APPLICATION OF THE HYBRID STRESS FINITE ELEMENT METHOD IN STABILITY ANALYSIS OF MINE EXCAVATIONS

Osman Aliu Momoh
B.Eng. (Mining), M.Eng. (Mining)

A thesis submitted to the Faculty of Graduate Studies
and Research, in partial fulfillment of the
requirements of the degree of
Doctor of Philosophy

Department of Mining and Metallurgical Engineering
McGill University
Montreal, Quebec
June, 1989

Short Title :

HYBRID STRESS F.E. METHOD IN STABILITY ANALYSIS OF MINE EXCAVATIONS

Osman Aliu Momoh
B.Eng. (Mining), M.Eng. (Mining)

A thesis submitted to the Faculty of Graduate Studies
and Research, in partial fulfillment of the
requirements of the degree of
Doctor of Philosophy

Department of Mining and Metallurgical Engineering
McGill University
Montreal, Quebec
June, 1989

ACKNOWLEDGMENTS

I wish to acknowledge my indebtedness and appreciations to my supervisor, Professor R. N. Yong, William Scott Professor of Civil Engineering and Applied Mechanics, for his abiding encouragement, guidance and advice throughout the period it has taken me to complete this dissertation.

My deep gratitude also goes to the mining staff of the Department of Mining and Metallurgical Engineering, in particular to Professor H. Mitri, who advised me and devoted a lot of time in careful review of the implementation procedures and in reading of this thesis.

Finally, this dissertation is dedicated to my parents and brothers, long since departed, who had supported, nurtured and prepared me to undertake this project during my early education, and to my wife and children, who have been patient and supportive throughout the period it has taken me to carry out the task.

RÉSUMÉ

Cette thèse concerne la formulation et le développment d'une nouvelle procédure numérique, appelée la méthode de contrainte hybride pour éléments finis. La formulation proposée considère la contrainte et le déplacement comme inconnus primaires et diverge de façon radicale des modèles d'éléments finis se basant sur le déplacement.

Parmi les caractéristiques spéciales de ce nouveau modèle, on retrouve la définition des équations de rigidité des éléments en termes de système de coordonnées locales plutôt que globales, l'usage efficace du schème de mise en mémoire pour la solution de données "Skyline" ainsi que l'usage rapide et efficace de routines de production et de modification de maille spécialement développées pour les applications géotechniques.

Les essais effectués sur les problèmes d'élasticité de solutions à forme fermée démontrent que la méthode proposée est précise et nécessite relativement moins d'éléments que les autres méthodes à éléments finis pour obtenir la même précision. Les études de cas effectuées sur deux problèmes de conception minière ont reproduit les résultats obtenus par d'autres procédures de modelage numérique dans des limites acceptables et ont démontré que la méthode proposée peut être utilisée pour solutionner de façon efficace et réaliste les problèmes de géoméchanique.

ABSTRACT

This thesis reports the formulation and implementation of a finite element model, which is based on the so called Hybrid Stress Finite Element Method. The proposed formulation is based on stresses and displacements as the primary unknowns and is radically different from displacement-based finite element models.

Among the special features of the model are the casting of the element stiffness equations in terms of local rather than global coordinate system, the use of the active column method known as the "Skyline" technique for the solution of equilibrium equations, and the use of fast and efficient mesh generation and alteration routines developed specially for geotechnical applications.

Verification tests carried out on problems in elasticity with closed form solutions show that the proposed method is accurate and that it requires relatively fewer elements than displacement-based finite element method to achieve the same accuracy. Results of case history studies done on two mine design problems matched those from other numerical modelling procedures within acceptable limits and showed that the proposed method can be used to solve geomechanics problems efficiently and realistically.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 The Role of Mining and Minerals in World Economy

Of the three traditional industries basic to human existence and development, viz, mining, agriculture and forestry, the only thing common to them is that their resources come from the earth. Agriculture and forestry depend on the land for their basic raw materials but in general do not completely and irreversibly deplete these resources. The nutrients which the farm and forest products take from the land are often returned to the land, sometimes in artificially contrived greater measures that ensure a better yield of the resource in the succeeding cycle of exploitation. Mining on the other hand depends on a resource which, once depleted cannot be replenished or replaced. While man can, to a large extent, control the basis of agriculture and forestry, the same is not true for minerals: man has no control over the process of forming the minerals nor the geologic setting of their deposition. The mining engineer is, as it were, constrained to work with the environment of the particular ore deposit in order to exploit it.

Throughout the ages, man has been excavating below the surface of the earth for minerals for various purposes, e.g. for ornaments, for health and medicinal purposes, for building and finally, for articles of war and security, Jumukis [1]. These were the basic needs of man and their satisfaction has usually resulted in an increased population worldwide as well as the modernization of science and technology. This increase in all aspects of human endeavours which minerals have helped to bring about has in itself resulted in an increased demand for greater quantities of minerals so that it is no great surprise that mining has progressed from the traditional small surface and near-surface openings to the great surface and underground excavations of today.

1.2 The Nature of Rock Masses

Rock differs from most other engineering materials in that it contains fractures of all sizes and shapes which render its structure discontinuous. There is therefore a clear distinction between a rock material and a rock mass Goodman [2], Brady and Brown [3]. A rock material is the term used to describe the intact rock between discontinuities while a rock mass is the total in-situ medium containing several discontinuities. These discontinuities are often faults, joints, folds, bedding planes, and other structural features. The structure of a rock mass describes the nature and distribution of the structural features within the rock mass. Rocks are therefore discontinuous, often heterogeneous and definitely anisotropic engineering materials and their response to the field forces of their physical environment is therefore influenced by their structures. For underground hard rock mines in particular, the predominant structural features are joints, dykes, shear zones, faults, and folds.

Joints are the most common and the most significant feature. Joints are breaks of geological origin along which there has been no visible displacement. They often occur in sets, the distinguishing feature of any set being the same orientation and size in space. Faults and joints are fractures with the difference that in faults, identifiable shear movement has taken place.

1.3 Rock Mass Response to Imposed Force Fields

In other fields of engineering that deal with materials which are homogeneous, isotropic and elastic, the study of their response to imposed forces is predictable and the application of the relevant theories of mechanics is well established. Even in the older but closely related field of soil mechanics, systematic documentation of case histories has been in progress for over 40 years and has now resulted in well-established empirical design methods which are also supported by theoretical analyses.

The following factors are generally recognized as influence factors in the consideration of the response of rock masses to the force fields of their physical environment:

1.3.1 Rock Fracture

The nature of rock masses renders their response to imposed forces very different from the response of conventional engineering materials such as steel. In general, the stress fields operating in rock masses at depth are compressive so that established theories which usually deal with tensile failure are not immediately applicable. Rocks are very weak in tension and every strong in compression. Under high confining stresses, they will fail by shear deformation and their strength is very dependent on the magnitude of the confining stress.

1.3.2 Size effects

The response of rock to imposed load is highly dependent on the size of the loaded volume. This fact is attributed to the discontinuous nature of a rock mass whose strength and deformational properties are influenced by the properties of the rock material and those of the various structural geological features.

1.3.3 Tensile Strength

The tensile strength of rock is one of its most distinguishing feature from other engineering materials and in general is an order of magnitude lower than its compressive strength. Rock is therefore conventionally described as a no-tension material, especially as joints and fractures can offer little or no resistance to tensile stresses.

1.3.4 Groundwater Effects

Groundwater affects the response of rock masses in two ways. Under the effective stress law, water under pressure in the joints defining rock blocks reduces the normal effective stress between the rock surfaces, thus reducing the potential shear resistance which can be mobilized by friction. Secondly, the mechanical strength of the rock mass is reduced by the deleterious action of the water on particular rocks and minerals such as clays.

1.3.5 Effect of Loading Rate

Many rocks are sensitive to the rate at which loads are applied. At low stress levels, most rate-sensitive rocks exhibit primary creep behaviour which is characterized by deformation at a decreasing rate to a final state under a constant load and by recovery to the initial undeformed state once the loads are removed. However, in some cases, particularly for salt type rocks such as potash, the stress level often exceeds the creep yield limit and secondary creep behaviour, in which displacements are no longer recoverable and the creep rate approaches a constant value, set in.

These factors are by no means exhaustive but it is clear that the response of a rock mass to the force fields of its physical environment depends highly on the nature of the rock mass and the environment. Thus, the need for constitutive models which are capable of representing these different types of rock mass behaviour is important.

1.4 Types of Underground Mining Excavations

Ore extraction by underground mining methods involves the creation of different types of openings for different purposes. These openings are best classified by the duration of time

they are expected to remain open. The first type of openings thus comprises the main shaft, level drives, crosscuts, ore haulage drives, ventilation drives and airways. These constitute the mine access and service openings. They are expected to remain open for the life of the mine and are usually developed in the host rock.

Access crosscuts, drill headings, access raises, extraction headings and ore passes which are directly related to ore development and production constitute the second type of mine openings. They are usually developed in the ore body or very close to it and are expected to remain open for the duration of the mining activity in their immediate vicinity. Many of these openings are eliminated by the mining process.

The third type of mine opening is developed in the ore source itself or the stope. In general a stope may be a free standing excavation with well defined boundary walls that are strong enough to support themselves, the excavation growing in size as the ore is mined. Sometimes however, a stope requires some kind of support either in the form of the broken ore itself or some engineered mechanical support. The lifetime of a stope is defined by the duration of active ore extraction. It is usually shorter than the life spans of the first two types.

It is clear from the foregoing that in order for safe and economic exploitation of an ore body to be possible, any mine opening must remain open for its expected life span, without the danger of precipitating collapse, due, perhaps to the factors that pertain to the mining process in combination with the rock mass properties and environment.

1.5 Sources of Instability in Underground Excavation

According to Bieniawski [4], there are four principal sources of instability encountered in underground mining excavations identified as follows:

1.5.1 Instability due to Adverse Structural Geology

This type of instability tends to occur in hard rocks which are faulted and jointed and where several sets of discontinuity are steeply inclined. Stability can sometimes be improved by a relocation or reorientation of the excavations. Usually however, extensive support system is required. In all cases, the cost of mining is high.

1.5.2 Instability due to High Stress

This is also generally associated with hard rocks and can occur when mining is at great depths or when very large excavations are created at shallow depths, or when very high

tectonic stresses are encountered. Changes in the shape of the excavations and repositioning the excavations with respect to one another sometimes helps but support is usually required.

1.5.3 Instability due to Weathering

Weathering and sometimes swelling occur generally in weak rocks generally found near the surface. Such rocks sometimes also occur as isolated seams within strong competent rocks. The resulting instability is best dealt with by protecting the surface of the rock from contact with moisture.

1.5.4 Instability due to Excessive Groundwater

Groundwater pressure or flow will cause instability in underground excavations when these excavations are below the water table, which is often the case. A consequence of this is that most underground mines routinely pump water out of the mine as an essential operation. The instability due to this source becomes a problem when it is associated with one of the above instabilities.

It is fair to state that the first two types of instabilities are the most serious to deal with. Indeed the instability due to excessive rock stresses precipitates some other forms of instabilities. As mining excavations grow in size, the loss of support previously provided by the mined rock results in the transfer of stresses to the adjacent rock masses which may then bear more than their capacity. The consequences of this include rock bursts, spalling of sides, creation of fractures and thus channels for groundwater flow.

1.6 The Need for Realistic Mine Design Techniques

In the past few decades, changes in mine economics as a result of several factors have resulted in the use of larger equipment with resulting increase in stope and access tunnel dimensions. The increasing volume of underground mining activities combined with the high cost of equipment and technology that accompany their operations lend to them an air of permanence formerly reserved to Civil Engineering works. Thus, a typical large mine nowadays has major shaft systems with their surrounding complex of haulage ways, ore passes, crusher stations, pump stations and underground engineering services. Consequently, such excavations must be secure against rockfalls and other forms of instability.

The ultimate objective in mine design is to control rock displacements into and around mine excavations, Brady and Brown [3]. Elastic displacements around mine excavations are typically small. Rock displacements of engineering consequence may involve such processes as fracture of intact rock, slip on a geological feature, excessive deflections of roof and floor rocks, or unstable failure in the system during which, stored potential energy is suddenly

released thus resulting in changes in the equilibrium configuration of the structure. These modes of rock response indicate a methodology of designing stable excavations in rock. This methodology includes the accurate determination of the strength and deformation properties of the orebody and the enclosing waste rock mass, the geologic parameters of the rock mass and the use of analytical methods to evaluate each of the possible modes of response of the rock mass.

Thus, the design of underground mining excavations plays an important role in modern mining philosophy and the science under which this design is practised is called ROCK MECHANICS. A widely accepted definition of rock mechanics is the one first offered by the *United States National Committee on Rock Mechanics* in 1964, and later modified in 1974, Judd [5]:

> "Rock Mechanics is the theoretical and applied science of the mechanical behaviour of rock and rock masses; it is that branch of mechanics concerned with the response of rock and rock masses to the force field of their physical environment."

Rock mechanics is thus of fundamental relevance to Mining Engineering because the act of creating mining excavations changes the force fields of the physical environment of the rocks. The study of the response of the rocks to these changes requires the application of analytical techniques developed specifically for the purpose. Rock mechanics is a branch of geomechanics, which is concerned with the mechanical responses of all geologic materials, including rocks.

Figure 1.1 illustrates the main parts of a rock mechanics program. It is apparent that numerical modelling, the area in which this thesis can be classified, is only a part of several multipass loops. It is however, an important area. Various models of mining methods from start up to end of mine life can be examined and in conjunction with mine monitoring, model calibration that accurately reflects the response of the mine structures to mining operations can be made.

The need to reduce the chances of failure of critical mine openings has made it very important that advanced techniques be developed for designing the mine openings and for general mining operations. Traditionally, mine design has been based on a combination of experience and relatively simple analytical models modified by empirical factors to account for the 'non ideal' nature of rock. With today's high tonnage, and bulk mining methods operating at greater depths, there is a need for rational methods for designing the mine geometry which explicitly consider the geological structure of the ore body and the host rocks and account for the interaction between the various openings. This need is largely being met by the appropriate use of the stress and displacement analysis techniques in the field of rock mechanics

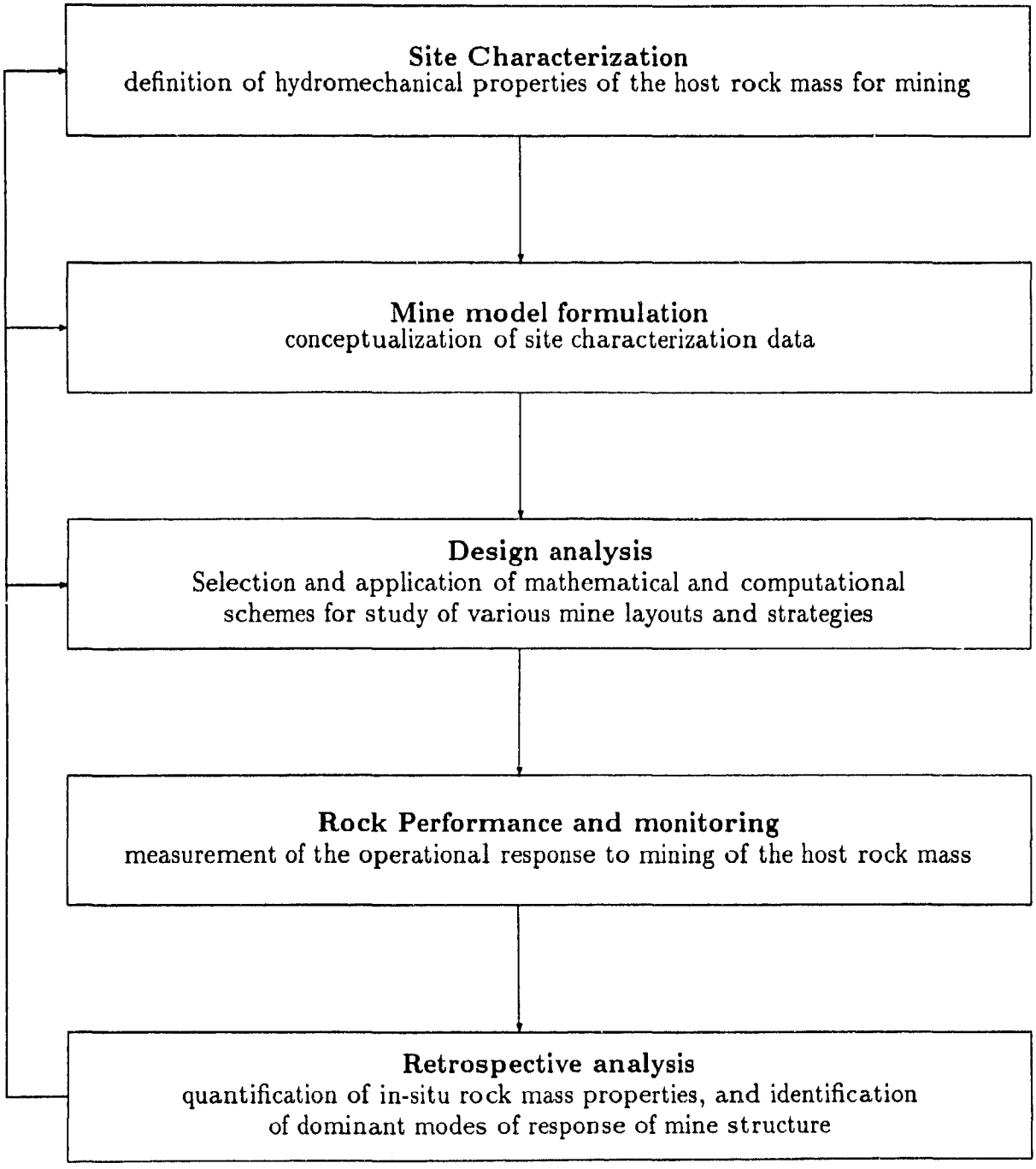


Figure 1.1: (after Brady and Brown [3]) Components and Logic of a Rock Mechanics Program

1.7 Objectives of this Thesis

There were four main objectives in undertaking this research:

1. To develop from mathematical principles, a numerical model for analysing stresses and displacements specifically tailored for excavations in rock masses, and based on the hybrid stress finite element technique.

2. To write a computer code of the method and verify it by using problems in elasticity with closed-form solutions.

3. To apply the computer code in solving actual mining stability problems which have been examined by different methods and compare the solutions, and,

4. To develop computer programs designed to ease the problems of generating and altering finite element meshes associated with excavations in rock masses.Ω

Chapter 1, this chapter, addresses the role that minerals and therefore, mines have played and are continuing to play in the life of man and in the world economy. It also sets the stage for the following chapters and stresses the importance of the work reported in the thesis. Subsequent to this chapter, an overview of the methods of stress and displacement analysis in geomechanics is given in Chapter 2. It is shown that the particular method chosen in this thesis is relatively new and unexploited in mining applications. Chapter 3 gives the detailed formulation of the relevant mathematics of the hybrid stress finite element using eight node isoparametric quadrilateral elements. This is followed in Chapter 4 by the mathematical formulation of the element stiffness matrix and the associated load vectors for the hybrid stress model.

In Chapter 5, a description of the computer programs is given. All the programs are written in Fortran 77. These programs include the hybrid stress computer code, two quadrilateral mesh generation programs, and a program for altering quadrilateral meshes for the purpose of creating voids or excavations in them. In Chapter 6, results of verification tests done on different types of problems with closed form solutions are discussed. Chapter 7 gives a detailed description of two case studies done on two Noranda mines. Discussions, conclusions, contributions and suggestions for further research then follow in Chapter 8.

There are five appendices in the thesis. Appendix A contains a listing of the hybrid stress program written from the reserach carried out. Appendices B, C and D contain listings of computer programs written for various purposes as described in Chapter 5. Appendix E contains some important mathematical derivations and matrices that are crucial to the programming of the hybrid stress finite element method.

Chapter 2

Methods of Stress and Displacement Analysis in Geomechanics

2.1 Introduction

The methods available in geomechanics for solving problems of stress and strain fall into three broad groups. These are, direct measurement and observational methods, analytical methods and, numerical methods as discussed below. Analytical and numerical methods start off by choosing a material model in form of the constitutive equations of linear elasticity. The theory of linear elasticity strictly applies to the case of a material with a stress-strain curve which is linear and completely reversible under all conditions. However, many materials which are not entirely linear elastic in their stress-strain behaviour, including rocks, are found to be linearly elastic for moderate departures from some condition which may be regarded as standard.

2.2 Direct Measurement and Observational Methods

Measurement of displacements and in-situ stresses forms an important part of mine design process before mine layout and planning, during the mining process, and after constructing important mine structures such as shafts, crusher stations and other permanent features in the mine. Measurements made before the excavation are used in the design process, those made during the excavation process are used to confirm the validity of the design and to provide a basis for any necessary changes in the original design. After construction, measurements are made to check the overall response of the excavation to changes in the surrounding rock masses, Hoek and Brown [6]. Data collected in the initial stage can also be used as input parameters into the numerical models discussed below.

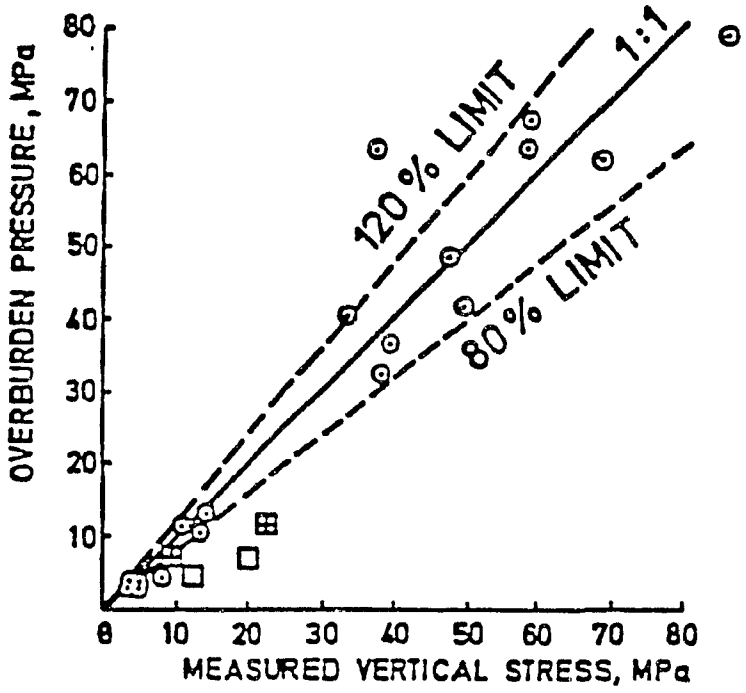


Figure 2.1: Correlation between Calculated and Measured Vertical Stresses, (after Bieniawski [4], p. 68)

The magnitudes and directions of the principal stresses which exist in the rock mass before the creation of an excavation play an important role in the subsequent response of the excavation to the changing stress field as more excavations are made. The creation of an excavation thus disturbs the initial in-situ stress, Bieniawski [4]. Ground or field stresses have been classified into virgin or original stresses, and induced stresses due to excavation. The virgin stresses themselves can be of gravitational, tectonic or residual type. Gravitational stresses are those due solely to the effect of gravity on the overburden rock. Tectonic stresses are those due to straining in the earth's crust, and residual stresses are stresses that remain after the cause has been removed.

Rock stress measurements feature several techniques which may involve either overcoring methods, flat jack methods or hydrofracturing. Such measurements have been conducted in several parts of the world and attempts have been made to correlate the values of the horizontal and vertical components with depth of overburden and with values calculated on the basis of gravity alone. Figures 2.1–2.4 depict some of these correlations.

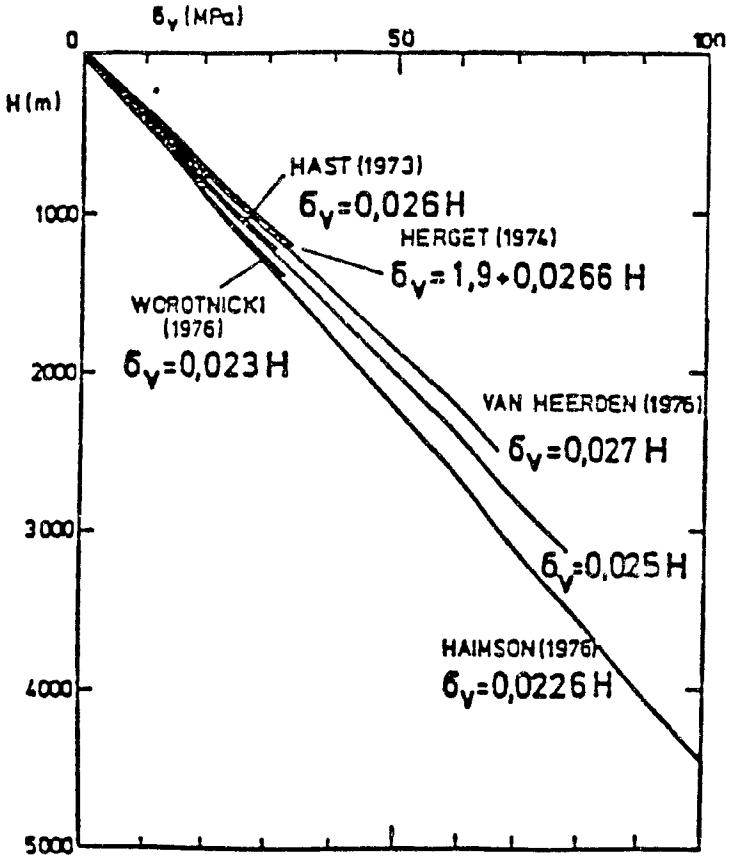


Figure 2.2: Plot of Vertical Stress versus Depth Below Surface,(after Bieniawski [4], p. 68)

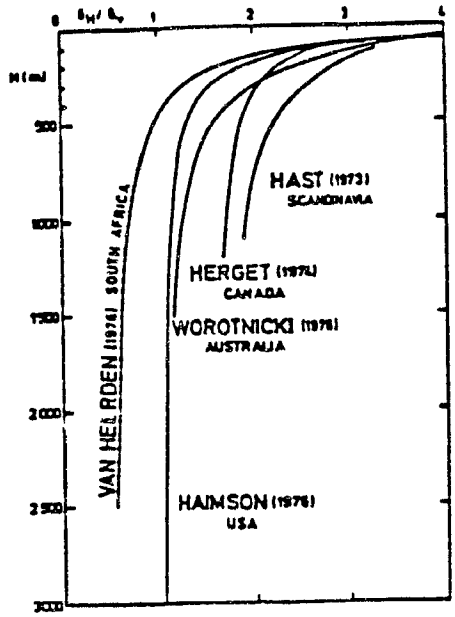


Figure 2.3: Variation of Ratio of Average Horizontal Stress to Vertical Stress with Depth below Suface, (After Bieniawski [4], p. 69)

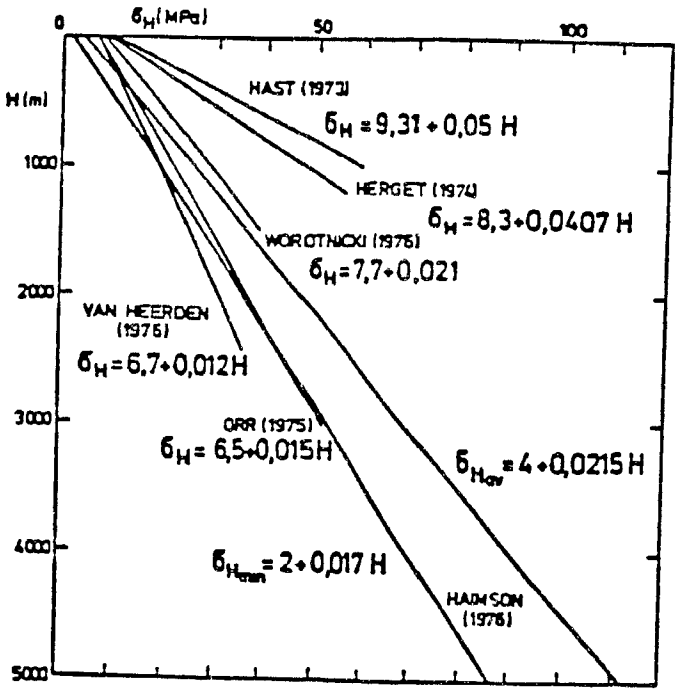


Figure 2.4: Comparison of Relationship Proposed for the Variation of Horizontal Stress Component with Depth below Surface, (After Bieniawski [4], p. 70)

2.3 Numerical Methods

Once the constitutive equation which represents the behaviour of the rock mass under stress has been identified and the boundary conditions have been specified, the problem of solving for the stresses and displacements around an underground excavation becomes one of solving a boundary value problem. Two approaches to numerical modeling are identified, both recognizing geologic structures as being discontinuous owing to joints, faults and bedding planes. These approaches are as follows:

2.3.1 Continuum Models.

The continuum model, of which there are two types, treats the rock mass as a continuum intersected by a number of discontinuities. Differential models characterize the entire region and include the finite difference and the finite element methods. These models utilize the classical theory of elasticity and plasticity to compute the stresses and displacements induced in the initially stressed rock following excavation. It is assumed that displacements are continuous everywhere within the rock mass. This leads to an idealization and simplification of its geometrical and mechanical properties.

The finite difference technique, Desai and Christian [15], was the first numerical approach

formulated on mathematical basis, to be applied to problems in continuum mechanics. The basic concept of the method is to replace all continuous derivatives by the ratio of changes in variables over a small but finite distance. Thus, the differential equations which govern the physical problem are transformed into a system of algebraic equations. A finite difference grid must be set up and interpolation functions which express the variations of the unknowns between the nodes need to be specified. The resulting system of equations is large but sparse.

For more complex excavation shapes and boundary conditions, the finite element method and the boundary element method are two of the several new schemes that can be used. The boundary element method is suitable for modelling unbounded problems in elasticity while the finite element method has advantages in its capability of modelling non-linear material behaviour, sequential excavations and other mining induced problems. Nowadays, hybrid computational schemes which combine finite element methods or boundary element methods with discrete element methods are coming into use, see Beer and Meek [16]. Details of the hybrid finite element method of this thesis are given in Chapter 3.

Integral or boundary element methods involve the discretization of the interior or exterior boundaries. The interfaces between different material types and discontinuities are treated as internal boundaries and are also discretized. Boundary element methods produce much smaller systems of algebraic equations than a finite element or finite difference method for the same problem. However, the smaller system of equations is not sparse nor is it banded as for the other continuum type models.

2.3.2 Discontinuum Models

Discontinuum models feature numerical procedures involving the equations of motion of particles or blocks rather than the continuum. These models are used whenever independent rock block movements must be taken into account, such as in block caving mining methods, rock slope failure, roof collapse or flow of ore in a stope or bin.

2.3.3 The Finite Element Method

The original concept of finite element methods for solids was developed by Turner, Clough, Martin and Topp [17] who applied the matrix displacement methods to plane stress problems using triangular and rectangular elements. Argyris [18] included in his treatise on matrix structural analysis a derivation of the stiffness matrix of a plane stress rectangular panel. The formulations of element stiffness matrices by early investigators were not based on the field equations of the entire elastic continuum.

In the early 1960's, it became apparent that the finite element method can be interpreted as an approximate Ritz method associated with a variational principle in continuum

mechanics, Pian [19], Pian and Tong [20], Washizu [21]. From such variational principles as illustrated by Figure 2.5, it is possible to derive numerous finite element models which may lead to different methods which involve only displacements, only force, or a hybrid of displacement and force. Washizu's contributions, as a typical example can lead to the derivation of finite element methods for solving initial strain problems, finite displacement problems, plasticity problems and others. Pian [22] has given a thorough summary of the application of the variational principles to the derivation of finite element equations for solids, and has classified finite element methods according to Washizu [21], Table 2.3.7.

In such methods, a solid continuum is first subdivided into an assemblage of discrete elements called finite elements, which are connected along continuous interelement boundaries. Piecewise continuous displacement and/or stress fields are then assumed in each element and the resulting equations from the application of the variational principles are simultaneous algebraic equations which may have either generalized displacements, generalized internal forces or stresses, or, both displacements and forces at the nodal points as unknowns to be evaluated.

The nature of the final matrix equations has been used as the basis for one type of classification of finite element methods and so, the three categories given above are often referred to as,

- the displacement or compatible method,
- the force or equilibrium method,
- the mixed method, and
- the hybrid method

Turner et al. [17], were the first to apply the technique of matrix displacement methods to plane stress problems using rectangular and triangular elements. The stiffness matrices were derived by the direct stiffness method but the formulation was not based on the field equations of the entire elastic continuum.

This study follows the definition of the hybrid finite element method given by Pian [23]. The hybrid stress method involves assumed equilibrating stresses only within each element and compatible displacements along the interelement boundaries. The hybrid displacement method involves assumed continuous displacement distribution within each element and equilibrating surface tractions along the interelement boundaries.

The displacement method is usually derived from the Principle of Stationary (minimum) Potential Energy, in which the displacements are assumed within the finite element such that they satisfy continuity conditions within the element and along the interelement boundaries.

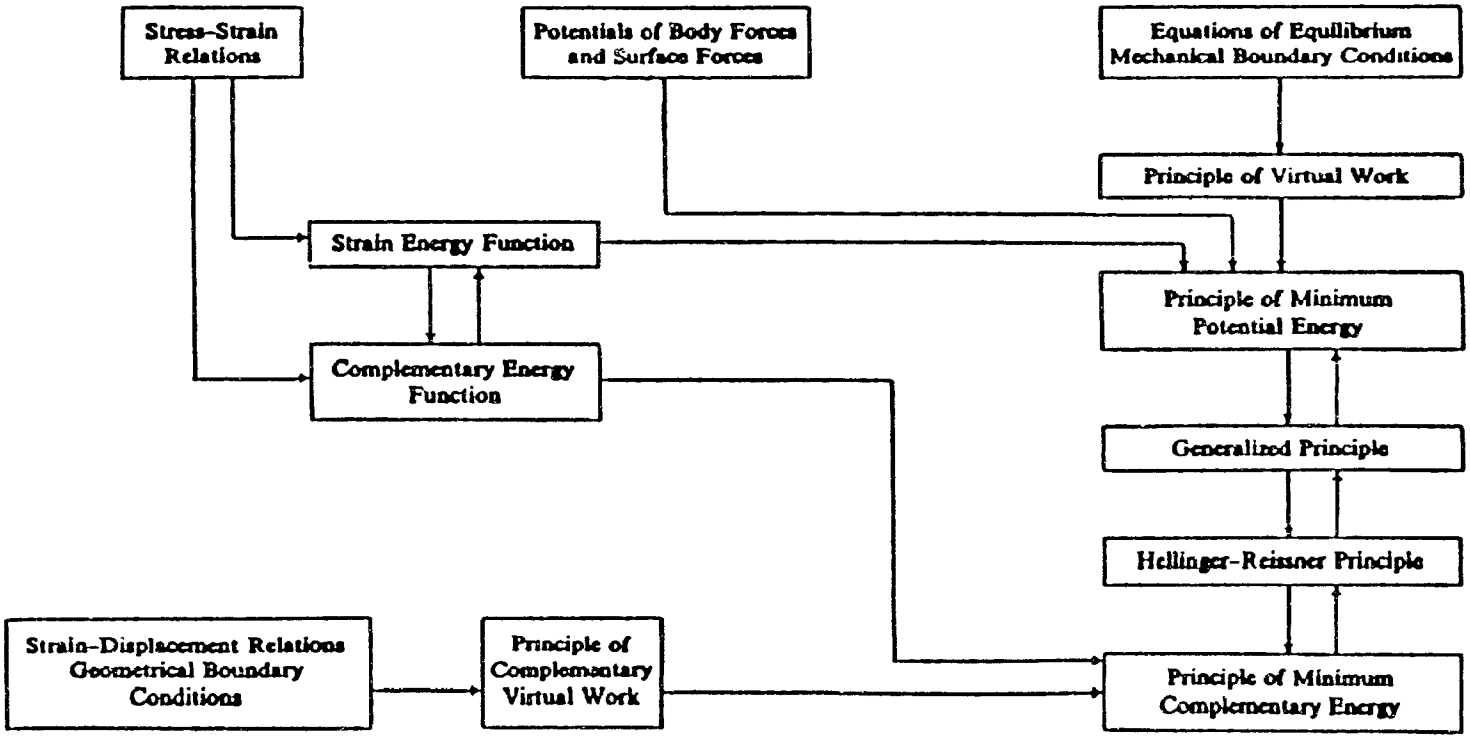


Figure 2.5: Variational Principles in the Small Displacement Theory of Elasticity

The stresses are then computed from the gradient of displacement functions. Boundary tractions and the equilibrium equations are not exactly satisfied within the element, as a result of which, the predicted stresses may not be accurate.

A method that can predict stresses with greater accuracy than the displacement method is the force method which is also known as the equilibrium method. It can be derived from the Principle of Stationary (minimum) Complementary Energy. The disadvantage of this method is that it is usually difficult to derive a stress field that satisfies the prescribed tractions and equilibrium equations. Also, the predicted displacements by this method may not be accurate, Pian and Tong [20].

The mixed method is often derived from the Hellinger-Reisner variational principle, in which the stress and displacement fields are assumed separately for each element.
The hybrid method can either be derived from the Modified Potential Energy or the Modified Complementary Energy Principles, which lead respectively to the Hybrid Displacement Model and the Hybrid Stress Model. In the Hybrid Displacement Model, the displacement field is assumed within the element and independent displacement and/or stress fields are assumed on the element boundaries. In the Hybrid Stress Model, an equilibrium stress field is assumed within the element and an independent displacement field is assumed on the element boundaries.

In geomechanics, Desai and Christian [15], used the Displacement Model to solve problems involving such complexities as material, geometric and boundary non-linearities. It is the author's belief that the hybrid stress method can be used to solve complex problems in geomechanics because of its specialized features some of which are discussed in the following subsections.

Pian, [24] concluded that the use of the hybrid stress model will yield a more flexible structure that the compatible model using the same boundary displacement approximations, and a more rigid structure that the equilibrium model using the same internal stress approximations. He further stated that the finite element method based on assumed stresses can provide more accurate stress estimation than the assumed displacement scheme. Appa Rao [76], in his studies on thick spheres and long thick cylinders also stated that the radial stresses obtained by the hybrid stress model are closer to the exact values than the stresses obtained by the assumed displacement model.

2.3.4 Improvement of Finite Element Solution by the Introduction of Special Elements for Prescribed Stress Boundaries

Computations using the element stiffness matrices formulated by the conventional assumed displacement method will not, in general, produce the correct stress components at the prescribed stress boundaries. As an example, at a stress-free boundary, spurious stresses will result. These can be prevented by the use of special free-boundary hybrid elements for which appropriate (column) vector of stresses can be made so as to give zero values of the generalized tractions on appropriate edges. Pian [24] and Yamada et al. [25] have shown that the use of the special stress-free boundary element can make considerable improvement on the finite element solution.

2.3.5 Improvement of Finite Element Solution for Problems Involving Stress Singularities

In the conventional finite element method based on an assumed displacement compatible method,the stress distribution within each element has been proved to converge to the exact solution if the corresponding stress distribution remains finite everywhere in the domain. However, for a problem involving a stress singularity such as the elastic solution at the tip of a sharp crack, it cannot be proved that the finite element method using the conventional compatible elements will necessarily converge to the exact solution. On the other hand, when the hybrid stress method is used, it is possible to include special stress terms which represent the correct stress singularity behaviour. By extracting the singular part of the solution in its correct analytical form, the nodal displacements in the finite element analysis correspond to a solution without singularity. Thus, the convergence of the finite element solution is assured, Tong and Pian [26].

The application of sections 2.4.2 and 2.4.3 to problems of joints and fractures is obvious. There is an extensive literature on the topic, Akin [27]], Benzley [28], Roshdy [29], and Atluri et al. [30] being typical of the keen interest.

2.3.6 Construction Sequences

Finite element simulation of construction sequences is now being done by the stress hybrid procedure. Lightner [31] has reviewed past work done in this field such as embankments, excavations and initial stress problems. Desai and Christian [15] also made extensive reviews on construction sequences. The extension of this principle to excavation sequences in hard rock mining, and to the design of mining methods for sill and crown pillars is also obvious.

2.3.7 Interface modelling

Modelling the interface between dissimilar materials such as soil and rock has been done in the past in several ways. The load transfer mechanism between two such unlike materials is affected by the behaviour at the interface. Thus, during loading, special consideration should be given to the modes of deformation such as slip, non-slip, opening, and closing of contact surfaces. Goodman, Taylor and Brekke [32] developed one of the earliest joint elements for use in this type of analysis. Zienkiewicz et al. [33] have used isoparametric elements for modelling interface behaviour. Ghaboussi, Wilson and Isenberg [34] have criticized the preceding models mainly on the basis of numerical ill-conditioning of the element stiffness matrices. They derived an interface element similar to the Goodman element with supposedly better properties. However, most interface elements thus far developed are inadequate in many respects that make their utility very limited. However, research is going on for development of better interface elements.

Table 2.1: Classification of Finite Element Methods for Solid Continua

(after Pian[23])

F.E MODEL	VARIATIONAL PRINCIPLE	VARIABLES	PARAMETERS IN π_{mc}	UNKNOWNS IN MATRIX EQUATIONS	MATRIX METHOD
Compatible	Potential Energy	Displ.: $\{u\} = [A]\{q\}$;	Nodal displ.: $\{q\}$;	$[q]$;	Displ.
Equilibrium I	Complim. Energy	Stress func.: $\{u\} = [B]\{p\}$;	Nodal stress function: $\{p\}$;	$\{p\}$;	Force
Equilibrium II	Modified Complim. Energy	Stress: $\{\sigma\} = [P]\{\beta\}$; Boundary displ.: $\{q\} = \int[\Phi^T]\{u_s\}ds$;	Stress param.: $\{\beta\}$; Generalized Bd.: displ.:, $\{q\}$;	$\{q\}$;	Displ.
hybrid Stress	Modified Complim. Energy	Stress: $\{\sigma\} = [P]\{\beta\}$; Boundary displ.: $\{u^T\} = [L]\{q\}$;	Stress param: $\{\beta\}$; Nodal displ.: $\{q\}$;	$\{q\}$;	Displ.
Hybrid Displ. I	Modified Potential Energy	Displacements: $\{u\} = [C]\{d\}$; Boundary traction: $[T] = [M]\{R\}$;	Displ. param:, $\{d\}$; Traction param:, $\{R\}$;	$\{R\}$;	Force
Hybrid Displ. II	Modified Potential Energy	Displacements: $\{u\} = [C]\{d\}$; Boundary traction: $[T] = [M]\{R\}$; Boundary displ.: $\{u\} = [L]\{q\}$;	Displ. param.: $\{d\}$; Traction param.. $\{R\}$; Nodal displ.: $\{q\}$;	$\{q\}$;	Displ.
Mixed Model	Reisner	Displacements: $\{u\} = [A]\{q\}$; Stress: $\{\sigma\} = [N]\{s\}$;	Nodal displ.: $\{q\}$; Nodal stresses: $\{s\}$;	$\{q\}$;	Displ.
				$\{q\}, \{s\}$;	Mixed
				$\{s\}$;	Force

Chapter 3

The Hybrid Stress Finite Element Method

3.1 Theoretical Basis of the Hybrid Stress Method

The Hybrid stress method was first introduced by Pian [19] in 1964 for the analysis of plane stress and plate bending problems. In this method, equilibrating stresses are assumed within each element in terms of undetermined stress coefficients and the displacements on the boundaries of the element are expressed in terms of the element nodal displacements. Subsequently, Pian and Tong [20] formulated the hybrid stress technique based on a modified stationary complementary energy principle, and they also considered the topics of convergence and bounds of the resulting equations.

There are two major attractions in this method: (i) that the unknowns of the overall equilibrium equations can be expressed in terms of the nodal displacements, and (ii) it is easier to construct compatible interpolation functions for displacements at interelement boundaries.

Among the several investigators who made useful contributions to the development of the hybrid stress method are Wissmann and Sprecht [35], Cook and Ladkany [36], Wolf [37], Spilker and Munir [38], and Ahmad and Irons [39]. Several investigators applied the hybrid stress method to the analysis of structures, and have reported that the technique predicts stresses and displacements with greater accuracy than does the displacement-based finite element method, Pian et al. [40], Barnard and Sharman [41], Tong et al. [42], Tong [43], and Spilker and Munir [44]. Ahmad and Irons [39] proved that the hybrid stress method is superior to the displacement method.

3.2 Hybrid Stress Model for Elastic Analysis

The modified complementary energy functional which has been used for the derivation of the hybrid stress method for an elastic continuum is expressed by Pian [19] as follows:

$$\Pi_{mc} = \sum_n \left[\int_{V_n} \frac{1}{2} D_{ijkl} \sigma_{ij} \sigma_{kl} \, dV - \int_{\partial V_n} T_i u_i \, dS + \int_{S_{\sigma n}} \bar{T}_i u_i \, dS \right] \tag{3.1}$$

where
σ_{ij} = stress tensor,
T_i = surface traction,
$\bar{T}_i$ = prescribed boundary tractions,
U_i = boundary displacements,
∂V_n = entire boundary of subregion V_n, and
$S_{\sigma n}$ = boundary along which prescribed tractions act.

These terms are illustrated in Figure 3.1 below.
D_{ijkl} is the elastic compliance matrix which, for a 2-dimensional body, is given by:

$$[D] = \frac{1}{2G} \begin{bmatrix} 1 & -\lambda & 0 \\ -\lambda & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

where
G, the rigidity modulus is equal to $E/[2(1+\nu)]$,
E is the modulus of elasticity or Young's modulus, and ν is the Poisson's ration.
For plane stress conditions, $\lambda = \nu/(1+\nu)$, and
for plane strain conditions, $\lambda = \nu$.

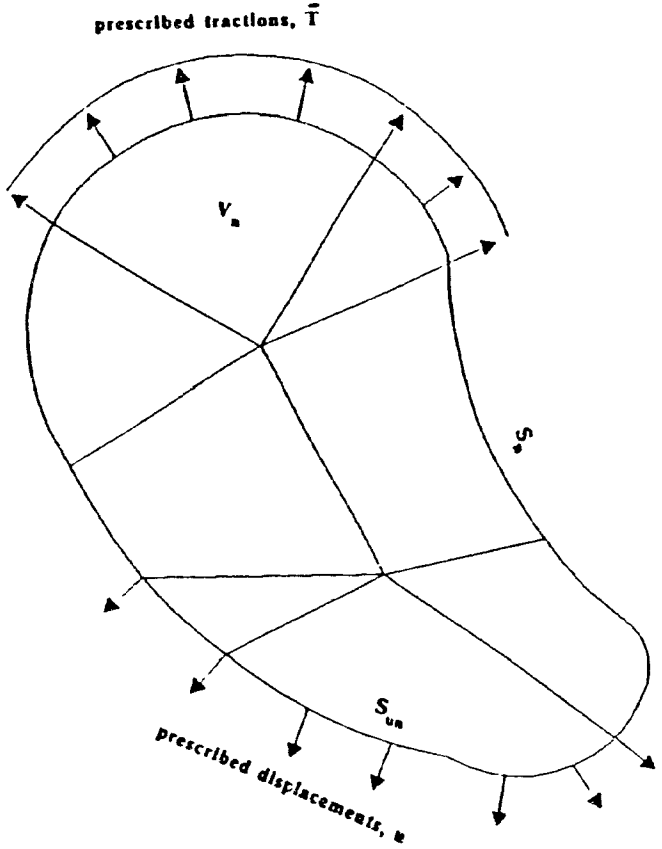
$\{\bar{T}\}$ is the vector of distributed surface tractions. The last integral in Equation (3.1) is evaluated on those element sides with externally applied loads.

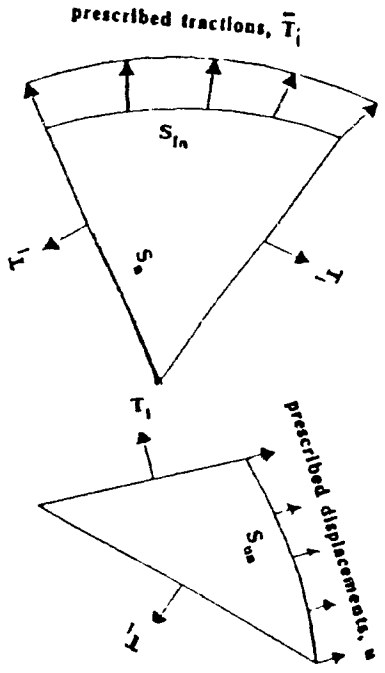
The stress distribution in an element can be expressed in terms of undetermined parameters $\{\beta\}$ by the equation

$$\{\sigma\} = [P]\{\beta\} + [P_b]\{\beta_b\} \tag{3.3}$$

where $[P]\{\beta\}$ satisfies the homogeneous equations of equilibrium and $[P_b]\{\beta_b\}$ is a particular solution of the equilibrium equations

$$\sigma_{ij,j} + F_i = 0 \tag{3.4}$$





$\partial V_n = S_n + S_{un} + S_{In}$

∂V_n = entire boundary of the continuum

S_{un} = portion of ∂V_n over which displacements are prescribed

S_{In} = portion of ∂V_n over which tractions are prescribed

S_n = portion of ∂V_n over which neither tractions nor displacements are prescribed

Figure 3.1: Notation for the Hybrid Stress Model

where F_i are the prescribed body forces. The matrices $[P]$ and $[P_b]$ are functions of the coordinates. $\{\beta\}$ are the so-called undetermined stress coefficients and $[P_b]$, $\{\beta_b\}$ are known. The surface tractions due to the assumed stress field can be expressed as:

$$T_i = \sigma_{ij}\nu_j,$$

or

$$\{T\} = [R]\{\beta\} + [R_b]\{\beta_b\} \tag{3.5}$$

where ν_j denotes the component of the unit vector in direction j. The inter-element boundary displacement functions are expressed in terms of an interpolation function $[L]$ and the generalized displacements q as

$$\{U\} = [L]\{q\} \tag{3.6}$$

Substituting Equations (3.3), (3.6) and (3.5) into Equation (3.1), we have:

$$\Pi_{mc} = \sum_n \left[\frac{1}{2} \{\beta\}^T [H] \{\beta\} + \{\beta\}^T [H_b] \{\beta_b\} - \{\beta\}^T [G] \{q\} + [\bar{Q}]^T\{q\} + C_n \right] \tag{3.7}$$

where

$$[H] = \int_{V_n} [P]^T [D] [P] \, dV \tag{3.8}$$

$$[H_b] = \int_{V_n} [P]^T [D] [P_b] \, dV \tag{3.9}$$

$$[G] = \int_{\partial V_n} [R]^T [L] \, dS \tag{3.10}$$

$$[G_b] = \int_{\partial V_n} [R_b]^T [L] \, dS \tag{3.11}$$

$$[\bar{Q}]^T = \int_{S_{\sigma_n}} \{\bar{T}\}^T [L] \{q\} \, dS - \{\beta_b\}^T [G_b] \{q\} \tag{3.12}$$

and

$$C_n = \int_{V_n} \frac{1}{2} \{\beta_b\}^T [P_b]^T [D] [P_b] \{\beta_b\} \, dV \tag{3.13}$$

3.2.1 Application of the Variational Principle

The variational principle states that of all admissible stresses and displacements, the solution which satisfies the equilibrium, compatibility and boundary conditions is distinguished by the stationary value of the functional with respect to variations in the stresses in the finite element, and displacements along the element boundaries.
Differentiation of Equation (3.12) with respect to $\{\beta\}$ and $\{q\}$ therefore, we get,

$$\frac{\partial \Pi_{mc}}{\partial\, \beta} = \sum_n ([H]\{\beta\} + [H_b]\{\beta_b\} - [G]\{q\}) \tag{3.14}$$

Equation (3.14) is zero at a maximum or minimum, at which state, β is given by

$$\{\beta\} = [H]^{-1}\,[G]\,\{q\} - [H]^{-1}\,[H_b]\,\{\beta_b\} \tag{3.15}$$

Note that $[H]$ is symmetric so that

$$[H] = [H]^T$$

and

$$[H^{-1}]^T = [H^T]^{-1}$$

Also,

$$\frac{\partial \Pi_{mc}}{\partial\, \{q\}} = \sum_n \left([\bar{Q}]^T - \{\beta_b\}^T\,[G]\right) \tag{3.16}$$

which, when set to zero yields

$$[\bar{Q}]^T = \{\beta_b\}^T\,[G] \tag{3.17}$$

Substituting Equations (3.15) into Equation (3.7), we have the final expression of Π_{mc} as:

$$\Pi_{mc} = -\frac{1}{2}\sum_n \{q\}^T\,[G]^T\,[H]^{-1}\,[G]\,\{q\} + \sum_n \{q\}^T\,[G]^T\,[H]^{-1}\,[H_b]\,\{\beta_b\} -$$

$$\frac{1}{2}\sum_n \{\beta_b\}^T\,[H_b]^T\,[H]^{-1}\,[H_b]\,\{\beta_b\} + \sum_n \left([Q]^T\{q\} + C_n\right) \tag{3.18}$$

which becomes

$$\Pi_{mc} = -\sum_{n} \left[\frac{1}{2} \{q\}^T [k] \{q\} - \{Q\}^T \{q\} + [B_n] \right] \tag{3.19}$$

where

$$[k] = [G]^T [H]^{-1} [G] \tag{3.20}$$

is the element stiffness matrix,

$$[Q] = [G]^t [H]^{-1} [H_b] \{\beta_b\} + \{\bar{Q}\} \tag{3.21}$$

is the element load vector, and

$$B_n = \frac{1}{2} \sum_{n} \{\beta_b\}^T [H_b]^T [H]^{-1} [H_b] \{\beta_b\} - C_n \tag{3.22}$$

is a constant.
The knowledge of the stiffness matrix and equivalent nodal forces of each element permits assembly of the overall stiffness matrix and the load vector of the system. After applying the prescribed boundary constraints, the system can be solved by any standard technique to obtain the nodal displacements.

3.3 Effects of Body Forces on the Hybrid Stress Formulation

The second part of Equation (3.3) represents the contribution to the element stresses due to body forces and may be written as

$$\{\sigma_b\} = [P_b]\{\beta_b\} \tag{3.23}$$

where the subscript b refers to body loads and the equation satisfies a particular solution of the equilibrium equations. For gravitational loading - which is usually the cause of body loads in geomechanics, the stress vector is given by

$$\sigma_x = \sigma_{xb} = 0.0$$

$$\sigma_y = \sigma_{y_b} = -\rho g y_b$$

$$\tau_{xy} = \tau_{xy_b} = 0.0$$

where
ρ = rock density,
g = gravitational acceleration,
y_b = thickness of overburden.

Matrix inversion is implied in the solution for $\{\beta_b\}$ in Equation (3.23) above. Since the $[P]$ matrix is 3x18 for reasons that will be explained later in Section 4.4, and Equation (3.23) is a particular solution of the equilibrium equations, the $[P_b]$ matrix may be chosen as the first three columns of the $[P]$ matrix for the element. Once the $[P_b]$ matrix has been evaluated as described, all the matrices and vectors which depend on it are then easily computed. $\{\beta_b\}$ can also be solved for from Equation (3.23).

3.4 Contributions to Equivalent Nodal Forces

The equivalent nodal forces constitute the load vector which is given by the equation

$$\{Q\} = [G]^T [H_b]^{-1} \{\beta_b\} + \{\bar{Q}\} = \{Q_b\} + \{\bar{Q}\} \tag{3.24}$$

where $\{Q_b\}$ = body load contribution to the load vector and Equations (3.21) are the contributions from prescribed loads including loads due to initial conditions such as stress, temperature or ground water pressure, (see next section). If there are no body loads, all body load terms in Equation (3.21) disappear and only Equation (3.12) remains.

3.5 Initial Stress Approach

Initial stresses in geomechanical applications at depth play a great role in the design and construction of excavations for mining and other uses such as for tunnels, underground storages, underground power plants and permanent waste disposal facilities. A construction sequence involves the alteration of the state of stress and may involve excavation, dewatering, deposition, and installation of support system. Each of these processes will alter the state of stress in the surrounding medium. In this study, emphasis is given to the simulation of excavation because it is the most common aspect in geomechanical applications.

Until recently, finite element modelling of excavation sequence has been based on the displacement method as originally proposed by Goodman and Brown [45]. Other investigators who have used variations of this technique are Lightner [31], Clough and Duncan [46], Mana [47] and Christian and Wong [48]. In this model, the geostatic stress, $\{\sigma_0\}$, prior to the start of excavation is determined from applied loads which are computed from the gravitational loading of the medium. Since this load is vertical, the proper lateral stresses are obtained

by the relation

$$\sigma_h = k_0 \, \sigma_v \tag{3.25}$$

where σ_h is the horizontal stress, σ_v the vertical stress and k_0 is referred to as the coefficient of lateral stress at rest. Poisson's ratio, ν, is then computed from the equation

$$k_0 = \frac{\nu}{1 - \nu} \tag{3.26}$$

Excavation is simulated incrementally as shown in Figure 3.2. In each increment, a stress-free surface is created by applying equivalent nodal forces on the excavation surface due to the increment of excavation. The equivalent nodal forces are then computed from the existing state-of-stress and applied with opposite sign to the excavated surface. The displacements, stresses and strains are then computed and added to the values for the previous step as shown in Equations (3.27), (3.28) and (3.29) below.

$$\sigma_i = \sigma_0 - \sum_j \Delta\sigma_j \tag{3.27}$$

$$q_i = q_0 - \sum_j \Delta q_j \tag{3.28}$$

$$\epsilon_i = \epsilon_0 - \sum_j \Delta\epsilon_j \tag{3.29}$$

The hybrid stress procedure adopted in this study for the initial stress approach follows the scheme illustrated in Figure 3.2. The stresses are computed directly on the boundary of the excavation by substituting the boundary coordinates into the assumed stress fields, using the excavated elements which have a common boundary with the unexcavated elements. The equivalent nodal forces are then computed by using the magnitudes of the boundary stresses and the interpolation functions used for the displacements of particular boundary, see Equation (3.6). After each step, the vector $\{\beta\}$ is accumulated for use in the computation of the stresses on the boundary of the excavation in the next step. This method works well for linear elastic media.

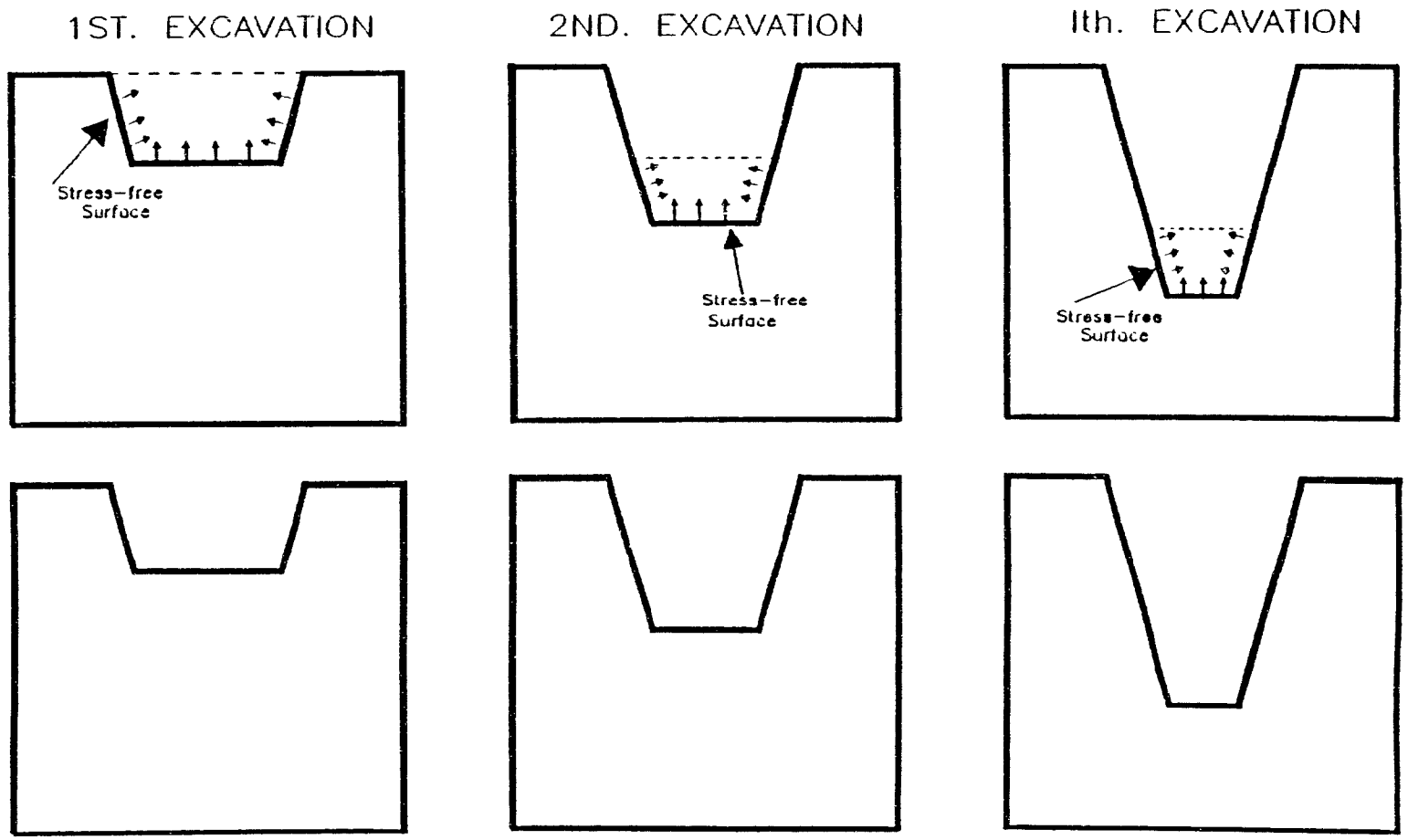


Figure 3.2: Simulation of Excavation Sequences

Chapter 4

Formulation of the Element Stiffness Matrix

4.1 Sign Convention and Constitutive Relations

The stresses at any point within a continuum are described by the nine components of the stress tensor; however, only six of them, three normal stresses and three shear stresses are independent. There are six strain components which are related to these stress components as follows:

$$\epsilon_i = D_{ij}\sigma_j \tag{4.1}$$

where the D_{ij} are the compliance coefficients. In general, there are 21 compliance coefficients but for isotropic materials, the number of independent coefficients reduces to 2. If a state of plane strain is considered, Equation (4.1) for plane strain then becomes

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{Bmatrix} = [D] \begin{Bmatrix} \sigma_x \\ \sigma_y \tau_{xy} \end{Bmatrix} \tag{4.2}$$

where the compliance coefficient matrix, $[D]$, is given in Equation 3.2. Equation (4.2) can be inverted to express the stresses in terms of strains.

The sign convention adopted is the same as in the theory of elasticity and continuum mechanics which can be stated as follows: The stress component, σ_{ij}, is positive if it acts in the positive j−direction on a plane whose outward drawn normal points in the positive i−direction. It is also positive if the i and j directions are both negative. By this definition, tensile stresses are positive and compressive stresses negative. All the stresses shown in

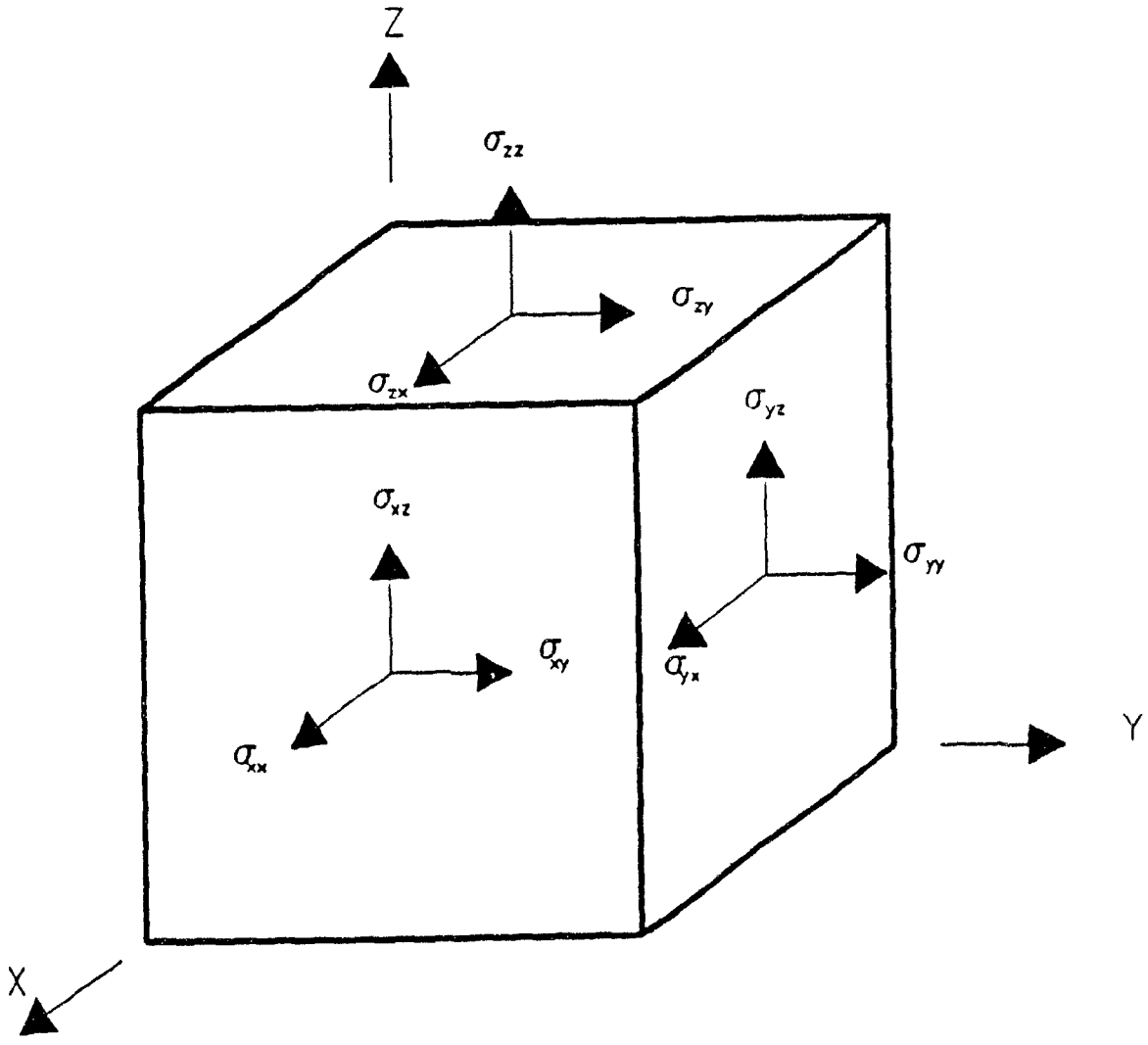


Figure 4.1: Components of the Stress Tensor

Figure 4.1 are positive. The kinematic analysis of the continuum, under assumption of small displacements, gives the necessary Equations relating the strains to the displacements, thus

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{Bmatrix} = \begin{Bmatrix} \partial u/\partial x \\ \partial v/\partial y \\ \partial u/\partial y + \partial v/\partial x \end{Bmatrix} \tag{4.3}$$

where u and v denote displacements along the x and y directions respectively. The final equilibrium of the body under the given load conditions must satisfy the equations of equilibrium which, in the absence of body forces, are given as

$$\begin{bmatrix} \partial/\partial x & \partial/\partial y \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix} = 0 \tag{4.4}$$

The analyses reported in this thesis were limited to bodies which were loaded within their elastic range.

4.2 Selection of an Admissible Stress Function

If the domain shown in Figure 3.1 has been discretized into, say, quadrilateral elements of arbitrary shapes but having straight sides, it then becomes necessary for the solution of plane problems to determine the stresses and displacements at every node of the domain. To accomplish this, a suitable stress function that satisfies equilibrium within the element boundary should be selected.

The solution of two-dimensional problems in elastic theory reduces to solving the two differential equations of equilibrium and the compatibility equations. Boundary conditions imposed on any particular problem are then used to determine the constants of integration. When body forces are absent or constant, the usual method of solving these equations is to introduce the Airy Stress function. This function uniquely defines the generalized stress components at any point within the element. There are many other ways of choosing such functions, see, for instance Zienkiewicz [49], Aziz [50], Carnahan et al. [51], Bathe and Wilson [52], and Cook [53].

Let the body forces be zero. Then the Airy Stress function, Φ, is defined as

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \partial^2\Phi/\partial y^2 \\ \partial^2\Phi/\partial x^2 \\ -\partial^2\Phi/\partial x\partial y \end{Bmatrix} \tag{4.5}$$

The two-dimensional equilibrium equations for zero body forces are, in matrix form, as follows:

$$\begin{Bmatrix} \partial/\partial x \\ \partial/\partial y \end{Bmatrix} \begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix} = 0 \tag{4.6}$$

Substituting Equations (4.1) into (4 2) shows that the Airy Stress function satisfies the equilibrium conditions.

The two-dimensional compatibility condition is as follows:

$$(\partial^2/\partial x^2 + \partial^2/\partial y^2)(\sigma_x + \sigma_y) = 0 \tag{4.7}$$

substituting Equation (4.5) into Equation (4.7) gives

$$\partial^4\Phi/\partial x^4 + 2\partial^4/\partial x^2\partial y^2 + \partial^4\Phi/\partial y^4 = 0 \tag{4.8}$$

Thus, any function, Φ, which satisfies Equation (4.8) also satisfies the compatibility and equilibrium conditions. Two-dimensional problems involving zero body forces thus reduce to solving a fourth-degree bi-harmonic equation as given in Equation (4.8) above. The Airy Stress function is thus a good basis for the construction of finite element equations and was used in this thesis along with 8−node isoparametric quadrilateral element as described later.

4.3 Local and Global Coordinates

In the context of finite element analysis, it is more convenient to use non-dimensional local coordinate system based on the element rather than use a global system. In finite element displacement methods, interpolation functions are used to relate the displacements at any point within the element to the nodal displacements. When the same interpolation functions are used for transformation of coordinates of the element, the element is called isoparametric. Figure 4.2 shows the global (x, y) coordinates and the local (s, t) coordinates of an 8−node quadrilateral element of arbitrary shape.

The global coordinates vary with the $x-$ and $y-$ values while the local coordinates vary between $\pm$ 1. Figure 4.3 shows the isoparametric mapping for an 8−node quadrilateral element of arbitrary shape.

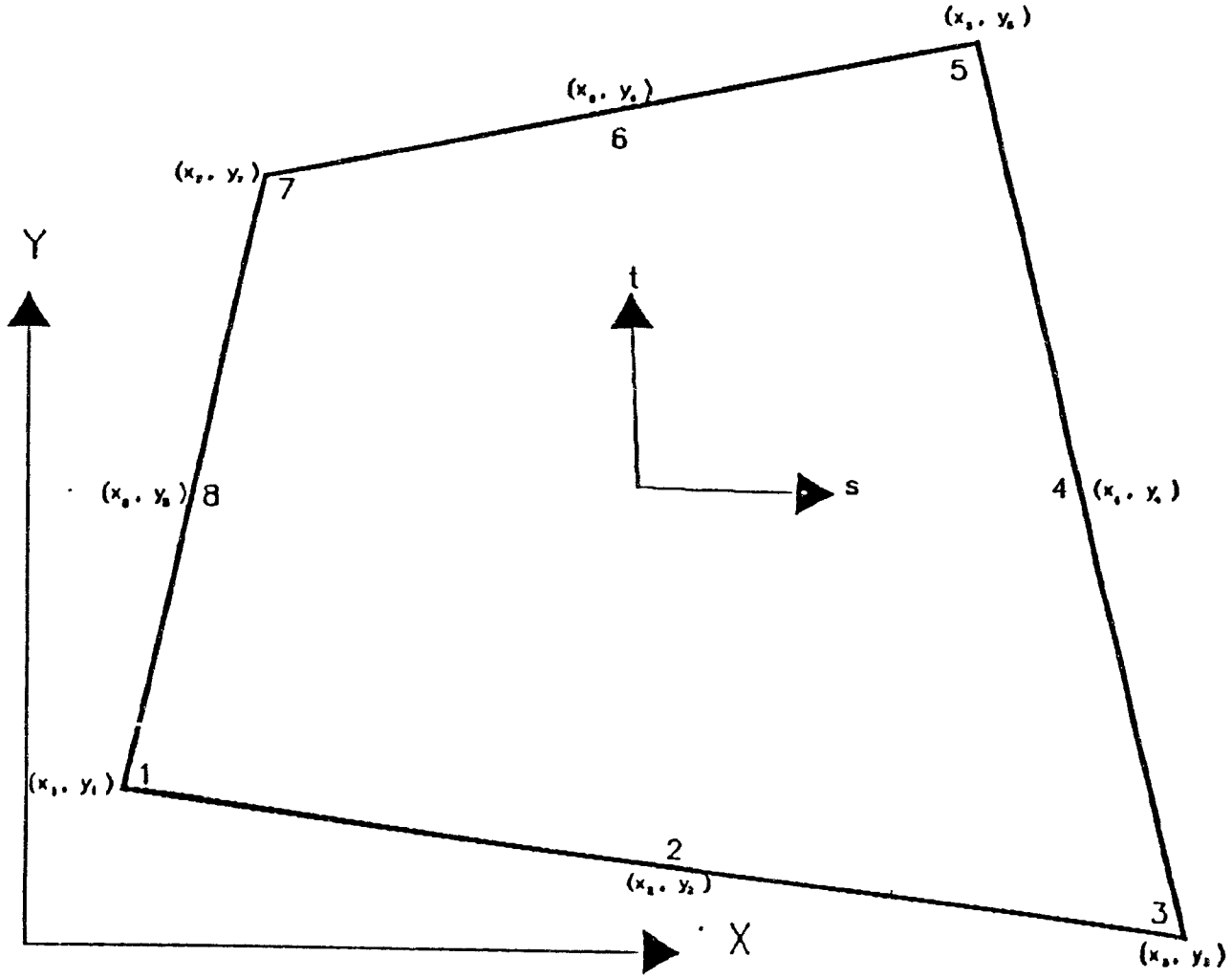


Figure 4.2: Global and Local Coordinate Systems

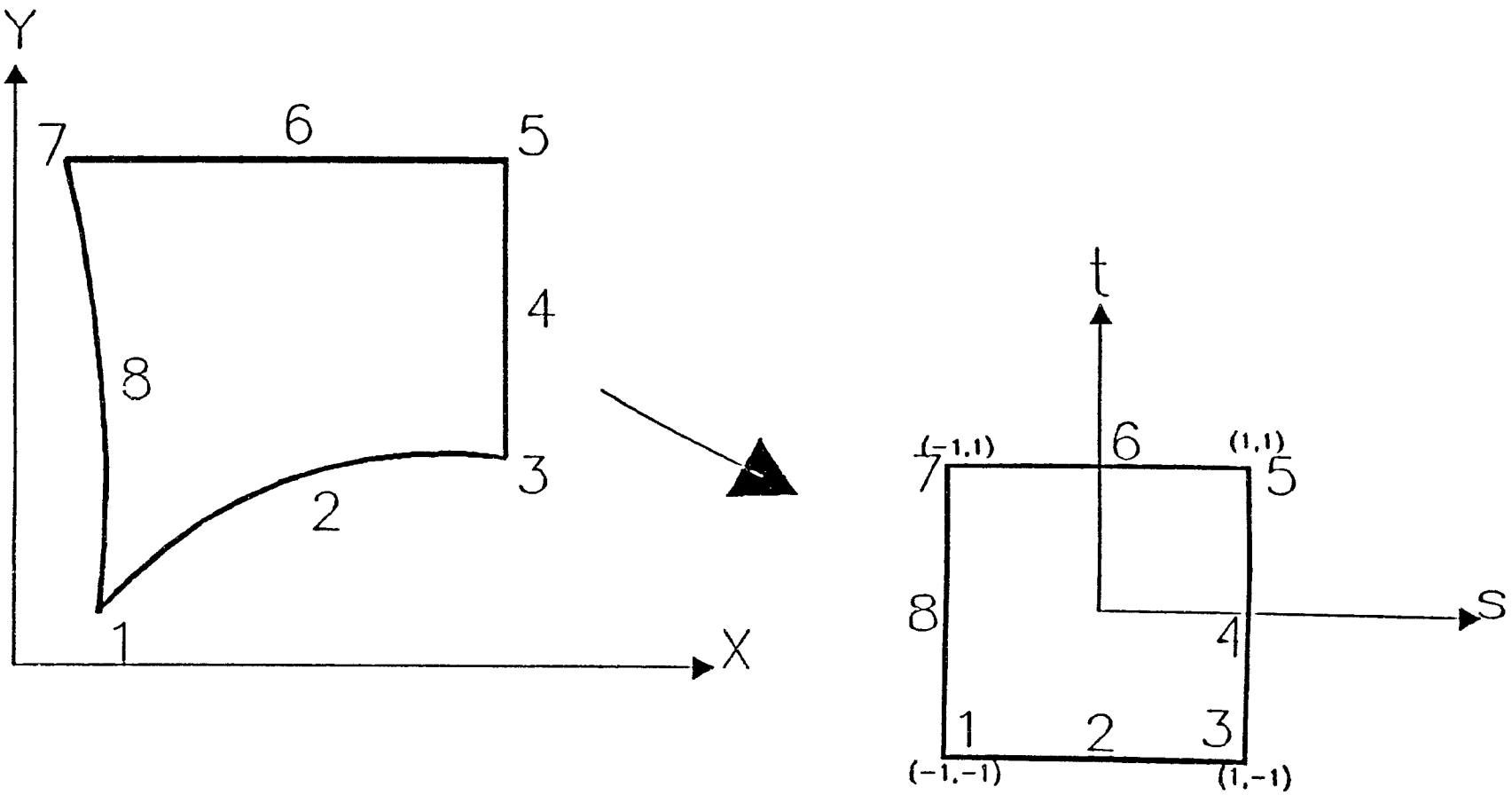


Figure 4.3: Mapping for 8-Node Isoparametric Element

The quadrilateral is mapped into the unit square. Details of the isoparametric concept appear in references [49] - [53], Akin [54], Press et al. [55], Korn and Korn [56], Abramowitz and Stegun [57].

A relation between the local and global coordinates for a $2-D$ element is expressed as

$$\left\{ \begin{array}{c} x \\ \\ y \end{array} \right\} = \left\{ \begin{array}{c} \sum N_i X_i \\ \\ \sum N_i Y_i \end{array} \right\} \tag{4.9}$$

where $n = 8$ for an 8-node quadrilateral element and N_i are the interpolation functions. There is a one-to-one correspondence between the global coordinates (x, y) and the local coordinates (s, t). By using the Chain Rule of differentiation, the following relation can be established between the local and global coordinates:

$$\left\{ \begin{array}{c} \partial/\partial s \\ \\ \partial/\partial t \end{array} \right\} = [J] \left\{ \begin{array}{c} \partial/\partial x \\ \\ \partial/\partial y \end{array} \right\} \tag{4.10}$$

where the Jacobian, $[J]$, is defines as

$$[J] = \left[\begin{array}{cc} \partial x/\partial s & \partial y/\partial s \\ \\ \partial x/\partial t & \partial y/\partial t \end{array} \right] = \left[\begin{array}{cc} \alpha_{11} & \alpha_{12} \\ \\ \alpha_{21} & \alpha_{22} \end{array} \right] \tag{4.11}$$

The interpolation functions used with the 8-node isoparametric element can be derived from the so called Serendipity elements and can be shown to be given as:

$$N_i(s,\ t) \ = \ (1 + \ s_i\ s)\ (1 + \ t_i\ t)\ (s_i\ s + \ t_i\ t - \ 1)\ /4 \tag{4.12}$$

for the corner nodes where i = 1, 3, 5, and 7. At midside nodes where i = 2, 4, 6 and 8, we have:

$$N_i(s,\ t) \ = \ s_i^2\ (1 - \ t^2)\ (1 + \ s_i\ s)\ /2 + \ t_i^2\ (1 - \ s^2\ )\ (1 + \ t_i\ t)\ /2 \tag{4.13}$$

The interpolation functions and their first and second derivatives are utilized in the formulation of the matrix equations. A complete listing is shown in Appendix E.

4.4 The Admissible Stress Function in terms of Local Coordinates

The Airy's Stress function Φ, can be expressed as follows:
$\Phi = f(s,t) = \Phi(x,y)$

We represent Φ as a series of terms involving the local coordinates and the undetermined stress multipliers, thus:

$$\beta_1 s^2 + \beta_3 st + \beta_2 t^2 +$$

$$\beta_4 s^3 + \beta_6 s^2 t + \beta_7 s t^2 + \beta_5 t^3 +$$

$$\beta_8 s^4 + \beta_{12} s^3 t + \beta_{11} s^2 t^2 + \beta_{10} s t^3 + \beta_9 t^4 +$$

$$\beta_{13} s^5 + \beta_{15} s^4 t + \beta_{17} s^3 t^2 + \beta_{18} s^2 t^3 + \beta_{16} s t^4 + \beta_{14} t^5 \qquad (4.14)$$

These 18 constants can be recognized as derived from the last four rows of Pascal's Triangle of order 6 which comprises 21 terms, with the first 2 rows missing. In Pascal's Triangle, the first row has 1 term, the second 2 terms etc. Thus a Pascal's Triangle of order 6 has $(1+2+3+4+5+6=)$ 21 terms. The missing terms are β, βs, and βt which would vanish even considering Equation (4.5) where the stress representation involves only second order differentiation of s and t. They are therefore not necessary. It can also be inferred that the highest terms to the power of 5 will result in a linear function in s and t on performing the fourth differentiation to satisfy the compatibility Equation (4.8).

By taking the second partial derivatives of Φ with respect to s and t, the [P] matrix can be represented as:

$$[P] = \frac{1}{\Delta^2} [S] \left([Z] - [F] [J]^{-1} [W] \right) \qquad (4.15)$$

where $\Delta = det|J|$
A full derivation of the [P] matrix is given in Appendix E.

A description of the hybrid computer program and the associated mesh generation and alteration programs is given in Chapter 5.

4.5 Consistent Load Vector due to Distributed Loads

Figure 4.4 illustrates an 8−node quadrilateral element loaded with uniformly varying loads on one side. The generalized forces and displacements for an 8−node quadrilateral element is shown in Figure 4.5

For an element side loaded with uniformly distributed loads w_x, w_y, the nodal force vector is given by the equation:

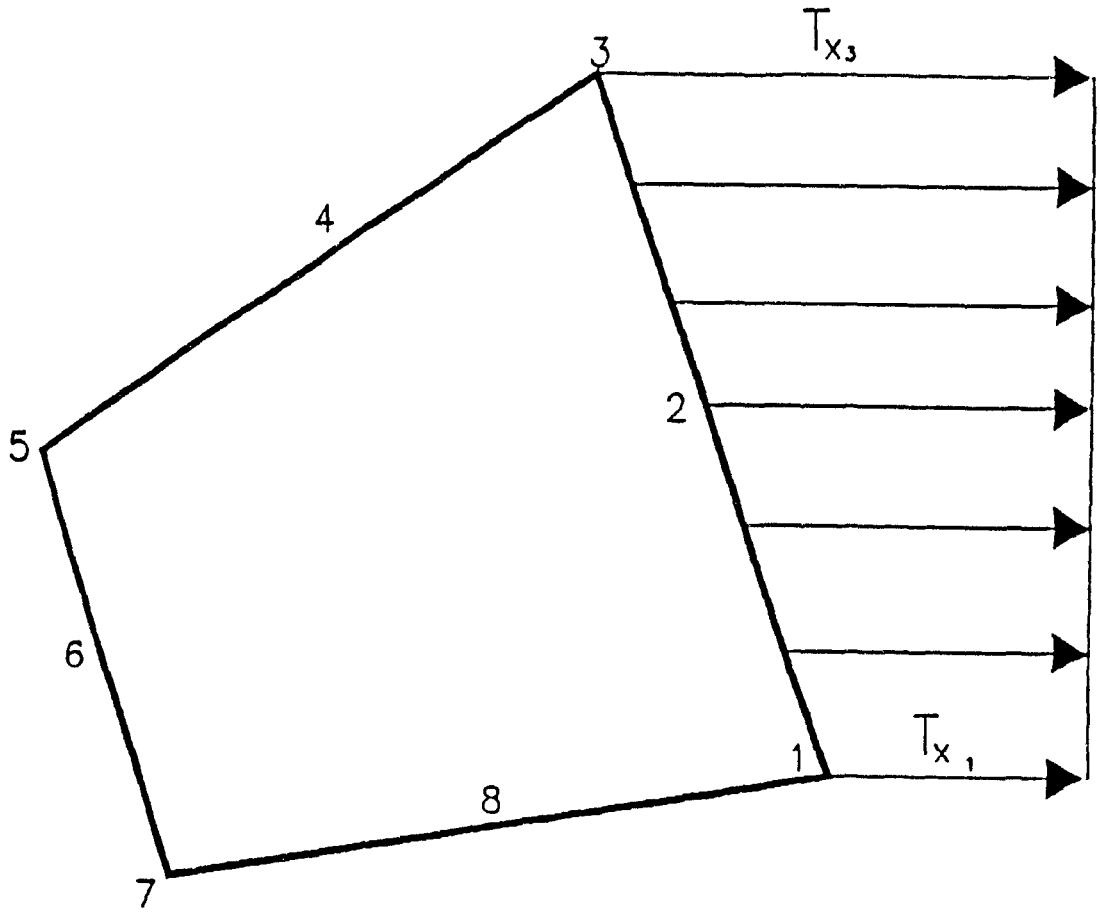


Figure 4.4: Uniformly Distributed Loads over an Element Side

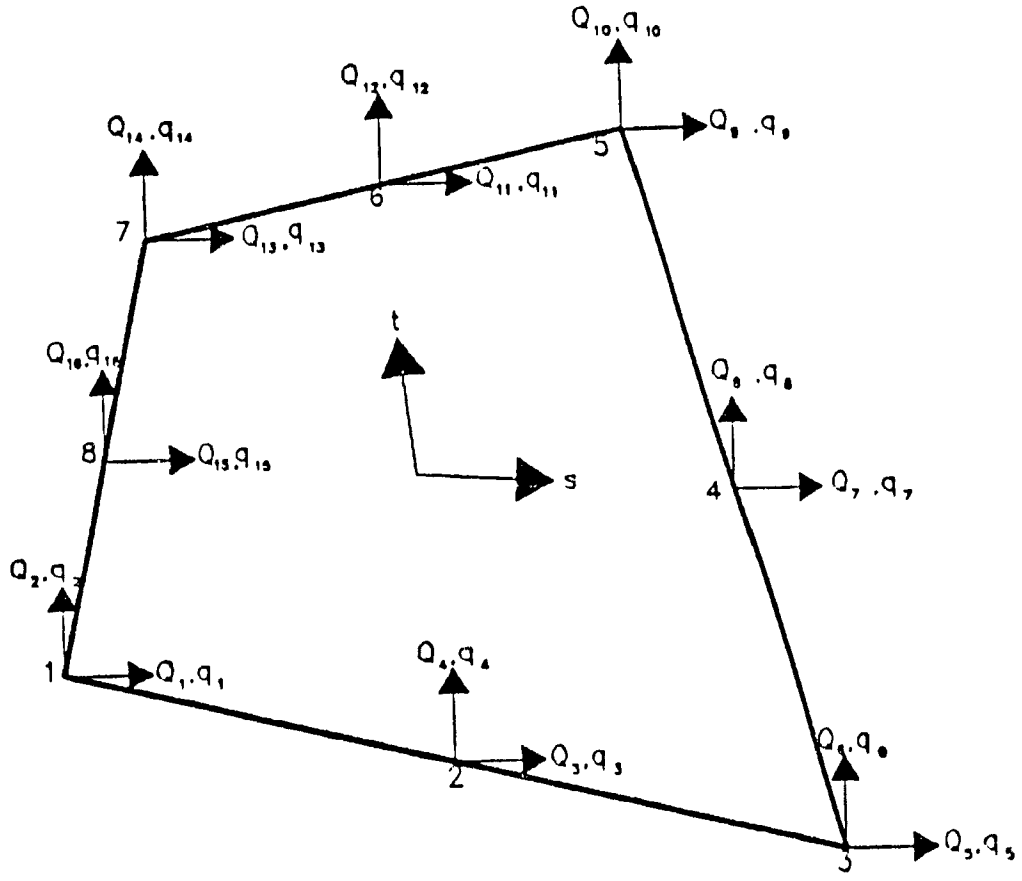


Figure 4.5: Generalized Forces (Q_i) and Displacements (q_i) for an 8-Node Quadrilateral Hybrid Stress Element

$$F^d = \int [N^b] \left\{ \begin{array}{c} w_x \\ \\ w_y \end{array} \right\} dr \tag{4.16}$$

where $[N^b]$ are the shape functions for that side, and dr is an element of length of the side. When local coordinates are used, one of them will be constant along any given side. In what follows, the side 1-2-3 of the element in Figure 4.4 will be used as example.

$$F^d = \left\{ \begin{array}{c} F_{x_1} \\ F_{y_1} \\ F_{x_2} \\ F_{y_2} \\ F_{x_3} \\ F_{y_3} \end{array} \right\} \qquad [N^b] = \left[\begin{array}{cc} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{array} \right] \tag{4.17}$$

dr is an element of length along the loaded side. Since, in general,
$r^2 = x^2 + y^2$, then
$dr = (x/r)dx + (y/r)dy = dx\cos\theta + dy\sin\theta$ By combining above expression for dr with the isoparametric representation of x and y, it can be shown that

$$dr = (11)[J] \left\{ \begin{array}{c} \cos\theta \\ \\ \sin\theta \end{array} \right\} \tag{4.18}$$

where $[J]$ is the Jacobian. The distributed loads, w_x, w_y can be linear or quadratic functions of the coordinates, thus,

$$\left\{ \begin{array}{c} w_x \\ \\ w_y \end{array} \right\} = [C] \left\{ \begin{array}{c} w_{x_1} \\ w_{y_1} \\ w_{x_3} \\ \\ w_{y_3} \end{array} \right\} \tag{4.19}$$

where,

$$[C] = \begin{bmatrix} C_1 & 0 & C_2 & 0 \\ 0 & C_1 & 0 & C_2 \end{bmatrix} \tag{4.20}$$

For linear representation,
$C_1 = (1-s)/2$,
$C_2 = (1+s)/2$,
and for quadratic representation,
$C_1 = (s^2+st)/2$,
$C_2 = (t^2-st)/2$.

It can be verified that at node 1, $w_x = w_1$, and at node 3, $w_x = w_3$ Similar expressions will hold for w_y. It is assumed that the values of the distributed loads are given at nodes 1 and 3 only. The integration in Equation (4.16) will be done numerically along the loaded side at a specified number of integration points.

4.6 Numerical Integration and the Gauss Quadrature Formula

In general, the integration required in the equations presented so far in this thesis involve matrices and higher order equations. The Gauss quadrature formula has proved to be an efficient and accurate method of integrating such equations in finite element analysis, see references [53] - [58]. In considering the evaluation of the definite integral, $I = \int_a^b f(x)dx$, by numerical integration, Gauss found that the number of points n, into which the interval (a,b) should be divided for the greatest accuracy should not be equally spaced but should be symmetrical with respect to the mid point of the interval.

Let $y = f(x)$. Define a change of variable

$$x(r) = \frac{1}{2}(b-a)r + \frac{1}{2}(b+a) \tag{4.21}$$

so that the non-dimensionalized limits of integration of r become -1 to $+1$. The new value of $y(r)$ is then given as

$$y = f(x) = f[\frac{1}{2}(b-a)r + \frac{1}{2}(b+a)] = \psi(r) \tag{4.22}$$

The original integral now becomes

$$I = \frac{1}{2}(b-a)\int_1^1 \psi(r)dr \tag{4.23}$$

Gauss showed that the integral in Equation (4.23) is given by

$$\int_1^1 \psi(r)dr = \sum_{i=1}^{i=n} W_i\psi(r_i) \tag{4.24}$$

where W_i and r_i represent tabulated values of the weight functions and the abscissae associated with the n points in the non-dimensional interval $(-1, 1)$. Thus, the final result of the integration is

$$I = \frac{1}{2}(b-a)\sum_{i=1}^{n} W_i\psi(r_i) \tag{4.25}$$

Gauss showed further that this equation will exactly integrate a polynomial of degree $(2n-1)$. In two dimensions, the quadrature formula for $\psi = \psi(\alpha, \beta)$ is found by first integrating with respect to α and then with respect to β, thus,

$$I = \int_{-1}^{1}\int_{-1}^{1} \psi(\alpha,\beta)d\alpha d\beta = \sum_{i=1}^{n}\sum_{j=1}^{n} W_iW_j\psi(\alpha_i,\beta_j) \tag{4.26}$$

The extension to three dimensions is obvious. Most of the numerical integration done in this thesis was carried out using the four point Gauss quadrature. Figure 4.6 illustrates the 4 x 4 Gauss integration points for an isoparametric quadrilateral element.

4.7 Steps in Isoparametric Hybrid Stress Formulation

Thus far, the relevant equations necessary for the computation of the necessary equations in Chapter 3 have been derived. On the element level, we have chosen a vector of functions - the Airy Stress functions expressed in local coordinates. These functions describe the generalized stress components at any point within a typical element.

The 8-node quadrilateral element has been chosen over the triangular varieties mainly because it enables higher order equations to be used in conformity with the stress function chosen.

Having formed the [P] matrix for the element, the [H] matrix Equation (3.8) can then be computed, using the [P] and the [D] matrices. If body forces are present, the $[H_b]$ matrix is also computed from $[P_b]$. Assembly of the [G] and the $[G_b]$ matrices follow the same procedure (Equations (3.10), (3.11)), such that the [R] and the $[R_b]$ matrices are derived from the [P] and the $[P_b]$ matrices respectively.

The next important component in the representation of the element stiffness matrix is the nodal force vector, expressed by Equation (3.12)The assembly of the nodal force vector is achieved by considering the imposed boundary conditions along with the boundary loads

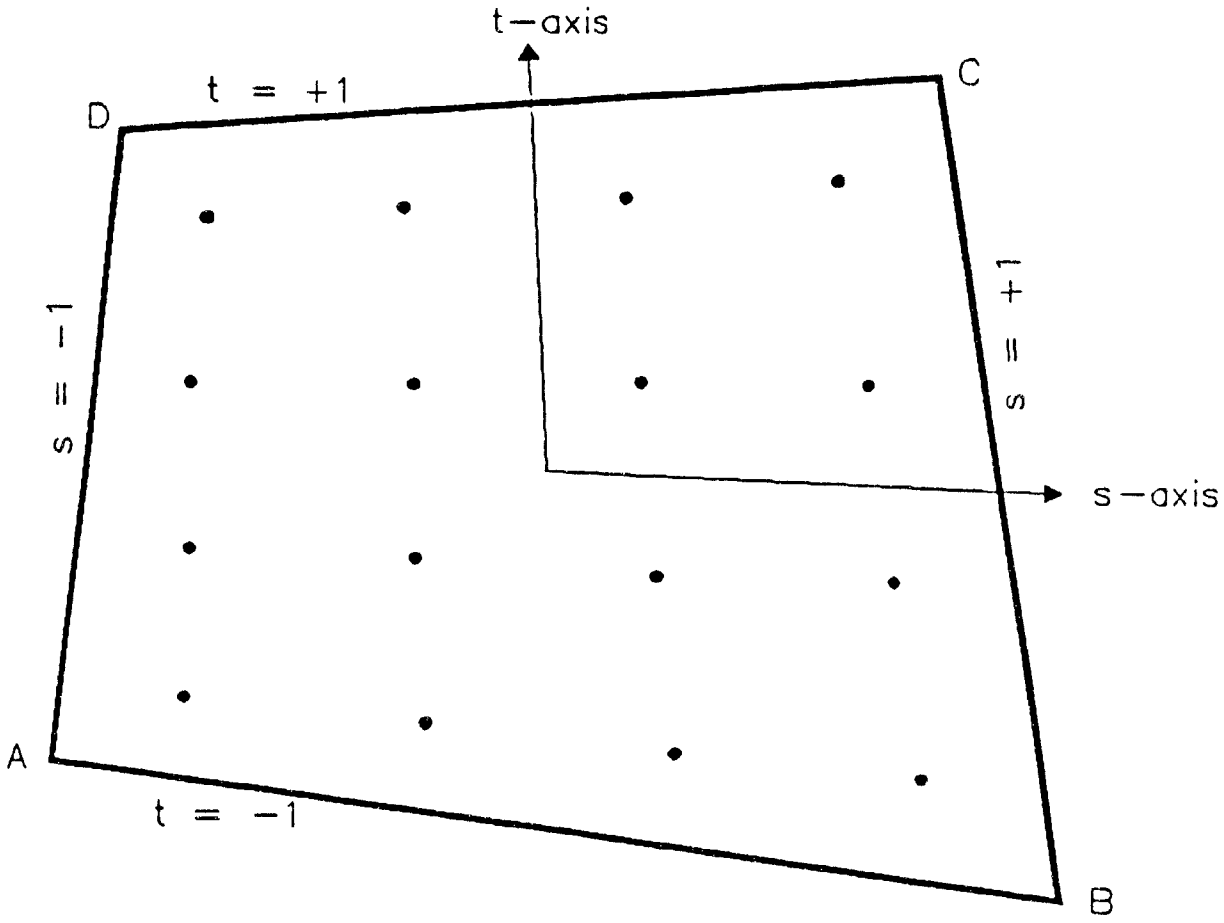


Figure 4.6: Numerical Integration Points for anIsoparametric Quadriiateral Element

acting on the system.

The stiffness matrix is assembled for the element from a knowledge of the [H] and the [G] matrices. Once the stiffness matrix and the load vector for the element have been assembled, they are transferred to the global stiffness matrix and the global load vector respectively. The solution for the displacements at the nodes then follows routinely from the equation:

$$\{Q\} = [K]\{X\} \tag{4.27}$$

where the vector $\{X\}$ represents the degree of freedom (d.o.f) displacements. The versatility of isoparametric formulation lies in its capability of expressing the parameters of interest as functions of nodal parameters. However, it is rare to have a finite element mesh where all the element sides are parallel to the global axes. With the Airy's stress function, Φ, expressed as polynomials in s and t local coordinates, its second derivatives with respect to x and y global axes cannot in general, be constant over the element. The ideal is to express Φ as polynomials in local orthogonal cartesian coordinate system, ϵ and η, unlike the s and t which are in general not orthogonal.

The axes of the local orthogonal system have been obtained by Ahmad and Irons [39] by firstly calculating the Jacobian at the centroid of the element and then performing the iterating scissors on the $[J]^T$ matrix to get the best orthogonal approximation. Consider a quadrilateral element with local axes (s,t) and the point O as centre, as shown in Figure 4.7.

Let ϵ and η axes represent an orthogonal system also centred at O, and let P be any point in the element with coordinates P(x,y) in the global system, P(s,t) in the local (s,t) system and $P(\epsilon,\eta)$ in the local (ϵ,η) system. In matrix form, the local coordinate transformation from the (s,t) system to the (ϵ,η) system is given by:

$$\begin{Bmatrix} \epsilon \\ \eta \end{Bmatrix} = \begin{bmatrix} \cos\theta_s & -\sin\theta_t \\ \sin\theta_s & \cos\theta_t \end{bmatrix} \begin{Bmatrix} s \\ t \end{Bmatrix} \tag{4.28}$$

Let

$$[J_{st}] = \begin{bmatrix} \partial x/\partial s & \partial y/\partial s \\ \partial x/\partial t & \partial y/\partial t \end{bmatrix} \tag{4.29}$$

$$[J_{\epsilon_n}] = \begin{bmatrix} \partial x/\partial \epsilon & \partial y/\partial \epsilon \\ \partial x/\partial \eta & \partial y/\partial \eta \end{bmatrix} \tag{4.30}$$

be the Jacobian in the (s,t) and the $(\epsilon-\eta)$ systems respectively. Then, it can be shown that

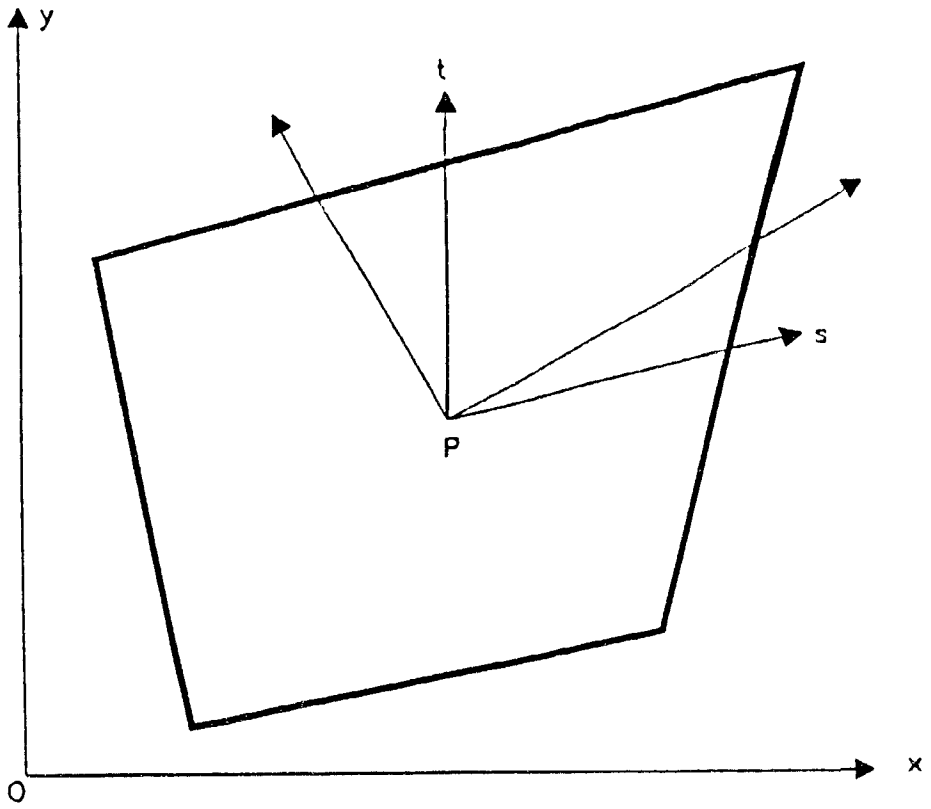


Figure 4.7: Mean Axes System for Stress Function

$$[J_{st}] = \begin{bmatrix} \partial\epsilon/\partial s & \partial\eta/\partial s \\ \partial\epsilon/\partial t & \partial\eta/\partial t \end{bmatrix} \tag{4.31}$$

and from Equation (4.28),

$$\begin{bmatrix} \partial\epsilon/\partial s & \partial\eta/\partial s \\ \partial\epsilon/\partial t & \partial\eta/\partial t \end{bmatrix} = \begin{bmatrix} \cos\theta_s & \sin\theta_s \\ -\sin\theta_t & \cos\theta_t \end{bmatrix} \tag{4.32}$$

The required local coordinate axes are obtained from Equation (4.31) as follows·

$$[J_{\epsilon,\eta}] = \begin{bmatrix} \cos\theta_t & -\sin\theta_s \\ -\sin\theta_t & \cos\theta_s \end{bmatrix} \tag{4.33}$$

The mean axes are generally expressed as the transpose of the $[J_{\epsilon,\eta}]$ matrix. One obvious way of defining the (ϵ, η) axes is to assume they are parallel to the global system. Then the (s,t) system is coincident to the (ϵ, η) system and the case is trivial.

4.8 Computer Implementation

The following are the steps in the solution by Finite element methods of a problem:

(a) Define the problem and define its physical geometry.

(b) Generate a (quadrilateral) mesh and assign nodal and element topological properties.

(c) Assemble the stiffness matrix for each element and transfer into global stiffness array.

(d) Assemble the Load vector in a similar manner to the stiffness matrix.

(e) Solve the resulting algebraic equations for the nodal displacements.

(f) Solve for nodal stresses or stresses within any element.

Of these six steps, the mesh generation (step b) and the solution for nodal stresses (step e) presented the greatest challenge apart from writing the computer code of the hybrid stress finite element method.

4.8.1 Mesh Generation Codes

It was apparent early in the research that a good and accurate mesh was highly important for the correct solution of a problem by the finite element method. It is only the simplest of meshes that can be generated by hand and even then, a thorough check is necessary. The author spent much effort in developing two mesh generation programs that work on the same principles, one specifically for generating quadrilateral meshes over regions surrounding circular holes and the other as a more general quadrilateral mesh generator over quadrilateral or nearly quadrilateral sections. Both programs require the minimum of input as described in Chapter 5.

Along with the more general mesh generator, a small program that enables one to create holes or empty spaces in an already existing mesh was also written. The usefulness of this program becomes apparent when one considers a very common method of mining deep ore bodies, i.e., open stoping or sequential excavation. Ordinarily, each new sequence requires an alteration to a previous mesh. By defining the elements and nodes that are to be eliminated in the next sequence, the companion program goes ahead and re-establishes a new mesh in which the zone formerly occupied by the elements and nodes is now void. The load vector is also automatically altered to reflect the new reality.

4.8.2 Solution of the Stiffness Equations

The assembly of the global stiffness matrix and load vector leads to a system of linear algebraic equations of the form

$$[K]\{X\} = \{Y\} \tag{1.34}$$

where $[K]$ is the global stiffness matrix, $\{X\}$ is the vector of unknown nodal displacements and $\{Y\}$ is the global load vector. These equations are often quite large, of the order of thousands in most cases of moderate to large problems, and the solution certainly calls for the use of the computer. It is known that from 30 to 50% of the computer time involved in a linear finite element analysis is associated with solving simultaneous equations. In non-linear and dynamic analysis, as much as 80% of the computer time is used for the same purpose.

However, certain properties of the square symmetric matrix in all finite element computations permit the use of a number of techniques to reduce not only the solution time, but the amount of computer storage required to perform the analysis. Fellipa [58], Mondkar and Powell [59], Meyer [60], George [61], Hood [62], Hinton and Owen [63], and Irons [64] among others, have written about methods of solving these types of equations.

For the hybrid stress finite element method, the $[K]$ matrix is symmetric, semi-positive definite, sparse and banded. Symmetry allows one to economize on computer storage since

only the elements in the upper or lower triangle need be stored. Because the matrix is positive definite, pivoting is never required in order to ensure a stable solution. Banding depends on the way the node numbers are assigned and the efficiency with which the the numbering was done. Since the solution time varies as the square of the bandwidth, the nodal numbering producing the minimum bandwidth results in the most economical solution. There is little doubt that the mesh generation programs described above result in the best banding of the $[K]$ matrix. The two methods that were used in this thesis are described below:

The Skyline Method

The Skyline solution method, also referred to as the profile, envelope, or variable bandwidth method first appealed to the author by its elegance. The symmetric and sparse $[K]$ matrix is stored in a linear array as a string of 'active' columns of the upper triangle. The active portion of each column is bounded by the diagonal and the furthest non-zero element. A second array of pointers holds the position of the diagonal elements in the active array and the algorithm used for solving the equations is the so-called modified Cholesky algorithm which decomposes the $[K]$ matrix into a lower triangular matrix $[L]$, and a diagonal matrix $[D]$ thus,

$$[K] = [L][D][L]^T \tag{4.35}$$

The factorization replaces elements of $[K]$ by elements of $[L]$ and $[D]$ matrices. Once the $[K]$ matrix has been factorized as above, the program can be invoked to solve Equation (4.34) for any right hand side vector Y. The solution vector is obtained in a three-stage process thus:

$[L]\{z\} = \{Y\}$ (forward substitution)
$\{b\} = [D]^{-1}\{z\}$ (scaling)
$[L]^T\{X\} = \{b\}$ (back substitution)

A working version of the program was written and tried on several problems. It was found that the size of the problem affected the time necessary to effect a solution, probably because of the large data handling overhead (addressing, fetching, storing, etc.). The program however, is quite accurate for solving linear simultaneous equations. Most of the problems reported in this thesis were solved by this method.

The Frontal Solution Scheme

This method, first introduced by Irons, [64], uses the Gauss Elimination and back substitution method to solve Equation (4.21). In doing so, however, it first assembles the element stiffness matrix and nodal loads into a global stiffness matrix and load vector. Its main feature over other methods is the delayed introduction of variables and their accelerated elimination. The active life of a node lasts from the time in which it first appears in an

element to the time it last appears in an element.

Thus, the ordering of the nodes is not as crucial as the ordering of the elements, an opposite requirement to that for banded solution schemes. The core storage requirements are at most the same as those for banded methods. The operation on zero coefficients are minimal and the total arithmetic operations are fewer than with other methods. On the other hand, an elaborate housekeeping system is required for the frontal method. A working program was amended from the frontal subroutine in the reference [64]. It was also found to be quite accurate in solving simultaneous equations. It is definitely faster than the Skyline method.

4.8.3 Computer Programs

A description of the hybrid computer program and the associated mesh generation and alteration programs is given in Chapter 5.

Chapter 5

Hybrid Stress Computer Programmes

Four programs are described in this chapter. Section 5.1 describes the hybrid stress computer program written according to the mathematical developments in the preceding chapters. The other three programs were written with the special need to generate 8-node quadrilateral meshes over mining excavations and for altering the mesh configurations.

5.1 The Hybrid Stress Computer Code

5.1.1 General

Figure 5.1 is a schematic flow chart of the hybrid stress computer code. It is written in FORTRAN 77 and can be described in terms of the major subdivisions into which the theory of the hybrid stress formulation falls. These subdivisions are, in order of occurrence, discretization of the continuum into finite elements, data preparation, assembly of the global stiffness matrix and the global load vector, solution for nodal displacements, and finally, solution for element nodal stresses. Mesh generation or the discretization of the continuum into finite elements is described in Section 5.2. A brief description of the main subroutines in Figure 5.1 follows. The computer code is given in Appendix A. The input file to the hybrid stress finite element program is structured as follows.

(a) Title line;

(b) Problem control data (1 line);

(c) Material properties data lines;

(d) Nodal coordinates and constraints lines;

(e) Element connectivity and element type lines;

(f) Distributed loads lines.

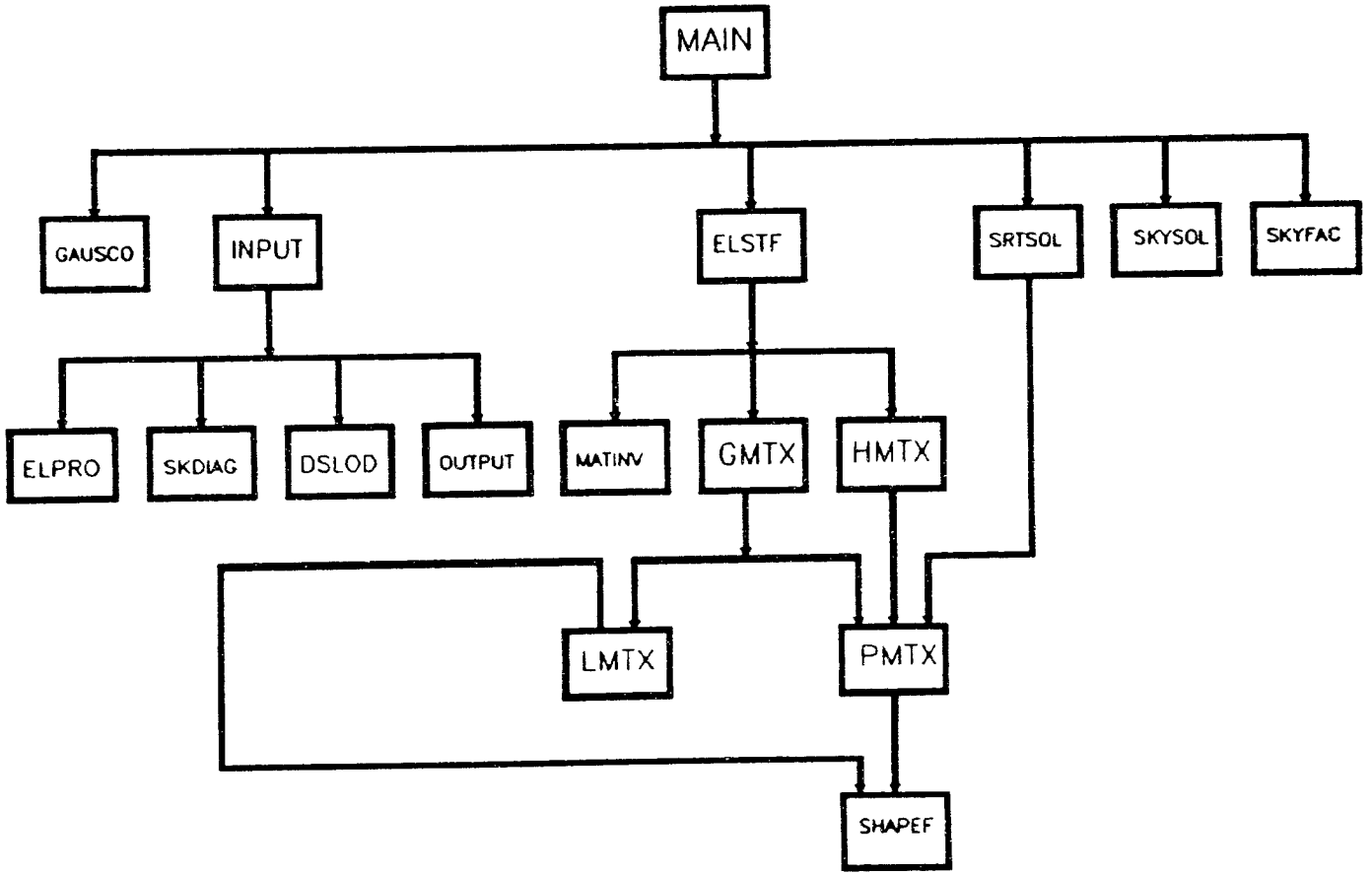


Figure 5.1: Schematic Flowchart of the Hybrid Stress Computer Programme

The input file is prepared by the mesh generation program (see Section 5.3) and is held in a designated file which must be present in the system for the program to work. Below is a description of the data file. It is freely formatted.

(a) Line 1: Problem Title Line, format 80A1

(b) Line 2: Problem specification data line.
READ NNODE NNEL NUMAT NTYPE NPOIN NDIST IPRNT THIC

NNODE =	Total number of nodes
NNEL =	Total number of elements
NUMAT =	Total number of material types
NTYPE =	Problem type parameter,
=	1 for plane stress
=	2 for plane strain
NPOIN =	Nodal point load parameter
=	0 for no point loads present, otherwise,
=	Total number of nodes with point loads
NDIST =	Distributed load parameter
=	0 for no distributed loads present, otherwise,
=	Total number of nodes with distributed loads
IPRNT =	Print output parameter
=	0, do not print output
=	1, print output
THIC =	Element thickness, assumed constant.

(c) NUMAT Lines: Element material properties line(s)
For each property type, READ PROPS(1) PROPS(2) PROPS(3)

PROPS(1) =	Young's modulus
PROPS(2) =	Poisson's ratio
PROPS(3) =	Density

(d) NNODE Lines: Nodal coordinate and constraint data.
These lines define the structural geometry of the discretized zone.
For each node, read N, XORD(N), YORD(N), JDOX(N), JDOY(N)

N = Node number
XORD(N) = x-coordinate
YORD(N) = y-coordinate
JDOX(N) = degree of freedom in x-direction
JDOY(N) = degree of freedom in y-direction
JDOX(N) = 0 for unconstrained displacement in x-direction
= 1 for constrained displacement in x-direction

(e) NNEL Lines: Element type and connectivity data.
For each element, read NUM, NELTYP, (NLM(J),J= 1,8)

NUM = Element number
NELTYP = Element type
NLM(J) = Jth. node
Numbering of elements is counterclockwise

(f) NPOIN Lines: Point load data lines.
If NPOIN is positive then , for each node under a point load,
read KNODE XFORCE YFORCE

KNODE = Node number
XFORCE = Point load in x-direction
YFORCE = Point load in y-direction

(g) NDIST Lines: Distributed load data lines.
Distributed loads are applied to some of the boundary element sides. If NDIST is positive, then follows NDIST lines of data of type:

NODE1, NODE2, NODE3, ELNUM, XFORCE1, YFORCE1, XFORCE3, YFORCE3
where

NODE1 = node number of first node of element side
NODE2 = node number of second node of element side
NODE3 = node number of third node of element side
ELNUM = element number
XFORCE1 = load in x-direction at NODE1
YFORCE1 = load in y-direction at NODE1
XFORCE3 = load in x-direction at NODE3
YFORCE3 = load in y-direction at NODE3

5.1.2 Subroutines for Massaging Data Input

GAUSCO is a small subroutine containing the 4-point Gauss quadrature data. It is called by the main program at the start of computation and is held in memory throughout the execution period.

The subroutine, INPUT, reads the input data file. The mesh generation program described later is capable of generating the entire data for a given problem from a minimum input information. The INPUT subroutine calls four other subroutines designed to complete the data preparation section.

SKDIAG is a subroutine to compute the diagonal element positions in the Skyline vector which stores the upper half of the stiffness matrix. This method of storage excludes all leading zero elements starting from the first row of each column. DSLOD is a subroutine for computing the applied load vector, {Q} of Equation (3.11). The subroutine, ELPRO computes the compliance matrix for all the different materials and stores them in a linear array termed CAR. In general, the compliance matrix is different for different material type. Assuming linear elastic behaviour of all materials, the matrix is symmetric and is described by three constants. More description is given in the subroutine in Appendix A.

OUTPUT is a subroutine which, at the user's choice, will direct all the input data to a terminal printer. It is called by the INPUT subroutine.

5.1.3 Subroutines for Assembling Global Stiffness Matrix

The subroutine ELSTF is called for each element by the main program for the purpose of computing the element stiffness matrix and summing the upper half of it into the global stiffness vector held in the skyline mode. The two important matrices involved are the [H] and [G] matrices of Equations (3.8) and (3.10) respectively. These two matrices are computed in

the subroutines HMTX and GMTX respectively. Computation is by numerical integration using the 4 point Gauss quadrature. Each of these two subroutines calls the PMTX subroutine which in turn calls the shape function subroutine, SHAPEF. The PMTX subroutine computes the [P] matrix and the SHAPEF subroutine contains the relevant equations of the 8-node isoparametric shape functions and their derivatives. LMTX contains a subset of SHAPEF subroutine values computed along the element sides for which one of the local variables is usually $\pm$ 1. It is called by those subroutines that compute values along element sides, namely, DSLOD and GMTX subroutines

The MATINV subroutine is used exclusively to invert the 18x18 square symmetric [H] matrix because it is in that form that this matrix is used in subsequent equations.

5.1.4 Subroutines for Solution of Nodal Displacements and Stresses

The two subroutines, SKYFAC and SKYSOL were adapted from the solution code by Fellipa [58]. SKYFAC factorizes the global stiffness matrix and SKYSOL solves Equation (3.23) for {q}, the nodal displacements. The displacement vector replaces the load vector and the array is then passed to the subroutine STRSOL by the main program. This subroutine solves for the stress components at any given point within the element. In the version of the program reported in this thesis, these stresses are computed at the corner nodes, midside nodes and the centre of each element.

5.2 Program for Meshes over Circular Holes, CIRCMESH

The program takes advantage of the fact that when circular holes are involved in finite element analysis, the continuum is often assumed to be uniform with respect to geometric and material properties. The resulting symmetry about the axis of the hole means that only a quadrant of the continuum need be analyzed. The input into the program consists of five lines described below.

Line 1: Title line
Line 2: NNY, NANG, R0, BM
Line 3: ((JXL(I,J),J= 1,2),I= 1,4)
Line 4: XL,YB,XR,YT,TH
Line 5: ASXX, ASYY

NNY = Number of element sides along x or y axis
R0 = Borehole radius
BM = y (or x) boundary magnification factor
NANG = Number of angular subdivisions of first quadrant (must be even)
JXL(I,J) = Condition code for Ith. Boundary in J-direction
= 0, means Ith. boundary can move in J-direction
= 1, means Ith. boundary cannot move in J-direction
J = 1, means x-direction
J = 2, means y-direction

In general, the left and right boundaries are parallel to the y-axis while the bottom and top boundaries are parallel to the x-axis.

XL = x-coordinate of left boundary (parallel to y-axis)
YB = y-coordinate of bottom boundary (parallel to x-axis)
XR = x-coordinate of right boundary (parallel to y-axis)
YT = y-coordinate of top boundary (parallel to x-axis)
ASXX = applied stress in x-direction
ASYY = applied stress in y-direction

The number of angular subdivision is kept even for the purpose of symmetry, since the program only has to generate the mesh over 45o arc and then reflect it over the 45o line to complete the generation. Unlike other mesh generation programs, the program first calculates the length of each radial line from the periphery of the borehole to the outer boundary, see Figure 5.2.

The number of corner nodes on any radial line is constant and equal to NNY+1. These nodes are placed on the radial line in such a way as to increase the density of elements near the borehole and create a coarse mesh farther away from it.

If r_1, r_2, r_3,.....r_n are the radial lengths from the origin to the first, second, third, etc corner node along any given line, the following algorithm is used to compute these lengths: $\Omega\, r_1 = r_0$, the radius of the hole, and,

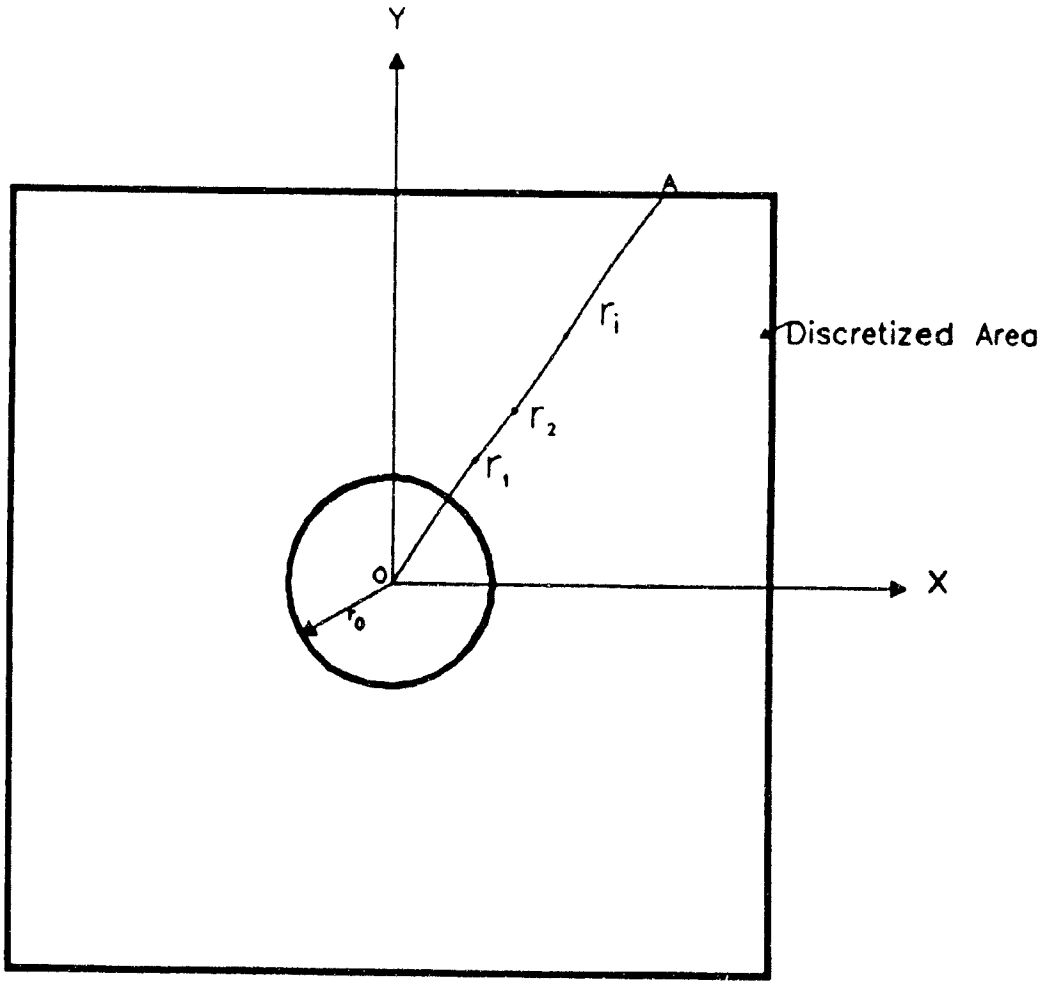


Figure 5.2: Coordinates of Corner Nodes in F.E. Mesh over Circular Holes

$r_i = r_{i-1} + r_0 a_{i-1}, \; i = 2, ..., NNY + 1$

The purpose is to compute the constant, a, from the knowledge of r_0 and the length of the radial line. Once it is computed, all the r_i's are then calculated from the recurrence relation above. It can be seen that r_i is greater that r_{i-1} so that smaller elements are generated near the hole and larger ones farther away.

For any given r_i, then,
x-coordinate of corner node $= r_i \cos\theta$,
y-coordinate of corner node $= r_i \sin\theta$

where θ is the angle between the horizontal x-axis and the radial line. Symmetry about the 45° line is utilized in both nodal coordinate generation and element node numbering. The midside nodes are computed at the middle points of the element sides.

5.3 A General Quadrilateral Mesh Generation Program, QUADMESH

This program will generate an 8-node quadrilateral mesh over an array of vertical lines whose lower end coordinates are specified. These end coordinates need not necessarily lie on the x-axis (assumed horizontal). The limits of the program are.

maximum number of nodes	1000
maximum number of elements	625
maximum number of sides with specified loads	125
maximum number of nodes with point loads	500
maximum number of element property zones	20
maximum number of different properties	10

The program essentially recognizes that straight-sided quadrilateral elements can be generated between two vertical lines or edges by drawing lines to join appropriate corner nodes. This recognition is illustrated in Figure 5.3.

In each of the illustrations, elements to the left of the diagram have been generated so that the nodal points on the left edge are already established. The left edge of the column is termed the front edge. The diagrams contain all the possible types of quadrilateral elements the program can generate. The continuum to be discretized is assumed to lie in the first quadrant. A concise description of the method of generating the quadrilateral elements follows.

(a) the program reads the specification of the first vertical line which may be, say, the

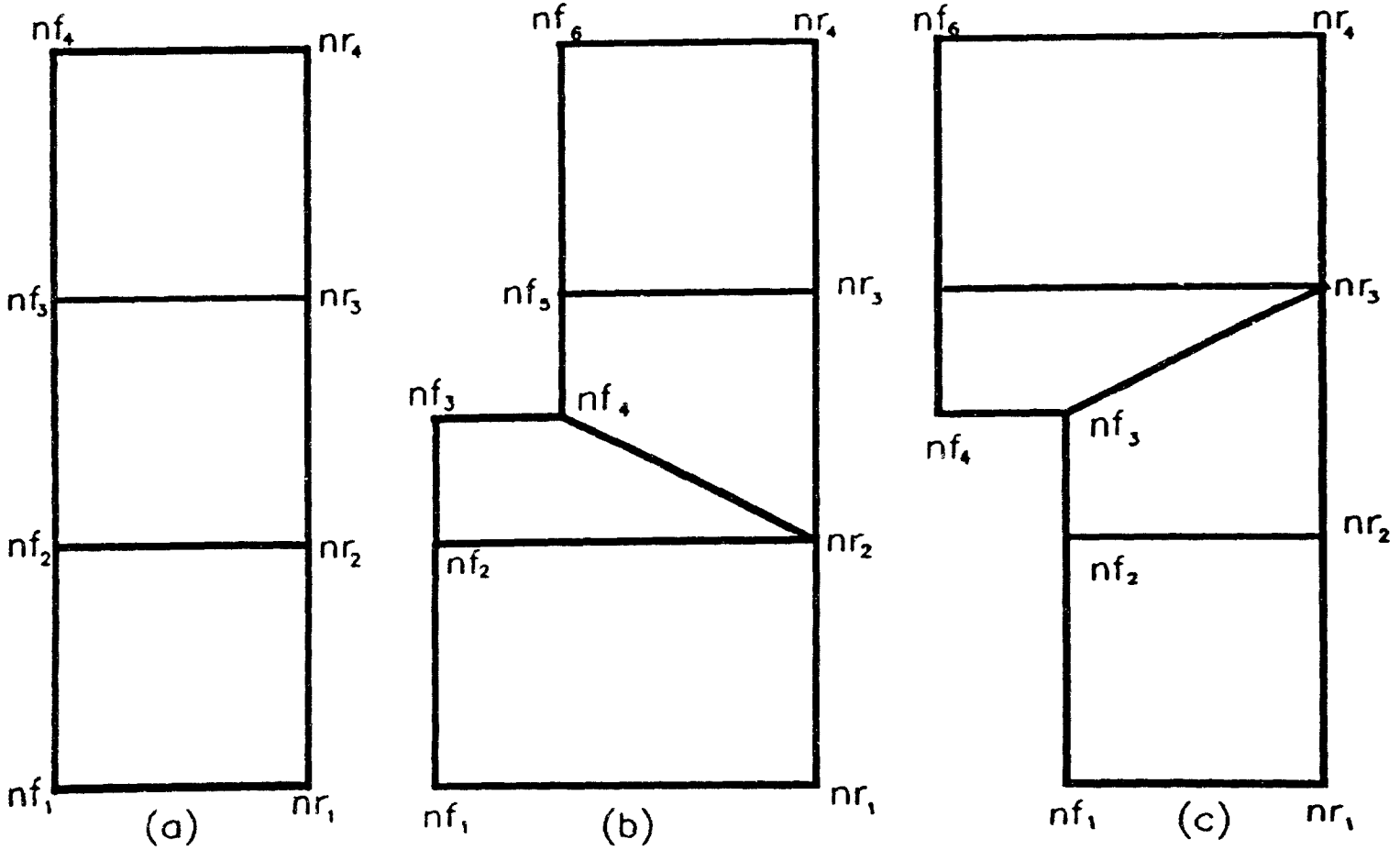


Figure 5.3: Types of Quadrilateral Elements Generated by the QUADMESH Program

y-axis. Since this is the first line read, the program generates the nodes along this line and tags the line as the front edge.

(b) the program then reads the specification of the next line and tags it the right edge. It then locates the first node on the front edge, n_{f_1}, which is not above the first node on the right edge, n_{r_1}, and the first node on the front edge, n_{f_1}, which is not below the last node on the right edge, n_{r_1}. Obviously, a line joining n_{f_1} to n_{r_1} is the base of the quadrilateral and is in a counter clockwise sense.

(c) the program then examines the next three corner nodes on the front edge to determine whether a turn through 90o has occurred. There are three possible turns. First, there could be no turn through 90o. Then, there may be a clockwise turn to the right followed immediately by an upward counter-clockwise turn, or lastly, there may be a counter-clockwise turn followed immediately by an upward clockwise turn.

(d) If there has been no turn, this gives rise to a quadrilateral element with corner nodes n_{f_1}, n_{r_1}, n_{r_2}, n_{f_2}. The base of the next element is tagged n_{f_2} - n_{r_2}, see Figure 5.3 (a).

(e) If there has been a clockwise turn through 90o followed by an upward turn, two quadrilateral elements are generated as shown in Figure 5.3 (b) and the base of the next element to be generated is tagged n_{f_3} - n_{r_3}.

(f) If there has been a counter-clockwise turn through 90o followed by an upward turn, two quadrilateral elements are generated as shown in Figure 5.3 (c) and the base of the next element to be generated is tagged n_{f_3} - n_{r_3}.

When all the elements in the column are generated, the front edge is then redefined to include only those nodes which are farthest away from the y-axis and which daylight to the right edge of the continuum. Examples of the meshes that can be generated by this program are shown in Figures 5.4 and 5.5.

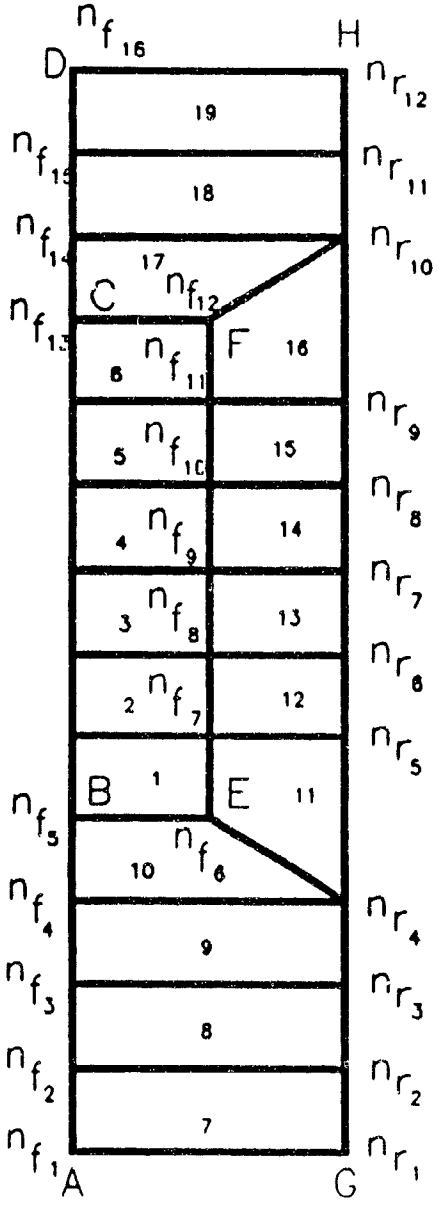


Figure 5.4
Example of a Simple Mesh
Generated by QUADMESH Program
with 3 Vertical Line Input, AD, EF, & GH

Note progression of front:
1st front: AD
2nd front: ABEFCD
3rd front: GH

Elements are numbered in the order
in which they are generated

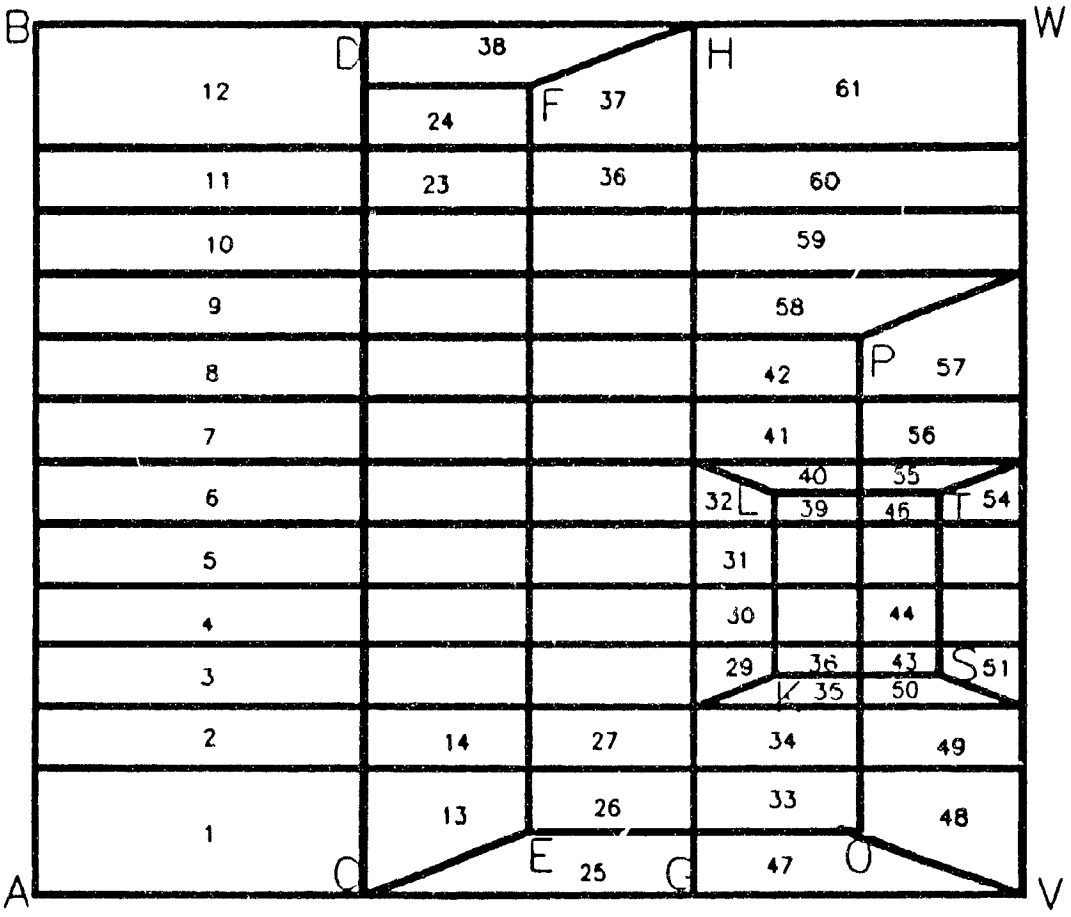


Figure 5.5: Example of a Complex Mesh Generated by QUADMESH Program with 8 Vertical Line Input

5.3.1 Input Data to QUADMESH

The input data into QUADMESH is free formatted and is described below.

1. Line 1: X_0, Y_0, NSUBX, NSUBY
2. Next line/s: If NSUBX ≥ 1, : DELX(1), DELX(2),.., DELX(NSUBX)
3. Next line/s: If NSUBY ≥ 1, : DELY(1), DELY(2),.., DELY(NSUBY)
4. Next line: 0.0 0.0 0 0

Steps 1, 2 and 3 above are repeated for all vertical lines or sets of vertical lines. Step 4 ends the ends the vertical line information necessary for mesh generation only. The program is also able to generate material properties, material types and distributed load information with a few more lines of input data as described below.

Next line: NPOIN, NDIST, NUMAT, NTYPE, YDLL, YDLR, XDLT, XDLB
Next NPOIN lines: NPO(I), XPO(I), YPO(I)
For each material type (I = 1, ...,NUMAT),
First line: YM(I), PR(I), MTYPE(I)
Second line: ZXL(I), ZXR(I), ZYB(I), ZYT(I)
If there is only one material type, then,
Second line: 0.0 0.0 0 0 0.0
Displacement Boundary Condition Codes:
JXL, JYL, JXB, JYB, JXR, JYR, JXT, JYT
This is the end of the input data.

5.4 A Program for Creating Voids in Discretized Media, VOIDMESH

Creating voids in a discretized medium is often necessary when simulating excavation sequences in a mine. The process involves deleting elements which represent the void, and the nodes common to them. A program was written for this purpose, to be used in conjunction with the mesh generation programs described above. Briefly, the initial mesh is generated to cover the entire medium with elements. Zones occupied by shafts, tunnels, and stopes are known and their boundaries are stored in a file which also holds information of the excavation sequence.

Creating an excavation will affect items (d), (e) and (f) above. The input file to the VOIDMESH program consists of the elements to be deleted and the affected nodes for each excavation. These are arranged in sequence. Creating the void essentially means ignoring the affected elements and nodes and updating the rest of the mesh data. The distributed loads are usually applied along the boundaries of the continuum and updating the mesh will

not affect their magnitudes, only the node numbering of the affected element sides. This program is therefore an indispensable part of the mesh generation programs.

Chapter 6

Verification Tests

6.1 Tests of the Reciprocity Theorem

Two series of patch tests were conducted using a square element of unit side with edges parallel to the coordinate axes. These tests were performed to show that the computer program is accurate and obeys the reciprocity principle. The form of this principle relevant to the developments in this thesis states that if a body is in equilibrium under two sets of forces, then the work done by the first set of forces acting through the displacements of the second set of forces is equal to the work done by the second set of forces acting through the displacements of the first set of forces, [7]. The tests consisted of subjecting a square plate to a uniform load on one edge while the opposite edge was pinned, see Figures 6.1 and 6.2.

Point loads were used in the first series of tests, and distributed loads were used in the second series. Each series consisted of paired tests in which the three nodes on any given side were pinned while the opposite nodes were loaded, first with compressive loads and then reversing the loads to make them tensile. Thus, each series consisted of four paired tests. Each pair of tests should yield stresses and displacements which are equal in magnitude but opposite in direction.

Using Figure 6.1 (a) as an example, nodes 4 and 8 should bulge out under compressive loading and should squeeze in under tensile loading. This is certainly so in the results shown in the Table 6.1. Using a simple one dimensional model, the average stress and displacement along the loaded edge (nodes 5, 6 and 7) can be computed as follows:

$$v = b.\sigma/E \tag{6.1}$$

where
v = vertical (y) displacement,
b = length = 1.0 ,
E = modulus of elasticity = 1.0,

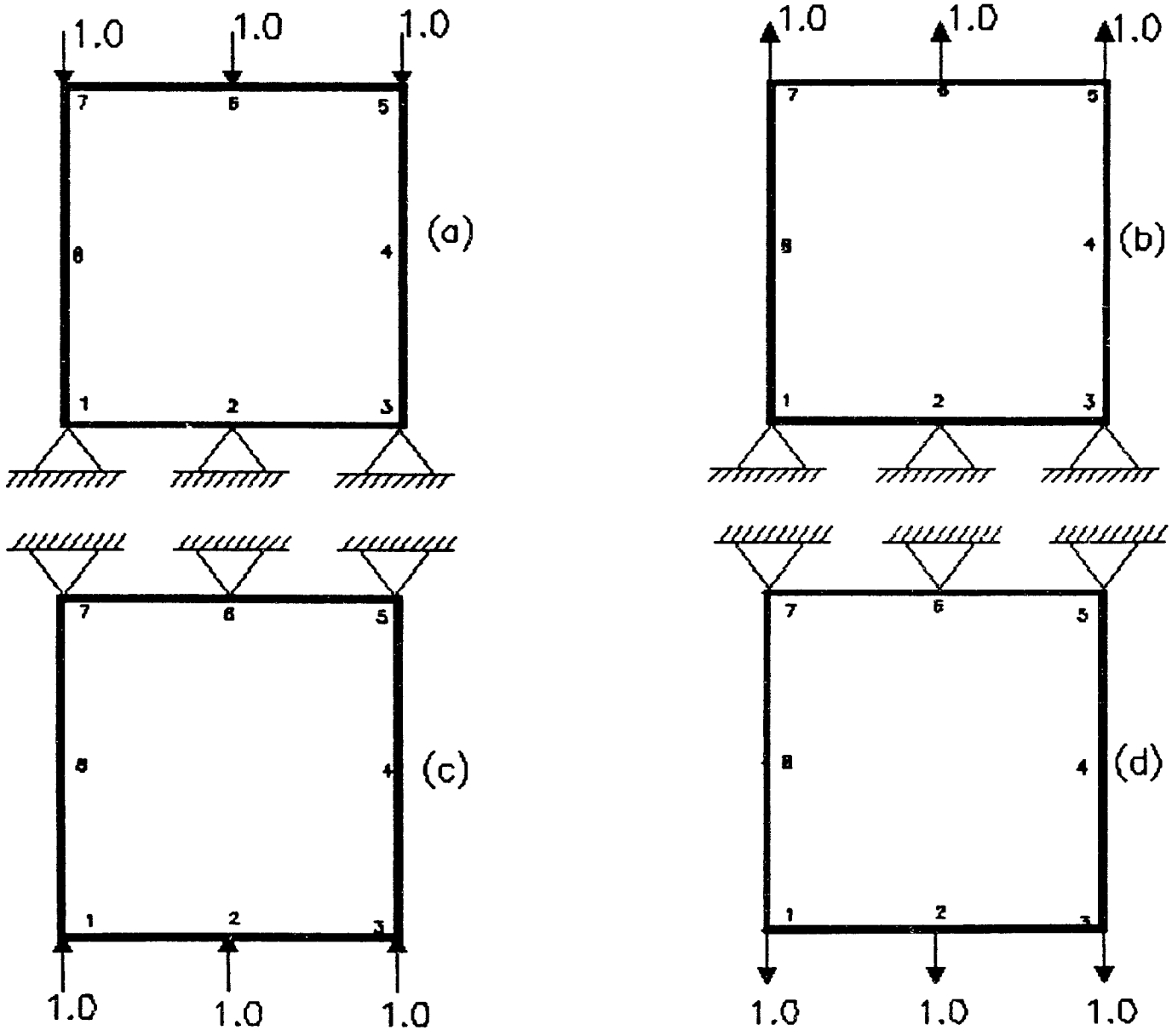


Figure 6.1: Tests of the Reciprocity Theorem

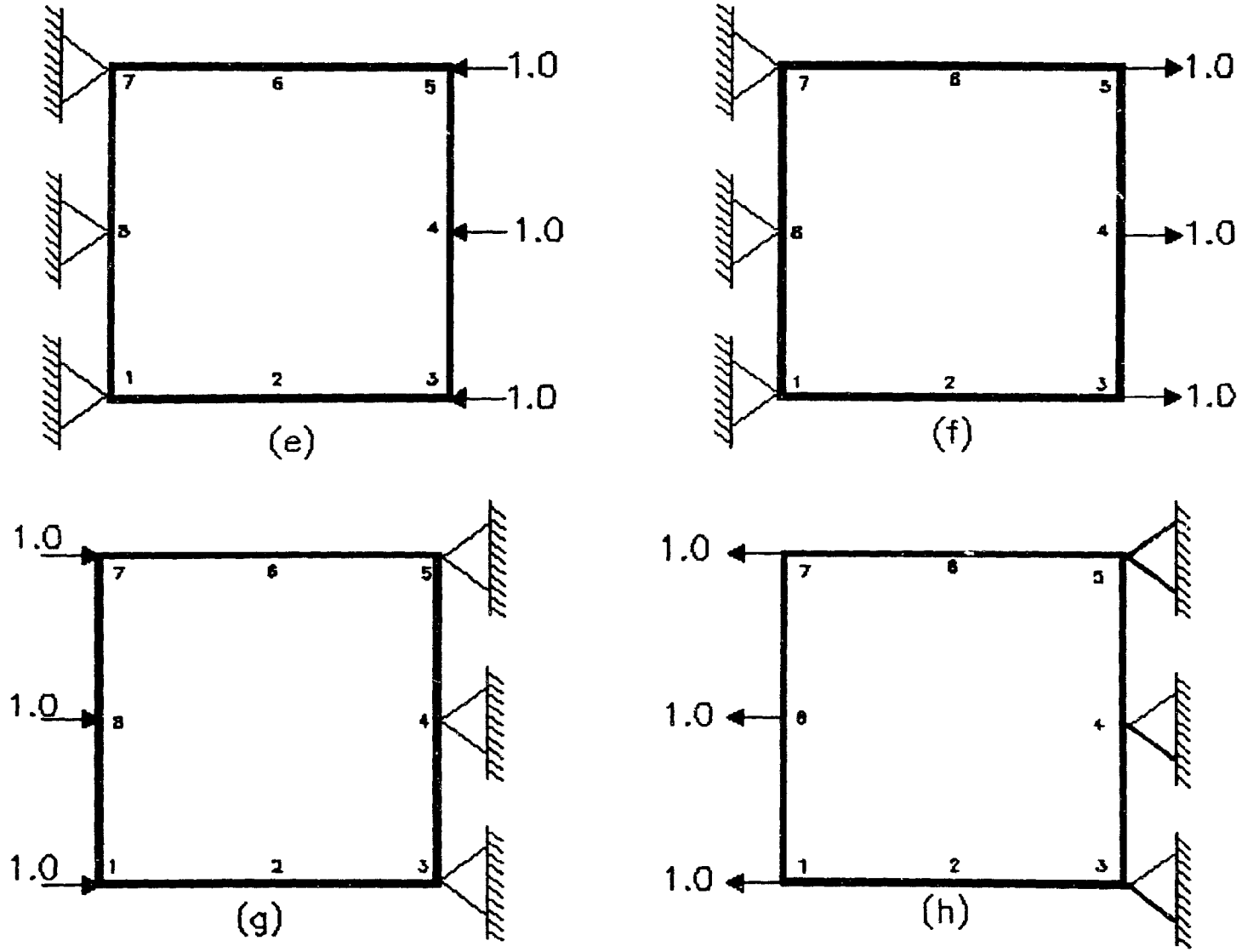


Figure 6.2: Tests of the Reciprocity Theorem, continued

σ = average stress = P/A,
A = cross sectional area = 1.0,
P = average load = -(0.5 + 1.0 + 0.5) = -2.0

Thus,
average stress = -2.0, and v = -2.0
Using the computed values from the program for this load case, we have
v = -1.069 + (-3.224 + 1.069)/3 = -1.787
which can be approximated to -2.0, considering that only one element is used in the analysis.

Only the displacements and stresses obtained for the load cases shown in Figures 6.1(a) - (d) are shown in Tables 6.1 - 6.8. The other four load cases shown in Figures 6.2(e) - (h) are the clockwise rotations through 90° of Figures 6.1(a) - (d) respectively. The results for these load cases are similarly the results of the tabulated load cases rotated clockwise through 90°. The results confirm the reciprocity principle.

6.2 The Cantilever Beam Problem

The cantilever beam problem is a useful beam problem to investigate numerically because it is simple to represent mathematically especially when the loading system is simple. Many authors in the field of Mechanics have dealt with it, see Valliapan [65], Popov [66], and Kreyszig [67]. Valliapan has considered the case of a cantilever beam of narrow cross-section of width b and depth $2d$, under a vertical end load P applied at the centre O, of the cross sectional end. In this example, b is small compared to $2d$ so that the beam can be considered to be in a state of plane stress. O is the origin of coordinates with the x-axis parallel to the longitudinal axis of the beam and the y-axis vertical. The length of the beam is L.

The boundary conditions are then as follows:

- $\sigma_y = 0$ for all x and y
- $\tau_{xy} = 0$ at the upper and lower surfaces, that is, $y = \pm d$
- τ_{xy} = P at the so-called neutral surface, that is, y = 0

If I is the moment of inertia about the x-axis, then the following equations of stress hold for the beam so described:

$$I = (2bd^3)/3$$

$$\sigma_x = -(P_{xy})/I$$

$$\tau_{xy} = -P(d^2 - y^2)/2I \tag{6.2}$$

The equations for stress are correct only if the boundary force at the free end is distributed parabolically. If the end load is a concentrated load, then by virtue of St. Venant's principle, the solution near the end of the beam will not be correct. This means that in a finite element analysis of the beam problem, the stresses and displacements at the end of the beam will not be correct. This effect can be mitigated by using a fine mesh. The equations of displacements are given as:

$$u_x = Py\{(3L^2 - 3x^2 - \nu y^2)/E + (y^2 - 3d^2)/G\}/6I$$

$$u_y = P(3\nu xy^2 + x^3 - 3L^2x + 2L^3)/6EI \tag{6.3}$$

The displacements at the so-called neutral surface (y = 0) are

$$u_x = 0$$

$$u_y = P(2L^3 - 3L^2x + x^3)/6EI \tag{6.4}$$

In the above expressions, E is modulus of elasticity and G is rigidity modulus, ν is Poisson's ratio.

$$G = E/2(1 + \nu) \tag{6.5}$$

Two computer models were run with the following specifications:

- P = -20.0
- L = 60.0
- 2d = 4.0
- b = 1.0
- E = 1.0E08
- q = 0.25,
- Number of elements used in first mesh = 10
- Number of elements used in second mesh = 20

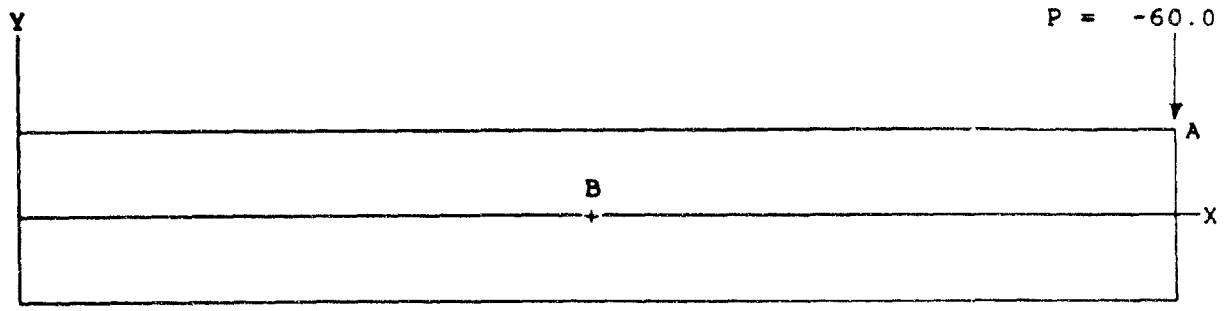


Figure 6.3: The Cantilever Beam under a Point Load, P

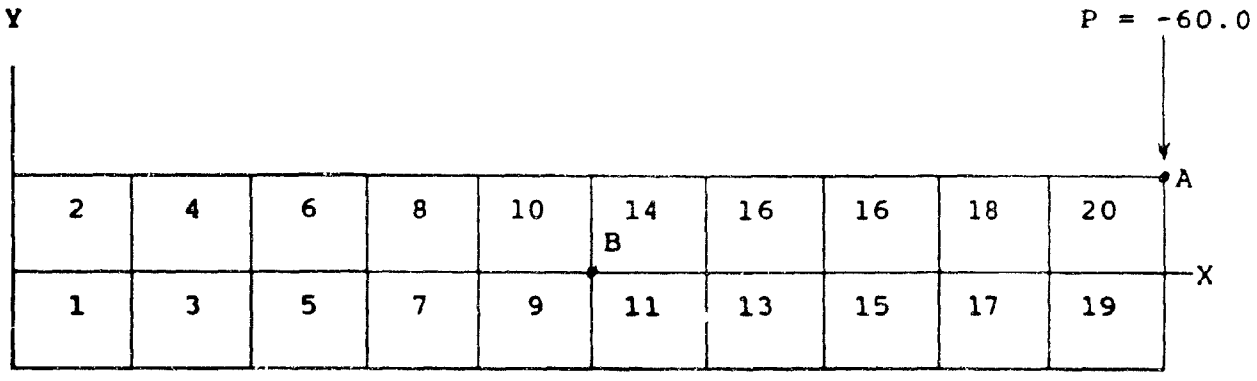


Figure 6.4: Discretization of the Cantilever Beam into 20 Elements

Figure 6 3 shows a free-body representation of a cantilever beam fixed on the left edge. Figure 6.4 shows a 20-element, finite element discretization of the beam.

Table 6 9 summarizes the hybrid test results for the displacements. A comparison with the exact mathematical solution is also included in the table. The table clearly illustrates that the results of the hybrid finite element programme are in very good agreement with the closed form analytical solutions.

6.3 The Hole-in-a-Plate Problem

The circular hole in a plate problem, Jaeger and Cook [7], Obert and Duvall [68], is a classic case that has been examined extensively in books on the theory of elasticity. It lends itself to an analytic solution and is therefore a proper case that can be verified by the hybrid finite element analysis.

Consider an infinite plate of thickness t with a circular hole of radius r_0 located with centre at the origin, O, as shown in Figure 6.5

Let S_x and S_y be the stresses at infinity applied in the x- and y- directions respectively. At a large distance from the hole, the polar components of stress, σ_r, σ_θ, $\tau_{r\theta}$, are due entirely to the applied stresses and are given in Equations 6.6 below.

$$\sigma_r = \frac{1}{2}(S_x + S_y)(1 - r\theta^2/r^2) + \frac{1}{2}(S_x - S_y)(1 + 3r\theta^4/r^4 - 4r\theta^2/r^2)\cos 2\theta$$

$$\sigma_\theta = \frac{1}{2}(S_x + S_y)(1 + r\theta^2/r^2) - \frac{1}{2}(S_x - S_y)(1 + 3r\theta^4/r^4)\cos 2\theta$$

$$\tau_{r\theta} = \frac{1}{2}(S_r - S_y)(1 - 3r\theta^4/r^4 + 2r\theta^2/r^2)\sin 2\theta \qquad (6\,6)$$

At the periphery of the borehole, r_θ = r and Equations (6.6) reduce to the following:

$$\sigma_r = \tau_{r\theta} = 0,$$

$$\sigma_\theta = (S_x + S_y) + 2(S_x - S_y)\cos 2\theta \qquad (6.7)$$

Points A and B (Figure 6.5), at the periphery of the borehole are of special interest because at $A, \theta = \pi/2$ and at $B, \theta = 0$. Also, at A, $\sigma_\theta = \sigma_x$ and at B, $\sigma_\theta = \sigma_y$

Three cases of applied stresses can be considered thus:

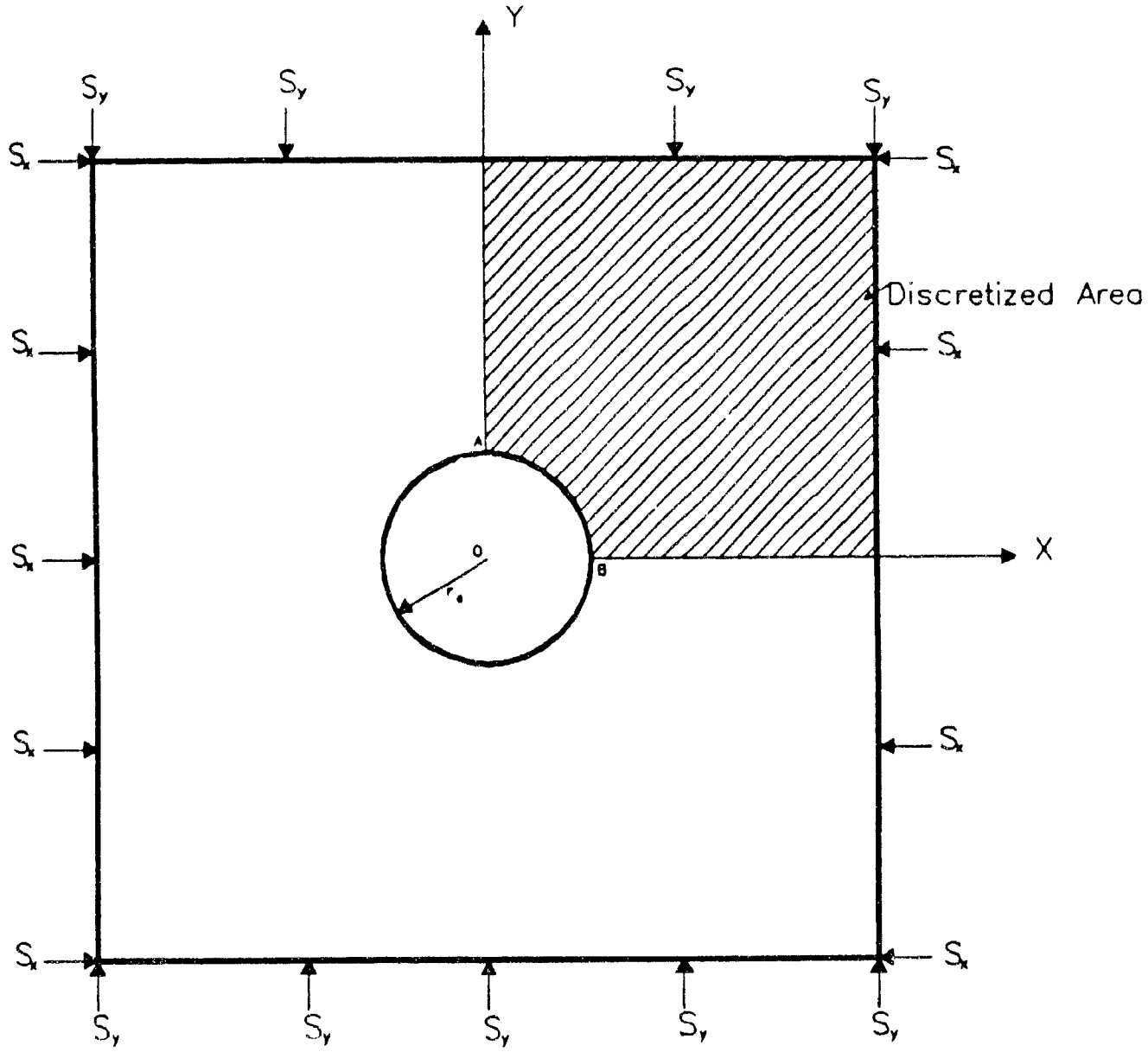


Figure 6.5: Circular Hole-in-a-Plate Problem showing Discretized Zone

case (a): $S_x = S_y = -p$, which leads to $\sigma_{x_A} = \sigma y_B = -2p$,

case (b): $S_x = 0$, which leads to $\sigma_{x_A} = -S_y$, $\sigma y_B = -3S_y$,

case (c): $S_y = 0$, which leads to $\sigma_{x_A} = 3S_x$, $\sigma y_B = -S_x$

In all cases, $\sigma_{x_B} = \sigma_{y_A} = \tau_{xy} = 0$

Case (c) is the complement of case (b) when the axes are exchanged so that only cases (a) and (b) need be tested in a finite element analysis. Four meshes of different finite element densities were used as shown in Figures 6 6 to 6.9 were used.

The meshes were 2x2, 4x4, 6x6 and 8x8 elements, the description indicating that there are equal number of elements along the x- and the y- axis. These meshes were generated by the program CIRCMESH. The results of the hybrid finite element tests are summarized in Tables 6.10 and 6.11. It is clear from these tables that the accuracy of the hybrid stress model is high and adequate for the types of real mining problems it would be employed to solve. Figure 6.10 shows the distribution of the horizontal stress along section AB of Figure 6.9 for case (a). The corresponding horizontal stress distribution along the same section for case (b) is shown in Figure 6.11

6.4 The Pure Beam Bending Problem

Desai and Abel [69] have analyzed the problem of a beam subjected to pure bending stresses. They used four-node elements in a finite element program based on the displacement model. In this test, the 8-node hybrid stress program is used not only to compare the results obtained with those by Desai and Abel but also to confirm that the hybrid stress model converges to the right answer rapidly with relatively fewer elements. Figure 6.12 shows the plane stress representation of the pure beam bending problem. Due to symmetry, only a quarter of the section is discretized as shown in Figure 6 12. The two points of interest are the middle of the discretized section and the far corner, marked X and Y respectively in Figure 6 12.

The material properties used were, as for Desai and Abel,

E = 30 x 106 psi (= 207 MPa), $\nu = 0.3$ and h = 1.0 inch

The results are summarized in Table 6.12. They clearly show that the hybrid stress model more accurately predicts the results than the displacement model. Also, no improvement in accuracy is gained by using greater density of elements from the 4-element, 21-node discretization.

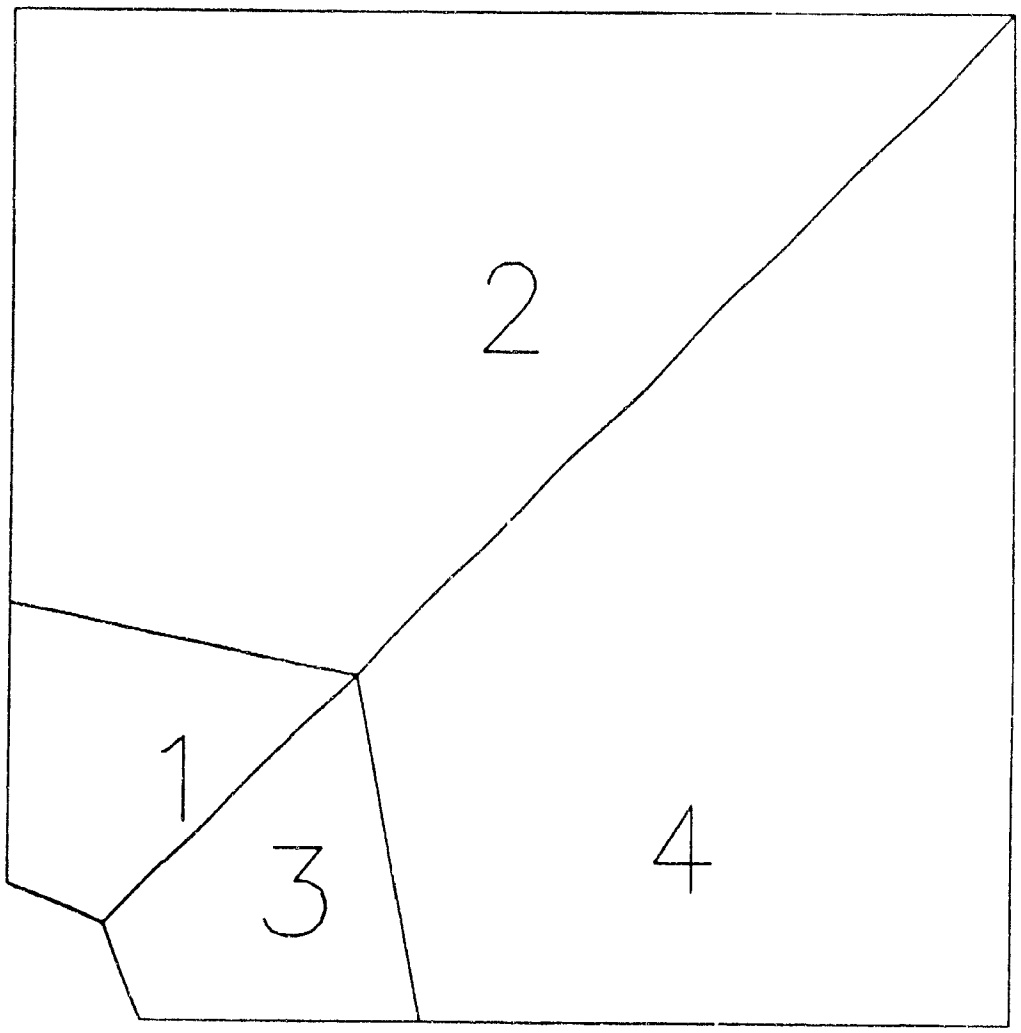


Figure 6.6: 21-Node, 4-Element Circular Mesh

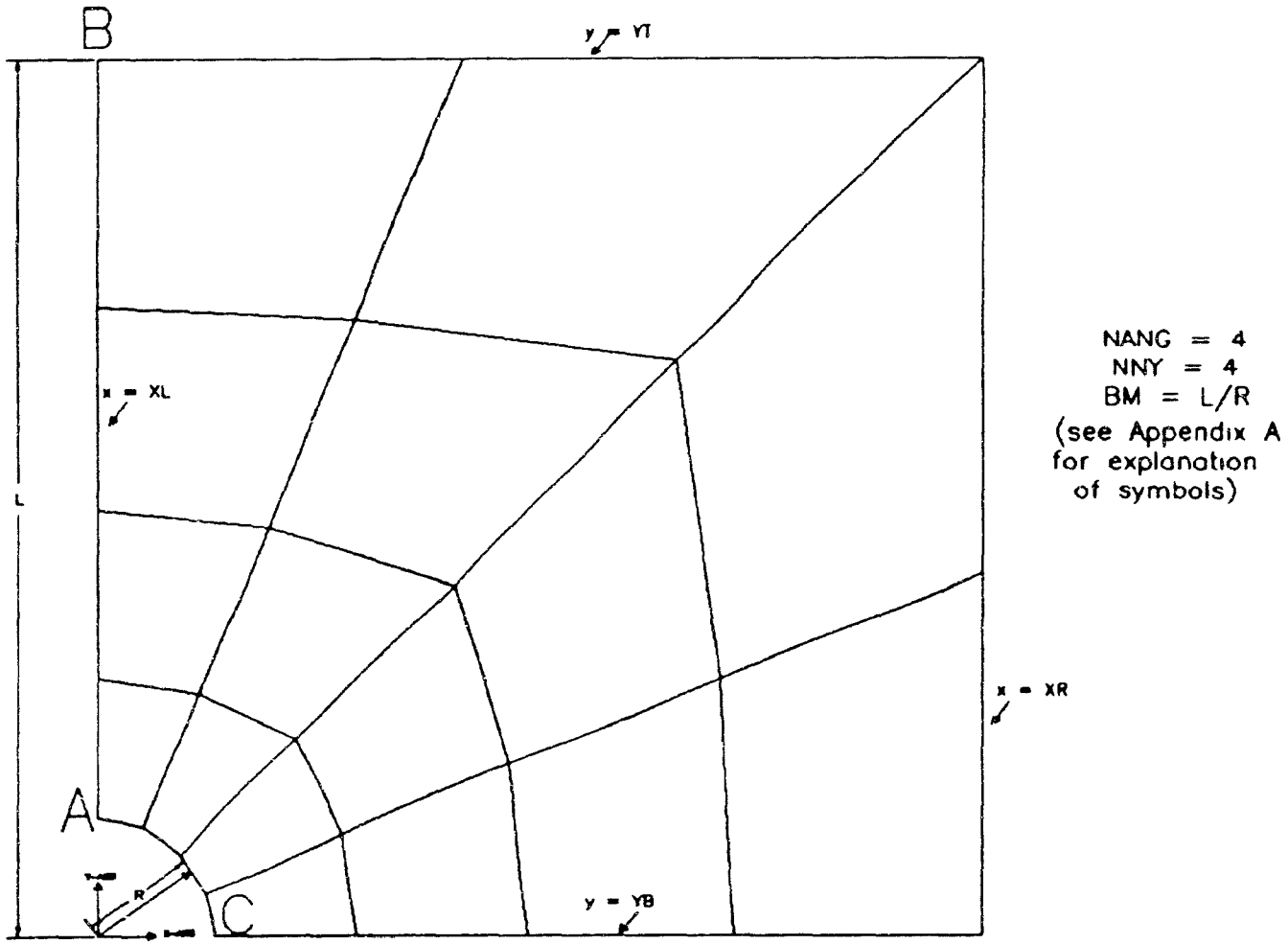


Figure 6.7: 65-Node, 16-Element Circular Mesh

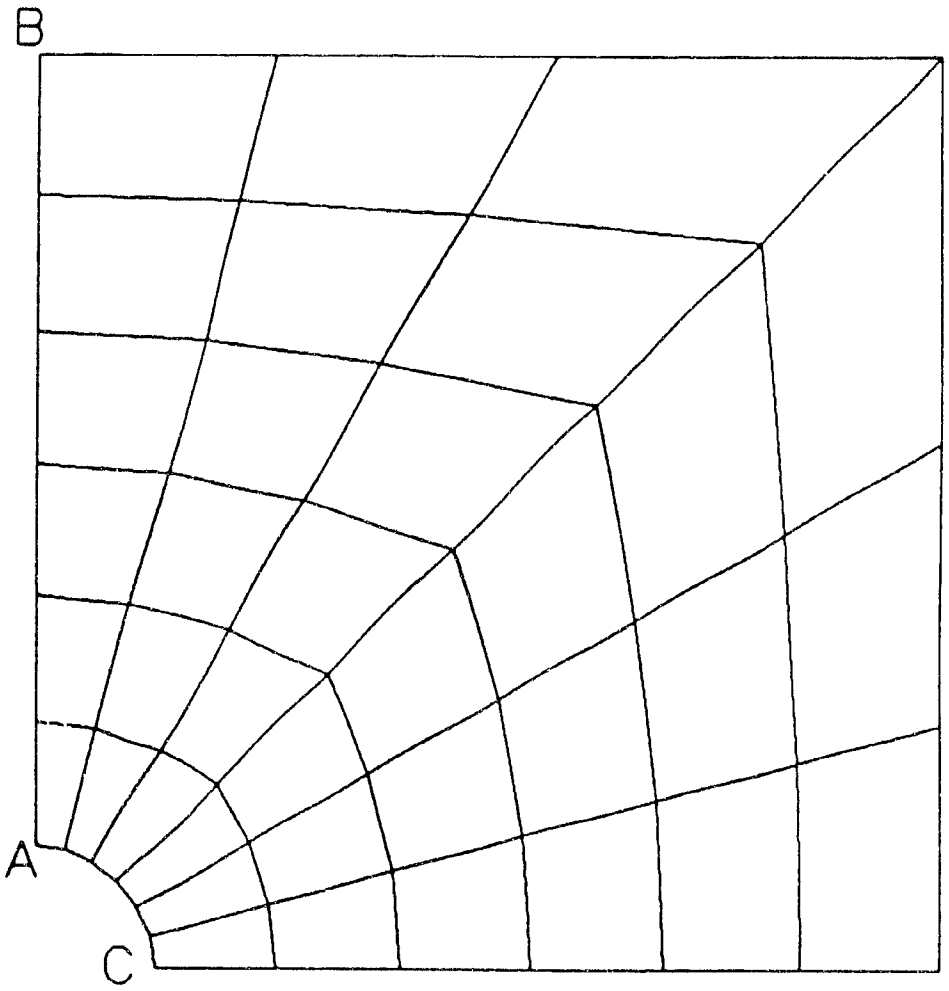


Figure 6.8: 133-Node, 36-Element Circular Mesh

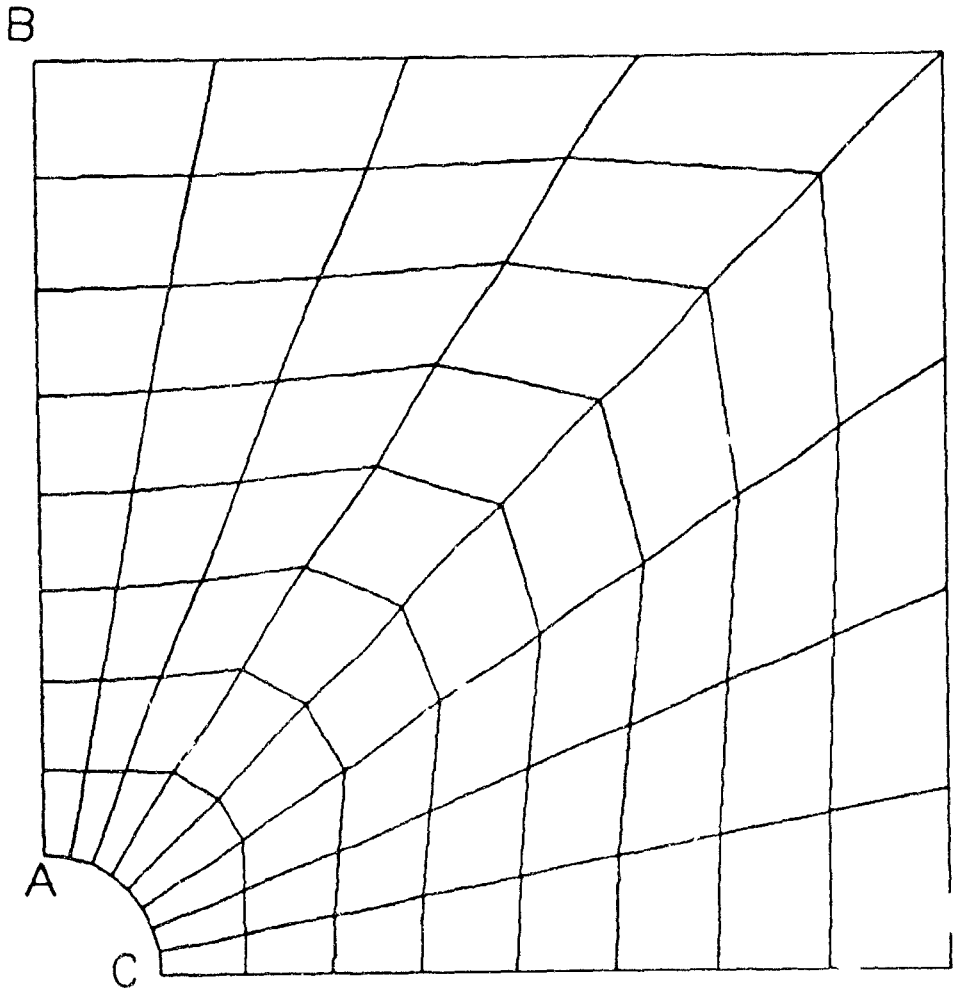


Figure 6.9: 255-Node, 64-Element Circular Mesh

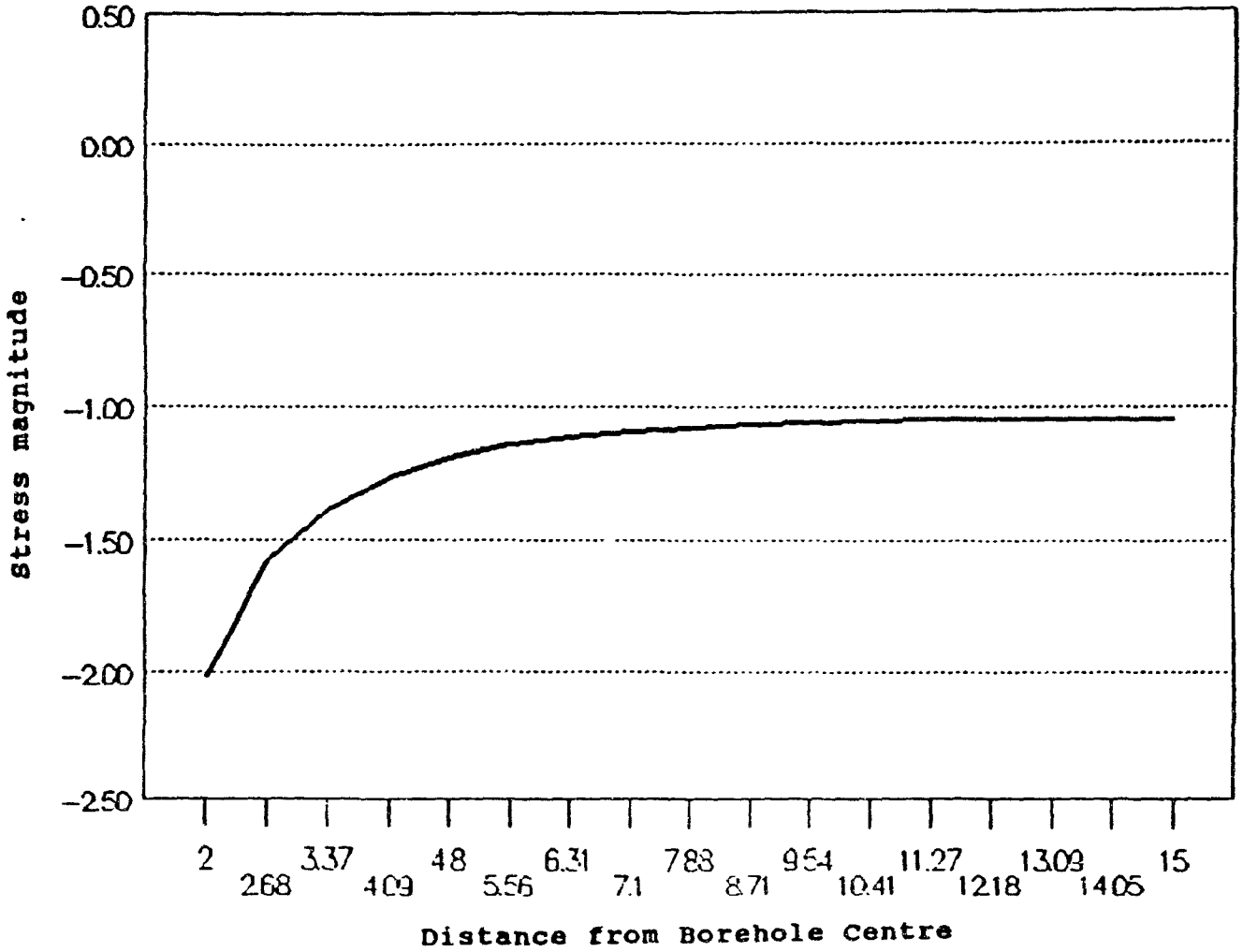


Figure 6.10: Stress Distribution Along Section AB of Figure 6.8, case (a)

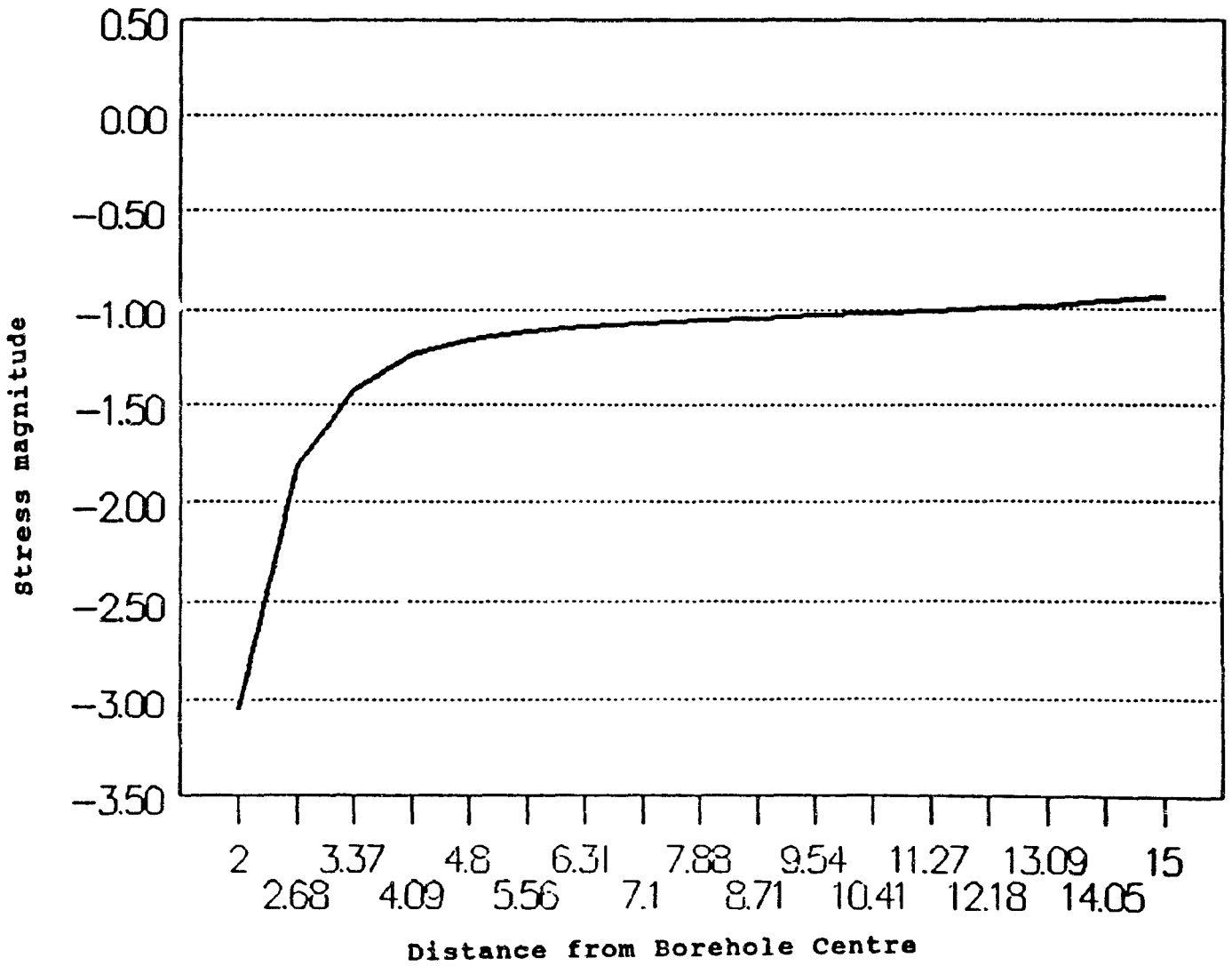


Figure 6.11: Stress Distribution Along Section AB of Figure 6.8, case (b)

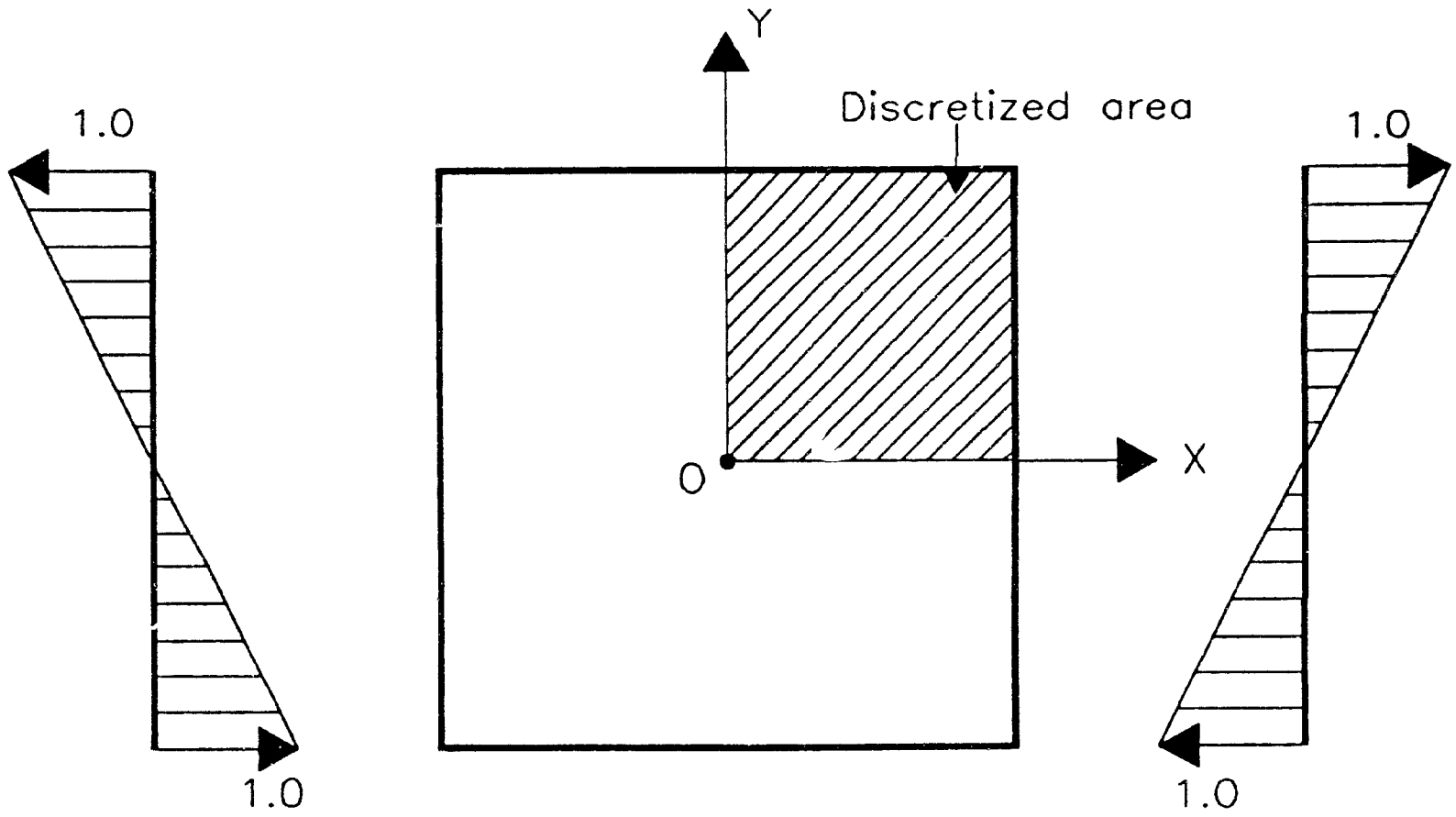


Figure 6.12: Free Body Representation of a Beam in Pure Bending

Table 6.1: Patch Test Displacement Results for Point Loads (from Figures 6.1(a) and (b))

Node	Figure 6.1(a) Displacements		Figure 6.1(b) Displacements	
	X-Disp.	Y-Disp.	X-Disp.	Y-Disp.
1	0.	0.	0.	0.
2	0.	0.	0.	0.
3	0.	0.	0.	0.
4	0.1424	-1.0440	-0.1424	1.0440
5	0.5664	-2.6635	-0.5664	2.6635
6	-0.0000	-1.6227	0.0000	1.6227
7	-0.5664	-2.6635	0.5664	2.6635
8	-0.1424	-1.0440	0.1424	1.0440

Table 6.2: Patch Test Displacement Results for Point Loads (from Figures 6.1(a) and (b))

Node	Figure 6.1(a) Stresses			Figure 6.1(b) Stresses		
	Sigma-x	Sigma-y	Tau-xy	Sigma-x	Sigma-y	Tau-xy
1	-0.5294	-1.7457	-0.2292	0.5294	1.7457	0.2292
2	-0.4396	-1.9444	-0.0000	0.4396	1.9444	0.0000
3	-0.5294	-1.7457	0.2292	0.5294	1.7457	-0.2292
4	-0.0689	-2.7357	0.0681	0.0689	2.7357	-0.0681
5	1.0803	-3.2239	-0.7743	-1.0803	3.2239	0.7743
6	0.4887	-1.0690	-0.0000	-0.4887	1.0690	0.0000
7	1.0803	-3.2239	0.7743	-1.0803	3.2239	-0.7743
8	-0.0689	-2.7357	-0.0681	0.0689	2.7357	0.0681

Table 6.3: Patch Test Displacement Results for Point Loads (from Figures 6.1(c) and (d))

Node	Figure 6.1(c) Displacements		Figure 6.1(d) Displacements	
	X-Disp.	Y-Disp.	X-Disp.	Y-Disp.
1	-0.5664	2.6635	0.5664	-2.6635
2	0.0000	1.6227	-0.0000	-1.6227
3	0.5664	2.6635	-0.5664	-2.6635
4	0.1424	1.0440	-0.1424	-1.0440
5	0.	0.	0.	0.
6	0.	0.	0.	0.
7	0.	0.	0.	0.
8	-0.1424	1.0440	0.1424	-1.0440

Table 6.4: Patch Test Stress Results for Point Loads (from Figures 6.1(c) and (d))

Node	Figure 6.1(c) Stresses			Figure 6.1(d) Stresses		
	Sigma-x	Sigma-y	Tau-xy	Sigma-x	Sigma-y	Tau-xy
1	1.0803	-3.2239	-0.7743	-1.0803	3.2239	0.7743
2	0.4887	-1.0690	0.0000	-0.4887	1.0690	-0.0000
3	1.0803	-3.2239	0.7743	-1.0803	3.2239	-0.7743
4	-0.0689	-2.7357	-0.0681	0.0689	2.7357	0.0681
5	-0.5294	-1.7457	-0.2292	0.5294	1.7457	0.2292
6	-0.4396	-1.9444	0.0000	0.4396	1.9444	-0.0000
7	-0.5294	-1.7457	0.2292	0.5294	1.7457	-0.2292
8	-0.0689	-2.7357	0.0681	0.0689	2.7357	-0.0681

Table 6.5: Patch Test Displacement Results for Distributed Loads (from Figures 6.1(a) and (b))

Node	Figure 6.1(a) Displacements		Figure 6.1(b) Displacements	
	X-Disp.	Y-Disp.	X-Disp.	Y-Disp.
1	0.	0.	0.	0.
2	0.	0.	0.	0.
3	0.	0.	0.	0.
4	-0.1270	0.5060	0.1270	-0.5060
5	-0.1104	0.9723	0.1104	-0.9723
6	-0.0000	0.9974	0.0000	-0.9974
7	0.1104	0.9723	-0.1104	-0.9723
8	0.1270	0.5060	-0.1270	-0.5060

Table 6.6: Patch Test Displacement Results for Distributed Loads from Figures 6.1 (c) and (d)

Node	Figure 6.1(c) Displacements		Figure 6.1(d) Displacements	
	X-Disp.	Y-Disp.	X-Disp.	Y-Disp.
1	-0.1104	0.9723	0.1104	-0.9723
2	0.0000	0.9974	-0.0000	-0.9974
3	0.1104	0.9723	-0.1104	-0.9723
4	0.1270	0.5060	-0.1270	-0.5060
5	0.	0.	0.	0.
6	0.	0.	0.	0.
7	0.	0.	0.	0.
8	-0.1270	0.5060	0.1270	-0.5060

Table 6.7: Patch Test Stress Results for Distributed Loads (from Figures 6.1(a) and (b))

Node	Figure 6.1(a) Stresses			Figure 6.1(b) Stresses		
	Sigma-x	Sigma-y	Tau-xy	Sigma-x	Sigma-y	Tau-xy
1	0.1921	1.0290	0.1072	-0.1921	-1.0290	-0.1072
2	0.2674	0.9392	0.0000	-0.2674	-0.9392	-0.0000
3	0.1921	1.0290	-0.1072	-0.1921	-1.0290	0.1072
4	-0.0268	0.9898	0.0072	0.0268	-0.9898	-0.0072
5	0.0219	0.8723	0.0494	-0.0219	-0.8723	-0.0494
6	0.0250	1.0319	0.0000	-0.0250	-1.0319	-0.0000
7	0.0219	0.8723	-0.0494	-0.0219	-0.8723	0.0494
8	-0.0268	0.9898	-0.0072	0.0268	-0.9898	0.0072

Table 6.8: Patch Test Stress Results for Distributed Loads (from Figures 6.1(c) and (d))

Node	Figure 6.1(c) Stresses			Figure 6.1(d) Stresses		
	Sigma-x	Sigma-y	Tau-xy	Sigma-x	Sigma-y	Tau-xy
1	-0.0219	-0.8723	-0.0494	0.0219	0.8723	0.0494
2	-0.0250	-1.0319	-0.0000	0.0250	1.0319	0.0000
3	-0.0219	-0.8723	0.0494	0.0219	0.8723	-0.0494
4	0.0268	-0.9898	0.0072	-0.0268	0.9898	-0.0072
5	-0.1921	-1.0290	-0.1072	0.1921	1.0290	0.1072
6	-0.2674	-0.9392	0.0000	0.2674	0.9392	-0.0000
7	-0.1921	-1.0290	0.1072	0.1921	1.0290	-0.1072
8	0.0268	-0.9898	-0.0072	-0.0268	0.9898	0.0072

Table 6.9: Comparative Displacements of the Cantilever Beam Bending Problem

Total No. of nodes	Total No. of elems.	Displacements at Point A (x 10^{-4})		Displacements at Point B (x 10^{-4})	
		u	v	u	v
		Hybrid stress solutions			
53	10	0.000	-0.002706	0.000	-0.0008469
85	20	0.000	-0.002705	0.000	-0.0008460
		Exact solution			
		0.000	-0.002700	0.00	-0.0008438

Table 6.10: Hybrid Stress Results for Hole-in-Plate Problem, case (a)

Total No. of nodes	Total No. of elems.	Stresses at Point A		Stresses at Point C	
		σ_x (-2.00)	σ_y (0.00)	σ_x (0.00)	σ_y (-2.00)
21	4	-1.838	-0.317	-0.317	-1.838
65	16	-1.974	-0.129	-0.129	-1.974
133	36	-1.998	-0.093	-0.093	-1.998
225	64	-2.018	-0.052	-0.052	-2.018

Note: Values enclosed in brackets are the theoretical values

Table 6.11: Hybrid Stress Results for Hole-in-Plate Problem, case (b)

Total No. of nodes	Total No. of elems.	Stresses at Point A		Stresses at Point C	
		σ_x (-1.00)	σ_y (0.00)	σ_x (0.00)	σ_y (3.00)
21	4	-1.838	-0.317	-0.317	-1.838
65	16	-1.974	-0.129	-0.129	-1.974
133	36	-1.998	-0.093	-0.093	-1.998
225	64	-2.018	-0.052	-0.052	-2.018

Table 6.12: Comparative Displacements of the Cantilever Beam Bending Problem

Total No. of nodes	Total No. of elems.	Displacements at Point X ($\times 10^{-4}$)		Displacements at Point Y ($\times 10^{-4}$)	
		u	v	u	v
		Hybrid stress solutions			
53	10	0.000	-0.002706	0.000	-0.0008469
85	20	0.000	-0.002705	0.000	-0.0008460
21	4	0.375	-0.319	1.500	-1.275
65	16	0.375	-0.318	1.498	-1.267
133	36	0.375	-0.319	1.500	-1.275
		Desai and Abel solutions [69]			
25	16	0.3679	-0.31236	1.4552	-1.2399
		Exact solution			
		0.3750	-0.3188	1.500	-1.275

Chapter 7

Case Histories

7.1 Introduction

The verification tests reported in Chapter 6 clearly demonstrated that the hybrid stress program can be relied upon to perform finite element analysis of a continuum to the same degree of accuracy and at less cost as any other finite element program with a different basis. In order to show that it can also be relied upon to carry out practical geomechanics type of analysis which typical mining operations perform in their ground control problems, two case histories were analyzed. Both case histories were taken from NORANDA Minerals operations in the Province of Quebec in Canada. This Chapter is therefore concerned with the correlation of numerical analyses on the two mines carried out by different methods on the one hand, and by the hybrid stress program on the other.

Noranda Inc. is a resource based company and mining is among its several areas of operation. The analyses described below indicated good correlation between hybrid stress results and those from other numerical procedures employed by Noranda Technology Centre. In each of the cases described, the same portion of a mining zone that was analyzed by a different numerical method was also analyzed by the hybrid finite element program developed in this thesis. Direct comparison of stresses and displacements was then made. The hybrid stress results revealed greater detail in part due to the 8-node discretization of the quadrilateral elements.

7.2 First Case Study: Geco Conveyor Drive Stability Analysis.

The descriptions in sections 7.2.1 to 7.2.3 inclusive were taken from an internal report written for Noranda Technology Centre by Bawden and Milne [70].

7.2.1 History

Geco mines is owned and operated by Noranda Minerals Inc., a natural resource-based company with interests in mining mainly in Canada. The Geco Cu-Zn-Ag deposit is in the Manitouwadge mining camp, in the Thunder Bay Mining District of northwestern Ontario, 320 kilometers east of Thunder Bay and about 150 kilometers northwest of Sault Ste Marie. The mining camp lies 5 kilometers east of the town of Manitouwadge. Geco Mines Ltd. was amalgamated with Noranda Mines Ltd. (now Noranda Inc) in 1964. The regional geological structure consists of a broad easterly limb of this synform. It consists of one vertical lenticular, continuous zone of mineralization which is interrupted by several diabase dykes and is offset by the Fox Creek Fault. The average horizontal length on any level is approximately 2400 feet with an average width of 65 feet. The bottom of the orebody plunges to the east at an average of 35 degrees as it follows the S-shaped dragfold which exists west of the orebody on each level. The orebody is made up of a core of massive sulphides consisting of pyrites, pyrrhotites, chalcopyrites, galena and minor amounts of gold. Appreciable amounts of silver are present, associated with the chalcopyrites and galena. The remainder of the massive core is made up of wall rock inclusions.

7.2.2 Mining

There are three main methods of mining. The most common method is the "bulk" method in which 70 ft wide, 300 ft high transverse stopes are taken at 120 and 150 ft intervals. Using large blast holes, the primary slices are blasted and the broken ore is drawn out the bottom of the muck pile while the void which is created above the broken ore is filled with waste rock supplied from a surface quarry. The fill material provides support for the weak walls. Upon completion of the stope, the waste rock fill is stabilized along the rib walls by introducing a mixture of hydraulically placed mill sands and cement suitably mixed.

Blast hole open stoping is practiced in the upper parts of the orebody where the transverse dimensions are less. Backfill with hydraulic sand is done after the stope is completely mined. The above two methods account for about 95% of the ore produced. The remaining 5% is mined by cut-and-fill methods in areas where the ore is too narrow to be mined by the other two methods and where more wall support is required. The stopes are typically 9 ft high and are taken along the strike. Again, hydraulic sand fill is used as backfill.

7.2.3 Stability Problems in the Conveyor Drift

The following is a description of the stability problems which ultimately led to a stability analysis of the mine, using the finite element program, SAP2D, according to Bawden and Milne [70], and Milne [71].

Since 1983, the mine has been carrying out the retreat phase of the transverse stoping, i.e., recovering the pillars. Since then, movement was being experienced in the conveyor drift and the 1-32 cross cut. The Mining Technology Division of the Noranda Technolgy Centre (NTC), based in Montreal then carried out field investigations as well as linear elastic finite element analysis of the area of concern.

Field Observations and Measurements by the Noranda Technology Centre

The field investigations comprised a limited amount of structural mapping and classification of the rock types. Three major joint sets were mapped, the most prominent of which was parallel to the foliation in the rock. Rock classification was done using the Barton (NGI) and the CSIR (RMR) systems. Closure measurements using extensometers were also undertaken. The results of these field investigations showed that the observed movements were due to the continued relaxation of the rock mass towards the stopes. The presence of a surface quarry above the mining zone might have caused some uplift but the reduction in the vertical stress due to this quarry was too small to consider. The relaxation mechanism was explained as the loss of a so-called clamping force across the foliation. This clamping force was supplied by compressive stresses acting across the foliation. With progressive stoping, the stresses are dispersed and the loss of the clamping force results in rock failure along the foliation. This mode of failure is well known at Geco. The backfill could not totally prevent this relaxation although it retarded it.

Computer Analysis by Noranda Technology Centre

A SAP2D finite element analysis was done to validate the field observations with the following assumptions according to Bawden and Milne [70], and Milne [71]. Vertical stresses were equal to the weight of the rock, meaning that they were gravitational. Bawden and Milne [70] claim that these conditions are similar to those expected in the Canadaian shield. However, it is known that horizontal stresses in the shield are quite often larger than can be accounted for by gravity alone. Maximum depth was taken as 420m.

A value for the Young's modulus of 26 GPa was estimated based on the RMR (CSIR) classification scheme. Poisson's ratio was taken as 0.30. Values for rock strength were not needed because the finite element model assumes a linearly elastic and homogeneous material. The main purpose of the finite element analysis was to show the effect of mining the adjacent stope to the surface, starting from the position it was in 1983 and ending in 1985 when the stope reached the surface. The computer results are illustrated in Figures 7.1, , , , , 7.6.

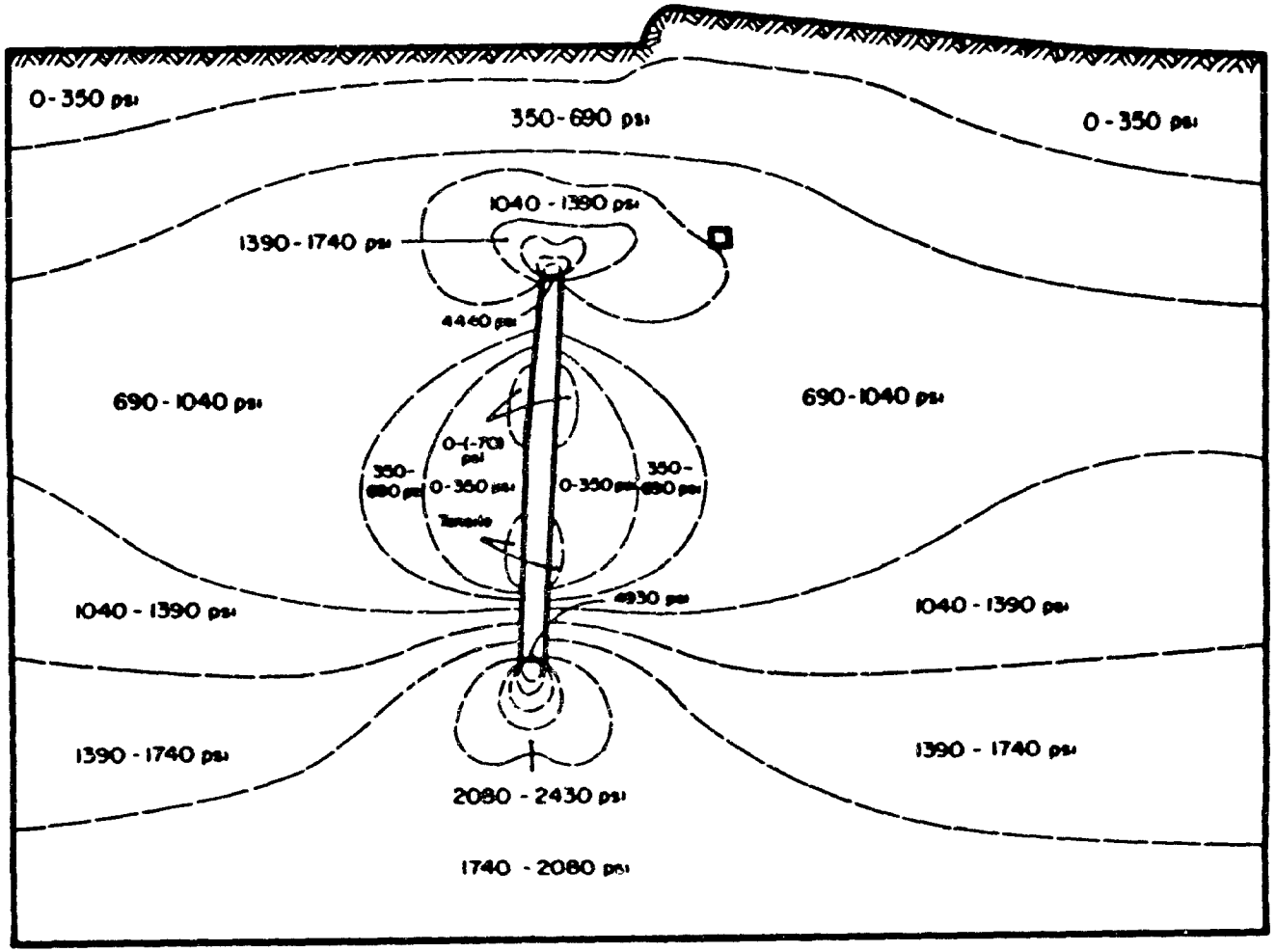


Figure 7.1: Geco Conveyor Drift Stability Analysis,
Maximum Stress Contours, Present Stope Extent, (by the NTC)

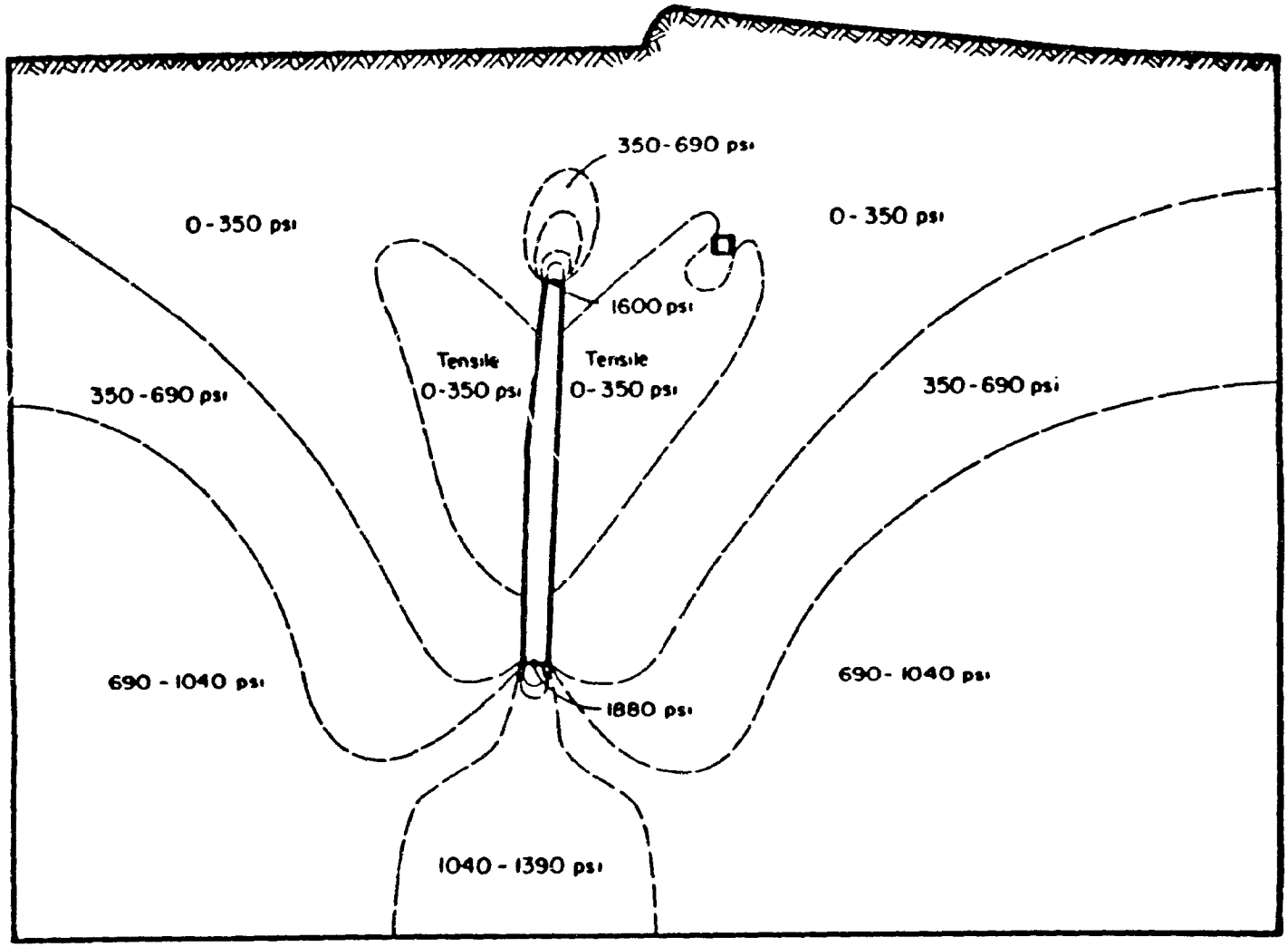


Figure 7.2: Geco Conveyor Drift Stability Analysis,
Minimum Stress Contours, Present Stope Extent, (by the NTC)

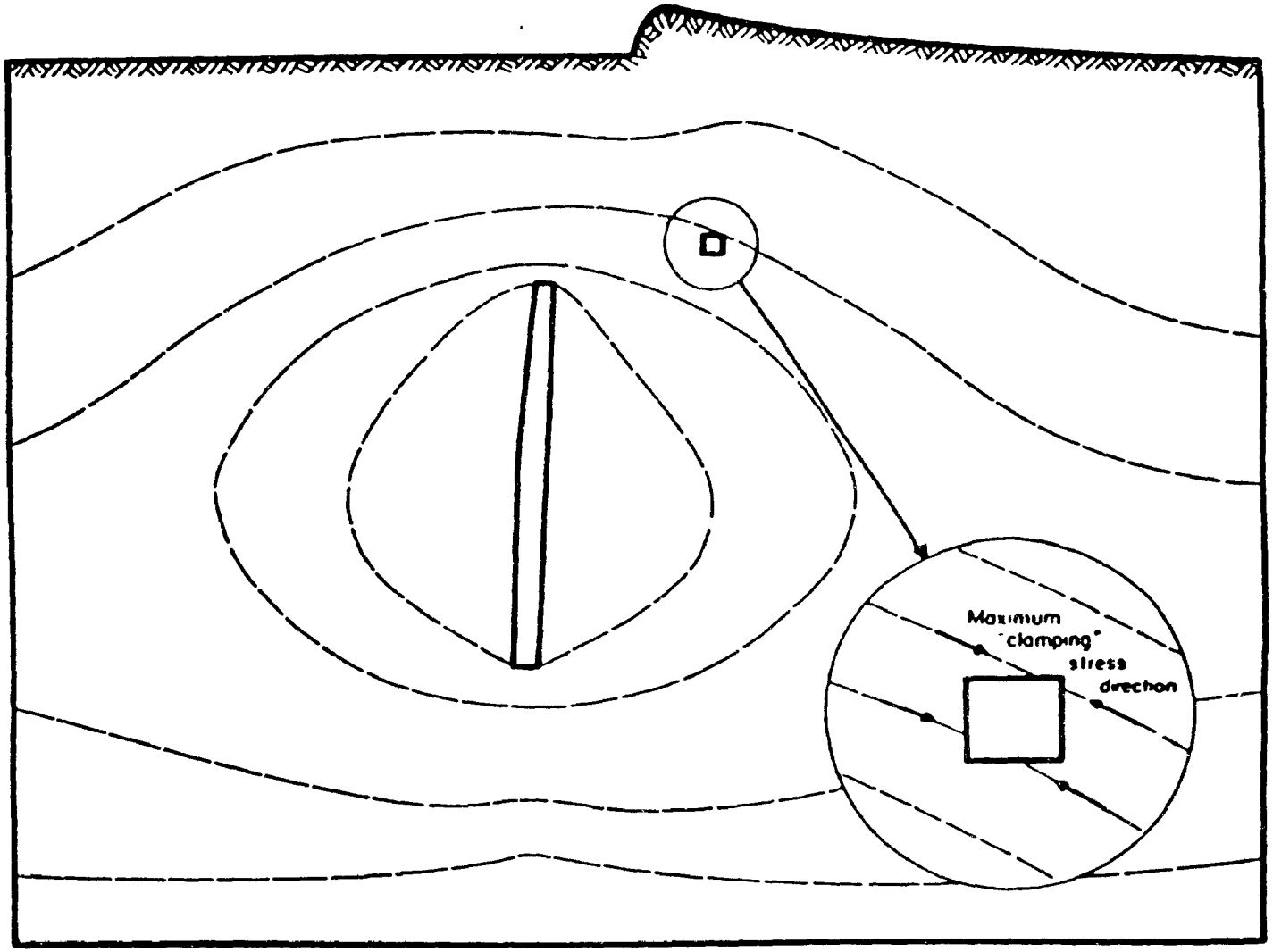


Figure 7.3: Geco Conveyor Drift Stability Analysis,
Maximum Principal Stress Directions, Present Stope Extent, (by the NTC)

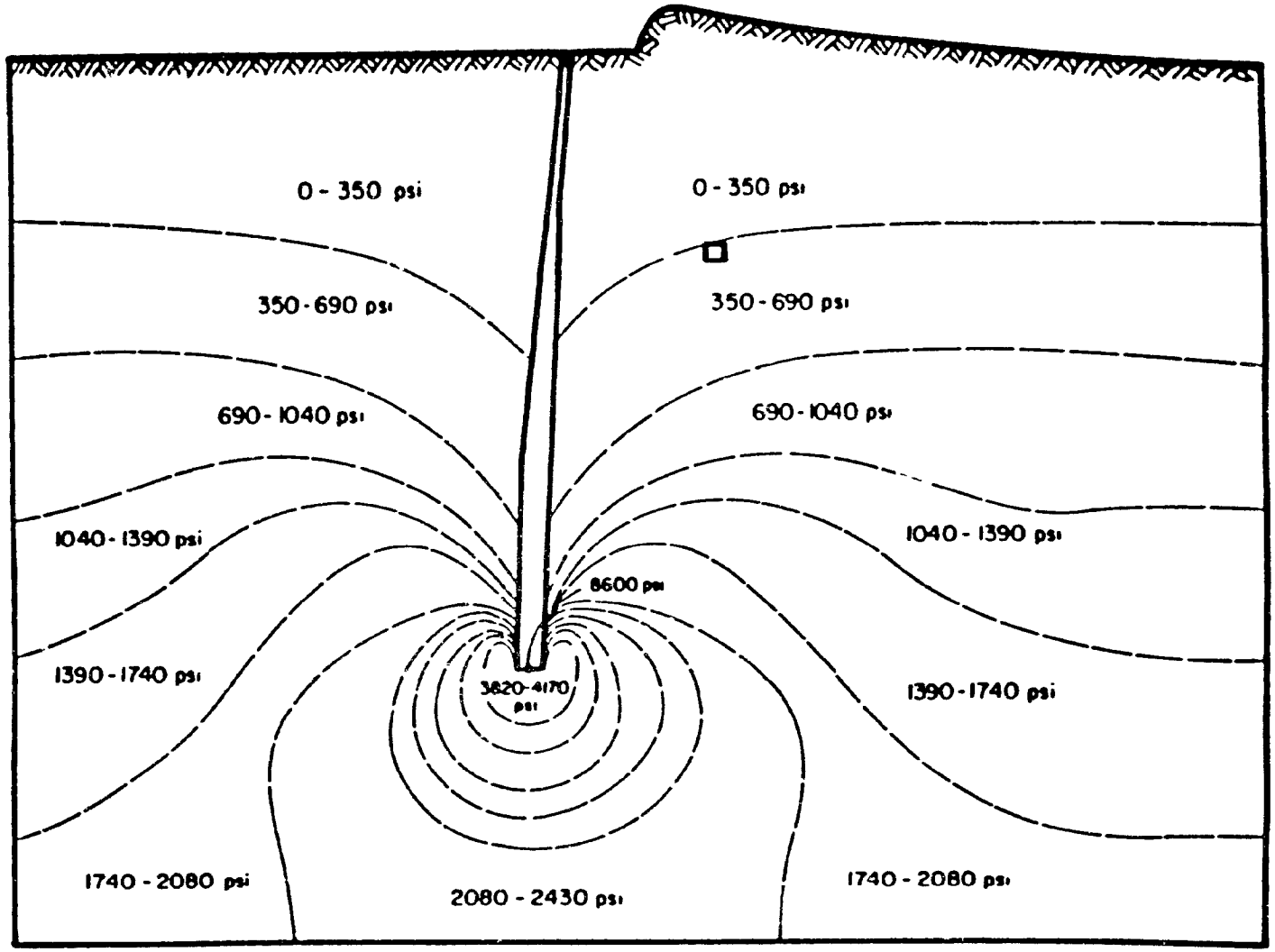


Figure 7.4: Geco Conveyor Drift Stability Analysis, Maximum Principal Stress Contours, Stope Extended to Surface, (by the NTC)

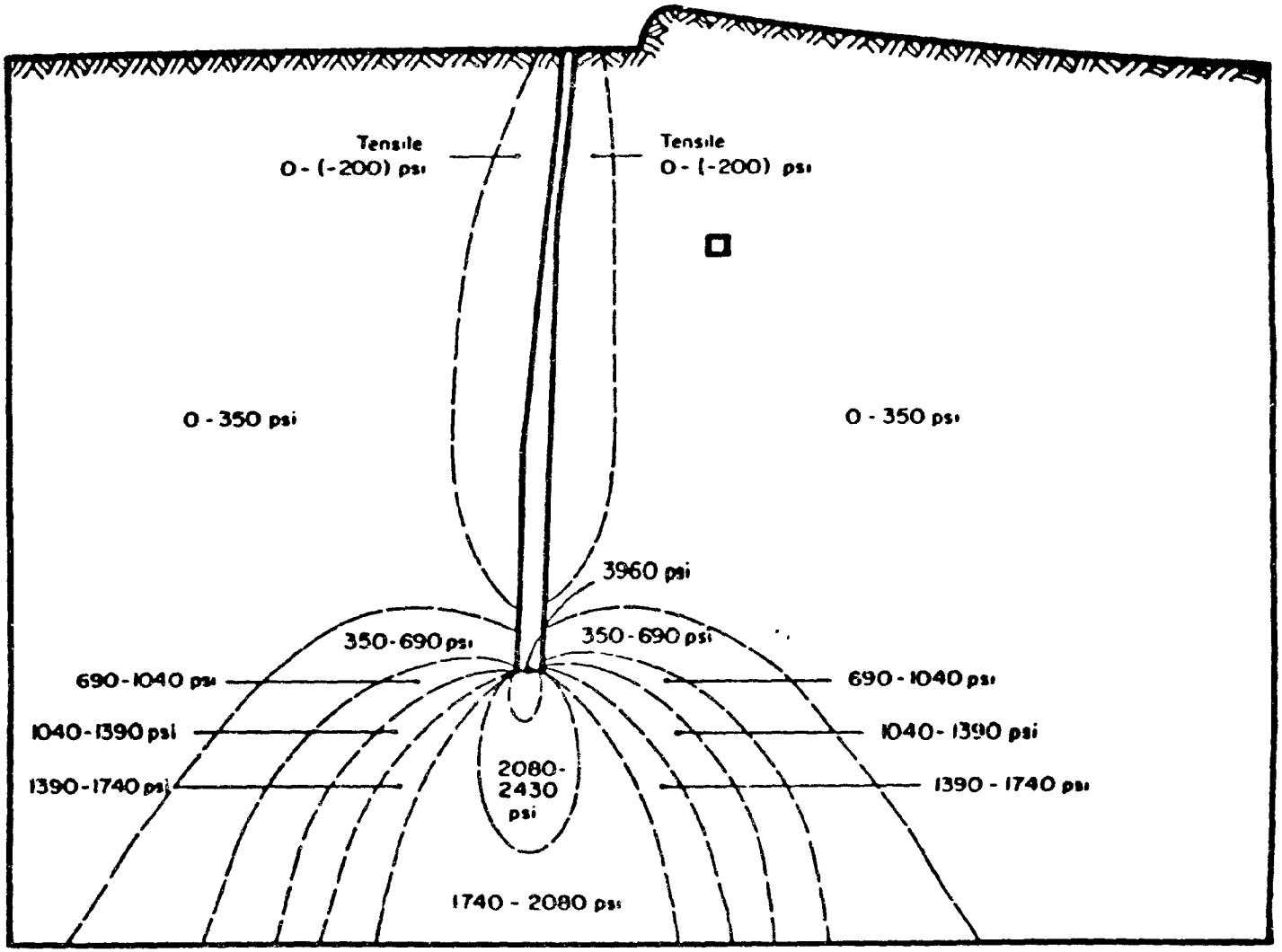


Figure 7.5: Geco Conveyor Drift Stability Analysis, Minimum Principal Stress Contours, Stope Extended to Surface, (by the NTC)

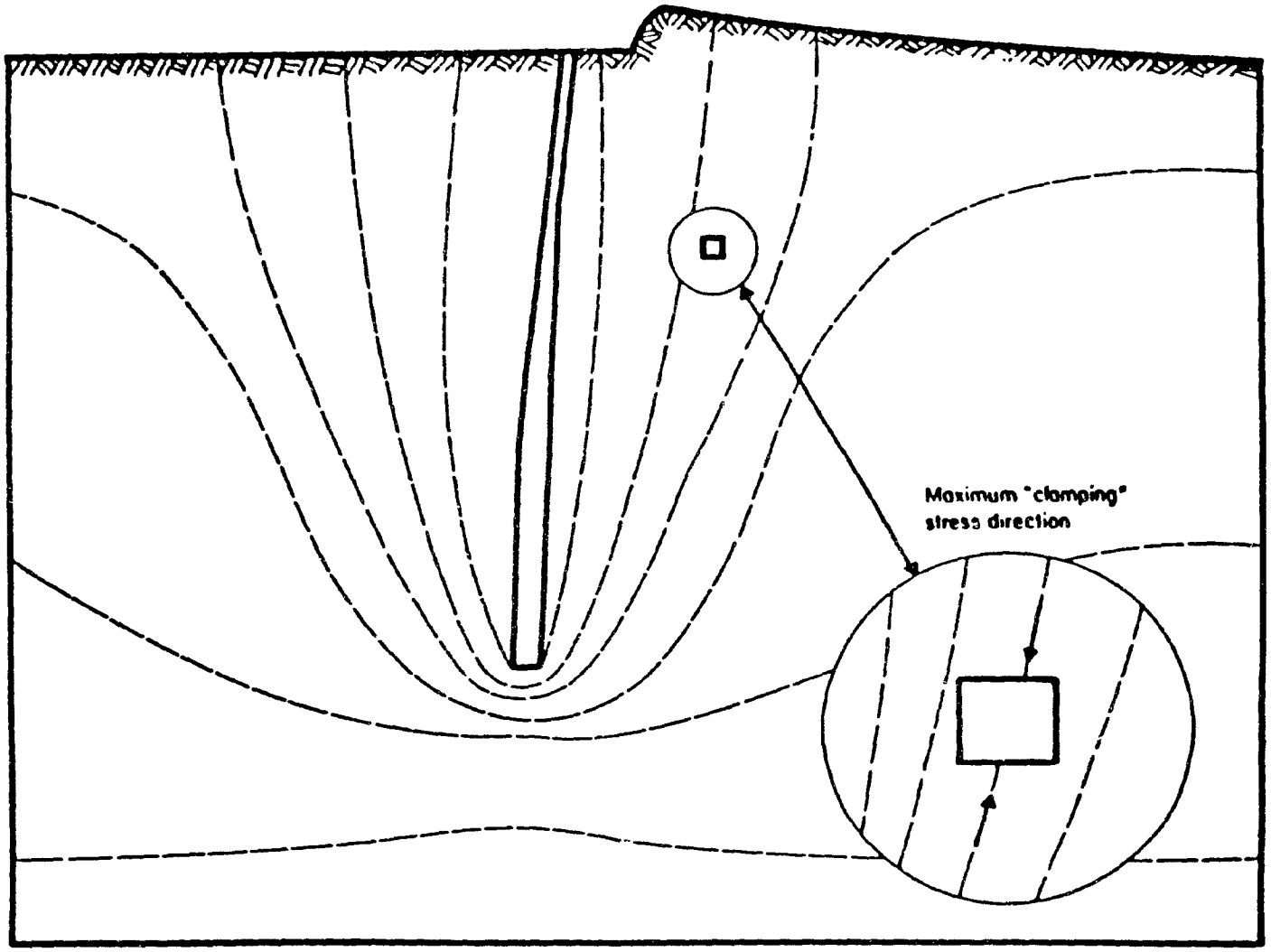


Figure 7.6: Geco Conveyor Drift Stability Analysis, Minimum Principal Stress Directions, Stope Extended to Surface, (by the NTC)

A summary of the stress history for the stope and the conveyor drift is given in Tables 7.1 and 7.2. At the beginning of the excavation cycle, the conveyor belt drift is quite stable. With the nearby stope at its 1985 position, the maximum stress induced around the conveyor drift is less than 1200 psi (8.3 MPa). This level of stress is insufficient to induce cracking and failure observed in the drift. Figures 7.1, 7.2 and 7.3 show the principal stresses in magnitudes and directions. The principal stresses around the drift appear to line up perpendicular to the foliation, thus confirming them as clamping stresses on the foliation.

As mining progresses to the surface, there is a reduction in the principal stresses sufficient to cause loosening and relaxation around the drift. These observations are depicted in the Figures 7.4, 7.5, and 7.6. Both the major and minor principal stresses decrease in magnitude but are still insufficient to cause failure. The minor principal stresses were slightly tensile around the drift. It is clear that the sides of the stopes are under some tensile stresses while the top and bottom of the stope are under compressive stresses. Figure 7.1 indicates that the bottom left corner of the conveyor drift and the corner opposite to it are under compressive stresses while the other two corners are under tensile stresses.

7.2.4 Hybrid Stress Modelling

This was done as part of the verification aspect of this thesis. The section of the mine analyzed was the same for the SAP2D analysis but the mesh generation was done using the QUADMESH generator described in Appendix A, starting with the initial position as at 1983. The same vertical section that was used by the Noranda Technology Centre in the SAP2D analysis was also used in this case study. It measured approximately 580 m in the horizontal direction by 490 m in the vertical. Displacement and load boundary conditions are shown in Figure 7.7.

Progressive excavation was achieved in four stages as shown in Figures 7.8 to 7.12. The magnitudes and directions of the principal stresses have been plotted for the initial position, first excavation and the final stage of extending the stope to the surface. These are shown in Figures 7.13, 7.14 and 7.15. The changing stress patterns following the excavations are summarized in Tables 7.3 to 7.6.

The inset in Table 7.6 describes the nodes of the conveyor drift. It can be seen that the walls of the stope are in tension (Tables 7.3 and 7.4 when the corner nodes for the initial and first excavation are ignored). The maximum tensile stress occurs towards the lower end of the stope walls and is greater than 6 MPa (870 psi). As stoping progresses, the walls are under sufficient tensile stress to cause spalling and/or massive failure. The floor and the back of the stope are under compressive stresses which increase gradually to a maximum of over 11 MPa (1595 psi) as mining progresses (Table 7.5).

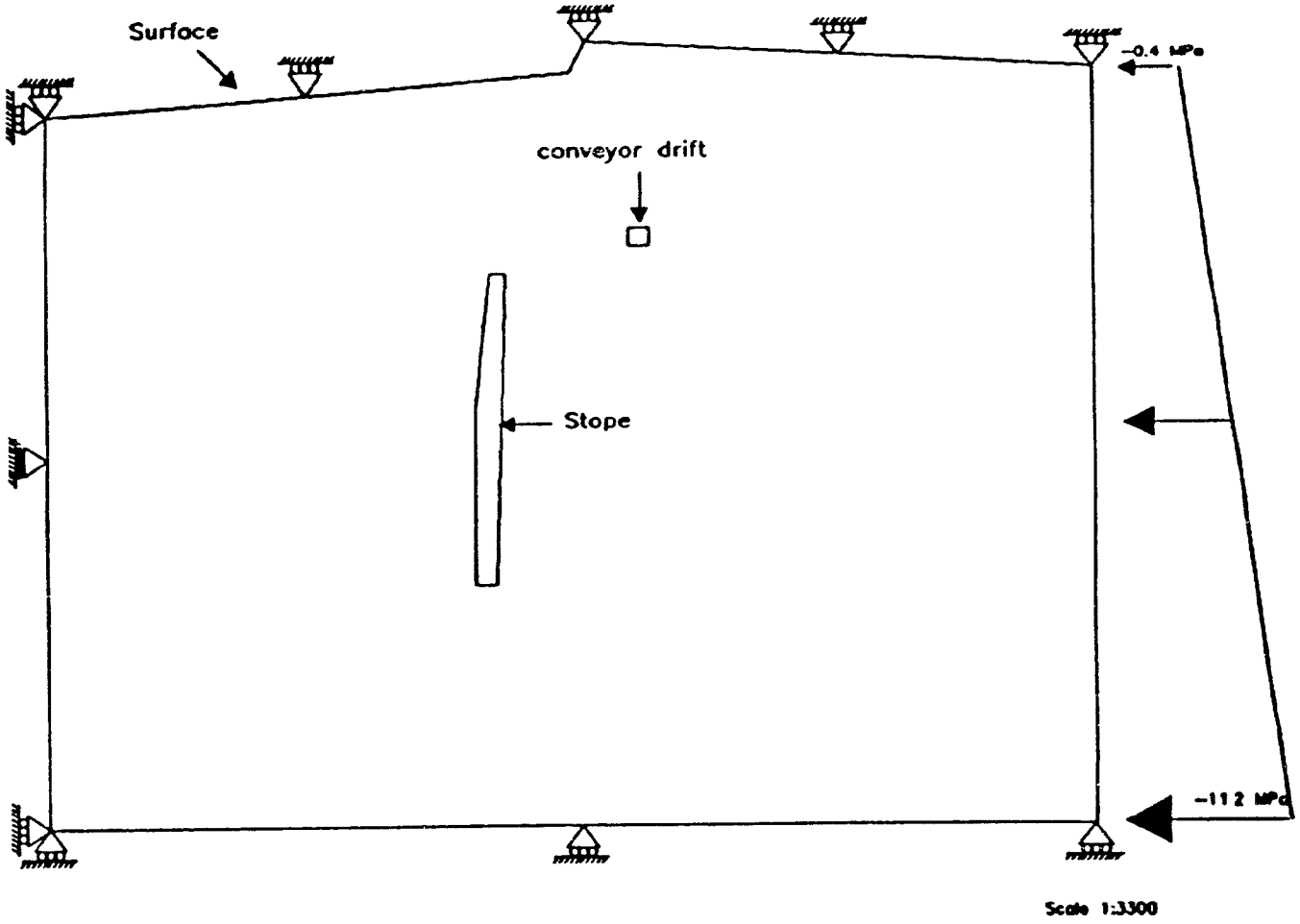


Figure 7.7: Geco Conveyor Drift Stability Analysis, Displacement and Load Boundary Conditions for Hybrid Stress Analysis

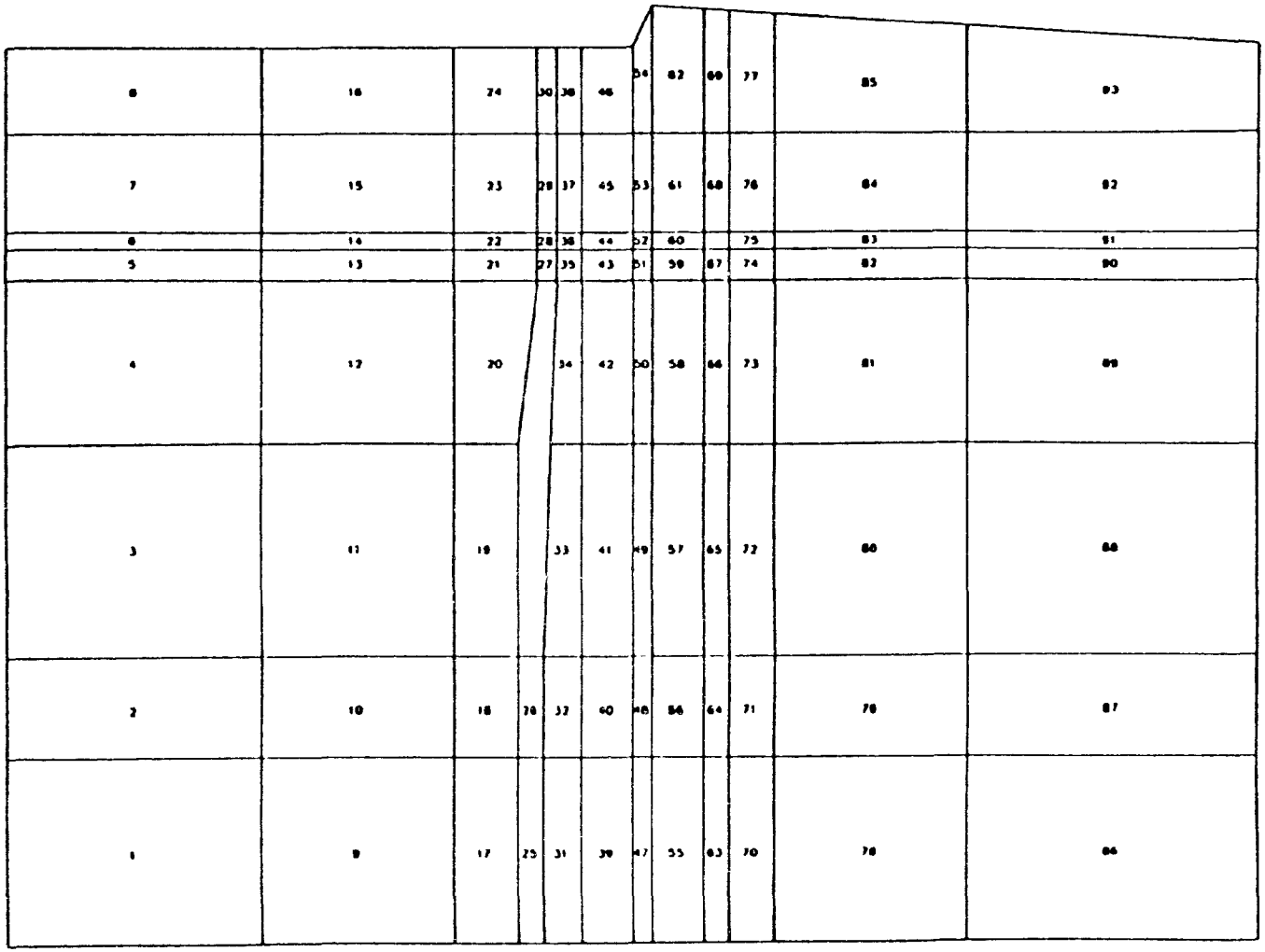


Figure 7.8: Geco Conveyor Drift Stability Analysis,8-Node Hybrid Mesh, Initial Position

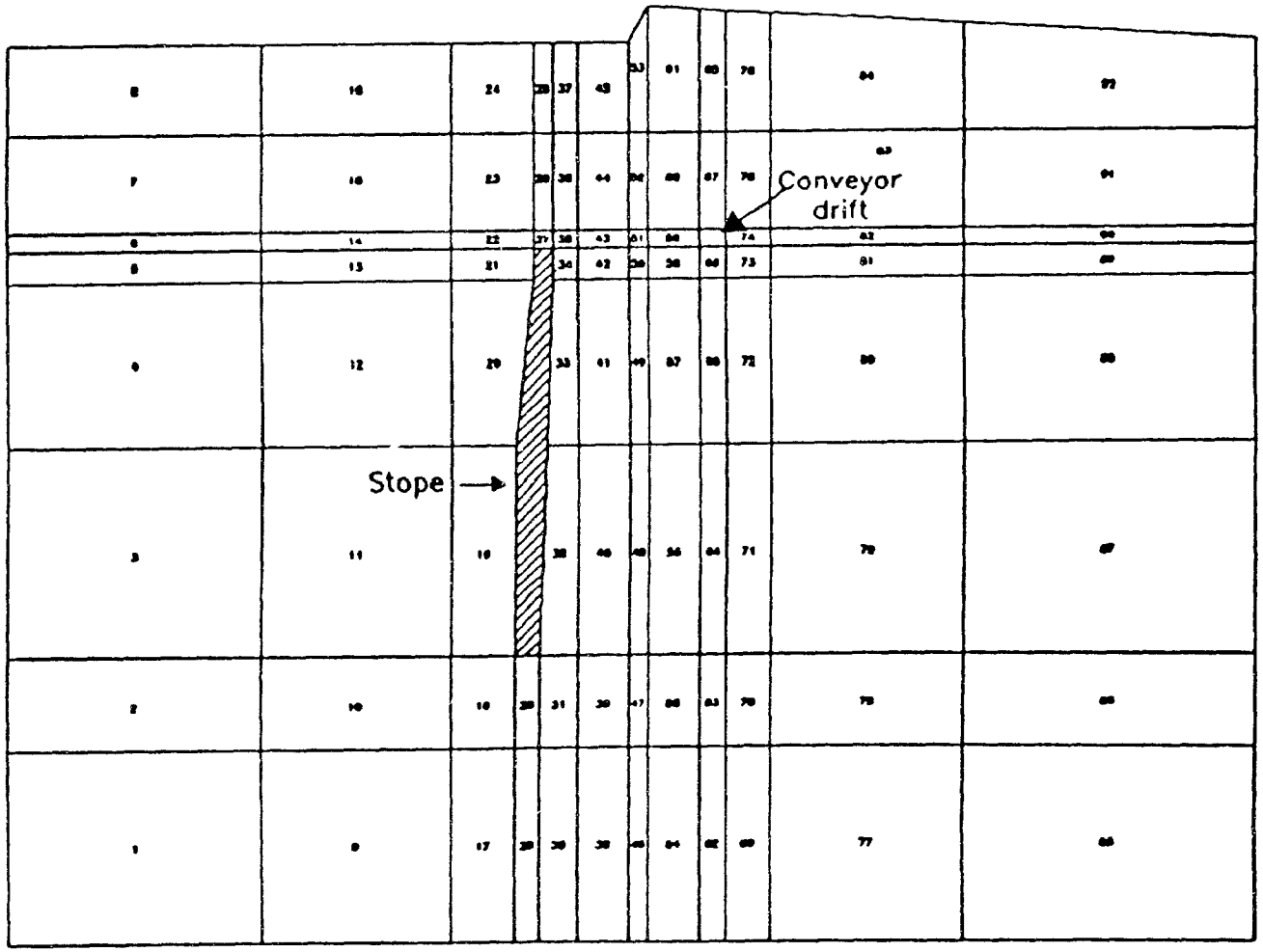


Figure 7.9: Geco Conveyor Drift Stability Analysis,8-Node Hybrid Mesh, First Excavation

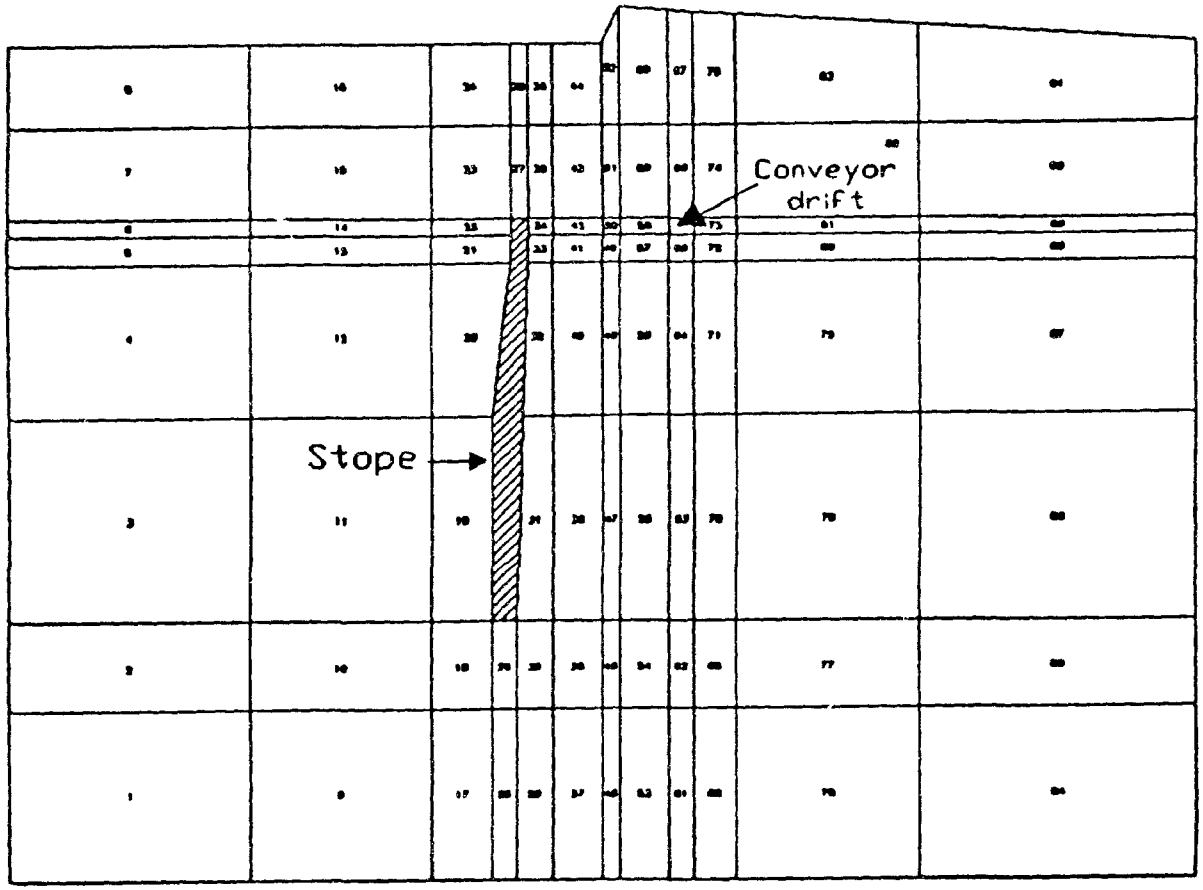


Figure 7.10: Geco Conveyor Drift Stability Analysis,
8-Node Hybrid Mesh, Second Excavation

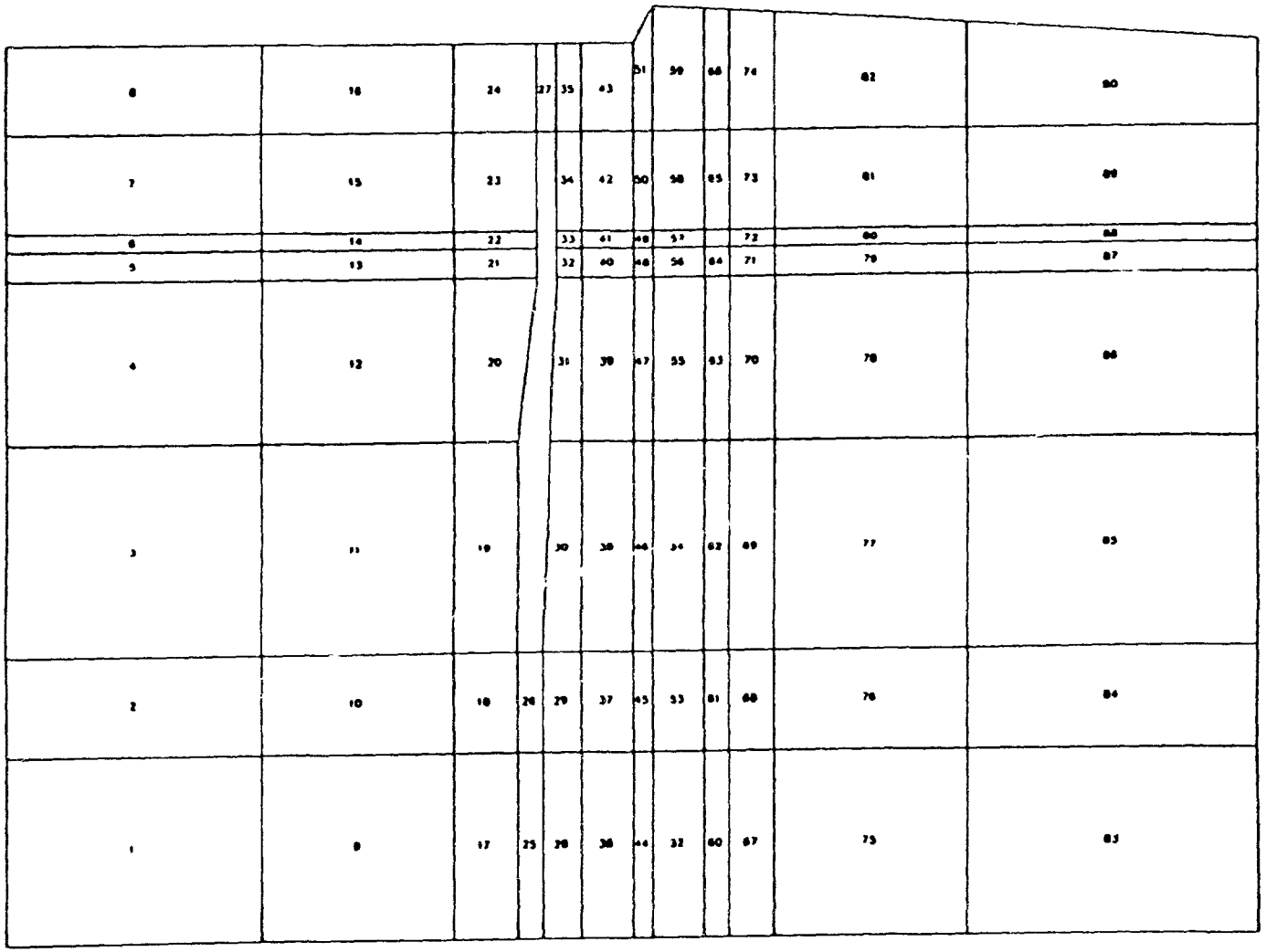


Figure 7.11: Geco Conveyor Drift Stability Analysis,
8-Node Hybrid Mesh, Third Excavation

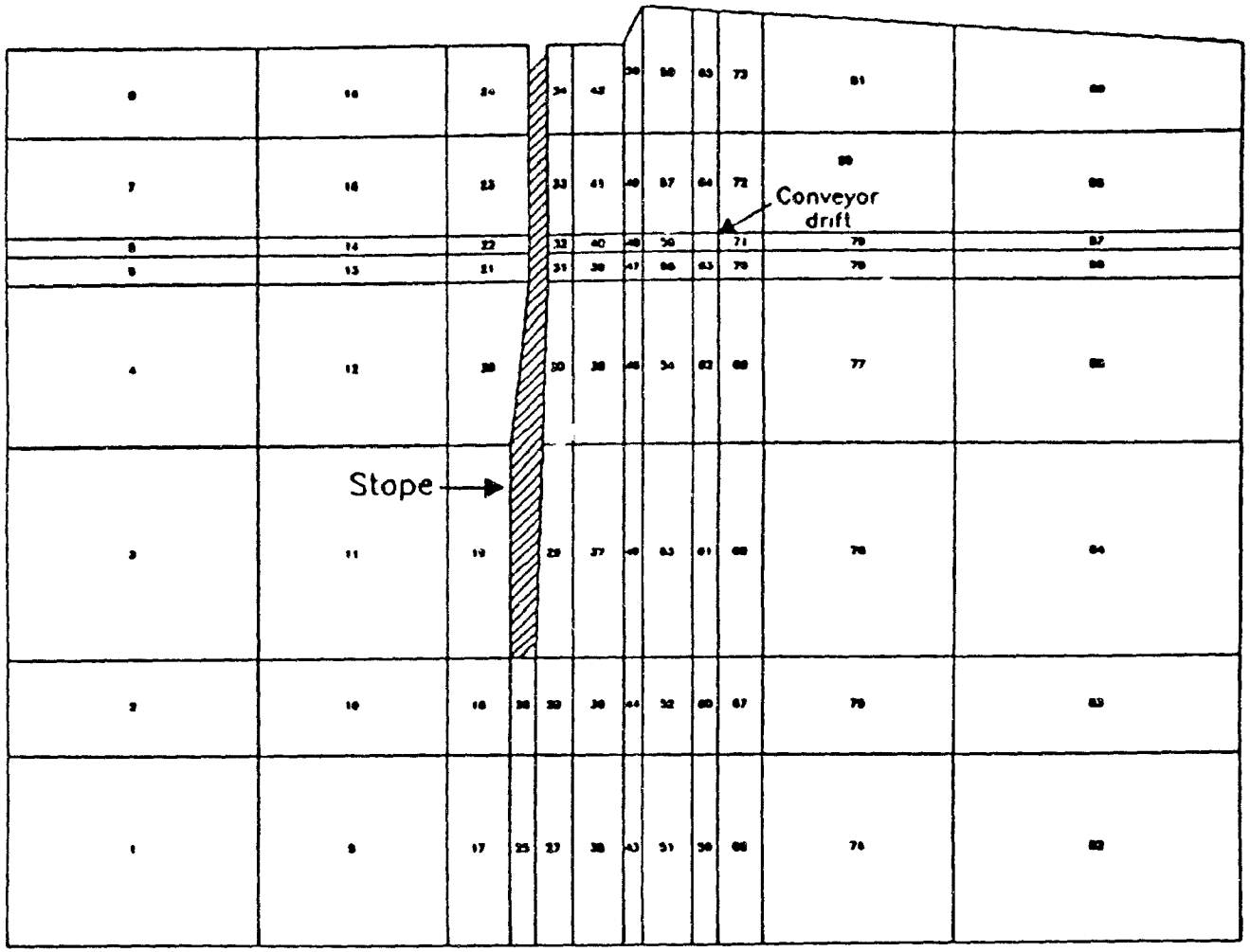


Figure 7.12: Geco Conveyor Drift Stability Analysis,8-Node Hybrid Mesh, Final Position

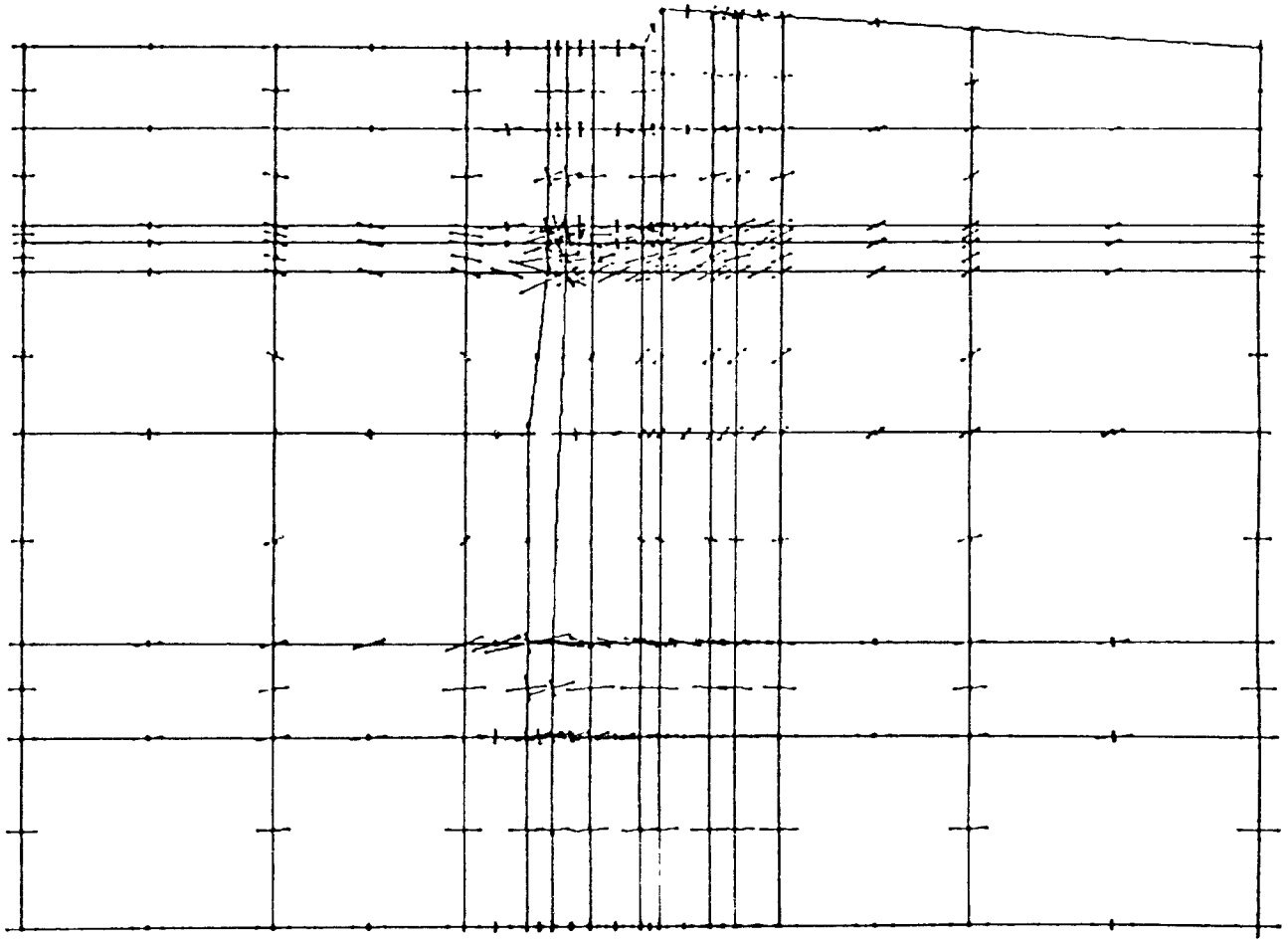


Figure 7.13: Geco Conveyor Drift Stability Analysis,Stress Tensor Plot, Initial Position

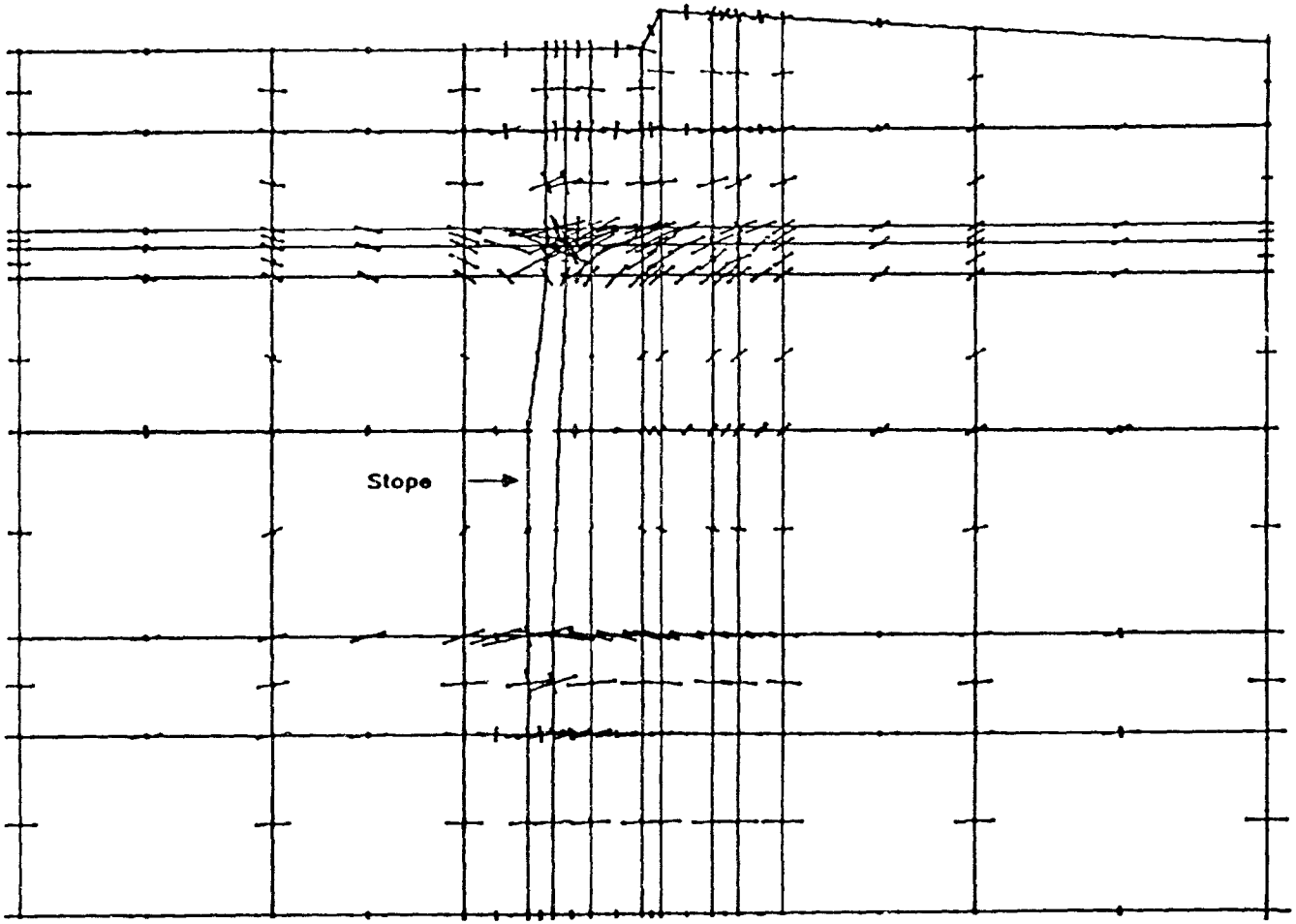


Figure 7.14: Geco Conveyor Drift Stability Analysis,Stress Tensor Plot, First Excavation

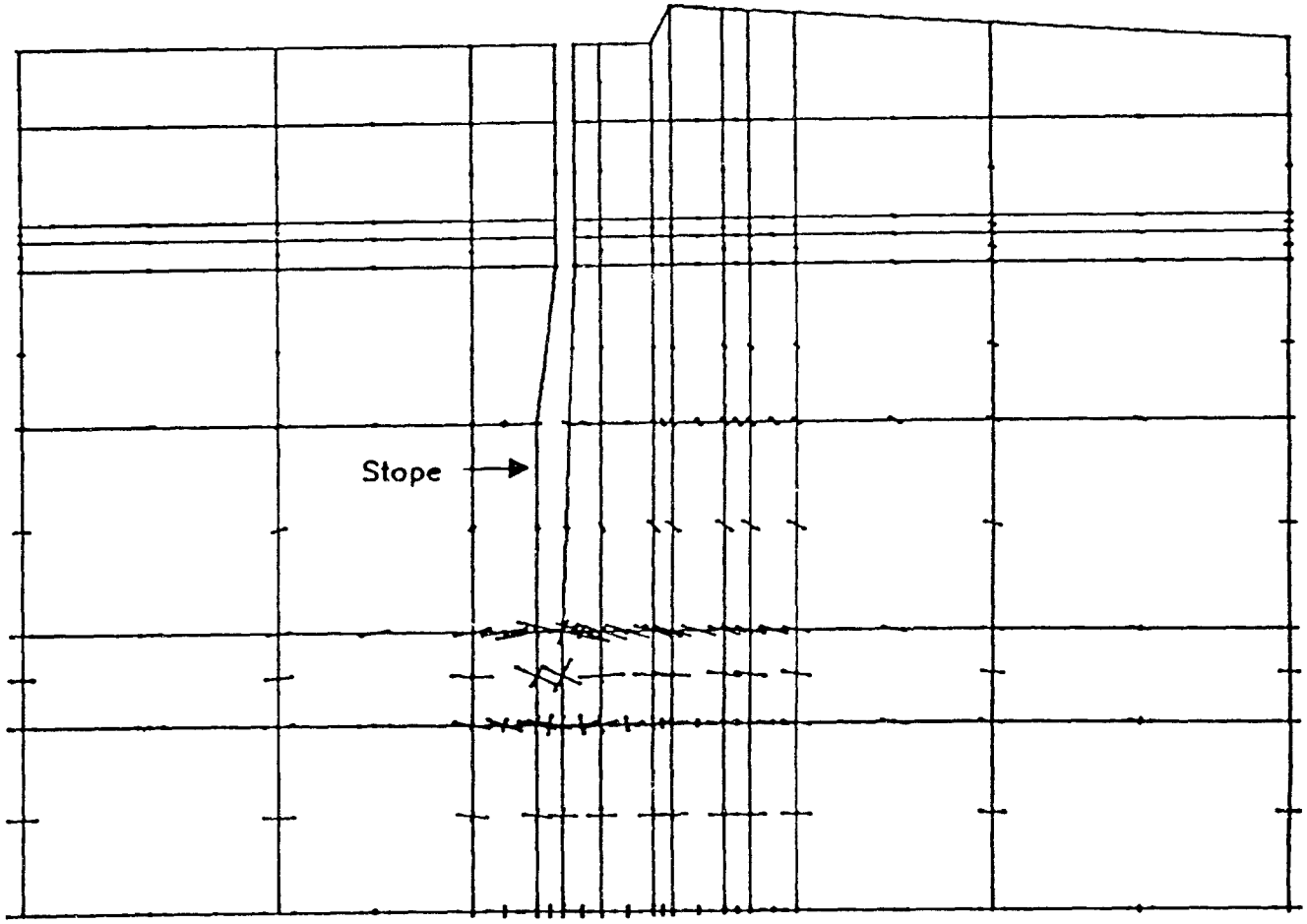


Figure 7.15: Geco Conveyor Drift Stability Analysis,Stress Tensor Plot, Final Position

The conveyor belt drift on the other hand shows a generally small minimum principal stress which is tensile, confirming the results of the SAP2D analysis. However, the maximum principal stress is compressive, starting off at high values and decreasing to near zero as the stope is mined to the surface (Table 7.6).

The general conclusion to be drawn from this analysis is that the hybrid model gives a better and more detailed description of the stress distributions at the various stages of the stoping operation and its effects on the conveyor drift. The findings of the SAP2D analysis have been duplicated.

7.3 Second Case History: Norita Transverse Stopes Stability Analysis

The descriptions in Sections 7.3.1 to 7.3.3 inclusive were taken from open reports written by Chauvin [72] and Goodier and Dube [73] while Section 7.4 is as reported by Bawden and Milne [70] and Stoesser [75] for the Noranda Technology Centre.

7.3.1 History

The Norita deposit, operated by Noranda Inc., Matagami Division, is located 2 kilometers north west of the town of Matagami in northern Quebec. Matagami town itself lies 180 kilometers north of Amos, Quebec, in one of the large volcanic centres in the so called Abitibi greenstone belt of the Superior Province.

7.3.2 Geology

The Norita deposit consists of five massive sulphides lenses containing over 4 million tonnes of ore grading 4.1% Zinc and 1.8% copper Chauvin [72], Goodier and Dube [73]. The orebody was discovered in 1957 but was only brought into production in 1976. Exhaustion of the ore is forecast for 1988. The five ore lenses are interbedded in a precambrian volcanic succession of rhyolitic and basaltic tuffs which strike approximately E-SE with a sub-vertical dip. The sulphide lenses have variable dimensions; the smallest (Upper Zone) has a height of 100 m, a strike length of 150 m and a 6 m average thickness across strike; the largest (A Zone) which is also the deepest, is 400 m in length, 210 m high and has a width of 1.2 m in its top section and over 25 m in its lower portion.

7.3.3 Mining Methods and Ground Control Problems

To the present day, the mine has passed through three distinct phases of ore extraction. These correspond directly to major ground control problems and to minor problems associ-

ated with dilution. The so-called Phase I mining was by the sub level retreat method which was used successfully between 1975 and 1978 in mining the Main, Lower and Marker Zones. This method was used in the "A" Zone which was discovered in 1977 and which appeared well suited for this type of mining.

The first sign of ground control problems was noticed in 1978 in the form of ground movements and spalling in the sill pillar drift of the 8-8 level, Bawden and Milne [74], and Stoesser [75]. Various actions taken to combat this deterioration included introducing extra ground support, lowering blast vibrations and thus decreasing pillar damage, monitoring ground movement and stress changes and computer simulations of the mining sequence. In addition, all development drifts were arched so as to improve their stability and lower stress build-up around them.

The mining methods changed from sublevel retreat to a modified sublevel caving and later to a transverse longhole open stoping which is still in use. Access to the stopes is from the footwall drift. Waste fill was introduced at the top of the "A" Zone to give some support to the walls. This so-called Phase II mining cycle began about 1981 with the development of a waste stope above the "A" Zone. The first waste stope was lost as a result of severe sloughing of the north and south walls of the 8-7 sub level which led to the loss of the waste dumping points and the remaining two production sub levels of the Marker Zone. A smaller waste stope was developed above the Marker Zone. For simulation purpose, the stopes were assumed to be 15 m wide rectangles.

7.3.4 Instrumentation and Modelling by Noranda Technology Centre

As a result of the deteriorating ground control situation, Noranda Technology Centre was consulted in 1985. Initial solution was to install extensometers and stress meters to quantify the visual ground movements in the sill pillar, shaft pillar and central pillar areas. Following extensive monitoring, two types of numerical modelling were done, using the MINTAB and the Boundary Element Analysis (BEA) programs.

The MINTAB program was used to simulate the extraction of the ore zone along the strike on the vertical plane. Initial runs were done with the entire mine as the model but after a few simulations it was apparent that mining of the "A" Zone had little effect on the remainder of the levels above it. Further modelling was done on the "A" Zone only. The model was correlated to the caving which had occurred before the mining method was changed.

The BEA program was used to complete the model in the third direction, i.e., along the strike in a horizontal direction. The computer runs indicated that the pillars between the

open stopes had failed, an observation that correlated well with the MINTAB runs and the site observations. In these runs, failure was based on the strength of the various rock types. If the stress was above the strength of the rock, that rock would be deemed to have failed.

The two models used - MINTAB and BEA, have serious limitations some of which are:

1. no multiple materials;
2. no discrete geologic structures;
3. no "post failure" behaviour;
4. no stress shedding from failed zones, and
5. only limited allowable movement of excavation boundaries.

However, they still allow a reasonable determination of the elastic stress distribution around mine openings.

Modelling the transverse stoping for the lower A zone to about 1986 indicated that induced stresses between the 9 and 10 levels, where the extraction ratio was low, would be predicted at around 60-80 MPa while in the lower 10-11 levels where the extraction ratios were somewhat higher, stresses would be predicted between 80 and 100 MPa. The proposed transverse mining sequence is shown in Figure 7.16.

The "lower stopes" refer to stopes between the 10 and 11 levels while the "upper stopes" refer to stopes between the 9 and 10 levels. The proposed sequence is then ordered as given below:

step 1 lower stopes 6, 14, 8, 12

step 2 upper stopes 6, 14

step 3 lower stopes 16, 4

step 4 upper stope 16

step 5 lower stope 2

step 6 upper stopes 4, 2

MINTAB simulations of above stoping sequence showed that at each step of the excavation, stress shedding to adjacent pillars occurred, often beyond the estimated strength of the pillars. The most highly stressed areas reached peak stress levels of up to 130 MPa as shown in Figure 7.17 which shows the stress distribution after mining of the upper stopes 6

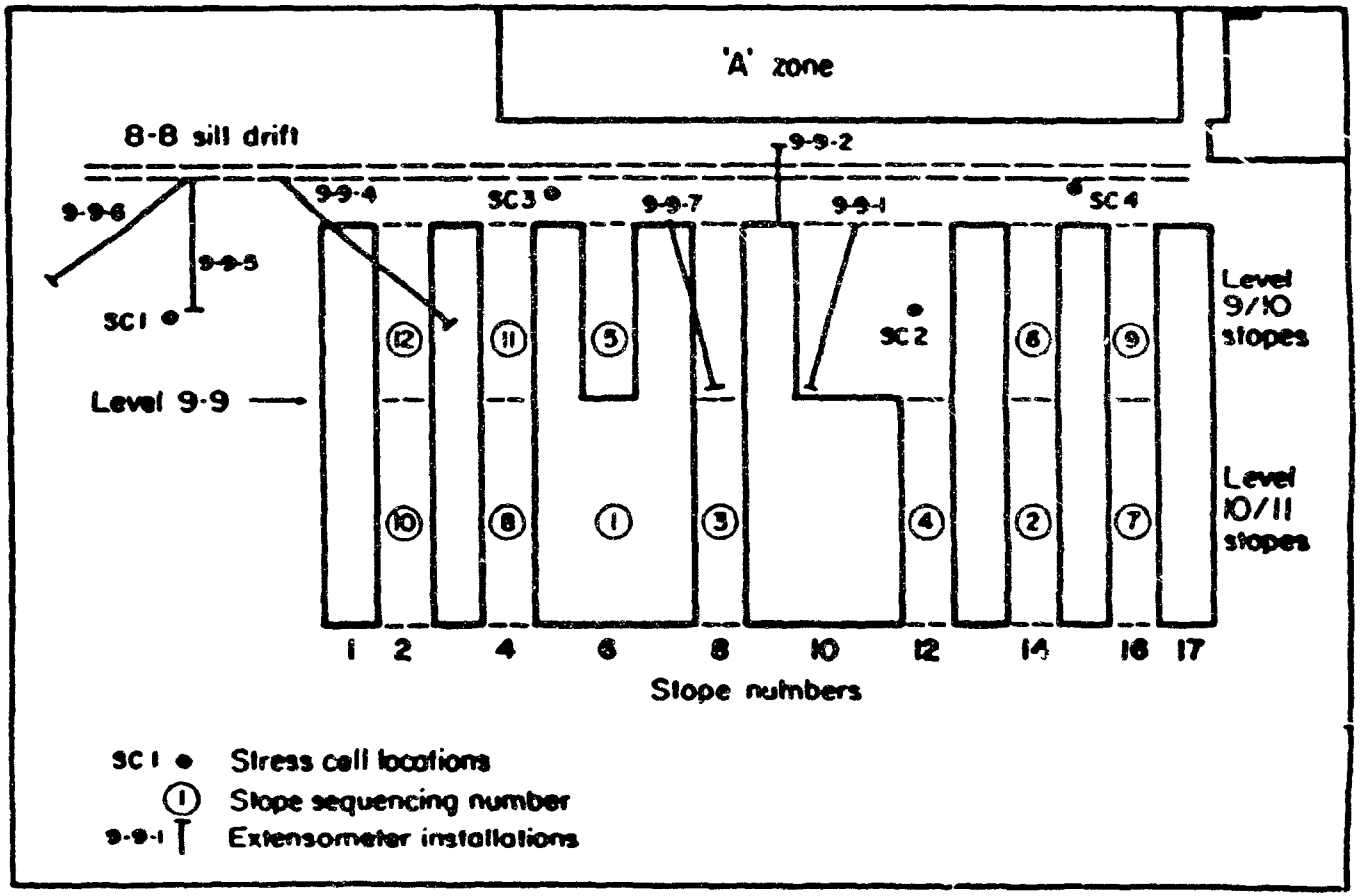


Figure 7.16: Norita Transverse Stope Stability Analysis, Stope Sequencing and Instrumentation, (by the NTC)

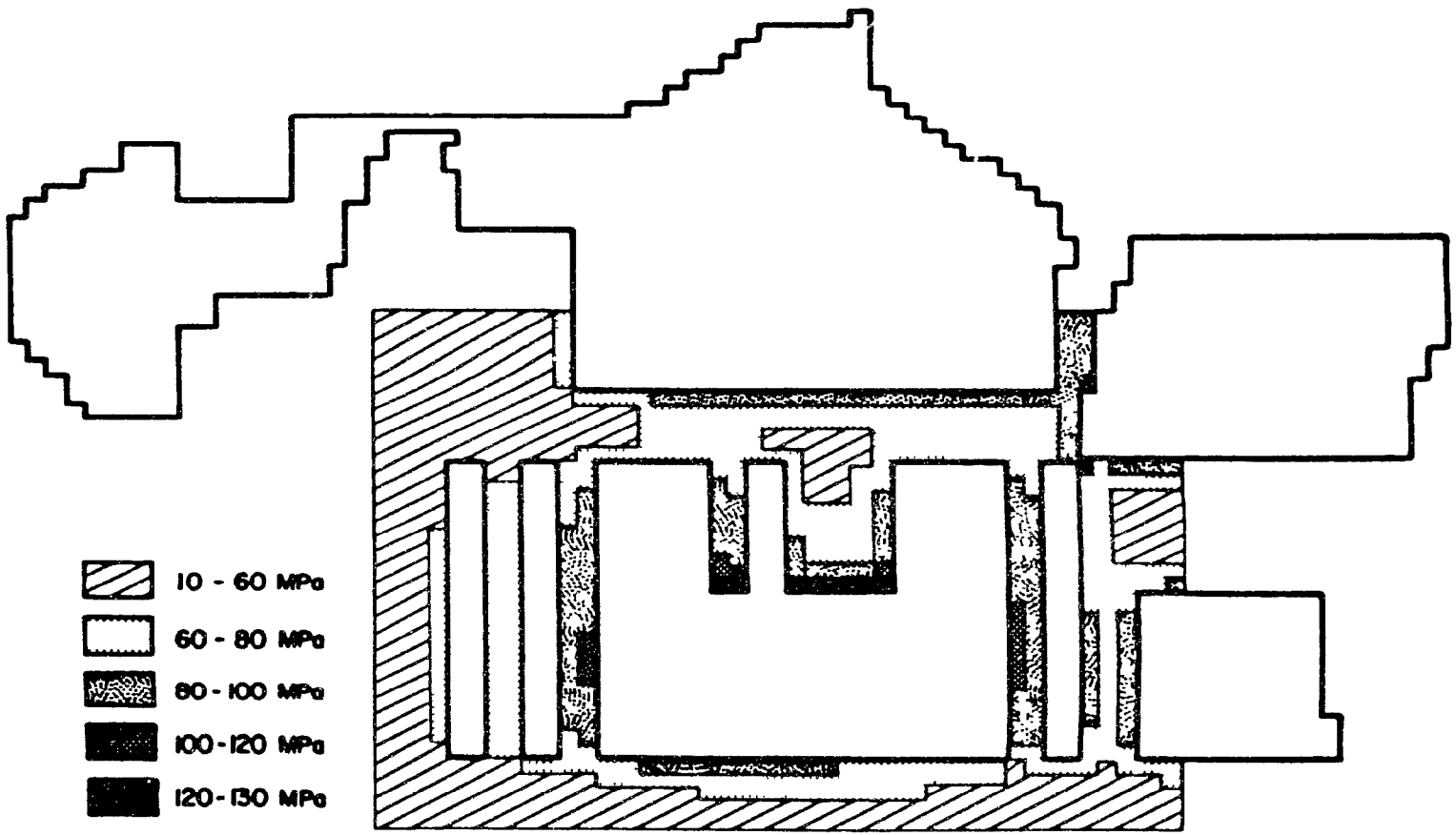


Figure 7.17: Norita Transverse Stope Stability Analysis, Mining of Upper Stopes 6,& 14, and Lower Stopes 6,8,14 & 12. (by the NTC)

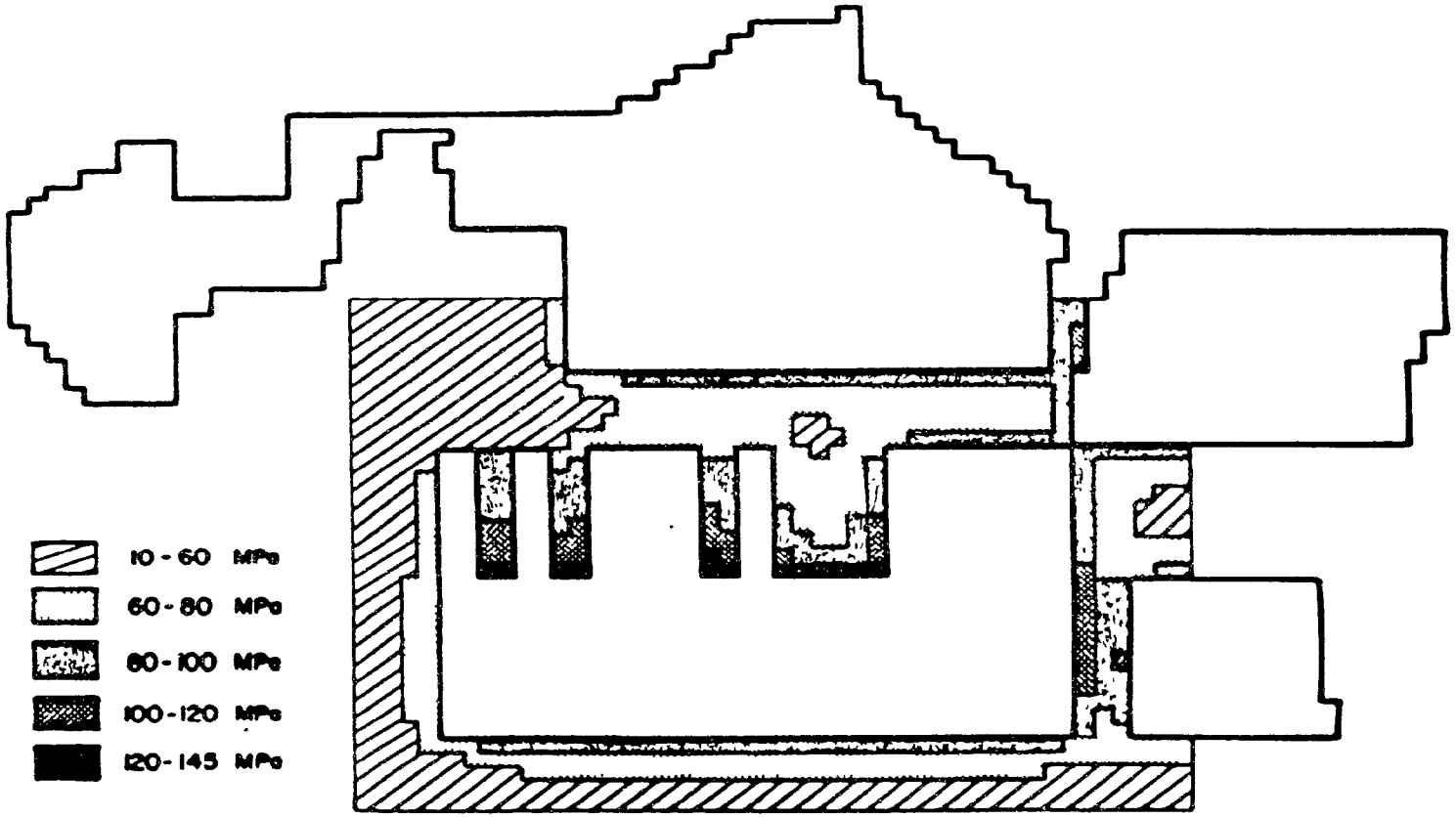


Figure 7.18: Norita Transverse Stope Stability Analysis, Mining of Upper Stopes 16 and all Lower Stopes. (by the NTC)

and 14, and the lower stopes 6, 8, 14 and 12.

Figure 7.18 shows the MINTAB model results for excavation of all the stopes up to step 5. At this stage, all of the lower 10/11 stopes have been excavated and backfilled but backfill properties were never included in the modelling. At this stage, the upper level pillars 2 and 4 have become very highly stressed, as high as 140 MPa. It is certain that these pillars would have failed and shed most of this stress to the east abutment and overlying sill pillar by this time. Stress levels in the centre of the main upper level pillar (level 9/10, pillars 8-12), however, remain relatively low, indicating that the core of the pillar remains intact.

7.3.5 Hybrid Stress Modelling

The hybrid stress finite element program was used to run simulations similar to the BEA runs described above. The hybrid model is a horizontal section taken through the lower level 10/11 stopes from Figure 7.16. The section modelled included the shaft and all the 17 stopes and pillars as shown in Figure 7.19 which also shows the boundary conditions imposed on the system.

Comparison of Figures 7.16 and 7.19 shows that the stope sequencing of the hybrid model follows the same order as for the MINTAB runs conducted by the Noranda Technology Centre. The hybrid modelling was done in four steps as given below:

step 1 Initial position with only the shaft excavated.

step 2 Position as at June 1986, with stopes 1, 3, 9, 10, 11, 13, 15 and 17 of Figure 7.19 excavated.

step 3 Lower stopes 6, 14, 8, 12 (sequence 1, 2, 3, and 4) excavated, and finally,

step 4 Lower stopes 16, 4 (sequence 5, 6, and 7) excavated.

The section analyzed has three principal material types with the massive sulphide orebody bounded on the south side by rhyolitic tuffs and on the north side by basaltic tuffs. Material properties had been determined in-situ from rock mass classification and from the laboratory from tests on intact rock cores by the NTC. As expected, there was a very large disparity in the properties with the laboratory measured values about an order of magnitude higher. The values used in the Hybrid simulation fall between the two extremes and are shown in Table 7.7.

The hybrid stress simulation of the mining of the lower stopes was carried out in four stages as described below. The first simulation, described the initial position which was assumed to begin with only the shaft in place. The second simulation described the position

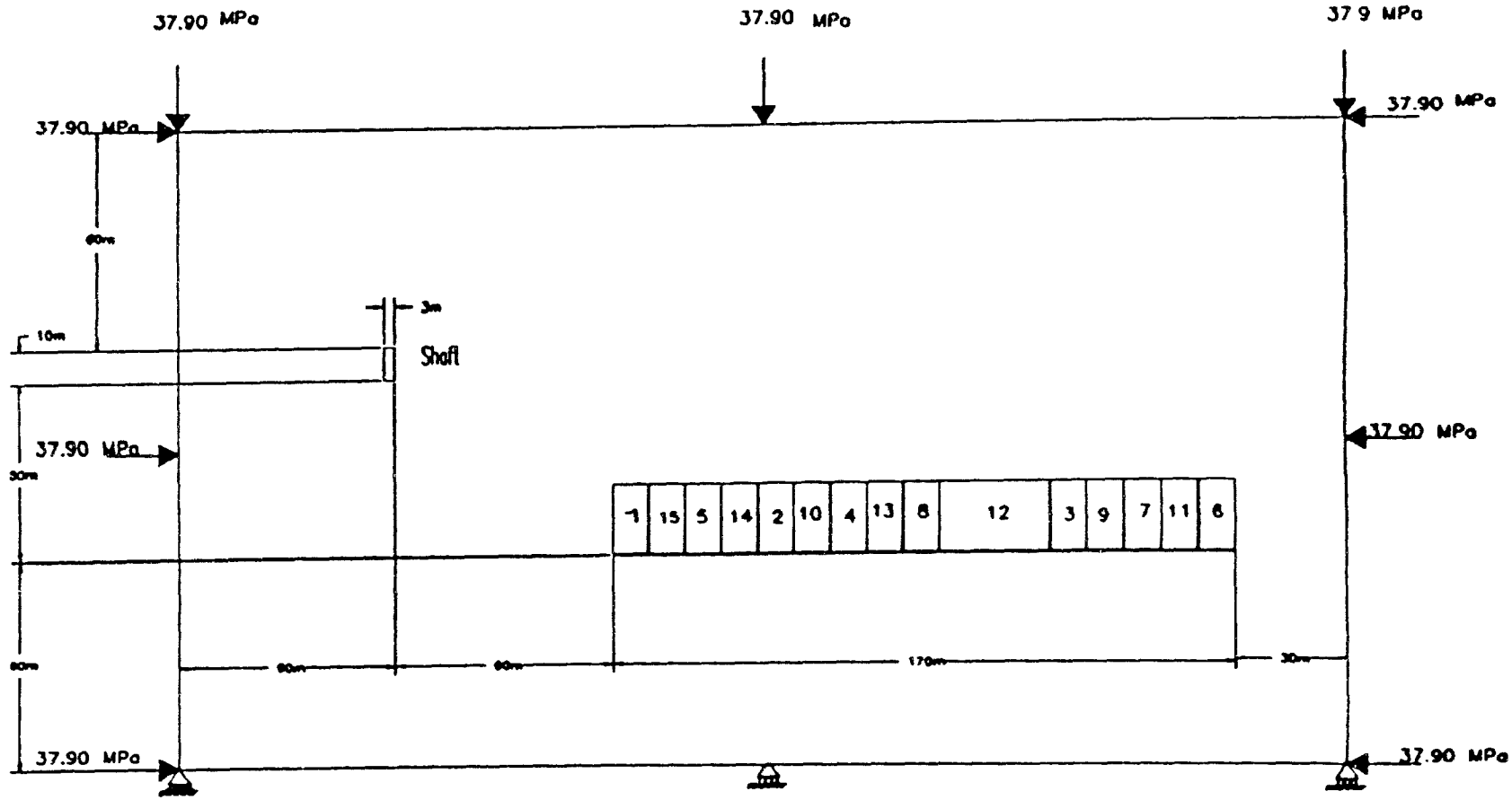


Figure 7.19: Norita Transverse Stope Stability Analysis,
Displacement and Load Boundary Conditions for Hybrid Stress Analysis

of the transverse stopes as at June 1986 and corresponded with the MINTAB simulation described above. During this step, stopes numbered 1, 3,9, 10, 11, 13, 15 and 17 were mined. The third run corresponded to the excavation of the lower stopes 6,14, 8 and 12 (stope equence 1, 2, 3 and 4). In the last simulation, the remaining lower stopes 16, 4 and 2 (sequence 5, 6 and 7) were taken. Figures 7.20 to 7.23 inclusive show the finite element meshes that were used at the various stages.

The original mesh covering the discretized zone without any excavation was generated using the QUADMESH program described in Appendix A. Meshes representing the various steps of the modelling (Figures 7.20 - 7 23) were then generated using the companion program REVISE to alter the original mesh. The program REVISE essentially deletes the nodes and elements in a specified window of a given mesh and reorders the remaining nodes and elements. Three material properties were used corresponding to the specification in the reference [70].

The results for the lower stopes are summarized in Tables 7.9 to 7.12. The critical areas for which stability was analyzed are the shaft, the west abutment, the east abutment and stope 2 and 16 which were not taken during the transverse stoping as indicated in the tables. The stress tensor plots for each step of the simulation are shown in Figures 7.24 to 7.27 inclusive.

The critical areas examined are best correlated to the MINTAB results with respect to the abutments and the pillars. An examination of these results shows that the hybrid stress results are slightly higher in magnitude than the values obtained by the MINTAB and BEA methods but the trend is essentially the same and leads to the following observations:

The Shaft Pillar

Table 7.9 shows the state of stress of the shaft region during the progress of the excavation. The shaft is a narrow slit about 3m wide by 8m long, the west and east walls representing the long sides while the north and south walls represent the short sides. It can be surmised from the theory of elasticity that there will be a high stress concentration therefore on the north and south rockmass surrounding the pillar. This is borne out by the results in Table 7.9. In general, the stress history shows a peak at about the start of the retreat cycle (June, 1986). The tensile stresses are very small compared to the compressive stresses, a maximum value of 11 MPa occurring in the north and south walls as of June 1986. Against this small tensile stress, everywhere else is under high compressive stresses. It is clear from comparison of these stresses with the strength values of the various rock types in Table 7.8 that the shaft is not endangered by the transverse stoping although the compressive stresses in the north and south walls are rather high. These results do not differ from the MINTAB runs.

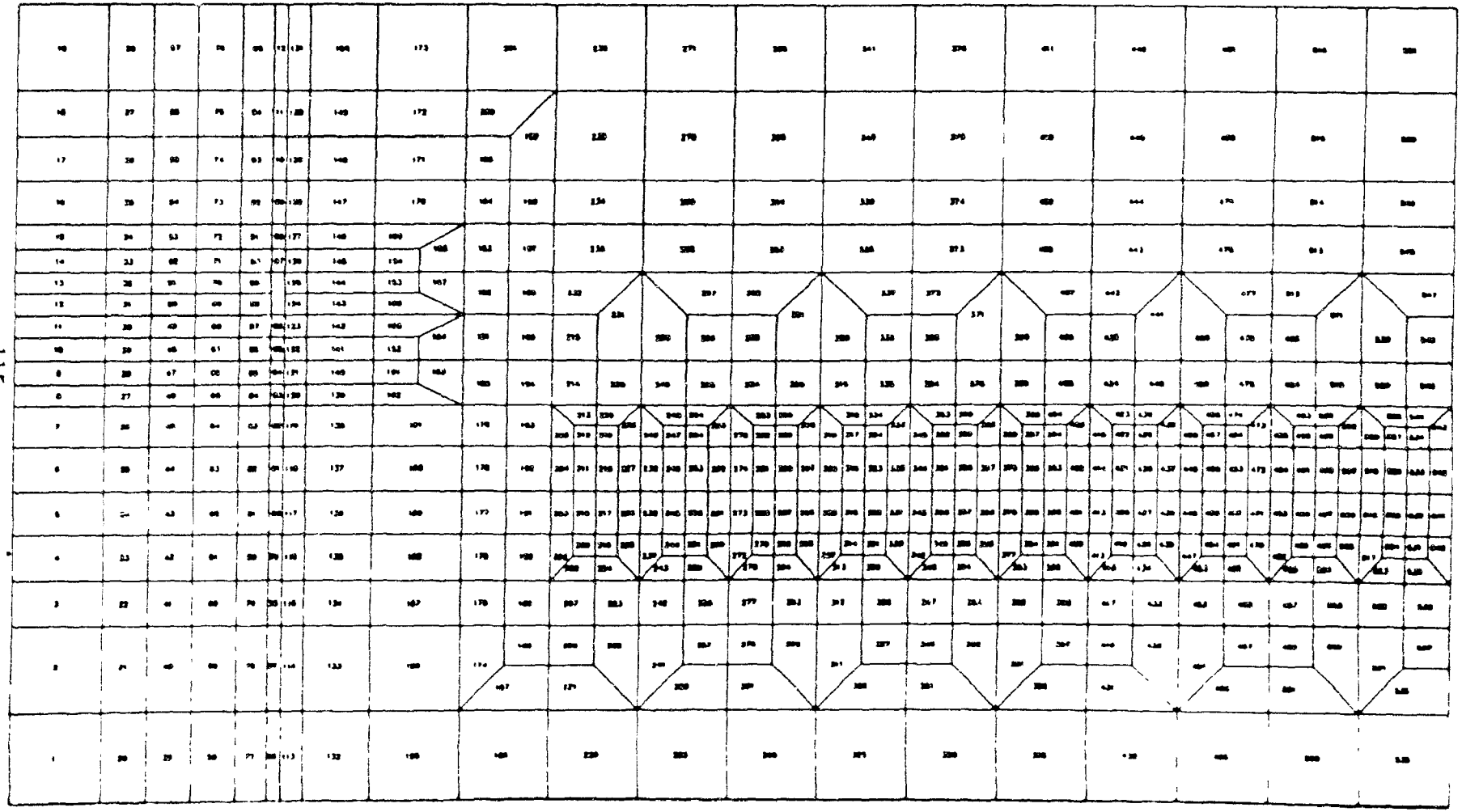


Figure 7.20. Norita Transverse Stope Stability Analysis,
Hybrid Stress Mesh for Initial Position with Shaft Excavated

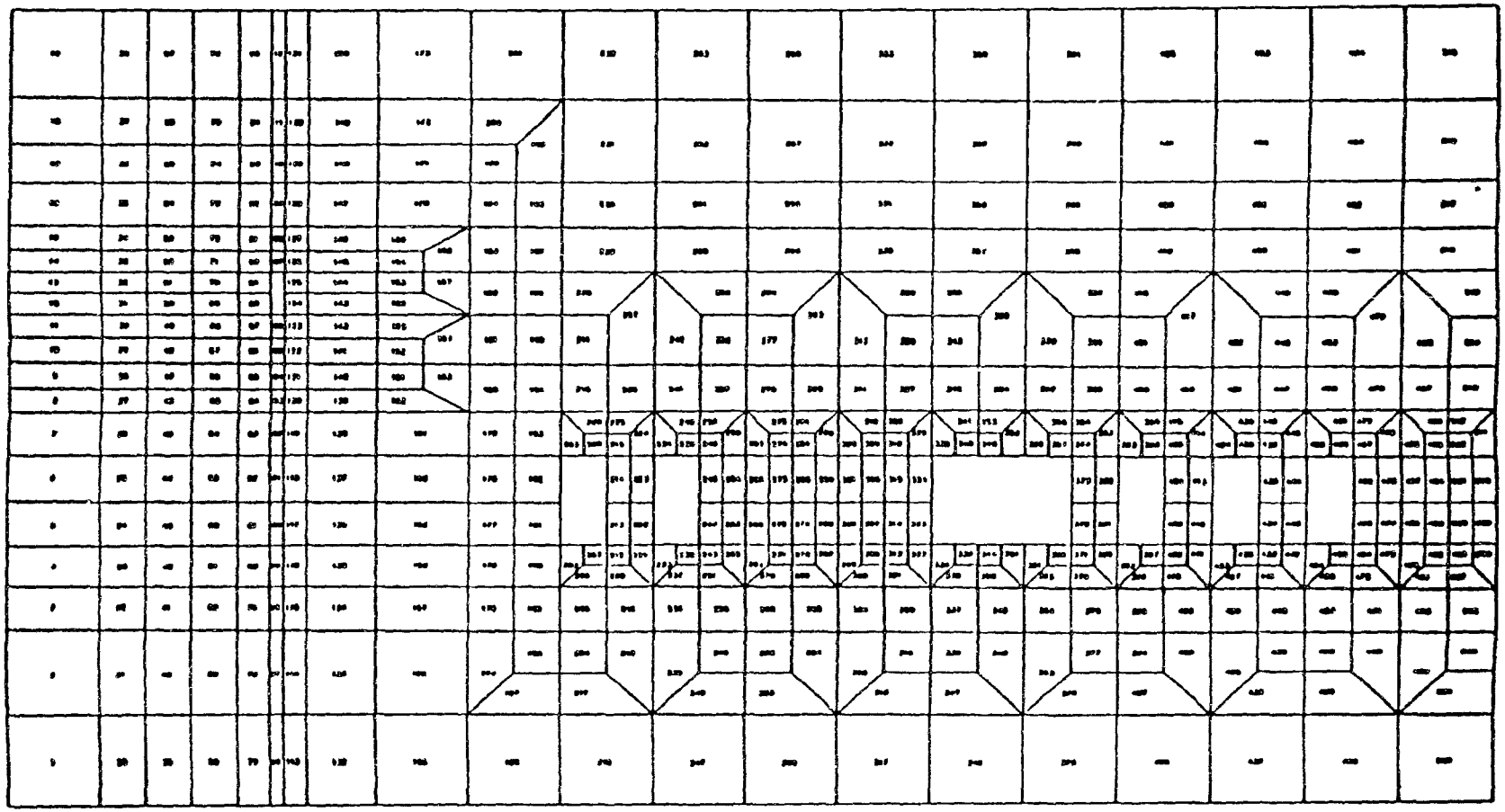


Figure 7.21: Norita Transverse Stope Stability Analysis,
Hybrid Stress Mesh for Position as at June 1986

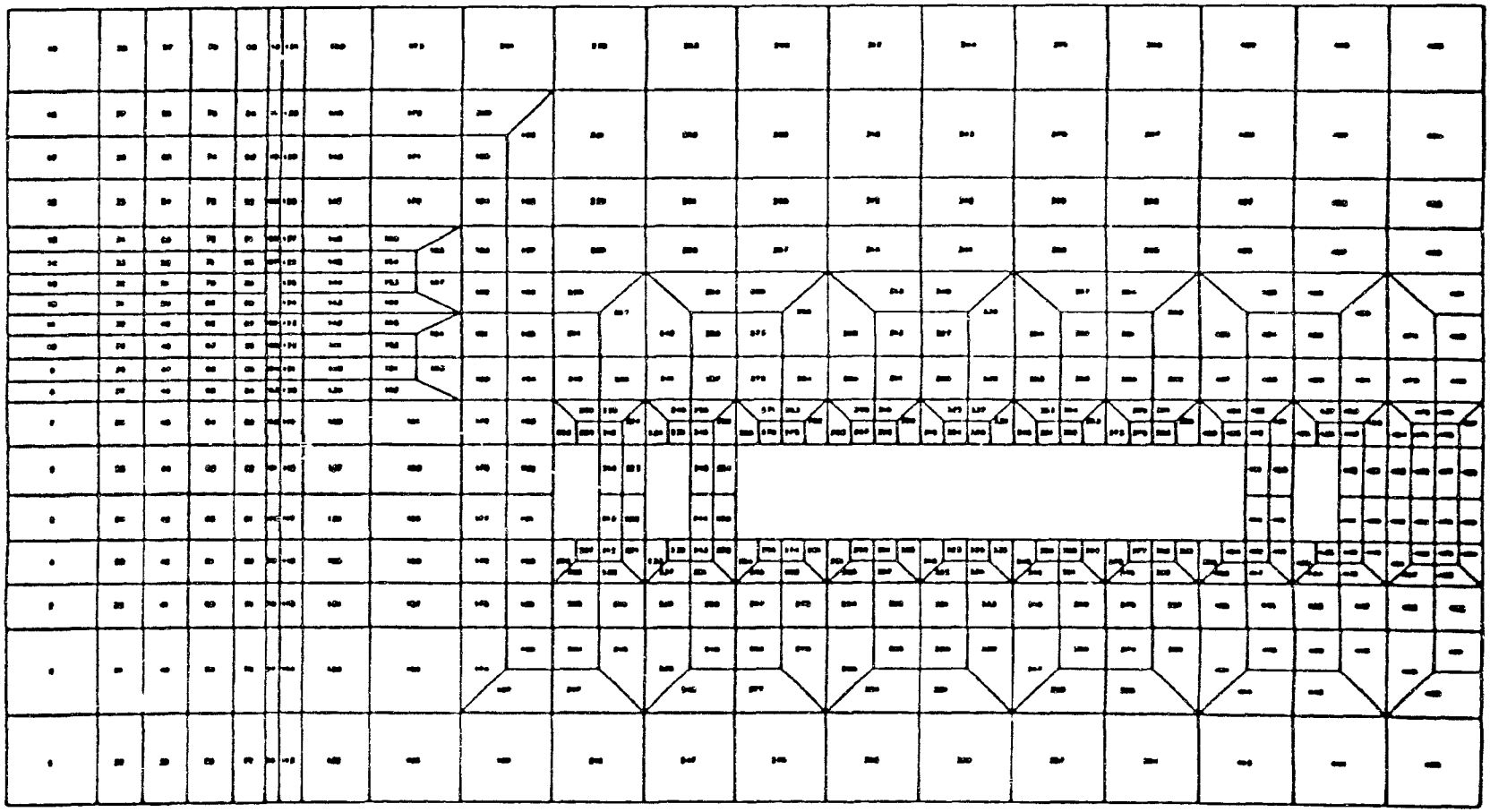


Figure 7.22: Norita Transverse Stope Stability Analysis,
Hybrid Stress Mesh for Stopes 6, 8, 12 and 14 Excavated

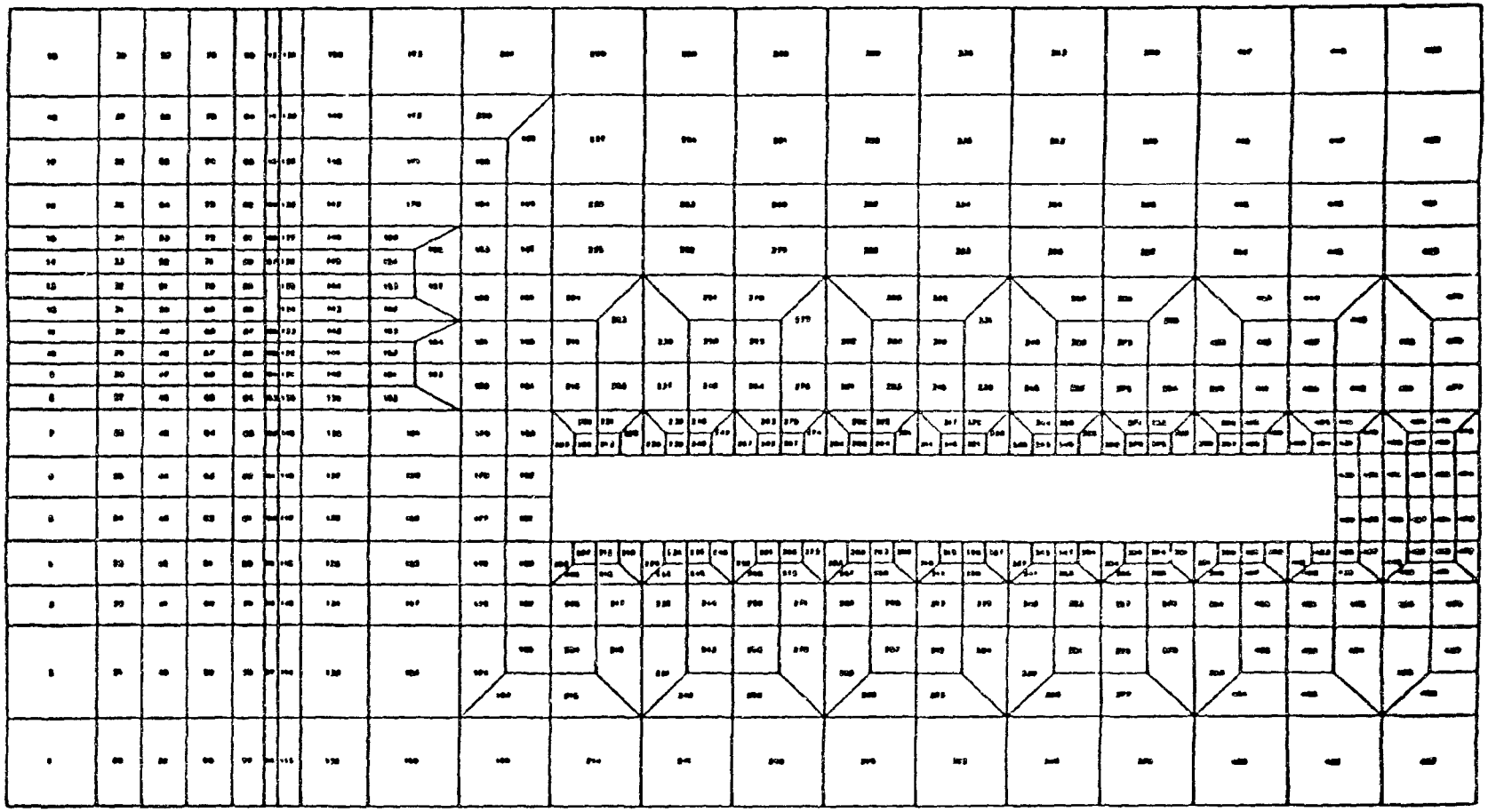


Figure 7.23: Norita Transverse Stope Stability Analysis,
Hybrid Stress Mesh, all Lower Stopes Excavated

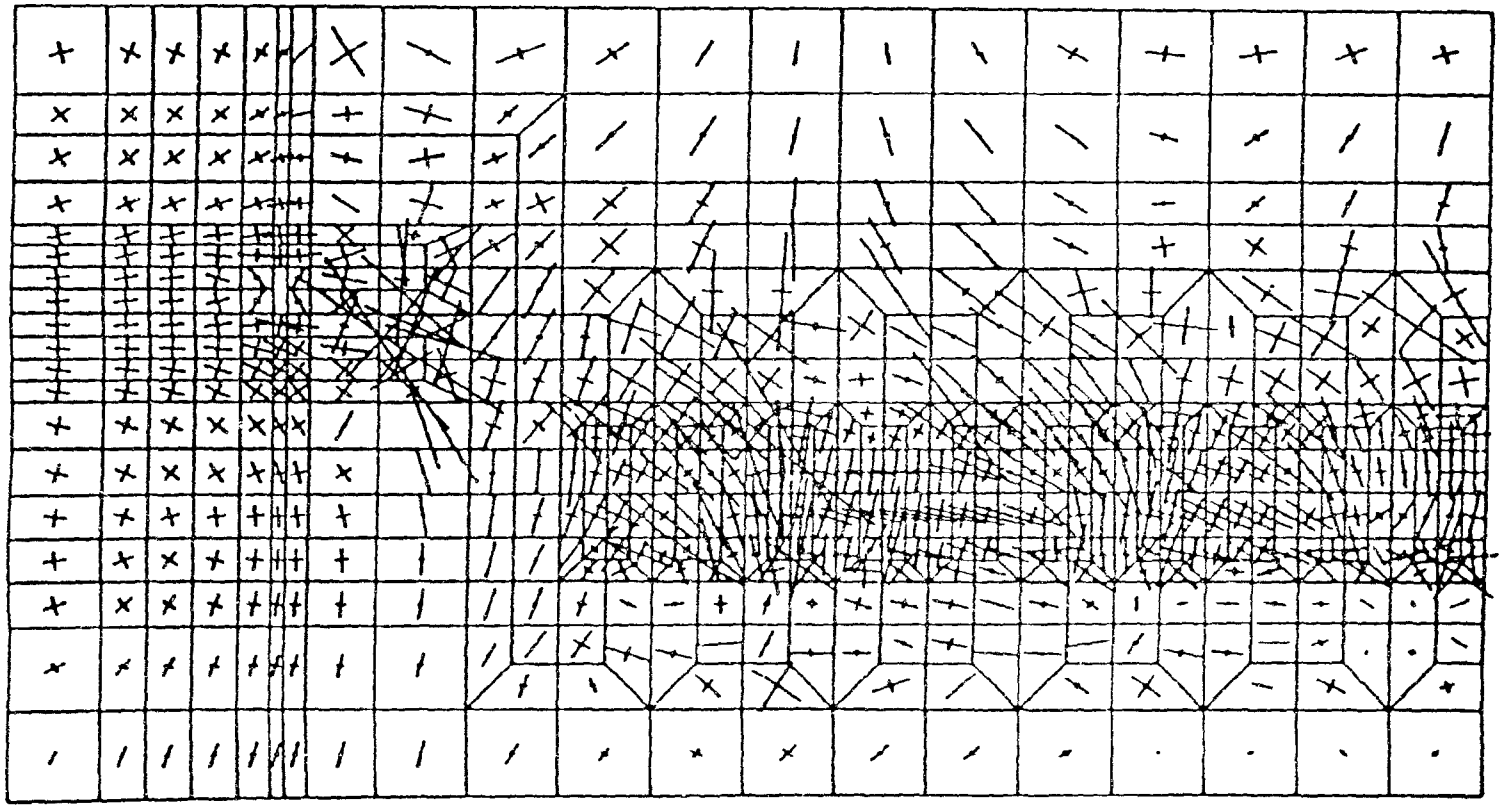


Figure 7.24: Norita Transverse Stope Stability Analysis,
Stress Tensor Plot for Initial Position with Shaft Excavated

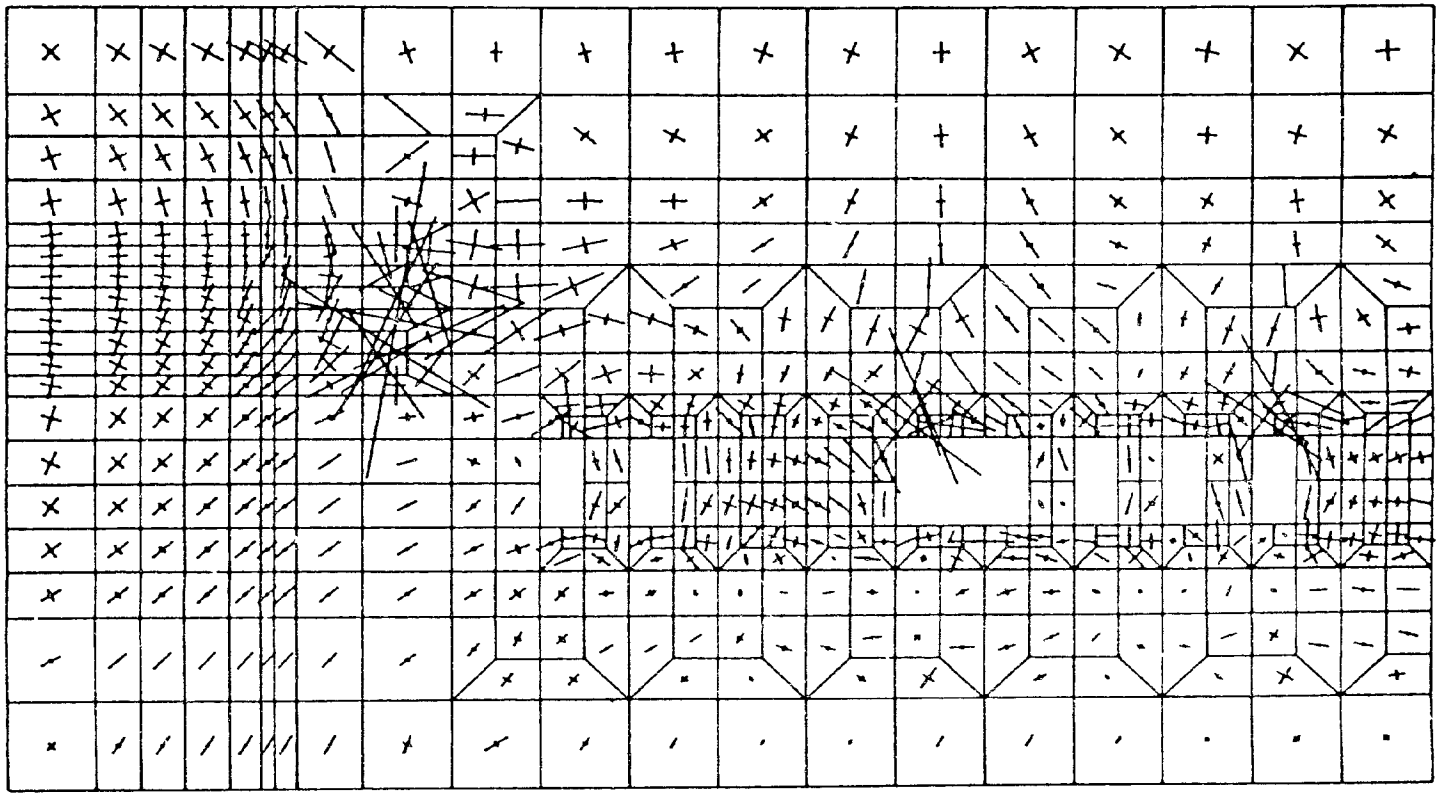


Figure 7.25: Norita Transverse Stope Stability Analysis,
Stress Tensor Plot for Position as at June 1986

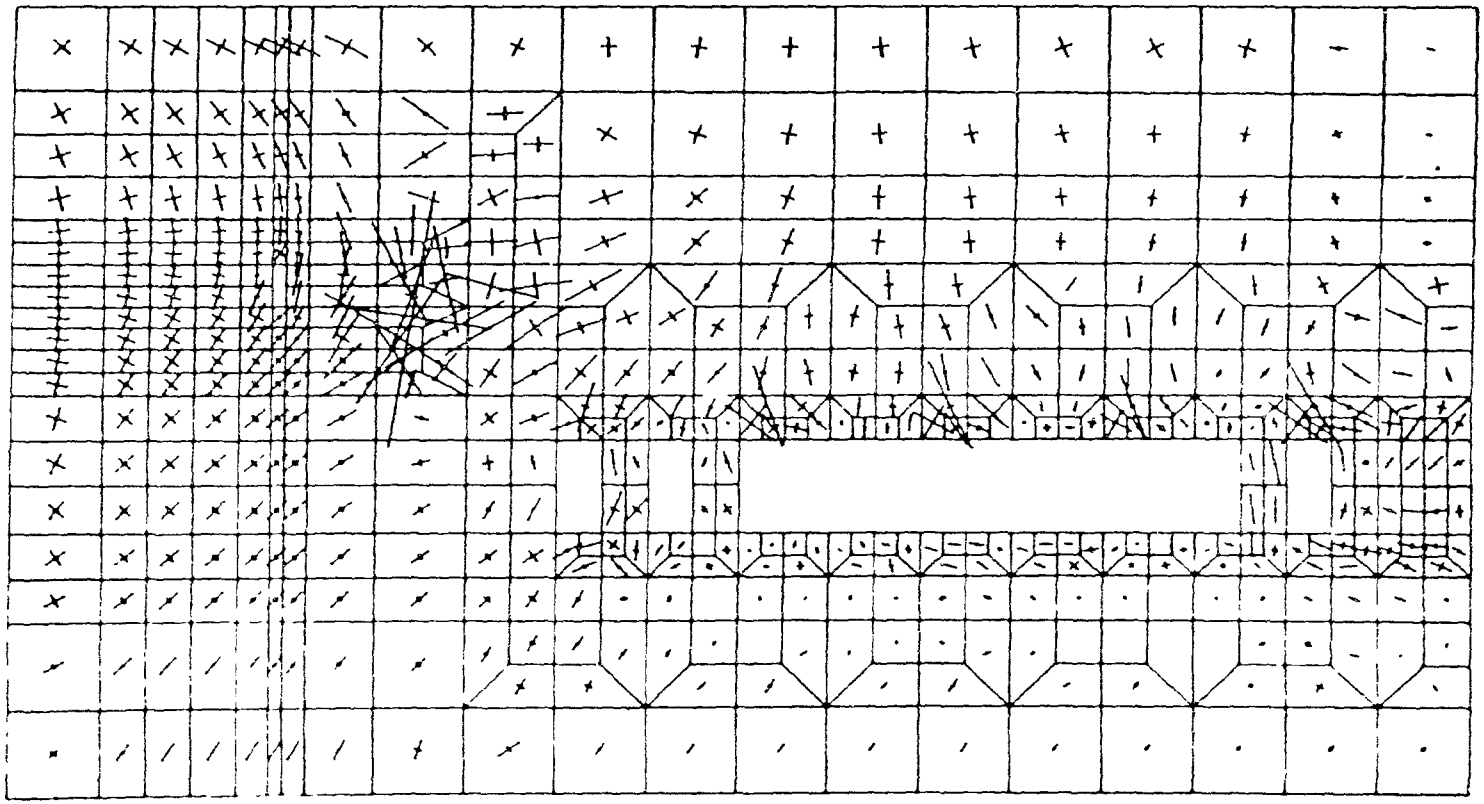


Figure 7.26: Norita Transverse Stope Stability Analysis,
Stress Tensor Plot for Stopes 6, 14, 8 and 12 Excavated

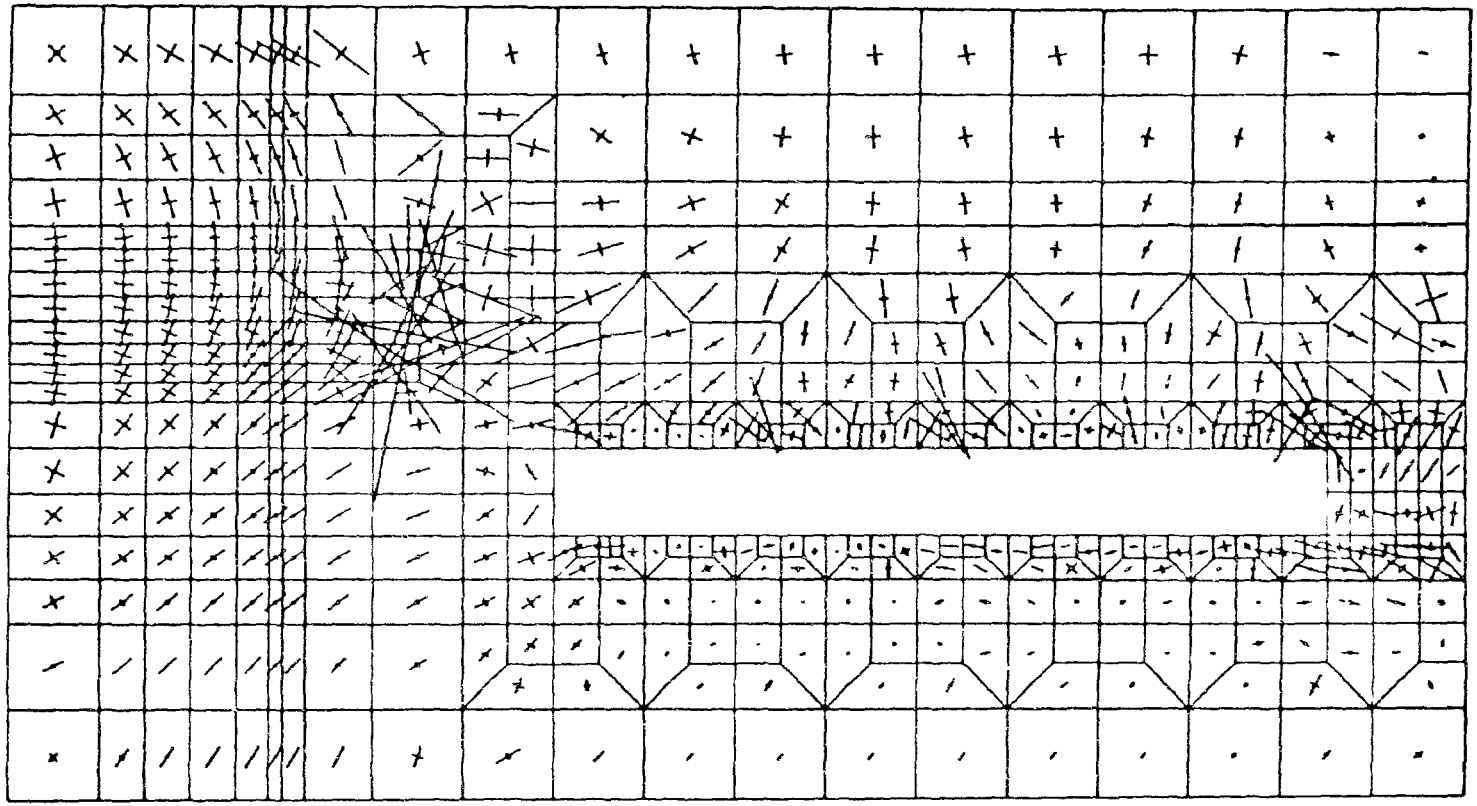


Figure 7.27: Norita Transverse Stope Stability Analysis,
Stress Tensor Plot, all Lower Stopes Excavated

The West and East Abutments

The west abutment lies immediately to the west of the No. 1 Stope and as such, it forms a contiguous part of the abutment between the stopes and the shaft pillar. Any indications of severe stressing of this area might be expected to spread towards the shaft in due course without remedial action. The east abutment on the other hand adjoins the Norita East ore zone which had not been mined at the time. Large stress concentrations in this area would also not be tolerated. The result of the Hybrid stress analysis is shown in Table 7.10. Before startup (Initial Position) the west abutment is still under compressive loads of between 60 and 70 MPa but thereafter, the stresses change to tensile and assume high values, reaching a peak of about 135 MPa at the beginning of June 1986. Although they later decline to about MPa tensile at the final simulation, the tensile stresses are considered high enough to have caused failure if no support of some kind was instituted. The values for the east abutment follow a pattern similar to that of the west abutment. It is worthy of note that the MINTAB simulation put the maximum values for these stresses at 60 - 80 MPa for the west abutment and at 80 - 100 MPa for the east abutment.

No. 2 Pillar

The hybrid stress results for the #2 Pillar are shown in Table 7.11 from where it is clear that tensile stresses predominate soon after the initial position is established. This pillar is best examined in two parts: the western part on the one hand and the central and eastern parts on the other. The western part appears to bear the highest tensile stresses which seem to peak around June 1986 with a maximum value of 149 MPa. With respect to the central and eastern parts of the pillar, the maximum principal stress is small but compressive while the minimum principal stress is tensile, with a maximum value of about 60 MPa. The explanation seems to be that as the transverse stoping operation continues, more and more stopes are being mined to the east of this pillar. With the support previously provided by the mined area gone, this pillar acts like a rod under bending moments with the western part under tension and the eastern part under compression. The MINTAB results from Figure 7.17 shows a stress regime of 60 - 80 MPa tensile in this pillar. Considering the hybrid stress results for the central and eastern parts of the pillar (60 MPa as above), there is a good correlation between the results.

No. 16 Pillar

An examination of the stress results for the No. 16 pillar as shown in Table 7.12 shows a reverse situation compared to the No. 2 Pillar. Thus, the eastern and central parts of the pillar show stress magnitudes similar to those of the eastern and central parts of the No. 2 pillar, while the eastern part of the pillar shows responses similar to the western part of the No. 2 pillar. However, the stress magnitudes are higher in the No. 16 pillar than in the No. 2 pillar, a maximum of 176 MPa tensile being recorded even before the final excavations

were taken. The same explanation for the No. 2 pillar seems to be true for the No. 16 pillar also. Again, the MINTAB results from Figure 7.17 show maximum stresses between 80 and 120 MPa for this pillar, values which are higher than those for the No. 2 pillar by between 20 and 40 MPa. The correlation between the hybrid stress results and the MINTAB results is also considered to be good for this pillar. This is the end of the case study on the Norita Transverse Stopes Stability Analysis. An analysis for the upper 9/10 level stope was considered but was not done because the stoping sequence was very similar to that of the lower 10/11 Level stopes. The results of such an analysis would therefore have constituted a duplicate of the presentation above for the lower 10/11 level stopes.

Table 7.1: Summary of SAP2D F.E. Analysis on Geco Conveyor Drift
Stress History of the Stope from Figs. 7.1 - 7.6
Analysis by the Noranda Technology Centre

	Initial Position (MPa)	Final Position (MPa)
Walls	$-0.50 \leq \sigma_1 \leq -4.76$ $-2.40 \leq \sigma_2 \leq -4.76$	$0.00 \leq \sigma_1 \leq -7.58$ $1.38 \leq \sigma_2 \leq -2.41$
Floor	$-13.79 \leq \sigma_1 \leq -33.79$ $-4.80 \leq \sigma_2 \leq -9.65$	$-13.79 \leq \sigma_1 \leq -59.30$ $-4.76 \leq \sigma_2 \leq -16.55$
Back	$-9.65 \leq \sigma_1 \leq -30.34$ $-2.40 \leq \sigma_2 \leq -11.03$	Not Applicable

Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.2: Summary of SAP2D F.E. Analysis on Geco Conveyor Drift
Stress History of the Drift from Figs. 7.1 - 7.6
Analysis by the Noranda Technology Centre

	Initial Position (MPa)	Final Position (MPa)
Walls	$\sigma_1 \leq -6.90$ $\sigma_2 \simeq -2.40$	$\sigma_1 \leq -2.76$ $\sigma_2 \leq -0.67$
Floor	$\sigma_1 \leq -10.34$ $\sigma_2 \simeq -2.40$	$\sigma_1 \simeq -2.76$ $\sigma_2 \simeq -0.67$
Back	$\sigma_1 \simeq -6.21$ $\sigma_2 \simeq -2.07$	$\sigma_1 \simeq -2.07$ $\sigma_2 \simeq 0.67$

Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.3: Summary of Hybrid Stress F.E. Analysis on Geco Conveyor Drift Stress History of the West Wall of the Stope

	Initial Excavation		First Excavation		Final Excavation	
Node	σ_2	σ_1	σ_2	σ_1	σ_2	σ_1
83	-5.35	-22.49	-5.64	-23.00	-7.11	-39.63
84	2.26	1.64	2.40	1.63	3.51	2.51
85	6.63	-3.23	5.29	-2.12	6.34	-3.80
86	1.76	1.27	3.18	0.23	2.10	0.50
87	-0.84	-20.87	5.44	-0.03	2.36	-0.60
88			0.89	-1.85	1.57	0.03
89			-3.27	-22.17	1.82	0.06
90					1.77	-0.01
91					1.80	0.02
92					1.70	-0.00
93					1.57	-0.00
94					1.50	0.00
95					1.46	-0.01

Note: Negative Values indicate compressive stresses, positive values, tensile

Table 7.4: Summary of Hybrid Stress F.E. Analysis on Geco Conveyor Drift Stress History of the East Wall of the Stope

	Initial Excavation		First Excavation		Final Excavation	
Node	σ_2	σ_1	σ_2	σ_1	σ_2	σ_1
103	-1.15	-19.72	-1.58	-20.73	-11.63	-42.12
104	4.49	1.86	4.64	1.78	3.69	0.70
105	5.36	-3.60	4.62	-2.46	5.40	-5.01
106	1.85	0.81	1.91	0.40	2.24	0.58
107	-8.27	-26.38	6.01	-1.96	3.72	-1.02
108			2.10	-8.10	2.90	0.02
109			11.52	-32.01	3.30	0.04
110					3.19	-0.01
111					3.21	0.02
112					3.20	-0.00
113					3.19	-0.00
114					3.20	0.00
115					3.21	-0.00

Note: Negative Values indicate compressive stresses, positive values, tensile

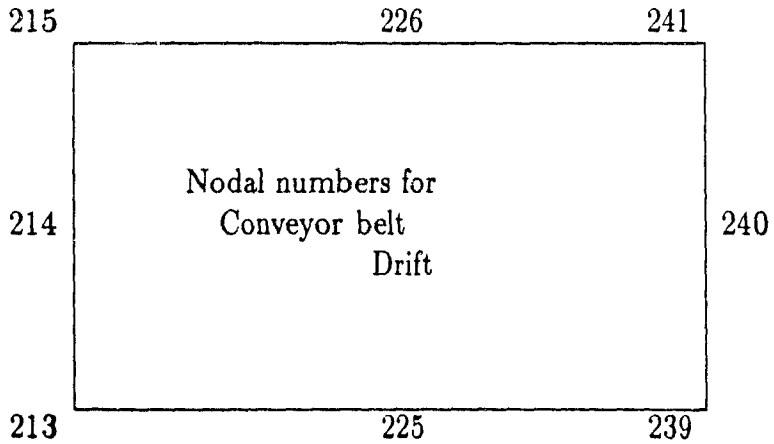
Table 7.5: Summary of SAP2D F.E. Analysis on Geco Conveyor Drift
Stress History of Back and Floor Walls of the Stope

	Initial Excavation		First Excavation		Last Excavation	
Node	σ_2	σ_1	σ_2	σ_1	σ_2	σ_1
			Floor			
Left corner	-5.35	-22.49	-5.64	-23.47	-7.11	-39.63
Mid corner	-2.20	-9.41	-2.38	-9.65	-6.39	-14.12
Right corner	-1.15	-19.72	-1.58	-20.73	-11.63	-42.12
			Back			
Left corner	-0.84	-20.87	-3.27	-22.17	-5.57	-25.74
Mid corner	1.98	-45.73	2.82	-40.73	1.57	-41.44
Right corner	-8.27	-26.38	-11.52	-32.01	-9.56	-28.24

Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.6: Summary of HYBRID F.E. Analysis on Geco Conveyor Drift
Stress History of the Conveyor Drift

	Initial Excavation		First Excavation		Last Excavation	
Node	σ_2	σ_1	σ_2	σ_1	σ_2	σ_1
213	-2.12	-11.18	-3.18	-11.72	0.52	0.25
214	0.79	-1.37	0.74	-1.82	0.02	-0.19
215	1.99	-4.47	2.50	-3.31	-0.29	-0.56
225	0.32	-9.66	0.38	-8.61	0.07	-0.20
226	0.18	-8.47	0.17	-8.05	0.02	-0.20
239	2.73	-2.46	3.31	-0.90	-0.61	-1.28
240	0.86	-1.21	0.81	-1.43	0.03	-0.80
241	-2.27	-9.20	-3.01	-9.65	0.23	-0.07



Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.7: Mechanical Properties of the Main Rock Types

Rock Type	Laboratory Measured values		In-situ RMR Classification Values		Actual Values Used	
	E (GPa)	Poisson's ratio, ν	E (GPa)	Poisson's ratio, ν	E (GPa)	Poisson's ratio, ν
Massive Sulphides	232.0	0.164	28.0	-	100.0	0.25
Basaltic Tuffs	95.0	0.261	10.0	-	50.0	0.25
Rhyolitic Tuffs	68.0	0.15	10.0	-	50.0	0.25

Table 7.8: Mechanical Properties of the Main Rock Types

Rock Type	Compressive Strength (MPa)	Tensile Strength (MPa)	Rupture Strength (MPa)
Massive Sulphides	316.0	30.39	32.90
Basaltic Tuffs	118.00	11.80	37.01
Rhyolitic Tuffs	98.30	15.24	18.14

Table 7.9: Summary of Hybrid Stress Analysis on Norita Transverse Stopes

Stress History of the Shaft Pillar

	Initial Position (MPa)	Position as at June 1986 (MPa)	Final Position (MPa)
West Wall	$-60.00 \leq \sigma_1 \leq -15.00$ $4.00 \leq \sigma_2 \leq 6.00$	$-42.00 \leq \sigma_1 \leq -38.00$ $6.00 \leq \sigma_2 \leq 8.00$	$-48.00 \leq \sigma_1 \leq -3.00$ $6.00 \leq \sigma_2 \leq 7.00$
East Wall	$-69.00 \leq \sigma_1 \leq -19.00$ $5.00 \leq \sigma_2 \leq 8.00$	$-130.00 \leq \sigma_1 \leq -9.00$ $-28.00 \leq \sigma_2 \leq 21.00$	$-57.00 \leq \sigma_1 \leq -10.00$ $6.00 \leq \sigma_2 \leq 9.00$
North Wall	$-104.00 \leq \sigma_1 \leq -93.00$ $-34.00 \leq \sigma_2 \leq -24.00$	$-137.00 \leq \sigma_1 \leq -107.00$ $-30.00 \leq \sigma_2 \leq 11.00$	$-118.00 \leq \sigma_1 \leq -99.00$ $-34.00 \leq \sigma_2 \leq 11.00$
South Wall	$-104.00 \leq \sigma_1 \leq -97.00$ $-37.00 \leq \sigma_2 \leq -15.00$	$-141.00 \leq \sigma_1 \leq -102.00$ $-30.00 \leq \sigma_2 \leq 11.00$	$-118.00 \leq \sigma_1 \leq -99.00$ $-34.00 \leq \sigma_2 \leq 11.00$

Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.10: Summary of Hybrid Stress Analysis on Norita Transverse Stopes

Stress History of the West and East Abutments

	Initial Position (MPa)	Position as at June 1986 (MPa)	Final Position (MPa)
West Abutment	$-67.00 \leq \sigma_1 \leq -60.00$ $-13.00 \leq \sigma_2 \leq -2.00$	$-31.00 \leq \sigma_1 \leq 7.00$ $14.00 \leq \sigma_2 \leq 135.00$	$-29.00 \leq \sigma_1 \leq -9.00$ $13.00 \leq \sigma_2 \leq 92.00$
East Abutment	$-58.00 \leq \sigma_1 \leq -38.00$ $51.00 \leq \sigma_2 \leq 77.00$	$-22.00 \leq \sigma_1 \leq 9.00$ $88.00 \leq \sigma_2 \leq 143.00$	$-27.00 \leq \sigma_1 \leq 14.00$ $123.00 \leq \sigma_2 \leq 161.00$

Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.11: Summary of Hybrid Stress Analysis on Norita Transverse Stopes
Stress History of the #2 Pillar

	Initial Position (MPa)	Position as at June 1986 (MPa)	Final Position (MPa)
West Wall	$-49.00 \leq \sigma_1 \leq -29.00$ $5.00 \leq \sigma_2 \leq 13.00$	$-18.00 \leq \sigma_1 \leq 24.00$ $49.00 \leq \sigma_2 \leq 149.00$	$-15.00 \leq \sigma_1 \leq 23.00$ $15.00 \leq \sigma_2 \leq 107.00$
Central core	$-43.00 \leq \sigma_1 \leq -22.00$ $0.00 \leq \sigma_2 \leq 43.00$	$-27.00 \leq \sigma_1 \leq -3.00$ $37.00 \leq \sigma_2 \leq 59.00$	$-28.00 \leq \sigma_1 \leq 0.00$ $30.00 \leq \sigma_2 \leq 41.00$
East Wall	$-38.00 \leq \sigma_1 \leq -24.00$ $-16.00 \leq \sigma_2 \leq 43.00$	$-22.00 \leq \sigma_1 \leq 4.00$ $2.00 \leq \sigma_2 \leq 58.00$	$-12.00 \leq \sigma_1 \leq 2.00$ $-2.00 \leq \sigma_2 \leq 51.00$

Note: Negative Values indicate compressive stresses,
positive values, tensile

Table 7.12: Summary of Hybrid Stress Analysis on Norita Transverse Stopes
Stress History of the #16 Pillar

	Initial Position (MPa)	Position as at June 1986 (MPa)	Final Position (MPa)
West Wall	$-30.00 \leq \sigma_1 \leq -25.00$ $71.00 \leq \sigma_2 \leq 91.00$	$-60.00 \leq \sigma_1 \leq -6.00$ $2.00 \leq \sigma_2 \leq 115.00$	$-20.00 \leq \sigma_1 \leq -3.00$ $3.00 \leq \sigma_2 \leq 116.00$
Central core	$-49.00 \leq \sigma_1 \leq -31.00$ $67.00 \leq \sigma_2 \leq 109.00$	$-17.00 \leq \sigma_1 \leq 22.00$ $35.00 \leq \sigma_2 \leq 198.00$	$0.00 \leq \sigma_1 \leq 27.00$ $62.00 \leq \sigma_2 \leq 91.00$
East Wall	$-52.00 \leq \sigma_1 \leq -37.00$ $85.00 \leq \sigma_2 \leq 129.00$	$-5.00 \leq \sigma_1 \leq 36.00$ $61.00 \leq \sigma_2 \leq 93.00$	$-11.00 \leq \sigma_1 \leq 17.00$ $50.00 \leq \sigma_2 \leq 176.00$

Note: Negative Values indicate compressive stresses,
positive values, tensile

Chapter 8

Discussions and Conclusions

8.1 Discussions

A hybrid stress finite element procedure for investigating stability problems in the field of geomechanics has been described and implemented. The theoretical basis, formulation of the computer code and the various tests to verify it have also been described in detail. Although emphasis has been laid on its use in mining and related fields, it was pointed out that its basis is essentially a continuum whose properties and response to externally imposed forces can be mathematically represented by a set of equations derived from the theory of elasticity. The essential difference between this (hybrid stress) formulation and the traditional displacement formulation was pointed out as leading to the advantage of using fewer elements for a more accurate solution.

The final equations to be solved, as in most cases of finite element analysis, are for the nodal displacements. Two solution methods were investigated and implemented. One of them, the so-called Skyline Storage method, required the assembly of the complete stiffness matirx and therefore, a larger storage space in the computer. The resulting overheads in data management causes longer delays in problem solution and is therefore ideally suited for small problems. The other method, termed the "frontal solution scheme", was also implemented. This method does not require the complete assembly of the stiffness matrix and therefore does not use up as much storage as the Skyline method. It is therefore ideally suited for use with large problems and may be easier to implement in a small desktop computer where data handling efficiency is of crucial importance.

Several closed form solutions to problems in elasticity were solved accurately, using the new code, thus proving and establishing the veracity of the method. The case histories dealt with in Chapter 6 are recent and current experiences of Noranda Minerals operations in Quebec and Ontario. The results of the hybrid modelling in both cases confirm the numerical modelling results done on the same problems by other methods although the

magnitudes of the stresses by the hybrid method were generally higher than those by other numerical procedures.

8.2 Notable Contributions

The finite element code was formulated in terms of stress and displacements as the primary unknowns. This is particulary relevant in the field of geomechanics where stress is the driving mechanism.

A particular feature of the formulation has been the derivation of the element stress functions (the Airy's Stress Function) in terms of element local coordinates. This scheme involves complex mathematical formulation but has the advantage that the basis of the stiffness matrix computation for each element is the same. It also leads to less computational errors than if the stiffness matrix for each element were computed on the basis of a set of global coordinate axes, especially for a quadrilateral element

Most finite element procedures using quadrilateral elements up till now have tended to concentrate on rectangular elements. The derivation of the relevant equations for a generalized quadrilateral has been one of the strong points of this research for, it allows curved and irregular edges of the continuum to be accurately modelled.

8.3 Suggestions for Future Work

Potential fields of applications of the hybrid stress finite element method are: in fracture mechanics, a field which is becoming very important as the role of fractures in the concentration and transmission of stress in rock masses - and consequently in mine design - become more apparent; in fluid flow, interface modelling, rock bolt modelling, and non-linear material response. Most of these fields of applications were not addressed in this thesis and could easily form the basis of further research in the application of the stress hybrid finite element method to geomechanics problems. A non-linear hybrid stress finite element version would form the basis of more realistic geomechanics application especially in the field of mining.

The computer code was written in FORTRAN 77 and implemented on a micro VAX II computer with a large memory. Although micro VAX II fortran programs are generally compatible with the more easily accessible IBM desktop computers, implementation of this code on a smaller computer may require changes to the size of the various arrays used in the program. The implementation of this code in another computer language such as "C" would definitely save on computer memory and would speed up programme execution. None of these items was attempted in the current research.

The present code is for 8-node quadrilateral elements. A 4-node quadrilateral element version will be less expensive to execute and should also produce accurate and acceptable results.

A simple graphics programme could be written to display the meshes and plot the stresses and displacements, rather than the present difficult method of exporting the output files to an external graphics package.

8.4 Conclusions

The objectives of the research reported in this thesis were met. A hybrid stress finite element computer code was written, along with mesh generation and alteration codes. The finite element code was successfully verified and two mining casse histories were examined.

Bibliography

[1] Jumukis,A.R. *Rock Mechanics.* Trans Tech Publications, 1983.

[2] Goodman, E.R. *Introduction to Rock Mechanics.* John Wiley and Sons, New York, 1980.

[3] Brady, B.H.G. and Brown, E.T. *Rock Mechanics for Underground Mining.* George Allen and Unwin, 1985.

[4] Bieniawski, Z.T. *Rock Mechanics design in Mining and tuneling.* A.A. Balkema, Amsterdam, 1984.

[5] Judd, W.R. *Rock Stress, Rock Mechanics and Research, State of Stress in the Earth's Crust.* Elsevier, New York, 1964.

[6] Hoek, E. and Brown, E.T. *Underground Excavations in Rock.* Institution of Mining and Metallurgy, London, England, 1980.

[7] Jaeger, J.C. and Cook, N.G.W. *Fundamentals of Rock Mechanics.* Chapman and Hall Ltd. and Science Paperbacks, New York, 1971.

[8] Muskhelishvili, N.I. *Some Basic Problems of the Mathematical Theory of Elasticity.* translated by J.R.M. Radok, Groningen, Noordhoff, fourth edition, 1953.

[9] Goodier, J.N. Concentration of stress around spherical inclusions and flaws. *Trans. Am. Soc. Mech. Engrs.*, 55:39–44, 1953.

[10] Goodman, R.E. On the distribution of stresses around tunnels in non-homogeneous rocks. *Proc. First Congress ISRM*, 249–255, 1966.

[11] Heller, S.R., Brock, J.S., and Bart, R. *The Stresses Around a Rectangular Opening with Rounded Corners in a Uniformly Loaded Plate.* Technical Report Report 1149, David Taylor Model Basin, 1958.

[12] Greenspan, M. Effect of a small hole on the stresses in a uniformly loaded plate. *Quarterly App. Math.*, 2:60–71, 1944.

[13] Timoshenko, S. and Goodier, J M *Theory of Elasticity.* McGraw-Hill Book Co , New York, 1951.

[14] Terzaghi, K. and Richart Jr., F E. Stresses in rocks around cavities *Geotechnique - The Int. J. Soil Mech.*, III.57-90, 1952.

[15] Desai, C S. and Christian, C. *Numerical Methods in Geotechnical Engineering* McGraw-Hill Book Co., New York, 1977

[16] Beer, G. and Meek, J.L. Infinite domain elements *Int. J. Numer Methods Eng.*, 17:43-52, 1981.

[17] Turner, M.J., Clough, R.W., Martin, H C., and Topp, L.J. Stiffness and deflection analysis of complex structures. *J. Aeronaut Sci.*, 23:805-823, 1956.

[18] Argyris, J.H. Energy theorems and structural analysis. *Aircraft Engineering*, 27:125-154, 1955.

[19] Pian, T.H.H. Derivation of element stiffness matrices by assumed stress distributions. *AIAA Journal*, 2(7), 1964.

[20] Pian, T.H.H. and Tong. P. Basis of finite element methods for solid continua *Int J. Numer Methods Eng.*, 1:3 28, 1969.

[21] Washizu, K. *Variational Methods in Elasticity and Plasticity* Pergamon Press, Oxford , 1968.

[22] Pian, T H.H. Formulations of finite element methods for solid continua. In Oden *et al.*, editor, *Recent Advances in Matrix Methods of Structural Analysis and Design*, pages 49–83, Univ. of Alabama Press, Tuscaloosa, Ala , 1969

[23] Pian, T.H.H. *Hybrid Models, Numerical and Computer Methods in Structural Mechanics.* Fenvers et al , 1973.

[24] Pian, T.H.H. Element stiffness matrices for boundary compatibility and for prescribed boundary stresses. *Proc. Conf on Matrix Methods in Structural Mech* , 1(AFFDL-TR-66-80.), 1966.

[25] Yamada, Y., Yoshimura, N , and Sakurai, T Plastic stress strain matrix and its application for the solution of elastic-plastic problems by the finite element method *Int J Mech Sci.*, 10:343-354, 1968

[26] Tong, P. and Pian, T H H. A variational principle and the convergence of finite element method based on assumed stress distribution *Int J. of Solids and Structures*, 5:463-472, 1969

[27] Akin, J.E. The generation of elements with singularities. *Int. J. Numer. Methods Eng.*, 10:1249–1259, 1976.

[28] Benzley, S.E. Representation of singularities with isoparametric finite elements. *Int. J. Numer. Methods Eng.*, 8:537-545, 1974.

[29] Roshdy S.B. On the use of isoparametric finite elements in linear fracture mechanics. *Int. J. Numer. Methods Eng.*, 10:25–37, 1976.

[30] Atluri, S.N., Kobayashi, A.S., and Nakagaki, M. An assumed displacement hybrid finite element model for linear fracture mechanics *Int. J. Fracture*, 11(2), 1975.

[31] Lightner, J.G. *A Mixed Finite Element Procedure for Soil-Structure Interaction including Simulation of Excavation Sequences.* PhD thesis, Dept. Civil Eng., Virginia Tech., Blacksburg, Va., U.S.A., 1981.

[32] Goodman, R.E. and Taylor, R.L. and Brekke, T.L. A model for the mechanics of jointed rocks. *J. Soil Mech. and Found. Div.*, ASCE, Vol. 94(SM3):637-659, 1968.

[33] Zienkiewicz, O.C., Valliapan, S., and King, I P. Elastoplastic solutions of engineering problems 'initial stress', finite element approach. *Int. J. Numer. Methods Eng.*, 1:75–100, 1969.

[34] Ghaboussi, J., Wilson, E.L., and Isenberg, J. Finite elements for rock joints and interfaces. *J. Soil Mech. and Found. Eng.*, ASCE Vol. 99(SM10), 1973.

[35] Wissman, J.W. and Sprecht, B. Inclusion of hybrid deformation elements in the class of simple deformation models. *Int. J. Numer. Methods Eng.*, 15:855–866, 1980.

[36] Cook, R.D. and Ladkany, S.G. Observations regarding assumed stress hybrid plate elements. *Int. J. Numer. Methods in Eng.*, 8:513–519, 1974.

[37] Wolf, J.P. Alternate hybrid stress finite element models. *Int. J. Numer. Methods Eng.*, 9:601–615, 1975.

[38] Spilker, R.L. and Munir, N.I. The hybrid stress model for thin plates. *Int. J. Numer. Methods Eng.*, 15:1239–1260, 1980.

[39] Ahmad, S. and Irons, B.M. An assumed stress approach to refined isoparametric finite element in three dimensions. *Proc. 1974 Int. Confr. Finite El. Methods in Engineering*, 1974.

[40] Pian,T.H.H., Tong, P., Luk, C.H., and Spilker, R.L. Elastic-plastic analysis by assumed stress hybrid model. *Proc. 1974 Int. Confr. Finite El Methods in Engineering*, 1974.

[41] Barnard, A.J. and Sharman, P.W. Elasto-plastic analysis using hybrid-stress finite elements. *Int. Confr. on Finite Elements in Nonlinear Solid and Structural Mechanics*, 1, 1977.

[42] Tong, P., Pian, T.H.H., and Lasry, S.J. A hybrid element approach to crack problems in plane elasticity. *Int. J. Numer Methods Eng.*, 7:297-308, 1973

[43] Tong, P. An assumed stress hybrid finite element method for an incompressible and near-incompressible material. *Int J. Solids Structures*, 5 455 461, 1969

[44] Spilker, R.L. and Munir, N A serendipity cubic diplacement hybrid-stress element for thin and moderately thick plates *Int. J Numer Methods Eng*, 15 1261 1278, 1980

[45] Goodman, R.E. and Brown, C B. Dead load stresses and the instability of slopes. *J Soil Mech. and Found Div., ASCE*, 89(SM 3), 1963

[46] Clough, G.W and Duncan, J.W *Finite Element Analysis of Port Allen and Old River Locks.* Technical Report TE 69-3, U S Army Engineers Waterways Experimental Station, 1969.

[47] Mana, A I. *Finite Element Analysis of Deep Excavation Behaviour In Soft Clay.* PhD thesis, Stanford University, Stanford, 1978.

[48] Christian, J.T. and Wong, I.H. Errors in simulation excavations in elastic media by finite elements. *Japanese Society of Soil Mechanics and Foundation Engineering*, 13(1), 1973.

[49] Zienkiewick, O C. *The Finite Element Method in Engineering Science* McGraw Hill, New York, 1971.

[50] Aziz, A.K., editor. *Symposium on Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, Univ of Maryland, 1972

[51] Carnahan, B., Luther, H.A , and Wilkes, J O *Applied Numerical Methods* John Wiley, New York, 1969.

[52] Bathe, K-J. and Wilson, E L. *Numerical Methods in Finite Element Analysis* Prentice-Hall Inc., New Jersey, 1976.

[53] Cook, R.D. *Concepts and applications in Finite Element Analysis* John Wiley and Sons, Inc , New York, second edition, 1981

[54] Akin, J.E *Application and Implementation of Finite Element Methods* Academic Press, Toronto., 1982

[55] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vettering, W.T. *Numerical Recipes - the Art of Scientific Computing.* Cambridge Univ. Press, London, first edition, 1986.

[56] Korn, G.A. and Korn, T.M. *Mathematical Handbook for Scientists and Engineers.* McGraw-Hill Book Co., New York, 1961.

[57] Abramowitz, M. and Stegun, I.A *Handbook of Mathematical Functions.* Dover Pub. Inc., New York., ninth edition, 1970.

[58] Fellipa, C.A. Solution of linear equations with skyline-stored symmetric matrix. *Computers and Structures*, 5:13-29, 1975.

[59] Mondkar, D.P. and Powell, G.H. Towards optimal in-core equation solving. *Computers and Structures*, 531-548, 1974.

[60] Meyer, C. Solution of linear equations, state-of-the-art. *J. Struct. Div., Proc.*, ASCE Vol. 99(ST7), 1973.

[61] George, A. A survey of sparse matrix methods in the direct solution of linear equations. *Proc., Summer Computer Simulation Confr.*, 15-20, 1973.

[62] Hood, P. Frontal solution program for unsymmetric matrices. *Int. J. Numer. Methods Eng.*, 10:379-399, 1976.

[63] Hinton, E. and Owen, D.R. *Finite Element Programming.* Academic Press, London, 1977.

[64] Irons, B.M. A frontal solution for finite element analysis. *Int. J. Numer. Methods Eng.*, 2:5-32, 1970.

[65] Valliapan, S. *Continuum Mechanics Fundamentals.* A. A. Balkema Publications, Rotterdam, 1981.

[66] Popov, E. P. *Mechanics of Materials.* Prentice Hall Inc., second edition, New Jersey, New York, 1978.

[67] Kreyszig, E. *Advanced Engineering Mathematics.* John Wiley and Sons, Inc., third edition, New York, 1972.

[68] Obert, L. and Duvall, R. *Rock Mechanics and the Design of Structures in Rock.* John Wiley and Sons, Inc , New York, 1976.

[69] Desai, C.S. and Abel, J.F. *Introduction to the Finite Element Method.* Van Nostrand Reinold Co., Amsterdam, 1972 Ω

[70] Bawden, W.F. and Milne, D. *Stability Analysis of Geco Conveyor Drift.* Internal Research Report RR 86-07: N-8404, Noranda Technology Centre, Montreal, 1986.

[71] Milne, D. *Rock Mechanics Review at GECO.* Internal Research Report RR 87-03: N-8404, Noranda Technology Centre, Montreal, 1987.

[72] Chauvin, J-P. *Ground Control, Monitoring and Simulation to Predict Ground Movement at Norita Mine, Matagami, Quebec.* Technical Report, Noranda Mine Engineering Fall Seminar, Manitouwadge, Ontario, September 1986.

[73] Goodier, A. and Dube, R. *Changes in Mining Methods to Overcome Ground Conditions at the Norita Mine.* Technical Report, 86th. AGM of the CIM, Ottawa, 1984

[74] Bawden, W.F. and Milne, D. *Rock Mechanics Design for Norita Mine* Internal Research Report RR 86-08: N-8404, Noranda Technology Centre Montreal, 1987.

[75] Stoesser, A.J. *Analysis of Norita Mine and Discussion of MINE12.X* Dept. of Civil Eng., Univ. of Toronto, Contract Report for Noranda Technology Centre, Montreal, 1987

[76] Appa Rao, T.V S.R., author. *An Assumed Stress Finite Element Model for the Analysis of an Axisymmetric Thick-Walled Pressure Vessel*, 1st. Int. Confr. on Struct. Mechs. in Reactor Technology, Thomas A. Jaeger, editor, Commission of the European Communities, Berlin, Germany, 6:315–332, 1971

Appendix A

LISTING OF THE HYBRID STRESS COMPUTER CODE, HYBRID.FOR

The listing of the hybrid stress computer program, HYBRID.FOR, is given in this Appendix. The program has been described in Chapter 5.

```
C*****************************************************************
CC
CC                A GENERAL PURPOSE LINEAR ELASTIC
CC              HYBRID STRESS FINITE ELEMENT PROGRAM
CC
CC         THE FOLLOWING LIMITS APPLY:
CC
CC   MAXIMUM NO. OF NODES                                      = 2000
CC   MAXIMUM NO. OF ELEMENTS                                   = 1000
CC   MAXIMUM NO. OF NODES WITH DISPL. CONSTRAINTS              = 200
CC   MAXIMUM NO. OF ELEMENT SIDES WITH DISTRIBUTED LOADS       = 100
CC   MAXIMUM NO. OF MATERIAL TYPES                             = 10
CC
CC    IF ANY OF ABOVE LIMITS IS EXCEEDED, PROGRAM MUST BE RECOMPILED
CC
CC    IN THIS EDITION, CONSTRAINED ROWS AND COLUMNS OF THE STIFFNESS
CC    MATRIX ARE NOT DELETED. SKYFAC/SKYSOL INVOCATION IS ON THE
CC    UNREDUCED  STIFFNESS MATRIX, LD VECTOR, AND QSLOD VECTOR.
CC
CC*****************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /BLK1/ PP(3,18),DB(3,8),DA(2,8),PB(3,3),SH(8),BD(3),DT
        COMMON /BLK2/ XX(2,8), SS(3,3),GPT(4),GWT(4),KXY(8)
        COMMON /OAKS/ HMX(18,18),GG(18,16),XLL(2,16),HB(18)
        COMMON /CONS/ THIC,NUMAT,NTYPE,NPOIN,NDIST,NNEL,
     +                NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT
        COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
        COMMON /BLOK/ SE(9),TE(9)
        COMMON        ASKY(500000),NUMEL(8000),QSLOD(4000),
     +                DISPL(4000),LDVEC(4000),
     +                XORD(2000),YORD(2000),
     +                NELTYP(800),DENS(30),CAR(60)
        LOGICAL       EXISTS
        CHARACTER*1   TITLE(80)
```

```
CC
CC  INITIALIZE VARIABLES
CC
            LIN1 = 1
            LIN2 = 2
            LIN3 = 3
            LIN4 = 4
            LIN7 = 7
            LIN8 = 8
            LIN9 = 9
CC
CC  CHECK THAT DATA FILE DECLARED NEW ARE NOT EXISTING.
CC  IF THEY EXIST, PURGE THEM FIRST.
CC
CC ---------------------------------------------------
            INQUIRE(FILE='COUTFILE',EXIST=EXISTS)
            IF(EXISTS) THEN
            OPEN(UNIT=LIN1,FILE='COUTFILE',STATUS='OLD')
            CLOSE(UNIT=LIN1,STATUS='DELETE')
            ENDIF
            OPEN(UNIT=LIN1,FILE='COUTFILE',STATUS='NEW',
     1      FORM='FORMATTED')
CC ---------------------------------------------------
CC
            OPEN(UNIT=LIN2,FILE='FILE2',STATUS='NEW',
     1        ACCESS='SEQUENTIAL')
            OPEN(UNIT=LIN3,FILE='FILE3',STATUS='NEW',
     1        ACCESS='SEQUENTIAL')
C           OPEN(UNIT=LIN4,FILE='FILE4',STATUS='NEW',
C    1        ACCESS='SEQUENTIAL')
            OPEN(UNIT=LIN7,FILE='FILE7',STATUS='NEW',
     1        ACCESS='SEQUENTIAL')
            OPEN(UNIT=LIN9,FILE='FILE9',STATUS='NEW',
     1        ACCESS='SEQUENTIAL')
            NQP=4
            CALL GAUSCO(NQP,GPT,GWT)
C
        LK = 1
C       DO 500 LK = 1,8
CC
CC  CALL THE MESH GENERATING SUBROUTINE
```

```
CC
      CALL INPUT(TITLE,XORD,YORD,QSLOD,NUMEL,LDVEC,NELTYP,DENS,CAR)
CC
      NAMX = NNEL*306
      DO 10 J=1,NWA
      ASKY(J)=0.
   10 CONTINUE
      DO 20 J=1,NEQ
      DISPL(J)=0.
   20 CONTINUE
CC
CC  COMPUTE ELEMENT STIFFNESS MATRIX. ASSEMBLE INTO GLOBAL ARRAY
CC
      NUM3 = NUMAT*3
      REWIND LIN2
      READ (LIN2,*) (XORD(I),YORD(I),I=1,NNODE)
      READ (LIN2,*) (NELTYP(I),I=1,NNEL)
      READ (LIN2,*) (NUMEL(I),I=1,NTOT)
      READ (LIN2,*) (DENS(I),I=1,NUM3)
      DO 40 NL=1,NNEL
      REWIND LIN9
      READ (LIN9,*) (QSLOD(I),I=1,NEQ)
      NN = NELTYP(NL)
      RHO = DENS(NN)
      N3 = NN*3
      DO 25 I = 1,3
      DO 25 J = 1,3
      SS(I,J) = 0.0
   25 CONTINUE
      SS(1,1) = CAR(N3-2)
      SS(1,2) = CAR(N3-1)
      SS(3,3) = CAR(N3)
      SS(2,1) = SS(1,2)
      SS(2,2) = SS(1,1)
      N1 = NL*8 - 7
      N2 = N1 + 7
      KK=0
      DO 30 J = N1,N2
      JJ=NUMEL(J)
      KK = KK+1
      XX(1,KK)=XORD(JJ)
```

```
      XX(2,KK)=YORD(JJ)
      KXY(KK)=JJ
 30   CONTINUE
CC
      CALL ELSTF(NL,NEQ,NQP,NWA,ASKY,QSLOD,LDVEC,THIC,RHO)
 40   CONTINUE
CC
CC  SCALE THE STIFFNESS MATRIX
CC
      ASMAX = 0.
      DO 60 I = 1,NWA
      DD = ABS(ASKY(I))
      IF(ASMAX-DD) 50,60,60
 50   ASMAX = DD
      ISP = I
 60   CONTINUE
CC
      DO 70 I = 1,NWA
      ASKY(I) = ASKY(I)/ASMAX
 70   CONTINUE
CC ------------------------------------------------------
      OPEN(UNIT=LIN4,FILE='FILE4',STATUS='NEW',
     1       ACCESS='SEQUENTIAL')
      WRITE(LIN4,*) (ASKY(I),I=1,NWA), (LDVEC(I),I=1,NEQ+1)
CC
      CALL SKYFAC(ASKY,LDVEC,NWA,0,NEQ)
CC
CC  SCALE THE LOAD VECTOR
CC
      DO 130 I = 1,NEQ
      QSLOD(I) = QSLOD(I)/ASMAX
 130  CONTINUE
CC
      CALL SKYSOL(ASKY,LDVEC,NWA,NEQ,0,1,QSLOD,DISPL,NEQ)
CC
 140  CONTINUE
CC
      CALL STRSOL(DISPL,XORD,YORD,NUMEL,DENS,CAR,NELTYP,
     1                  THIC,NUMAT,NTYPE,NPOIN,NDIST,NNEL,
     1                  NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT)
 500  CONTINUE
```

```
          CLOSE(UNIT=LIN2,STATUS='DELETE')
          CLOSE(UNIT=LIN3,STATUS='DELETE')
          CLOSE(UNIT=LIN4,STATUS='DELETE')
          CLOSE(UNIT=LIN7,STATUS='DELETE')
          CLOSE(UNIT=LIN9,STATUS='DELETE')
        STOP
        END
CC ***********************************************************************
      BLOCK DATA
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BLOK/ SE(9),TE(9)
      DATA SE/-1.0, 0.0, 1.0, 1.0, 1.0, 0.0,-1.0, -1.0, 0.0/
      DATA TE/-1.0, -1.0, -1.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0/
      END
CC ***********************************************************************
      SUBROUTINE ELSTF(NL,NEQ,NQP,NWA,ASKY,QSLOD,
     1 LDVEC,THIC,RHO)
      IMPLICIT REAL*8 (A-H,O-Z)
CC
      COMMON /BLK2/ XX(2,8), SS(3,3),GPT(4),GWT(4),KXY(8)
      COMMON /OAKS/ HMX(18,18),GG(18,16),XLL(2,16),HB(18)
      COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
      DIMENSION     QSLOD(1),ASKY(NWA),LDVEC(NEQ+1)
      DIMENSION     AMX(18,16),BS(16,16),HG(18)
      INTEGER       NEQ,NQP,NL,NWA
CC
CC  FORM THE HMX MATRIX
CC
      CALL HMTX(NQP,NL)
CC
CC INVERT THE H MATRIX
CC
      CALL MATINV(HMX,18)
CC
CC  FORM THE GG MATRIX
CC
      CALL GMTX(QSLOD,NEQ,NL,NQP,THIC,RHO)
CC
      DO 60 I=1,18
      DO 60 J=1,16
      DD=0.
```

```
      DO 50 K=1,18
   50 DD=DD+HMX(I,K)*GG(K,J)
      AMX(I,J)=DD
   60 CONTINUE
   65  format(6f12.4)
      WRITE(LIN3,*) AMX
      DO 80 I=1,16
      DO 80 J=1,16
      DD=0.
      DO 70 K=1,18
   70 DD=DD+GG(K,I)*AMX(K,J)
      BS(I,J)=DD*THIC
   80 CONTINUE
CC
CC  FORM (HG) = INV.(HMX)*(HB)
CC
      DO 120 I=1,18
      HG(I)=0.
      DO 110 J=1,18
      HG(I)=HG(I)+HMX(I,J)*HB(J)
  110 CONTINUE
  120 CONTINUE
      WRITE(LIN3,*) HG
CC
CC  STORE TRANS.(GG) X (HG) IN FIRST 16 LOCATIONS OF (HB)
CC
      DO 130 I=1,16
      HB(I)=0.
      DO 130 J=1,18
      HB(I)=HB(I)+GG(J,I)*HG(J)
  130 CONTINUE
CC
CC---  EQUIVALENT NODAL FORCES
CC   SUM APPROPRIATE COMPONENTS OF HB INTO GLOLAL QSLOD VECTOR ONLY
CC
      REWIND LIN9
      READ (LIN9,*) (QSLOD(I),I=1,NEQ)
      DO 140 J=1,16
      J1=(J+1)/2
      K=KXY(J1)*2-MOD(J,2)
      QSLOD(K)=QSLOD(K)+HB(J)
```

```
 140   CONTINUE
       REWIND LIN9
       WRITE (LIN9,*) (QSLOD(I),I=1,NEQ)
CC
CC   BS(16,16) IS ELEMENT STIFFNESS MATRIX.
CC   ELEMENTS OF BS ARE NOW SUMMED INTO ASKY(*)
CC
       IF(NL.EQ.1) GOTO 145
 145   DO 180 II=1,16
       I=(II+1)/2
       L=KXY(I)
       LL=L*2-MOD(II,2)
       DO 180 JJ=II,16
       J=(JJ+1)/2
       M=KXY(J)
       MM=M*2-MOD(JJ,2)
       IF(LL-MM) 150,150,160
 150   KK=IABS(LDVEC(MM+1))+LL-MM
       GOTO 170
 160   KK=IABS(LDVEC(LL+1))-LL+MM
 170   ASKY(KK)=ASKY(KK)+BS(II,JJ)
 180   CONTINUE
CC
CC
       RETURN
       END
CC ********************************************************************
       SUBROUTINE GAUSCO(N,A,W)
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION A(N),W(N)
CC
CC  -----   N=NO. OF GAUSS POINTS IN ONE DIMENSION
CC
       NLES1=N-1
       GOTO(10,20,30), NLES1
  10   A(1)=-0.577350269189626
       A(2)=-A(1)
       W(1)=1.0
       W(2)=1.0
       GOTO 100
  20   A(1)=-0.774596669241493
```

```
      A(2)=0.
      A(3)=-A(1)
      W(1)=0.555555555555556
      W(2)=0.888888888888889
      W(3)=W(1)
      GOTO 100
   30 A(1)=-0.861136311594053
      A(2)=-0.339981043584856
      A(3)=-A(2)
      A(4)=-A(1)
      W(1)=0.347854845137454
      W(2)=0.652145154863546
      W(3)=W(2)
      W(4)=W(1)
  100 RETURN
      END
CC ***********************************************************************
      SUBROUTINE GMTX(QSLOD,NEQ,NL,NQP,THIC,RHO)
      IMPLICIT REAL*8 (A-H,O-Z)
CC
      COMMON /BLK1/ PP(3,18),DB(3,8),DA(2,8),PB(3,3),SH(8),BD(3),DT
      COMMON /BLK2/ XX(2,8), SS(3,3),GPT(4),GWT(4),KXY(8)
      COMMON /OAKS/ HMX(18,18),GG(18,16),XLL(2,16),HB(18)
      COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
      DIMENSION QSLOD(NEQ),GB(16),PC(2),BB(2,3),AP(3),BP(3),CP(3)
      INTEGER       NEQ,NQP
CC
CC  INITIALIZE
CC
CC
      DO 10 I=1,18
      DO 10 J=1,16
   10 GG(I,J)=0.
CC
CC  NUMERICAL INTEGRATION OVER EACH ELEMENT SIDE
CC
      DO 130 IS=1,4
      DO 15 J=1,16
   15 GB(J)=0.
CC
CC  COMPUTE LIMITS OF XLL MATRIX
```

```
CC
      N1=IS*4-3
      N2=N1+5
CC
CC  COMPUTE COMPONENTS OF SIDE IN X- AND Y-DIRECTIONS
CC
      I1=IS*2-1
      I3=I1+2
      IF(I3.GT.8) I3=I3-8
      CY=XX(2,I3)-XX(2,I1)
      CX=XX(1,I3)-XX(1,I1)
      RAD=SQRT(CX**2+CY**2)
      CS=CY/RAD
      SC=-CX/RAD
      BB(1,1) = CS
      BB(1,2) = 0.0
      BB(1,3) = SC
      BB(2,1) = 0.0
      BB(2,2) = SC
      BB(2,3) = CS
CC
      DO 110 II=1,NQP
       PC(1) = 0.0
       PC(2) = 0.0
      SI=GPT(II)
      TI=SI
      WI=GWT(II)
      GOTO(20,30,40,50), IS
  20  TI=-1.0
      GOTO 60
  30  SI=1.0
      GOTO 60
  40  TI=1.0
      GOTO 60
  50  SI=-1.0
CC
CC   ---   NUMERICAL INTEGRATION OVER SIDE IS
CC
  60  CALL LMTX(SI,TI,XLL,IS)
CC
      CALL PMTX(SI,TI,PP,PB,BD,DELT,XX,RHO,NL)
```

```
CC
CC      COMPUTE G AND GB MATRICES
CC      MULTIPLY TRANS. OF BD BY GB AND STORE RESULT AS GB VECTOR
CC
        DO 70 I=1,18
        DO 65 K = 1,3
        AP(K) = PP(K,I)
        IF(I.LE.3) THEN
        CP(K) = PB(K,I)
        ENDIF
  65    CONTINUE
        DO 67 J = 1,2
        DO 66 K = 1,3
        BP(K) = BB(J,K)
  66    CONTINUE
        DD = DOTPRD(AP,BP,3)
        PP(J,I)=RAD * WI * THIC * DD/2.0
CC
        IF(I.LE.3) THEN
        DD = DOTPRD(CP,BP,3)
        D1 = RAD * WI * DD * THIC/2.0
        PC(J) = PC(J) + D1*BD(I)
        ENDIF
CC
  67    CONTINUE
  70    CONTINUE
CC
       DO 90 I = 1,18
       AP(1) = PP(1,I)
       AP(2) = PP(2,I)
       DO 80 JK = N1,N2
       J=JK
       IF(J.GT.16) J=J-16
       BP(1) = XLL(1,J)
       BP(2) = XLL(2,J)
       GG(I,J) = GG(I,J) + DOTPRD(AP,BP,2)
  80   CONTINUE
  90   CONTINUE
       DO 100 JK = N1,N2
       J=JK
       IF(J.GT.16) J=J-16
```

```
      BP(1) = XLL(1,J)
      BP(2) = XLL(2,J)
      GB(J) = GB(J) + DOTPRD(PC,BP,2)
  100 CONTINUE
  110 CONTINUE
CC
CC--- EQUIVALENT NODAL FORCES
CC    SUM GB VECTOR INTO QSLOD
CC
      DO 120 JJ = 1,16,2
      I1=(JJ+1)/2
      J1=KXY(I1)*2-1
      QSLOD(J1)=QSLOD(J1)-GB(JJ)
      QSLOD(J1+1)=QSLOD(J1+1)-GB(JJ+1)
  120 CONTINUE
  130 CONTINUE
      RETURN
      END
CC **********************************************************************
      SUBROUTINE SHAPEF(S,T,DA,DB,SH)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION       DA(2,8),DB(3,8),SH(8)
CC
      S1=(1.-S)/2.
      S2=(1.+S)/2.
      T1=(1.-T)/2.
      T2=(1.+T)/2.
CC
CC  SH(I)   = SHAPE FUNCTION
CC  DA(1,I) = FIRST DERIVATIVE OF SHAPE FUNCTION WITH RESPECT TO S
CC  DA(2,I) = FIRST DERIVATIVE OF SHAPE FUNCTION WITH RESPECT TO T
CC  DB(1,I) = 2ND DERIVATIVE OF SHAPE FUNCTION WITH RESPECT TO S
CC  DB(2,I) = 2ND DERIVATIVE OF SHAPE FUNCTION WITH RESPECT TO T
CC  DB(3,I) = 2ND DERIVATIVE OF SHAPE FUNCTION WITH RESPECT TO S & T
CC
      DO 20 J=1,8
      SH(J)=0.
      DO 20 I=1,3
      IF (I.LE.2) DA(I,J)=0.
      DB(I,J)=0.
   20 CONTINUE
```

```
CC
      DB(1,1)=T1
      DB(1,2)=-2.*T1
      DB(1,3)=T1
      DB(1,5)=T2
      DB(1,6)=-2.*T2
      DB(1,7)=T2
CC
      DB(2,1)=S1
      DB(2,3)=S2
      DB(2,4)=-2.*S2
      DB(2,5)=S2
      DB(2,7)=S1
      DB(2,8)=-2.*S1
CC
      DB(3,1)=1.25-S2-T2
      DB(3,2)=S
      DB(3,3)=T2-S2-.25
      DB(3,4)=-T
      DB(3,5)=S2+T2-0.75
      DB(3,6)=-S
      DB(3,7)=S2-T2-0.25
      DB(3,8)=T
CC
      DA(1,1)=T1*(S+T/2.)
      DA(1,2)=-S*(1.-T)
      DA(1,3)=T1*(S-T/2.)
      DA(1,4)=T1*(1.+T)
      DA(1,5)=T2*(S+T/2.)
      DA(1,6)=-S*(1.+T)
      DA(1,7)=T2*(S-T/2.)
      DA(1,8)=-T2*(1.-T)
CC
      DA(2,1)=S1*(T+S/2.)
      DA(2,2)=-S1*(1.+S)
      DA(2,3)=S2*(T-S/2.)
      DA(2,4)=-T*(1.+S)
      DA(2,5)=S2*(T+S/2.)
      DA(2,6)=S2*(1.-S)
      DA(2,7)=S1*(T-S/2.)
      DA(2,8)=-T*(1.-S)
```

```
CC
      SH(1)=-(1.-S)*(1.-T)*(S+T+1.)/4.
      SH(2)=(1.-S*S)*(1.-T)/2.
      SH(3)=(1.+S)*(1.-T)*(S-T-1.)/4.
      SH(4)=(1.-T*T)*(1.+S)/2.
      SH(5)=(1.+S)*(1.+T)*(S+T-1.)/4.
      SH(6)=(1.-S*S)*(1.+T)/2.
      SH(7)=-(1.-S)*(1.+T)*(S-T+1.)/4.
      SH(8)=(1.-T*T)*(1.-S)/2.
CC
      RETURN
      END
CC **********************************************************************
      SUBROUTINE PMTX(S,T,PP,PB,BD,DT,XX,RHO,NN)
      IMPLICIT REAL*8 (A-H,O-Z)
CC -------------------------------------
CC  THIS SUBROUTINE COMPUTES THE P MATRIX
CC  AT AN INTEGRATION POINT
CC -------------------------------------
      COMMON /CONS/ THIC,NUMAT,NTYPE,NPOIN,NDIST,NNEL,
     1              NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT
      DIMENSION       PP(3,18), PB(3,3), BD(3), XX(2,8)
      DIMENSION       DB(3,8),DA(2,8),SH(8)
      DIMENSION       QQ(2,18),ABC(3,3),DC(2),XJB(2,2)
CC
      CALL SHAPEF(S,T,DA,DB,SH)
CC
CC   FORM THE INVERSE OF THE JACOBIAN MATRIX.
CC   DETERMINANT OF JACOBIAN IS DT
CC
      DO 20 I=1,2
      DO 20 J=1,2
      XJB(I,J)=0.
      DO 10 K=1,8
  10  XJB(I,J)=XJB(I,J)+DA(I,K)*XX(J,K)
  20  CONTINUE
      DT=XJB(1,1)*XJB(2,2)-XJB(1,2)*XJB(2,1)
CC
      DD = XJB(1,1)/DT
      XJB(1,1) = XJB(2,2)/DT
      XJB(2,2) = DD
```

```
      XJB(1,2) = -XJB(1,2)/DT
      XJB(2,1) = -XJB(2,1)/DT
CC
CC  ---  [ABC] MATRIX
CC
      ABC(1,1)=XJB(2,1)**2
      ABC(1,2)=XJB(2,2)**2
      ABC(1,3)=2.*XJB(2,1)*XJB(2,2)
      ABC(2,1)=XJB(1,1)**2
      ABC(2,2)=XJB(1,2)**2
      ABC(2,3)=2.*XJB(1,1)*XJB(1,2)
      ABC(3,1)=-XJB(1,1)*XJB(2,1)
      ABC(3,2)=-XJB(1,2)*XJB(2,2)
      ABC(3,3)=-XJB(1,1)*XJB(2,2)-XJB(1,2)*XJB(2,1)
CC
      DO 40 I=1,3
      DO 40 J=1,18
      PP(I,J)=0.
      IF(I.LE.2) QQ(I,J)=0.
 40   CONTINUE
CC
      PP(1,1)=2.
      PP(1,4)=6.*S
      PP(1,6)=2.*T
      PP(1,8)=12.*S*S
      PP(1,11)=2.*T*T
      PP(1,12)=6.*S*T
      PP(1,13)=20.*S*S*S
      PP(1,15)=12.*S*S*T
      PP(1,17)=6.*S*T*T
      PP(1,18)=2.*T*T*T
CC
      PP(2,2)=2.
      PP(2,5)=6.*T
      PP(2,7)=2.*S
      PP(2,9)=12.*T*T
      PP(2,10)=6.*S*T
      PP(2,11)=2.*S*S
      PP(2,14)=20.*T*T*T
      PP(2,16)=12.*S*T*T
      PP(2,17)=2.*S*S*S
```

```
      PP(2,18)=6.*S*S*T
CC
      PP(3,3)=1.
      PP(3,6)=2.*S
      PP(3,7)=2.*T
      PP(3,10)=3.*T*T
      PP(3,11)=4.*S*T
      PP(3,12)=3.*S*S
      PP(3,15)=4.*S*S*S
      PP(3,16)=4.*T*T*T
      PP(3,17)=6.*S*S*T
      PP(3,18)=6.*S*T*T
CC
      QQ(1,1)=2.*S
      QQ(1,3)=T
      QQ(1,4)=3.*S*S
      QQ(1,6)=2.*S*T
      QQ(1,7)=T*T
      QQ(1,8)=4.*S*S*S
      QQ(1,10)=T*T*T
      QQ(1,11)=2.*S*T*T
      QQ(1,12)=3.*S*S*T
      QQ(1,13)=5.*S*S*S*S
      QQ(1,15)=4.*T*S*S*S
      QQ(1,16)=T*T*T*T
      QQ(1,17)=3.*S*S*T*T
      QQ(1,18)=2.*S*T*T*T
CC
      QQ(2,2)=2.*T
      QQ(2,3)=S
      QQ(2,5)=3.*T*T
      QQ(2,6)=S*S
      QQ(2,7)=2.*S*T
      QQ(2,9)=4.*T*T*T
      QQ(2,10)=3.*S*T*T
      QQ(2,11)=2.*S*S*T
      QQ(2,12)=S*S*S
      QQ(2,14)=5.*T*T*T*T
      QQ(2,15)=S*S*S*S
      QQ(2,16)=4.*S*T*T*T
      QQ(2,17)=2.*T*S*S*S
```

```
      QQ(2,18)=3.*S*S*T*T
CC
      DO 60 I=1,2
      DC(I)=0.
      DO 60 JJ=1,8,2
      J = (JJ-1)/2
      DC(I) = DC(I)+0.25*(XX(I,JJ)*(-1.0)**J)
   60 CONTINUE
CC
      D1=DC(1)
      D2=DC(2)
      DO 70 I=1,2
      DC(I) = D1*XJB(1,I)+D2*XJB(2,I)
   70 CONTINUE
CC
      DO 80 J=1,18
      PP(3,J)=PP(3,J)-DC(1)*QQ(1,J)-DC(2)*QQ(2,J)
   80 CONTINUE
CC
      DO 100 J=1,18
      D1=PP(1,J)
      D2=PP(2,J)
      D3=PP(3,J)
      DO 100 I=1,3
      DD = D1*ABC(I,1)+D2*ABC(I,2)+D3*ABC(I,3)
      PP(I,J) = DD
  100 CONTINUE
CC
CC  BODY LOAD ARRAYS
CC
      DO 110 I = 1,3
      BD(I) = 0.
      DO 110 J = 1,3
      PB(I,J) = PP(I,J)
  110 CONTINUE
CC
      IF(GRAV.EQ.0.0) GOTO 150
      DO 120 I=1,3
      J=I+1
      K=I+2
      IF(J.GT.3) J=J-3
```

```
      IF(K.GT.3) K=K-3
      BD(I) = PB(3,J)*PB(1,K)-PB(3,K)*PB(1,J)
 120  CONTINUE
CC
      DET=PB(2,1)*BD(1)+PB(2,2)*BD(2)+PB(2,3)*BD(3)
      IF(DET.EQ.0) THEN
      WRITE(*,*) 'ZERO DETERMINANT AT BODY LOAD SECTION.'
      WRITE(*,*) 'PROGRAM ABORTS IN SUBROUTINE PMTX'
      STOP
      ENDIF
CC
      Y0=0.
      DO 130 I=1,8
 130  Y0=Y0+SH(I)*XX(2,I)
      DO 140 I=1,3
      BD(I)=RH0*BD(I)/DET
 140  CONTINUE
CC
 150  RETURN
      END

CC ***************************************************************************
      SUBROUTINE HMTX(NQP,NL)
      IMPLICIT REAL*8 (A-H,0-Z)
CC
      COMMON /BLK1/ PP(3,18),DB(3,8),DA(2,8),PB(3,3),SH(8),BD(3),DT
      COMMON /BLK2/ XX(2,8), SS(3,3),GPT(4),GWT(4),KXY(8)
      COMMON /OAKS/ HMX(18,18),GG(18,16),XLL(2,16),HB(18)
      COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
      DIMENSION     PQ(18,3),AP(3),BP(3),CP(3)
      INTEGER       NQP,NL
CC
CC  INITIALIZE
CC
       RH0 = 1.0
       DO 10 I = 1,18
       HB(I) = 0.0
       DO 10 J = 1,18
       HMX(I,J) = 0.0
  10   CONTINUE
CC
CC  NUMERICAL INTEGRATION PART, AT EVERY INTEGRATION POINT.
```

```
CC
        DO 100 II = 1,NQP
        SI = GPT(II)
        WI = GWT(II)
        DO 100 JJ = 1,NQP
        TI = GPT(JJ)
        WJ = GWT(JJ)
CC
CC  FORM THE P MATRIX AT THE INTEGRATION POINT
CC
       CALL PMTX(SI,TI,PP,PB,BD,DT,XX,RHO,NL)
CC
CC
CC  ---- FORM THE P TRANSPOSE X S MATRIX
CC
       DO 50 I=1,18
       DO 30 K = 1,3
       AP(K) = PP(K,I)
  30   CONTINUE
       DO 50 J=1,3
       DO 40 K = 1,3
       BP(K) = SS(K,J)
  40   CONTINUE
       PQ(I,J) = DOTPRD(AP,BP,3)
  50   CONTINUE
CC
CC  --- FORM THE (PQ MATRIX  X P TRANSPOSE
CC
       DO 90 I=1,18
       DO 60 K = 1,3
       AP(K) = PQ(I,K)
  60   CONTINUE
       DO 90 J=1,18
       DO 70 K = 1,3
       BP(K) = PP(K,J)
       IF(J.GT.3) GOTO 70
       CP(K) = PB(K,J)
  70   CONTINUE
       DD = DOTPRD(AP,BP,3)
       HMX(I,J) = HMX(I,J) + DD * WI * WJ * DT
       IF(J.GT.3) GOTO 90
```

```
      DD = DOTPRD(AP,CP,3)
      HB(I)=HB(I) + DD * WI * WJ * DT * BD(J)
   90 CONTINUE
  100 CONTINUE
CC
      RETURN
      END
CC ********************************************************************
      SUBROUTINE LMTX(S,T,XLL,IS)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XLL(2,16)
CC
      S1=(1.-S)/2.
      S2=(1.+S)/2.
      T1=(1.-T)/2.
      T2=(1.+T)/2.
      DO 20 I=1,2
      DO 20 J=1,16
      XLL(I,J)=0.
   20 CONTINUE
CC
      GOTO(30,40,50,60),IS
   30 XLL(1,1)=-S*S1
      XLL(1,3)=4.*S1*S2
      XLL(1,5)=S*S2
      XLL(2,2)=XLL(1,1)
      XLL(2,4)=XLL(1,3)
      XLL(2,6)=XLL(1,5)
      GOTO 70
CC
   40 XLL(1,5)=-T*T1
      XLL(1,7)=4.*T1*T2
      XLL(1,9)=T*T2
      XLL(2,6)=XLL(1,5)
      XLL(2,8)=XLL(1,7)
      XLL(2,10)=XLL(1,9)
      GOTO 70
CC
   50 XLL(1,9)=S*S2
      XLL(1,11)=4.*S1*S2
      XLL(1,13)=-S*S1
```

```
      XLL(2,10)=XLL(1,9)
      XLL(2,12)=XLL(1,11)
      XLL(2,14)=XLL(1,13)
      GOTO 70
CC
   60 XLL(1,13)=T*T2
      XLL(1,15)=4.*T1*T2
      XLL(1,1)=-T*T1
      XLL(2,14)=XLL(1,13)
      XLL(2,16)=XLL(1,15)
      XLL(2,2)=XLL(1,1)
   70 CONTINUE
      RETURN
      END
CC ****************************************************************
      SUBROUTINE MATINV(HH,N)
      IMPLICIT REAL*8 (A-H,O-Z)
CC
CC  THIS ROUTINE INVERTS THE HH MATRIX AND RETURNS IT
CC  ALSO AS HH
CC
      COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
      DIMENSION HH(N,N),A(18,18)
CC
       L10 = 10
      DO 20 I=1,N
      DO 20 J=1,N
      A(I,J)=0.0
      IF(I.NE.J) GOTO 20
      A(I,I)=1.0
  20  CONTINUE
CC
      DO 80 J=1,N
CC
CC   ROW OPERATION. OBTAIN UNITY IN (I,I) POSITION BY
CC   DIVIDING ROW I BY H(I,I)
CC
      DD=HH(J,J)
      DO 40 K=1,N
      HH(J,K)=HH(J,K)/DD
      A(J,K)=A(J,K)/DD
```

```
 40   CONTINUE
CC
CC   COLUMN OPERATION. FOR THE JTH. ROW OF COLUMN I,
CC   (I.NE.J), ADD -A(J,I) TIMES THE ITH. ROW TO THE
CC   JTH. ROW SO AS TO OBTAIN ZEROS IN THE OFF-DIAGONAL
CC   ELEMENTS OF COLUMN I.
CC
      DO 70 I=1,N
      IF(I.EQ.J) GOTO 70
      DD=HH(I,J)
      DO 60 K=1,N
      HH(I,K)=HH(I,K)-HH(J,K)*DD
      A(I,K)=A(I,K)-A(J,K)*DD
 60   CONTINUE
 70   CONTINUE
 80   CONTINUE
      DO 100 I=1,N
      DO 100 J=1,N
      HH(I,J)=A(I,J)
100   CONTINUE
150   CONTINUE
      RETURN
      END
CC ********************************************************************
      SUBROUTINE INPUT(TITLE,XORD,YORD,QSLOD,NUMEL,
     +                 LDVEC,NELTYP,DENS,CAR)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONS/ THIC,NUMAT,NTYPE,NPOIN,NDIST,NNEL,
     +              NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT
      COMMON /BLK2/ XX(2,8), SS(3,3),GPT(4),GWT(4),KXY(8)
      COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
      DIMENSION     PROPS(8,3),DLOAD(200),JDSIDE(200)
      DIMENSION     NLM(800,8),PL(800,2),JJX(4000)
      DIMENSION     NPL(800),JDOF(2000,2)
      DIMENSION     XORD(1),YORD(1),QSLOD(1),NUMEL(1),LDVEC(1),
     +              NELTYP(1),DENS(1),CAR(1)
      CHARACTER*1   TITLE(80)
C
CC
CC  READ PROBLEM TITLE, PROBLEM SPECIFICATION DATA
CC
```

```
      OPEN(UNIT=LIN8,FILE='INFILE',STATUS='OLD',
     +     ACCESS='SEQUENTIAL',FORM='FORMATTED')
      READ(LIN8,1) TITLE
    1 FORMAT(80A1)
      WRITE(*,*) TITLE
      READ(LIN8,*) NNODE,NNEL,NUMAT,NTYPE,NPOIN,NDIST,
     + IPRNT,THIC
      WRITE(*,*) NNODE,NNEL,NUMAT,NTYPE,NPOIN,
     + IPRNT,THIC
CC
CC  INITIALIZE ARRAYS
CC
CC  GRAV    = 0. => THERE ARE NO GRAVITY LOADS
CC        NE. 0. => THERE ARE GRAVITY LOADS = ACTUAL VALUE
CC
CC  QSLOD        = NODAL DOF LOADS
CC
CC  NNODE    = TOTAL NO. OF NODES
CC  NNEL     = TOTAL NO. OF ELEMENTS
CC  IPRNT    = PRINT OUTPUT DATA OPTION
CC           = 0  => DO NOT PRINT
CC           = 1  => PRINT
CC  THIC     = ELEMENT THICKNESS, ASSUMED CONSTANT
CC  DEPTH    = DEPTH OF ORIGIN BELOW SURFACE (NECESSARY IF GRAV > 0.)
CC
CC  NUMAT       = TOTAL NO. OF MATERIAL TYPES
CC  NTYPE       = PROBLEM TYPE PARAMETER
CC              = 1 => PLANE STRESS
CC              = 2 => PLANE STRAIN
CC  NUMBERING OF THE BOUNDARY EDGES IS ANTICLOCKWISE, STARTING FROM
CC  THE LEFTMOST VERTICAL BOUNDARY
CC
      NEQ = NNODE*2
      DO 2 I = 1,NEQ
      QSLOD(I) = 0.0
    2 CONTINUE
CC
CC  READ ELEMENT PROPERTIES DATA
CC
CC  PROPS(I,1)  = YOUNG'S MODULUS FOR MATERIAL TYPE I
CC  PROPS(I,2)  = POISSON'S RATIO FOR MATERIAL TYPE I
```

```
CC  PROPS(I,3)  = DENSITY FOR MATERIAL TYPE I
CC  SS(3,3)     = COMPLIANCE MATRIX
CC
        IJ = 0
        DO 10 I = 1,NUMAT
        READ(LIN8,*) (PROPS(I,J),J=1,3)
        DO 10 J = 1,3
        IJ = IJ+1
        CAR(IJ) = 0.0
        DENS(IJ) = 0.0
  10    CONTINUE
        CALL ELPROP(NUMAT,NTYPE,PROPS,CAR,DENS)
       N3 = NUMAT*3
CC
CC  READ NODAL DATA
CC
CC  XORD(I) = X-COORDINATE OF NODE I
CC  YORD(I) = Y-COORDINATE OF NODE I
CC  JDOF(I,1) = X DOF OF NODE I
CC  JDOF(I,2) = Y DOF OF NODE I
CC  JDOF(I,J) = 0 => NODE I IS UNCONSTRAINED IN DIRECTION J
CC  JDOF(I,J) = 1 => NODE I IS CONSTRAINED IN DIRECTION J
CC
        NEQ = 0
        DO 20 I = 1,NNODE
        READ(LIN8,*) N, XORD(N), YORD(N), JDOF(N,1), JDOF(N,2)
        DO 20 J = 1,2
        NEQ = NEQ+1
        JJX(NEQ) = JDOF(N,J)
  20    CONTINUE
CC
CC  READ ELEMENT GEOMETRIC DATA
CC
CC  NLM(I,J) = NODE J OF ELEMENT I (J = 1,2, ..,8)
CC  NELTYP(I) = PROPERTY TYPE OF ELEMENT I
CC
        KK = 0
        DO 30 I = 1,NNEL
        READ(LIN8,*) N,NELTYP(I),(NLM(I,J),J=1,8)
 30     CONTINUE
CC
```

```
      NTOT = 0
      DO 40 I = 1,NNEL
      DO 40 J = 1,8
      NTOT = NTOT+1
      NUMEL(NTOT) = NLM(I,J)
 40   CONTINUE
CC
CC  --- MAXIMUM HALF BANDWIDTH
CC
     MAXBAN=2*(NLM(1,5)-NLM(1,1)+1)
CC
CC  FORM DIAGONAL POSITION POINTER FOR SKYLINE STORAGE SCHEME
CC
      CALL SKDIAG(NEQ,NNODE,NNEL,JJX,LDVEC,NUMEL,NTOT,NWA)
CC
CC  READ LOADING DATA
CC
CC  NPOIN = NODAL POINT LOAD PARAMETER
CC        = 0 => NO POINT LOAD
CC        > 0 => THERE ARE POINT LOADS
CC
CC      IF NPOIN > 0, THEN FOLLOWS NPOIN LINES OF DATA OF TYPE
CC
CC      KNODE  XFORCE  YFORCE
CC
CC      WHERE
CC          KNODE  = NODAL NUMBER
CC          XFORCE = POINT LOAD IN X-DIRECTION
CC          YFORCE = POINT LOAD IN Y-DIRECTION
CC
CC  NDIST = DISTRIBUTED LOAD PARAMETER
CC        = 0 => THERE ARE NO DISTRIBUTED LOADS
CC        > 0 => THERE ARE DISTRIBUTED LOADS ON NDIST BOUNDARY EDGES
CC
CC         IF NDIST > 0, THEN FOLLOWS NDIST LINES OF DATA OF TYPE
CC
CC          KSIDE   KFUNC   FORCE1   FORCE2
CC
CC      WHERE
CC          KSIDE  = BOUNDARY NUMBER
CC          KFUNC  = DISTRIBUTION FUNCTION
```

```
CC                = 1, LINEAR
CC                = 2, QUADRATIC
CC          FORCE1 = DISTRIBUTED LOAD VALUE AT FIRST END
CC          FORCE2 = DISTRIBUTED LOAD VALUE AT SECOND END
CC
        IF(NPOIN.EQ.0) GOTO 60
         J = 0
        DO 50 I = 1,NPOIN
        READ(LIN8,*) KK, XF, YF
        J = J+1
        NPL(J) = KK
        PL(J,1) = XF
        PL(J,2) = YF
        QSLOD(KK+KK-1) = QSLOD(KK+KK-1)+XF
        QSLOD(KK+KK) = QSLOD(KK+KK)+YF
  50    CONTINUE
CC
CC   DISTRIBUTED LOAD INPUT
CC
CC   NDIST = NO. OF SIDES WITH UNIFORMLY DISTRIBUTED LOADS.
CC   JDSIDE(I,J) = ARRAY DESCRIBING GEOMETRIC PARAMETERS OF ELEMENT
CC                 SIDE WITH UNIFROMLY DISTRIBUTED LOADS.
CC           I = 1,2, ..., NDIST,
CC           J = 1,2,3,4
CC           J = 1 --> FIRST NODE ON THE ELEMENT SIDE
CC           J = 2 --> SECOND NODE ON THE ELEMENT SIDE
CC           J = 3 --> THIRD NODE ON THE ELEMENT SIDE
CC           J = 4 --> ELEMENT NUMBER
CC
CC   DLOAD(I,J) = JTH COMPONENT OF DISTRIBUTED LOAD ON ITH SIDE
CC           I = 1,2, ...., NDIST,
CC           J = 1, --> WX1, X-COMPONENT AT FIRST NODE
CC           J = 2, --> WY1, Y-COMPONENT AT FIRST NODE
CC           J = 3, --> WX3, X-COMPONENT AT THIRD NODE
CC           J = 4, --> WY3, Y-COMPONENT AT THIRD NODE
CC
  60    IF(NDIST.EQ.0) GOTO 80
        write(*,*) 'reading distributed load data'
        DO 70 I = 1,NDIST
        J1 = 4*I-3
        J2 = 4*I
```

```
      READ(LIN8,*) (JDSIDE(J),J=J1,J2),(DLOAD(J),J=J1,J2)
 70   CONTINUE
      NDTOT = NDIST*4
 7    FORMAT(4I7,4F10.3)
CC
      CALL DSLOD (NUMEL,NTOT,NDTOT,GPT,GWT,QSLOD,XORD,YORD,
     +                  DLOAD,JDSIDE,THIC,NQP,NEQ,NNODE)
CC
 80   CONTINUE
CC
CC  WRITE OUTPUT FILES FOR OTHER PROCESSES
CC
      WRITE(LIN9,*) (QSLOD(I),I=1,NEQ)
 500  FORMAT(' ',6F12.4)
      WRITE(LIN2,*) (XORD(I),YORD(I),I=1,NNODE)
      WRITE(LIN2,*) (NELTYP(I),I=1,NNEL)
      WRITE(LIN2,*) (NUMEL(I),I=1,NTOT)
      WRITE(LIN2,*) (DENS(I),I=1,N3)
CC
CC  PPINT INPUT DATA
CC
      IF(IPRNT.EQ.0) GOTO 100
      CALL OUTPUT(NLM,NELTYP,NPL,PL,DL,PROPS,DLOAD,JDSIDE,
     1            TITLE,THIC,NUMAT,NTYPE,NPOIN,NDTOT,NNEL,
     1            NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT)
CC
 100  CONTINUE
      RETURN
      END

CC ***************************************************************************
      SUBROUTINE ELPROP(NUMAT,NTYPE,PROPS,CAR,DENS)
      IMPLICIT REAL*8 (A-H,O-Z)
CC
CC  THIS SUBROUTINE COMPUTES THE COMPLIANCE MATRIX FOR ALL THE
CC  DIFFERENT MATERIALS AND STORES THEM IN A LINEAR ARRAY CAR(*)
CC
CC  IN GENERAL, THE COMPLIANCE MATRIX WILL BE DIFFERENT FOR EACH
CC  MATERIAL TYPE. ASSUMING LINEAR ELASTIC CASE, THIS MATRIX IS
CC  SYMMETRIC AND IS DESCRIBED BY THREE CONSTANTS,VIZ
CC  SS(1,1), SS(1,2) AND SS(3,3).
CC  THESE THREE CONSTANTS MAY BE COMPUTED FOR EACH MATERIAL TYPE
```

```
CC  AND STORED IN A COMPLIANCE ARRAY CAR(3I-2),CAR(3I-1)AND
CC  CAR(3I) WHERE
CC                I = 1,2, ..., NUMAT,
CC                CAR(3I-2) => SS(1,1)
CC                CAR(3I-1) => SS(1,2) OR SS(2,1)
CC                CAR(3I)   => SS(3,3)
CC
      DIMENSION PROPS(8,3), CAR(1),DENS(1)
CC
CC  GRAV = ACCELERATION DUE TO GRAVITY (KM/SEC SQ)
CC
      GRAV = 9.80665
      DO N = 1,NUMAT
       N3 = N*3
        YM = PROPS(N,1)
         PR = PROPS(N,2)
          BETA = YM/(1.0 - PR*PR)
            IF(NTYPE.EQ.2) THEN
           BETA = BETA*(1.0 - PR)*(1.0 - PR)/(1.0 - 2.0*PR)
          ENDIF
CC
CC  COMPLIANCE MATRIX CONSTANTS
CC
          GAMA = 1.0/(BETA*(1.0 - PR*PR))
           CAR(N3-2) = GAMA
            CAR(N3-1) = -PR*GAMA
             CAR(N3)   = 2.0*(1.0 + PR)/YM
               DENS(N) = GRAV*PROPS(N,3)*1.0E-03
      ENDDO
CC
      RETURN
      END
CC  ****************************************************************
       SUBROUTINE DSLOD (NUMEL,NTOT,NDTOT,GPT,GWT,QSLOD,XORD,YORD,
     +                   DLOAD,JDSIDE,THIC,NQP,NEQ,NNODE)
       IMPLICIT REAL*8 (A-H,O-Z)
       COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
       DIMENSION QSLOD(1),GPT(1),GWT(1)
       DIMENSION XORD(1),YORD(1)
       DIMENSION NUMEL(1),DLOAD(1),JDSIDE(1)
       DIMENSION XLL(2,16),XX(2,8),KXY(8)
```

```
      DIMENSION CZ(2,4),IX(3), IZ(3), BZ(2)
      DIMENSION SH(8),DA(2,8),DB(3,8),AZ(16),WZ(4)
CC
      NNEL = NTOT/8
CC
C     WRITE(*,*) 'IN DSLOD'
CC
      NDIST = NDTOT/4
CC
      DO 500 NBD = 1,NDIST
      NB4 = 4*NBD
      NB1 = NB4-4
      NJ = JDSIDE(NB4)
      IF(NJ.EQ.0) GOTO 500
CC
CC  FORM THE [XX] AND {KXY} ARRAYS
CC
      N1 = NJ*8-7
      N2 = N1+7
      KK = 0
      DO 20 J = N1,N2
      JJ=NUMEL(J)
      KK = KK+1
      XX(1,KK)=XORD(JJ)
      XX(2,KK)=YORD(JJ)
      KXY(KK)=JJ
   20 CONTINUE
CC
CC FORM THE {IZ} AND {WZ} ARRAYS
CC
      DO 30 I = 1,4
      WZ(I) = DLOAD(NB1+I)
      IF(I.LE.3) IZ(I) = JDSIDE(NB1+I)
 30   CONTINUE
CC
CC  LENGTH OF THIS SIDE
CC
       I1 = IZ(1)
       I2 = IZ(3)
       X1 = XORD(I1) -XORD(I2)
       Y1 = YORD(I1) -YORD(I2)
```

```
        RADI = SQRT(X1**2 + Y1**2)
CC
CC  NODAL NUMBERING IS SEQUENTIAL FOR THE ELEMENT, IE
CC  1,2,3, ...,8 WHERE
CC  SIDE 1,(KL = 1), IS DESCRIBED BY NODES 1,2,3
CC  SIDE 2,(KL = 2), IS DESCRIBED BY NODES 3,4,5
CC  SIDE 3,(KL = 3), IS DESCRIBED BY NODES 5,6,7
CC  SIDE 4,(KL = 4), IS DESCRIBED BY NODES 7,8,1
CC  FIND THE KL VALUE FOR THIS SIDE
CC
       DO 50 I = 1,8,2
       I2 = (I+1)/2
       IX(1) = KXY(I)
       IX(2) = KXY(I+1)
       I1 = I+2
       IF(I1.GT.8) I1 = I1-8
       IX(3) = KXY(I1)
CC
CC  COMPARE {IX} AND {IZ} ARRAYS
CC
        KL = 3
        DO 40 J = 1,3
        IF(IX(J).EQ.IZ(J)) KL = KL-1
 40     CONTINUE
        IF(KL.EQ.0) THEN
        KL = I2
        GOTO 60
        ENDIF
 50     CONTINUE
CC
 60     CONTINUE
        I1 = KL*2-1
        I2 = I1+2
        J1 = KXY(I1)
        J2 = KXY(I2)
CC
CC   NUMERICAL INTEGRATION OVER SIDE KL
CC
       DO 300 II = 1,NQP
       SI = GPT(II)
       TI = SI
```

```
      WI = GWT(II)
      GOTO (70,80,90,100) KL
 70   TI = -1
      GOTO 110
 80   SI=1.0
      GOTO 110
 90   TI=1.0
      GOTO 110
 100  SI=-1.0
CC
 110  CONTINUE
      CI = SI
      IF(ABS(SI).EQ.1.0) CI = TI
CC
      CALL SHAPEF(SI,TI,DA,DB,SH)
      CALL LMTX(SI,TI,XLL,KL)
CC
CC
CC   FORM THE CZ MATRIX
CC
       DO 152 I = 1,2
       DO 152 J = 1,4
       CZ(I,J) = 0.0
 152   CONTINUE
CC
       CZ(1,1) = (1.0-CI)/2.0
       CZ(1,3) = (1.0+CI)/2.0
       CZ(2,2) = (1.0-CI)/2.0
       CZ(2,4) = (1.0+CI)/2.0
CC
CC   [CZ] * {WZ} = {BZ}
CC
       DO 170 I = 1,2
       DO 160 J = 1,4
       AZ(J) = CZ(I,J)
 160   CONTINUE
       BZ(I) = DOTPRD(AZ,WZ,4)
 170   CONTINUE
CC
CC  NOW FORM THE CONSISTENT LOAD VECTOR FOR THIS SIDE
CC
```

```
        DO 185 I = 1,16
        DD = BZ(1)*XLL(1,I) + BZ(2)*XLL(2,I)
        AZ(I) = DD*THIC*WI*RADI/2.0
 185    CONTINUE
CC
CC  ASSOCIATE CONSISTENT LOAD VECTOR WITH GLOBAL LOAD ARRAY
CC
         DO 210 I = 1,8
         I1 = I*2
         J = KXY(I)*2
         QSLOD(J-1) = QSLOD(J-1) + AZ(I1-1)
         QSLOD(J) = QSLOD(J) + AZ(I1)
 210     CONTINUE
CC
 300    CONTINUE
 500    CONTINUE
CC
        RETURN
        END
CC **********************************************************************
       SUBROUTINE OUTPUT(NLM,NELTYP,NPL,PL,DL,
     1                   PROPS,DLOAD,JDSIDE,TITLE,
     1               THIC,NUMAT,NTYPE,NPOIN,NDTOT,NNEL,
     1               NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT)
       IMPLICIT REAL*8 (A-H,O-Z)
CC
       COMMON /BLK1/ PP(3,18),DB(3,8),DA(2,8),PB(3,3),SH(8),BD(3),DT
       COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
       DIMENSION     DL(4,2),PL(65,2),NLM(64,8),NPL(65)
       DIMENSION     NELTYP(1),PROPS(NUMAT,1)
       DIMENSION     DLOAD(1),JDSIDE(1)
       CHARACTER*1   TITLE(80),STAR(80)
CC
CC     PRINT RESULTS
CC
        NDIST = NDTOT/4
        DO 5 I = 1,80
        STAR(I) ='*'
 5      CONTINUE
        IF(IPRNT.EQ.0) GOTO 700
CC
```

```
      WRITE(LIN1,10) STAR
      WRITE(LIN1,10) (TITLE(I),I = 1,80)
      WRITE(LIN1,10) STAR
 10   FORMAT(/,80A1)
      WRITE(LIN1,500) NNODE,NNEL,NUMAT,THIC
CC
C     WRITE(LIN1,530)
C     DO 20 I = 1,NNODE
C     WRITE(LIN1,550) I,XORD(I),YORD(I),JDOF(I,1),JDOF(I,2)
C 20  CONTINUE
CC
C     WRITE(LIN1,510)
C     DO 30 I=1,NNEL
C     WRITE(LIN1,520) I,NELTYP(I),(NLM(I,J),J=1,8)
C 30  CONTINUE
CC
C     WRITE(LIN1,650)
C     DO 25 I = 1,NUMAT
C     WRITE(LIN1,615) I, PROPS(I,1),PROPS(I,2),PROPS(I,3)
C 25  CONTINUE
CC
C     IF(NTYPE.EQ.1) THEN
C     WRITE(LIN1,580)
C     ELSE
C     WRITE(LIN1,590)
C     ENDIF
CC
C     IF(NPOIN.GT.0) THEN
C     WRITE(LIN1,610)
C     DO 40 I = 1,NPOIN
C     WRITE(LIN1,620) NPL(I),PL(I,1),PL(I,2)
C 40  CONTINUE
C     ENDIF
CC
C     IF(NDIST.GT.0) THEN
C     WRITE(LIN1,640)
C     DO 50 I = 1,NDIST
C     J1 = 4*I-3
C     J2 = 4*I
C     WRITE(LIN1,645) JDSIDE(J2),(JDSIDE(J),J=J1,J2-1),
C    1                (DLOAD(J),J=J1,J2)
```

```
C 50   CONTINUE
C      ENDIF
CC
C      WRITE(LIN1,560)
C      WRITE(LIN1,570) (LDVEC(I),I=2,NEQ+1)
CC
C      WRITE(LIN1,675)
C       DO 60 I = 1,NEQ,2
C       J = (I+1)/2
C       WRITE(LIN1,690) J,QSLOD(I),QSLOD(I+1)
C 60    CONTINUE
C      WRITE(LIN1,10) STAR
C      WRITE(LIN1,680)
C      WRITE(LIN1,10) STAR
CC
 500   FORMAT(' ',//,5X,'TOTAL NO. OF NODES           =',I10/,
     +               5X,'TOTAL NO. OF ELEMENTS        =',I10/,
     +               5X,'TOTAL NO. OF MATERIAL TYPES  =',I10/,
     +               5X,'ELEMENT THICKNESS            =',F10.3/)
C
 510   FORMAT(' ',/,15X,'ELEMENT NODE NUMBERING',/,
     + ' ELEM  ELEM  NODE1  NODE2  NODE3  NODE4',
     + '  NODE5  NODE6  NODE7  NODE8',/,
     + '  NO   TYPE',/)
 520   FORMAT(I4,4X,2,8(4X,I3))
 530   FORMAT(/,'  NODE    X-COORDINATE    Y-COORDINATE     ',
     + 'X-DOF    Y-DOF',/)
 540   FORMAT(' ',I5,2(6X,F10.5))
 550   FORMAT(' ',I5,2(6X,F10.5),2(6X,I3))
 560   FORMAT(/,' ',' DIAGONAL ELEMENT POSITIONS IN THE ',
     + 'SKYLINE VECTOR. ',/,' (NEGATIVE VALUE',
     + ' INDICATES CONSTRAINED D.O.F)'/)
 570   FORMAT(10I7)
 580   FORMAT(/,20X,' PROBLEM TYPE        PLANE STRESS',/)
 590   FORMAT(/,20X,' PROBLEM TYPE        PLANE STRAIN',/)
 610   FORMAT(20X,' POINT LOAD DATA',/,
     + 10X, ' NODE NO.     X-FORCE        Y-FORCE',/)
 615   FORMAT(10X,I5,8X,F9.3,2(8X,F5.3))
 620   FORMAT(10X,I5,8X,F9.3,5X,F8.3)
 640   FORMAT(20X,' DISTRIBUTED LOAD DATA',/,
     + 5X,'ELEM. NO.  NODE1  NODE2  NODE3  XLOAD1',
```

```
     + 2X,'YLOAD1  XLOAD3  YLOAD3',/)
 645   FORMAT(2(5X,I5),2(4X,I3),1X,4(1X,F7.2))
 650   FORMAT(/,20X,' ELEMENT MATERIAL PROPERTIES ',//,
     + 10X, ' MATERIAL     YOUNG"S        POISSON"S    DENSITY',/,
     + 10X, '   TYPE         MOD           RATIO      (KG/CU.M)',/)
 675   FORMAT(/,10X,' NODAL LOADS ',/,
     + '  NODE       X-LOAD          Y-LOAD',/)
 680   FORMAT(/,10X,' NODAL DISPLACEMENTS ',/,
     + '  NODE   X-DISPLACEMENT  Y-DISPLACEMENT',/)
 690   FORMAT(1X,I4,2(6X,E10.3))
CC
 700   CONTINUE
       RETURN
       END
CC ***************************************************************************
       SUBROUTINE SKDIAG(NEQ,NNODE,NNEL,JJX,LDVEC,NUMEL,NTOT,NWA)
       IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
       DIMENSION JJX(1),LDVEC(1),KH(2000)
       DIMENSION NUMEL(1),JD(2000)
CC
CC  ---  VECTOR LDVEC HOLDS THE POSITION NUMBERS OF THE DIAGONAL ELEMENTS
CC
       DO 20 I=1,NNODE
  20   JD(I)=0
       DO 50 II=1,NNEL
       I1 = II*8 - 7
       I2 = I1 + 7
CC
CC  ---  FIND SMALLEST NODE ( = NODE 1)
CC
       N1=100000
       DO 30 J=I1,I2
       JJ=NUMEL(J)
       IF(N1.GT.JJ) N1=JJ
  30   CONTINUE
CC
CC  ---  FIND SKYLINE POSITION
CC
       NSK=N1*2-2
CC
```

```
CC  ---  FIND THOSE NODES WITHOUT SKYLINE
C   ---  HEIGHT OF DIAGONAL ELEMENT BELOW SKYLINE = D.O.F-NSK
CC
      DO 40 JJ=I1,I2
      J=NUMEL(JJ)
      IF(JD(J).EQ.1) GOTO 40
      J1=J*2-1
      J2=J*2
      KH(J1)=J1-NSK
      KH(J2)=J2-NSK
      JD(J)=1
 40   CONTINUE
 50   CONTINUE
CC
CC  ---  ESTABLISH SKYLINE DIAGONAL VECTOR
CC
      LDVEC(1)=0
      DO 60 I=1,NEQ
      LDVEC(I+1)=LDVEC(I)+KH(I)
 60   CONTINUE
CC
      NWA=LDVEC(NEQ+1)
      DO 70 I=1,NEQ
      J=I+1
      IF(JJX(I).GT.0) THEN
           LDVEC(I+1)=-LDVEC(I+1)
      ENDIF
 70   CONTINUE
CC
      RETURN
      END
CC **********************************************************************
       SUBROUTINE SKYFAC (A,LD,NWA,NBEG,NEND)
       IMPLICIT REAL*8 (A-H,O-Z)
CC
CC  PERFORMS SYMMETRIC INCORE FACTORIZATION OF A
CC  SPARSE, BANDED, SKYLINE-STORED, SYMMETRIC MATRIX.
CC
       COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
       DIMENSION        A(1), LD(1), V(20000)
       REAL        AIJ2, D,  EPSMAC, UMAX, UMIN
```

```
      LOGICAL     SINGAB
      EQUIVALENCE (AIJ2,D,DMAX)
CC
      DATA  EPSMAC/1.49E-8/, UMAX /1.00/, UMIN /0.625/
CC
CC   INITIALIZATION
CC
      REWIND LIN4
      READ (LIN4,*) (A(I),I=1,NWA), (LD(I),I=1,NEND+1)
 9000 FORMAT(1X,4E20.14)
 909  FORMAT(10I7)
      IKJ  = 0
      SINGAB = .TRUE.
CC
CC   COMPUTE SQUARED LENGTHS OF UNCONSTRAINED ROWS NBEG+1 THRU NEND
CC
  200 NBEGP1 = NBEG+1
      DO 1000 I = NBEGP1, NEND
      II = LD(I+1)
      IF(II)  1000,1000,400
  400 V(I) = A(II)**2
      M = II-I
      K = MAX0(NBEGP1,IABS(LD(I))-M+1)
      L = MIN0(NEND,I) - 1
      IF(K-L)  500,500,1000
  500 DO 800 J = K,L
      IF(LD(J+1))  800,800,600
  600 AIJ2 = A(M+J)**2
      V(I) = V(I) + AIJ2
      V(J) = V(J) + AIJ2
  800 CONTINUE
 1000 CONTINUE
CC
CC   FACTORIZATION SECTION
CC
      DO 4000 J = NBEGP1,NEND
CC
CC   COMPUTE KU SUPERDIAGONAL ENTRIES OF JTH COLUMN OF [U]
CC   IF UNCONSTRAINED
CC
      JJ = LD(J+1)
```

```
      IF(JJ)  4000,4000,1200
 1200 D = A(JJ)
      JMJ = IABS(LD(J))
      JK = JJ - JMJ
      KU = JK - 1
      IF(KU.EQ.0)  GOTO 2200
      DO 2000 K = 1,KU
      I = J - JK + K
      V(K) = 0.0
      II = LD(I+1)
      IF(II)  2000,2000,1800
 1800 M = MIN0(II - IABS(LD(I)),K) - 1
      IJ = JMJ + K
      V(K) = A(IJ) - DOTPRD(A(II-M),V(K-M),M)
      A(IJ) = V(K)*A(II)
 2000 CONTINUE
CC
CC   COMPUTE DIAGONAL ELEMENT
CC
      D = D - DOTPRD(A(JMJ+1),V,KU)
 2200 CONTINUE
CC
CC  SINGULARITY TEST
CC
      TOLROW = 8.0 * EPSMAC * SQRT(V(J))
      IF(ABS(D).GT.TOLROW) GOTO 2500
      IF(SINGAB) GOTO 6000
      D = TOLROW
 2500 A(JJ) = 1.0/D
 4000 CONTINUE
 5000 CONTINUE
CC
      REWIND LIN4
      WRITE(LIN4,*) (A(I),I=1,NWA), (LD(I),I=1,NEND+1)
      RETURN
CC
CC  ERROR EXIT
CC
 6000 WRITE(*,*) 'FATAL ERROR'
      STOP
      END
```

```
CC ***********************************************************************
        SUBROUTINE SKYSOL(A,LD,NWA,N,IOP,IBX,B,X,NQ)
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
CC
        DIMENSION  X(1), A(1), B(1), LD(1)
        REAL       BI,   BXFAC
        REAL       XI, XNORM
        EQUIVALENCE  (BI, XI, XNORM)
CC
CC    INITIALIZATION
CC
        REWIND(LIN4)
        READ (LIN4,*) (A(I),I=1,NWA), (LD(I),I=1,NQ+1)
 909    FORMAT(10I7)
 9000   FORMAT(1X,4E20.14)
        KREF = 1
        BXFAC = 0.0
        IF(IBX.EQ.0) GOTO 200
        BXFAC = 1.0
        DO 150 I = 1,N
  150   X(I) = B(I)
  200   IF(IOP.GT.0) GOTO 1800
        IF(IBX.EQ.0) GOTO 1100
CC
CC    RHS MODIFICATION
CC
        DO 1000 I = 1,N
        II = LD(I+1)
        IF(II)  300, 1000, 1000
  300   BI = B(I)
        IF(BI.EQ.0.0) GOTO 1000
        II = -II
        K = I - II + IABS(LD(I)) + 1
        DO 900 J = K,N
        JJ = LD(J+1)
        IF(JJ)  900,900,400
  400   M = J-I
        IF(M) 500,600,600
  500   X(J) = X(J) - A(II+M)*BI
        GOTO 900
```

```
 600    IJ = JJ - M
        IF(IJ-IABS(LD(J))) 900,900,800
 800    X(J) = X(J) - A(IJ)*BI
 900    CONTINUE
1000    CONTINUE
CC
CC   FORWARD SUBSTITUTION PASS
CC
1100    DO 1500 I = 1,N
        II = LD(I+1)
        IF(II) 1200,1200,1300
1200    X(I) = 0.0
        GOTO 1500
1300    IMI = IABS(LD(I))
        M = II - IMI -1
        X(I) = X(I) - DOTPRD(A(IMI+1),X(I-M),M)
1500    CONTINUE
        IF(IOP.NE.0) GOTO 5000
CC
CC   SCALING PASS
CC
1800    DO 2000 I = 1,N
        II = IABS(LD(I+1))
2000    X(I) = A(II)*X(I)
CC
CC   BACK SUBSTITUTION PASS
CC
        I = N
        DO 3000 K = 1,N
        II = LD(I+1)
        IF(II) 2200,2200,2400
2200    X(I) = BXFAC*B(I)
        GOTO 2800
2400    M = II - IABS(LD(I)) -1
        IF(M.EQ.0) GOTO 2800
        DO 2500 J = 1,M
        X(I-J) = X(I-J) - A(II-J)*X(I)
2500    CONTINUE
2800    I = I -1
3000    CONTINUE
 210    FORMAT(1X,I4,6X,E10.3,6X,E10.3)
```

```
CC
CC   CONSTRAINED RHS RECOVERY
CC
 4000    IF(IBX.LE.0)  GOTO 5000
         DO 4800 I = 1,N
         II = LD(I+1)
         IF(II) 4200,4200,4800
 4200    IMI = IABS(LD(I))
         M = -II -IMI -1
         B(I) = DOTPRD(A(IMI+1),X(I-M),M)
         DO 4600 J = 1,N
         IJ = IABS(LD(J+1)) + I - J
         IF(IJ-IABS(LD(J))) 4600,4600,4400
 4400    B(I) = B(I) + A(IJ)*X(J)
 4600    CONTINUE
 4800    CONTINUE
 5000     CONTINUE
         REWIND LIN4
         WRITE(LIN4,*) (X(I),I=1,NQ)
         RETURN
         END
CC ******************************************************************
         SUBROUTINE STRSOL(DISPL,XORD,YORD,NUMEL,DENS,CAR,NELTYP,
     +                     THIC,NUMAT,NTYPE,NPOIN,NDIST,NNEL,
     +                     NNODE,NEQ,NQP,NWA,MAXBAN,NTOT,IPRNT)
         IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /BLK1/ PP(3,18),DB(3,8),DA(2,8),PB(3,3),SH(8),BD(3),DT
        COMMON /BLK2/ XX(2,8), SS(3,3),GPT(4),GWT(4),KXY(8)
        COMMON /OAKS/ HMX(18,18),GG(18,16),XLL(2,16),HB(18)
        COMMON /LINE/ LIN1,LIN2,LIN3,LIN4,LIN7,LIN8,LIN9
        COMMON /BLOK/ SE(9),TE(9)
        DIMENSION    QQ(16),SIG(6),PX(2000),SPX(2000,6)
        DIMENSION    AMX(18,16),BMX(18),BETA(18), DS(3)
        DIMENSION    DENS(1),CAR(1),NUMEL(1),XORD(1),
     +               YORD(1),DISPL(1),NELTYP(1)
CC
         NUM3 = NUMAT*3
         REWIND LIN2
         READ (LIN2,*) (XORD(I),YORD(I),I=1,NNODE)
         READ (LIN2,*) (NELTYP(I),I=1,NNEL)
         READ (LIN2,*) (NUMEL(I),I=1,NTOT)
```

```
      READ (LIN2,*) (DENS(I),I=1,NUM3)
CC
CC READ DISPLACEMENT DATA FROM LIN4 AND WRITE
CC STRESSES AT ELEMENT CENTRES TO IT
CC
      REWIND LIN4
      READ (LIN4,*) (DISPL(I),I=1,NEQ)
      REWIND LIN4
      WRITE(LIN1,630)
      DO 7 I = 1,NEQ,2
      J = (I+1)/2
      WRITE(LIN1,650) J,XORD(J),YORD(J),DISPL(I),DISPL(I+1)
    7 CONTINUE
      PI=3.141592654
      DO 10 I=1,NNODE
      PX(I)=0.
      DO 10 J=1,6
      SPX(I,J)=0.
   10 CONTINUE
C     WRITE(LIN1,260)
CC
      REWIND LIN3
      NAM = 0
      DO 180 II=1,NNEL
      I = NELTYP(II)
      RHO = DENS(I)
      READ(LIN3,*) AMX
      READ(LIN3,*) BMX
CC
c     WRITE(LIN1,280) II
      N1 = II*8 - 7
      N2 = N1 + 7
      JJ = 0
      XTX = 0.0
      YTY = 0.0
      DO 20 KK= N1,N2
      J=NUMEL(KK)
      JJ = JJ+1
      XX(1,JJ)=XORD(J)
      XX(2,JJ)=YORD(J)
      XTX = XTX + XORD(J)/8.0
```

```
        YTY = YTY + YORD(J)/8.0
        KXY(JJ)=J
        JJ1=JJ*2-1
        JJ2=JJ*2
        J1=J*2-1
        J2=J*2
        QQ(JJ1)=DISPL(J1)
        QQ(JJ2)=DISPL(J2)
   20   CONTINUE
CC
CC  ----MATRIX AMX*QQ
CC
        DO 60 I=1,18
        DD=0.
        DO 40 J=1,16
   40   DD=DD+AMX(I,J)*QQ(J)
        BETA(I)=DD - BMX(I)
   60   CONTINUE
CC
CC  ---  NOW SOLVE FOR NODAL STRESSES AT ELEMENT CENTRE
CC       AND CORNER NODES ONLY
CC
CC  ---    SIG(J) = STRESSES AT NODE I
CC             J  = 1 => SIGMA-X
CC             J  = 2 => SIGMA-Y
CC             J  = 3 => TAU-XY
CC             J  = 4 => SIGMA-1
CC             J  = 5 => SIGMA-2
CC             J  = 6 => ANGLE OF SIGMA-1 FROM X-AXIS
CC
c       WRITE(LIN1,191)
        DO 140 I=1,9
        DO 75 J = 1,6
        SIG(J) = 0.
   75   CONTINUE
        S1=SE(I)
        T1=TE(I)
        CALL PMTX(S1,T1,PP,PB,BD,DELTA,XX,RHO,II)
        B1 = BD(1)
        B2 = BD(2)
        B3 = BD(3)
```

```
        DO 80 J = 1,3
        BD(J) = B1*PB(J,1) + B2*PB(J,2) + B3*PB(J,3)
   80   CONTINUE
        DO 120 J=1,3
        DD=0.
        DO 100 K=1,18
  100   DD=DD+PP(J,K)*BETA(K)
        DS(J)=DD + BD(J)
        SIG(J) = DS(J)
  120   CONTINUE
CC
        S1=(DS(1)+DS(2))/2.
        S2=(DS(1)-DS(2))/2.
        S3=DS(3)
        SIG(6)=ATAN2(S3,S2)*90.0/PI
        RS=SQRT(S2**2+S3**2)
        SIG(4)=S1+RS
        SIG(5)=S1-RS
CC
CC  SUM SIGMA-X, SIGMA-Y AND TAU-XY FOR THE RESPECTIVE NODES
CC
        IF(I.LT.9) THEN
          LL=KXY(I)
            PX(LL)=PX(LL)+1.0
              DO 125 K=1,3
                SPX(LL,K)=SPX(LL,K)+SIG(K)
 125             CONTINUE
C           WRITE(LIN1,591) KXY(I),(SIG(J),J=1,6)
        ELSE
C           WRITE(LIN1,599) (SIG(J),J=1,6)
CC
CC  WRITE STRESSES AT ELEMENT CENTRE TO LIN4
CC
            WRITE(LIN4,*) XTX, YTY, (SIG(J),J=4,6)
        ENDIF
  140   CONTINUE
  180   CONTINUE
CC
CC  ---  AVERAGE NODAL STRESSES
CC
        DO 190 I=1,NNODE
```

```
      IF(PX(I).EQ.0.) GOTO 190
      DO 185 J=1,3
      SPX(I,J)=SPX(I,J)/PX(I)
  185 CONTINUE
      S1=(SPX(I,1)+SPX(I,2))/2.
      S2=(SPX(I,1)-SPX(I,2))/2.
      IF(S2.EQ.0.) GOTO 190
      S3=SPX(I,3)
      SPX(I,6)=ATAN2(S3,S2)*90.0/PI
      S2=SQRT(S2**2+S3**2)
      SS1=S1+S2
      SS2=S1-S2
      SPX(I,4) = SS1
      SPX(I,5) = SS2
C
  190 CONTINUE
      WRITE(LIN1,192)
      WRITE(LIN1,191)
      DO 193 I=1,NNODE
      WRITE(LIN1,205) I,(SPX(I,J),J=1,6)
  193 CONTINUE
CC
CC  OUTPUT STRESSES AT ELEMENT CENTRES
CC
      REWIND LIN4
      WRITE(LIN1,300)
      WRITE(LIN1,640)
      DO 250 I=1,NNEL
      READ (LIN4,*) XTX,YTY, (SIG(J),J=4,6)
      WRITE(LIN1,660) I,XTX, YTY, (SIG(J),J=4,6)
  250 CONTINUE
  192 FORMAT(' ',20X,/,'  AVERAGE NODAL STRESSES',)
  191 FORMAT(/,'    NODE  SIGMA-X   SIGMA-Y    TAU-XY   ',
     1 'SIGMA-1   SIGMA-2   X-ANGLE',/)
  300 FORMAT(' ',20X,/,' STRESSES AT ELEMENT CENTRES',)
  591 FORMAT(' ','NODE',I3,5F10.4,F10.2)
  599 FORMAT(' ','CENTRE ',5F10.4,F10.2)
  205 FORMAT(' ',I6,5F10.4,F10.2,F10.2)
  210 FORMAT(1X,I4,2(6X,F10.4))
  260 FORMAT(' ',/,20X,' ELEMENT NODAL STRESSES')
  280 FORMAT(' ',/,20X,' ELEMENT NO. ',I2,/)
```

```
630   FORMAT(/,'  NODE   X-COORD.  Y-COORD.   X-DISPL.    Y-DISPL.',/)
640   FORMAT(/,'  ELEM   X-CENTR.  Y-CENTR.   SIGMA-1    ',
     1          'SIGMA-2   X-ANGLE',/)
650   FORMAT(' ',I4,2F10.2,2F12.6)
660   FORMAT(' ',I4,2F10.2,2X,2F10.4,F10.2)
      RETURN
      END
```

Appendix B

LISTING OF A GENERAL PURPOSE 8-NODE QUADRILATERAL MESH GENERATOR, QUADMESH

```
CC
The listing of a general purpose 8-node quadrilateral mesh
generator called QUADMESH, is given in this appendix. The program
has been described in Chapter 5.
CC
CC  A GENERAL PURPOSE 8-NODE QUADRILATERAL MESH GENERATOR
CC
CC
      INTEGER    NLM(5000),JJX(2000),JDSIDE(500)
      INTEGER    MTYPE(20),NODRT(50),NODFR(50),NOMID(50),ND(50)
      INTEGER    DLOAD(500),NPO(500),NSP(100),KSP(100),NOTEM(50)
      INTEGER    NELTYP(1000),JDOX(2000),JDOY(2000),LXY(8)
      INTEGER    NUMEL,NUMAT,NDIST,NUM,NNODE,LIN,LNP
      REAL       XX(2000),YY(2000),XV(100),YV(100)
      REAL       XPO(500),YPO(500)
      REAL       DELX(100),DELY(100),XF(50),YF(50),XR(50)
      REAL       YR(50),XT(50),YT(50),XZ(4),YZ(4)
      REAL       ZXL(20),ZXR(20),ZYB(20),ZYT(20),YM(10),PR(10)
      REAL       YDLL,YDLR,XDLT,XDLB,XR,YT
      CHARACTER*1   TITLE(70)
CC
         LIN = 11
         LNP = 12
         OPEN(LIN,FILE='meshin',STATUS='OLD',ACCESS='SEQUENTIAL')
         OPEN(LNP,FILE='meshout',STATUS='OLD',FORM='FORMATTED')
CC ------------------------------------------------------------
CC
CC  THIS PROGRAM WILL GENERATE AN 8-NODE QUADRILATERAL MESH OVER
CC  AN ARRAY OF VERTICAL LINES WHOSE LOWER END COORDINATES ARE
CC  SPECIFIED. THE LIMITS OF THE PROGRAM ARE:
CC
CC       MAXIMUM NO. OF NODES                        = 1000
CC       MAXIMUM NO. OF ELEMENTS                     = 625
CC       MAXIMUM NO. OF SIDES WITH DISTRIB. LOADS = 125
```

```
CC       MAXIMUM NO. OF NODES WITH POINT LOADS    = 500
CC       MAXIMUM NO. OF ELEMENT PROPERTY ZONES    = 20
CC       MAXIMUM NO. OF DIFFERENT PROPERTIES      = 10
CC
CC  IF THESE MAXIMUM ARE EXCEEDED, PROGRAM WILL ABORT.
CC
CC  THE INPUT DATA FOR EACH PROBLEM IS AS FOLLOWS:
CC
CC   1. LINE 1 : XO, YO, NSUBX, NSUBY
CC
CC   2. NEXT LINE(S),
CC      IF NSUBX IS > 0, : DELX(1), DELX(2), ....., DELX(NSUBX)
CC
CC   3. NEXT LINE(S),
CC      IF NSUBY IS > 0, : DELY(1), DELY(2), ....., DELY(NSUBY)
CC
CC   THE ABOVE THREE STEPS ARE REPEATED FOR ALL VERTICAL LINES
CC   OR SETS OF LINES.
CC
CC   4. NEXT LINE         : 0.0  0.0  0  0
CC      THIS ENDS THE VERTICAL LINE INFORMATION.
CC
CC   5. NEXT LINE: NPOIN,NDIST,NUMAT,NTYPE,YDLL,YDLR,XDLT,XDLB
CC
CC   6. NEXT NPOIN LINES   : NPO(I),XPO(I),YPO(I)
CC
CC   7.  FOR EACH MATERIAL TYPE (I = 1, ...,NUMAT),
CC        FIRST LINE  :     YM(I),PR(I),MTYPE(I)
CC        SECOND LINE :     ZXL(I),ZXR(I),ZYB(I),ZYT(I)
CC
CC       IF THERE IS ONLY ONE MATERIAL TYPE,  THEN,
CC        SECOND LINE : 0.0  0.0  0.0  0.0
CC
CC   8. DISPLACEMENT BOUNDARY CONDITION CODES
CC      :JXL, JYL, JXB, JYB, JXR, JYR, JXT, JYT
CC
CC  THIS IS THE END OF THE INPUT DATA
CC ------------------------------------------------------------
CC  THE FOLLOWING DEFINITIONS ARE PERTINENT:
CC
CC  XO       = X-COORD. OF VERTICAL LINE
```

```
CC  YO        = Y-COORD. OF BOTTOM OF VERTICAL LINE
CC  NSUBX     = NO. OF SUBDIVS. ALONG X-DIRECTION
CC            = 0, THERE IS ONLY ONE VERTICAL LINE
CC  NSUBY     = NO. OF SUBDIVS. ALONG Y-DIRECTION
CC          = 0, THERE IS ONLY ONE POINT TO CONSIDER (= TRIVIAL)
CC  DELX*   = LENGTH OF SUBDIVS. IN X-DIRECTION,
CC            STARTING FROM LEFT
CC  DELY*   = LENGTH OF SUBDIVS. IN Y-DIRECTION,
CC            STARTING FROM BOTTOM
CC
CC  XR*, YR* = ARRAY OF NODAL COORDINATES OF RIGHT EDGE NODES
CC  XF*, YF* = ARRAY OF NODAL COORDINATES OF FRONT EDGE NODES
CC  XZ*, YZ* = ARRAY OF CORNER NODE COORDINATES
CC               OF CURRENT ELEMENT
CC  NRY       = NO. OF CORNER NODES ON RIGHT EDGE
CC  NF        = NO. OF CORNER NODES ON FRONT EDGE
CC  NVERT     = VERTICAL LINE COUNTER
CC  ICOL      = COLUMN COUNTER
CC  NODRT*    = ARRAY OF NODE NUMBERS OF RIGHT EDGE CORNER NODES
CC  NODFR*    = ARRAY OF NODE NUMBERS OF FRONT EDGE CORNER NODES
CC  NOMID*    = ARRAY OF NODE NUMBERS OF MIDSIDE NODES CONNECTING
CC               FRONT AND RIGHT EDGES
CC  NOTEM*    = ARRAY OF NODE NUMBERS EQUIVALENT TO NOMID*
CC  LASTNODE = LAST NODE COUNTER
CC
CC   DISTRIBUTED LOAD INFO.
CC
CC  THE LEFT AND BOTTOM EDGES ARE ASSUMED TO BE AXES OF SYMMETRY
CC  SO THAT DISTRIBUTED LOADS CAN ONLY BE APPLIED ALONG THE TOP
CC  AND RIGHT EDGES. DISTRIBUTED LOADS ALONG THE RIGHT EDGE
CC  ARE ASSUMED TO VARY LINEARLY DOWNWARDS WHILE THOSE ALONG THE
CC  TOP ARE ASSUMED TO BE CONSTANT.
CC
CC     XDLT    = XLOAD AT TOP RIGHT HAND CORNER OF STRUCTURE,
CC     XDLB    = XLOAD AT BOTTOM RIGHT HAND CORNER OF STRUCTURE,
CC     YDLL    = YLOAD AT TOP LEFT HAND CORNER OF STRUCTURE,
CC     YDLR    = YLOAD AT TOP RIGHT HAND CORNER OF STRUCTURE,
CC    NDIST = TOTAL NO. OF ELEMENT SIDES WITH DISTRIBUTED LOADS.
CC    ZXL(I)   = X-CORD OF LEFT BOUNDARY OF ITH ZONE
CC    ZXR(I)   = X-CORD OF RIGHT BOUNDARY OF ITH ZONE
CC    ZYB(I)   = Y-CORD OF BOTTOM BOUNDARY OF ITH ZONE
```

```
CC    ZYT(I)    = X-CORD OF TOP BOUNDARY OF ITH ZONE
CC    MTYPE(I)  = MATERIAL TYPE OF ITH ZONE
CC
CC  POINT LOAD INFO.
CC
CC    NPOIN     = TOTAL NO. OF NODES WITH POINT LOAD.
CC    NPO(I)    = ITH NODE POINT WITH POINT LOAD
CC    XPO(I)    = X-LOAD ON NPO(I)
CC    YPO(I)    = Y-LOAD ON NPO(I)
CC
CC  MATERIAL PROPERTY INFO.
CC
CC    NUMAT     = NO. OF DISTINCT ZONES OF THE DISCRETIZED
CC                DOMAIN WITH DIFF. MATERIAL PROPERTIES
CC              = TOTAL NO. OF DIFFERENT MATERIALS
CC    NTYPE     = PROBLEM TYPE,
CC              = 1 => PLANE STRAIN
CC              = 2 => PLANE STRESS
CC    YM(I)     = YOUNG'S MODULUS OF ITH MATERIAL TYPE
CC    PR(I)     = POISSON'S RATIO OF ITH MATERIAL TYPE
CC
CC  DISPLACEMENT BOUNDARY CONDITION CODES
CC    JXL = X-CONDTION CODE FOR LEFT BOUNDARY
CC    JYL = Y-CONDTION CODE FOR LEFT BOUNDARY
CC    JXB = X-CONDTION CODE FOR LOWER BOUNDARY
CC    JYB = Y-CONDTION CODE FOR LOWER BOUNDARY
CC    JXR = X-CONDTION CODE FOR RIGHT BOUNDARY
CC    JYR = Y-CONDTION CODE FOR RIGHT BOUNDARY
CC    JXT = X-CONDTION CODE FOR TOP BOUNDARY
CC    JYT = Y-CONDTION CODE FOR TOP BOUNDARY
CC     VALUE = 1 => NO MOVEMENT ALLOWED IN THE GIVEN DIRECTION,
CC     VALUE = 0 => MOVEMENT IS ALLOWED IN THE GIVEN DIRECTION
CC
CC  INITIALIZATION
CC
        XMAX = 0.0
        XMIN = 1.0E10
        YMAX = 0.0
        YMIN = 1.0E10
        NNODE = 0
        NUM = 0
```

```
      NUMEL = 0
      JSP = 0
      LASTNODE = 0
      ICOL = 0
      JNS = 0
      NVERT = 0
CC
     DO 5 I = 1,100
        NSP(I) = 0
          KSP(I) = 0
          DELX(I) = 0.0
        DELY(I) = 0.0
 5    CONTINUE
      DO 10 I = 1,2000
 10   JJX(I) = 0
CC
      READ(LIN,15) TITLE
 15   FORMAT(80A1)
 20   READ(LIN,*) X0,Y0,NSUBX, NSUBY
      IF(NSUBX.EQ.0.AND.NSUBY.EQ.0) GOTO 500
      IF(NSUBX.NE.0) THEN
      READ(LIN,*) (DELX(I),I=1,NSUBX)
      ENDIF
      IF(NSUBY.NE.0) THEN
         READ(LIN,*) (DELY(I),I=1,NSUBY)
      ENDIF
CC
      DXX = X0
      INX = 0
 25   INX = INX+1
      IF(INX.GT.NSUBX+1) GOTO 20
      IF(INX.GT.1) DXX = DXX + DELX(INX-1)
      NRY = NSUBY+1
      ICOL = NVERT
      NVERT = NVERT+1
      XR(1) = DXX
      YR(1) = Y0
      DO 35 I = 1,NSUBY
        XR(I+1) = DXX
         YR(I+1) = YR(I) + DELY(I)
 35   CONTINUE
```

```
        IF(NVERT.EQ.1) THEN
          NF = NRY
            DO 30 I = 1,NF
             XF(I) = XR(I)
             YF(I) = YR(I)
 30         CONTINUE
          GOTO 25
        ENDIF
CC
CC  LOCATE FIRST NODE, (NF1), ON FRONT EDGE WHICH IS NOT ABOVE
CC  FIRST NODE, (NR1), ON RIGHT EDGE AND FIRST NODE, (NF2), ON
CC  FRONT EDGE WHICH IS NOT BELOW LAST NODE, (NR),
CC  ON RIGHT EDGE.
CC
        J = 1+NF
        NF0 = 0
        NF2 = J
        DO 40 I = 1,NF
          J = J-1
            IF(YF(I).LT.YR(1)) NF0 = NF0+1
          IF(YF(J).GE.YR(NRY)) NF2=NF2-1
 40     CONTINUE
        NF1 = NF0
        IF(YF(NF0+1).LE.YR(1)) NF1 = NF0+1
CC
CC  DETERMINE NO. OF EXTRA EDGES GENERATED BY THE FRONT TURNING
CC  THROUGH 90 DEGREES
CC
        NM = NRY
        IF(ICOL.EQ.1) GOTO 45
          DO 50 I = NF1,NF
           IF(YF(I).EQ.YF(I+1)) THEN
          IF(YF(I).GE.YR(1).AND. YF(I).LE.YR(NRY)) NM = NM+1
        ENDIF
 50     CONTINUE
CC
CC  MAP NODFR* INTO GLOBAL NODE NUMBERS
CC
 45     CONTINUE
        IF(ICOL.EQ.1) THEN
          NP = 2
```

```
          CALL MAP(NODFR,NF,LASTNODE,NP)
        ENDIF
CC
CC  MAP NOMID* INTO GLOBAL NODE NUMBERS
CC
          NP = 1
          CALL MAP(NOMID,NM,LASTNODE,NP)
CC
        IF(NF0.GT.0) THEN
          JSP = JSP+1
          NSP(JSP) = NOMID(1)
        ENDIF
        IF(NF2.LT.NF) THEN
           JSP = JSP+1
           NSP(JSP) = NOMID(NSUBY+1)
        ENDIF
CC
CC  MAP NODRT* INTO GLOBAL NODE NUMBERS
CC
          NP = 2
          CALL MAP(NODRT,NRY,LASTNODE,NP)
CC
CC  BUILD 8-NODE QUADRILATERAL ELEMENTS IN COLUMN
CC  SORT NSP ARRAY IN ASCENDING ORDER W.R.T Y-COORD.
CC
      DO 79 I = 1,MSP-1
      II = KSP(I)
      DO 77 J = I+1,MSP
      JJ = KSP(J)
      IF(YY(II).GT.YY(JJ)) THEN
        KSP(I) = JJ
        KSP(J) = II
        II = JJ
      ENDIF
 77      CONTINUE
 79      CONTINUE
CC
          JR = 1
          JM = 1
          JF = NF1
          ISP = 0
```

```
          DO 90 II = 1,NM-1
          JF1 = JF+1
          JF2 = JF+2
          NUM = NUM+1
CC
          IF(JF2.GT.NF) JF2=NF
            LXY(1) = NODFR(JF)
             LXY(2) = NOMID(JM)
              LXY(3) = NODRT(JR)
             LXY(7) = NODFR(JF1)
            LXY(8) = NODFR(JF)+1
            XZ(1) = XF(JF)
             YZ(1) = YF(JF)
              XZ(2) = XR(JR)
              YZ(2) = YR(JR)
             XZ(4) = XF(JF1)
            YZ(4) = YF(JF1)
CC
          IF(JF1.EQ.JF2) GOTO 600
          IF(YF(JF1)-YF(JF)) 400,300,200
 200      IF(YF(JF2)-YF(JF1)) 290,250,220
 220      IF(XF(JF1)-XF(JF)) 620,600,620
CC
CC  CASES 1 AND 4
CC
 600        LXY(4) = NODRT(JR)+1
               LXY(5) = NODRT(JR+1)
                 LXY(6) = NOMID(JM+1)
                  XZ(3) = XR(JR+1)
                 YZ(3) = YR(JR+1)
               GOTO 60
CC
 250      IF(XF(JF2)-XF(JF1)) 600,280,270
CC
CC  CASE 3
CC
 270         LXY(4) = NOMID(JM+1)
               LXY(5) = NODFR(JF2)
                ISP = ISP+1
            LXY(6) = KSP(ISP)
                   XZ(3) = XF(JF2)
```

```
                YZ(3) = YF(JF2)
            JF = JF+1
            JR = JR-1
                GOTO 60
CC
CC  CASES 5 AND 6
CC
 620          LXY(4) = NODRT(JR)+1
               LXY(5) = NODRT(JR+1)
              LXY(6) = NOMID(JM+1)
               ISP = ISP+1
                 LXY(8) = KSP(ISP)
                  XZ(3) = XR(JR+1)
                 YZ(3) = YR(JR+1)
              GOTO 60
CC
CC  CASE 2
CC
 300          LXY(4) = NOMID(JM+1)
               LXY(5) = NODFR(JF2)
              LXY(6) = NODFR(JF1)+1
               ISP = ISP+1
                 LXY(8) = KSP(ISP)
                  XZ(3) = XF(JF2)
                 YZ(3) = YF(JF2)
            JF = JF+1
            JR = JR-1
                GOTO 60
 280        WRITE(*,*) 'ERROR DETECTED AT STATEMENTS 250 - 280'
            STOP
 290        WRITE(*,*) 'ERROR DETECTED AT STATEMENTS 200 - 290'
            STOP
 400        WRITE(*,*) 'ERROR DETECTED AT STATEMENT 400 '
            STOP
CC
 60        CONTINUE
           DO 70 I = 1,8
           NUMEL = NUMEL+1
           NLM(NUMEL) = LXY(I)
 70        CONTINUE
CC
```

```
CC  NODAL COORDS. OF CORNER NODES
CC
          JJ = 0
          IA = 8*(NUM-1)
          DO 75 I = 1,8,2
          LL = NLM(IA+I)
          JJ = JJ+1
          IF(JJX(LL).EQ.1) GOTO 75
          NNODE = NNODE+1
          XX(LL) = XZ(JJ)
          YY(LL) = YZ(JJ)
          JJX(LL) = 1
          IF(XMIN.GT.XX(LL)) XMIN = XX(LL)
          IF(XMAX.LT.XX(LL)) XMAX = XX(LL)
          IF(YMIN.GT.YY(LL)) YMIN = YY(LL)
          IF(YMAX.LT.YY(LL)) YMAX = YY(LL)
 75       CONTINUE
CC
CC  NODAL COORDS. OF MIDSIDE NODES
CC
          DO 80 I = 1,8,2
          J = I+2
          IF(J.GT.8) J=J-8
          LL = NLM(IA+I+1)
          KK = NLM(IA+I)
          JJ = NLM(IA+J)
          IF(JJX(LL).EQ.1) GOTO 80
          NNODE = NNODE+1
          XX(LL) = (XX(KK)+XX(JJ))/2.0
          YY(LL) = (YY(KK)+YY(JJ))/2.0
          JJX(LL) = 1
 80       CONTINUE
CC
          JF = JF+1
          JR = JR+1
          JM = JM+1
CC
 90       CONTINUE
          MSP = 0
CC
           DO 46 I = 1,NM
```

```
 46       NOTEM(I) = NOMID(I)
CC
CC  CHECK WHICH OF NSP(*) ELEMENTS LIE ON THE FRONT
CC
         DO 47 II = 1,JSP
         I = NSP(II)
         XV(II) = XX(I)
         YV(II) = YY(I)
 47      CONTINUE
         CALL CHECK(NSP,JSP,KSP,MSP,XX,YY,NODFR,
     1                          NF,XV,YV,LASTNODE)
CC
CC  UPDATE THE FRONT EDGE
CC
         JJ = 0
         IF(NF2.EQ.NF) GOTO 105
         DO 100 I = NF2,NF
         JJ = JJ+1
         XT(JJ) = XF(I)
         YT(JJ) = YF(I)
         ND(JJ) = NODFR(I)
 100     CONTINUE
CC
 105     NF = 0
         IF(NF0.EQ.0) GOTO 109
         DO 107 I = 1,NF1
         NF = NF+1
         XF(NF) = XF(I)
         YF(NF) = YF(I)
         NODFR(NF) = NODFR(I)
 107     CONTINUE
CC
 109     DO 110 I = 1,NRY
         NF = NF+1
         XF(NF) = XR(I)
         YF(NF) = YR(I)
         NODFR(NF) = NODRT(I)
 110     CONTINUE
CC
         IF(JJ.EQ.0) GOTO 125
         DO 120 I = 1,JJ
```

```
          NF = NF+1
          XF(NF) = XT(I)
          YF(NF) = YT(I)
          NODFR(NF) = ND(I)
 120      CONTINUE
 125      CONTINUE
CC
CC  CHECK WHICH OF NSP(*) ELEMENTS LIE ON THE UPDATED FRONT
CC
       CALL CHECK(NSP,JSP,KSP,MSP,XX,YY,NODFR,NF,XV,YV,LASTNODE)
CC
          GOTO 25
 500      CONTINUE
       WRITE(*,*) 'MAXIMUM X-VALUE =', XMAX
       WRITE(*,*) 'MINIMUM X-VALUE =', XMIN
       WRITE(*,*) 'MAXIMUM Y-VALUE =', YMAX
       WRITE(*,*) 'MINIMUM Y-VALUE =', YMIN
CC
CC  NODAL CONSTRAINTS, ELEMENT MATERIAL PROPERTIES
CC  AND DISTRIBUTED LOADS
CCO
          READ (LIN,*) NPOIN,NDIST,NUMAT,NTYPE,YDLL,YDLR,XDLT,XDLB
          NDISZ = 4*NDIST
          IF(NPOIN.GT.0) THEN
            DO 510 I = 1,NPOIN
 510        READ(LIN,*) NPO(I),XPO(I),YPO(I)
          ENDIF
          IF(NUMAT-1) 520,530,530
 520      WRITE(*,*) 'ERROR! NEGATIVE VALUE READ FOR NO.',
     1               'OF MATERIAL TYPES', NUMAT
          STOP
 530      DO 550 I = 1,NUMAT
          READ (LIN,*) YM(I),PR(I),MTYPE(I)
 550      READ (LIN,*) ZXL(I),ZXR(I),ZYB(I),ZYT(I)
          IF(NUMAT.EQ.1) THEN
         ZXL(1) = XMIN
         ZXL(1) = XMAX
         ZXL(1) = YMIN
         ZXL(1) = YMAX
         ENDIF
         READ (LIN,*) JXL, JYL, JXB, JYB, JXR, JYR, JXT, JYT
```

```
CC
      CALL MATEL(XX,YY,JDOX,JDOY,NNODE,NUM,NUMEL,
     .     NLM,NELTYP,JDSIDE,DLOAD,MTYPE,ZXL,NDISZ,
     .     ZXR,ZYB,ZYT,YDLL,YDLR,XDLT,XDLB,XMAX,XMIN,YMAX,
     .     JXL, JYL, JXB, JYB, JXR, JYR, JXT, JYT,
     .     YMIN,LNP,NUMAT,NTYPE,YM,PR,NPOIN,NPO,XPO,YPO,TITLE)
CC
      STOP
      END
CC-----------------------------------------------
      SUBROUTINE MAP(NCELL,NN,LL,IJ)
      DIMENSION NCELL(NN)
CC
      LL = LL-IJ+1
      DO 50 I = 1,NN
      LL = LL+IJ
      NCELL(I) = LL
 50   CONTINUE
      RETURN
      END
CC ---------------------------------------------
      SUBROUTINE CHECK(NSP,JSP,KSP,MSP,XF,YF,NOD,NF,XV,YV,LL)
      DIMENSION  NSP(JSP),KSP(JSP),XF(LL),YF(LL),NOD(NF)
      DIMENSION     XV(JSP),YV(JSP)
CC
      DO 86 II = 1,JSP
      XI = XV(II)
      YI = YV(II)
      DO 84 JJ = 1,NF-1
      J = NOD(JJ)
      K = NOD(JJ+1)
      XJ = (XF(J)+XF(K))/2.0
      YJ = (YF(J)+YF(K))/2.0
      IF(XI.EQ.XJ.AND.YI.EQ.YJ) THEN
      MSP = MSP+1
      KSP(MSP) = NSP(II)
      GOTO 86
      ENDIF
 84   CONTINUE
CC
 86   CONTINUE
```

```
CC
CC  CHECK FOR REPEATED NODES
CC
        K = MSP
        DO 150 II = 1,MSP-1
        I = KSP(II)
        DO 140 JJ = II+1,MSP
        J = KSP(JJ)
        IF(I.EQ.J) THEN
        K = JJ-1
        IF(JJ.EQ.MSP) GOTO 140
        DO 130 KK = JJ+1,MSP
        K = K+1
 130    KSP(K) = KSP(KK)
        ENDIF
 140    CONTINUE
        MSP = K
 150    CONTINUE
CC
        RETURN
        END
CC ----------------------------------------------
        SUBROUTINE MATEL(XORD,YORD,JDOX,JDOY,NNODE,NNEL,NUMEL,
     +       NLM,NELTYP,JDSIDE,DLOAD,MTYPE,ZXL,NDISZ,
     +       ZXR,ZYB,ZYT,YDLL,YDLR,XDLT,XDLB,XMAX,XMIN,YMAX,
     +       JXL, JYL, JXB, JYB, JXR, JYR, JXT, JYT,
     +       YMIN,LNP,NUMAT,NTYPE,YM,PR,NPOIN,NPO,XPO,YPO,TITLE)
CC
        DIMENSION  XORD(NNODE), YORD(NNODE),DLOAD(NDISZ)
        DIMENSION  NLM(NUMEL), NELTYP(NNEL),JDOX(NNODE)
        DIMENSION  YM(NUMAT), PR(NUMAT),JDOY(NNODE)
        DIMENSION  NPO(NPOIN),XPO(NPOIN), YPO(NPOIN)
        INTEGER    JDSIDE(NDISZ),MTYPE(NUMAT),NODE(4)
        INTEGER    NNEL,NUMAT,NNODE,ML,ML,MT,IN
        REAL       ZXL(NUMAT),ZXR(NUMAT),ZYB(NUMAT),ZYT(NUMAT)
        REAL       XL,XR,YB,YT,XXL,XXR,YYB,YYT
        CHARACTER*1   TITLE(70)
CC ------------------------------------------------------
CC     YB       = Y-CORD OF BOTTOM OF STRUCTURE,
CC     YT       = Y-CORD OF TOP OF STRUCTURE,
CC     YBE      = BOTTOM Y-COORD. OF ELEMENT
```

```
CC                (ASSUMED PARALLEL TO X-AXIS)
CC     YTE      = TOP Y-COORD. OF ELEMENT
CC                (ASSUMED PARALLEL TO X-AXIS)
CC     YDLL     = YLOAD AT TOP LEFT CORNER OF ELEMENT,
CC     YDLR     = YLOAD AT TOP RIGHT CORNER OF ELEMENT,
CC     XDLT     = XLOAD AT TOP RIGHT CORNER OF ELEMENT,
CC     XDLB     = XLOAD AT BOTTOM RIGHT CORNER OF ELEMENT,
CC ------------------------------------------------
        NDIST = NDISZ/4
        THIC = 1.0
        NPOIN = 0
        DENS = -0.0264
        IPRNT = 1
CC
CC  NODAL CONSTRAINTS
CC
        DO 10 I = 1,NNODE
        JDOX(I) = 0
        JDOY(I) = 0
CC
CC CHECK IF NODE IS ON LEFT BOUNDARY
CC
        IF(XORD(I).EQ.XMIN) THEN
          JDOX(I) = JXL
          JDOY(I) = JYL
        ENDIF
CC
CC CHECK IF NODE IS ON LOWER BOUNDARY
CC
        IF(YORD(I).EQ.YMIN) THEN
          JDOX(I) = JXB
          JDOY(I) = JYB
        ENDIF
CC
CC CHECK IF NODE IS ON RIGHT BOUNDARY
CC
        IF(XORD(I).EQ.XMAX) THEN
          JDOX(I) = JXR
          JDOY(I) = JYR
        ENDIF
CC
```

```
CC CHECK IF NODE IS ON TOP BOUNDARY
CC
      IF(YORD(I).EQ.YMAX) THEN
        JDOX(I) = JXT
        JDOY(I) = JYT
      ENDIF
CC
CC  CHECK FOR CORNER NODES
CC
      IF(XORD(I).EQ.XMIN .AND. YORD(I).EQ.YMIN) THEN
        IF(JXL.EQ.1 .OR.JXB.EQ.1) JDOX(I) = 1
        IF(JYL.EQ.1 .OR.JYB.EQ.1) JDOY(I) = 1
      ENDIF
CC
      IF(XORD(I).EQ.XMIN .AND. YORD(I).EQ.YMAX) THEN
        IF(JXL.EQ.1 .OR.JXT.EQ.1) JDOX(I) = 1
        IF(JYL.EQ.1 .OR.JYT.EQ.1) JDOY(I) = 1
      ENDIF
CC
      IF(XORD(I).EQ.XMAX .AND. YORD(I).EQ.YMAX) THEN
        IF(JXR.EQ.1 .OR.JXT.EQ.1) JDOX(I) = 1
        IF(JYR.EQ.1 .OR.JYT.EQ.1) JDOY(I) = 1
      ENDIF
CC
      IF(XORD(I).EQ.XMAX .AND. YORD(I).EQ.YMIN) THEN
        IF(JXB.EQ.1 .OR.JXR.EQ.1) JDOX(I) = 1
        IF(JYB.EQ.1 .OR.JYR.EQ.1) JDOY(I) = 1
      ENDIF
CC
 10     CONTINUE
CC
CC  ELEMENT MATERIAL PROPERTIES
CC
       IF(NUMAT.EQ.1) THEN
         DO 20 II = 1,NNEL
 20      NELTYP(II) = 1
         GOTO 45
       ENDIF
        DO 40 II = 1,NNEL
        IA = 8*(II-1)
        ML = NLM(IA+1)
```

```
      MR = NLM(IA+3)
      MT = NLM(IA+5)
      XXL = XORD(ML)
      XXR = XORD(MR)
      YYB = YORD(ML)
      YYT = YORD(MT)
      DO 30 JJ = 1,NUMAT
      XL = ZXL(JJ)
      XR = ZXR(JJ)
      YB = ZYB(JJ)
      YT = ZYT(JJ)
      IF(XXL.GE.XL.AND.XXR.LE.XR) THEN
        IF(YYB.GE.YB.AND.YYT.LE.YT) THEN
            NELTYP(II) = MTYPE(JJ)
            GOTO 40
        ENDIF
      ENDIF
 30   CONTINUE
 40   CONTINUE
 45   CONTINUE
      WRITE(LNP,505) TITLE
      WRITE(LNP,500) NNODE,NNEL,NUMAT,NTYPE,NPOIN,NDIST,
     1 IPRNT,THIC
      DO 50 I = 1,NUMAT
 50   WRITE(LNP,510) YM(I), PR(I), DENS
      DO 55 I = 1,NNODE
      WRITE(LNP,80) I,XORD(I),YORD(I),JDOX(I),JDOY(I)
 55   CONTINUE
CC
      DO 60 I = 1,NNEL
      IA = 8*I-7
      IB = IA+7
      WRITE(LNP,70) I,NELTYP(I),(NLM(J),J=IA,IB)
 60   CONTINUE
CC
 70   FORMAT(10I6)
 80   FORMAT(I6,2F8.3,2I5)
CC
CC  DISTRIBUTED LOADS, ASSUMED TO BE APPLIED TO THE
CC  TOP AND/OR RIGHT BOUNDARIES ONLY.
CC
```

```
      GRADX = (YDLL-YDLR)/(XMAX-XMIN)
      GRADY = (XDLT-XDLB)/(YMAX-YMIN)
      DO 90 I = NDISZ
 90   DLOAD(I) = 0.0
CC
      IN = 0
      DXB = XDLB
      DYL = YDLL
      DO 160 II = 1,NNEL
      IA = 8*(II-1)
      DO 150 JJ = 1,2
      IJ = JJ*2
      DO 100 KK = 1,3
      IJ = IJ+1
 100  NODE(KK) = NLM(IA+IJ)
      GOTO(110,130), JJ
 110  CONTINUE
CC
CC  RIGHT BOUNDARY LOADS
CC
     IF(XDLB.EQ.0.0 .AND. XDLT.EQ.0.0) GOTO 150
      XN = XORD(NODE(3))
      IF(XN.EQ.XMAX) THEN
        IN = IN+1
        IB = 4*IN-4
        DO 120 I = 1,3
 120     JDSIDE(IB+I) = NODE(I)
         JDSIDE(IB+4) = II
          YBB = YORD(NODE(1))
         YBT = YORD(NODE(3))
        DXT = GRADY*(YBT - YBB) + DXB
       DLOAD(IB+1) = DXB
      DLOAD(IB+3) = DXT
      DXB = DXT
     ENDIF
      GOTO 150
 130  CONTINUE
     IF(YDLL.EQ.0.0 .AND. YDLR.EQ.0.0) GOTO 160
CC
CC  TOP BOUNDARY LOADS
CC
```

```
      YN = YORD(NODE(3))
      IF(YN.EQ.YMAX) THEN
        IN = IN+1
         IB = 4*IN-4
          DO 140 I = 1,3
140         JDSIDE(IB+I) = NODE(I)
             JDSIDE(IB+4) = II
              XBR = XORD(NODE(1))
             XBL = XORD(NODE(3))
            DYR = GRADY*(XBR - XBL) + DYL
           DLOAD(IB+2) = DYR
          DLOAD(IB+4) = DYL
         DYL = DYR
       ENDIF
CC
 150    CONTINUE
 160    CONTINUE
CC
       IF(NDIST.GT.0) THEN
         DO 170 I = 1,NDIST
          IA = 4*I-3
          IB = IA+3
 170      WRITE(LNP,180) (JDSIDE(J),J=IA,IB),(DLOAD(J),J=IA,IB)
       ENDIF
CC
       IF(NPOIN.GT.0) THEN
          DO 175 I = 1,NPOIN
 175       WRITE(LNP,185) NPO(I),XPO(I),YPO(I)
       ENDIF
CC
 180   FORMAT(4I5, 4F10.3)
 185   FORMAT(I5, 2F10.3)
 505   FORMAT(70A1)
 510   FORMAT(1F12.3,2F10.3)
 500   FORMAT(7I5,F6.2)
CC
       STOP
       END
```

Appendix C

LISTING A SPECIAL PURPOSE 8-NODE QUADRILATERAL MESH GENERATOR OVER CIRCULAR OPENINGS, QUADMESH

The listing of a special purpose 8-node quadrilateral mesh generator called CIRCMESH, is given in this appendix. The program has been described in Chapter 5.

```
CC*********************************************************
CC  INPUT IS AS FOLLOWS:
CC
CC  LINE 1:     READ(LIN,1) TITLE
CC  LINE 2:     READ(LIN,*) NNY,NANG,RO,BM
CC  LINE 3:     READ(LIN,*) ((JXL(I,J),J=1,2),I=1,4)
CC  LINE 4:     READ(LIN,*) XL,YB,XR,YT,TH
CC  LINE 5:     READ(LIN,*) ASXX, ASYY
CC
CC  EXPLANATIONS:
CC
CC  TITLE = TITLE, NOT MORE THAN 75 CHARACTERS.
CC
CC   NNY  = NO. OF ELEMENT SIDES ALONG X OR Y AXIS
CC   RO   = BOREHOLE RADIUS
CC   BM   = Y (OR X) BOUNDARY MAGNIFICATION FACTOR
CC   NANG = NO. OF ANGULAR SUBDIVS. OF FIRST QUADRANT
CC          - (MUST BE EVEN)
CC
CC       JXL(I,J) = CONDITION CODE FOR BOUNDARY I IN J DIRECTION
CC              J = 1, MEANS X-DIRECTION
CC              J = 2, MEANS Y-DIRECTION
CC       JXL(I,J) = 0, MEANS BOUNDARY CAN MOVE IN J DIRECTION
CC                 = 1, MEANS BOUNDARY CANNOT MOVE IN J DIRECTION
CC  I=1,  BOUNDARY # 1 : ( XL = LEFT BOUNDARY X-COORDINATE)
CC  I=2,  BOUNDARY # 2 : ( YB = BOTTOM BOUNDARY Y-COORDINATE)
CC  I=3,  BOUNDARY # 3 : ( XR = RIGHT BOUNDARY X-COORDINATE)
CC  I=4,  BOUNDARY # 4 : ( YT = TOP BOUNDARY Y-COORDINATE)
CC
```

```
CC    ASXX = APPLIED STRESS IN X-DIRECTION
CC    ASYY = APPLIED STRESS IN Y-DIRECTION
CC
CC   OTHER EXPLANATIONS ARE:
CC
CC   IPRNT     = PRINT OUTPUT DATA OPTION
CC             = 0  => DO NOT PRINT
CC             = 1  => PRINT
CC
CC   NUMAT         = TOTAL NO. OF MATERIAL TYPES
CC   NTYPE         = PROBLEM TYPE PARAMETER
CC                 = 1 => PLANE STRESS
CC                 = 2 => PLANE STRAIN
CC   NDIST     = NO. OF ELEMENT SIDES WITH DISTRIBUTED LOADS
CC   NPOIN     = NO. OF NODES WITH POINT LOADS
CC
CC ---------------------------------------
CC
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL       XORD(225),YORD(225),DLOAD(512),PROPS(3)
      REAL       THIC,XR,YT,ASXX,ASYY
      INTEGER    NLM(64,8),JDSIDE(512),NUMEL(512),ELTYP(64)
      INTEGER    JJX(450),JDOX(225),JDOY(225),JXL(4,2),MM(10)
      INTEGER    NTOT,NNEL,NNODE,NEQ, ISIDE, NSIDE
      LOGICAL    LESS, MORE, EXISTS
      CHARACTER*5   TITLE(15)
      PARAMETER (PI=3.141592654)

CC
      LOUT = 4
      LIN = 3
CC
      LESS = .FALSE.
      MORE = .FALSE.
CC
      OPEN(LIN,FILE='meshin',STATUS='OLD',FORM='FORMATTED')
CC
CC   CHECK THAT OUTPUT FILE DECLARED NEW DOES NOT  ALREADY EXIST.
CC   IF IT EXISTS, PURGE IT FIRST.
CC
CC ---------------------------------------------------
```

```
CC
      INQUIRE(FILE='meshout',EXIST=EXISTS)
      IF(EXISTS) THEN
       OPEN(UNIT=LOUT,FILE='meshout',STATUS='OLD')
       CLOSE(UNIT=LOUT,STATUS='DELETE')
       ENDIF
       OPEN(UNIT=LOUT,FILE='meshout',STATUS='NEW',
     1       FORM='FORMATTED')
CC ---------------------------------------------------
CC
CC  READ PROBLEM TITLE
CC
        READ(LIN,1) TITLE
        WRITE(*,1) TITLE
        WRITE(LOUT,1) TITLE
 1      FORMAT(15A5)
       READ(LIN,*) NNY,NANG,RO,BM
       READ(LIN,*) ((JXL(I,J),J=1,2),I=1,4)
  10   FORMAT(8I5)
  15   FORMAT(2I5,6F10.1)
  20   FORMAT(7F10.1)
       ISIDE = NANG*2
       NSIDE = 0
       PI4=PI/2./FLOAT(NANG)
       NN=-NNY-1
       NP=NNY+1
CC
          ENDIF
CC
CC  TOP EDGE
CC
        IF(YORD(ISS(1)).EQ.YT.AND.YORD(ISS(3)).EQ.YT) THEN
           IF(ASYY.LT.0.0) THEN
          XXYY = ASYY
          ICHEK = 2
           ENDIF
        ENDIF
CC
 80     CONTINUE
        IF(ICHEK.EQ.0) GOTO 90
         JDS(KK+1) = ISS(1)
```

```
          JDS(KK+2) = ISS(2)
           JDS(KK+3) = ISS(3)
             JDS(KK+4) = NL
          IF(ICHEK.EQ.1) THEN
                 DDL(KK+1) = XXYY
                DDL(KK+2) = 0.0
               DDL(KK+3) = 0.0
             DDL(KK+4) = XXYY
          ENDIF
          IF(ICHEK.EQ.2) THEN
               DDL(KK+2) = XXYY
              DDL(KK+1) = 0.0
             DDL(KK+3) = 0.0
           DDL(KK+4) = XXYY
         ENDIF
         KK = KK+4
 90      CONTINUE
 100     CONTINUE
         NS = KK
         RETURN
         END
           NLM(NL,6)=IM+1
           NLM(NL,7)=IL+2
           NLM(NL,8)=IL+1
CC
        DO 28 K = 1,8
        NTOT = NTOT+1
        NUMEL(NTOT) = NLM(NL,K)
  28    CONTINUE
        IL=IL+2
        IM=IM+1
        IR=IR+2
  30    CONTINUE
CC
CC  -------    NODAL COORDINATES
CC
        ANG=-PI4
        MANG=NANG/2+1
        BMR=BM*RO
        DO 85 J=1,MANG
CC
```

```
CC    CALCULATE RADIAL LENGTHS OF CORNER POINTS ON RADIAL LINE
CC
        ANG=ANG+PI4
        BZ=BM/DCOS(ANG)-1.0
        EPS=0.00001
        XA=0.5
        RISE=1.0
        LOOP = 0
        DENOM = 1000.0
  50    LOOP = LOOP + 1
        IF(LOOP.GT.2000) THEN
        WRITE(*,*) 'TOO MANY LOOPS !!'
        STOP
        ENDIF
        XA=XA*RISE
        BA = 0.0
        WB = 1.0
        DO 55 I = 1,NNY
        WB = WB*XA
        BA = BA+WB
  55    CONTINUE
CC
  59    BAZ = BA-BZ
        IF(BAZ) 60,70,65
  60    LESS = .TRUE.
CC
        IF(MORE) THEN
CC
         IF(DABS(BAZ).LE.0.001) GOTO 70
         XA = XA/RISE
         RISE = RISE + EPS*DENOM
         DENOM = DENOM/10.0
            MORE = .FALSE.
          ENDIF
         RISE = RISE + EPS*DENOM
         GOTO 50
CC
  65      MORE = .TRUE.
CC
          IF(LESS) THEN
CC
```

```
          IF(DABS(BAZ).LE.0.001) GOTO 70
          XA = XA/RISE
          RISE = RISE - EPS*DENOM
          DENOM = DENOM/10.0
          LESS = .FALSE.
          GOTO 68
           ENDIF
          XA = XA/RISE
 68        RISE = RISE - EPS*DENOM
          GOTO 50
CC
  70    CONTINUE
CC
        NN=NN+NNY
        MM(J)=NN+2
        WB=0.
        DO 80 I=1,NP
        WB=WB+R0*XA**(I-1)
        NN=NN+2
        XORD(NN)=WB*DSIN(ANG)
        YORD(NN)=WB*DCOS(ANG)
        IF(DABS(BMR-XORD(NN)).LE.EPS) XORD(NN)=BMR
        IF(DABS(BMR-YORD(NN)).LE.EPS) YORD(NN)=BMR
  80    CONTINUE
  85    CONTINUE
CC
CC   NOW ROTATE ABOUT 45 DEGREES TO COMPLETE CORNER COORDS.
CC
        DO 95 I=2,MANG
        ML=MM(MANG-I+1)-2
        NN=NN+NNY
        DO 90 J=1,NP
        ML=ML+2
        NN=NN+2
        XORD(NN)=YORD(ML)
        YORD(NN)=XORD(ML)
  90    CONTINUE
  95    CONTINUE
CC
CC  ---  INTERMEDIATE NODES
CC
```

```
      DO 110 II=1,NNEL
      DO 100 JJ=2,8,2
      J=NLM(II,JJ)
      IF(JJX(J).GT.0) GOTO 100
      K1=JJ-1
      K2=JJ+1
      IF(K2.GT.7) K2=1
      J1=NLM(II,K1)
      J2=NLM(II,K2)
      XORD(J)=(XORD(J1)+XORD(J2))/2.
      YORD(J)=(YORD(J1)+YORD(J2))/2.
      JJX(J)=1
 100  CONTINUE
 110  CONTINUE
      READ(LIN,*) XL,YB,XR,YT,THIC
      READ(LIN,*) ASXX, ASYY
CC
CC  READ ELEMENT PROPERTIES DATA
CC
CC  PROPS(1)  = YOUNG'S MODULUS FOR MATERIAL TYPE I
CC  PROPS(2)  = POISSON'S RATIO FOR MATERIAL TYPE I
CC  PROPS(3)  = DENSITY FOR MATERIAL TYPE I
CC
      READ(LIN,*) PROPS(1),PROPS(2),PROPS(3)
      IF(ASXX.NE.0.0) NSIDE = NSIDE+ISIDE
      IF(ASYY.NE.0.0) NSIDE = NSIDE+ISIDE
CC
CC          FORM THE NODAL D.O.F ARRAY, JDOX(I),
CC          JDOY(I) (I=1,2, ..,NNODE)
CC
CC  J = 1 MEANS X-DIRECTION
CC  J = 2 MEANS Y-DIRECTION
CC  JDOX(I) = 0 MEANS NODE I IS NOT CONSTRAINED IN X DIRECTION
CC  JDOX(I) = 1 MEANS NODE I IS CONSTRAINED IN X DIRECTION
CC  JDOY(I) = 0 MEANS NODE I IS NOT CONSTRAINED IN Y DIRECTION
CC  JDOY(I) = 1 MEANS NODE I IS CONSTRAINED IN Y DIRECTION
CC
      DO 165 I = 1,NNODE
CC
      X1=(XORD(I)-XL)/2.
      X2=(XORD(I)-XR)/2.
```

```
      Y1=(YORD(I)-YB)/2.
      Y2=(YORD(I)-YT)/2.
      IF(DABS(X1).LE.0.001) XORD(I)=XL
      IF(DABS(X2).LE.0.001) XORD(I)=XR
      IF(DABS(Y1).LE.0.001) YORD(I)=YB
      IF(DABS(Y2).LE.0.001) YORD(I)=YT
CC
      J = 0
      K = 0
      XI = XORD(I)
      YI = YORD(I)
      JDOX(I) = 0
      JDOY(I) = 0
      IF(XI.EQ.XL) THEN
        J = 1
        GOTO 162
      ENDIF
      IF(XI.EQ.XR) THEN
        J = 3
      ENDIF
 162  IF(YI.EQ.YB) THEN
        K = 2
       GOTO 164
      ENDIF
      IF(YI.EQ.YT) THEN
        K = 4
      ENDIF
 164  IF(J.EQ.0.AND.K.EQ.0) GOTO 165
      IF(J.EQ.1.OR.J.EQ.3) JDOX(I) = JXL(J,1)
      IF(K.EQ.2.OR.K.EQ.4) JDOY(I) = JXL(K,2)
 165  CONTINUE
CC
CC  DISTRIBUTED LOADS
CC
       CALL SIDELOD(JDSIDE,DLOAD,NSIDE,XORD,YORD,NUMEL,
     1              NNODE,NTOT,ASXX,ASYY,XL,XR,YB,YT)
CC
CC  MATERIAL TYPE VECTOR, ELTYP(I)
CC
      DO 167 I = 1,NNEL
      ELTYP(I) = 1
```

```
 167   CONTINUE
       NDIST = NSIDE/4
       IPRNT = 1
       NUMAT = 1
       NTYPE = 1
       NPOIN = 0
CC
CC     PRINT RESULTS
CC
        WRITE(LOUT,2) NNODE,NNEL,NUMAT,NTYPE,NPOIN,NDIST,
     1  IPRNT,THIC
        WRITE(*,2) NNODE,NNEL,NUMAT,NTYPE,NPOIN,NDIST,
     1  IPRNT,THIC
        WRITE(LOUT,241) PROPS(1),PROPS(2),PROPS(3)
 241    FORMAT(3F10.4)
 2      FORMAT(7I6,F6.2)
C
C      WRITE(LOUT,250) NNODE,NNEL,PROPS(1),PROPS(2),
CC   1 THIC,PROPS(3), ASXX,ASYY
C      WRITE(LOUT,280) NNODE
       DO 240 I = 1,NNODE
 240   WRITE(LOUT,290) I,XORD(I),YORD(I),JDOX(I),JDOY(I)
C      WRITE(LOUT,260) NNEL
       DO 245 I=1,NNEL
 245   WRITE(LOUT,270) I,ELTYP(I),(NLM(I,J),J=1,8)
CC
       DO 500 I = 1,NSIDE,4
       J = I+3
       WRITE(LOUT,410) (JDSIDE(K),K=I,J),(DLOAD(K),K=I,J)
 500   CONTINUE
 410   FORMAT(4I6,4F10.4)
 250   FORMAT(' ',//,5X,'TOTAL NO. OF NODES           =',I10,/,
     1               5X,'TOTAL NO. OF ELEMENTS        =',I10/,
     2               5X,'MODULUS OF ELASTICITY  (MPA) =',F10.2/,
     3               5X,'POISSONS RATIO               =',F10.4/,
     4               5X,'ELEMENT THICKNESS (M)        =',F10.4/,
     5               5X,'MATERIAL DENSITY (MPA/M)     =',F10.6/,
     8               5X,'APPLIED STRESS IN X-DIRECTION=',F10.3/,
     9               5X,'APPLIED STRESS IN Y-DIRECTION=',F10.3/)
 260   FORMAT(' ',/,15X,'ELEMENT NODE NUMBERING',/,
     1 15X,'TOTAL NO. OF ELEMENTS =',I3,/,
```

```
     1 ' ELEM  ELEM  NODE1  NODE2  NODE3  NODE4',
     1 '  NODE5  NODE6  NODE7  NODE8',/,
     1 '  NO   TYPE',/)
 270   FORMAT(I4,4X,I2,8(4X,I3))
 280   FORMAT(/,2X,'TOTAL NO. OF NODES =',I5,/,
     1 '  NODE    X-COORDINATE  Y-COORDINATE  X-DOF    Y-DOF',/)
 289   FORMAT(' ',I5,2(6X,F10.5))
 290   FORMAT(' ',I5,2(4X,F10.5),2(4X,I3))
 350   FORMAT(10I7)
       STOP
       END
CC ------------------------------
       SUBROUTINE SIDELOD(JDS,DDL,NS,XORD,YORD,NUM,NN,NTOT,
     1                    ASXX,ASYY,XL,XR,YB,YT)
       DIMENSION  XORD(NN),YORD(NN),NUM(NTOT)
       DIMENSION  JDS(NS),DDL(NS),NLM(8),ISS(3)
       INTEGER    KK,NNEL,NN,NS,ICHEK
CC
       KK = 0
       NNEL = NTOT/8
       NL = 0
       DO 100 II = 1,NTOT,8
       NL = NL+1
       J = II - 1
       DO 50 I = 1,8
       J = J+1
 50    NLM(I) = NUM(J)
       DO 90 IS = 1,4
       ICHEK = 0
       ISS(1) = IS*2-1
       ISS(2) = IS*2
       ISS(3) = IS*2+1
       IF(ISS(3).GT.8) ISS(3) = ISS(3)-8
CC
CC  CHECK IF SIDE CORRESPONDS TO ANY EDGE
CC
       DO 60 I = 1,3
       ISS(I) = NLM(ISS(I))
 60    CONTINUE
CC
CC  LEFT EDGE
```

```
CC
        IF(XORD(ISS(1)).EQ.XL.AND.XORD(ISS(3)).EQ.XL) THEN
            IF(ASXX.GT.0.0) THEN
          XXYY = ASXX
          ICHEK = 1
          GOTO 80
           ENDIF
         ENDIF
CC
CC  RIGHT EDGE
CC
        IF(XORD(ISS(1)).EQ.XR.AND.XORD(ISS(3)).EQ.XR) THEN
           IF(ASXX.LT.0.0) THEN
          XXYY = ASXX
          ICHEK = 1
          GOTO 80
           ENDIF
        ENDIF
CC
CC  BOTTOM EDGE
CC
        IF(YORD(ISS(1)).EQ.YB.AND.YORD(ISS(3)).EQ.YB) THEN
           IF(ASYY.GT.0.0) THEN
          XXYY = ASYY
          ICHEK = 2
          GOTO 80
           ENDIF
        ENDIF
CC
CC  TOP EDGE
CC
        IF(YORD(ISS(1)).EQ.YT.AND.YORD(ISS(3)).EQ.YT) THEN
           IF(ASYY.LT.0.0) THEN
          XXYY = ASYY
          ICHEK = 2
           ENDIF
        ENDIF
CC
 80     CONTINUE
        IF(ICHEK.EQ.0) GOTO 90
         JDS(KK+1) = ISS(1)
```

```
            JDS(KK+2) = ISS(2)
              JDS(KK+3) = ISS(3)
                JDS(KK+4) = NL
            IF(ICHEK.EQ.1) THEN
                 DDL(KK+1) = XXYY
                DDL(KK+2) = 0.0
               DDL(KK+3) = 0.0
             DDL(KK+4) = XXYY
            ENDIF
            IF(ICHEK.EQ.2) THEN
                 DDL(KK+2) = XXYY
                DDL(KK+1) = 0.0
               DDL(KK+3) = 0.0
             DDL(KK+4) = XXYY
            ENDIF
          KK = KK+4
90        CONTINUE
100       CONTINUE
          NS = KK
          RETURN
          END
```

Appendix D

LISTING OF A COMPUTER PROGRAM TO CREATE EXCAVATIONS IN A FINITE ELEMENT MESH, VOIDMESH

The listing of a special purpose 8-node quadrilateral mesh generator called CIRCMESH, is given in this appendix. The program has been described in Chapter 5.

```
CC
      REAL          XORD(2000),YORD(2000), DLOAD(200,4)
      REAL          YM(10),PR(10),DENS(10)
      INTEGER       NLM(800,8),IDOF(2000),JDOF(2000),JJX(2000)
      INTEGER       ISIDE(200,4),ICELL(200)
      INTEGER       DELEM(200),NOD(300),NNODE,NNEL,IJ,NS
      CHARACTER*14 FILEA,FILEB
      CHARACTER*5   TITLE(15)
      LOGICAL       EXISTS
CC ------------------------------------------------------------
CC
CC  PROGRAM TO CREATE EXCAVATIONS IN A FINITE ELEMENT MESH.
CC
CC  THIS IS OFTEN NECESSARY WHEN SIMULATING EXCAVATION SEQUENCES
CC  IN A MINE. THE PROCESS INVOLVES DELETEING ELEMENTS AND THE
CC  NODES COMMON TO THEM, OR, ALTERNATIVELY, DELETING NODES AND
CC  THE ELEMENTS ATTACHED TO THEM. IT IS NECESSARY TO HAVE A
CC  COMPUTER PROGRAMME THAT WILL REVISE AN
CC  EXISTING MESH FOR THIS OPERATION.
CC
CC  THE INPUT FILE TO THE FRONT PROGRAMME IS
CC   STRUCTURED AS FOLLOWS:
CC
CC  1. TITLE LINE
CC  2. PROBLEM DATA LINE
CC  3. MATERIAL PROPERTIES LINE(S)
CC  4. NODAL COORDINATES (AND CONSTRAINT) LINES
CC  5. ELEMENT CONNECTIVITY (AND TYPE) LINES, AND
CC  6. POINT AND/OR DISTRIBUTED LOAD LINES
CC
```

```
CC  CREATING AN EXCAVATION WILL AFFECT ITEMS 4,5, AND 6 ABOVE.
CC  IT IS EASIER TO SPECIFY BOTH NODES AND
CC  ELEMENTS INVOLVED ALTHOUGH
CC  IN THEORY, GIVEN ONLY EITHER THE NODES OR THE ELEMENTS,THE
CC  PROGRAM SHOULD BE ABLE TO VERIFY THE OTHER
CC   MISSING COMPONENTS.
CC
CC  WE ASSUME THAT THE ITEMS TO BE DELETED ARE RANKED IN ORDER OF
CC  INCREASING MAGNITUDE, SAY, N(1),N(2),N(3),....N(I).
CC
CC  (A)  NODAL COORDINATE REVISION
CC
CC  OPERATION (A1)
CC
CC     NODES N(1)+1 TO N(2)-1 DECREASE BY 1
CC     NODES N(2)+1 TO N(3)-1 DECREASE BY 2
CC     .
CC     ....... ..... ...... ........
CC     .
CC     NODES N(I-1)+1 TO N(I) DECREASE BY (I-1)
CC
CC
CC
CC  (B)  ELEMENT CONNECTIVITY REVISION
CC
CC  OPERATION (B1)
CC
CC     ELEMENTS N(1)+1 TO N(2)-1 DECREASE BY 1
CC     ELEMENTS N(2)+1 TO N(3)-1 DECREASE BY 2
CC     .
CC     ....... ..... ...... ........
CC     .
CC     ELEMENTS N(I-1)+1 TO N(I) DECREASE BY (I-1)
CC
CC OPERATION (B2)
CC
CC   IDENTIFY THE ELEMENT NODES AND APPLY OPERATION (A1) ABOVE.
CC
CC
CC  (C)     DISTRIBUTED LOAD SIDES REVISION
CC
```

```
CC
CC  FOR EACH LOADED SIDE,
CC
CC  (1) IDENTIFY THE ELEMENT NO. AND APPLY OPERATION (B1) ABOVE,
CC
CC  (2) IDENTIFY THE NODES AND APPLY OPERATION (A1) ABOVE.
CC
CC  AT THE END OF THE EXERCISE, THE TOTAL NODES AND ELEMENTS CC
CC  WILL BE DECREASED BY THE AMOUNTS DELETED. THE PROGRAM
CC  WRITTEN FOR THIS PURPOSE IS CALLED REVISE.F
CC
CC  NELD     = NO. OF ELEMENTS TO DELETE, (MAX = 200)
CC  DELEM(*) = ARRAY HOLDING THE ELEMENT NUMBERS TO BE DELETED
CC  NDD      = NO. OF NODES TO DELETE (MAX = 300)
CC  NOD(*)   = ARRAY HOLDING THE ELEMENT NUMBERS TO BE DELETED
CC ------------------------------------------------------------
CC
      LIN3 = 3
      LIN1=7
      LIN2=8
  3     FORMAT(14A1)
      WRITE(*,*)
      WRITE(*,*) 'NAME OF INPUT FILE ?'
      WRITE(*,*)
      READ (*,*) FILEA
      OPEN(LIN3,FILE= FILEA,STATUS='OLD',ACCESS='SEQUENTIAL',
     1            FORM='FORMATTED')
C        WRITE(*,*)
C        WRITE(*,*) 'TOTAL NO. OF ELEMENTS TO DELETE ?'
C        WRITE(*,*)
         READ (LIN3,*) NELD
C        WRITE(*,*)
C        WRITE(*,*) 'ELEMENTS TO DELETE ?'
C        WRITE(*,*)
         READ(LIN3,*) (DELEM(I),I=1,NELD)
C        WRITE(*,*)
C        WRITE(*,*) 'TOTAL NO. OF NODES TO DELETE ?'
C        WRITE(*,*)
         READ (LIN3,*) NDD
C        WRITE(*,*)
C        WRITE(*,*) 'NODES TO DELETE ?'
```

```
C       WRITE(*,*)
        READ(LIN3,*) (NOD(I),I=1,NDD)
C       WRITE(*,*)
C       WRITE(*,*) 'NAME OF OUTPUT FILE ?'
C       WRITE(*,*)
        READ (LIN3,10) FILEB
CC
CC  SORT DELEM* AND NOD* IN ASCENDING ORDER IF NECESSARY
CC
        CALL SORT2(DELEM,NELD)
        CALL SORT2(NOD,NDD)
CC
        OPEN(LIN1,FILE='newdata.dat',STATUS='OLD',
     1  ACCESS='SEQUENTIAL', FORM='FORMATTED')
CC
CC  CHECK THAT OUTPUT FILE IS NOT ALREADY EXISTING.
CC  IF IT EXISTS, PURGE IT FIRST.
CC
CC ---------------------------------------------------
        INQUIRE(FILE=FILEB,EXIST=EXISTS)
           IF(EXISTS) THEN
          OPEN(UNIT=LIN2,FILE=FILEB,STATUS='OLD')
          CLOSE(UNIT=LIN2,STATUS='DELETE')
        ENDIF
        OPEN(UNIT=LIN2,FILE=FILEB,STATUS='NEW',FORM='FORMATTED')
CC ---------------------------------------------------
CC
CC NNODE = NO. OF NODES.
CC  NNEL = NO. OF ELEMENTS.
CC    NS = NO. OF LOADED SIDES.
CC
        READ(LIN1,10) TITLE
        WRITE(*,10) TITLE
C       WRITE(*,*) 'ENTER NEW TITLE'
        READ (LIN3,10) TITLE
 10     FORMAT(15A5)
        WRITE(LIN2,10) TITLE
        READ(LIN1,*) NNODE,NNEL,N7,N8,N9,NS,N1,D2
        DO 20 I = 1,N7
        READ(LIN1,*) YM(I),PR(I),DENS(I)
 20     CONTINUE
```

```
      DO 30 I = 1,NNODE
 30   READ(LIN1,*) IJ,XORD(IJ),YORD(IJ),IDOF(IJ),JDOF(IJ)
      DO 40 I = 1,NNEL
 40   READ(LIN1,*) N,JJX(N),(NLM(N,J),J=1,8)
      DO 50 I = 1,NS
 50   READ(LIN1,*) (ISIDE(I,J),J=1,4), (DLOAD(I,J),J=1,4)
CC
      CLOSE (LIN1)
CC
CC  NODAL COORD. REVISION
CC
      CALL ALTNOD(XORD,YORD,IDOF,JDOF,NNODE,NOD,NDD)
CC
CC  ELEMENT CONNECTIVITY REVISION
CC
      NZ = 8
      NNDD = NNODE+NDD
CC
       DELEM(NELD+1) = NNEL+1
       DO 100 II = 1,NNEL
         DO 70 I = 1,8
 70      ICELL(I) = NLM(II,I)
       JJ = II
       MINUS = 0
      DO 80 I = 1,NELD+1
CC
       IF(II.EQ.DELEM(I)) GOTO 100
       IF(II.LT.DELEM(I)) THEN
            MINUS = I-1
         GOTO 85
          ENDIF
 80      CONTINUE
 85    JJ = JJ-MINUS
       JJX(JJ) = JJX(II)
       CALL ALTER(ICELL,NZ,NNDD,NOD,NDD)
       DO 90 I = 1,8
       NLM(JJ,I) = ICELL(I)
 90    CONTINUE
 100   CONTINUE
       NNEL = JJ
CC
```

```
CC  LOADED SIDES REVISION
CC
CC  CHANGE THE ELEMENT NUMBERS OF THE ELEMENTS CONTAINING
CC  THE SIDES
CC
         NZ = NS
         DO 110 J = 1,NS
 110    ICELL(J) = ISIDE(J,4)
         NOT = 0
         CALL ALTER(ICELL,NZ,NNEL,DELEM,NELD)
CC
          DO 120 J = 1,NS
 120      ISIDE(J,4) = ICELL(J)
CC
       DO 150 II = 1,NS
       NZ = 3
CC
CC  CHANGE THE NODE NUMBERS OF THE NODES DESCRIBING THE SIDES
CC
        DO 130 J = 1,3
 130    ICELL(J) = ISIDE(II,J)
         NOT = 0
        CALL ALTER(ICELL,NZ,NNDD,NOD,NDD)
       DO 140 J = 1,3
 140    ISIDE(II,J) = ICELL(J)
 145    FORMAT(F12.2,2F12.4)
 150    CONTINUE
CC
       WRITE(LIN2,*) NNODE,NNEL,N7,N8,N9,NS,N1,D2
       DO 155 I = 1,N7
       WRITE(LIN2,145) YM(I),PR(I),DENS(I)
 155    CONTINUE
       DO 160 I = 1,NNODE
 160    WRITE(LIN2,200) I,XORD(I),YORD(I),IDOF(I),JDOF(I)
       DO 170 I = 1,NNEL
 170    WRITE(LIN2,210) I,JJX(I), (NLM(I,J),J=1,8)
       DO 180 I = 1,NS
 180    WRITE(LIN2,220) (ISIDE(I,J),J=1,4), (DLOAD(I,J),J=1,4)
        CLOSE(LIN2)
 190    CONTINUE
 200    FORMAT(I5,2F10.3,2I5)
```

```
 210    FORMAT(10I5)
 220    FORMAT(4I5,4F10.3)
CC
       STOP
       END
CC*****************
      SUBROUTINE ALTNOD(XX,  YY  ,ID,  JD,  NN,  ND, KK)
      DIMENSION XX(NN),YY(NN),ID(NN),JD(NN),ND(KK+1)
CC
      ND(KK+1) = NN+1
      DO 50 II = 1,NN
      JJ = II
      MINUS = 0
        DO 30 I = 1,KK+1
CC
         IF(II.EQ.ND(I)) GOTO 50
         IF(II.LT.ND(I)) THEN
              MINUS = I-1
           GOTO 40
            ENDIF
  30        CONTINUE
  40              JJ = JJ-MINUS
           XX(JJ) = XX(II)
           YY(JJ) = YY(II)
           ID(JJ) = ID(II)
           JD(JJ) = JD(II)
  50     CONTINUE
      NN = JJ
CC
      RETURN
      END
CC  ----------------------------------------
      SUBROUTINE ALTER(IC,  LL,  MM,  ND, KK)
      DIMENSION IC(LL),ND(KK+1)
CC
      ND(KK+1) = MM+1
      MINUS = 0
      DO 50 II = 1,LL
      JJ = IC(II)
        DO 30 I = 1,KK+1
CC
```

```
          IF(JJ.EQ.ND(I)) GOTO 50
          IF(JJ.LT.ND(I)) THEN
               MINUS = I-1
            GOTO 40
             ENDIF
  30        CONTINUE
  40        CONTINUE
                  JJ = JJ-MINUS
                    IC(II) = JJ
  50      CONTINUE
CC
          RETURN
          END
CC ---------------------------------------
         SUBROUTINE SORT2(JD,NN)
CC
CC   SORTS A LINEAR ARRAY OF NN ITEMS IN ASCENDING ORDER
CC   IN A DOUBLE-ENDED OPERATION.
CC   THERE ARE   NN*(NN+1)/4      ITERATIONS FOR NN EVEN, AND,
CC               (NN-1)*(NN+3)/4  ITERATIONS FOR NN ODD.
CC   THIS COMPARES TO NN*(NN+1)/2 ITERATIONS FOR A SINGLE-ENDED
CC   SORT, A SAVING OF NEARLY 50 PERCENT.
CC
         DIMENSION  JD(200)
         INTEGER    JMIN,JMAX,IMIN,IMAX,JJ,KK,ID,IE
CC
         JJ = 0
         KK = NN+1
         JK = NN/2
         IE = 10000000
CC
         DO 60 I = 1,JK
         IMIN = IE
         IMAX = 0
         JMIN = IE
         JMAX = 0
         JJ = JJ+1
         KK = KK-1
CC
         DO 20 J = JJ,KK
         ID = JD(J)
```

```
      IF(IMAX.LT.ID) THEN
       JMAX = J
      IMAX = ID
      ENDIF
      IF(IMIN.GT.ID) THEN
      JMIN = J
      IMIN = ID
      ENDIF
   20 CONTINUE
CC
      IF(JMIN.EQ.IE .OR. JMAX.EQ.0) GOTO 60
      IF(JMIN.EQ.JJ) GOTO 30
      IJ = JD(JJ)
      JD(JJ) = JD(JMIN)
      JD(JMIN) = IJ
CC
   30 IF(JMAX.EQ.KK) GOTO 40
      IF(JMAX.EQ.JJ) JMAX = JMIN
CC
      IJ = JD(KK)
      JD(KK) = JD(JMAX)
      JD(JMAX) = IJ
   40 CONTINUE
   60 CONTINUE
CC
      RETURN
      END
```

APPENDIX E
DERIVATION OF THE [P], [R] AND [L] MATRICES

E.1 Formulation of the $[P]$ Matrix

E.1.1 The Jacobian

There is a one-to-one correspondence between the global coordinates (x, y) and the local coordinates (s, t). By using the Chain Rule of differentiation, the following relation can be established between the local and global coordinates:

$$\begin{Bmatrix} \partial/\partial s \\ \\ \partial/\partial t \end{Bmatrix} = [J] \begin{Bmatrix} \partial/\partial x \\ \\ \partial/\partial y \end{Bmatrix} \tag{E.1}$$

where the Jacobian, $[J]$, is defines as

$$[J] = \begin{bmatrix} \partial x/\partial s & \partial y/\partial s \\ \\ \partial x/\partial t & \partial y/\partial t \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \tag{E.2}$$

Note that $[J^{-1}]^T = [J^T]^{-1}$ and that

$$[J]^{-1} = \begin{bmatrix} \alpha_{22} & -\alpha_{12} \\ \\ -\alpha_{21} & \alpha_{11} \end{bmatrix} / \Delta \tag{E.3}$$

where

$$\Delta = det|J| = \alpha_{11}\alpha_{22} - \alpha_{12}\alpha_{21}$$

By taking the second partial derivatives of the Airy Stress Function Φ with respect to local coordinates s and t, it can be shown that

$$[B] = [J]^{-1} [A] [J^{-1}]^T \tag{E.4}$$

where

$$[A] = [A_0] - [A_x] - [A_y] \tag{E.5}$$

$$[B] = \begin{bmatrix} \partial^2\Phi/\partial x^2 & \partial^2\Phi/\partial x\partial y \\ \partial^2\Phi/\partial x\partial y & \partial^2\Phi/\partial y^2 \end{bmatrix} = \begin{bmatrix} \sigma_y & -\tau_{xy} \\ -\tau_{xy} & \sigma_x \end{bmatrix} \tag{E.6}$$

The components of $[A]$ can be written as follows,

$$[A] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{E.7}$$

where

$$[A_0] = \begin{bmatrix} \partial^2\Phi/\partial s^2 & \partial^2\Phi/\partial s\partial t \\ \partial^2\Phi/\partial s\partial t & \partial^2\Phi/\partial t^2 \end{bmatrix} \tag{E.8}$$

$$[A_x] = \partial\Phi/\partial x \begin{bmatrix} \partial^2 x/\partial s^2 & \partial^2 x/\partial s\partial t \\ \partial^2 x/\partial s\partial t & \partial^2 x/\partial t^2 \end{bmatrix} \tag{E.9}$$

$$[A_y] = \partial\Phi/\partial y \begin{bmatrix} \partial^2 y/\partial s^2 & \partial^2 y/\partial s\partial t \\ \partial^2 y/\partial s\partial t & \partial^2 y/\partial t^2 \end{bmatrix} \tag{E.10}$$

Each of the 2x2 matrices in the expressions above is symmetric and the $[B]$ matrix can be expressed entirely in terms of the undetermined stress multipliers, β_i.

Equation (E.4) can be expanded by substituting Equations (E.3) and (E.7) into it to yield:

$$[B] = \begin{bmatrix} \sigma_y & -\tau_{xy} \\ -\tau_{xy} & \sigma_x \end{bmatrix} = \begin{bmatrix} \alpha_{22} & -\alpha_{12} \\ -\alpha_{21} & \alpha_{11} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \alpha_{22} & -\alpha_{21} \\ -\alpha_{12} & \alpha_{11} \end{bmatrix} / \Delta^2 \quad \text{(E.11)}$$

which can be expanded and rearranged to yield:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{1}{\Delta^2} [S] \begin{Bmatrix} A_{11} \\ A_{22} \\ A_{12} \end{Bmatrix} \quad \text{(E.12)}$$

where

$$[S] = \begin{bmatrix} \alpha_{21}^2 & \alpha_{11}^2 & -2\alpha_{11}\alpha_{21} \\ \alpha_{22}^2 & \alpha_{12}^2 & -2\alpha_{12}\alpha_{22} \\ \alpha_{21}\alpha_{22} & \alpha_{11}\alpha_{12} & -(\alpha_{11}\alpha_{22} + \alpha_{12}\alpha_{21}) \end{bmatrix} \quad \text{(E.13)}$$

Equation (E.5) can be expanded by substituting Equations (E.7), (E.8), (E.9) and (E.10) into it, and the resulting expression can be rearranged to yield:

$$\begin{Bmatrix} A_{11} \\ A_{22} \\ A_{12} \end{Bmatrix} = [Z]\{\beta\} - [F][J]^{-1} [W] \{\beta\} \quad \text{(E.14)}$$

where

$$[Z]\{\beta\} = \begin{Bmatrix} \partial^2\Phi/\partial s^2 \\ \partial^2\Phi/\partial t^2 \\ \partial^2\Phi/\partial s\partial t \end{Bmatrix} \quad \text{(E.15)}$$

$$[F] = \begin{bmatrix} \partial^2 x/\partial s^2 & \partial^2 y/\partial s^2 \\ \partial^2 x/\partial t^2 & \partial^2 y/\partial t^2 \\ \partial^2 x/\partial s\partial t & \partial^2 y/\partial s\partial t \end{bmatrix} \quad \text{(E.16)}$$

$$[J]^{-1}\,[W]\,\{\beta\} = \left\{ \begin{array}{c} \partial\Phi/\partial x \\ \\ \partial\Phi/\partial y \end{array} \right\} \tag{E.17}$$

From equations (E.12), and (E.14), it can be deduced that the [P] matrix is given by the expression:

$$[P] \;=\; \frac{1}{\Delta^2}\,[S]\,\Big([Z] \;-\; [F]\,[J]^{-1}\,[W]\Big) \tag{E.18}$$

$[W]$ is a 2x18 matrix whose coefficients are functions of s and t as derived from the first partial derivative of Φ with respect to s and t. $[Z]$ is a 3x18 matrix with coefficients as functions of s and t, as derived from the second partial derivatives of Φ with respect to s and t.

The matrix, $[F]$, when expanded in full results in the following expression:

$$[F] \;=\; \left[\begin{array}{c} \partial^2 N_1/\partial s^2 \;\; \partial^2 N_2/\partial s^2 \;\; \ldots \;\; \partial^2 N_8/\partial s^2 \\ \\ \partial^2 N_1/\partial t^2 \;\; \partial^2 N_2/\partial t^2 \;\; \ldots \;\; \partial^2 N_8/\partial t^2 \\ \\ \partial^2 N_1/\partial s\partial t \;\; \partial^2 N_2/\partial s\partial t \;\; \ldots \;\; \partial^2 N_8/\partial s\partial t \end{array} \right] \left[\begin{array}{cc} x_1 & y_1 \\ \\ x_2 & y_2 \\ \\ \vdots & \vdots \\ \\ x_8 & y_8 \end{array} \right] = \left[\begin{array}{cc} 0 & 0 \\ \\ 0 & 0 \\ \\ X_c & Y_c \end{array} \right] \tag{E.19}$$

where

$X_c = (x_1 - x_3 + x_5 - x_7)/4,$

$Y_c = (y_1 - y_3 + y_5 - y_7)/4$

This leads to further simplification in the computation of the $[P]$ matrix. The full expressions for $[W]$ and $[Z]$ are as follows:

$$[W]^T = \begin{bmatrix} 2s & 0 \\ 0 & 2t \\ t & s \\ 3s^2 & 0 \\ 0 & 3t^2 \\ 2st & s^2 \\ t^2 & 2st \\ 4s^3 & 0 \\ 0 & 4t^3 \\ t^3 & 3st^2 \\ 2st^2 & 2s^2t \\ 3s^2t & s^3 \\ 5s^4 & 0 \\ 0 & 5t^4 \\ 4s^3t & s^4 \\ t^4 & 4st^3 \\ 3s^2t^2 & 2s^3t \\ 2st^3 & 3s^2t^2 \end{bmatrix}$$

$$[Z]^T = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 6s & 0 & 0 \\ 0 & 6t & 0 \\ 2t & 0 & 2s \\ 0 & 2s & 2t \\ 12s^2 & 0 & 0 \\ 0 & 12t^2 & 0 \\ 0 & 6st & 3t^2 \\ 2t^2 & 2s^2 & 4st \\ 6st & 0 & 3s^2 \\ 20s^3 & 0 & 0 \\ 0 & 20t^3 & 0 \\ 12s^2t & 0 & 4s^3 \\ 0 & 12st^2 & 4t^3 \\ 6st^2 & 2s^3 & 6s^2t \\ 2t^3 & 6s^2t & 6st^2 \end{bmatrix} \tag{E.20}$$

E.2 Formulation of the $[G]$ Matrix

The $[G]$ and $[G_b]$ are represented by the equations

$$[G] = \int_{\partial V_n} [R]^T [L] ds$$

$$[G_b] = \int_{\partial V_n} [R_b]^T [L] ds$$

where $[R]$ and $[R_b]$ are expressions of the tractions acting on the element side and $[L]$ is the expression for the displacement of the side.

E.2.1 Surface Tractions due to Assumed Stress Fields

Consider the forces acting on the sides of a typical quadrilateral element ABCD as shown in Figure E.1.

The boundary surface tractions can be expressed as:

$$\{T\}^T = \{T_{AB}\ T_{BC}\ T_{CD}\ T_{DA}\} \tag{E.21}$$

where

$$\{T_{AB}\} = \{-\tau_{AB}, -\sigma_{n_{AB}}\}$$

$$\{T_{BC}\} = \{\sigma_{n_{BC}}, \tau_{BC}\}$$

$$\{T_{CD}\} = \{\tau_{CD}, \sigma_{n_{CD}}\}$$

$$\{T_{DA}\} = \{-\sigma_{n_{DA}}, -\tau_{DA}\} \tag{E.22}$$

The boundary surface tractions can also be expressed as:

$$T_i = \sigma_{ij} \nu_j \tag{E.23}$$

where, for any particular boundary j, ν_j is the direction cosine of the normal to it.

Consider the boundary surface AB whose outward-drawn normal makes an angle θ with the

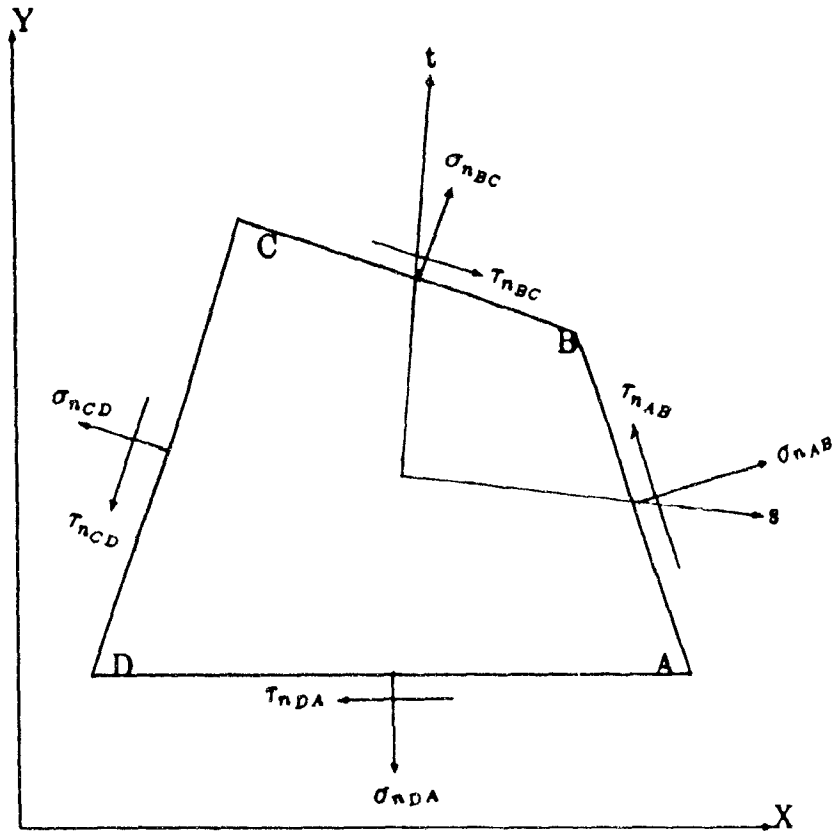


Figure E.1: Generalized Boundary Forces Acting on the Sides of a Quadrilateral Element

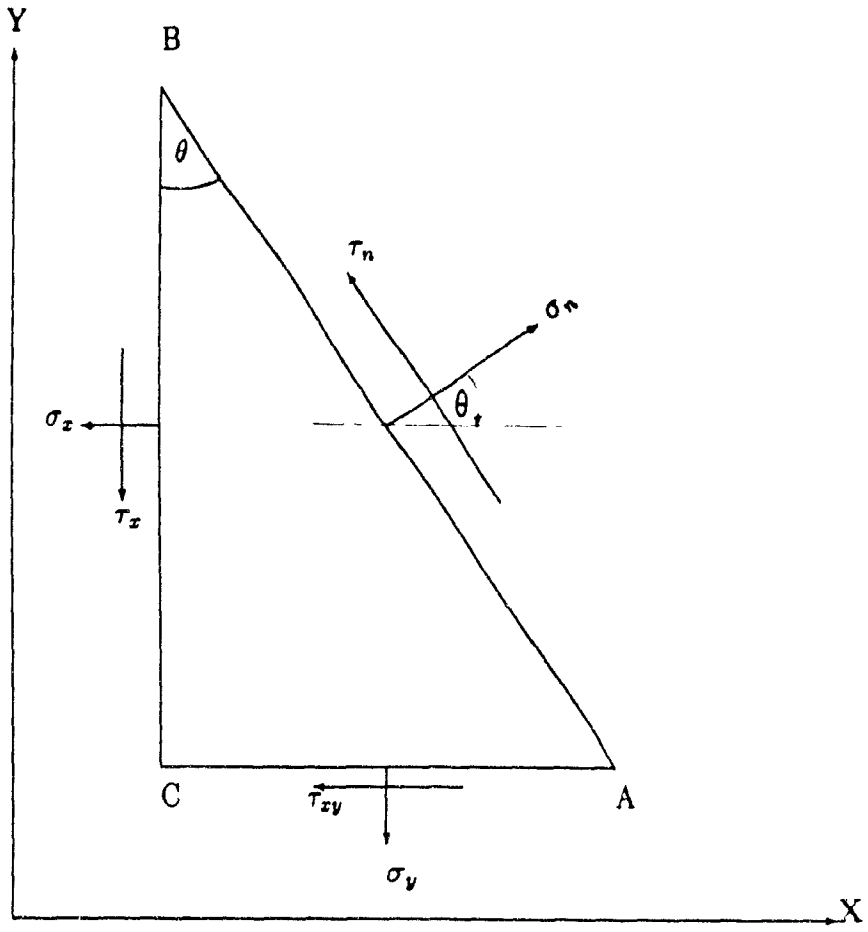


Figure E.2: Components of Stress Acting on Side AB of a Quadrilateral Element

positive direction of the x-axis as shown in Figure E.2.

If the stresses acting are σ_x, σ_y, τ_{xy} as shown, then

$$\begin{Bmatrix} T_x \\ T_y \end{Bmatrix} = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \{\sigma\} \tag{E.24}$$

where the stresses are given by Equation (3.2) and the surface tractions by Equation (3.4) of Chapter 3.
From Equations (3.2), (3.4) of Chapter 3, and (E.24),

$$[R] = \Delta t \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} [P] \tag{E.25}$$

$$[R_b] = \Delta t \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} [P_b] \tag{E.26}$$

where Δt is the thickness of the element.

E.2.2 The [L] Matrix

Consider the inter-element boundary AB of Figure E.1. Using 8-node isoparametric elements, let the node numbering be anti-clockwise as shown. Let u_s, v_s be the horizontal and vertical components of the inter-element displacements and let (u_i, v_i) be the global horizontal and vertical components of the displacements of node i where $(1 \leq i \leq 3)$.
Since we are using isoparametric formulation, we can write:

$$\begin{Bmatrix} u_s \\ v_s \end{Bmatrix} = \begin{Bmatrix} \sum_1^3 N_i u_i \\ \sum_1^3 N_i v_i \end{Bmatrix} = [N_{AB}] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \tag{E.27}$$

where the subscripted components in

$$[N_{AB}] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \tag{E.28}$$

are the shape functions for nodes 1, 2 and 3 respectively of side AB. Similar expressions can be written for the other sides of the quadrilateral element. The 6x1 vector of displacements in Equation (E.27) is precisely part of the 16-component global displacement vector, $\{q\}$, of the element. $\{q\}$ comprises 2-component global displacements per node. Thus, the inter-element displacements of side AB can be expressed in terms of the entire $\{q\}$ vector by expanding the 2x6 $[N_{AB}]$ matrix to 2x16 by filling the rest of the positions with zeros. Thus, if we consider the entire element sides, numbered 1 - 4, then, the $[L]$ matrix is seen to be an 8x16 matrix with each side subscribing a 2x16 sub-matrix.

When the quadrilateral element is expressed in terms of the local (s,t) coordinates as shown in Figure E.3, and if we write:

$$\begin{Bmatrix} s_i \\ s_j \\ t_i \\ t_j \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} (1-s) \\ (1+s) \\ (1-t) \\ (1+t) \end{Bmatrix} \tag{E.29}$$

then, the complete [L] matrix is given by

$$[L]^T = \begin{bmatrix}
-ss_i & 0 & 0 & 0 & 0 & 0 & -tt_i & 0 \\
0 & -ss_i & 0 & 0 & 0 & 0 & 0 & -tt_i \\
4s_is_j & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4s_is_j & 0 & 0 & 0 & 0 & 0 & 0 \\
ss_j & 0 & -tt_i & 0 & 0 & 0 & 0 & 0 \\
0 & ss_j & 0 & -tt_i & 0 & 0 & 0 & 0 \\
0 & 0 & 4t_it_j & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4t_it_j & 0 & 0 & 0 & 0 \\
0 & 0 & tt_j & 0 & ss_j & 0 & 0 & 0 \\
0 & 0 & 0 & tt_j & 0 & ss_j & 0 & 0 \\
0 & 0 & 0 & 0 & 4s_is_j & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4s_is_j & 0 & 0 \\
0 & 0 & 0 & 0 & -ss_i & 0 & tt_j & 0 \\
0 & 0 & 0 & 0 & 0 & -ss_i & 0 & tt_j \\
0 & 0 & 0 & 0 & 0 & 0 & -tt_i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -tt_i
\end{bmatrix} \tag{E.30}$$

E.3 Interpolation Functions for the 8-Node Quadrilateral Element

The interpolation functions used with the 8-node isoparametric element can be derived from the so called Serendipity elements and can be shown to be given as.

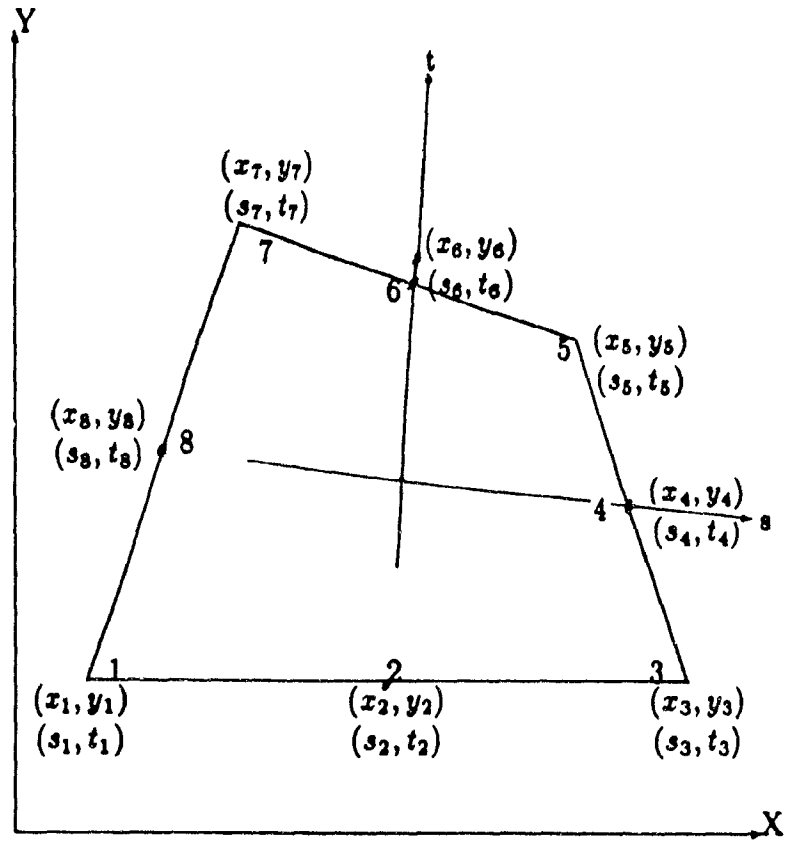


Figure E.3: Global and Local Coordinate System for Quadrilateral Element

$$N_i(s,\ t)\ =\ (1+\ s_i\ s)\ (1+\ t_i\ t)\ (s_i\ s+\ t_i\ t-\ 1)\ /4 \qquad \text{(E.31)}$$

for the corner nodes where i = 1, 3, 5, and 7. At midside nodes where i = 2, 4, 6 and 8, we have:

$$N_i(s,\ t)\ =\ s_i^2\ (1-\ t^2)\ (1+\ s_i\ s)\ /2+\ t_i^2\ (1-\ s^2\ )\ (1+\ t_i\ t)\ /2 \qquad \text{(E.32)}$$

The interpolation functions and their first and second derivatives are utilized in the formulation of the matrix equations. They are shown in the table below:

i	N_i	$\partial N_i/\partial s$	$\partial N_i/\partial t$	$\partial^2 N_i/\partial s^2$	$\partial^2 N_i/\partial t^2$	$\partial^2 N_i/\partial s\partial t$
1	$-s_it_i(s+t+1)$	$\frac{1}{2}t_i(2s+t)$	$\frac{1}{2}s_i(2t+s)$	t_i	s_i	$(s_i-t_j+0.25)$
2	$4s_is_jt_i$	$-2st_i$	$-2s_is_j$	$-2t_i$	0	s
3	$s_jt_i(s-t-1)$	$\frac{1}{2}t_i(2s-t)$	$\frac{1}{2}s_j(2t-s)$	t_i	s_j	$(t_j-s_j-0.25)$
4	$4s_jt_it_j$	$2t_it_j$	$-2s_jt$	0	$-2s_j$	$-t$
5	$s_jt_j(s+t-1)$	$\frac{1}{2}t_j(2s+t)$	$\frac{1}{2}s_j(2t+s)$	t_j	s_j	$(s_j-t_i+0.25)$
6	$4s_is_jt_j$	$-2st_j$	$2s_is_j$	$-2t_j$	0	$-s$
7	$s_it_j(-s+t-1)$	$\frac{1}{2}t_j(2s-t)$	$\frac{1}{2}s_i(2t-s)$	t_j	s_i	$(s_j-t_j-0.25)$
8	$4s_it_it_j$	$-2t_it_j$	$-2ts_i$	0	$-2s_i$	t

(E.33)

where

$$\begin{Bmatrix} s_i \\ s_j \\ t_i \\ t_j \end{Bmatrix} = \frac{1}{2}\begin{Bmatrix} (1-s) \\ (1+s) \\ (1-t) \\ (1+t) \end{Bmatrix} \qquad \text{(E.34)}$$

Node numbering is sequentially counter clockwise, starting from the node nearest to the bottom left corner as in Figure 4.2 of Chapter 4.