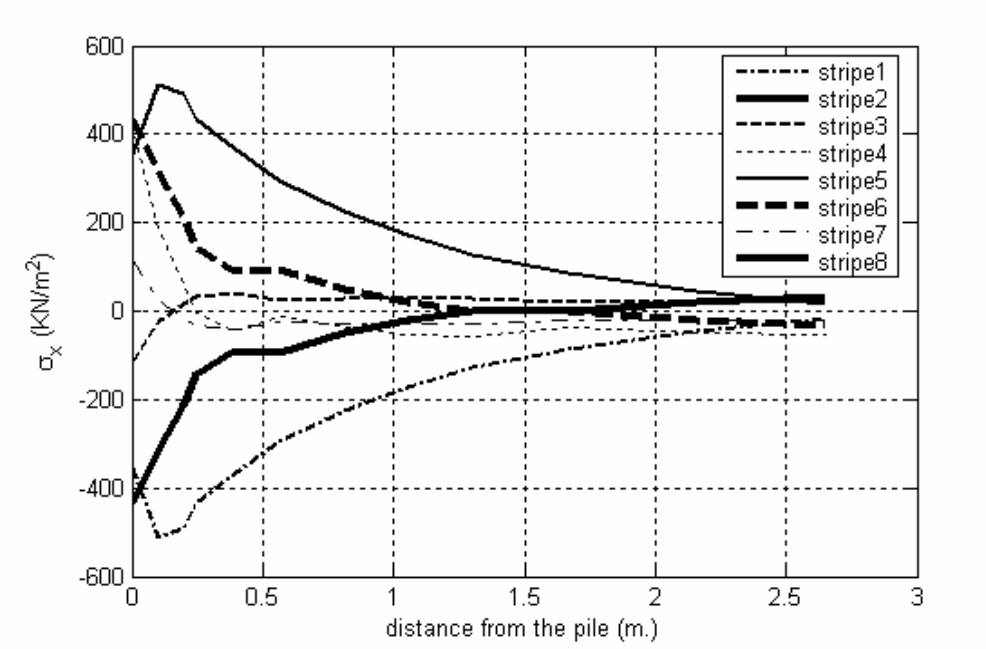


Figure 5.48 Variation of σx with distance from the pile

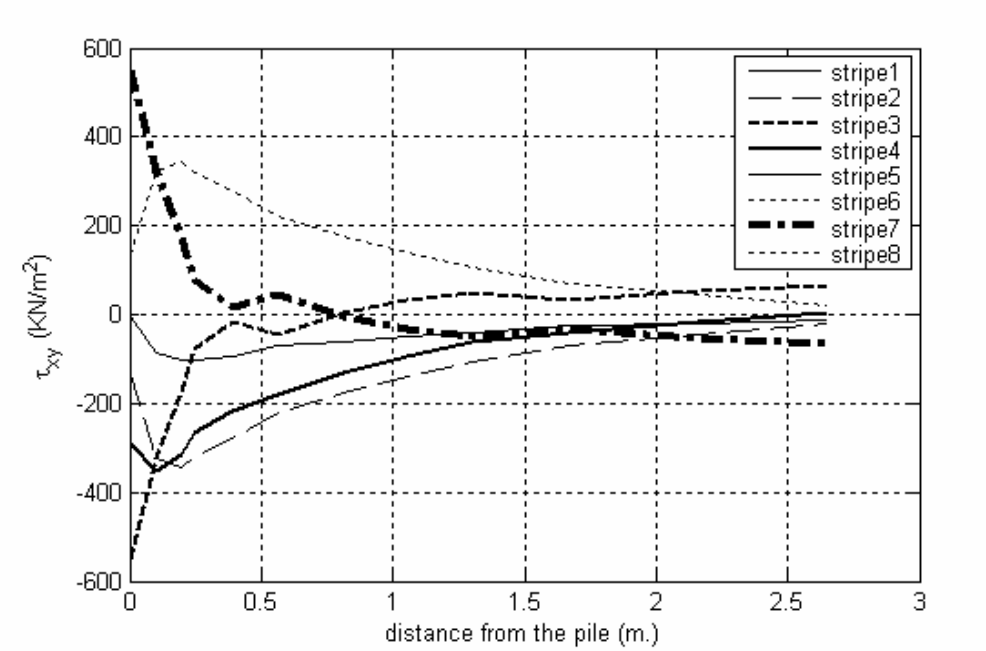


Figure 5.49 Variation of τ_{xy} with distance from the pile

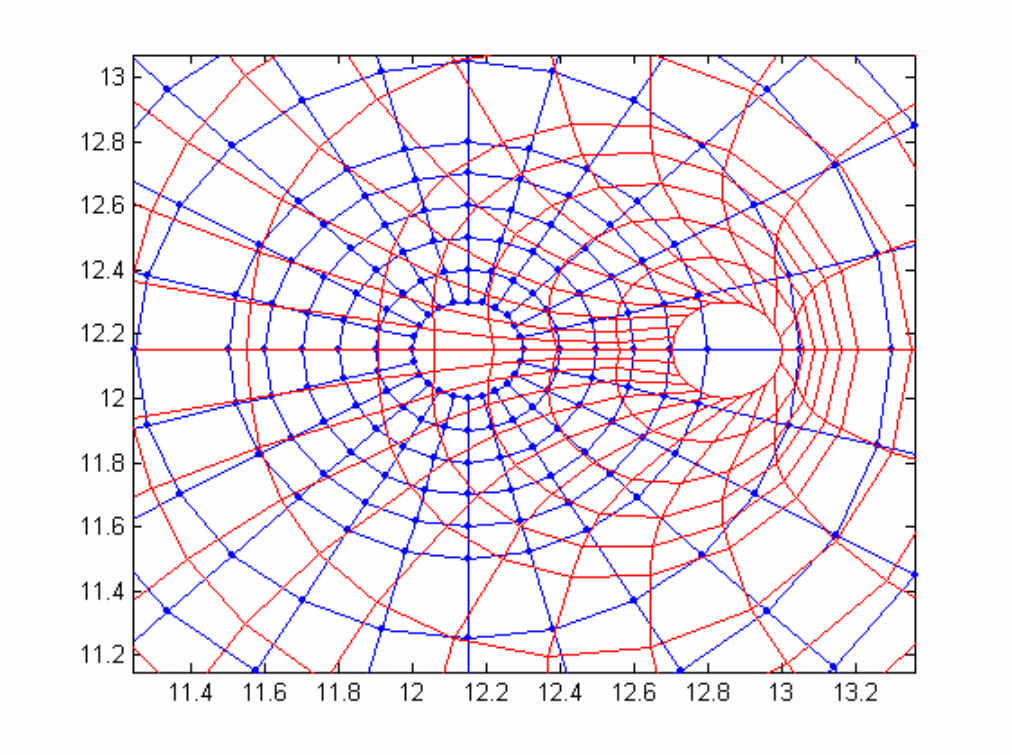


Figure 5.50 Deformed mesh under exaggerated loading conditions

CHAPTER SIX

SOIL CONSTITUTIVE MODEL

6.1 Introduction

Liquefaction and cyclic mobility are considered among the major phenomena in which non-linear behavior and cumulative damage effects can occur and hence, a full modelling of this soil behavior by means of computational techniques is necessary.

First of all, saturated soils are two-phase media: a soil skeleton and water occupying the pores within the skeleton. The geotechnical engineering problems emerge basically from the interaction of these two parts under various loading and boundary conditions (i.e. static/dynamic loading, drained/undrained assumptions).

Solution of boundary value problems of saturated sands under dynamic loading conditions requires numerical methods. Such methods can succeed in achieving this task provided that the following aspects of the solution methodology are properly handled:

1. Description of a mathematical model for fluid-skeleton interaction such as Biot's equations.
2. Definition of an adequate soil constitutive model
3. Development of an efficient computational scheme (usually the finite element method) (Zienkiewicz et al., 1988).

Deformation of saturated cohesionless soils undergoing large deformations due to dynamic loads may be defined either by the term "liquefaction" or "cyclic mobility". In general "liquefaction" refers to large and permanent deformations (on the order of several meters) of saturated cohesionless soils. "Cyclic mobility", on the other hand, is used for limited larger deformations of the soil as the excess pore water pressure attains high values temporarily.

One should note that soil layers surrounding a pile foundation generally exhibits cyclic mobility type of behavior undergoing alternating cycles of contraction and dilation, which may result in temporary increase of excess pore water pressure ratio up to %100 followed by the generation of negative excess pore water pressure, which causes strength recovery. Therefore, soils around a laterally displacing pile do not develop full liquefaction due to the action of the pile itself (i.e. inertial interaction).

Because of the above mentioned soil behavior around the laterally loaded pile, the strain space plasticity model (Iai et al., 1992) has been chosen as the preferred soil constitutive model of this thesis study since it is capable of modeling both contractive and dilative soil behavior under dynamic loading. Details of the strain space plasticity model are given in the following subsections along with brief presentation of a literature survey regarding soil plasticity models.

6.2 Dynamic Sand Behavior Under Monotonic Loading Conditions

Soil plasticity models attempt to capture various aspects of sand behavior under dynamic loading. Although dynamic conditions are of interest, sand response under monotonic loading is also important since strength and stiffness degradations are described with respect to monotonic parameters.

6.2.1 Sand Behavior Under Drained Monotonic Loading Conditions

A deposit of sand is composed of an assemblage of particles in equilibrium where intergranular forces are transmitted through contact points. When shear stress is applied, the resulting deformation is always accompanied by a volume change in drained condition.

This shear-induced volume change occurs as a result of the rearrangement of the sand grains. This rearrangement usually occurs in the form of slip-down and rollover particle motion modes. The slip-down movement of the sand grains tends to reduce the volume by repacking the sand aggregate into a denser state for loose to medium-

dense sands. When slip-down motion of sand grains occurs, they usually settle down in the voids and do not follow the direction of shearing. Hence, slip-down motion takes place without mobilizing large shear strains. For this reason, it is meaningful to observe volume reduction at the earlier stage of shear tests performed on sand samples either in loose or dense state. Exluding earlier stages of shear tests on dense sands; this mode of particle movement is usually seen in loose sand samples. On the other hand, rollover motion mode tends to increase the volume and this is a general characteristic of dense sands. Rollover mode generates higher strain levels compared with the slip-down mode. Therefore, it is understandable why dilation occurs at a later stage of shear stress application where the sand is largely deformed.

Figure 6.1 shows a typical loose and dense sand behaviour under drained monotonic loading conditions. As revealed in the stress-strain curve (Figure 6.1a), during drained shearing of loose sands, the deviatoric stress increases to a maximum value accompanied by a decrease in void ratio (Figure 6.1b). There is a critical value of the void ratio at which continuous deformation occurs with no change in deviatoric stress. This value of the void ratio is called as the critical void ratio. On the other hand, when a dense sand specimen is sheared, the deviatoric stress reaches a peak value afterwhich it decreases to a value very close to the maximum deviatoric stress for the loose sand (Figure 6.1a). Volume change behavior of dense sand (Figure 6.1b) shows that dense sand decreases in volume slightly at the begining, then dilates up to a value very close to the critical void ratio.

Confining pressure is the other important factor, which affects the volume change (dilatancy) behavior of both loose and dense sand samples during drained shear. Figures 6.2 and 6.3 respectively show influence of confining pressure on loose and dense sand samples during drained shear. According to these figures, loose sands may exhibit dilative behaviour under lower confining pressures whereas dense sands may exhibit loose sand behavior showing a decrease in volume at high confining stresses (Holtz and Kovacs, 1981; Ishihara, K., 1996).

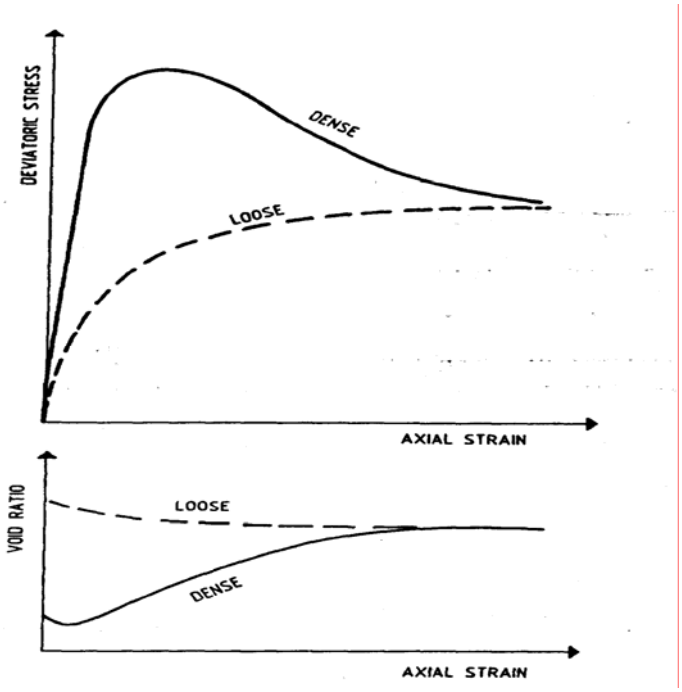


Figure 6.1 Loose and dense sand behavior under drained conditions (Pastor et al., 1985)

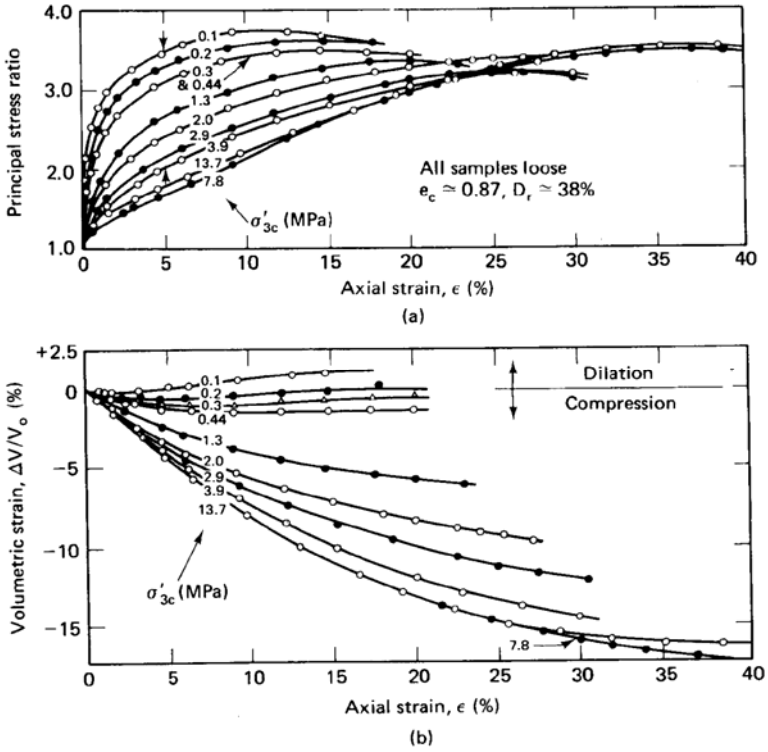


Figure 6.2 Drained triaxial test results on loose Sacremento River Sand a) Principal stress ratio versus axial strain b) Volumetric strain versus axial strain (Holtz and Kovacs, 1981)

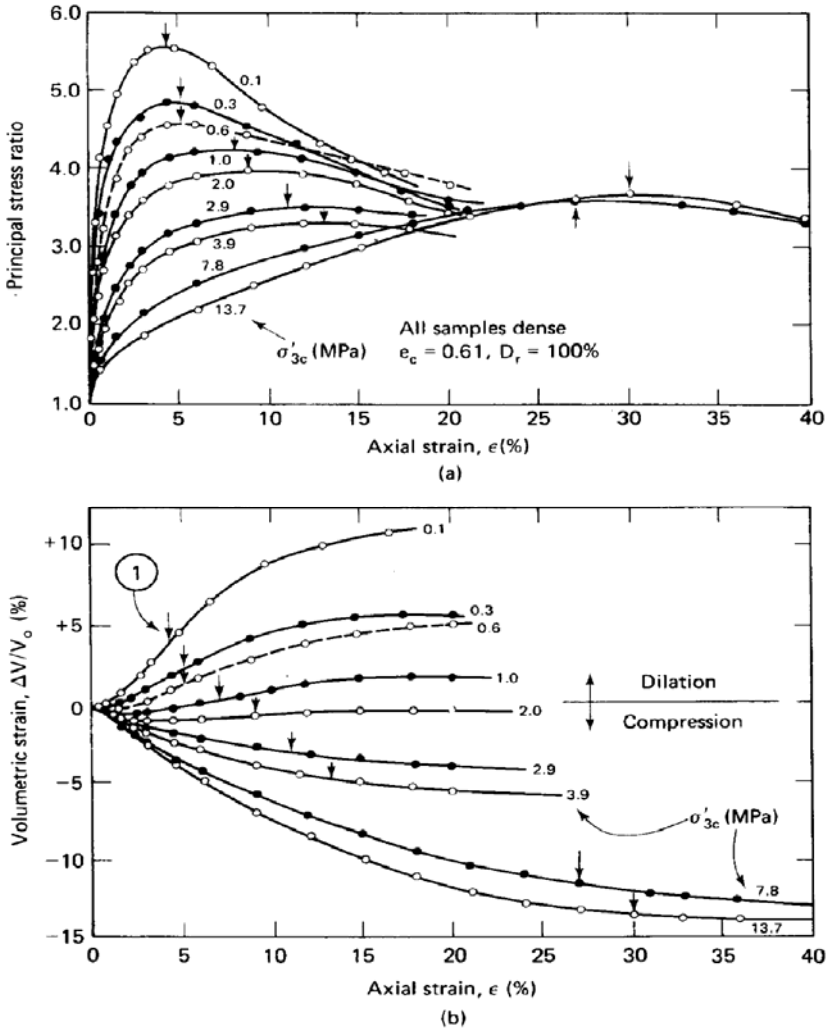


Figure 6.3 Drained triaxial test results on dense Sacremento River Sand a) Principal stress ratio versus axial strain b) Volumetric strain versus axial strain (Holtz and Kovacs, 1981)

6.2.2 Sand Behavior Under Undrained Monotonic Loading Conditions

Figure 6.4 shows different types of stress-strain relations obtained from undrained shear tests on saturated sand samples at different relative densities. According to this figure, it can be concluded that under undrained monotonic loading conditions, the sand behaviour can be grouped as:

1. Contractive (strain-softening)- Flow of the sand occurs
2. Dilative (strain-hardening)- Non-flow behavior
3. Initially contractive and then dilative-flow with limited deformation

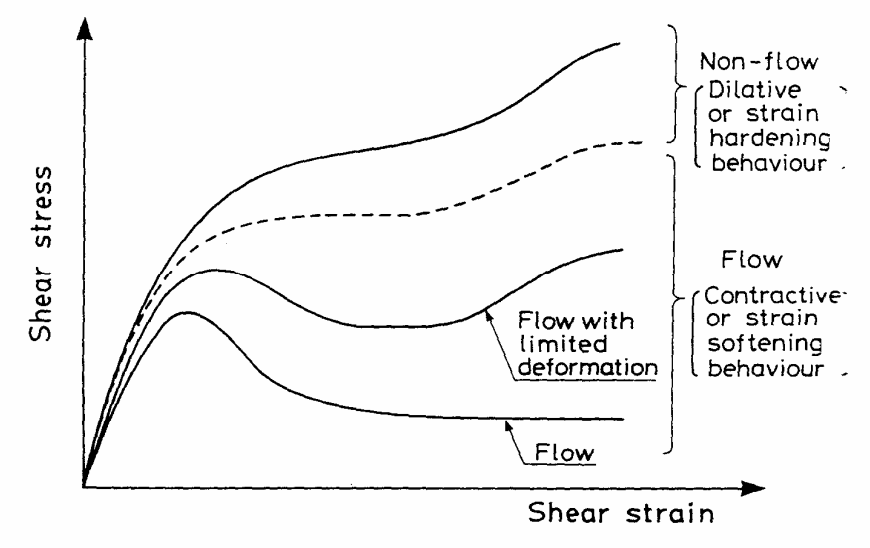


Figure 6.4 Classification of undrained behavior of sandy soils based on their contractive and dilative responses (Ishihara, K., 1996)

Additionally, Figure 6.5 helps to understand how pore pressure changes during contractive and dilative behavior of sand specimens. Figure 6.5a, 6.5b and 6.5c respectively show deviatoric stress-mean effective confining pressure, shear stress-shear strain and pore pressure-shear strain curves of loose to dense sands.

Contractive (strain-softening) sand behavior:

Under undrained conditions, loose saturated sands exhibit contractive behavior resulting in flow failure of the saturated sand. Here ‘contractive’ means that when sand specimen is sheared in the drained condition, it would contract and decrease in volume. However, under undrained conditions, this tendency to contract increases the pore pressure decreasing the mean effective stress and shear strength. In Figure 6.4, this is represented by the flow curve at the bottom of the graph. It is seen in this curve that shear stress is greatly reduced after reaching a peak value. The sudden drop in shear stress is accompanied by unlimitedly large strains under constant shear stress. This behavior is also illustrated by the curve (a) of Figures 6.5 a, b and c for monotonic loading condition.

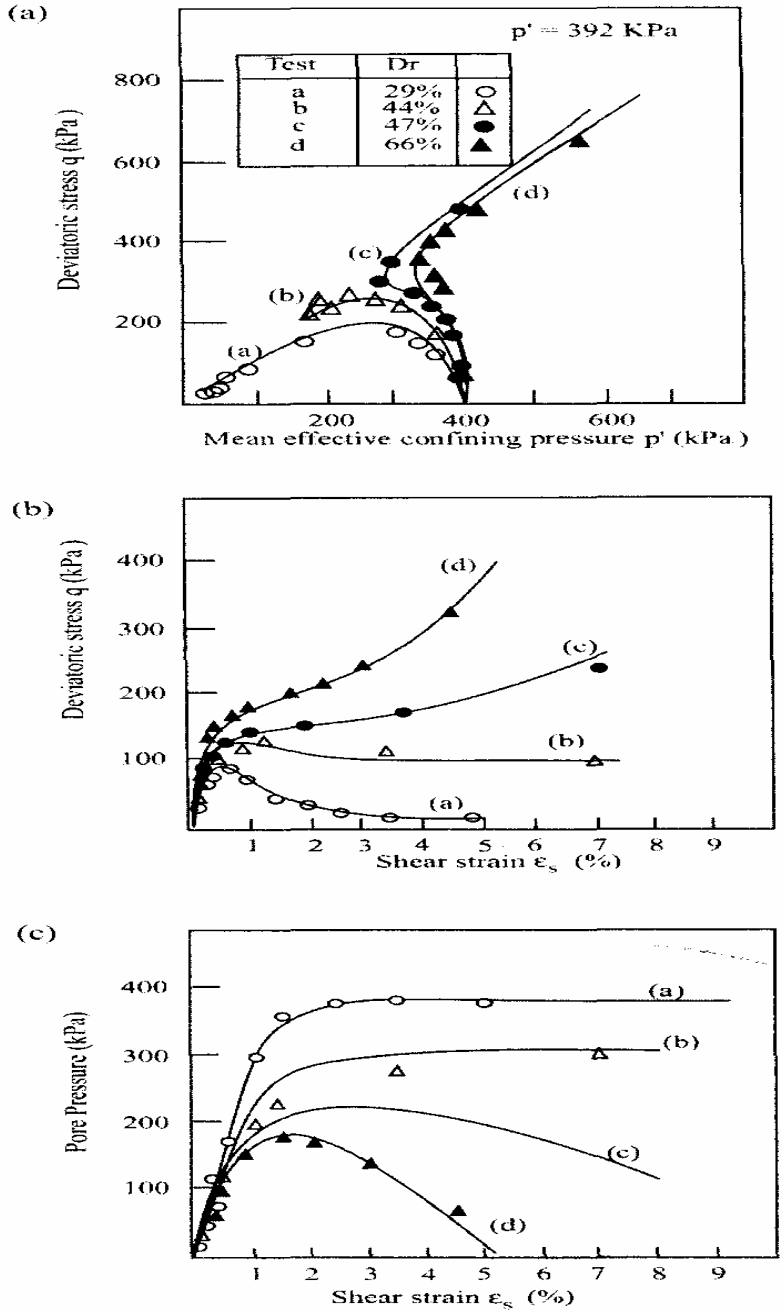


Figure 6.5 Undrained sand behavior during monotonic loading (data from Castro 1969) (a) p′-q, (b) deviatoric stress-shear strain, and (c) pore pressure -shear strain variations (Zienkiewicz et al., 1999)

Dilative (strain-hardening) behavior:

Under undrained monotonic loading conditions, dense saturated sands present dilative behavior. In this type of soil response, shear stress increases with increasing

strain and therefore no flow type failure takes place. This behavior is called dilative since skeleton of monotonically sheared sands under drained conditions would increase their volume. However, under undrained conditions, this tendency of volume increase decreases the pore pressure causing mean effective stress and shear strength to increase. Figure 6.4 shows this strain-hardening type of soil response. The curves of (c) and (d) of Figures 6.5a, 6.5b and 6.5c also illustrate dilative dense sand response under monotonic undrained loading.

Initially contractive and then dilative behavior:

Moderately dense saturated sands first show a contractive behavior at moderate shear strains but then starts to dilate as the strain increases. This kind of behavior results in flow with limited deformation (Figure 6.5b). Under drained shear conditions, volume of this sand would first decrease and then increase with increasing shear strain. However, during undrained shear, tendency to decrease volume will generate positive excess pore water pressure. This behavior is then reversed at higher shear strains causing negative excess pore water pressure development.

Similar to drained sand behavior confining pressure has a conrolling effect over the undrained response as well. Figures 6.6, 6.7 and 6.8 demonstrate this influence for loose, medium dense and dense Toyoura Sand samples at different confining pressures, respectively. One can notice in these figures that all soil samples demonstrate dilative behavior sooner or later in the tests until they reach to the steady state condition. The degree of dilation, however, depends on the confining pressure for a particular void ratio. Initially contractive behavior is increasingly suppressed as the void ratios of the samples are decreased. Note that dense sand samples dominantly exhibit dilation irrespective of the applied confining pressure.

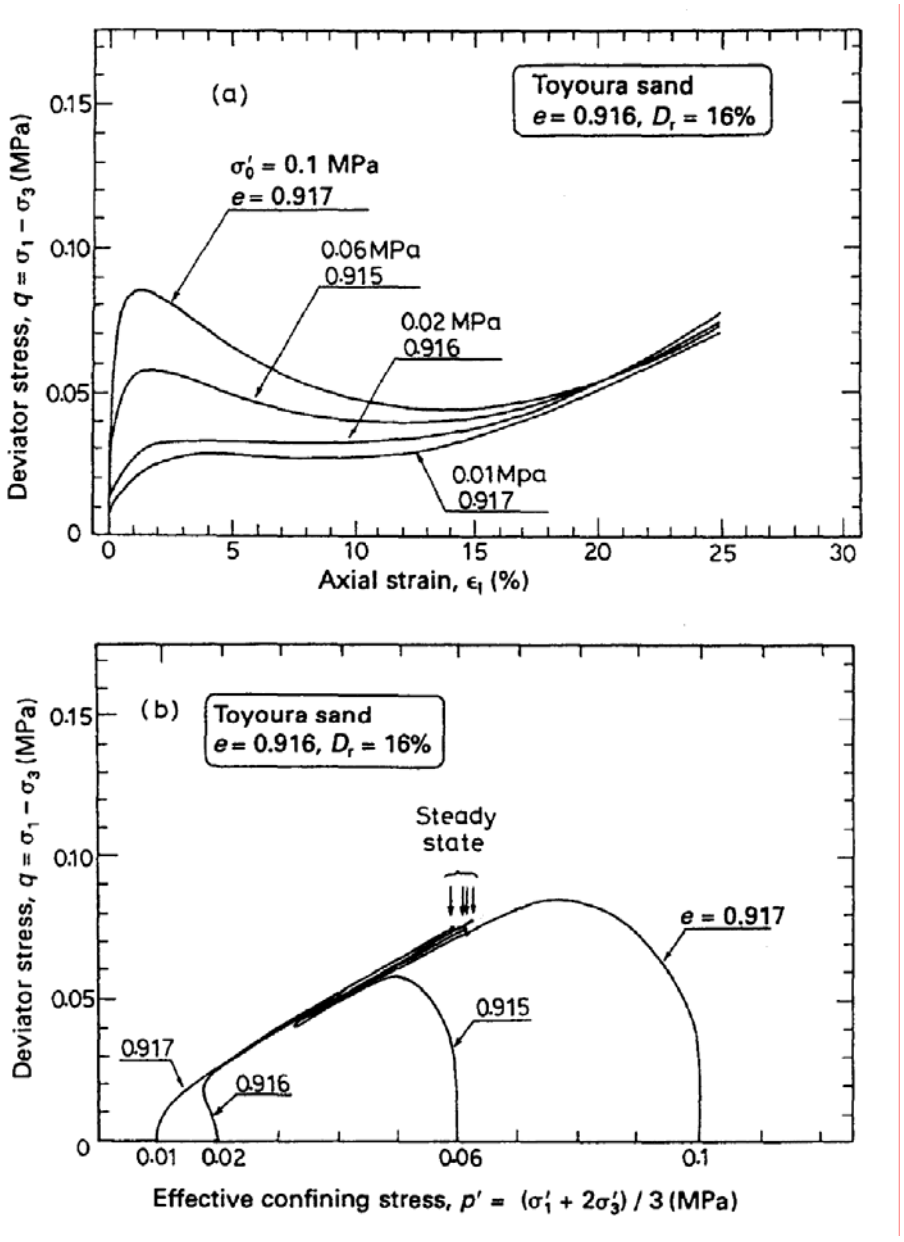


Figure 6.6 Undrained behavior of loose Toyoura Sand (Ishihara, 1996)

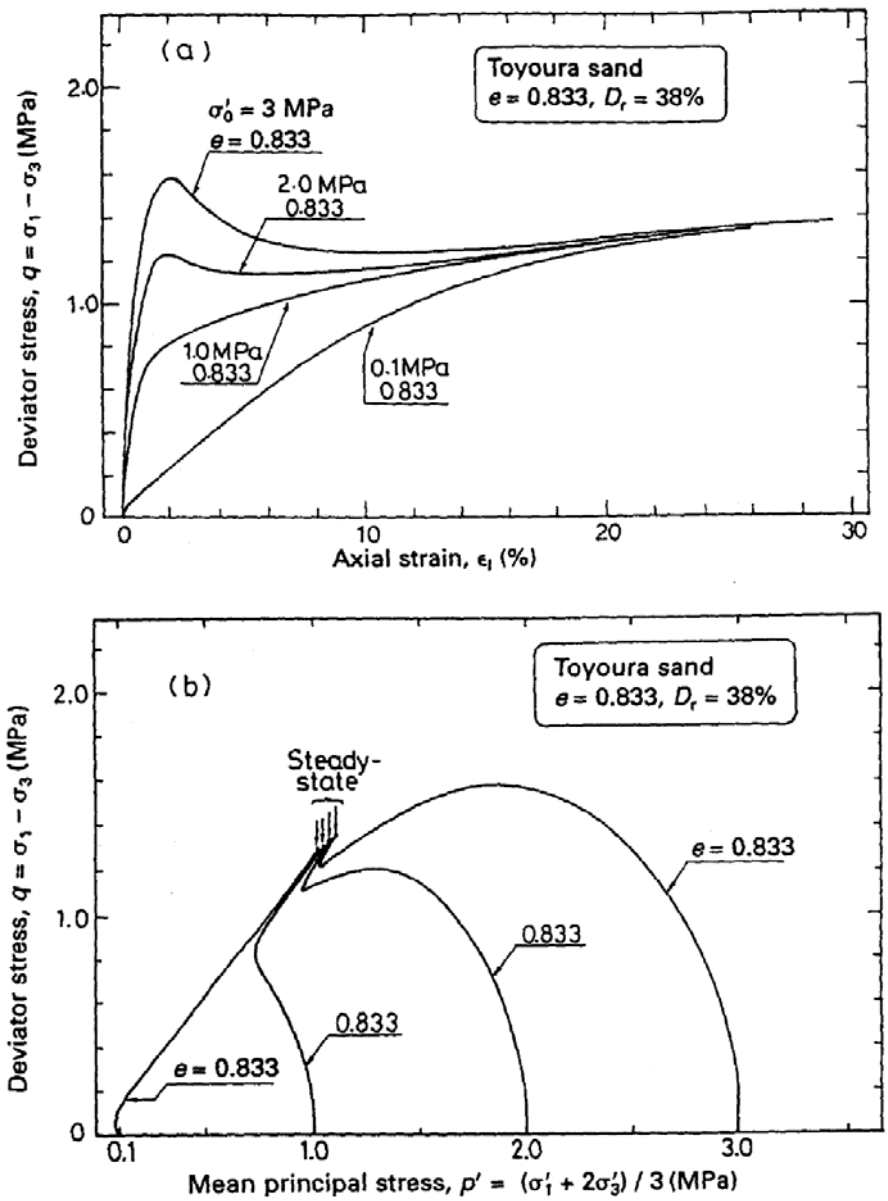


Figure 6.7 Undrained behavior of medium dense Toyoura Sand (Ishihara, 1996)

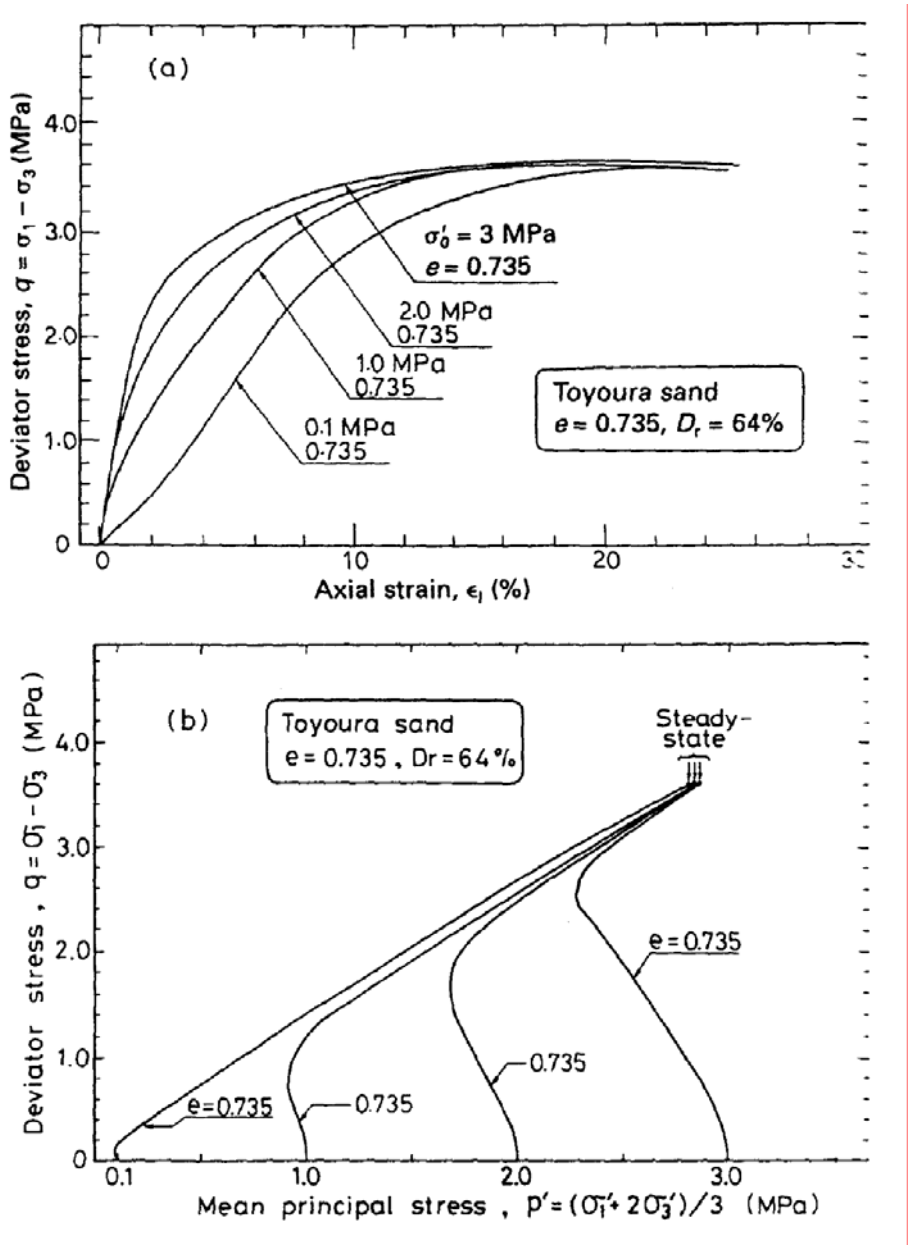


Figure 6.8 Undrained behavior of dense Toyoura Sand (Ishihara, 1996)

6.3 Strain Space Plasticity Model for Cyclic Mobility

6.3.1 Main Features of the Model

Strain space plasticity model for cyclic mobility (Iai,S., Matsunaga, Y., Kameoka,T.,1992) can be categorised among the modern elasto-plastic soil constitutive models for sands under dynamic loading conditions. The form presented in the context of this thesis allows the use of this model with only undrained conditions since the loading path is established using an empirical path formulation called as liquefaction front in the model. The model has been updated for three dimensional case (Iai,S., 1993a; Iai,S., 1993b) and also for drained conditions (Takeshima et al., 2004).

In the context of this thesis, two dimensional and undrained version of the strain space plasticity model for cyclic mobility is incorporated into the finite element code as a constitutive soil model to represent the gradual change of effective stress state and hence the change in strength and stiffness of the saturated sand medium around the rigid pile due to the changes in pore water pressure under dynamic loading conditions.

A model for cyclic mobility should be able to represent rapid or persistent increase in shear strain ranging from 1% to 10% at which cracks, settlement and other important type of deformation initiate. Such large shear strains are only generated when the effective stres path becomes very close to the failure line. This causes difficulty in numerical analysis because a very small error in the computed effective stresses causes a very large error in the computed strains, sometimes causing divergence in the solution process. None of the existing models can deal with this problem; however, attempt made in the model to overcome this difficulty has been reported to be successful (Iai, S. et al., 1990).

Basic characteristics of this model can be defined as follows:

1. It is a generalised plasticity-multiple mechanism type model that means the model is developed in the framework of generalised plasticity theory defined in strain space (loading and unloading is defined in the strain space).

2. Complex behavior under dynamic loading conditions is represented by a single volumetric mechanism and several shear mechanisms (i.e. multimechanism model). The motivating factor behind this approach is, any model casted in terms of stress or strain invariants will predict no response to a pure rotation of principal stress axes. This kind of model can produce accurate results if rotation is accompanied by an increase of the deviatoric strain as occurs in hollow cylinder device. Multimechanism models can represent pure rotation of principal stress axes as the stress state varies in some mechanisms (Zienkiewicz et al., 1988).

3. The undrained effective stress path is idealised with the concept of liquefaction front, which is defined in the effective stress space as an envelope of stress points gradually approaching the failure line. The correct simulation of the undrained effective stress path is important because realistic representation for plastic volumetric strains due to dilatancy can only be achieved if the stress path is traced correctly under dynamic loading conditions.

4. Dilatancy is treated as an additional volumetric strain component in such a manner that strains due to creep and temperature are treated in the constitutive equation.

5. Most of the soil constitutive models dealing with cyclic mobility of sand suffer from the numerical vulnerability when the effective stress path comes very close to the failure line. Hence, this model solves this problem successfully (Iai et al., 1992).

6.3.2 Main Components of the Model

Strain space plasticity model for cyclic mobility is based on the following components:

1. Loading Criterion: Definition of loading/unloading in the framework of generalised plasticity theory defined in the strain space.
2. Decomposition of the Complex Mechanism into Simple Mechanisms
   2a. Modelling of the Volumetric Mechanism
   2b. Modelling of the Shear Mechanism
   2c. Incremental Stress-Strain Relation
3. Undrained Effective Stress Path
4. A methodology to overcome numerical vulnerability when the effective stress path comes very close to the failure line.

6.3.2.1 Definition for Loading and Unloading (Loading Criterion)

In plasticity models, definitions for loading and unloading are necessary because material behavior depends on the direction of loading. Loading and unloading is defined as in Equations 6.1 and 6.2 when expressed in the framework of generalised plasticity defined in strain space. The basics of these definitions were established by Mroz and Norris (1982) and then updated for multiple mechanism approach. A brief review of classical and generalized plasticity theories can be found in Appendix-A of the thesis.

The relation between the effective stress increment and the strain increment is:

$$d\sigma' = Dd\varepsilon \tag{6.1}$$

where D is the tangent stiffness matrix.

If the dependence of the material behaviour on direction is considered;

$$\begin{aligned} d\sigma' &= D_L d\varepsilon \qquad \text{if } n^T d\varepsilon > 0 \\ d\sigma' &= D_U d\varepsilon \qquad \text{if } n^T d\varepsilon < 0 \end{aligned} \tag{6.2}$$

If multiple mechanism approach is adopted;

$$d\sigma' = \sum_{i=0}^{I} d\sigma'^{(i)}$$

$$d\sigma' = \sum_{i=0}^{I} D^{(i)} d\varepsilon \tag{6.3}$$

If the dependence of material behaviour on direction is defined for each mechanism separately,

$$\begin{aligned} d\sigma'^{(i)} &= D_L{}^{(i)} d\varepsilon \qquad \text{if } n^{(i)T} d\varepsilon > 0 \\ d\sigma'^{(i)} &= D_U{}^{(i)} d\varepsilon \qquad \text{if } n^{(i)T} d\varepsilon < 0 \end{aligned} \tag{6.4}$$

Plasticity theory in strain space assures the uniqueness of stress increment for each mechanism, which demands that;

$$\begin{aligned} D_L{}^{(i)} &= R_L{}^{(i)} n_L{}^{(i)} n^{(i)T} \\ D_U{}^{(i)} &= R_U{}^{(i)} n_U{}^{(i)} n^{(i)T} \end{aligned} \tag{6.5}$$

Where;

$D_L{}^{(i)}, D_U{}^{(i)}$: tangential stiffness matrix for each mechanism,for loading and unloading respectively.

$n_L{}^{(i)}, n_U{}^{(i)}$: arbitrary vectors defining the directions of stress increments

$R_L{}^{(i)}, R_U{}^{(i)}$: define the magnitudes of stress increments

Finally, the constitutive relation including all the mechanisms can be given by (6.6);

$$d\sigma' = \sum_{i=0}^{I} R_{L/U}{}^{(i)} n_{L/U}{}^{(i)} n^{(i)T} d\varepsilon \tag{6.6}$$

in associated form, $n_{L/U} = n$

For the preceding formulations and for the ones coming next, two dimensional behaviour of soil under plane strain conditions are considered, and hence the stres and strain vectors are defined as:

$$\sigma' = \begin{Bmatrix} \sigma'_x \\ \sigma'_y \\ \tau_{xy} \end{Bmatrix}, \ \varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{6.7}$$

In addition, compressive stress and the contractive strain are assumed negative. The sign of the strains are determined by the nodal point deformations corresponding to x and y directions.

6.3.2.2 Decomposition of the Complex Mechanism into Simple Mechanisms

Constitutive relation (6.6) is composed of I+1 mechanisms of which the first one is the volumetric mechanism and the rest (I) are the shear mechanisms. Basically, there are two main mechanisms which are volumetric mechanism and the shear mechanism. However, when the effects of principal stress axes rotation has to be taken into account, the shear mechanism has to be divided into simple mechanisms which is the basic philosophy of the multimechanism models.

Volumetric Mechanism:

Volumetric mechanism is the first mechanism and such a mechanism is defined for representing the cumulative pore pressure build up with occasional recovery of the mean effective stress. Dilatancy represents this phenomena which is expressed in terms of plastic volumetric strain. For the modelling of volumetric mechanism, the approach taken fort he densification model by Zienkiewicz et al. (1978). This approach is simple and the data obtained from undrained cyclic loading conditions can be directly used. The first mechanism of (6.6) can be expressed as :

$$d\sigma'^{(0)} = R_{L/U}{}^{(0)} n^{(0)} n^{(0)T} d\varepsilon$$

$$d\sigma'^{(0)} = K n^{(0)} n^{(0)T} (d\varepsilon - d\varepsilon_p) \tag{6.8}$$

in matrix form;

$$\begin{Bmatrix} d\sigma'_x \\ d\sigma'_y \\ d\tau_{xy} \end{Bmatrix}^0 = K \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \left(\begin{Bmatrix} d\varepsilon_x \\ d\varepsilon_y \\ d\gamma_{xy} \end{Bmatrix} - \begin{Bmatrix} d\varepsilon_p / 2 \\ d\varepsilon_p / 2 \\ 0 \end{Bmatrix} \right) \tag{6.9}$$

$$n^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \tag{6.10}$$

Here, the additional volumetric strain $d\varepsilon_p$ is assumed to represent the whole part of the plastic volumetric strain generated by transient and cyclic loads. Then the strain increment $d\varepsilon - d\varepsilon_p$ becomes elastic and it is reasonable to represent $R_{L/U}{}^{(0)}$ with the elastic tangent bulk modulus of the soil skeleton known as rebound modulus.

$$R_{L/U}{}^{(0)} = K = K_a \left(\frac{\sigma'_m}{\sigma'_{ma}} \right)^{0.5} \tag{6.11}$$

where,

K : elastic tangent bulk modulus of the soil skeleton (rebound modulus)

$\sigma'_m = \dfrac{\sigma'_x + \sigma'_y}{2}$: mean effective stress

K_a: elastic tangent bulk modulus of soil skeleton at σ'_{ma}

(Iai,S.,1990)

Shear Mechanism:

The second mechanism to be modelled is the shear mechanism, which is based on the approach by Towhata and Ishihara (1985a). This approach has the ability to represent soil behaviour under principal stress axes rotation by decomposing the complex shear mechanism into a set of one dimensional seperate shear shear mechanisms as shown in Figure 6.9 and Figure 6.10.

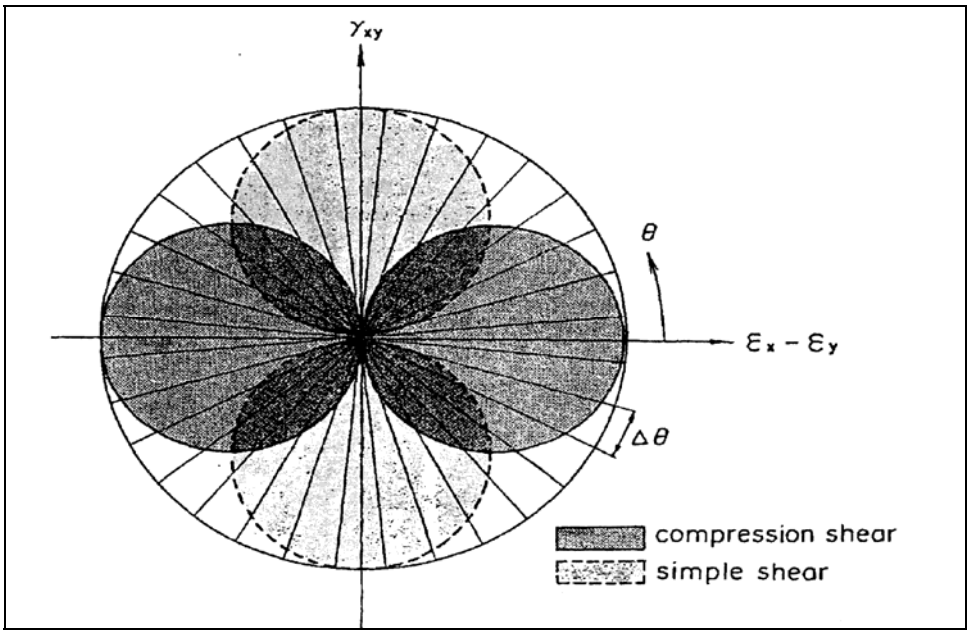


Figure 6.9 Schematic figure for multiple shear mechanisms (pairs of circles indicate Mobilized virtual shear strain in positive and negative modes of compression shear and simple shear) (Iai, 1990)

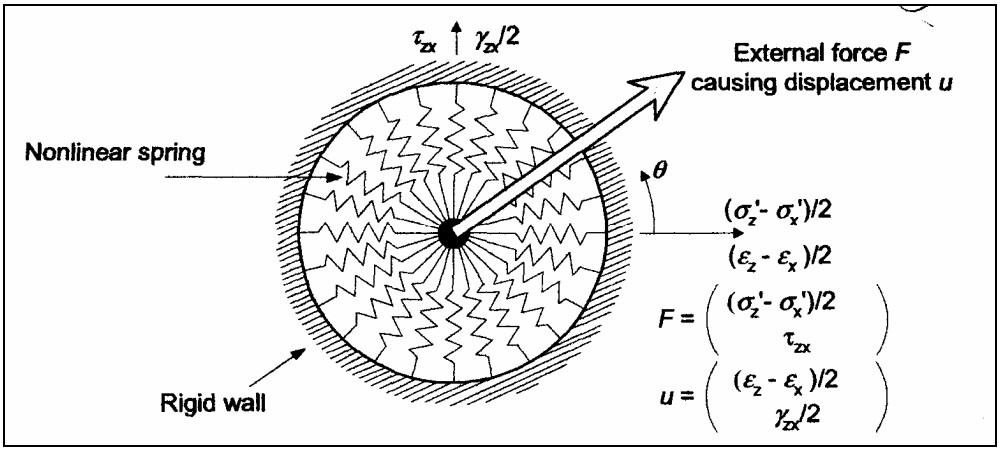


Figure 6.10 Multiple shear spring model (after Towhata and Ishihara) (Takeshima et al.,2004)

Shear mechanism part of the constitutive relation (Eqn.6) is expressed as;

$$d\sigma' = \sum_{i=1}^{I} R_{L/U}{}^{(i)} n^{(i)} n^{(i)T} d\varepsilon \qquad (6.12)$$

I : number of mechanisms

i: one dimensional stres-strain relation defined in a virtual simple shear which is mobilized at angle $\theta_i / 2 + \pi / 4$ to the x axis. $\theta_i = 0$ represents virtual compression mode, $\theta_i = \pi / 2$ represents virtual simple shear mode in the x axis direction (Figure 6.9)

$R_{L/U}^{(i)}$: tangent shear modulus for each mechanism

$$n^{(i)^T} = \begin{bmatrix} \cos\theta_i \\ -\cos\theta_i \\ \sin\theta_i \end{bmatrix} \tag{6.13}$$

$$\theta_i = (i-1)\Delta\theta$$
$$\Delta\theta = \pi / I$$

Hence, the incremental stress – strain relation for the whole system is given as:

$$d\sigma' = Kn^{(0)}n^{(0)^T}(d\varepsilon - d\varepsilon_p) + \sum_{i=1}^{I} R_{L/U} n^{(i)}n^{(i)^T} d\varepsilon$$

or

$$n^{(i)} d\varepsilon_p = 0 \text{ for i} = 1....\text{I} \tag{6.14}$$
$$d\sigma' = D(d\varepsilon - d\varepsilon_p)$$
$$D = Kn^{(0)}n^{(0)^T} + \sum_{i=1}^{I} R_{L/U}{}^{(i)} n^{(i)}n^{(i)^T}$$

The spring group (Figure 6.10) represents the two dimensional shear stress-shear strain behavior of the two dimensional soil element (actual shear) by using one-dimensional springs which are called virtual shear mechanisms. Each virtual shear mechanism behaves according to the one dimensional shear stress-shear strain relation developed by using Masing rules.

The shear stress-shear strain behaviour of each spring is controlled by hyperbolic model for the construction of the backbone and the revised Masing rules for the construction of hysteresis curve. Masing rules are revised by Iai due to the fact that the hyperbolic relation does not represent the realistic hysteresis loop when the amplitude of cyclic shear strain becomes as large as a few percent; at this strain level

the hysteresis loop given by Masing's rule consumes about twice the energy as those observed in the laboratory tests (Iai, 1994).

Construction of initial loading, unloading from initial loading and reloading/reunloading portions of the shear stress-shear strain relations of each virtual shear mechanism are given in the following paragraphs.

When a soil element is exposed to strain increment of dε, the virtual shear strain increment in each virtual shear mechanism can be expressed as:
Total strain increment vector:

$$d\varepsilon = \begin{Bmatrix} d\varepsilon_x \\ d\varepsilon_y \\ d\gamma_{xy} \end{Bmatrix} \tag{6.15}$$

Virtual shear strain increment for the i th. virtual shear mechanism:

$$d\gamma^{(i)} = n^{(i)^T} d\varepsilon \tag{6.16}$$

$$d\gamma^{i} = \left(d\varepsilon_x - d\varepsilon_y\right)\cos\theta_i + d\gamma_{xy}\sin\theta_i$$

Loading criterion for each mechanism can be defined as follows:

$$\begin{aligned} n^{(i)^T} d\varepsilon > 0 \quad & \text{loading} \\ n^{(i)^T} d\varepsilon < 0 \quad & \text{unloading} \\ n^{(i)^T} d\varepsilon = 0 \quad & \text{neutral} \end{aligned} \tag{6.17}$$

Initial loading:
Shear stress- shear strain relation belonging to each mechanism is defined as :

$$\begin{aligned} Q^{(i)} &= \frac{\dfrac{\gamma^{(i)}}{\gamma_v}}{1+\left|\dfrac{\gamma^{(i)}}{\gamma_v}\right|} Q_v \\ R_L^{(i)} &= \frac{1}{\left(1+\left|\dfrac{\gamma^{(i)}}{\gamma_v}\right|\right)^2} \frac{Q_v}{\gamma_v} \Delta\theta \end{aligned} \tag{6.18}$$

in normalised form;

$$\xi = \frac{\gamma^{(i)}}{\gamma_v} \qquad \eta = \frac{Q^{(i)}}{\gamma_v}$$
$$\eta = \frac{\xi}{1+|\xi|} \tag{6.19}$$

in these formulations;

$\gamma^{(i)}$: virtual shear strain for the i th. mechanism

$\xi^{(i)}$: normalised virtual shear strain

$Q^{(i)}$: virtual shear stress for the i th. mechanism

$\eta^{(i)}$: normalised virtual shear stress

$R_L^{(i)}$: virtual tangent shear modulus for the i th.mechanism

Q_v : virtual shear strength

γ_v : virtual reference strain

Unloading from initial loading:

In normalised form, unloading from initial loading is represented by (6.20)

$$\frac{\eta - \eta_r}{2} = f\left(\frac{\xi - \xi_r}{2}\right) \qquad \text{(In normalised form)} \tag{6.20}$$

(ξ_r, η_r) : coordinates of the reversal point

When unloading occurs from initial loading the reversal points will be on the backbone . Hence, the coordinates of the reversal point from the backbone named as (ξ_B, η_B) and Eqn. (6.20) becomes

$$\frac{\eta - \eta_B}{2} = f\left(\frac{\xi - \xi_B}{2}\right) \tag{6.21}$$

However, Masing's Rule when combined with the hyperbolic relation does not represent the realistic hysteresis rule when the amplitude of cyclic shear strain

becomes as large as a few percent, at this strain level, the hysteresis loop given by Masing's rule consumes about twice the energy as those observed in the laboratory tests (Iai et al.,1998). To remedy this problem numerically, some scale parameters are introduced:

$$\xi' = \frac{\xi}{a} \qquad \eta' = \frac{\eta}{b}$$
$$\xi_B' = \frac{\xi_B}{a} \qquad \eta_B' = \frac{\eta_B}{b} \tag{6.22}$$
$$\frac{\eta' - \eta'_B}{2} = f\left(\frac{\xi' - \xi'_B}{2}\right)$$

Parameters a and b are found by considering the damping factors of each mechanism and the damping factor of the actual shear as follws:

$$b = \frac{a + |\xi_B|}{1 + |\xi_B|} \tag{6.23}$$

In order to find *b*, *a* should be found first:

$$h(|\xi_B|) = \frac{\left|\frac{\xi_B}{\xi_h}\right|}{1 + \left|\frac{\xi_B}{\xi_h}\right|} h_v \tag{6.24}$$

$h(|\xi_B|)$:realistic damping factor to be represented by each virtual simple shear mechanism. The strain level is $|\xi_B|$.

h_v : limiting value of virtual damping factor when virtual strain level is infinity.

ξ_h : a parameter similar to virtual reference strain.

Hysteresis loop defined by Eqn. (6.24) produces the damping as a nondimensional damping factor parameter:

$$D(|\xi'_B|) = \frac{4}{\pi}(1+\frac{1}{|\xi'_B|})(1-\frac{\ln(1+|\xi'_B|)}{|\xi'_B|}) - \frac{2}{\pi} \tag{6.25}$$

a is defined using the following relationship;

$$D(\left|\frac{\xi_B}{a}\right|) = h(|\xi_B|) \tag{6.26}$$

In these formulations ξ_h is formulated by;

$H(|\gamma_{xyB}|) = \dfrac{\sum_{i=1}^{I} W^{(i)} h(|\gamma_B{}^{(i)} / \gamma_v|)}{\sum_{i=1}^{I} W^{(i)}}$: damping factor for actual shear (6.27)

$\overline{H}(|\gamma_{xyB}|) = \dfrac{\left|\frac{\gamma_{xyB}}{\gamma_m}\right|}{1+\left|\frac{\gamma_{xyB}}{\gamma_m}\right|} H_m$: actual damping factor to be measured in the laboratory. The relation is proposed by Hardin and Drnevich (1972) (6.28)

ξ_h is found by;

$$H(|\gamma_{xyB}|) = \overline{H}(|\gamma_{xyB}|) \tag{6.29}$$

Finally the unloading curve is given in the non normalised form as:

$$\begin{aligned} &\frac{\frac{Q^{(i)}}{Q_v} - \eta_B}{2b} = f(\frac{\frac{\gamma^{(i)}}{\gamma_v} - \xi_B}{2a}) \\ &R_U^{(i)} = g(\frac{\frac{\gamma^{(i)}}{\gamma_v} - \xi_B}{2a}) \frac{Q_v}{\gamma_v} \frac{b}{a} \Delta\theta \\ &g(\xi) = \frac{1}{(1+|\xi|)^2} \end{aligned} \tag{6.30}$$

Reloading and reunloading:

$$\frac{\eta^{'}-\eta^{'}_{r}}{2c}=f(\frac{\xi^{'}-\xi^{'}_{r}}{2c})$$
$$c=\frac{1}{2}\frac{\left|\pm\xi_{B}^{'}-\xi^{'}_{r}\right|(\pm\eta_{B}^{'}-\eta^{'}_{r})}{(\pm\xi_{B}^{'}-\xi^{'}_{r})-(\pm\eta_{B}^{'}-\eta^{'}_{r})} \quad (6.31)$$

$$\frac{\frac{Q^{(i)}}{Q_v}-\eta_r}{2bc}=f(\frac{\frac{\gamma^{(i)}}{\gamma_v}-\xi_r}{2ac})$$
$$R^{(i)}_{L/U}=g(\frac{\frac{\gamma^{(i)}}{\gamma_v}-\xi_r}{2ac})\frac{Q_v}{\gamma_v}\frac{b}{a}\Delta\theta \quad (6.32)$$

The shear strength parameters for the virtual shear mechanisms are defined as:

$$Q_v=\frac{\tau_m}{\sum_{i=1}^{\mathrm{Im}}\sin\theta_i\Delta\theta}$$
$$\gamma_v=\frac{Q_v}{G_m}\sum_{i=1}^{\mathrm{Im}}\sin\theta_i\Delta\theta \quad (6.33)$$

Q_v: Virtual mechanism shear strength

τ_m: Actual shear strength

γ_v:Virtual reference strain

G_m: Shear modulus for the actual shear mechanism at low strain level

6.3.2.3 Simulation of the Undrained Effective Stress Path

It was stated previously that sands try to change volume under monotonic and cyclic loading conditions which is known as dilatancy. This tendency to change volume causes pore pressure build up (either increase or decrease) under undrained conditions which changes the mean effective stresses. Related with the mean

effective stresses soil stiffness and strength change too. Dilatancy is represented in terms of plastic volumetric strain. The change in the value of the plastic volumetric strains with the progress of cyclic mobility (for any boundary value problem) can be traced close to reality if the laboratory undrained effective stress paths could be simulated numerically. In the strain space plasticity model for cyclic mobility, components of the numerical simulation of the undrained effective stress path are:

1. Relation between the plastic volumetric strain and the excess pore pressure or mean effective stres
2. Modelling of the effective stress path (Liquefaction front)
3. Relation between the liquefaction front parameter (S0) and shear work (shear work correlation)
4. Threshold limit

Relation Between the Plastic Volumetric Strain and the Excess Water Pressure /Mean Effective Stress:

Plastic volumetric strain needs to be specified in order to use the constitutive relation of the model. As the plastic volumetric strains represent the effects of dilatancy, it is natural to think that they are closely related with the change in pore pressure or effective mean stress under constant total confining pressure. This relation is derived considering that, under constant total mean effective stress σ'_{m0} with undrained condition (no change in volume), volumetric strain due to change in effective stress is only due to the volumetric strain of pore water (Iai, 1993).

$$\Delta V = 0,\ \frac{\Delta V}{V} = \varepsilon_x + \varepsilon_y = 0 \quad \text{(undrained condition, no change in volume)} \tag{6.33}$$

$$\varepsilon_e + \varepsilon_p - \varepsilon_{e0} = \frac{n}{K_f}\left(\sigma'_{m0} - \sigma'_m\right) \tag{6.34}$$

$$\varepsilon_p = \frac{n}{K_f}\left(\sigma'_{m0} - \sigma'_m\right) + \left(\frac{\sigma'_m}{-B}\right)^{0.5} + \varepsilon_{e0}$$

Where,

ε_p : plastic volumetric strain

ε_e : elastic volumetric strain

ε_{e0} : initial elastic volumetric strain

n : porosity

K_f : bulk modulus of water

σ'_m : mean effective stres

σ'_{m0} : initial mean effective stres (under undrained conditions, it represents the total mean effective stres)

$$B = \left[\frac{0.5 K_a}{(-\sigma'_{ma})^{0.5}} \right]^{0.5} \tag{6.35}$$

Modelling of Effective Stress Path (Liquefaction Front):

In this model, effective stress path for cyclic mobility is modelled by using the concepts phase transformation line and the envelope of stress points at equal shear work (liquefaction front). Phase transformation line is a straight-line seoarating the dilative zone from the contractive zone in stress space. The envelope of stress points at equal shear work is shown in Figure 6.11. As shear work is accumulated by cyclic shear under undrained conditions, the envelope of stress points at equal shear work gradually moves from the initial envelope to the failure line (Figure 6.11). This envelope is called the liquefaction front and is used in the modeling of cyclic mobility. In order to approximate the shape of the liquefaction front, it should be defined in normalised stress space defined with effective mean stress ratio and deviatoric stress ratio (Figure 6.12).

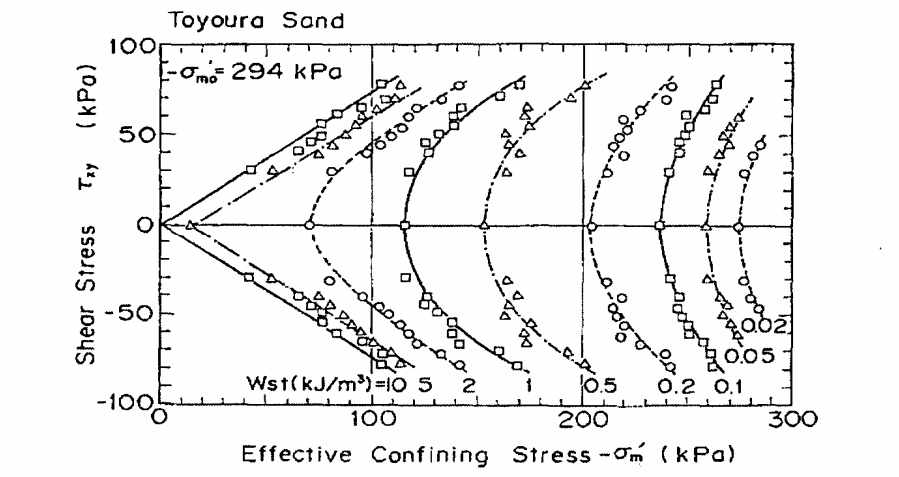


Figure 6.11 Envelope of stres points at equal shear work (after Towhata and Ishihara, 1985b)

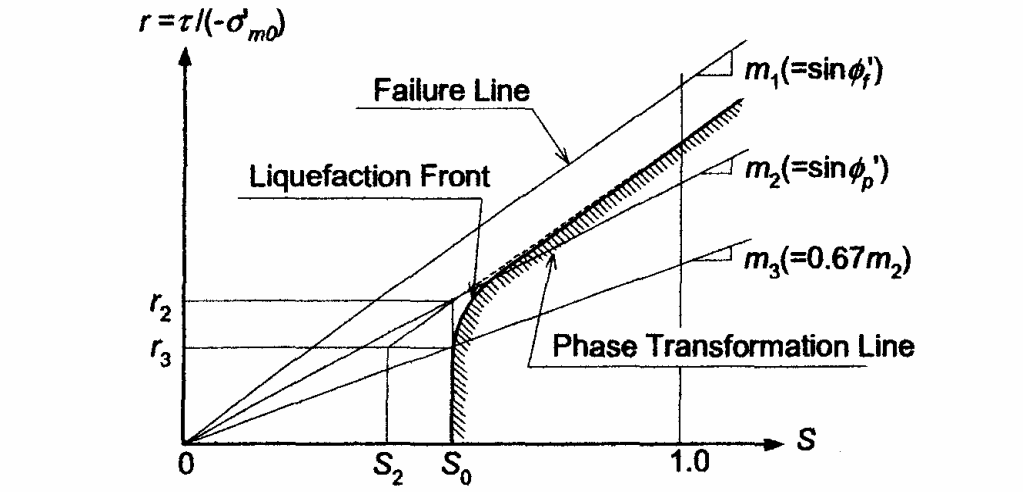


Figure 6.12 Schematic figure of liquefaction front, state variable S and shear stress ratio(Iai, S., 1992)

$S = \frac{\sigma'_m}{\sigma'_{m0}}$:effective mean stres ratio (6.36)

$r = \frac{\tau}{-\sigma'_{m0}}$: deviatoric stres ratio (6.37)

$\sigma'_{m0} = \frac{\sigma'_{x0} + \sigma'_{y0}}{2}$: initial effective mean stress (6.38)

$$\tau = \frac{\sigma_1^{'} - \sigma_3^{'}}{2} = \sqrt{\tau_{xy}^2 + \frac{\sigma_x^{'} - \sigma_y^{'}}{2}} \quad \text{: deviatoric stress} \tag{6.39}$$

The shape of the liquefaction front is approximated by the following function:

$$\begin{aligned}
&if\ r < r_3 \quad S = S_0 \\
&if > r_3 \qquad S = S2 + \sqrt{(S_0 - S2)^2 + (\frac{r - r_3}{\mathrm{m}_1})^2} \\
&m_1 = \sin\phi_f \\
&m_2 = \sin\phi_p \\
&m_3 = 0.67 m_2 \\
&r_2 = m_2 S0 \\
&r_3 = m_3 S0 \\
&S2 = S0 - \frac{r_2 - r_3}{\mathrm{m}_1}
\end{aligned} \tag{6.40}$$

S_0:liquefaction front parameter. It is a measure which defines the state of liquefaction (should be defined by a function of shear work)

S_0=1 initial stres state if r<m3

S_0=0 limiting state at which failure occurs due to liquefaction

m1: inclination of failure line

m2: inclination of phase transformation line

m3: introduced for assuring the smooth transition from one zone to the other.

Shear Work Correlation:

The liquefaction front parameter S0 is given by a function of shear work.

$$w = \frac{W_s}{W_n}$$

$$W_n = \frac{\tau_{m0}\gamma_{m0}}{2}$$

$$if\ w < w_1 \quad S0 = 1 - 0.6\left(\frac{w}{w_1}\right)^{p_1} \tag{6.41}$$

$$if\ w > w_1\ \ S0 = (0.4 - S1)\left(\frac{w_1}{w}\right)^{p_2} + S1$$

$$if\ w = w_1\ \ S0 = 0.4$$

Here,

S1, w1, p1, p2 are the material parameters which characterise the liquefaction properties of the cohesionless soil.

w: normalised shear work.

Threshold Limit:

In order to obtain the shear work correlation, value of plastic shear work should be determined. Total shear work concept was used by Towhata and Ishihara (1985b) to find the value plastic shear work. Also, it is a known fact that there exists a threshold limit in the amplitude of cyclic shear strain or shear stress below which no pore pressure build up occurs. This part of the shear work (this part is related with the elastic shear work) should be subtracted from the total shear work in order to find the correct plastic shear work which is related with pore pressure build up.

Total shear work is defined as:

$$dW_{st} = \frac{\left(\sigma_x^{'} - \sigma_y^{'}\right)}{2} d(\varepsilon_x - \varepsilon_y) + \tau_{xy} d\gamma_{xy} \tag{6.42}$$

$$if\ sign(dW_{st}) = -1 \Rightarrow dW_{st} = abs(dW_{st})$$

Elastic shear work is defined as:

$$dW_{se} = \left| \tau d\left(\frac{\tau}{G_m^*}\right) \right| \tag{6.43}$$

Plastic shear work is defined by:

$$dW_s = dW_{st} - c_1 . dW_{se} \tag{6.44}$$

c_1: specifies the treshold level

$$if\ \text{sign}(dW_s) = -1 \Rightarrow dW_s = 0$$

6.3.2.4 Scheme for Numerical Robustness

A numerically robust scheme is necessary for the following reasons:

Problem 1: With the progress of cyclic loading, excess pore water pressure builds up which changes the mean effective stresses and hence the shear strength and the shear modulus of the soil. If the current values of these parameters are obtained from the initial values as in equations 6.45, some problems arise. One of these problems is the difficulty in numerical analysis when the effective stress path becomes close to the failure line. In such a case, even a very small error in the computed effective stress causes a very large error in the computed strains.

$$\tau_m = \tau_{m0}\frac{\sigma_m^{'}}{\sigma_{m0}^{'}}$$
$$G_m = G_{m0}\left(\frac{\sigma_m^{'}}{\sigma_{m0}^{'}}\right)^{0.5} \qquad (6.45)$$
$$\gamma_m = \frac{\tau_m}{G_m} \qquad \gamma_m = \gamma_{m0}\left(\frac{\sigma_m^{'}}{\sigma_{m0}^{'}}\right)^{0.5}$$

Problem 2: Shear strains due to cyclic mobility range from about 1% to 10%. Such large strains are generated only when the stress path becomes very close to the failure line. Although the effective stress path converges to a limiting stress path, the stress-strain loop does not converge to a closed hysteresis loop due to significant increases in the values of shear strain. Some of the existing models can't simulate this increase in shear strain although they are numerically robust in simulating the cyclic mobility.

Problem 3: This part of the problems is related with the shape of the hysteresis loop. Hysteresis loop simulated with this formulation is more of a convex nature even the stress path is in the dilative zone. These problems can be solved by the introduction of an additional strength $\Delta\tau_m$ for shear strength. The related scheme is given as follows:

if $S_0 > 0.4$	if $S_0 < 0.4$	
$\tau_m = \tau_{m0} S$	$\tau_m = \tau_{m0} S + \Delta\tau_m$	
$G_m = \dfrac{\tau_m}{\gamma_{m0}}$	$G_m = \dfrac{\tau_m}{\gamma_m}$	(6.46)
$\gamma_m = \gamma_{m0}$	$\gamma_m = \dfrac{\gamma_{m0}}{\dfrac{S_0}{0.4}}$	

In Figure 6.13, an overall flow chart of the model is given:

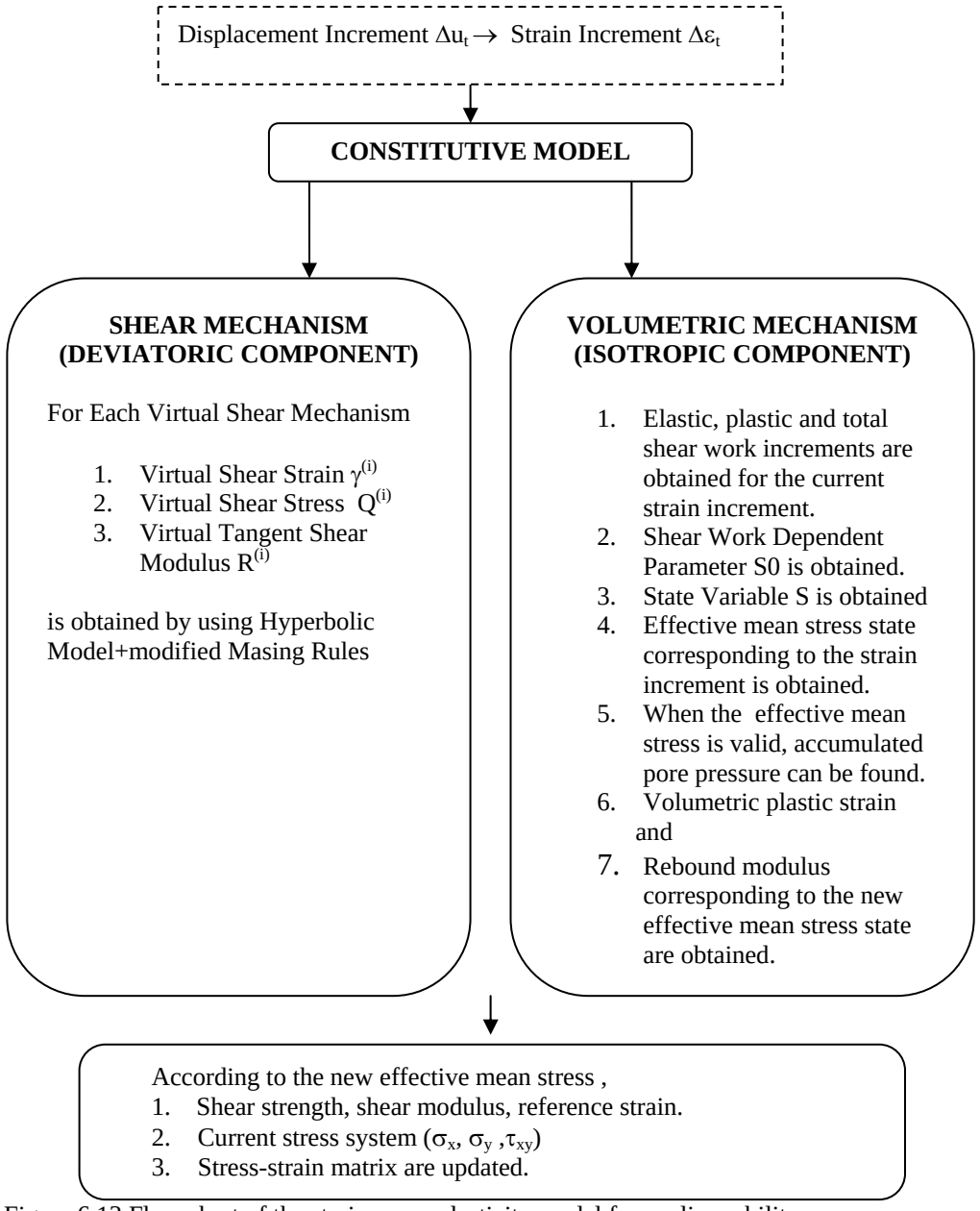


Figure 6.13 Flow chart of the strain space plasticity model for cyclic mobility

CHAPTER SEVEN

THE FINITE ELEMENT CODE

7.1 Introduction

In this chapter, the components of the nonlinear dynamic finite element code developed for the time domain analysis of the problem stated in the second chapter of this thesis will be introduced. The code is written in Matlab 7.0 of which the capabilities on matrix operations and figure options are extensive. This is a very useful feature for the prelimanary coding purposes of dynamic nonlinear finite element code due to the fact that nonlinearity under dynamic loading conditıons is not an easy task to deal with using higher level of programming languages. A lot of things may go wrong and the code developer should be patient, should have an instinctive feeling about how the system will behave and understand the mathematical basis of the finite element method very well. Below is a quote from Irons and Ahmad (1986): 'We have no idea of how many nervous breakdowns are precipitated by the persistent failure of computer programs to solve nonlinear problems…'. In the following sections, the components of the code as well as its feautures and some shortcomings are given.

7.2 Components of the Dynamic Nonlinear Finite Element Code

General flow chart of the FEM algorithm is presented in Figure 7.1. One should note that the main subroutine, which assigns major tasks to other subroutines and functions as a controller algorithm is named as "iai_main_undrained.m". Subroutines of the FEM code were generated as distinct small programs. These files are called during FEM analysis and are expected to interact each other seamlessly. Maximum care has been spend in order to establish sound communication channels among forty six Matlab subroutine files.

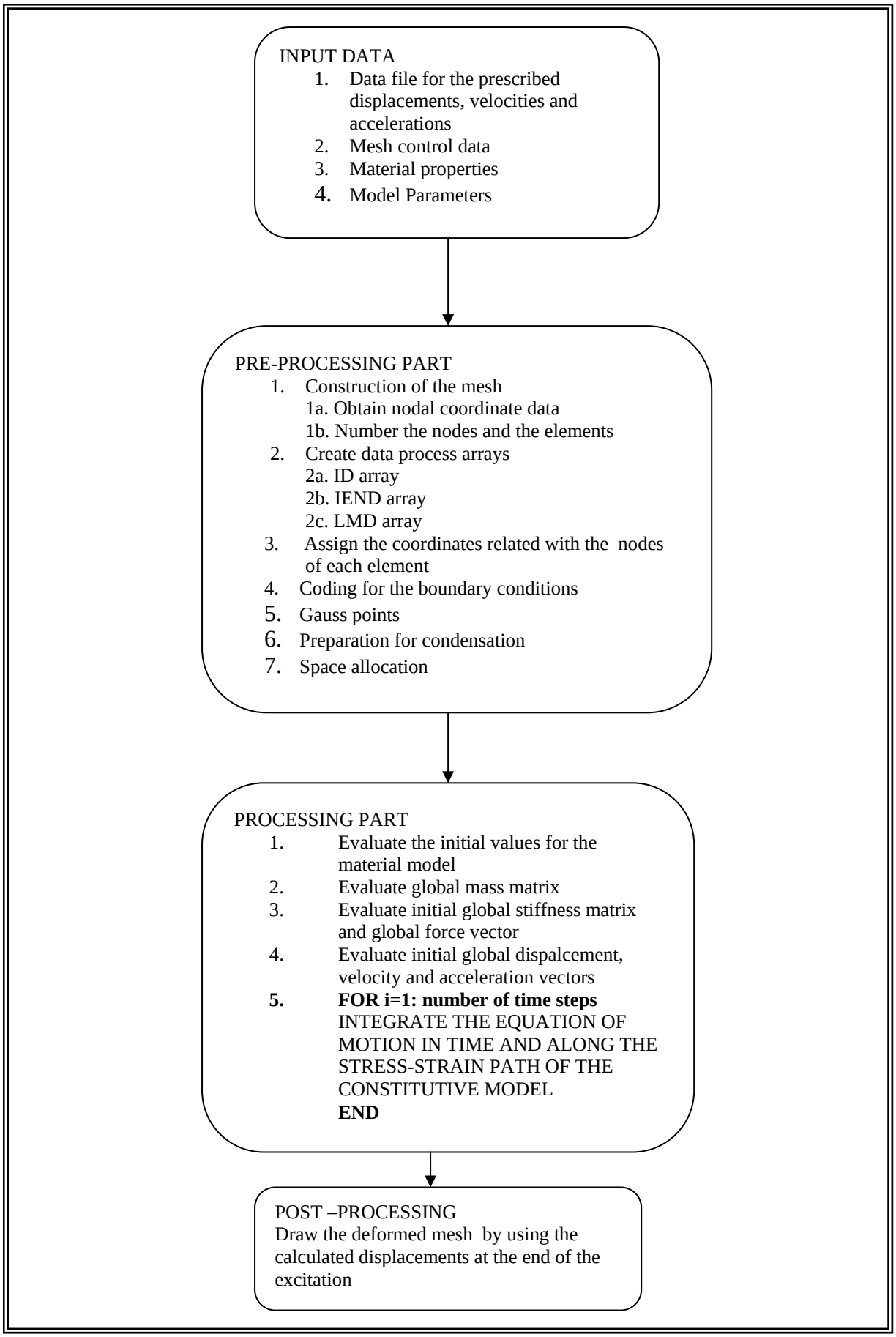


Figure 7.1 Brief flow chart of the dynamic nonlinear FEM code

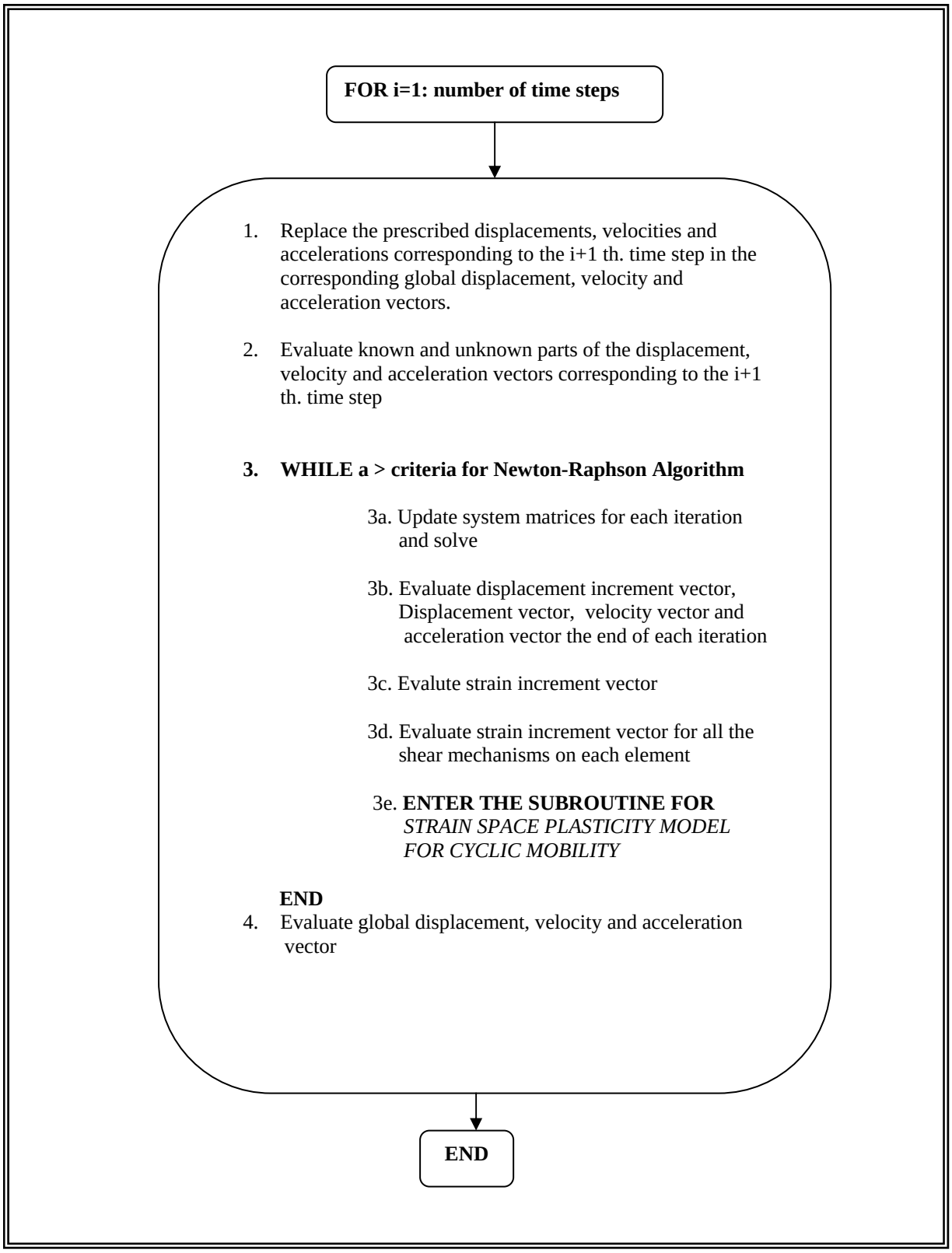


Figure 7.1 Brief flow chart of the dynamic nonlinear FEM code (continue)

7.3 Sample Applications of the FEM Code

7.3.1 Single Element Case

This section is devoted to the discussion of the sample runs performed to check the validity of the program in case of soil-pile interaction problem stated in Chapter 2. Initially, sample runs were obtained for the single soil element case afterwhich the assemblage of soil elements for SSPSI problem analysis was attempted. Typical results obtained from a single element run are presented which are in good agreement with the behaviour expected. In this single element run, harmonic boundary pile displacement shown in Figure 7.2. is applied to the element. The single soil-pile element is shown at the interface of the pile in Figure 7.3.

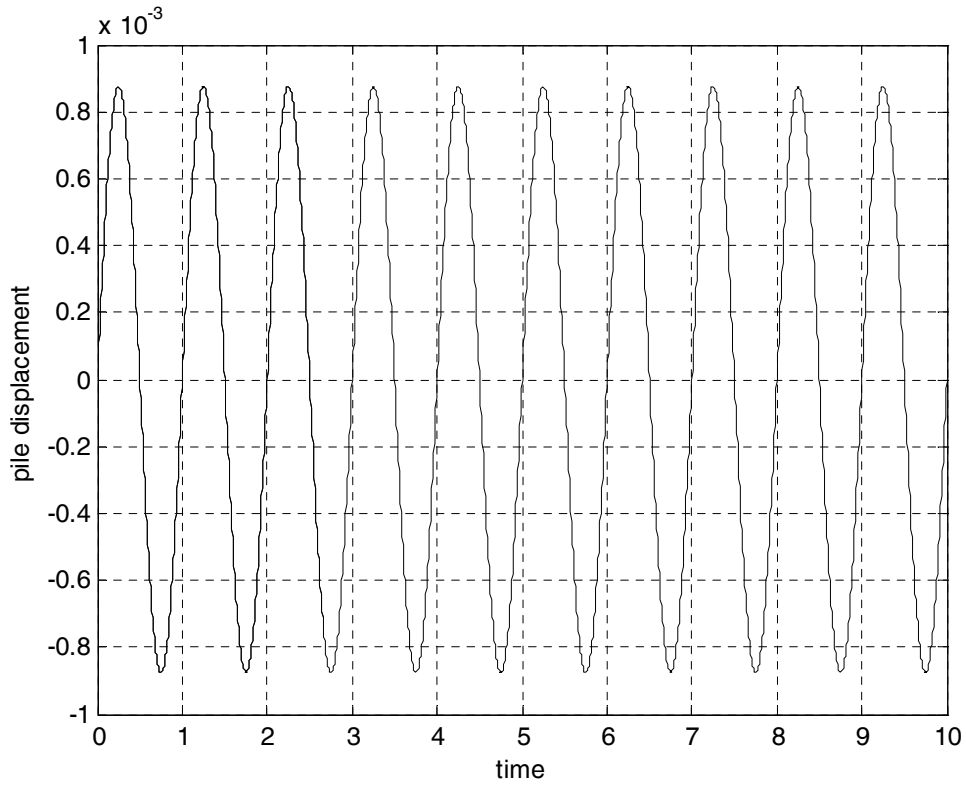


Figure 7.2 Harmonic displacement applied at the pile head

At nodes 2 and 3 of the element in Figure 6.3, the soil element is hold against displacements bıth in x and y directions. For nodes 1 and 4, the pile displacement increment is the boundary condition along the x direction while these nodes are hold

in the y direction. The initial loading and hysteresis loops of the single soil element due to the mentioned dynamic loading conditions is shown in Figure 7.4. In addition, the initial loading and hysteresis loops for the virtual shear mechanisms are given in Figure 7.5. The close similarity in shape between the actual and virtual shear mechanisms are due to the fact that virtual shear mechanisms construct the behaviour of the actual shear mechanism.

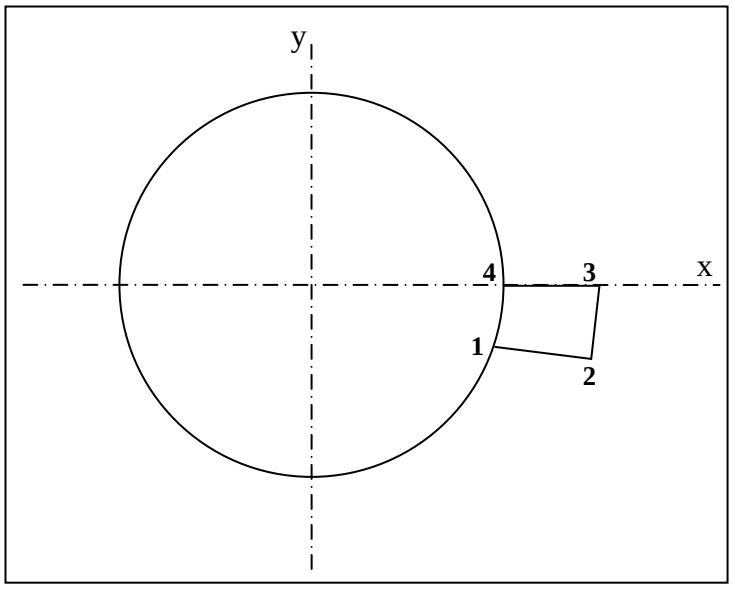


Figure 7.3 Single element case

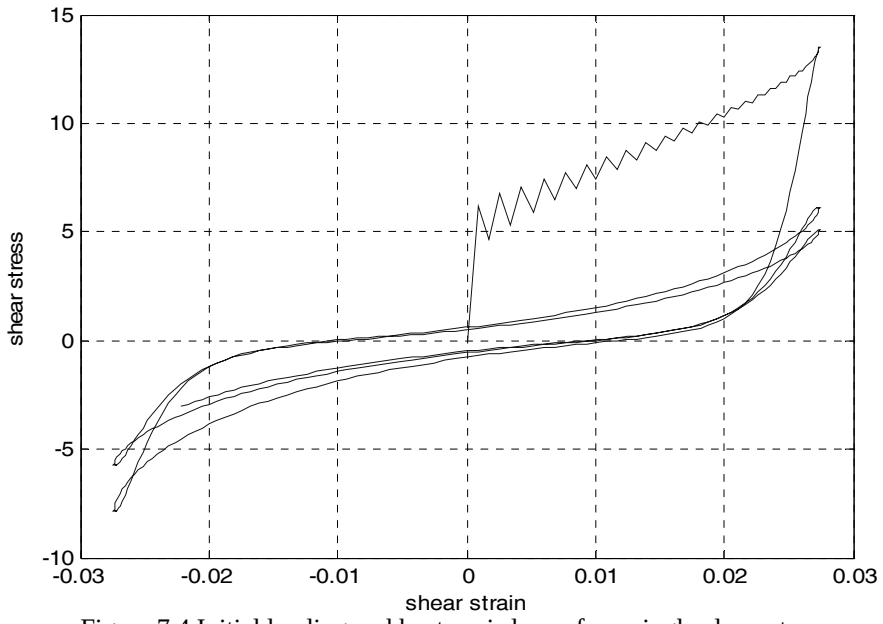


Figure 7.4 Initial loading and hysteresis loops for a single element

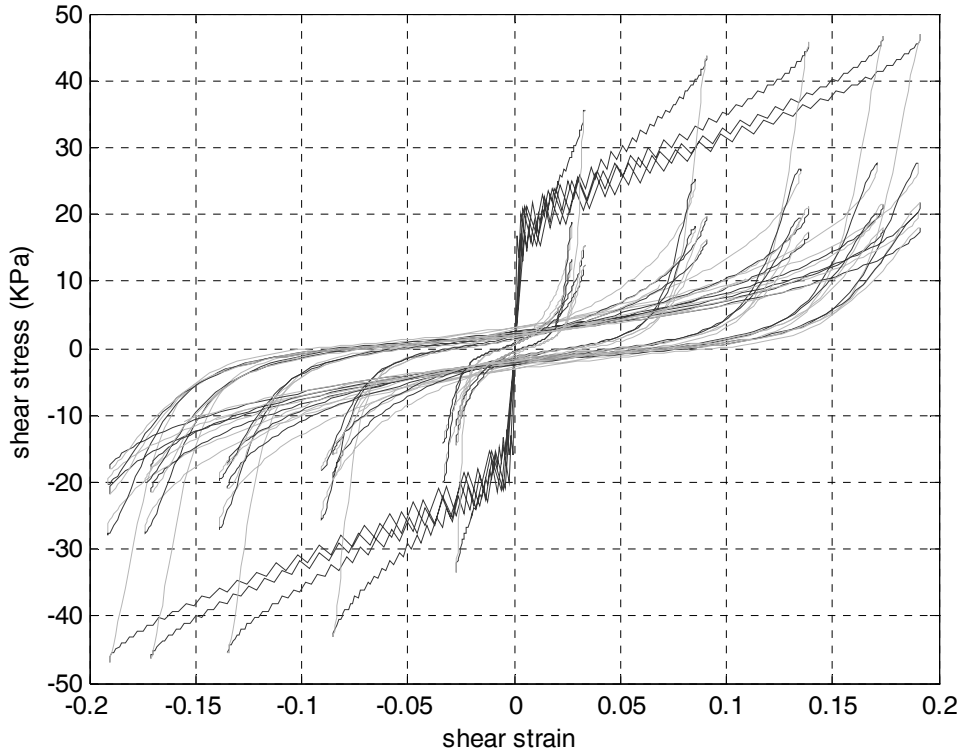


Figure 7.5 Initial loading and hysteresis loops for the virtual shear mechanisms

According to Figure 7.4, it is clearly observed that ,the code successfully represents the change in strength and stiffness of the soil single soil element under the applied dynamic loading conditions.The S-shape of the loops is due to the change in the effective mean stress state of the soil element (which is dilative for the studied case) occuring in parallel with the development of pore water pressure. The variation of effective mean stres, pore water pressure and the shear stres of the soil element with time are shown in Figures 7.6, 7.7 and 7.8, respectively.

The variation of effective mean stress state of the soil element causes changes in the shear modulus, shear strength and reference strain as shown in Figure 7.9, 7.10 and 7.11, respectively.

A closer examination of the figures reveal that soil parameters such as shear strength, stiffness, effective mean stress are all dependent on the computed excess pore water pressure, which is the expected type of performance from the model. The

sudden reduction in shear modulus, however, is somehow large and needs further study.

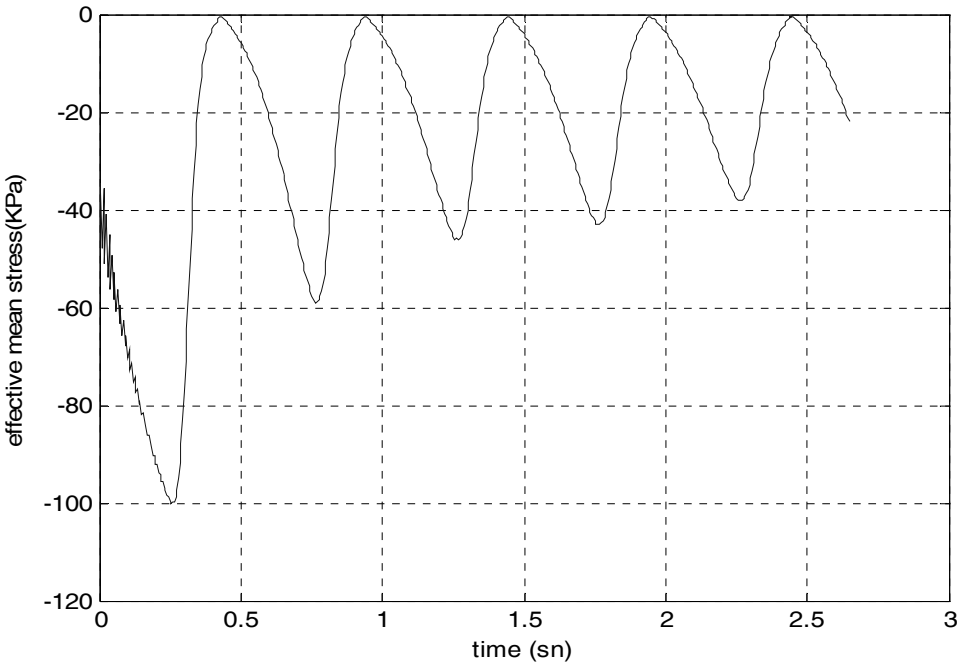


Figure 7.6 Variation of effective mean stress with time

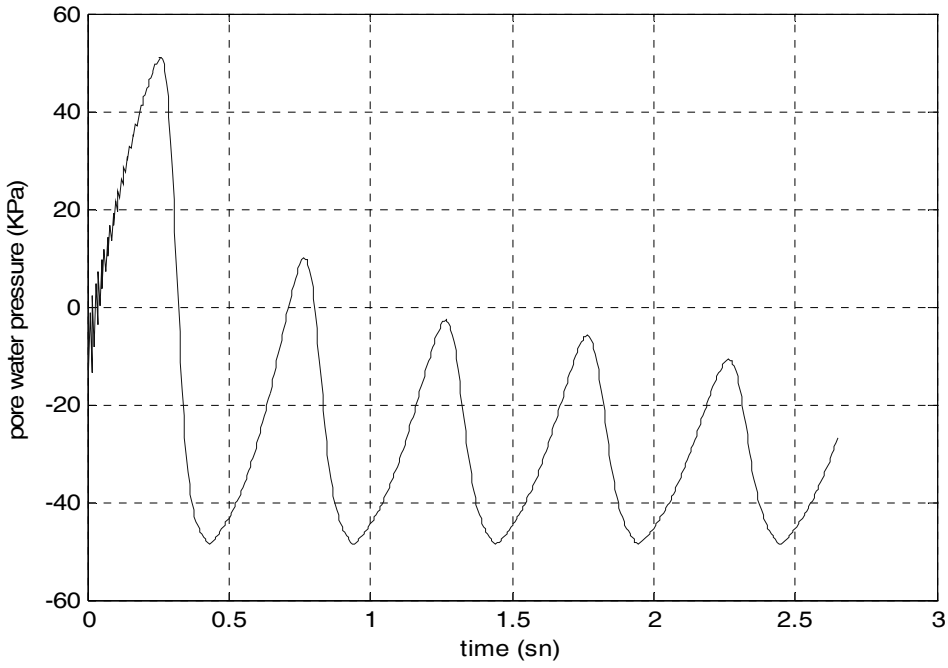


Figure 7.7 Variation of pore water pressure with time

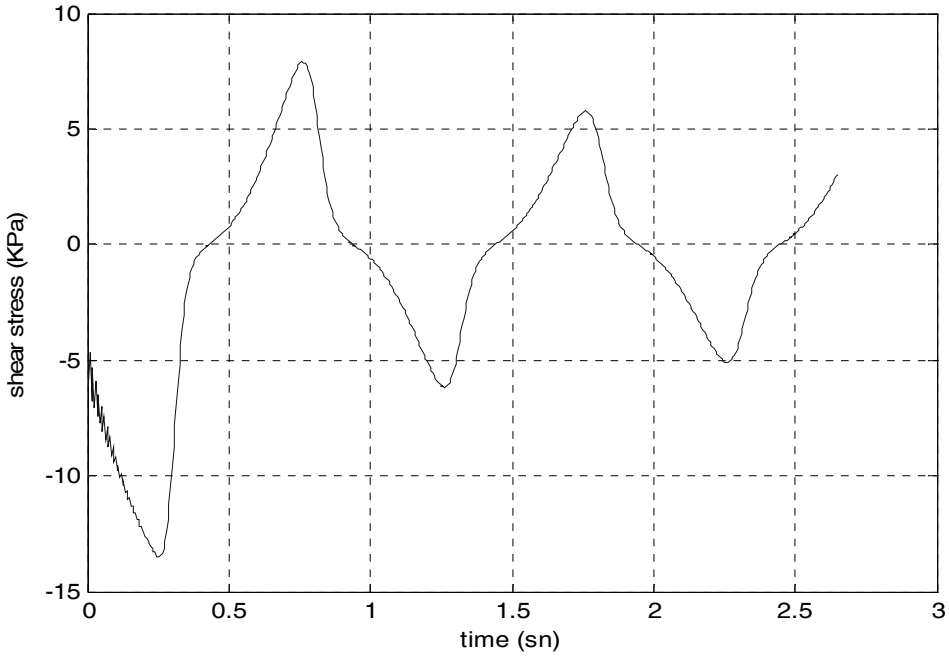


Figure 7.8 Variation of shear stress with time

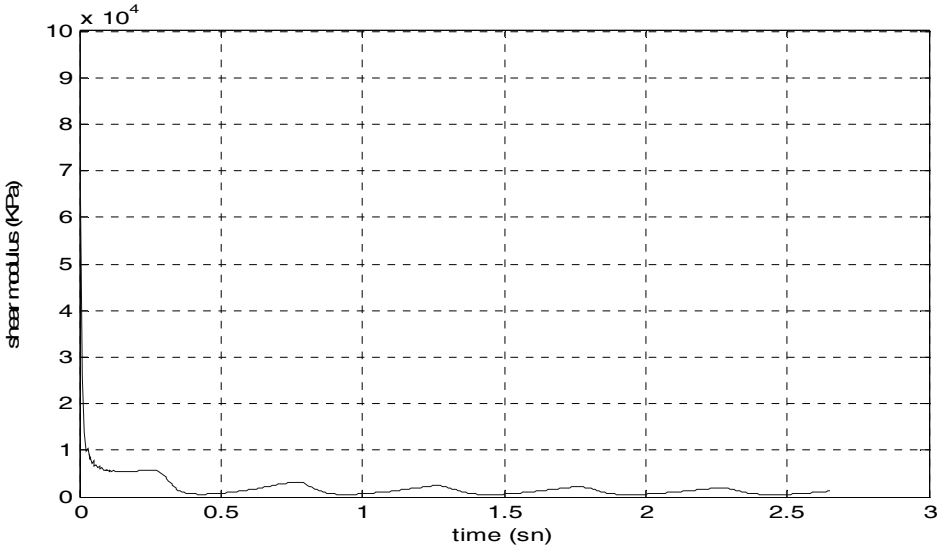


Figure 7.9 Variation of shear modulus with time

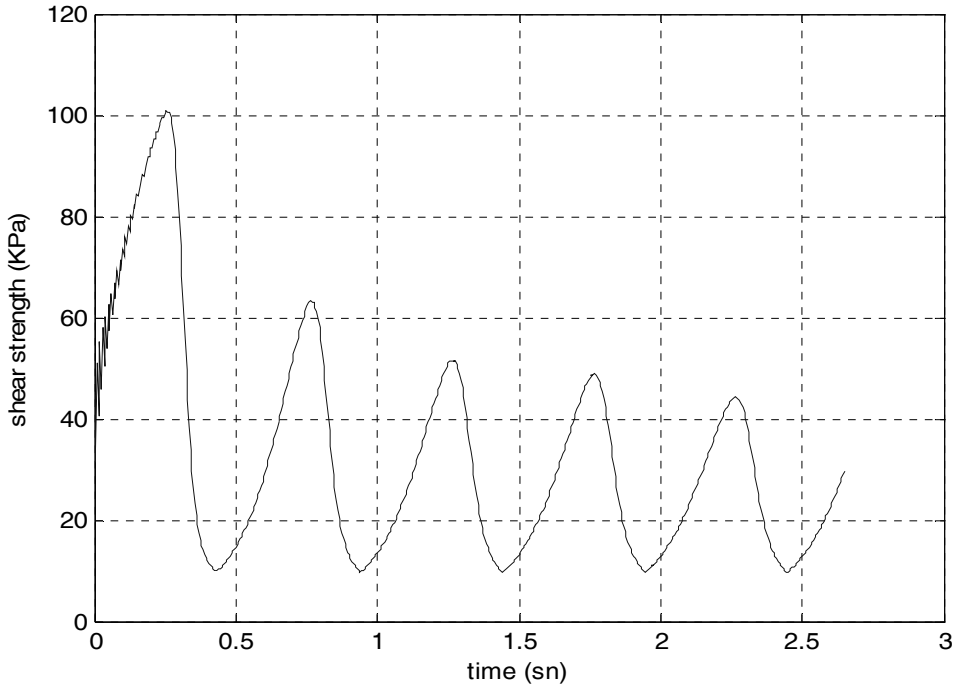


Figure 7.10 Variation of shear strength with time

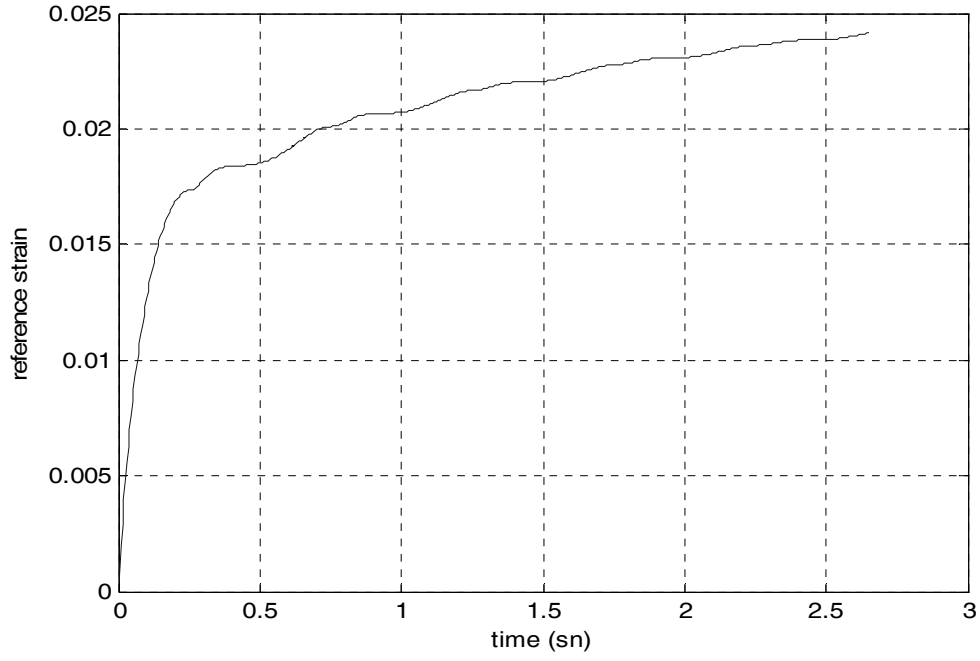


Figure 7.11 Variation of the reference strain with time

At this point it will be instructive to present Figure 7.12 which represents the numerical simulation of undrained behaviour of saturated sand sample under strain

controlled cyclic loading conditions. The soil constitutive model is one of the several constitutive models used in the verification studies of the VELACS Project. The soil-pile interaction problem investigated in this thesis is a displacement controlled problem and therefore, Figure 7.12 supports Figure 7.4 which represents the shear stress-shear strain curve of the single element case. The single element case shows that the model part of the FEM code works.

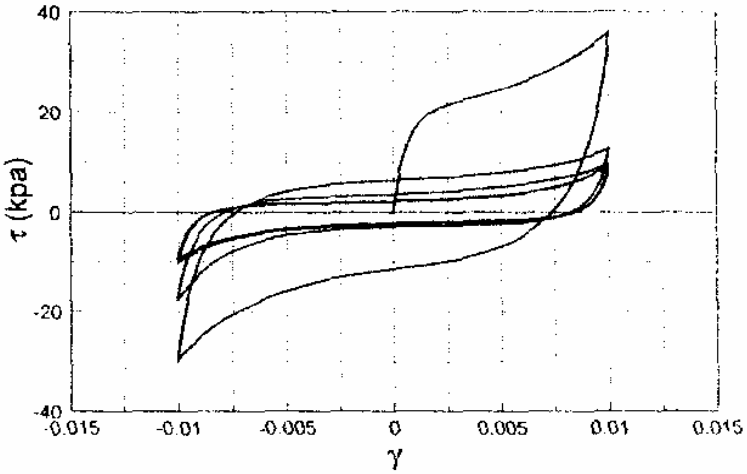


Figure 7.12 Undrained Simple Shear Response to a Strain Controlled Cyclic Loading (Isotropic Initial Confining Pressure, No lateral Strains are allowed) (Lie, 1993)

7.3.2 Assembled Case

The real interaction between the rigid pile section and the saturated sand medium under dynamic loading conditions was analysed by using the mesh shown in Figure 7.13. This is the same mesh used in linear dynamic analysis in Chapter 5 of this thesis. The results will be analysed by using the stripes of elements which enable the analysis of the behavior according to direction all around the rigid pile section.

During nonlinear dynamic finite element analysis, erroneous results can occur because of the convergence problems due to inadequate selection of the mesh, of the time integration parameters and mishandling of the nonlinear integration algorithm (Newton-Raphson algorithm). Figure 7.14 shows such a problem in which the finite

element mesh can not demonstrate the expected deformation mode since the solution quickly converges out after a few time steps. This was one of the problems encountered during the construction of the FEM code. At this point, such a problem has been necessary to mention since it emphasizes the difficulties of dealing with nonlinear dynamic finite element analysis.

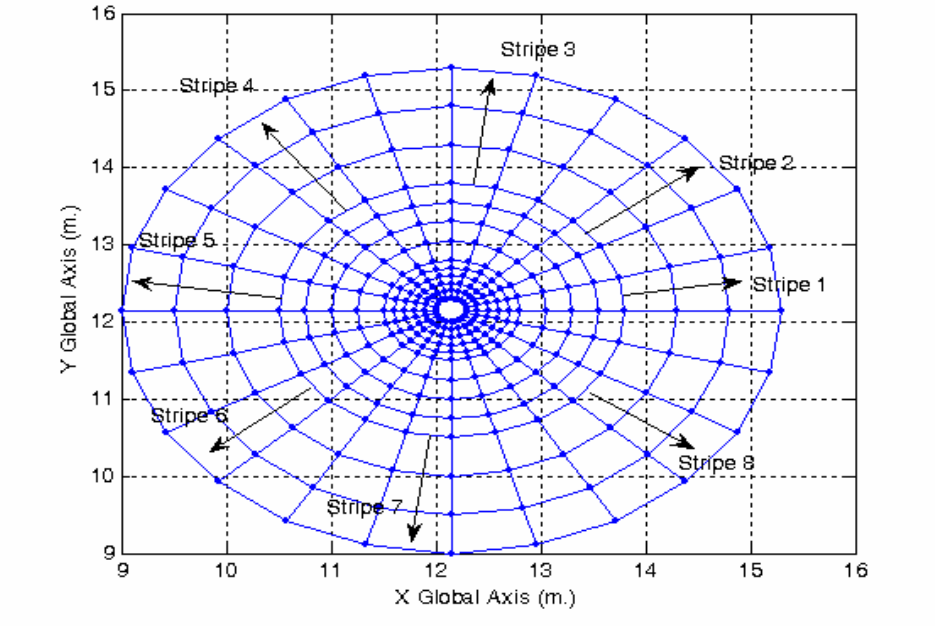


Figure 7.13 Finite element mesh used in nonlinear dynamic FEM analysis

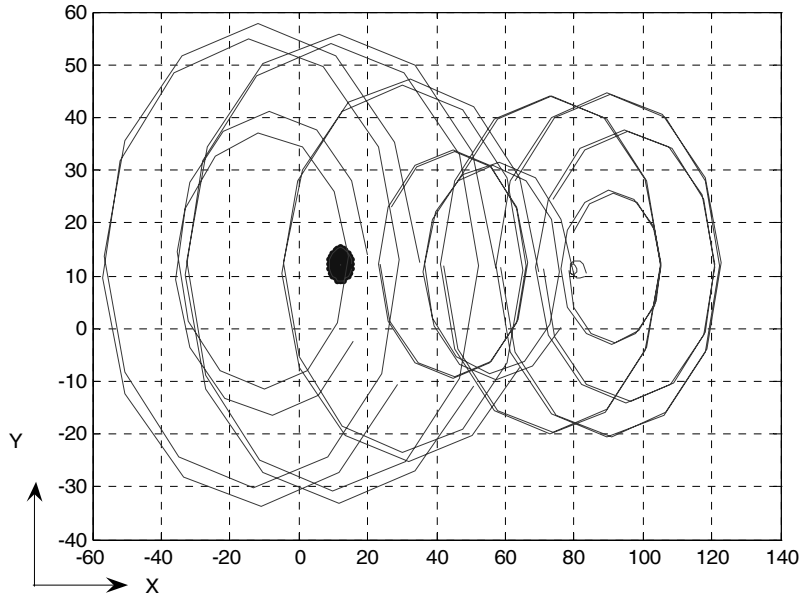


Figure 7.14 Inappropriately deformed mesh of the SSPSI problem analysis

7.3.2.1 Analysis Results

A sample run using the model parameters of the loose Fuji River Sand (Iai, et al., 1992) was performed under cyclic excitation with frequency of 1 Hz. Prescribed sinusoidal displacements, velocities and accelerations applied on the pile boundary with the soil medium under the frequency of 1Hz have been presented in Figures 5.19 through 5.21. Uncoupled analysis has been performed. Model parameters used in the sample run are given as:

$K_a = 270500\ KPa$
$G_{ma} = 103700\ KPa$
$p_1 = 0.45$
$p_2 = 1.4$
$w_1 = 2.0$
$S_1 = 0.0035$
$c_1 = 1.0$
$\sin\phi_f = 0.87$
$\sin\phi_p = 0.42$
$H_m = 0.3$
$n = 0.45$
$K_f = 2x10^6$ KPa

Figures 7.15 through 7.23 represent the variation of σ_x with time for the selected elements on each stripe. Tendency of σ_x to decrease in the neighbourhood of stripes 3 and 7 is the expected behaviour and very similar to the behaviour observed in the dynamic linear analysis. In addition, sign of σ_x is negative when the pile applies compressive stresses on the sand medium while they become negative in case of extension stresses (i.e when the pile is unloading). Change of sign with the direction loading is best demonstrated in Figure 7.23 which is compatible with the expected. Here, the variation of σ_x with time is demonstrated for the first band of elements around the pile. Comparing the Figures 7.15 through 7.23, it will be seen that

maximum normal stresses are obtained in the bands of elements very near the rigid-pile section.

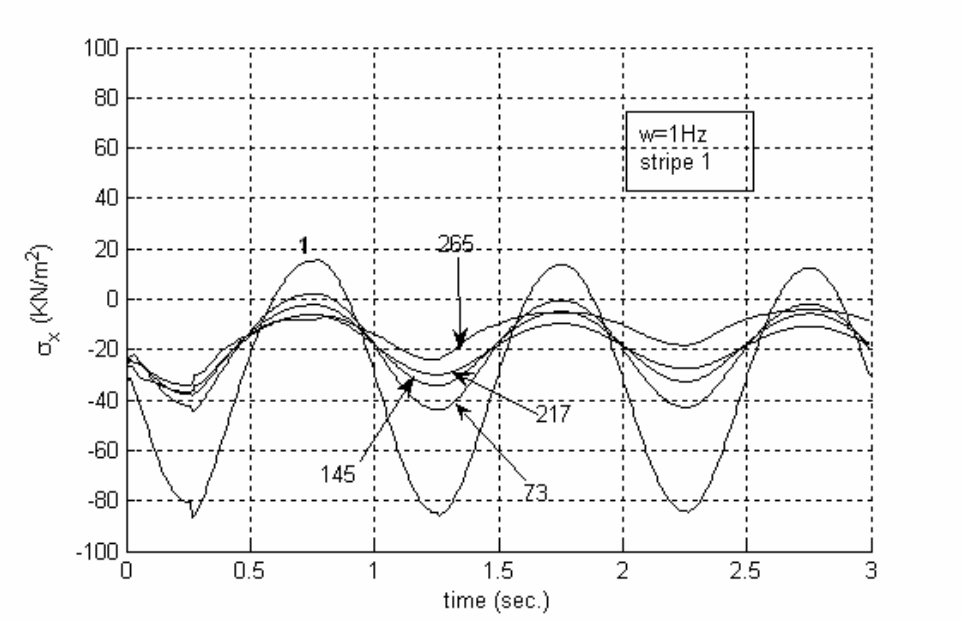


Figure 7.15 Variation of σ_x with time for the elements (1, 49, 121, 193, 265) along stripe 1

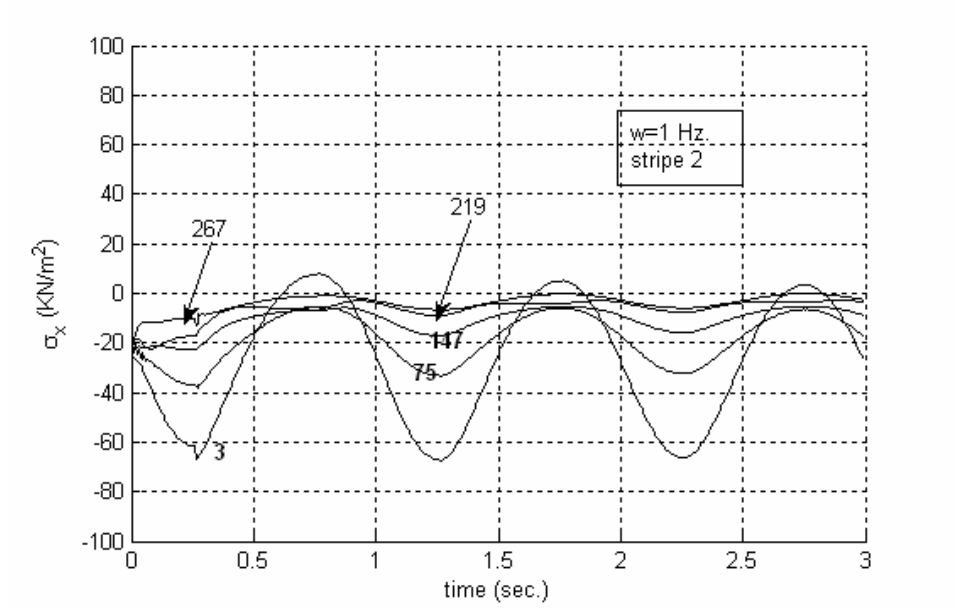


Figure 7.16 Variation of σ_x with time for the elements (3, 75, 147, 219, 267) along stripe 2

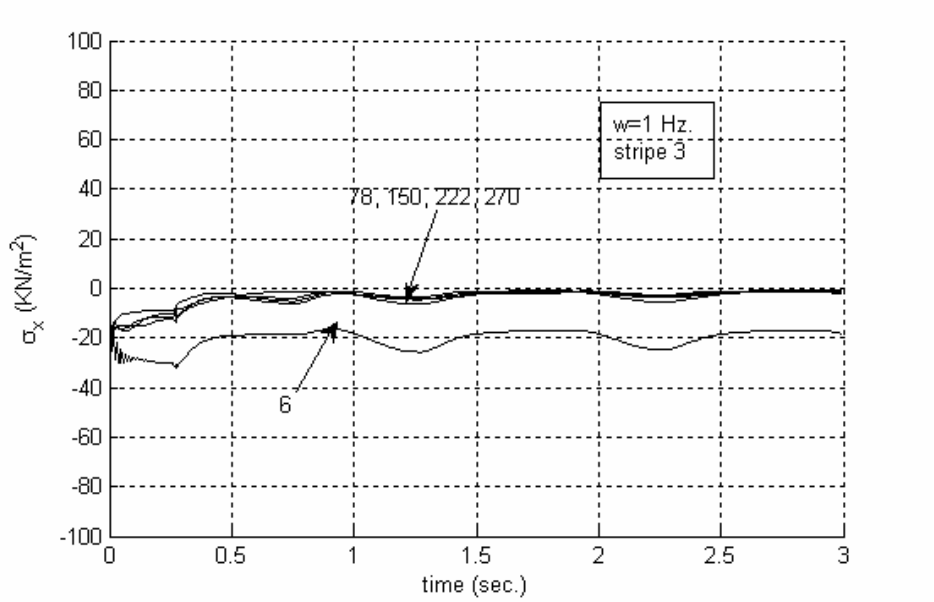


Figure 7.17 Variation of σ_x with time for the elements (6, 78, 150, 222, 270) along stripe 3

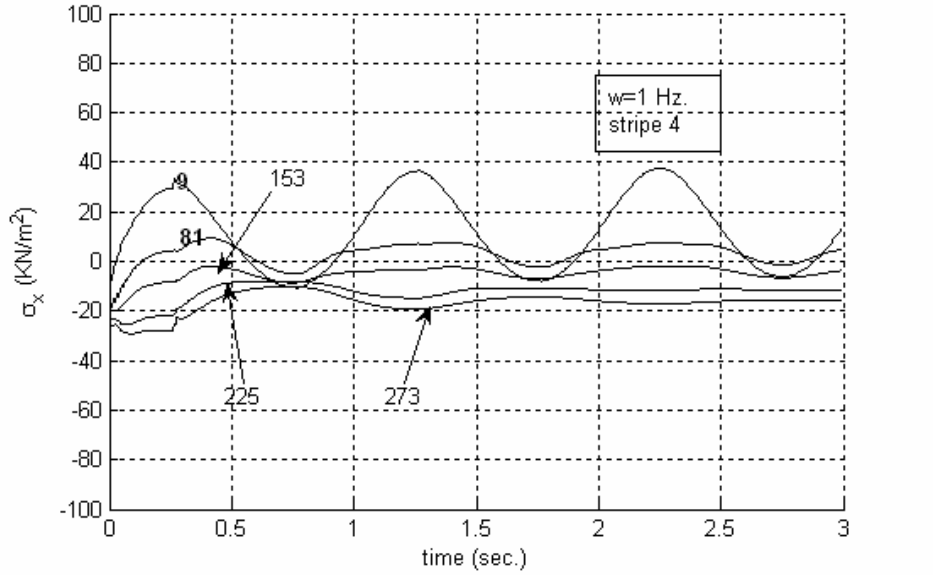


Figure 7.18 Variation of σ_x with time for the elements (9, 81, 153, 225, 273) along stripe 4

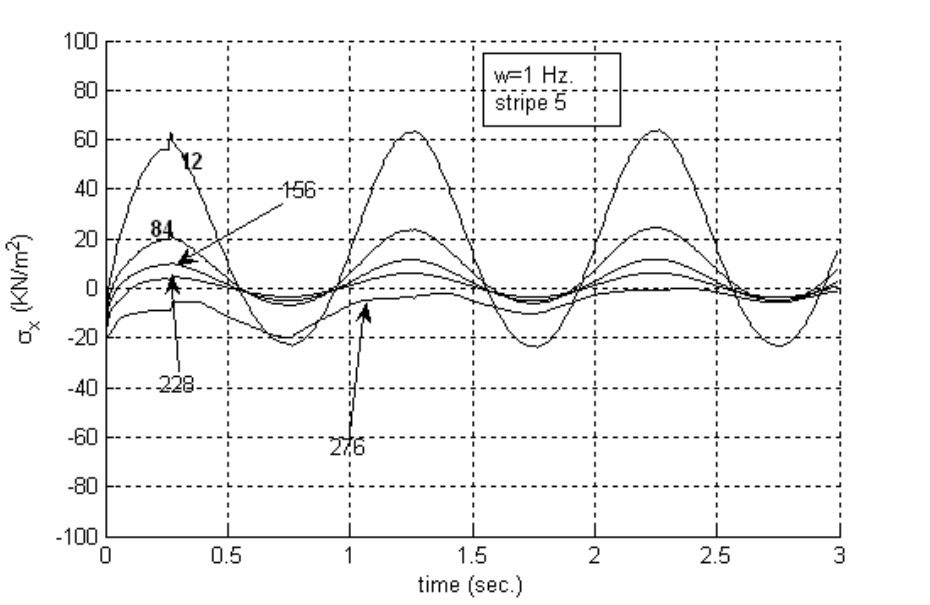


Figure 7.19 Variation of σ_x with time for the elements (12, 60, 156, 204, 276) along stripe 5

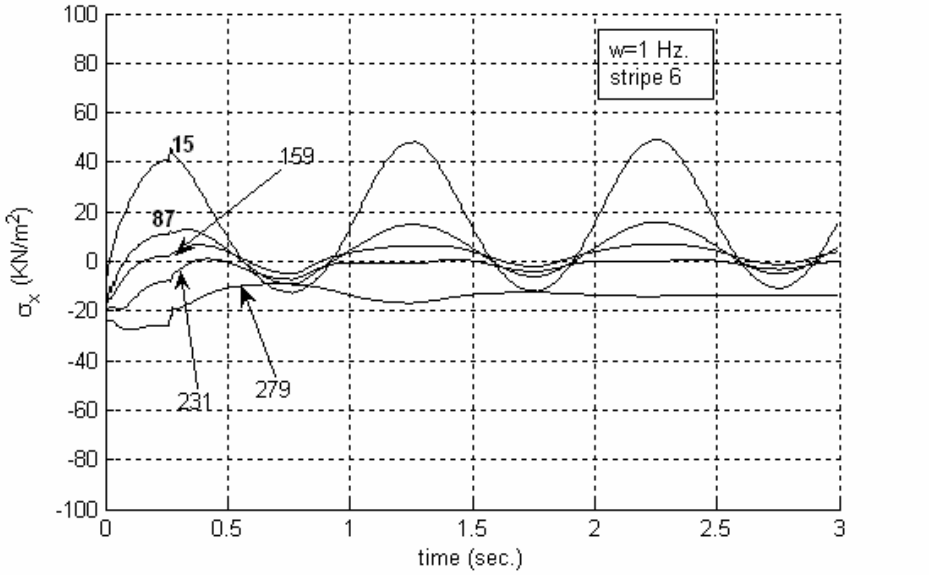


Figure 7.20 Variation of σ_x with time for the elements (15, 87, 135, 207, 279) along stripe 6

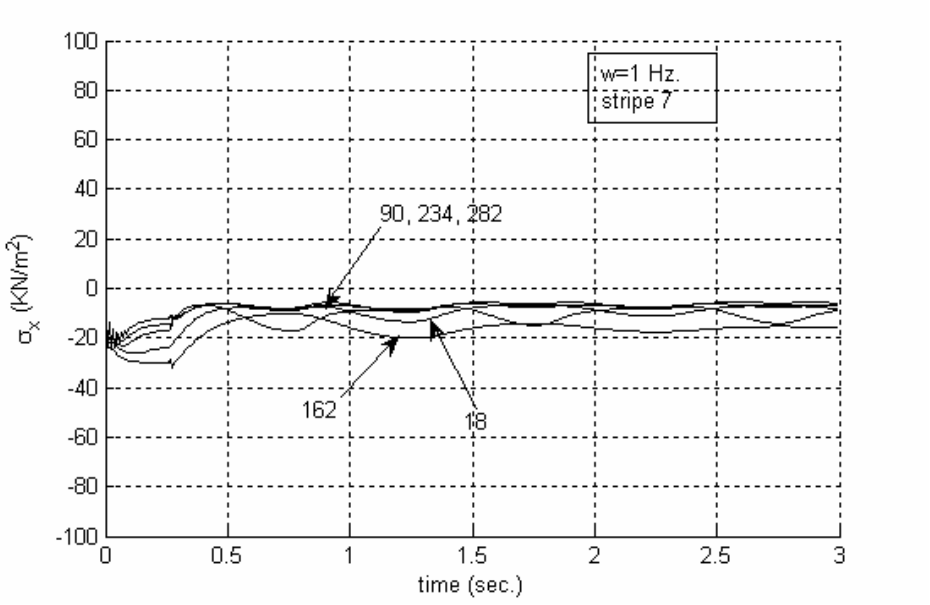


Figure 7.21 Variation of σ_x with time for the elements (18, 90, 186, 234, 282) along stripe 7

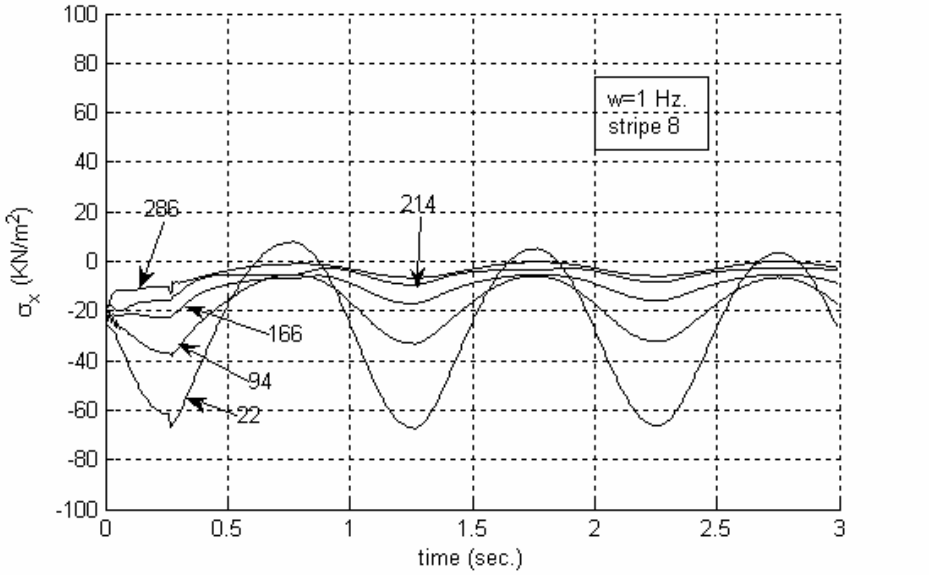


Figure 7.22 Variation of σ_x with time for the elements (22,118, 166, 238, 286) along stripe 8

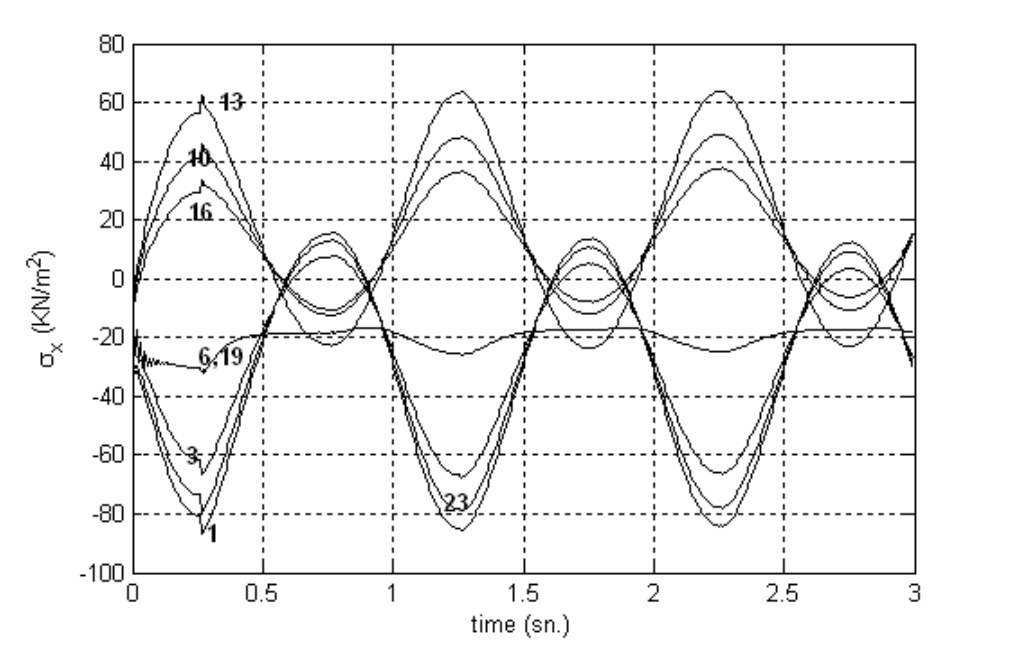


Figure 7.23 Variation of σ_x with time for the elements in the first band around the rigid pile section

Figures 7.24 and 7.25 represent the variation of σ_x and σ_y with the radial distance from the pile for each stripe, respectively. The calculated data corresponding to the end of initial loading (i.e 40 th. time step) is used for the construction of these figures. σ_x and σ_y have the tendency to decrease with the radial distance from the pile for all the stripes other than 3 and 7. These stresses reach their minimum values at stripes 3 and 7.

Figures 7.26 through 7.35 demonstrate the variation of shear stress and shear strain in the deforming medium around the rigid pile section. In Figures 7.34 and 7.35, calculated data represents the stress state corresponding to the end of the initial loading. Shear stress and shear strain reach their maximum value at stripes 3 and 7 which is expected. Besides, in the radial directions, shear stres and shear strain gradually increase at a distance in the vicinity of the pile and as the radial distance increases, shear stres and shear strain gardually decrease. This is the behaviour in all the stripes other than stripes 3 and 7. In these stripes, shear stres and shear strain have the max values at the pile boundary and they gradually decrease with the radial distance.

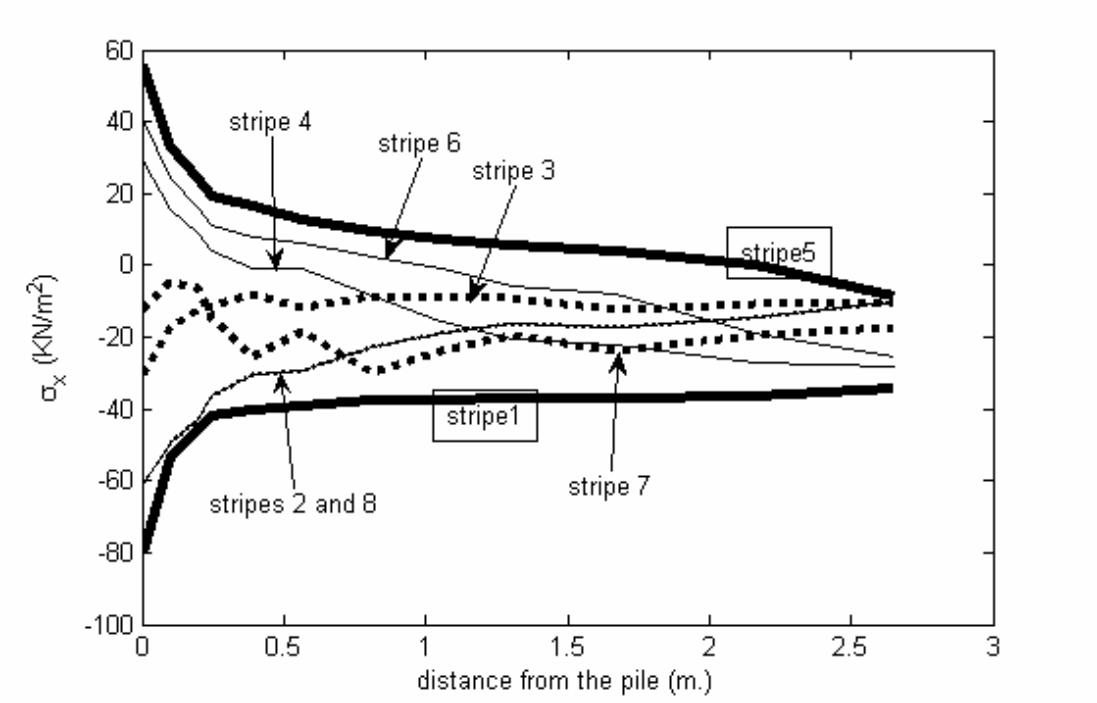


Figure 7.24 Variation of σ_x with radial distance from the pile for each stripe
(Data represents the stress state at the end of the initial loading, 40 th. time step)

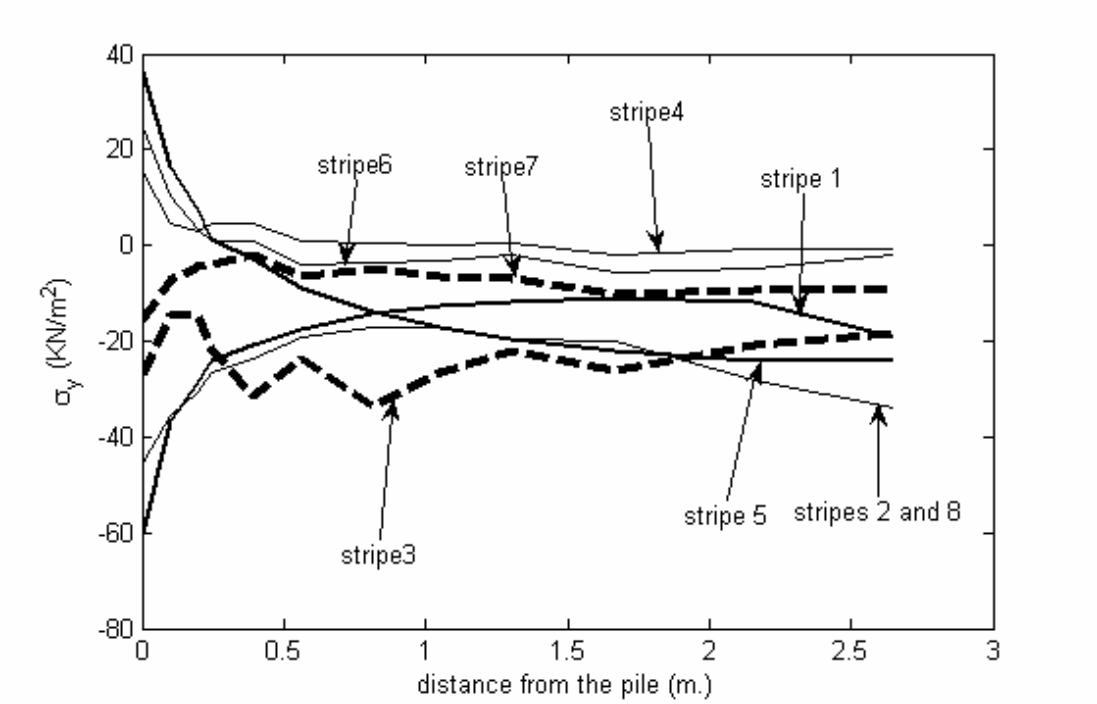


Figure 7.25 Variation of σ_y with radial distance from the pile for each stripe
(Data represents the stress state at the end of the initial loading, 40 th. time step)

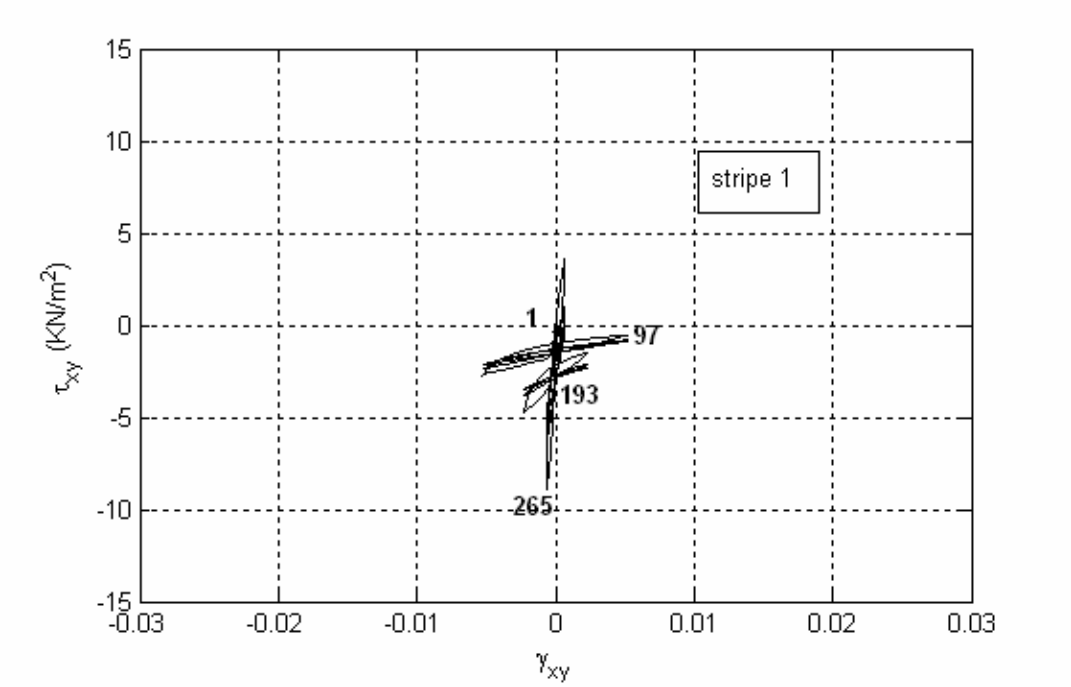


Figure 7.26 Variation of τ_{xy} with time along stripe 1

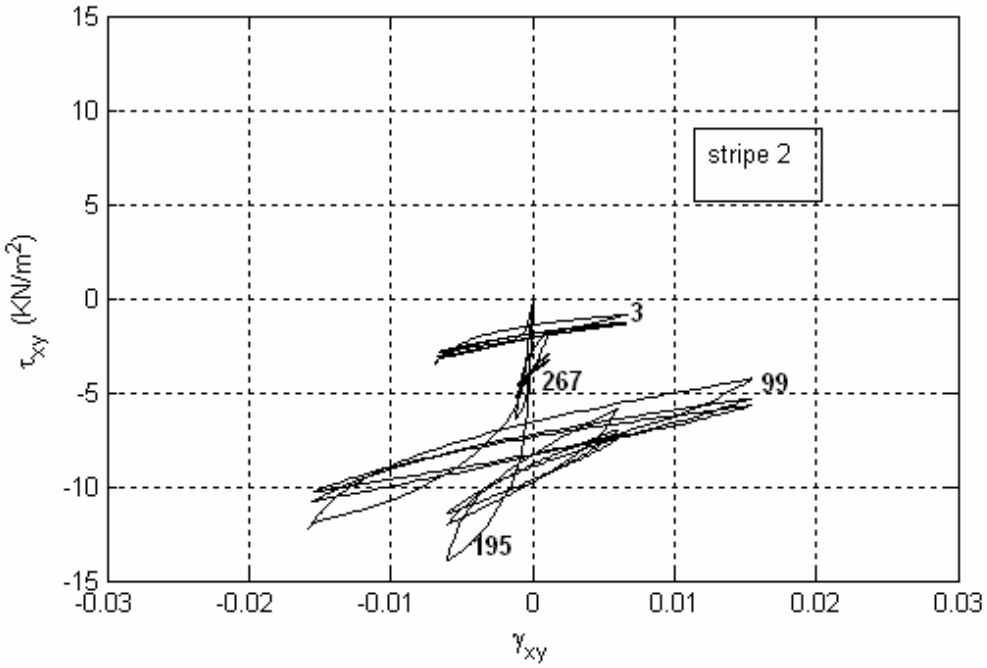


Figure 7.27 Variation of τ_{xy} with time along stripe 2

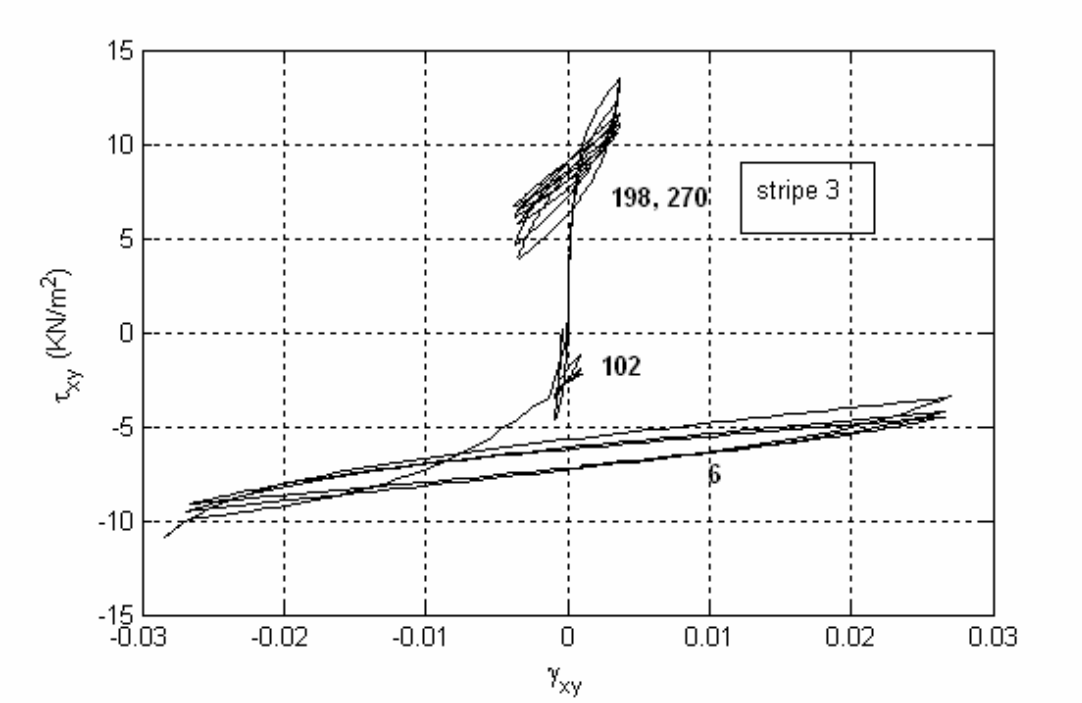


Figure 7.28 Variation of τ_{xy} with time along stripe 3

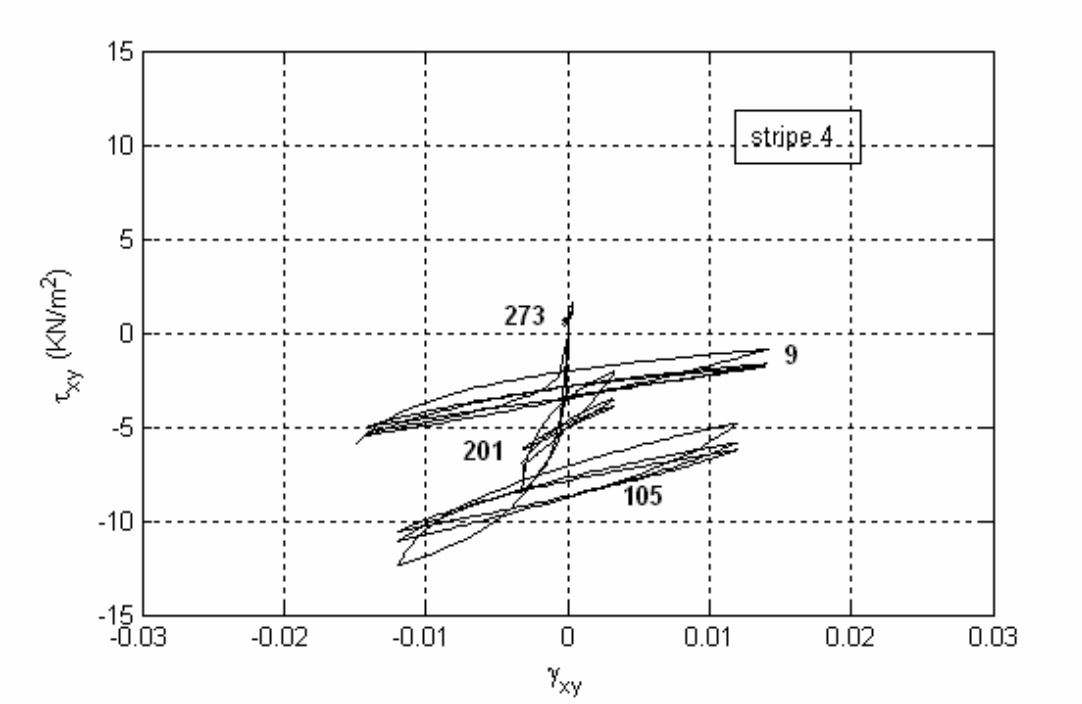


Figure 7.29 Variation of τ_{xy} with time along stripe 4

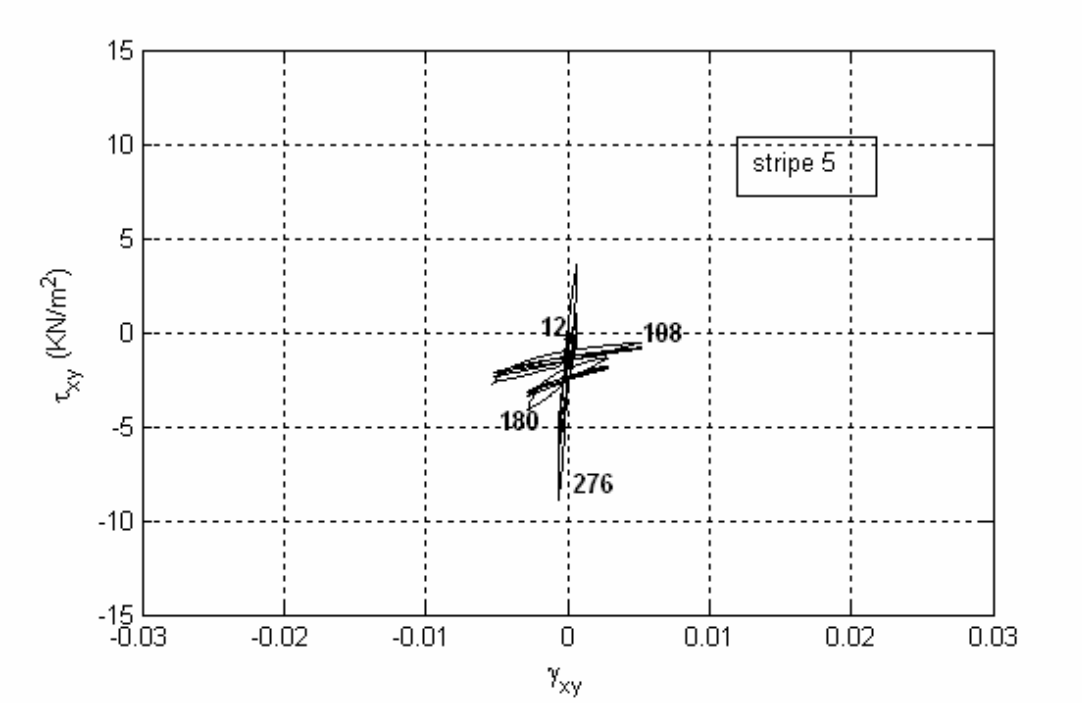


Figure 7.30 Variation of τ_{xy} with time along stripe 5

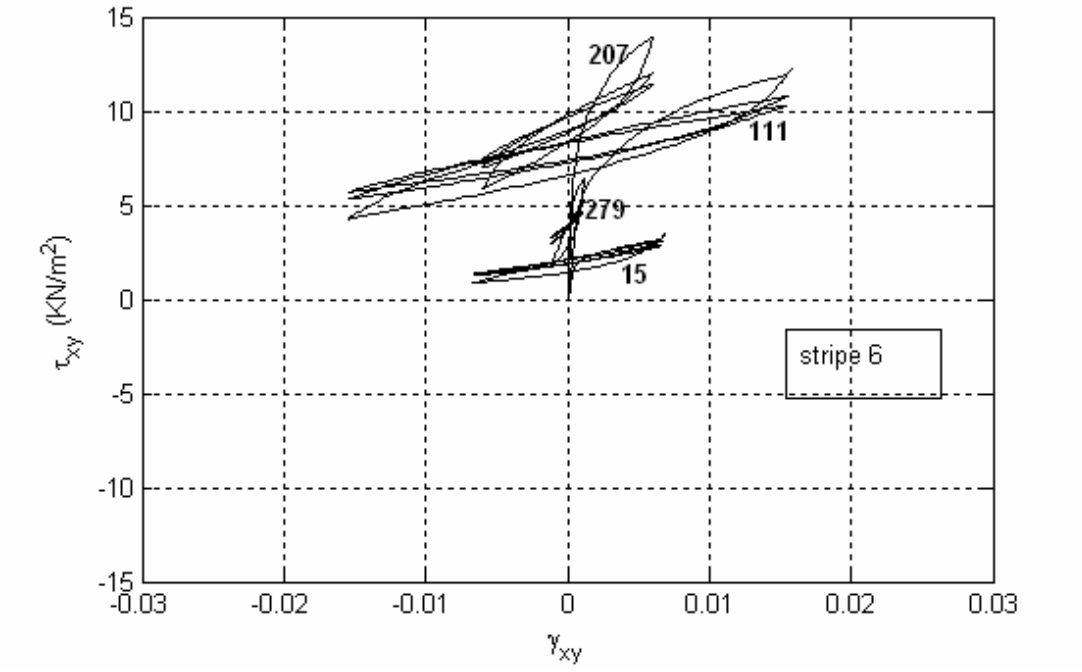


Figure 7.31 Variation of τ_{xy} with time along stripe 6

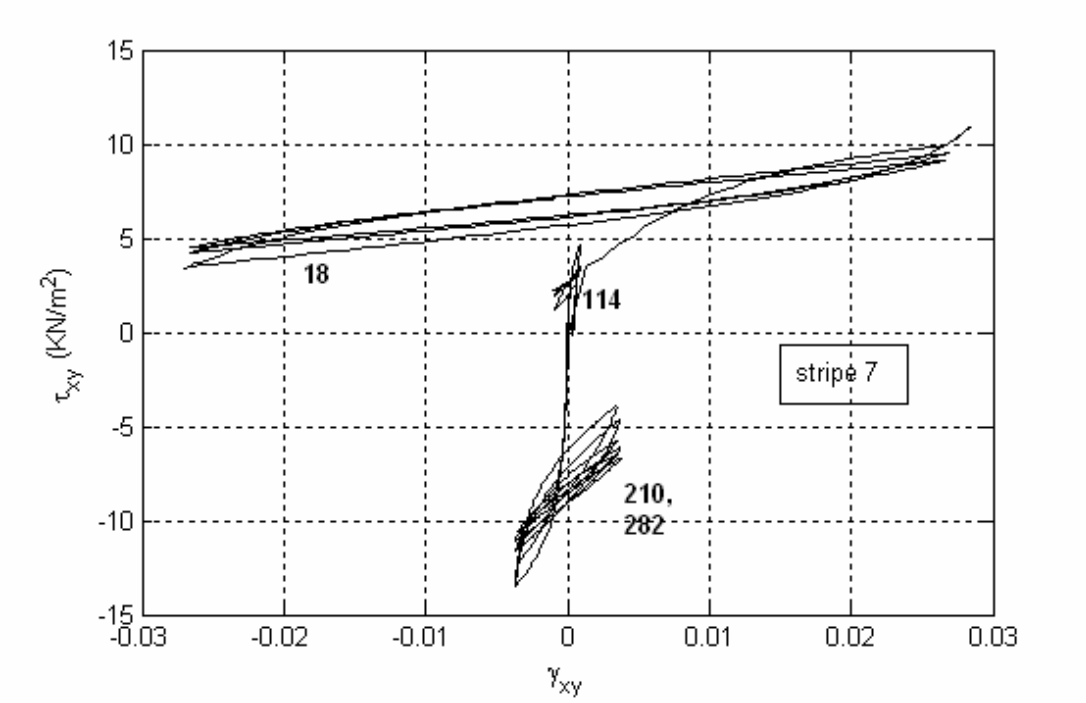


Figure 7.32 Variation of τ_{xy} with time along stripe 7

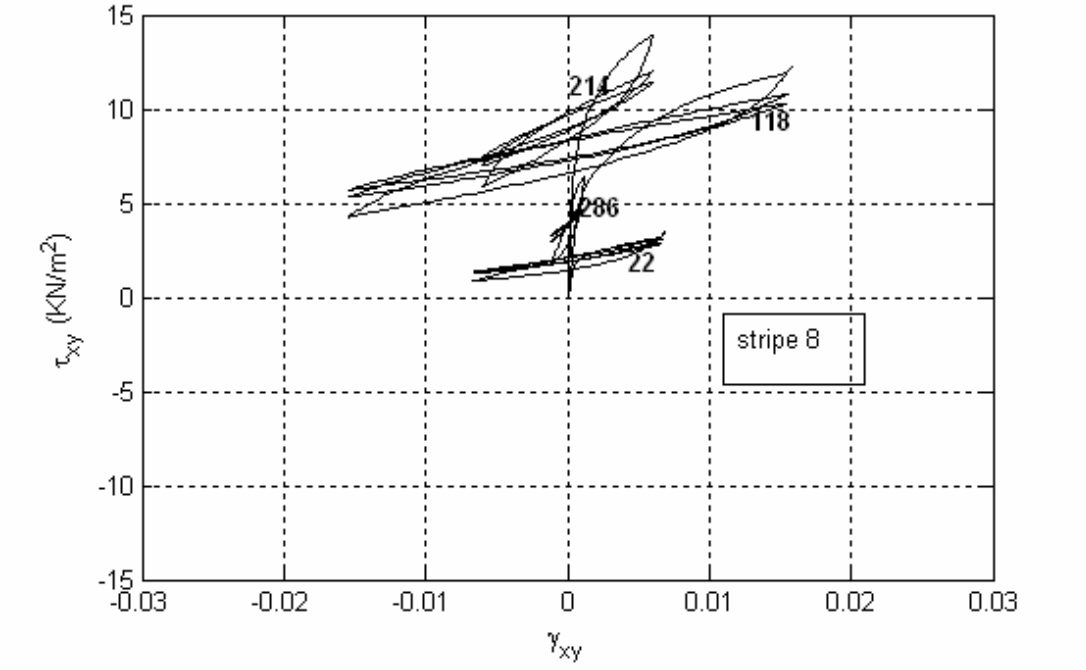


Figure 7.33 Variation of τ_{xy} with time along stripe 8

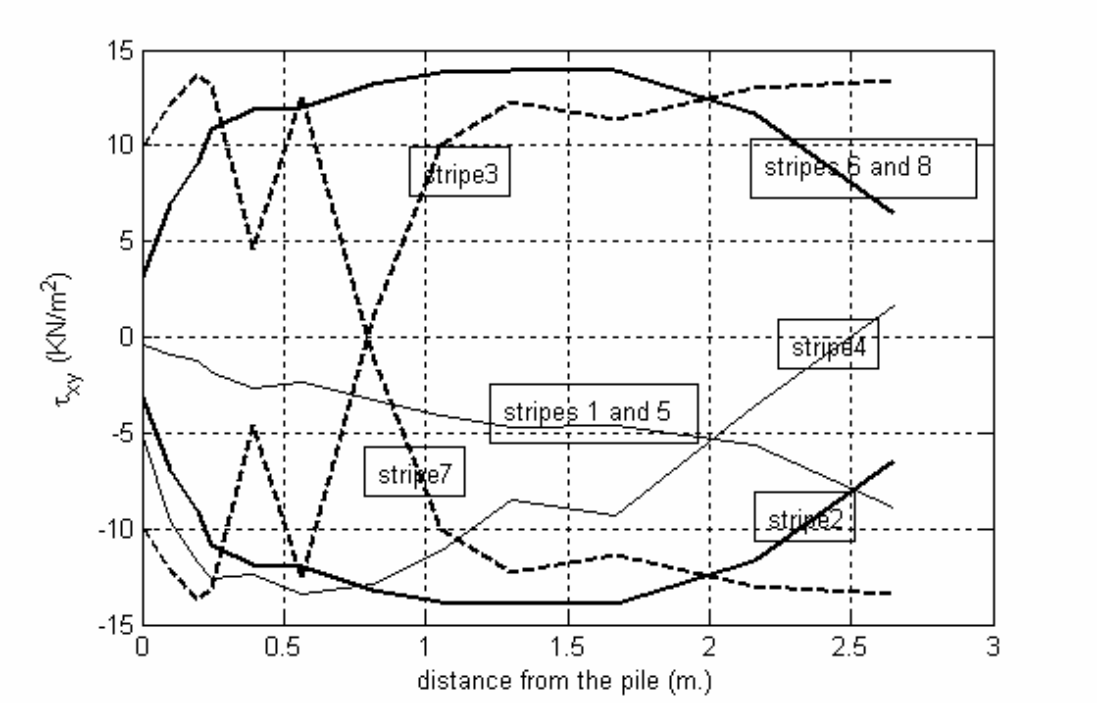


Figure 7.34 Variation of τ_{xy} with radial distance from the pile for each stripe
(Data represents the stress state at the end of the initial loading, 40 th. time step)

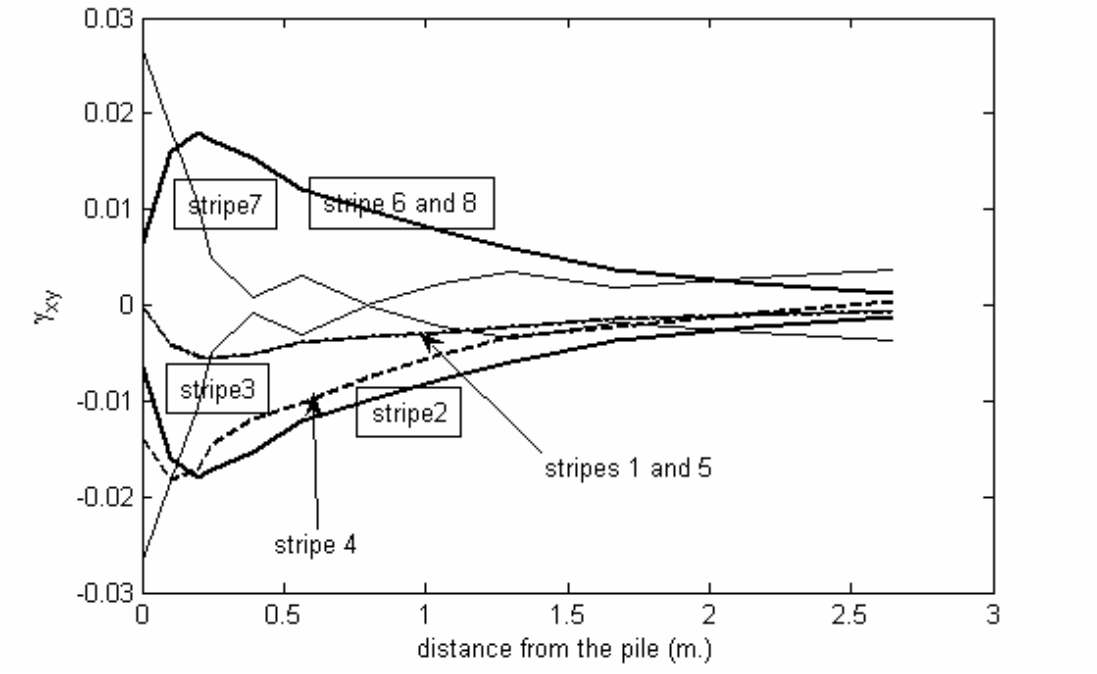


Figure 7.35 Variation of γx_y with radial distance from the pile for each stripe
(Data represents the stress state at the end of the initial loading, 40 th. time step)

CHAPTER EIGHT

CONCLUSIONS

An attempt has been made to develop a nonlinear dynamic finite element algorithm to study seismic-soil-pile-structure interaction problem. The motivation behind this attempt was the investigation of seismic inertial soil-pile-structure interaction effects on the response of the surrounding soil to the laterally deforming pile. Emphasis has been given to the understanding of excess pore pressure development since this parameter governs the mechanism by determining effective stress field around the pile.

Such an analysis needs the mathematical formulation of the dynamic porous medium, an adequate soil constitutive model representing the problem, the robust numerical procedures to be used in analysis with the finite element method and adequate determination of boundary conditions.

In the proposed model, sinusoidal displacement boundary conditions are applied at the X global direction degree of freedoms of the nodes along the perimeter of the pile section. The nodes on the outer perimeter of the sand medium are constrained in both global X and Y directions. The fact that the sand medium should follow the rigid pile section while the pile makes cyclic displacements is represented adequately by this boundary condition configuration. Physically, the pile moving in the global positive X direction, generates compressive strains on the soil medium in front (right half of the mesh) and passive lateral stress state is formed. The reverse is true for the soil medium in back of the pile in which extension strains are generated and in active state of stress. Therefore, the failure conditions should be defined seperately for active and passive parts which should be taken into account in the constitutive model. Only with this modification, the model is expected to work properly for left and right halves of the finite element mesh.

The finite element discretization and solution of dynamic governing differential equations were made using well established FEM solution procedures. Governing equations for the dynamic behavior of porous medium were presented by Biot (1941) which are considered to be fully coupled. However, it is possible to obtain simpler coupled sets by applying adequate assumptions related with the nature of the problem. u-p coupled set and the coupled

formulation with undrained assumption can be used for the undrained dynamic analysis of saturated sand medium. u-p coupled set is suitable for implicit integration procedures. Additionally, when undrained conditions are considered, it is advised to apply more elaborate solution procedures (Zienkiewicz et.al, 1999). On the other hand, coupled system with the undrained assumption can be solved by using explicit integration procedures (Zienkiewicz et al., 1978).

Coupled formulation with undrained assumption is used in the current version of the code which is solved using implicit integration procedures. In addition, u-p coupled formulation set which is completely effective stress based, can be effectively used in order to determine the distribution of pore pressures in saturated sand medium around the pile under dynamic loading conditions.

A state of the art strain space plasticity constitutive model has been employed in the finite element program. It should be noted that both contractive and dilative soil behavior are accounted for in the utilized model. The effect of principal stress axes rotation is modelled in the framework of generalised plasticity by representing the shear mechanism with multiple shear mechanisms placed in different directions. Besides, the model involves numerical robustness features enabling computation of the soil response near failure line where the soil attains small strength and stiffness.

Linear dynamic analysis of the soil-pile section interaction gave satisfactory results about the stress distribution around the pile under cyclic excitation (i.e sinusoidal prescribed displacements applied at the pile boundary with the soil medium). In case of nonlinear dynamic analysis, in which the strain space plasticity model has been incorporated into the FEM code, single element case gave satisfactory results indicating that the model subroutine of the code works properly. In case of nonlinear dynamic analysis with the assembled mesh, calculated stress distribution around the pile was compatible with the expected.

It should be stated that a very comprehensive work was made and a nonlinear FEM algorithm was developed which can be considered as a basis for a powerful analysis tool for the study of seismic soil dynamic and soil-pile interaction problems. The study was rather useful in the accumulation of a knowledge base for the analysis of similar nonlinear problems in other engineering fields.

REFERENCES

Altun, S. (2002). Asal Eksen Yönlerinin Döngüsünün Kumların Drenajsız Koşullardaki Dinamik Mukavemet Özelliklerine Etkisi. *Beşinci Uluslararası İnşaat Mühendisliğinde Gelişmeler Kongresi*. İstanbul Teknik Üniversitesi, İstanbul.

Ashour, M., Norris, G., & Pilling, P. (1998). Lateral loading of a pile in layered soil using the strain wedge model. *Journal of Geotechnical and Geoenvironmental Engineering*, 124(4), 203-315.

Bathe, K. J. (1996). *Finite element procedures* (6th. ed.). New Delhi:Prentice Hall of India.

Biot, M.A. (1941). General theory of three-dimensional consolidation. *J.appl.Phys*., (12), 155-164.

Finn, L. D. W., Lee, K. W., & Martin, G. R., (1977). An effective stres model for liquefaction. *ASCE, Journal of Geotechnical Engineering Division*, (GT6), 517-533.

Finn, W. L. D. (1979). Critical Review of dynamic effective stress analysis. In *Proc. 2nd U.S. National Conference on Earthquake Engineering* (853-867). Stanford University.

Ghaboussi, J., & Dikmen, S. U. (1978). Liquefaction analysis of horizontally layered sands. *Proc. ASCE*, 104 (GT3), 341-356.

Gutierrez, M., Ishihara, K. & Towhata, I. (1991). Flow Theory for Sand During Rotation of Principal Stress Direction. *Soils and Foundations*, 31(4), 121-132

Gutierrez, M., Ishihara, K. & Towhata, I. (1991). Noncoaxiality and stress-dilatancy relations for granular materials. In Beer, Booker & Carter, (Eds.) *Computer Methods and Advances in Geomechanics* (625-630). Balkema, Rotterdam.

Gutierrez, M., Ishihara, K. & Towhata, I. (1993). Model for the Deformation of Sand During Rotation of Principal Stress Directions. *Soils and Foundations*, 33(3), 105-117.

Hamada, M. & O'Rourke, T.D. (1992). Case studies of liquefaction and lifeline performance during past earthquakes. *Technical Report NCEER-92-0001.*

Holtz, R.D., & Kovacs, W.D. (1981). *An introduction to geotechnical engineering.* New Jersey: Prentice Hall.

Hughes, T. J. R. (2000). *The finite element method-linear static and dynamic finite element analysis.* New York: Dover Publications.

Iai, S. (2002) 'Analysis of soil deformation around a cylindrical rigid body', US-JApan Seminar Anchorage, Alaska, June 26-27.

Iai, S. (1993). Concept of effective strain in constitutive modeling of granular materials. *Soils and Foundations*, 33(2), 171-180.

Iai, S. (1993a). Micromechanical background to a strain space multiple mechanism model for sand. *Soils and Foundations*, 33(1), 102-117.

Iai, S. (1993b). Three dimensional formulation and objectivity of a strain space multiple mechanism model for sand. *Soils and Foundations*, 33(1), 192-199.

Iai, S., Matsunaga, Y., Kameoka, T. (1992). Strain space plasticity model for cyclic mobility. *Soils and Foundations*, 32(2), 1-15.

Iai, S., Matsunaga, Y., Kameoka, T. (1990). Strain space plasticity model for cyclic mobility. *Report of the Port and Harbour Research Institute*, 29(4), 27-56.

Iai, S., Matsunaga, Y., Kameoka, T. (1990). Parameter identification for a cyclic mobility model. *Report of the Port and Harbour Research Institute*, 29(4), 57-83.

Irons, B. and Ahmad, S., (1986) 'Techniques of finite elements', Ellis Horwood Series in Engineering Science, pp.529.

Ishihara, K. (1996). *Soil behavior in earthquake engineering.* Oxford: Clarendon Press.

Ishihara, K., Tatsuoka, F. & Yasuda, S., (1975). Undrained deformation and liquefaction of sand under cyclic stres. *Soils and Foundations,* 15 (1), 29-44.

Ishihara, K. & Towhata, I., (1982). Dynamic response analysis of level ground based on the effective stres method. In G. N. Pande and O. C Zienkiewicz (ed.). *Soil Mechanics-Transient and Cyclic Loads (133-172).* John Wiley and Sons,

Ishihara, K., Yamazaki, A. & Haga, K (1985). Liquefaction of K_0-Consolidated Sand Under Cyclic Rotation of Principal Stress Direction with Lateral Constraint. *Soils and Foundations*, 25(4), 63-74.

Kagawa, T. & Kraft, L. M. (1980). Lateral load-deflection relationships of piles subjected to dynamic loadings. *Soils and Foundations,* 20(4), 19-36.

Kagawa,T. & Kraft, L. M., (1981). Lateral pile response during earthquakes. *ASCE, Journal of Geotechnical Engineering Division,* (GT12), 1713-1731.

Kagawa, T. (1983). Lateral pile group response under seismic loading. *Soils and Foundations,* 23(4), 75-86.

Kaynia, A. M. & Kausel, E. (1982). Dynamic stiffness and seismic reponse of pile groups. *Research Report, R82-03*. Cambridge: MIT.

Lambe, T.W., & Whitman, R.V. (1969). *Soil Mechanics*. John Wiley & Sons.

Lewis, R.W. & Schrefler, B.A. (1987). *The finite element method in the deformation and consolidation of porous media.* Chichester: J.Wiley & Sons.

Liou, C. P., Streeter, V. L., & Richart, F. E., (1977). Numerical model for liquefaction. *Proceedings, ASCE*. 103 (GT6), 589-606.

Martin, T. R., Finn, W. D. L. & Seed, H. B. (1975). Fundamentals of liquefaction under cyclic loading. *Proceedings, ASCE,* 101 (GT5), 423-38.

Matsuoka, H. & Suzuki, Y. (1988). A constitutive model of sands directly expressed in general coordinates for evaluating principal stress rotation and the comparison of its predicted values with the measured values. In Saada & Bianchini (Ed.) *Constitutive Equations for Granular Non-Cohesive Soils* (403-426). Balkema, Rotterdam.

Miura, K., Miura, S. & Toki, S. (1986). Deformation Behavior of Anisotropic Dense Sand Under Principal Stress Axes Rotation. *Soils and Foundations*, 26(1), 36-52.

Mizuno, H. (1987). Pile damage during earthquakes in Japan (1923-1983). *Proceedings, Dynamic Response of Pile foundations, Geotechnical engineering division,* ASCE, Atlantic City, Spring Convention.

Mould Jr., J.C., Sture, S. & Ko, H.Y. (1985). Sand deformation tests with rotating principal stress directions. In *Fifth International Conference on Numerical Methods in Geomechanics* (531-538). Nagoya, Japan.

Nishimura, S. & Towhata, I., (2004). A three dimensional stress-strain model of sand undergoing cyclic rotation of principal stres axes. *Soils and Foundations*, 44(2), 103-116.

Nofal, H., (1998). *Analysis of nonlinear soil-pile ınteraction under dynamic lateral loading.* Dissertation, University of California.

Novak, M., (1974). Dynamic stiffness and damping of piles. *Canadian Geotechnical Journal,* 11(4), 574-598.

Ozden G., (1999). *Soil pile interaction ın loose cohesionless submerged soils.* Ph.D. Thesis, pp.258.

Ozden, G. & Kagawa, T. (2000). Experimental cyclic p-y curves for saturated loose sand. In *The International Conference of Geotechnical & Geological Engineering, Proceedings, Vol.2 Extended Abstracts and CD-ROM of complete manuscripts*. Australia.

Pastor, M., Zienkiewicz, O.C., Leung, K.H. (1985). Simple model for transient soil loading in earthquake analysis. II. Non-associative models for sands. *International Journal for numerical and Analytical Methods in Geomechanics*, 9, 477-498.

Potts, D.M., & Zdravkovic, L. (1999). *Finite element analysis in geotechnical engineering.* Londan: Thomas Telford.

Pradel, D., Ishihara, K. & Gutierrez, M. (1990). Yielding and flow of sand under principal stress axes rotation. *Soils and Foundations* 30(1), 87-99.

Ross, G., Seed, H., & Migliacio, R. (1973). Performance of highway bridge foundations. *In the Great Alaska Earthquake of 1964-Engineering, Comm. on the Alaska Earthquake of the Division of Eart Sciences, Natl. Research Council, Natl. Academy of Sciences, Wash., D.C.*

Sanchez-Salinero, I. (1983). Dynamic stiffness of pile groups: approximate solutions. *NSF Report.*

Sato, T., Shibata, T. & Kosaka, M. (1980). Dynamic behaviour and liquefaction of saturated soil. *In Proceedings, Int. Symp. Soils under Cyclic and Transient Loading* (523-532). Vol.2. Swansea.

Seed, H. B. & Idriss, I. M. (1967). Analysis of soil liquefaction: Niigata Earthquake. *Journal of Soil Mechanics and Foundation Division, ASCE*, 93(3), 83-108.

Sherif, M.A., Ishibashi, I. & Tsuchida, C., (1978). Pore pressure prediction during earthquake loading. *Soils and Foundations*, 18 (4), 19-30.

Symes, M.J.P.R., Gens, A. & Hight, D.W. (1984). Undrained anisotropy and principal stress rotation in saturated sand. *Géotechnique*, 34(1), 11-27.

Takeshima, Y., Sawada, S., Iai, S., Ichii, K., Ozutsumi,O., Adachi, M., Yoshida, A., Ikeda, T., Ootsuka, N. & Umeki,Y. (2004). Modelling of drainage behaviour for dynamic effective stres analysis by undrained condition. *13th. World Conference on Earthquake Engineering, Vancouver,B.C., Canada, Paper No.3146*

Towhata, I. & Ishihara, K. (1985). Modelling Soil Behavior Under Principal Stress Axes Rotation. In *Fifth International Conference on Numerical Methods in Geomechanics* (523-530). Nagoya, Japan.

Towhata, I. & Ishihara, K. (1985). Shear work and pore water pressure in undrained shear. *Soils and Foundations*, 25(3), 73-84.

Wang, S. & Reese, L. C. (1998). Design of pile foundations in liquefied soils. In Dakoulas, P. et.al (Ed.). *Proceedings, Geotechnical Earthquake Engineering and Soil Dynamics III, ASCE, Geotechnical Special Publication, No.75* (1331-1345) .

Yamazaki, F., Towhata, I. & Ishihara, K. (1985). Numerical model for liquefaction problem under multi-directional shearing. In *Fifth International Conference on Numerical Methods in Geomechanics* (399-406). Nagoya, Japan.

Yoshimi,Y., Kuwabara, F., & Tokimatsu, K. (1978). Two-dimensional pore pressure changes in sand deposits during earthquakes. In *Proceedings of 2nd Int Conf. Microzonation for Safer Construction, Research and Application Vol.3* (853-863). San Francisco.

Zienkiewicz, O. C., Chang, C. T., & Hinton, E. (1978). Non-linear seismic response and liquefaction. *International Journal of Numerical and Analytical Methods in Geomechanics,* 2, 381-404.

Zienkiewicz, O. C. & Shiomi, T. (1984). Dynamic behaviour of saturated porous media: the generalised Biot Formulation and its numerical solution. *International Journal for Numerical and Analytical Methods in Geomechanics,* 8, 71-96.

Zienkiewicz, O. C. & Taylor, R. L. (1991). *The finite element method-solid and fluid mechanics, dynamics and nonlinearity Vol 2. (4th.ed.).* McGraw-Hill International Editions.

Zienkiewicz, O. C. & Taylor, R. L. (1989). *The finite element method-basic formulation and linear problems Vol 1. (5th.ed.).* McGraw-Hill International Editions.

Zienkiewicz, O. C. & Taylor, R. L. (1985). Coupled Problems-A Simple Time-Stepping Procedure. *Communications in Applied Numerical Methods,* 1, 233-239.

Zienkiewicz, O. C., Chan, A. H. C., Pastor, M. (1988). Simple models for soil behavior and applications to problems of soil liquefaction. In Numerical Methods in Geomechanics (169-180). Balkema, Rotterdam.

Zienkiewicz, O. C., Chan, A. H. C., Pastor, M., Schrefler, B. A., Shiomi, T. (1999). *Computational geomechanics with special reference to earthquake engineering.* John Wiley & Sons.

Zienkiewicz, O. C., Pastor, M., Chan, A.H.C., Xie, Y.M. (1991). Computational approaches to the dynamics and statics of saturated and unsaturated soils. In P.K. Banerjee, R. Butterfield, (Ed). *Advanced Geotechnical Analysis*, *Developments in Soil MEchanics and Foundation Engineering -4* (1-43). Balkema, Rotterdam, Elsevier Applied Science.

Zienkiewicz, O. C., Leung, K.H., Hinton, E., Chang, C.T. (1982). Liquefaction and permanent deformation under dynamic conditions-numerical solution and constitutive relations. In G.N. Pande, O.C. Zienkiewicz, (Ed). *Soil Mechanics-Transient and Cyclic Loads* (71-103). John Wiley and Sons Ltd.

Zienkiewicz, O. C., Chang, C.T., Bettess, P. (1980). Drained, undrained, consolidating and dynamic behavior assumptions in soils. *Geotechnique*, 30(4), 233-239.

Zienkiewicz, O. C., Bettess, P. (1982). Soils and other saturated media under transient, dynamic conditions; general formulation and the validity of various simplifying assumptions. In G.N. Pande, O.C. Zienkiewicz, (Ed). *Soil Mechanics-Transient and Cyclic Loads* (1-16). John Wiley and Sons Ltd.

APPENDIX – A

LITERATURE SURVEY FOR SOIL PLASTICITY MODELS FOR DYNAMIC SAND BEHAVIOR

A.1 Literature Survey for Soil Plasticity Models for Dynamic Sand Behavior

This part of the thesis is devoted to the brief literature survey of elasto-plastic soil constitutive models giving special attention to the ones for liquefaction and cyclic mobility. The development of elasto-plastic soil constitutive models has been possible within the frameworks of the three great theories, which are classical plasticity theory, generalised plasticity theory, and critical state theory. Therefore, it is possible to group elasto-plastic models as follows:

1. Elasto-plastic soil constitutive models within the framework of Classical Plasticity
2. Elasto-plastic soil constitutive models within the framework of Generalised Plasticity
3. Elasto-plastic soil constitutive models within the framework of Critical State Theory

A.1.1 Classical Plasticity Theory

A.1.1.1 Brief Definition of Plastic Material

Plasticity may be defined as the presence of irrecoverable strains on load removal. Figure 5.1 shows the uniaxial behaviour of a material for (a) non-linear elastic and plastic, (b) ideal plastic and (c) strain hardening material behaviors. In order to understand if a material is nonlinear elastic or nonlinear plastic, it is necessary to look at the unloading portion of the material behaviour which means non-linear behaviour on loading alone does not determine whether the material behaves elastically or plastically. As shown in three of the materials in Figure 1, unloading portion determines the plastic behaviour of the material. If a material is nonlinear elastic, unloading portion follows the same path as the loading portion. On the other hand, if a material is nonlinear plastic, only some part of the strains are recovered and the unloading portion follows a history-dependent different path.

A.1.1.2 Definition of Yield and the Yield Surface

Yield is among the basic concepts in plastic nonlinear behaviour. The stress at which the strains are indeterminate is called the yield stress. For all the stresses below this limit, linear or nonlinear elasticity relationship is assumed. Yield stress is clearly observed in

Figure A1. (b) for an ideal plastic material. For strain hardening / softening plastic behaviour shown in Figure A1. (c), yield stress is determined by a *parameter κ (it may be plastic strain ε_p)* . Yield stress as defined in this paragraph is for uniaxial state of stress. For multiaxial states of stress, yielding depends on a surface which is defined by the yield criterion (A1) which is also verified by experimental studies. Yielding can only occur when the stresses satisfy the general yield criterion. In the interior of the yield surface, there is no plastic deformation. An example for a yield surface is shown in Figure A2.

$$F(\sigma,\kappa) = 0 \qquad \text{(A1)}$$

Here ;

$F(\sigma,\kappa)$: yield criterion

κ : hardening parameter (material state parameter)

Position, size and shape of the yield surface depends on the material state parameter. In other words, the shifts of the yield surface depend on the material state parameter (or parameters).

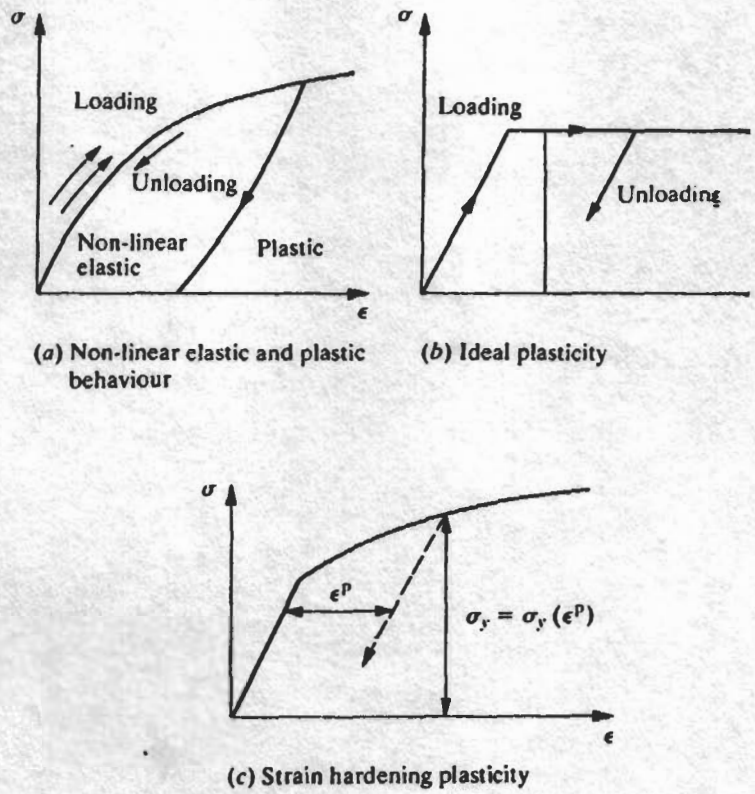


Figure A1. Uniaxial Behaviour of Real Materials (Zienkiewicz and Taylor,1991)

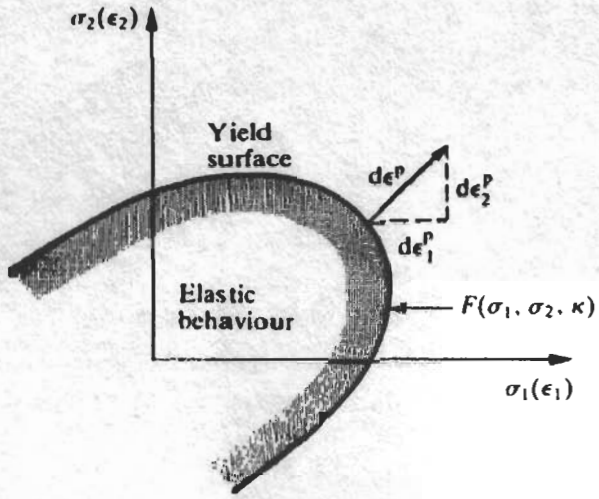


Figure A2. Yield Surface and Normality Criterion in Two-Dimensional Stress Space (Zienkiewicz and Taylor,1991)

A.1.1.3 Definition of Flow Rule and the Potential Surface

The next essential concept of plastic behaviour is the *flow rule* (normality principle). In plasticity, a rule is needed for defining the plastic strain increments. Von Mises first suggested that this behavior is related with the yield surface and this hypothesis was verified by the studies of other researchers working in the area. Hence, the relation between the plastic strain increments and the yield criterion is shown in Eqn.A2 and for the n dimensional stress space Eqn.A3 can be written.

$$d\varepsilon^p = d\lambda \frac{\partial F}{\partial \sigma} \qquad \text{(A2)}$$

$$d\varepsilon_n^p = d\lambda \frac{\partial F}{\partial \sigma_n} \qquad \text{(A3)}$$

Here;

$d\varepsilon^p$: plastic strain increment

$d\lambda$: proportionality constant (yet undetermined)

The restriction of this rule is that it requires the plastic strain increment vector to be normal to the yield surface. This restriction can be removed by using another function in Eqn.(A2) called *potential function (Q)* instead of the yield function as shown in Eqn.(A4) and Eqn.(A5).

$$Q = Q(\sigma, \kappa) \tag{A4}$$

$$d\varepsilon^p = d\lambda \frac{\partial Q}{\partial \sigma} \tag{A5}$$

If Q=F this case is called associated plasticity.

If Q≠F this case is called non-associated plasticity.

Associative and non-associative concepts reveal that the soil behaviour predictions by the Classical Plasticity theory present a sharp transition from the elastic to elastoplastic regime, with a discontinuity in the derivative of stress-strain curves.

A.1.1.4 Yield and Failure Surfaces

In the preceding paragraphs, it is stated that when the stress state reaches the yield surface (stated by eqn. (A1)), irreversible (plastic strains) strains occur, for all the stress states below this limit, the material is elastic. If the stress state has reached the yield surface and in addition, if the material state parameter κ is constant, failure occurs. This behaviour (material state parameter being constant) means that there is only one failure surface which is at the same time equal to the yield surface. However, in case of materials with hardening/softening, yield and failure surfaces can be different .So the difference between the yield surface inside which the material behaviour is elastic and the failure surface on which the failure occurs should be made clear. This behaviour will be explained in the following paragraphs.

A.1.1.5 Concepts of Hardening, Softening and Failure

Figure A3. shows a clay specimen being loaded from an initial state P1 to failure at P3. Yield surfaces for changing values of the material state parameter are given in the form f-(κ)=0. These yield surfaces are expressed in the space of deviatoric invariant and mean (hydrostatic) invariant. The change of the state parameter which determines the shape ,size and the position of the yield surfaces should be represented by some kind of law

(hardening law). In the literature, state parameter increment is usually related with plastic strain increment Eqn.(A6) or plastic work increment Eqn.(A7).

$$d\kappa = d\varepsilon^{p}_{volumetric} \tag{A6}$$

$$d\kappa = \sigma_1 d\varepsilon_1^p + \sigma_2 d\varepsilon_2^p + ... = \sigma^T d\varepsilon^p \tag{A7}$$

In Figure A.3 (a) P1,P2 and P3 shows the stages of loading and that for each stress state, the size of the yield surface gets bigger due to the increase of the hardening parameter (or state parameter). This means the elastic domain gets bigger and hence the material is harder. However, the corresponding stress-strain curve (Figure A.3(b)) shows that the slopes of the stress-strain curve corresponding to P1,P2 and P3 is in contradiction with the material getting harder. The explanation for this contradiction is, the incremental response of the material is harder for Figure A.3 (a).

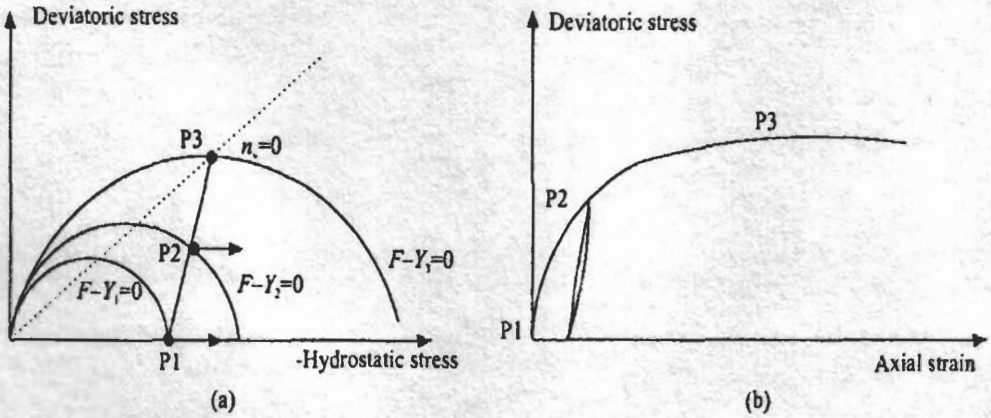


Figure A.3. Typical Hardening Behaviour of Clay a) Yield surfaces b) Stress-Strain Curve Showing Permanent Strain upon Unloading (Zienkiewicz,et.al.,1999)

Figure A.4 represents the material behaviour in which the yield and failure surfaces are the same, hence there is only one surface for both the yield and the failure. In addition, Figure A.5 shows the case for softening material behaviour in which the yield surfaces gets smaller.

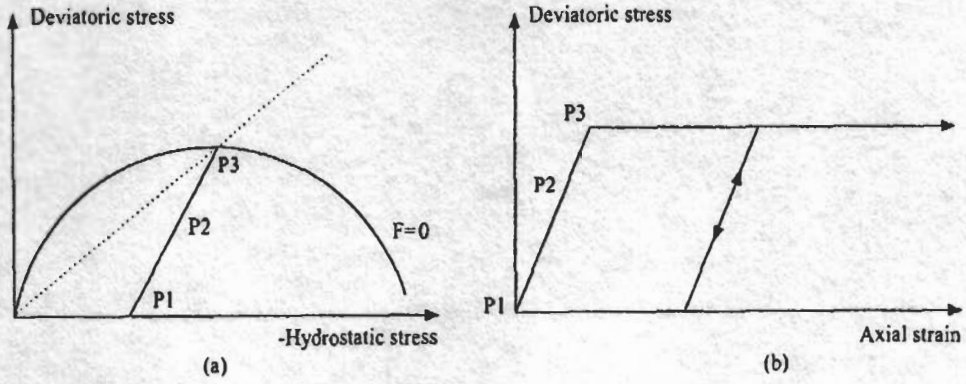


Figure A.4. Ideal Plasticity(κ=constant) a) Stress path b) Stress-Strain Curve (Zienkiewicz,et.al.,1999)

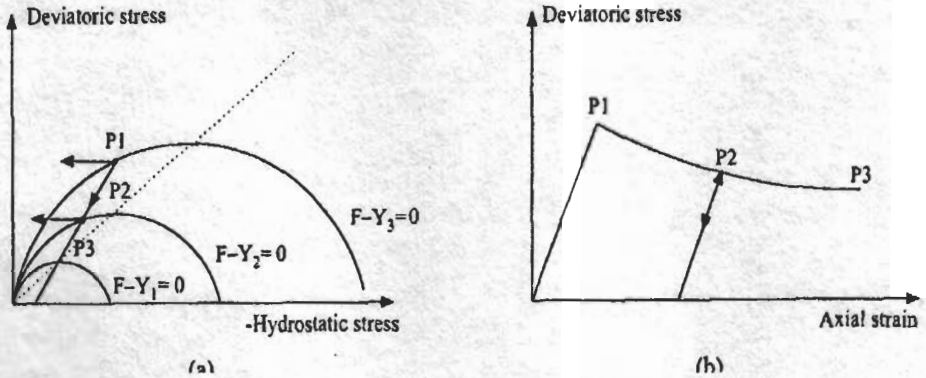


Figure A.5. Softening Behaviour a) Stress Path b) Stress-Strain Curve (Zienkiewicz, et.al.,1999)

A.1.1.6 Total, Elastic and Plastic Strain Increments

In plasticity theory, a strain increment occuring due to a stress increment is always composed two parts as elastic strain increment part and the plastic strain increment part.

$$d\varepsilon = d\varepsilon^e + d\varepsilon^p \qquad \text{(A8)}$$

Replacing Eqn.(A5) in Eqn.(A8), Eqn.(A9) is obtained.

$$d\varepsilon = D^{-1}d\sigma + \frac{\partial Q}{\partial \sigma}d\lambda \qquad \text{(A9)}$$

Here;

$d\varepsilon$:total strain increment

$d\varepsilon^e$: elastic strain increment

$d\varepsilon^p$: plastic strain increment

D : elastic stress-strain matrix

A.1.1.7 Definition for 'Loading' and 'Unloading'

Loading and unloading rules in the framework of classical plasticity theory can be stated as follows:

Plastic loading : If the elastic stress increment ($d\sigma^e$=D$d\varepsilon$) puts the stress outside the yield surface (In this case plastic straining will occur).

Elastic loading : If the new stress is still inside the yield surface due to elastic stress increment ($d\sigma e$=D$d\varepsilon$) or if the new stress is in the unloading direction. (In these cases, no plastic straining will occur.)

A.1.1.8 Incremental Stress-Strain Relationship

When plastic loading occurs, the stress state in on the yield surface and the derivative of the yield criterion will be as shown in Eqn.(A10).

$$dF = \frac{\partial F}{\partial \sigma_1} d\sigma_1 + \frac{\partial F}{\partial \sigma_2} d\sigma_2 + ... + \frac{\partial F}{\partial \kappa} d\kappa = 0 \qquad \text{(A10)}$$

After a set of calculations, the incremental stress –strain relation will be given as in eqn.(A11) and Eqn.(A12).

$$d\sigma = D^*_{ep} d\varepsilon \qquad \text{(A11)}$$

$$D^*_{ep} = D - D\left\{\frac{\partial Q}{\partial \sigma}\right\}\left\{\frac{\partial F}{\partial \sigma}\right\}^T D\left[A + \left\{\frac{\partial F}{\partial \sigma}\right\}^T D\left\{\frac{\partial Q}{\partial \sigma}\right\}\right]^{-1} \qquad \text{(A12)}$$

(Zienkiewicz and Taylor,1991; Zienkiewicz et al.,1999)

A.1.2 Generalised Plasticity Theory

Earlier constitutive models have been derived by using the classical plasticity theory (Work of Cambridge Group,(1960), Roscoe et al., (1958), Schofield and Wroth, (1968)). Later on, theory of generalised plasticity was introduced (Mroz and Zienkiewicz,1984). In fact, modelling with the classical plasticity theory is a special case of generalised plasticity theory. With generalised plasticity theory, it is possible to define the plastic laws without having to define yield or flow surfaces which is very suitable for the construction of plastic laws for liquefaction and cyclic mobility phenomena. The classical plasticity theory allows accurate modelling of soil behaviour under monotonic loading conditions, however, it fails to represent the soil behaviour when the stress path is within the yield surface such as under repeated loading and unloading. For example, the key factor for simulating the liquefaction and cyclic mobility phenomena close to reality lies in the fact that plastic deformation should be introduced for such paths (stres paths within the yield surface) Generalised plasticity was very successful in the solution of such complex soil dynamics problems because the concept of yield surface is not explicitly used and which allows the direct definition of parameters (Zienkiewicz, Chan, Pastor, 1988).

A.1.2.1 Basic Formulations

In all the soil constitutive model theories, the essential criteria is to formulate the plastic behaviour characterised by the irreversibility of stress paths (permanent strain changes). In generalised plasticity theory, the following relations formulate this phenomena:

$$d\sigma = D^{*} d\varepsilon \qquad \text{(A13)}$$

Here;

D^{*} : matrix depending on stress σ, state parameter κ and the direction of the applied stress or strain increment ($d\sigma$ or $d\varepsilon$). The direction of applied stress or strain can be expressed with the concepts 'loading' and 'unloading'. If the direction of 'loading' is expressed in the stress space with the vector n and considering the state parameters as shown in Figure A.7, the plastic loading and unloading can be described as:

$$n^T d\sigma > 0 \quad \text{for loading} \tag{A14}$$

$$n^T d\sigma < 0 \quad \text{for unloading}$$

$$n^T d\sigma = 0 \quad \text{neutral (only elastic straining ocurs)}$$

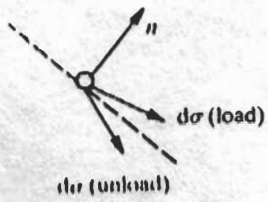


Figure A.7 Loading and Unloading Directions in Stress Space (Zienkiewicz and Taylor,1991)

Thus, incremental stress-strain relations for loading and unloading can be written as:

$$d\sigma = D_L^* d\varepsilon \quad \text{for loading} \tag{A15}$$

$$d\sigma = D_U^* d\varepsilon \quad \text{for unloading}$$

D_U^* and D_L^* : tangent matrices for unloading and loading respectively. (depend only on the state parameter described by σ and κ).

D_U^* and D_L^* should be such matrices that in the neutral direction of the stress increment $d\sigma$, the corresponding strain increments should be equal which gives way to:

$$d\varepsilon = D_L^{*-1} d\sigma = D_u^{*-1} d\sigma$$

$$D_L^{*-1} \equiv D^{-1} + \frac{n_{gL} n^T}{H_L} \tag{A16}$$

$$D_U^{*-1} \equiv D^{-1} + \frac{n_{gU} n^T}{H_U}$$

$$\frac{n^T d\sigma}{H_L} \equiv d\lambda$$

$$d\varepsilon = D_L^{*-1} d\sigma = D^{-1} d\sigma + \frac{n_{gL} n^T}{H_L} d\sigma = D^{-1} d\sigma + n_{gL} d\lambda \tag{A17}$$

Where,

n_{gL} and n_{gU} : arbitrary unit stres vector for loading and unloading

H_L, H_U : appropriate plastic moduli depending on σ and κ.

(Zienkiewicz and Taylor,1991)

A.1.2.2 Multi-mechanism Approach in the Framework of Generalised Plasticity

In the framework of Generalised Plasticity, it is sometimes necessary to adopt an approach called 'multimechanism approach'. For instance, when the effects of principal stres axes rotation are to be taken into account multimechanism model become the best choice. The philosophy of the multimechanism models depends on the fact that the deformation of the material can be considered as the result of the deformations produced by M separate mechanisms, all of them subjected to the same state of stress (Zienkiewicz et al., 1988).

APPENDIX – B
DYNAMIC NONLINEAR FINITE ELEMENT CODE SUBROUTINES

```
clc
clear all
tic
%                              INPUT DATA PART
global_variables
input_data_new_pile
z=1
%-------------------------------------------------------------------------
%                           PRE-PROCESSING PART

% 1. Assign the time,displacement,velocity and acceleration values in the
%    data file to the related variables
assign_data_file
% 2. Create variables related with nodal point coordinate data
%    2.1. Obtain angles
obtain_angles
%    2.2. Obtain number of boundaries (KK)
obtain_KK
%    2.3. Obtain nodal point coordinate data
nodal_coordinate_data
%    2.4. Draw undeformed mesh
undeformed_mesh
%    2.5 Number the nodes
number_nodes
% 3. Create Data Process Arrays
%    3.1. ID Array
ID_array
%    3.2. IEND Array
IEND_array
%    3.3. LMD Array
LMD_array
% 4. Assign the coordinates of each element
assign_coord_to_element
% 5. Boundary Condition Coding
BC_coding
% 6. Gauss Point Data
gauss_points
% 7. Preparation for Condensation Solution
prep_for_condensation
% 8. Preperation for tension part
prep_for_tension
% 9. Space Allocation
allocate_space_disp_vel_accel
allocate_space_for_the_material_model
%-------------------------------------------------------------------------
%                              PROCESSING PART

%CONSTRUCTION OF ELEMENT MATRICES AND ASSEMBLY
calculate_initial_values_for_the_material_model
               %evaluate_constitutive_matrix_beginning
               evaluate_mass_matrix % yüzeydeki noktalar için kontrol et

               evaluate_stiffness_matrix
               evaluate_damping_matrix

evaluate_constant_force_vector %Assembly'de force vector yok olacek
```

```
par1=1-beta1/beta2;
par2=(1-beta1)*delta_t-beta1/beta2*(0.5-beta2)*delta_t;
par3=beta1/(beta2*delta_t);
par4=1/(beta2*delta_t*delta_t);
par5=1/(beta2*delta_t);
par6=(0.5-beta2)/beta2;


displacement_un(:,z+1)=displacement_un(:,z)+displacement_un_inc;
accel_un(:,z+1)=par4*displacement_un_inc-par5*velocity_un(:,z)-par6*accel_un(:,z);
velocity_un(:,z+1)=velocity_un(:,z)+((1-beta1)*accel_un(:,z)+beta1*accel_un(:,z+1))↙
*delta_t;
   %velocity_un(:,z+1)=par1*velocity_un(:,z)+par3*displacement_un_inc+par2*accel_un(:,↙
z);
%-------------------------------------------------------------------------

for te=1:k7
displacement(unknown(te,1),z+1)=displacement_un(te,z+1);
velocity(unknown(te,1),z+1)=velocity_un(te,z+1);
accel(unknown(te,1),z+1)=accel_un(te,z+1);
displacement_inc(unknown(te,1),z)=displacement_un_inc(te,1);
end

for te=1:k3
displacement(known(te,1),z+1)=displacement_kn(te,z+1);
velocity(known(te,1),z+1)=velocity_kn(te,z+1);
accel(known(te,1),z+1)=accel_kn(te,z+1);
displacement_inc(known(te,1),z)=displacement_kn_inc(te,1);
end
%-------------------------------------------------------------------------
% Stresses and strains at the end of each time step
%for el=1:num_of_elem
%stress{el,z+1}=stress_k{el,z+1}(:,5);
%strain{el,z+1}=strain_k{el,z+1}(:,5);
%end
```

```
if z==1
XX_draw=XX;
YY_draw=YY;
end
if z>1
XX_draw=ux';
YY_draw=uy';
end

hold on
t=1;
for j=1:2:nodes*NDOFD
    if j==nodes*NDOFD
    end
    ux(t,1)=XX_draw(1,t)+displacement_inc(j,z);
    t=t+1;
end
t=1;
for j=2:2:nodes*NDOFD
    if j==nodes*NDOFD
    end
    uy(t,1)=YY_draw(1,t)+displacement_inc(j,z);
    t=t+1;
end

%hold on
%plot(ux,uy,'r');
%hold on

 %   for tt=1:360/angle

  %      plot(ux(KK_no{tt}(:,1),1),uy(KK_no{tt}(:,1),1),'r');
   %     hold on
   %   end
    %
%hold on
%-------------------------------------------------------------------------
% B Matrix Change
%XX=ux';
%YY=uy';
%assign_coord_to_element
%-------------------------------------------------------------------------
```

```
%if z==1
%XX_draw=XX;
%YY_draw=YY;
%end
%if z>1
%XX_draw=ux';
%YY_draw=uy';
%end

hold on
%t=1;
%for j=1:2:nodes*NDOFD
%    if j==nodes*NDOFD
%    end
%    ux(t,1)=XX_draw(1,t)+displacement_inc(j,z);
%    t=t+1;
%end
%t=1;
%for j=2:2:nodes*NDOFD
%    if j==nodes*NDOFD
%    end
%    uy(t,1)=YY_draw(1,t)+displacement_inc(j,z);
%    t=t+1;
%end

hold on
tt=1:(360/angle):nodes;
[tt1 tt2]=size(tt);
for i=1:tt2
    if i==tt2
    plot(ux(tt(i):nodes,1),uy(tt(i):nodes,1),'r');
    hold on
    ad1=[ux(nodes,1);ux(tt(i),1)];
    ad2=[uy(nodes,1);uy(tt(i),1)];
    plot(ad1,ad2,'r');
    hold on

        break
    end
    plot(ux(tt(i):tt(i+1)-1,1),uy(tt(i):tt(i+1)-1,1),'r');
    hold on
    ad1=[ux(tt(i+1)-1,1);ux(tt(i),1)];
    ad2=[uy(tt(i+1)-1,1);uy(tt(i),1)];
    plot(ad1,ad2,'r');
    hold on
end

%plot(ux,uy,'r');
hold on
    for tt=1:360/angle

        plot(ux(KK_no{tt}(:,1),1),uy(KK_no{tt}(:,1),1),'r');
        hold on
    end
    %
hold on
```

```
d_TSW(el,z)=((sigma_x(el,z+1)-sigma_y(el,z+1))/2)*(d_ex(el,z)-d_ey(el,z))+to_xy(el,z+1)
*d_gama_xy(el,z);
%d_TSW(el,z)=((sigma_x(el,z)-sigma_y(el,z))/2)*(d_ex(el,z)-d_ey(el,z))+to_xy(el,z)
*d_gama_xy(el,z);

if sign(d_TSW(el,z))==-1
    d_TSW(el,z)=abs(d_TSW(el,z));
end
%
% CALCULATE tau(i) and d_tau(i)
%---------------------------------
tau(el,z+1)=(to_xy(el,z+1)^2+((sigma_x(el,z+1)-sigma_y(el,z+1))/2)^2)^(1/2);
%tau(el,z+1)=(to_xy(el,z)^2+((sigma_x(el,z)-sigma_y(el,z))/2)^2)^(1/2);

d_tau(el,z)=tau(el,z+1)-tau(el,z);
%
% Deviatoric Stress Ratio(r)

DSR(el,z+1)=tau(el,z+1)/(-sigma_m(el,1));

% CALCULATE d_ESW(i), d_PSW(i)
%---------------------------------
d_ESW(el,z)=abs(tau(el,z+1)*(d_tau(el,z)/Gm_star(el,z)));

  m_1=sin(fi_f*pi/180);
  m_2=sin(fi_p*pi/180);
  m_3=0.67*m_2;
d_PSW(el,z)=d_TSW(el,z)-c_1*d_ESW(el,z);

if DSR(el,z+1)/S(el,z)>m_3
  RR=(m_1-DSR(el,z+1)/S(el,z))/(m_1-m_3);
  d_PSW(el,z)=RR*(d_TSW(el,z)-c_1*d_ESW(el,z));
end
if DSR(el,z+1)/0.4>m_3
  RR=(m_1-DSR(el,z+1)/0.4)/(m_1-m_3);
  d_PSW(el,z)=RR*(d_TSW(el,z)-c_1*d_ESW(el,z));
end

        if sign(d_PSW(el,z))==-1
           d_PSW(el,z)=0;
        end

  PSW(el,z+1)=PSW(el,z)+d_PSW(el,z);
  ESW(el,z+1)=ESW(el,z)+d_ESW(el,z);
  TSW(el,z+1)=TSW(el,z)+d_TSW(el,z);
%
% CALCULATE FN
%-----------------------------
FN(el,1)=To_m(el,1)*ref_m(el,1);
%
% CALCULATE w(i)
%-----------------------------
w(el,z+1)=PSW(el,z+1)/FN(el,1);
%
% CALCULATE S0(i)
```

```
%------------------------------
% Liquefaction Front Parameter (S0)
if w(el,z+1)<w_1
    S0(el,z+1)=1-0.6*(w(el,z+1)/w_1)^p_1;
end
if w(el,z+1)>w_1
    S0(el,z+1)=(0.4-S_1)*(w_1/w(el,z+1))^p_2+S_1;
end
if w(el,z+1)==w_1
    S0(el,z+1)=0.4;
end

% Effective Mean Stress Ratio (S)
m_1=sin(fi_f*pi/180);
m_2=sin(fi_p*pi/180);
m_3=0.67*m_2;
r_2(el,z+1)=m_2*S0(el,z+1);
r_3(el,z+1)=m_3*S0(el,z+1);
S_2(el,z+1)=S0(el,z+1)-(r_2(el,z+1)-r_3(el,z+1))/m_1;
%
if DSR(el,z+1)<r_3(el,z+1)
    S(el,z+1)=S0(el,z+1);

end
if DSR(el,z+1)>r_3(el,z+1)
    S(el,z+1)=S_2(el,z+1)+((S0(el,z+1)-S_2(el,z+1))^2+((DSR(el,z+1)-r_3(el,z+1))/m_1)↙
^2)^0.5;
end
```

```
if S0(el,z+1)>0.4
    To_m(el,z+1)=To_m(el,1)*S(el,z+1);
    G_m(el,z+1)=To_m(el,z+1)/ref_m(el,1);
    ref_m(el,z+1)=ref_m(el,1);
end
if S0(el,z+1)<0.4
    delta_r_m(el,z+1)=(m_1-m_2)*(0.4-S0(el,z+1));
    delta_to_m(el,z+1)=delta_r_m(el,z+1)*(-sigma_m(el,1));

    To_m(el,z+1)=To_m(el,1)*S(el,z+1)+delta_to_m(el,z+1);
    ref_m(el,z+1)=ref_m(el,1)/(S0(el,z+1)/0.4);
    G_m(el,z+1)=To_m(el,z+1)/ref_m(el,z+1);
end
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
To_max_f(el,z+1)=To_m(el,z+1)/sum(delta_tetha*sin(tetha));
ref_f(el,z+1)=(To_max_f(el,z+1)/G_m(el,z+1))*sum(delta_tetha*(sin(tetha).*sin(tetha)));
%-------------------------------------------------------------------------
Gm_star(el,z+1)=G_m(el,1)*(sigma_m(el,1)*S(el,z+1)/sigma_m(el,1))^0.5;
%Gm_star(el,z+1)=G_m(el,1)*(sigma_m(el,z+1)/sigma_m(el,1))^0.5;


% Rebound Modulus
KG(el,z+1)=K_a*(sigma_m(el,1)*S(el,z+1)/sigma_va)^0.5;
%KG(el,z+1)=K_a*(sigma_m(el,z+1)/sigma_va)^0.5;
```

```
par1=1-beta1/beta2;
par2=(1-beta1)*delta_t-beta1/beta2*(0.5-beta2)*delta_t;
par3=beta1/(beta2*delta_t);
par4=1/(beta2*delta_t*delta_t);
par5=1/(beta2*delta_t);
par6=(0.5-beta2)/beta2;
%
%displacement_unk=zeros(k7,1);
%displacement_un_inck=zeros(k7,1);

%[L,U]=lu(effective_stiffness);

%u(:,1)=U\(L\effective_force_vector);

%u(:,1)=cgs(effective_stiffness,effective_force_vector);
%u(:,1)=pinv(effective_stiffness)*effective_force_vector;
%-------------------------------------------------------------------------
%displacement_un_inc=pinv(effective_stiffness_aa)*(effective_force_vector_aa);
%displacement_un_inc=cgs(effective_stiffness_aa,effective_force_vector_aa);
[L,U]=lu(effective_stiffness_aa);

displacement_un_inck=U\(L\effective_force_vector_aa);


%%%% cgs birsürü artık
%%%% bırakıyor.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
par1=1-beta1/beta2;
par2=(1-beta1)*delta_t-beta1/beta2*(0.5-beta2)*delta_t;
par3=beta1/(beta2*delta_t);
par4=1/(beta2*delta_t*delta_t);
par5=1/(beta2*delta_t);
par6=(0.5-beta2)/beta2;
%-------------------------------------------------------------------------
% Displacement, velocity, acceleration values for the k th. iteration and
% the total displacement increment
% VERSION 1
            %displacement_unk=displacement_unk_1+displacement_un_inck;
            %velocity_unk=par1*velocity_un(:,z)+par2*accel_un(:,z)-par3*displacement_un
(:,z)+par3*displacement_unk_1+par3*displacement_un_inck;
            %accel_unk=par4*displacement_unk_1+par4*displacement_un_inck-
par4*displacement_un(:,z)-par5*velocity_un(:,z)-par6*accel_un(:,z);
            %
 % VERSION 2
            displacement_unk=displacement_unk_1+displacement_un_inck;
            velocity_unk=velocity_unk_1+par3*displacement_un_inck;
            accel_unk=accel_unk_1+par4*displacement_un_inck;

            displacement_un_inc=displacement_un_inc+displacement_un_inck;

            RA=zeros(k7,1);

RB=GM_MATRIX_KNOWN_UNKNOWN*accel_unk+GM_MATRIX_KNOWN*accel_knk+GDAMP_KNOWN_UNKNOWN*velo
city_unk+GDAMP_KNOWN*velocity_knk+GK_MATRIX_KNOWN_UNKNOWN*displacement_unk+GK_MATRIX_KN
OWN*displacement_knk;

%-------------------------------------------------------------------------

                for te=1:k7
                displacement_k(unknown(te,1),z+1)=displacement_unk(te,1);
                velocity_k(unknown(te,1),z+1)=velocity_unk(te,1);
                accel_k(unknown(te,1),z+1)=accel_unk(te,1);
                end

                for te=1:k3
                displacement_k(known(te,1),z+1)=displacement_knk(te,1);
                velocity_k(known(te,1),z+1)=velocity_knk(te,1);
                accel_k(known(te,1),z+1)=accel_knk(te,1);
                end
                %"""""""""""""""""""""""""""""""""""""""
                for te=1:k7
                displacement_k_1(unknown(te,1),z+1)=displacement_unk_1(te,1);
                velocity_k_1(unknown(te,1),z+1)=velocity_unk_1(te,1);
                accel_k_1(unknown(te,1),z+1)=accel_unk_1(te,1);
                end

                for te=1:k3
                displacement_k_1(known(te,1),z+1)=displacement_knk_1(te,1);
                velocity_k_1(known(te,1),z+1)=velocity_knk_1(te,1);
                accel_k_1(known(te,1),z+1)=accel_knk_1(te,1);
                end
                %"""""""""""""""""""""""""""""""""""""""
                for te=1:k7
```

```
                displacement_inck(unknown(te,1),1)=displacement_un_inck(te,1);
                end

                for te=1:k3
                displacement_inck(known(te,1),1)=displacement_kn_inck(te,1);
                end
                %"""""""""""""""""""""""""""""""""""
                for te=1:k7
                REACTION(unknown(te,1),1)=RA(te,1);
                end

                for te=1:k3
                REACTION(known(te,1),1)=RB(te,1);
                end

if iter==1

    gk=GK_MATRIX;
    gm=GM_MATRIX;
    gd=GDAMP;
end
%-------------------------------------------------------------------------
            %evaluate_stresses_at_each_Gaussian_point_update_D
            %evaluate_stresses_at_each_Gaussian_point_update_D1
%-------------------------------------------------------------------------
            % criterias
            % 1.displacement
            criteria_d=norm(displacement_un_inck,2)/norm(displacement_unk,2)
            %criteria_d=norm(displacement_inck,2)/norm(displacement_k,2)

            % 2.force
            %ust1=GM_MATRIX*accel_k+GDAMP*velocity_k+GK_MATRIX*displacement_k;
            %ust2=GM_MATRIX*accel_k_1+GDAMP*velocity_k_1+GK_MATRIX*displacement_k_1;
            %ust3=gm*accel(:,z)+gd*velocity(:,z)+gk*displacement(:,z);
            %criteria_f=norm(ust1-ust2)/norm(ust1-ust3);
            residual
            % 3.energy
            %ust=(displacement_un_inck)'*(-GK_MATRIX_KNOWN_UNKNOWN*displacement_unk_1-
GDAMP_KNOWN_UNKNOWN*velocity_unk_1-GM_MATRIX_KNOWN_UNKNOWN*accel_unk_1);
            %alt=(aa)'*(-gk*displacement_un(:,z)-gm*accel_un(:,z)-gd*velocity_un(:,z));
            %criteria_e=ust/alt;
%-------------------------------------------------------------------------
%For the next iteration
            displacement_unk_1=displacement_unk;
            velocity_unk_1=velocity_unk;
            accel_unk_1=accel_unk;
```

```
for te=1:k7
displacement_inc(unknown(te,1),z)=displacement_un_inc(te,1);
end

for te=1:k3
displacement_inc(known(te,1),z)=displacement_kn_inc(te,1);
end

u=displacement_inc(:,z);

            for el=1:num_of_elem
                u_el=[u(LMDO(1,1,el),1);u(LMDO(2,1,el),1);u(LMDO(1,2,el),1);u(LMDO(2,2,el),1);u(LMDO(1,3,el),1);u(LMDO(2,3,el),1);u(LMDO(1,4,el),1);u(LMDO(2,4,el),1)];
                ab=strain_k{el,z};
                ba=stress_k{el,z};
                for g=1:IGAUSD
                   disp_shape_func_derv
                   disp_jaco_c_detjaco
                   b_matrix_disp
                   %
                   d_strain_k{el,z}(:,g)=B*u_el;
                   strain_k{el,z+1}(:,g)=ab(:,g)+B*u_el;

                  % stress_k{el,z+1}(:,g)=ba(:,g)+DD{el,z}*B*u_el;
                end

                % At the middle of the element
                d_ex(el,z)=d_strain_k{el,z}(1,5);
                d_ey(el,z)=d_strain_k{el,z}(2,5);
                d_gama_xy(el,z)=d_strain_k{el,z}(3,5);
                %
                ey(el,z+1)=ey(el,z)+d_ey(el,z);
                ex(el,z+1)=ex(el,z)+d_ex(el,z);
                gama_xy(el,z+1)=gama_xy(el,z)+d_gama_xy(el,z);



             for im=1:I_m
             d_ksi_store(el,im,z)=(d_ex(el,z)-d_ey(el,z))*cos(tetha(im))+d_gama_xy(el,z)*sin(tetha(im));
             ksi_store(el,im,z+1)=(ex(el,z+1)-ey(el,z+1))*cos(tetha(im))+gama_xy(el,z+1)*sin(tetha(im));
             %ksi_store(el,im,z+1)=ksi_store(el,im,z)+d_ksi_store(el,im,z);

             ksi_store_norm(el,im,1:z+1)=ksi_store(el,im,1:z+1)/ref_f(el,z);
             d_ksi_store_norm(el,im,1:z)=d_ksi_store(el,im,1:z)/ref_f(el,z);

             end



         end
```

```
for el=1:num_of_elem


                  for im=1:I_m

                      % initial loading
                        if (disp(z+1,1)>disp(z,1))&&initial_finish(el,im)==0;
                           initial_loading
                           continue
                        end

                      % unloading from initial loading
                        if portion(el,im)==1&&(disp(z+1,1)<disp(z,1))&&initial_finish
(el,im)==0;
                           unload_from_initial_begin
                           continue
                        end
                        %
                        if portion(el,im)==2&&(disp(z+1,1)<disp(z,1))&&initial_finish
(el,im)==1&&relunl_finish(el,im)==0;
                           unload_from_initial_continue
                           continue
                        end

                        %reloading
                        if (disp(z+1,1)>disp(z,1))&&initial_finish(el,im)
==1&&relunl_finish(el,im)==0;
                            reloading_begin
                            continue
                        end
                        if (disp(z+1,1)>disp(z,1))&&initial_finish(el,im)
==1&&relunl_finish(el,im)==1;
                           reloading_continue
                           continue
                        end

                        %reunloading
                        if portion(el,im)>2&&(disp(z+1,1)<disp(z,1))&&initial_finish
(el,im)==1&&relunl_finish(el,im)==1;
                           reunloading_begin
                           continue
                        end
                        if portion(el,im)>2&&(disp(z+1,1)<disp(z,1))&&initial_finish
(el,im)==1&&relunl_finish(el,im)==0;
                           reunloading_continue
                           continue
                        end




                  end

%
          update_the_new_stress_system4;
```

```
            s_s0;
            due_to_S1
            update_actual_shear;

            stress{el,z+1}=[sigma_x(el,z+1);sigma_y(el,z+1);to_xy(el,z+1)];
            strain{el,z+1}=[ex(el,z+1);ey(el,z+1);gama_xy(el,z+1)];

end
```

```
portion(el,im)=1;

nu(el,im,z+1)=((ksi_store(el,im,z+1)/ref_f(el,z))/(1+abs(ksi_store(el,im,z+1)/ref_f
(el,z))))*To_max_f(el,z);
R(el,im,z+1)=(1/(1+abs(ksi_store(el,im,z+1)/ref_f(el,z)))^2)*(To_max_f(el,z)/ref_f(el,
z))*delta_tetha;
                                %Calculate Norm
nu_norm(el,im,1:z+1)=ksi_store_norm(el,im,1:z+1)./(1+abs(ksi_store_norm(el,im,1:
z+1)));
max_ksi_norm(el,im)=ksi_store_norm(el,im,z+1);
initial_sign(el,im)=sign(d_ksi_store_norm(el,im,z));
```

```
portion(el,im)=portion(el,im)+1;
initial_finish(el,im)=1;
turning_point(el,im,portion(el,im)-1)=z;
%
reversal_initial_ksi(el,im)=ksi_store(el,im,z);
reversal_initial_nu(el,im)=nu(el,im,z);
%
ref_p(el,im,portion(el,im)-1)=ref_f(el,z);
To_max_f_p(el,im,portion(el,im)-1)=To_max_f(el,z);
%
reversal_initial_ksi_norm(el,im)=ksi_store_norm(el,im,z);
reversal_initial_nu_norm(el,im)=nu_norm(el,im,z);

ayla1=ksi_store(el,:,z);
ayla2=nu(el,:,z);
ayla3=ksi_store_norm(el,:,z);


ksih(el,im)=find_ksih(gama_xy(el,z),ref_m(el,z),ayla3,ayla1,ayla2);

[a_back(el,im),b_back(el,im)]=find_a_b_parameters(im,ayla3);

                                % Calculate Normal Ksi and Nu
          A1=(ksi_store(el,im,z+1)/ref_f(el,z)-reversal_initial_ksi_norm(el,im))*
*b_back(el,im)/a_back(el,im);
          B1=1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-reversal_initial_ksi_norm(el,im))*
/(2*a_back(el,im)));
          nu(el,im,z+1)=(A1/B1+reversal_initial_nu_norm(el,im))*To_max_f(el,z);
                                % Calculate Normalised Ksi,Nu
          A11=(ksi_store_norm(el,im,z+1)-reversal_initial_ksi_norm(el,im))*b_back(el,*
im)/a_back(el,im);
          B11=1+abs((ksi_store_norm(el,im,z+1)-reversal_initial_ksi_norm(el,im))/*
(2*a_back(el,im)));
          nu_norm(el,im,z+1)=A11/B11+reversal_initial_nu_norm(el,im);
                                % Calculate Tangent Shear Modulus
          C1=1/((1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-reversal_initial_ksi_norm(el,*
im))/(2*a_back(el,im))))^2);
          R(el,im,z+1)=C1*(To_max_f(el,z)/ref_f(el,z))*delta_tetha*(b_back(el,im)*
/a_back(el,im));
```

```
                                A1=(ksi_store(el,im,z+1)/ref_f(el,z)-↙
reversal_initial_ksi_norm(el,im))*b_back(el,im)/a_back(el,im);
                                B1=1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-↙
reversal_initial_ksi_norm(el,im))/(2*a_back(el,im)));
                                nu(el,im,z+1)=(A1/B1+reversal_initial_nu_norm(el,im))↙
*To_max_f(el,z);

                                A11=(ksi_store_norm(el,im,z+1)-reversal_initial_ksi_norm↙
(el,im))*b_back(el,im)/a_back(el,im);
                                B11=1+abs((ksi_store_norm(el,im,z+1)-↙
reversal_initial_ksi_norm(el,im))/(2*a_back(el,im)));
                                nu_norm(el,im,z+1)=A11/B11+reversal_initial_nu_norm(el,↙
im);

                                C1=1/((1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-↙
reversal_initial_ksi_norm(el,im))/(2*a_back(el,im))))^2);
                                R(el,im,z+1)=C1*(To_max_f(el,z)/ref_f(el,z))*delta_tetha*↙
(b_back(el,im)/a_back(el,im));
```

```
  portion(el,im)=portion(el,im)+1;
  %
  relunl_finish(el,im)=1;
  turning_point(el,im,portion(el,im)-1)=z;
  %
  reversal_ksi(el,im,portion(el,im)-1)=ksi_store(el,im,z);
  reversal_nu(el,im,portion(el,im)-1)=nu(el,im,z);
  %
  ref_p(el,im,portion(el,im)-1)=ref_f(el,z);
  To_max_f_p(el,im,portion(el,im)-1)=To_max_f(el,z);
  %
  reversal_ksi_norm(el,im)=ksi_store_norm(el,im,z);
  reversal_nu_norm(el,im)=nu_norm(el,im,z);

  cback1=1/2*abs((reversal_initial_ksi_norm(el,im)-reversal_ksi_norm(el,im))/a_back(el,↙
im));
  cback2=(reversal_initial_nu_norm(el,im)-reversal_nu_norm(el,im))/b_back(el,im);
  cback3=(reversal_initial_ksi_norm(el,im)-reversal_ksi_norm(el,im))/a_back(el,im)-↙
(reversal_initial_nu_norm(el,im)-reversal_nu_norm(el,im))/b_back(el,im);
  c_back(el,im)=cback1*cback2/cback3;

  A1=(ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm(el,im))*b_back(el,im)/a_back↙
(el,im);
  B1=1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm(el,im))/(2*a_back(el,im)↙
*c_back(el,im)));
  nu(el,im,z+1)=(A1/B1+reversal_nu_norm(el,im))*To_max_f(el,z);

  A11=(ksi_store_norm(el,im,z+1)-reversal_ksi_norm(el,im))*b_back(el,im)/a_back(el,im);
  B11=1+abs((ksi_store_norm(el,im,z+1)-reversal_ksi_norm(el,im))/(2*a_back(el,im)↙
*c_back(el,im)));
  nu_norm(el,im,z+1)=A11/B11+reversal_nu_norm(el,im);

  C1=1/((1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm(el,im))/(2*a_back↙
(el,im)*c_back(el,im))))^2);
  R(el,im,z+1)=C1*(To_max_f(el,z)/ref_f(el,z))*delta_tetha*(b_back(el,im)/a_back(el,↙
im));
```

```
                                A1=(ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm↙
(el,im))*b_back(el,im)/a_back(el,im);
                                B1=1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-↙
reversal_ksi_norm(el,im))/(2*a_back(el,im)*c_back(el,im)));
                                nu(el,im,z+1)=(A1/B1+reversal_nu_norm(el,im))*To_max_f↙
(el,z);

                                A11=(ksi_store_norm(el,im,z+1)-reversal_ksi_norm(el,im))↙
*b_back(el,im)/a_back(el,im);
                                B11=1+abs((ksi_store_norm(el,im,z+1)-reversal_ksi_norm↙
(el,im))/(2*a_back(el,im)*c_back(el,im)));
                                nu_norm(el,im,z+1)=A11/B11+reversal_nu_norm(el,im);

                                C1=1/((1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-↙
reversal_ksi_norm(el,im))/(2*a_back(el,im)*c_back(el,im))))^2);
                                R(el,im,z+1)=C1*(To_max_f(el,z)/ref_f(el,z))*delta_tetha*↙
(b_back(el,im)/a_back(el,im));
```

```
 portion(el,im)=portion(el,im)+1;
 relunl_finish(el,im)=0;
 turning_point(el,im,portion(el,im)-1)=z;


 reversal_ksi(el,im,portion(el,im)-1)=ksi_store(el,im,z);
 reversal_nu(el,im,portion(el,im)-1)=nu(el,im,z);
 %
 ref_p(el,im,portion(el,im)-1)=ref_f(el,z);
 To_max_f_p(el,im,portion(el,im)-1)=To_max_f(el,z);

 reversal_ksi_norm(el,im)=ksi_store_norm(el,im,z);
 reversal_nu_norm(el,im)=nu_norm(el,im,z);

 cback1=1/2*abs((-reversal_initial_ksi_norm(el,im)-reversal_ksi_norm(el,im))/a_back(el,↙
im));
 cback2=(-reversal_initial_nu_norm(el,im)-reversal_nu_norm(el,im))/b_back(el,im);
 cback3=(-reversal_initial_ksi_norm(el,im)-reversal_ksi_norm(el,im))/a_back(el,im)-(-↙
reversal_initial_nu_norm(el,im)-reversal_nu_norm(el,im))/b_back(el,im);
 c_back(el,im)=cback1*cback2/cback3;


 A1=(ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm(el,im))*b_back(el,im)/a_back↙
(el,im);
 B1=1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm(el,im))/(2*a_back(el,im)↙
*c_back(el,im)));
 nu(el,im,z+1)=(A1/B1+reversal_nu_norm(el,im))*To_max_f(el,z);

 A11=(ksi_store_norm(el,im,z+1)-reversal_ksi_norm(el,im))*b_back(el,im)/a_back(el,im);
 B11=1+abs((ksi_store_norm(el,im,z+1)-reversal_ksi_norm(el,im))/(2*a_back(el,im)*c_back↙
(el,im)));
 nu_norm(el,im,z+1)=A11/B11+reversal_nu_norm(el,im);

 C1=1/((1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm(el,im))/(2*a_back(el,↙
im)*c_back(el,im))))^2);
 R(el,im,z+1)=C1*(To_max_f(el,z)/ref_f(el,z))*delta_tetha*(b_back(el,im)/a_back(el,↙
im));
```

```
                                A1=(ksi_store(el,im,z+1)/ref_f(el,z)-reversal_ksi_norm
(el,im))*b_back(el,im)/a_back(el,im);
                                B1=1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-
reversal_ksi_norm(el,im))/(2*a_back(el,im)*c_back(el,im)));
                                nu(el,im,z+1)=(A1/B1+reversal_nu_norm(el,im))*To_max_f
(el,z);

                                A11=(ksi_store_norm(el,im,z+1)-reversal_ksi_norm(el,im))
*b_back(el,im)/a_back(el,im);
                                B11=1+abs((ksi_store_norm(el,im,z+1)-reversal_ksi_norm
(el,im))/(2*a_back(el,im)*c_back(el,im)));
                                nu_norm(el,im,z+1)=A11/B11+reversal_nu_norm(el,im);

                                C1=1/((1+abs((ksi_store(el,im,z+1)/ref_f(el,z)-
reversal_ksi_norm(el,im))/(2*a_back(el,im)*c_back(el,im))))^2);
                                R(el,im,z+1)=C1*(To_max_f(el,z)/ref_f(el,z))*delta_tetha*
(b_back(el,im)/a_back(el,im));
```

```
part11=[1 0 0;0 1 0;0 0 1]-KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1/(2*KG(el,z)))*[1 1
0];
part22=KG(el,z)*[1;1;0]*[1 1 0];
part33=KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1/(2*KG(el,z)))*[1 1 0]*[1;1;0]
*K_f/porosity*[1 1 0];
part44=-KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1+KG(el,z)*porosity/K_f)/KG(el,z)
*K_f/porosity*[1 1 0];
G1=zeros(3,3);
for im=1:I_m
    G1=G1+R(el,im,z)*[cos(tetha(im));-cos(tetha(im));sin(tetha(im))]*[cos(tetha(im)) -
cos(tetha(im)) sin(tetha(im))];
end
part55=G1;
dstress=pinv(part11)*(part22+part33+part44+part55)*[d_ex(el,z);d_ey(el,z);d_gama_xy(el,
z)];

dstress_t=pinv(part1)*(part2+part3+part4+part5)*[d_ex(el,z);d_ey(el,z);d_gama_xy(el,
z)];
%-------------------------------------------------------------------------
%part1=KG(el,z)*[1;1;0]*[1 1 0];
%part2=-KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1/KG(el,z)+porosity/K_f)*K_f/porosity*[1
1 0];
%G1=zeros(3,3);
%for im=1:I_m
 %   G1=G1+R(el,im,z)*[cos(tetha(im));-cos(tetha(im));sin(tetha(im))]*[cos(tetha(im)) -
cos(tetha(im)) sin(tetha(im))];
%end

%part3=G1;
%part4=[1;1;0]*K_f/porosity*[1 1 0];
%dstress=(part1+part2+part3)*[d_ex(el,z);d_ey(el,z);d_gama_xy(el,z)];

%dstress_t=(part1+part2+part3+part4)*[d_ex(el,z);d_ey(el,z);d_gama_xy(el,z)];

% Stress Increment
d_sigma_x(el,z)=dstress(1,1);
d_sigma_y(el,z)=dstress(2,1);
d_to_xy(el,z)=dstress(3,1);
%
d_sigma_xt(el,z)=dstress_t(1,1);
d_sigma_yt(el,z)=dstress_t(2,1);
d_to_xyt(el,z)=dstress_t(3,1);

%
sigma_x(el,z+1)=sigma_x(el,z)+d_sigma_x(el,z);
sigma_y(el,z+1)=sigma_y(el,z)+d_sigma_y(el,z);
to_xy(el,z+1)=to_xy(el,z)+d_to_xy(el,z);

sigma_x_t(el,z+1)=sigma_x_t(el,z)+d_sigma_xt(el,z);
sigma_y_t(el,z+1)=sigma_y_t(el,z)+d_sigma_yt(el,z);
to_xy_t(el,z+1)=to_xy_t(el,z)+d_to_xy(el,z);
```

```
% Jacobien, C Matrix, Determinant of Jacobien
XS=0.0;
XT=0.0;
YS=0.0;
YT=0.0;
%

    for k=1:4
%
XS=XS+DNS(k)*XCORD(el,k);
XT=XT+DNT(k)*XCORD(el,k);
%
YS=YS+DNS(k)*YCORD(el,k);
YT=YT+DNT(k)*YCORD(el,k);
%
end
%
%
% JACOBIEN MATRIX
SJ(1,1)=XS;
SJ(2,1)=XT;
SJ(1,2)=YS;
SJ(2,2)=YT;
%
%
% C MATRIX

C(1,1)=YT;
C(2,1)=-YS;
C(1,2)=-XT;
C(2,2)=XS;
%
%
% DETERMINANT OF JACOBIEN
SD=XS*YT-XT*YS;
if SD==0||SD<0
    pause;
end

%
CNX(1)=C(1,1)*DNS(1)+C(2,1)*DNT(1);
CNX(2)=C(1,1)*DNS(2)+C(2,1)*DNT(2);
CNX(3)=C(1,1)*DNS(3)+C(2,1)*DNT(3);
CNX(4)=C(1,1)*DNS(4)+C(2,1)*DNT(4);

CNY(1)=C(1,2)*DNS(1)+C(2,2)*DNT(1);
CNY(2)=C(1,2)*DNS(2)+C(2,2)*DNT(2);
CNY(3)=C(1,2)*DNS(3)+C(2,2)*DNT(3);
CNY(4)=C(1,2)*DNS(4)+C(2,2)*DNT(4);
```

```
% B Matrix

B=1/SD*[CNX(1) 0 CNX(2) 0 CNX(3) 0 CNX(4) 0;
        0 CNY(1) 0 CNY(2) 0 CNY(3) 0 CNY(4);
        CNY(1) CNX(1) CNY(2) CNX(2) CNY(3) CNX(3) CNY(4) CNX(4)];
```

```
% Shape Function Matrix
NN=[DN(1) 0 DN(2) 0 DN(3) 0 DN(4) 0;0 DN(1) 0 DN(2) 0 DN(3) 0 DN(4)];

EMASS1=ro*SD*transpose(NN)*NN*WEIGHDS(g)*WEIGHDT(g);
for kont=1:8
    EMASS1(kont,:)=(EMASS1(:,kont))';
end
EM_MATRIX=EM_MATRIX+EMASS1;
```

```
%Style 1:
for it=1:ND
    KB1(it)=LMDO(1,it,el);
    KB2(it)=LMDO(2,it,el);

    IC1(it)=NDOFD*(it-1)+1;
    IC2(it)=NDOFD*(it-1)+2;
    end

for is=1:ND
          for ip=1:ND
              Y1(KB1(is),KB1(ip))=Y1(KB1(is),KB1(ip))+YY1(IC1(is),IC1(ip));% tek↙
satırlar tek kolonlara
              Y1(KB1(is),KB2(ip))=Y1(KB1(is),KB2(ip))+YY1(IC1(is),IC2(ip));% tek↙
satırlar cift kolonlara
          end


          for ip=1:ND
              Y1(KB2(is),KB1(ip))=Y1(KB2(is),KB1(ip))+YY1(IC2(is),IC1(ip));% cift↙
satırlar tek kolonlara
              Y1(KB2(is),KB2(ip))=Y1(KB2(is),KB2(ip))+YY1(IC2(is),IC2(ip));% cift↙
satırlar cift kolonlara
          end
     end


%-------------------------------------------------------------------------
% Style 2:
%index(1)=LMDO(1,1,el);
%index(2)=LMDO(2,1,el);
%index(3)=LMDO(1,2,el);
%index(4)=LMDO(2,2,el);
%index(5)=LMDO(1,3,el);
%index(6)=LMDO(2,3,el);
%index(7)=LMDO(1,4,el);
%index(8)=LMDO(2,4,el);

%for ir=1:8
%irs=index(ir);
 %    for ic=1:8
  %       ics=index(ic);
   %      Y1(irs,ics)=Y1(irs,ics)+YY1(ir,ic);
   % end
%end
```

```
%[k5 k6]=size(AS);
%
%kot=[];
%for ip=1:k5
%for is=1:k3
 %  if AS(ip,1)==known(is,1)
  %     kot(ip)=is;
   %    break
 %  end
%end
%end
%for is=1:k5
%    for ip=1:k5
 %        Y2_new(kot(is),kot(ip))=Y2(AS(is),AS(ip))+YY2(AT(is),AT(ip));% tek satırlar↙
tek kolonlara
  %  end
%end
%-------------------------------------------------------------------------
for is=1:k3
    for ip=1:k3
        Y2_new(is,ip)=Y2(known(is,1),known(ip,1));% tek satırlar tek kolonlara
    end
end
```

```
%[k5 k6]=size(AS);
%
%kot=[];
%for ip=1:k5
%for is=1:k7
 %  if AS(ip,1)==unknown(is,1)
  %     kot(ip)=is;
   %    break
%   end
%end
%end

%for is=1:k5
 %   for ip=1:k5
  %       Y3_new(kot(is),kot(ip))=Y3(AS(is),AS(ip))+YY3(AT(is),AT(ip));% tek satırlar↙
tek kolonlara
   % end
%end
%--------------------------------------------------------------------------
for is=1:k7
    for ip=1:k7
        Y3_new(is,ip)=Y3(unknown(is,1),unknown(ip,1));% tek satırlar tek kolonlara
    end
end
```

```
%[k55 k66]=size(AS_un);
%[k77 k88]=size(AS_n);

%
%kot_n=[];
%for ip=1:k77
%for is=1:k3
 %  if AS_n(ip,1)==known(is,1)
  %     kot_n(ip)=is;
   %    break
%   end
%end
%end
%
%kot_un=[];
%for ip=1:k55
%for is=1:k7
 %  if AS_un(ip,1)==unknown(is,1)
  %     kot_un(ip)=is;
   %    break
   %end
%end
%end



%for is=1:k55
 %   for ip=1:k77
  %       Y4_new(kot_un(is),kot_n(ip))=Y4(AS_un(is),AS_n(ip))+YY4(AT_un(is),AT_n(ip));%↙
tek satırlar tek kolonlara
   % end
%end
%-------------------------------------------------------------------------
for is=1:k7
    for ip=1:k3
        Y4_new(is,ip)=Y4(unknown(is,1),known(ip,1));% tek satırlar tek kolonlara
    end
end
```

```
GK_MATRIX=zeros(nodes*NDOFD,nodes*NDOFD);
GK_MATRIX_KNOWN=zeros(k3,k3);
GK_MATRIX_UNKNOWN=zeros(k7,k7);
GK_MATRIX_UNKNOWN_KNOWN=zeros(k7,k3);
GK_MATRIX_KNOWN_UNKNOWN=zeros(k3,k7);

for el=1:num_of_elem


        EK_MATRIX=zeros(8,8);


         for g=1:IGAUSD
             KMU1=zeros(8,8);
             disp_shape_func_derv
             disp_jaco_c_detjaco
             b_matrix_disp

             k_matrix

         end
                 % EK_MATRIX=(tmat{el,1})'*EK_MATRIX*(tmat{el,1});

         %
         Y1=GK_MATRIX;
         YY1=EK_MATRIX;
         assemble_overall1
         GK_MATRIX=Y1;
         clear Y1
         clear YY1
         %



end
% Assemblage for condensation part
         %
         % known part
         known_mass
         Y2=GK_MATRIX;
         Y2_new=GK_MATRIX_KNOWN;

         assemble_overall2
         GK_MATRIX_KNOWN=Y2_new;
         clear Y2
         clear Y2_new

         %
         % umknown part
         umknown_mass
         Y3=GK_MATRIX;
         Y3_new=GK_MATRIX_UNKNOWN;

         assemble_overall3
         GK_MATRIX_UNKNOWN=Y3_new;
         clear Y3
```

```
clear Y3_new

%
% unknown-known part
unknown_known_mass
Y4=GK_MATRIX;
Y4_new=GK_MATRIX_UNKNOWN_KNOWN;

assemble_overall4
GK_MATRIX_UNKNOWN_KNOWN=Y4_new;
clear Y4
clear Y4_new

% known-unknown part
GK_MATRIX_KNOWN_UNKNOWN=(GK_MATRIX_UNKNOWN_KNOWN)';
```

```
%Drained
if drainage==1
KMU1=SD*transpose(B)*DD(el,z)*B*WEIGHDS(g)*WEIGHDT(g);    %
for kont=1:8
    KMU1(kont,:)=(KMU1(:,kont))';
end

EK_MATRIX=EK_MATRIX+KMU1;     %
end
%Undrained
%1
if drainage==2
part1=[1 0 0;0 1 0;0 0 1]-KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1/(2*KG(el,z)))*[1 1
0];
part2=KG(el,z)*[1;1;0]*[1 1 0];
part3=-KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1+KG(el,z)*porosity/K_f)/KG(el,z)
*K_f/porosity*[1 1 0];
G1=zeros(3,3);
for im=1:I_m
    G1=G1+R(el,im,z)*[cos(tetha(im));-cos(tetha(im));sin(tetha(im))]*[cos(tetha(im)) -
cos(tetha(im)) sin(tetha(im))];
end

part4=G1;
part5=[1;1;0]*K_f/porosity*[1 1 0];
DDU=pinv(part1)*(part2+part3+part4+part5);
%-------------------------------------------------------------------------
%2.
%part1=KG(el,z)*[1;1;0]*[1 1 0];
%part2=-KG(el,z)*[1;1;0]*[1 1 0]*[0.5;0.5;0]*(1/KG(el,z)+porosity/K_f)*K_f/porosity*[1
1 0];
%G1=zeros(3,3);
%for im=1:I_m
 %   G1=G1+R(el,im,z)*[cos(tetha(im));-cos(tetha(im));sin(tetha(im))]*[cos(tetha(im)) -
cos(tetha(im)) sin(tetha(im))];
%end

%part3=G1;
%part4=[1;1;0]*K_f/porosity*[1 1 0];
%DDU=(part1+part2+part3+part4);




KMU1=SD*transpose(B)*DDU*B*WEIGHDS(g)*WEIGHDT(g);    %
for kont=1:8
    KMU1(kont,:)=(KMU1(:,kont))';
end
EK_MATRIX=EK_MATRIX+KMU1;     %
end
```

```
% displacement
displacement_un(:,1)=displacement(unknown,1);
displacement_kn(:,1)=displacement(known,1);
% velocity

%
%b1=2*daire_no{1}(1:(360/angle)/4+1,1)-1;
%b2=2*daire_no{1}(((360/angle)/4)*3+1:(360/angle),1)-1;
%bak1=[b1;b2];
%
b1=2*daire_no{1}(1:360/angle,1)-1;
bak1=b1;
%


velocity(bak1,1)=vel(1,1);
velocity_un(:,1)=velocity(unknown,1);
velocity_kn(:,1)=velocity(known,1);

for te=1:k7
velocity(unknown(te,1),1)=velocity_un(te,1);
end
for te=1:k3
velocity(known(te,1),1)=velocity_kn(te,1);
end


% acceleration
%accel(:,1)=(GM_MATRIX)\(-GDAMP*velocity(:,1)-GK_MATRIX*displacement(:,1));
%accel_un(:,1)=accel(unknown,1);
%accel_kn(:,1)=accel(known,1);
%      ------------------------------------------------------------------
[L,U]=lu(GM_MATRIX_UNKNOWN);

accel_un(:,1)=U\(L\(-GK_MATRIX_UNKNOWN*displacement_un(:,1)-↙
GK_MATRIX_UNKNOWN_KNOWN*displacement_kn(:,1)-GDAMP_UNKNOWN*velocity_un(:,1)-↙
GDAMP_UNKNOWN_KNOWN*velocity_kn(:,1)-GM_MATRIX_UNKNOWN_KNOWN*accel_kn(:,1)));



%accel_un(:,1)=pinv(GM_MATRIX_UNKNOWN)*(-GK_MATRIX_UNKNOWN*displacement_un(:,1)-↙
GK_MATRIX_UNKNOWN_KNOWN*displacement_kn(:,1)-GDAMP_UNKNOWN*velocity_un(:,1)-↙
GDAMP_UNKNOWN_KNOWN*velocity_kn(:,1)-GM_MATRIX_UNKNOWN_KNOWN*accel_kn(:,1));
for te=1:k7
accel(unknown(te,1),1)=accel_un(te,1);
end
for te=1:k3
accel(known(te,1),1)=accel_kn(te,1);
end

%accel(:,1)=zeros(NDOFD*nodes,1);
```

```
beta1=3;
beta2=3.0625;


%beta1=2;
%beta2=1.5625;

%beta1=1.5;
%beta2=0.8;


%beta1=0.7;
%beta2=0.36;

%beta1=0.8;
%beta2=0.4225;

%beta1=1;
%beta2=0.5625;


%beta1=0.5;
%beta2=0.25;
delta_t=time(2,1)-time(1,1);

% Newmark değiştir
% Mesh küçült
% delta_t crit küçült
% B matrisi
```

```
%b1=2*daire_no{1}(1:(360/angle)/4+1,1)-1;
%b2=2*daire_no{1}(((360/angle)/4)*3+1:(360/angle),1)-1;
%bak1=[b1;b2];
%
b1=2*daire_no{1}(1:360/angle,1)-1;
bak1=b1;


%b1=2*daire_no{1}(1:7,1)-1;
%b2=2*daire_no{1}(19:24,1)-1;
%bak1=[b1;b2];
%
%b1=2*daire_no{1}(1,1)-1;
%bak1=b1;



            displacement(bak1,z+1)=disp(z+1,1);
            displacement_un(:,z+1)=displacement(unknown,z+1);
            displacement_kn(:,z+1)=displacement(known,z+1);



%

            velocity(bak1,z+1)=vel(z+1,1);
            velocity_un(:,z+1)=velocity(unknown,z+1);
            velocity_kn(:,z+1)=velocity(known,z+1);
%

            accel(bak1,z+1)=acc(z+1,1);
            accel_un(:,z+1)=accel(unknown,z+1);
            accel_kn(:,z+1)=accel(known,z+1);

%-------------------------------------------------------------------------
```

```
criteria_d=0.5;
criteria_e=0.5;
criteria_f=0.5;
iter=0;
% k_1
displacement_unk_1=displacement_un(:,z);
velocity_unk_1=velocity_un(:,z);
accel_unk_1=accel_un(:,z);
displacement_knk_1=displacement_kn(:,z);
velocity_knk_1=velocity_kn(:,z);
accel_knk_1=accel_kn(:,z);
% k
displacement_knk=displacement_kn(:,z+1);
velocity_knk=velocity_kn(:,z+1);
accel_knk=accel_kn(:,z+1);
displacement_unk=zeros(k7,1);
velocity_unk=zeros(k7,1);
accel_unk=zeros(k7,1);
% inc
displacement_un_inc=zeros(k7,1);
displacement_kn_inc=displacement_kn(:,z+1)-displacement_kn(:,z);
% inck
displacement_un_inck=zeros(k7,1);
displacement_kn_inck=displacement_kn_inc;
%
```

```
%evaluate_constitutive_matrix

evaluate_stiffness_matrix
evaluate_damping_matrix
evaluate_effective_stiffness
evaluate_effective_force
boundary_conditions_and_solution33k;
```

```
par1=1-beta1/beta2;
par2=(1-beta1)*delta_t-beta1/beta2*(0.5-beta2)*delta_t;
par3=beta1/(beta2*delta_t);
par4=1/(beta2*delta_t*delta_t);
par5=1/(beta2*delta_t);
par6=(0.5-beta2)/beta2;


% VERSION 1
%effective_stiffness_aa=[];
%effective_stiffness_ab=[];
%effective_stiffness_ba=[];
%effective_stiffness_bb=[];
%
%effective_stiffness_aa=par4*GM_MATRIX_UNKNOWN+par3*GDAMP_UNKNOWN+GK_MATRIX_UNKNOWN;
%↙
effective_stiffness_ab=par4*GM_MATRIX_UNKNOWN_KNOWN+par3*GDAMP_UNKNOWN_KNOWN+GK_MATRIX_↙
UNKNOWN_KNOWN;
%effective_stiffness_ba=transpose(effective_stiffness_ab);
%effective_stiffness_bb=par4*GM_MATRIX_KNOWN+par3*GDAMP_KNOWN+GK_MATRIX_KNOWN;

% VERSION 2

effective_stiffness_aa=[];
effective_stiffness_ab=[];
effective_stiffness_ba=[];
effective_stiffness_bb=[];

effective_stiffness_aa=par4*GM_MATRIX_UNKNOWN+par3*GDAMP_UNKNOWN+GK_MATRIX_UNKNOWN;
```

```
par1=1-beta1/beta2;
par2=(1-beta1)*delta_t-beta1/beta2*(0.5-beta2)*delta_t;
par3=beta1/(beta2*delta_t);
par4=1/(beta2*delta_t*delta_t);
par5=1/(beta2*delta_t);
par6=(0.5-beta2)/beta2;


% VERSION 1
%effective_force_vector_aa=[];
%effective_force_vector_bb=[];
%
%c1=-GM_MATRIX_UNKNOWN_KNOWN*accel_kn(:,z+1)-GDAMP_UNKNOWN_KNOWN*velocity_kn(:,z+1)-↙
GK_MATRIX_UNKNOWN_KNOWN*displacement_kn(:,z+1);
%c2=(par4*GK_MATRIX_UNKNOWN+par3*GDAMP_UNKNOWN)*displacement_un(:,z);
%c3=(par5*GM_MATRIX_UNKNOWN-par1*GDAMP_UNKNOWN)*velocity_un(:,z);
%c4=(par6*GM_MATRIX_UNKNOWN-par2*GDAMP_UNKNOWN)*accel_un(:,z);
%c5=-(par4*GM_MATRIX_UNKNOWN+par3*GDAMP_UNKNOWN+GK_MATRIX_UNKNOWN)*displacement_unk_1;
%effective_force_vector_aa=c1+c2+c3+c4+c5;

% VERSION 2
effective_force_vector_aa=[];
effective_force_vector_bb=[];
c1=-GM_MATRIX_UNKNOWN*accel_unk_1-GDAMP_UNKNOWN*velocity_unk_1-↙
GK_MATRIX_UNKNOWN*displacement_unk_1;
c2=-GM_MATRIX_UNKNOWN_KNOWN*accel_kn(:,z+1)-GDAMP_UNKNOWN_KNOWN*velocity_kn(:,z+1)-↙
GK_MATRIX_UNKNOWN_KNOWN*displacement_kn(:,z+1);

effective_force_vector_aa=c1+c2;
```

```
if IGAUSD==4

%                          2 x 2 Gauss Quadrature


 % GAUSS POINTS FOR DISPLACEMENT
% ____________________________
%FOR INTEGRATION POINT 1
SGD(1)=-0.577350269189626;
TGD(1)=-0.577350269189626;
WEIGHDS(1)=1.0;
WEIGHDT(1)=1.0;
%FOR INTEGRATION POINT 2
SGD(2)=0.577350269189626;
TGD(2)=-0.577350269189626;
WEIGHDS(2)=1.0;
WEIGHDT(2)=1.0;
%FOR INTEGRATION POINT 3
SGD(3)=0.577350269189626;
TGD(3)=0.577350269189626;
WEIGHDS(3)=1.0;
WEIGHDT(3)=1.0;
%FOR INTEGRATION POINT 4
SGD(4)=-0.577350269189626;
TGD(4)=0.577350269189626;
WEIGHDS(4)=1.0;
WEIGHDT(4)=1.0;
%Weight ler 1 olduğu için eleman matrislerinde yazılmamıştır.
% PLAXIS ten veya CHANDRAKANT'tanveya LOGAN'daN YAZ

end
%-------------------------------------------------------------------------

if IGAUSD==9
%                          3 x 3 Gauss Quadrature


 % GAUSS POINTS FOR DISPLACEMENT
% ____________________________
%FOR INTEGRATION POINT 1
SGD(1)=-0.774596669241483;
TGD(1)=-0.774596669241483;
WEIGHDS(1)=0.555555555555556;
WEIGHDT(1)=0.555555555555556;
%FOR INTEGRATION POINT 2
SGD(2)=0;
TGD(2)=-0.774596669241483;
WEIGHDS(2)=0.888888888888888;
WEIGHDT(2)=0.555555555555556;
%FOR INTEGRATION POINT 3
SGD(3)=0.774596669241483;
TGD(3)=-0.774596669241483;
WEIGHDS(3)=0.555555555555556;
WEIGHDT(3)=0.555555555555556;
%FOR INTEGRATION POINT 4
SGD(4)=-0.774596669241483;
```

```
TGD(4)=0;
WEIGHDS(4)=0.555555555555556;
WEIGHDT(4)=0.888888888888888;
%FOR INTEGRATION POINT 5
SGD(5)=0;
TGD(5)=0;
WEIGHDS(5)=0.888888888888888;
WEIGHDT(5)=0.888888888888888;
%FOR INTEGRATION POINT 6
SGD(6)=0.774596669241483;
TGD(6)=0;
WEIGHDS(6)=0.555555555555556;
WEIGHDT(6)=0.888888888888888;
%FOR INTEGRATION POINT 7
SGD(7)=-0.774596669241483;
TGD(7)=0.774596669241483;
WEIGHDS(7)=0.555555555555556;
WEIGHDT(7)=0.555555555555556;
%FOR INTEGRATION POINT 8
SGD(8)=0;
TGD(8)=0.774596669241483;
WEIGHDS(8)=0.888888888888888;
WEIGHDT(8)=0.555555555555556;
%FOR INTEGRATION POINT 9
SGD(9)=0.774596669241483;
TGD(9)=0.774596669241483;
WEIGHDS(9)=0.555555555555556;
WEIGHDT(9)=0.555555555555556;
end

% No Singularity
% Numerical Integration'dan kaynaklanabilecek Singularity kontrol
% edildi.Hughes'da.
% Nonlinear analizde higher order numerical integration kullanılmalı....
```

```
% preparation for known part
ak=0;
[k1 k2]=size(BC_node_xy);
for ii=1:k1
    if (BC_node_xy(ii,2)==1)||(BC_node_xy(ii,2)==-1)
        ak=ak+1;
        known(ak,1)=BC_node_xy(ii,1)*2-1;
    end
    if (BC_node_xy(ii,3)==1)||(BC_node_xy(ii,3)==-1)
        ak=ak+1;
        known(ak,1)=BC_node_xy(ii,1)*2;
    end
end

[k3 k4]=size(known);

% preparation for unknown part
ak=0;
sonlu=0;
for ii=1:nodes*2
    sonlu=0;
    for ik=1:k3
        if ii==known(ik,1)
            sonlu=1;
            break

        end
    end
    if sonlu==0
        ak=ak+1;
        unknown(ak,1)=ii;
    end
end

[k7 k8]=size(unknown);
```

```
DD=cell(num_of_elem,1);
DD(:,:)={zeros(3,3)};
%
stress=cell(num_of_elem,f1);
stress(:,:)={zeros(3,1)};
%
strain=cell(num_of_elem,f1);
strain(:,:)={zeros(3,1)};
%
stress_k=cell(num_of_elem,f1);
stress_k(:,:)={zeros(3,IGAUSD)};
%
strain_k=cell(num_of_elem,f1);
strain_k(:,:)={zeros(3,IGAUSD)};
%
d_strain_k=cell(num_of_elem,f1);
d_strain_k(:,:)={zeros(3,IGAUSD)};

%-------------------------------------------------------------------------

displacement=zeros(nodes*NDOFD,f1);
velocity=zeros(nodes*NDOFD,f1);
accel=zeros(nodes*NDOFD,f1);
%
displacement_un=zeros(k7,f1);
displacement_kn=zeros(k3,f1);
%
velocity_un=zeros(k7,f1);
velocity_kn=zeros(k3,f1);
%
accel_un=zeros(k7,f1);
accel_kn=zeros(k3,f1);
%-------------------------------------------------------------------------
displacement_k=zeros(nodes*NDOFD,f1);
velocity_k=zeros(nodes*NDOFD,f1);
accel_k=zeros(nodes*NDOFD,f1);

displacement_unk=zeros(k7,1);
displacement_knk=zeros(k3,1);
%
velocity_unk=zeros(k7,1);
velocity_knk=zeros(k3,1);
%
accel_unk=zeros(k7,1);
accel_knk=zeros(k3,1);
%-------------------------------------------------------------------------

displacement_k_1=zeros(nodes*NDOFD,f1);
velocity_k_1=zeros(nodes*NDOFD,f1);
accel_k_1=zeros(nodes*NDOFD,f1);
%
displacement_unk_1=zeros(k7,1);
displacement_knk_1=zeros(k3,1);
%
velocity_unk_1=zeros(k7,1);
velocity_knk_1=zeros(k3,1);
```

```
%
accel_unk_1=zeros(k7,1);
accel_knk_1=zeros(k3,1);
%--------------------------------------------------------------------------

displacement_inc=zeros(nodes*NDOFD,f1);
displacement_un_inc=zeros(k7,1);
displacement_kn_inc=zeros(k3,1);
%--------------------------------------------------------------------------

displacement_inck=zeros(nodes*NDOFD,1);
displacement_un_inck=zeros(k7,1);
displacement_kn_inck=zeros(k3,1);
%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
stress_check=zeros((round(KK)-1)*bb1,1);
%--------------------------------------------------------------------------
ux=zeros(nodes,1);
uy=zeros(nodes,1);
```

```
sigma_v0=zeros(num_of_elem,1);
sigma_x=zeros(num_of_elem,fl);
sigma_y=zeros(num_of_elem,fl);
sigma_m=zeros(num_of_elem,fl);
to_xy=zeros(num_of_elem,fl);

sigma_x_t=zeros(num_of_elem,fl);
sigma_y_t=zeros(num_of_elem,fl);
to_xy_t=zeros(num_of_elem,fl);
sigma_m_t=zeros(num_of_elem,fl);

pore=zeros(num_of_elem,fl);

%
ex=zeros(num_of_elem,fl);
ey=zeros(num_of_elem,fl);
gama_xy=zeros(num_of_elem,fl);
%
G_m=zeros(num_of_elem,fl);
To_m=zeros(num_of_elem,fl);
ref_m=zeros(num_of_elem,fl);
%
To_max_f=zeros(num_of_elem,fl);
ref_f=zeros(num_of_elem,fl);
%
e_e=zeros(num_of_elem,fl);
S=zeros(num_of_elem,fl);
S0=zeros(num_of_elem,fl);
e_p=zeros(num_of_elem,fl);
%
PSW=zeros(num_of_elem,fl);
ESW=zeros(num_of_elem,fl);
TSW=zeros(num_of_elem,fl);
%
Gm_star=zeros(num_of_elem,fl);
tau=zeros(num_of_elem,fl);
%
R=zeros(num_of_elem,I_m,fl);
KG=zeros(num_of_elem,fl);
%
DD=cell(num_of_elem,fl);
DD(:,:)={zeros(3,3)};
%
ksi_store=zeros(num_of_elem,I_m,fl);
nu=zeros(num_of_elem,I_m,fl);
d_ksi_store=zeros(num_of_elem,I_m,fl);
%
d_ex=zeros(num_of_elem,fl);
d_ey=zeros(num_of_elem,fl);
d_gama_xy=zeros(num_of_elem,fl);
d_ee=zeros(num_of_elem,fl);
d_ep=zeros(num_of_elem,fl);
%
d_sigma_x=zeros(num_of_elem,fl);
d_sigma_y=zeros(num_of_elem,fl);
d_to_xy=zeros(num_of_elem,fl);
```

```
d_pore=zeros(num_of_elem,f1);
d_porel=zeros(num_of_elem,f1);
d_sigma_m=zeros(num_of_elem,f1);
%
d_sigma_xt=zeros(num_of_elem,f1);
d_sigma_yt=zeros(num_of_elem,f1);
d_to_xyt=zeros(num_of_elem,f1);
d_sigma_mt=zeros(num_of_elem,f1);

%
initial_finish=zeros(num_of_elem,I_m);
relunl_finish=zeros(num_of_elem,I_m);
initial_sign=zeros(num_of_elem,I_m);
portion=zeros(num_of_elem,I_m);
ksih=zeros(num_of_elem,1);
a_back=zeros(num_of_elem,I_m);
b_back=zeros(num_of_elem,I_m);
c_back=zeros(num_of_elem,I_m);
gol=zeros(num_of_elem,1);
```

```
for el=1:num_of_elem
    sigma_v0(el,1)=-(sat_gama-9.81)*depth;
    %
    sigma_x(el,1)=K*sigma_v0(el,1);
    sigma_y(el,1)=K*sigma_v0(el,1);
    sigma_m(el,1)=(sigma_x(el,1)+sigma_y(el,1))/2;
    to_xy(el,1)=0.0;

    sigma_x_t(el,1)=K*sigma_v0(el,1);
    sigma_y_t(el,1)=K*sigma_v0(el,1);
    to_xy_t(el,1)=0.0;
    sigma_m_t(el,1)=(sigma_x_t(el,1)+sigma_y_t(el,1))/2;
    %-----------------------------------------------------------------
    gama_xy(el,1)=0.0;
    ex(el,1)=0.0;
    ey(el,1)=0.0;
    %-----------------------------------------------------------------
    G_m(el,1)=G_ma*(sigma_m(el,1)/sigma_va)^0.5;
    To_m(el,1)=-sigma_m(el,1)*sin(fi_f*pi/180);
    ref_m(el,1)=To_m(el,1)/G_m(el,1);
    %-----------------------------------------------------------------
    delta_tetha=pi/(I_m);
                %for im=1:I_m;
                %tetha(im)=(im-1)*delta_tetha;
                %end
    im=1:1:I_m;
    tetha(im)=(im-1)*delta_tetha;
    %-----------------------------------------------------------------
    To_max_f(el,1)=To_m(el,1)/sum(delta_tetha*sin(tetha));
    ref_f(el,1)=(To_max_f(el,1)/G_m(el,1))*sum(delta_tetha*(sin(tetha).*sin(tetha)));
    %-----------------------------------------------------------------
    hv=H_m;
    %-----------------------------------------------------------------
    e_e(el,1)=-((sigma_m(el,1)/-BB)^0.5);
    %e_e(el,1)=0;
    %-----------------------------------------------------------------
    S(el,1)=1;
    S0(el,1)=1;
    e_p(el,1)=0;
    %-----------------------------------------------------------------
    PSW(el,1)=0;
    ESW(el,1)=0;
    TSW(el,1)=0;
    %-----------------------------------------------------------------
    Gm_star(el,1)=G_m(el,1)*(sigma_m(el,1)/sigma_m(el,1))^0.5;
    %-----------------------------------------------------------------
    tau(el,1)=(to_xy(el,1)^2+((sigma_x(el,1)-sigma_y(el,1))/2)^2)^(1/2);
    %-----------------------------------------------------------------
    %G1=0;
    %G2=0;
    %G3=0;
    for im=1:I_m
        %R{el}(im,1)=G_m0(el)*delta_tetha;
        R(el,im,1)=(To_max_f(el,1)/ref_f(el,1))*delta_tetha;
      %  G1=G1+(R(el,im,1)*(cos(tetha(im)))^2);
       % G2=G2+(R(el,im,1)*cos(tetha(im))*sin(tetha(im)));
```

```
      %  G3=G3+(R(el,im,1)*(sin(tetha(im)))^2);
    end
    %-------------------------------------------------------------------
    KG(el,1)=K_a*(sigma_m(el,1)/sigma_va)^0.5;
    %-------------------------------------------------------------------
    %DD{el,1}=KG(el,1)*[1 1 0;1 1 0;0 0 0]+G1*[1 -1 0;-1 1 0;0 0 0]+G2*[0 0 1;0 0 -1;1↙
-1 0]+G3*[0 0 0;0 0 0;0 0 1];
    %-------------------------------------------------------------------
end
```

```
GM_MATRIX=zeros(nodes*NDOFD,nodes*NDOFD);
GM_MATRIX_KNOWN=zeros(k3,k3);
GM_MATRIX_UNKNOWN=zeros(k7,k7);
GM_MATRIX_UNKNOWN_KNOWN=zeros(k7,k3);
GM_MATRIX_KNOWN_UNKNOWN=zeros(k3,k7);
%
for el=1:num_of_elem


        EM_MATRIX=zeros(8,8);


         for g=1:IGAUSD
             EMASS1=zeros(8,8);
             disp_shape_func_derv
             disp_jaco_c_detjaco
             b_matrix_disp

             mass_matrix

         end
         %EM_MATRIX=(tmat{el,1})'*EM_MATRIX*(tmat{el,1});
         % Normal assemblage
         Y1=GM_MATRIX;
         YY1=EM_MATRIX;
         assemble_overall1
         GM_MATRIX=Y1;
         clear Y1
         clear YY1
         %


end
% Assemblage for condensation part
         %
         % known part
         known_mass
         Y2=GM_MATRIX;
         Y2_new=GM_MATRIX_KNOWN;

         assemble_overall2
         GM_MATRIX_KNOWN=Y2_new;
         clear Y2
         clear Y2_new

         %
         % unknown part
         unknown_mass
         Y3=GM_MATRIX;
         Y3_new=GM_MATRIX_UNKNOWN;

         assemble_overall3
         GM_MATRIX_UNKNOWN=Y3_new;
         clear Y3
         clear Y3_new
```

```
%
% unknown-known part
unknown_known_mass
Y4=GM_MATRIX;
Y4_new=GM_MATRIX_UNKNOWN_KNOWN;

assemble_overall4
GM_MATRIX_UNKNOWN_KNOWN=Y4_new;
clear Y4
clear Y4_new

% known-unknown part
GM_MATRIX_KNOWN_UNKNOWN=(GM_MATRIX_UNKNOWN_KNOWN)';
```

```
Si=SGD(g);
Ti=TGD(g);
% Displacement Shape Functions

DN(1)=1.0/4.0*(1-Si)*(1-Ti);

DN(2)=1.0/4.0*(1+Si)*(1-Ti);

DN(3)=1.0/4.0*(1+Si)*(1+Ti);

DN(4)=1.0/4.0*(1-Si)*(1+Ti);

% Derivatives of Displacement Shape Functions

DNS(1)=1.0/4.0*(Ti-1);

DNS(2)=1.0/4.0*(1-Ti);

DNS(3)=1.0/4.0*(1+Ti);

DNS(4)=1.0/4.0*(-1-Ti);
%

DNT(1)=1.0/4.0*(Si-1);

DNT(2)=1.0/4.0*(-1-Si);

DNT(3)=1.0/4.0*(1+Si);

DNT(4)=1.0/4.0*(1-Si);
```

```
%
%CALCULATION TAKING INTO ACCOUNT THE CHANGING TIME DEPENDENT DISP. B.C.
initial_acceleration_velocity_displacement
Newmark_parameters
efe=80;
for z=1:efe
          if z==efe
              break
          end

           evaluate_time_dependent_prescribed_disp_vel_acc
           at_the_beginning_of_the_time_step


           while criteria_d>0.0001

               iter=iter+1
            %if iter==2
            %     stop
            % end

               update_system_matrices_for_each_iteration_and_solve
               at_the_end_of_each_iteration
               strains_for_the_material_model
               Iai_model6


           end
           disp_vel_acc_at_the_end_of_time_step
           %deformed_mesh
             help_drawing1_aha
end
 help_drawing1
%-------------------------------------------------------------------------
%                      POST-PROCESSING PART

           %strains_and_stresses
           %support_reactions
           %soil_reactions


%-------------------------------------------------------------------------
toc
```

```
%   Displacement, Velocity Acceleration Data File for Prescribed Nodes
load A005Hz001sn4m.txt;
%-------------------------------------------------------------------------
%                          Loading Conditions
%         1: Drained
%         2: Undrained
drainage=2;
%-------------------------------------------------------------------------
%                           Mesh Control Data
%
%
angle=15;                          %5            9             15
cap=0.3;
reference=40*cap+cap/2;
radius=cap/2;
%
boundary_no=4;
boundary_sub(1)=0.1;               %0.025        0.05          0.1
boundary_sub(2)=0.25;              %0.10         0.25          0.25
boundary_sub(3)=0.5;               %0.25         0.5           0.5
%
boundary(1)=reference+cap/2;
boundary(2)=boundary(1)+0.5;
boundary(3)=boundary(2)+1.0;
boundary(4)=boundary(3)+1.5;
%-------------
%angle=15;           %5
%
%cap=0.0365;   %2;
%reference=40*cap+cap/2;
%radius=cap/2;
%
%boundary_no=5;
%boundary_sub(1)=0.010;
%boundary_sub(2)=0.020;
%boundary_sub(3)=0.050;
%boundary_sub(4)=0.1;

%
%boundary(1)=reference+cap/2;
%boundary(2)=boundary(1)+0.05;
%boundary(3)=boundary(2)+0.10;
%boundary(4)=boundary(3)+0.20;
%boundary(5)=boundary(4)+0.10;


NDOFD=2;
ND=4;
%
%-------------------------------------------------------------------------
%                        Material PRoperties
elasticity_modulus=50000;
porosity=0.40;

sat_gama=21;
gama=20;
```

```
gama_water=9.81;
Dr=0.47;
poisson=0.35;
K=0.5;
%if drainage==1
 %  ro=(gama)/9.81;
%end
%if drainage==2
   ro=(sat_gama-gama_water)/9.81;
   %ro=(gama)/9.81;
%end
depth=4;
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%               Material Properties Related With Iai Model

%                                        Type of Mechanism          Kind of the↙
Parameters
%                                        -------------------↙
----------------------------
    sigma_va=-80;        %KPa(kN/m2)                                 Reference↙
Confining Stress

    K_a=70480;          %KPa(kN/m2)       Elastic Volumetric        Rebound Modulus↙
at sigma_m0=sigma_ma
    BB=(0.5*K_a/((-sigma_va)^0.5))^2;
    %BB=0.25*(K_a^2)/(-sigma_va);
    G_ma=152700;        %KPa(kN/m2)       Elastic Shear             Initial shear↙
modulus at sigma_m0=sigma_ma
    fi_f=35;            %deg              Plastic Shear             Shear↙
Resistance Angle (fully developed friction angle(internal friction angle)
    fi_p=30;            %deg              Plastic Dilatancy         Phase↙
Transformation Angle
    H_m=0.3;            %-                Plastic Shear             Hysteretic↙
Damping Factor at Large Strain Level
%   material parameters which characterise the cyclic mobility of the cohesionless soil
    p_1=0.50;           %-                Plastic Dilatancy         Initial Phase↙
of Dilatancy
    p_2=0.74;           %-                Plastic Dilatancy         Final Phase of↙
Dilatancy
    w_1=4.1;            %-                Plastic Dilatancy         Overall↙
dilatancy
    S_1=0.005;          %-                Plastic Dilatancy         Ultimate Limit↙
of Dilatancy
    c_1=1.0;            %-                Plastic Dilatancy         Threshold Limit↙
(parameter for specifying the threshold level)
    porosity=0.42;      %                 Continuity Condition      Porosity
                        %                 of two phase medium
    K_f=2000000;        % kPa(kN/m2)      Continuity Condition      Elastic Bulk↙
Modulus of Water
                        %                 of two phase medium


% NUMBER OF VIRTUAL SHEAR MECHANISMS
%-----------------------------------------
              % Conditions Related With Mechanisms
```

```
            % I_m=1000; (infinite number of shear mechanisms)
            % I_m=a number<1000 (finite number of mechanisms)

      I_m=12;              % number of virtual shear mechanisms


%-------------------------------------------------------------------------

%                    Number of Gauss Points
IGAUSD=9;
%-------------------------------------------------------------------------
```

```
disp=A005Hz001sn4m(:,2);
vel=A005Hz001sn4m(:,3);
acc=A005Hz001sn4m(:,4);
time=A005Hz001sn4m(:,1);
[f1 f2]=size(disp);
```

```
a(1)=0;
t=0;
for i=2:(360/angle)
    a(i)=a(i-1)+angle;
end
%--------------------------------------------------------------------------
%a=0:angle:360;
```

```
KK=1;  %En sonda ekleyeceğimiz 1'i ilk başta ekleyelim!
for i=2:boundary_no
KK=KK+(boundary(i)-boundary(i-1))/boundary_sub(i-1);
end
%-------------------------------------------------------------------------
bounk(1)=1;
bounk(2)=(((boundary(2)-boundary(1))/boundary_sub(1))+1);
bounk(boundary_no)=KK;
for i=3:boundary_no-1
    bounk(i)=bounk(i-1)+((boundary(i)-boundary(i-1))/boundary_sub(i-1));
end
%
num_of_elem=(round(KK)-1)*(360/angle);
nodes=(360/angle)*round(KK);
%
%[da1 da2]=size(daire_x{1});
%bounds=2*da1;
```

```
t=0;
for k=1:round(KK)
        for i=2:boundary_no
            if k==1
               radius_inc=0;
               radius=radius+radius_inc;
            end

            if k>bounk(i-1)&&k<=bounk(i)
               radius_inc=boundary_sub(i-1);
               radius=radius+radius_inc;
            end

        end


for i=1:(360/angle)
    t=t+1;
    if (a(i)<=90)&&(a(i)>=0)
    y(i,k)=radius*sin(a(i)*pi/180);
    x(i,k)=(radius^2-y(i,k)^2)^0.5;
    %
    y(i,k)=reference+abs(y(i,k));
    x(i,k)=reference+abs(x(i,k));
    end
    if (a(i)<=180)&&(a(i)>90)
    y(i,k)=abs(radius*sin((180-a(i))*pi/180));
    x(i,k)=abs((radius^2-y(i,k)^2)^0.5);
    %
    y(i,k)=reference+y(i,k);
    x(i,k)=reference-x(i,k);
    end
    if (a(i)<=270)&&(a(i)>180)
    x(i,k)=abs(radius*sin((270-a(i))*pi/180));
    y(i,k)=abs((radius^2-x(i,k)^2)^0.5);
    %
    y(i,k)=reference-y(i,k);
    x(i,k)=reference-x(i,k);
    end
    if (a(i)<=360)&&(a(i)>270)
    x(i,k)=abs(radius*sin((a(i)-270)*pi/180));
    y(i,k)=abs((radius^2-x(i,k)^2)^0.5);
    %
    y(i,k)=reference-y(i,k);
    x(i,k)=reference+x(i,k);
    end
    daire_no{k}(i,1)=t;
end
daire_x{k}=x(:,k);
daire_y{k}=y(:,k);

end


    for tt=1:360/angle
        for t=1:round(KK)
```

```
t=0;
for k=1:round(KK)
        for i=2:boundary_no
            if k==1
               radius_inc=0;
               radius=radius+radius_inc;
            end

            if k>bounk(i-1)&&k<=bounk(i)
               radius_inc=boundary_sub(i-1);
               radius=radius+radius_inc;
            end

        end


for i=1:(360/angle)
    t=t+1;
    if (a(i)<=90)&&(a(i)>=0)
    y(i,k)=radius*sin(a(i)*pi/180);
    x(i,k)=(radius^2-y(i,k)^2)^0.5;
    %
    y(i,k)=reference+abs(y(i,k));
    x(i,k)=reference+abs(x(i,k));
    end
    if (a(i)<=180)&&(a(i)>90)
    y(i,k)=abs(radius*sin((180-a(i))*pi/180));
    x(i,k)=abs((radius^2-y(i,k)^2)^0.5);
    %
    y(i,k)=reference+y(i,k);
    x(i,k)=reference-x(i,k);
    end
    if (a(i)<=270)&&(a(i)>180)
    x(i,k)=abs(radius*sin((270-a(i))*pi/180));
    y(i,k)=abs((radius^2-x(i,k)^2)^0.5);
    %
    y(i,k)=reference-y(i,k);
    x(i,k)=reference-x(i,k);
    end
    if (a(i)<=360)&&(a(i)>270)
    x(i,k)=abs(radius*sin((a(i)-270)*pi/180));
    y(i,k)=abs((radius^2-x(i,k)^2)^0.5);
    %
    y(i,k)=reference-y(i,k);
    x(i,k)=reference+x(i,k);
    end
    daire_no{k}(i,1)=t;
end
daire_x{k}=x(:,k);
daire_y{k}=y(:,k);

end


    for tt=1:360/angle
        for t=1:round(KK)
```

```
            KK_no{tt}(t,1)=daire_no{t}(tt,1);
        end
    end
```

```
for i=1:(360/angle)
    %for k=1:(boundary/radius_inc)
        for k=1:round(KK)

    X{i}(k,1)=daire_x{k}(i,1);
    Y{i}(k,1)=daire_y{k}(i,1);
    end
end
%---------------------------------------

%for k=1:(boundary/radius_inc)
    for k=1:round(KK)

plot(x(:,k),y(:,k))
hold on
end
hold on
for k=1:round(KK)    %son çizgiyi çizmek için
    dogx=[x(i,k);x(1,k)];
    dogy=[y(i,k);y(1,k)];
    line(dogx,dogy);
    hold on
end
hold on
%---------------------------------------
for i=1:(360/angle)
    plot(X{i}(:,1),Y{i}(:,1))
end
hold on
for i=1:(360/angle)
    plot(X{i}(:,1),Y{i}(:,1),'.')
end
%---------------------------------------
```

```
for k=1:round(KK)
    for i=1:(360/angle)
        XX((360/angle)*(k-1)+i)=x(i,k);
        YY((360/angle)*(k-1)+i)=y(i,k);
        NODE_NO((360/angle)*(k-1)+i)=daire_no{k}(i,1);
    end
end
```

```
% ID Array
IDDO=zeros(NDOFD,nodes);
%
%  ID Array for displacement elements
%  ____________________________________
NEQD=1;
for k=1:nodes
for i=1:NDOFD
   IDDO(i,k)=NEQD;
   NEQD=NEQD+1;
end
end
NEQD=NEQD-1;
```

```
i=0;
k=1;
elem=1;
while elem<=num_of_elem
i=i+1;
    if i==(360/angle)
    IEND(elem,1)=daire_no{k}(i,1);
    IEND(elem,2)=daire_no{k+1}(i,1);
    IEND(elem,3)=daire_no{k+1}(1,1);
    IEND(elem,4)=daire_no{k}(1,1);

        k=k+1;
        i=1;
        elem=elem+1;
    end

    if round(k)==round(KK)


        break
    end


    IEND(elem,1)=daire_no{k}(i,1);
    IEND(elem,2)=daire_no{k+1}(i,1);
    IEND(elem,3)=daire_no{k+1}(i+1,1);
    IEND(elem,4)=daire_no{k}(i+1,1);

    elem=elem+1;

end
```

```
%  LM Array
for i=1:num_of_elem
for j=1:ND
for k=1:NDOFD
LMDO(k,j,i)=IDDO(k,IEND(i,j));
end
end
end
```

```
%          Assign the Coordinates of the Nodes of Each Element
for cs=1:num_of_elem
    XCORD(cs,1)=XX(IEND(cs,1));
    XCORD(cs,2)=XX(IEND(cs,2));
    XCORD(cs,3)=XX(IEND(cs,3));
    XCORD(cs,4)=XX(IEND(cs,4));

    YCORD(cs,1)=YY(IEND(cs,1));
    YCORD(cs,2)=YY(IEND(cs,2));
    YCORD(cs,3)=YY(IEND(cs,3));
    YCORD(cs,4)=YY(IEND(cs,4));
end
```

```
%                    Displacement Boundary Conditions

%       =1 (Constrained- for supports)
%       =0 (free)
%       =-1 PRescribed displacement other than support conditions
%-------------------------------------------------------------------------
IB=transpose(NODE_NO(1:nodes));
[IB1 IB2]=size(daire_no{1});
[IB3 IB4]=size(IB);
for ib=1:IB1
    NF(ib,1)=-1;
    NF(ib,2)=0;
end
for ib=(round(KK)-1)*IB1+1:IB3
    NF(ib,1)=1;
    NF(ib,2)=1;
end
%-------------------------------------------------------------------------
%BC_node=2*(360/angle);
%BC_node_xy=zeros(BC_node,3);
%BC_node_xy(:,1)=[daire_no{1};daire_no{round(KK)}];
%s0=zeros((360/angle),1);
%s1=zeros((360/angle),1);
%s2=zeros((360/angle),1);
%
%s0(:,1)=-1;
%s1(:,1)=1;
%s2(:,1)=0;
%BC_node_xy(:,2)=[s0;s1];
%BC_node_xy(:,3)=[s2;s1];
%---------------------------
%BC_node=(360/angle)+13;
%BC_node_xy=zeros(BC_node,3);
%BC_node_xy(:,1)=[daire_no{1}(1:7,1);daire_no{1}(19:24,1);daire_no{round(KK)}];
%s0=zeros(13,1);
%s1=zeros((360/angle),1);
%s2=zeros(13,1);
%
%s0(:,1)=-1;
%s1(:,1)=1;
%s2(:,1)=1;
%BC_node_xy(:,2)=[s0;s1];
%BC_node_xy(:,3)=[s2;s1];
%---------------------------
%BC_node=(360/angle)+1;
%BC_node_xy=zeros(BC_node,3);
%BC_node_xy(:,1)=[daire_no{1}(1,1);daire_no{round(KK)}];
%s0=zeros(1,1);
%s1=zeros((360/angle),1);
%s2=zeros(1,1);
%
%s0(:,1)=-1;
%s1(:,1)=1;
%s2(:,1)=1;
%BC_node_xy(:,2)=[s0;s1];
%BC_node_xy(:,3)=[s2;s1];
```

```
%-----------------------------
%BC_node=(360/angle)+(360/angle)/2+1;
%BC_node_xy=zeros(BC_node,3);
%BC_node_xy(:,1)=[daire_no{1}(1:(360/angle)/4+1,1);daire_no{1}(((360/angle)/4)*3+1:
(360/angle),1);daire_no{round(KK)}];
%s0=zeros((360/angle)/2+1,1);
%s1=zeros((360/angle),1);
%s2=zeros((360/angle)/2+1,1);
%
%s0(:,1)=-1;
%s1(:,1)=1;
%s2(:,1)=1;
%BC_node_xy(:,2)=[s0;s1];
%BC_node_xy(:,3)=[s2;s1];
%-----------------------------
%BC_node=2*(360/angle);
%BC_node_xy=zeros(BC_node,3);
%BC_node_xy(:,1)=[daire_no{1};daire_no{round(KK)}];
%s0=zeros((360/angle)/4+1,1);
%s1=zeros((360/angle)/4*2-1,1);
%s2=zeros((360/angle)/4,1);
%s3=zeros((360/angle),1);
%
%s0(:,1)=-1;
%s1(:,1)=0;
%s2(:,1)=-1;
%s3(:,1)=1;

%BC_node_xy(:,2)=[s0;s1;s2;s3];
%BC_node_xy(:,3)=[s3;s3];
%-----------------------------
BC_node=2*(360/angle);
BC_node_xy=zeros(BC_node,3);
BC_node_xy(:,1)=[daire_no{1};daire_no{round(KK)}];
s3=zeros((360/angle),1);
s4=zeros((360/angle),1);
%
s3(:,1)=-1;
s4(:,1)=1;

BC_node_xy(:,2)=[s3;s4];
BC_node_xy(:,3)=[s4;s4];
%-----------------------------
```