LOWER AND UPPER BOUND LIMIT ANALYSES FOR STABILITY PROBLEMS IN GEOTECHNICAL ENGINEERING

LI DAZHONG

Ph.D

The Hong Kong Polytechnic University

2014

THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

LOWER AND UPPER BOUND LIMIT ANALYSES FOR STABILITY PROBLEMS IN GEOTECHNICAL ENGINEERING

LI Dazhong

A Thesis submitted in partial fulfilment of the requirements

for the degree of philosophy

DECEMBER 2013

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledge has been made in the text.

______________________________ (Signed)

LI Dazhong (Name of Student)

Abstract of thesis entitled

LOWER AND UPPER BOUND LIMIT ANALYSES FOR STABILITY PROBLEMS IN GEOTECHNICAL ENGINEERING

submitted by **LI Dazhong**

for the degree of Doctor of Philosophy

at The Hong Kong Polytechnic University in December 2013

Stability problems are important issues in geotechnical engineering, and many methods have been developed over the past forty years. However, virtually all of the existing methods have either practical or theoretical limitations when applied under general conditions. With the advances in computer hardware and numerical algorithms for convex programming, there are growing interests in the application of numerical limit analysis for the stability problems for the Ultimate Limit State (ULS) design. Lower and upper bound limit analyses are becoming competitive due to its ability to provide rigorous lower and upper bounds to the exact solutions without arbitrary assumptions.

In this research, the Finite-Element-based Limit Analysis (FELA) in which the stress and velocity field are discretised with finite dimensional element spaces is discussed. Stability problems are formulated as the solution of convex optimizations in the present study. The Mohr-Coulomb (MC) yield criterion for plane strain analysis and the Drucker Prager (DP) yield criterion for full three-dimensional analysis are transformed to second order conic constraints such that the resulting large scale optimization problem could be solved by the standard Second Order Cone Programming (SOCP) solvers with great efficiency. For MC materials under the full three-dimensional condition, hyperbolic smoothing in the

meridional plane and round-off of corners in the octahedral plane are applied to render the yield function to be everywhere differentiable. The NLP optimization stemming from the formulation is then solved efficiently by the primal-dual interior point algorithms. This technique will ensure that the developed method can achieve high accuracy while maintaining a fast solution suitable for complicated practical problems.

Slip bands for most of the geotechnical problems are highly localized and the quality of the FELA solution is considerably affected by the configuration of the mesh. A variety of strategies for error estimation and mesh adaptation are investigated in the current work. The residual error estimator and recovery-based error estimators that have been proposed for the FEM are tailored for the FELA. Comparisions on the performances of these error estimators are made to the yield criterion slackness and the deformation based method in this study.

With an iterative local refinement technique, a coarse mesh without prior knowledge on the approximate solution can be used as the initial discretisation input. The optimal mesh distribution and the failure mechanism will be obtained as part of the solution, which enables solutions with satisfactory degree of accuracy to be obtained from a poor initial mesh and the need for the prior knowledge in generating high quality mesh is then removed, which is of great practical value for complicated problems. The method as developed in the present study can usually solve a problem within five to fifteen minutes (for several thousand elements), which is considered to be acceptable so that the present work can be useful to both theoretical study as well as practical application.

To consider the effect of the nonlinearity of the yield envelope on the collapse load, a power-type nonlinear yield criterion has also been studied in a systematic manner under the framework of the FELA. Comparisons are made between the results from the present study and those from the literature obtained by either conventional upper bound methods or limit equilibrium method.

To facilitate the extending of the FELA, a code library written in objective-oriented programming language C++ has been developed. Abstract classes such as *OptSolver*, *YieldCriterion,* and *LAsystem etc.,* are developed such that integration of other yield criteria and more powerful solvers that will appear in later development can be greatly simplified.

LIST OF PUBLICATIONS

Cheng, Y. M., **Li, D. Z.,** Li, L., Sun, Y. J., Baker, R., and Yang, Y. (2011). "Limit equilibrium method based on an approximate lower bound method with a variable factor of safety that can consider residual strength." Computers and Geotechnics, 38(5), 623-637.

Li, D. Z., and Cheng, Y. M. (2012). "Lower bound limit analysis using nonlinear failure criteria." 2012 International Conference on Structural Computation and Geotechnical Mechanics. Procedia Earth and Planetary Science City: Kun Ming China, 170-174.

Li, D. Z., Cheng, Y. M., Wang, J. A., et al. (2013). “Finite-element-based limit analysis with mesh adaptation and applications in geotechnical engineering. ”Chinese Journal of Geotechnical Engineering, 35(5) ,922-929

Li, D. Z., and Cheng, Y. M. (2013). “Automatic mesh adaptation in limit analysis.” Soils and Foundations (Under Review)

Cheng, Y. M., **Li, D. Z.,** N. Li, Lee, Y. Y and Au, S. K. (2013). “An application of the global optimization method for the solution of partial differential equations with applications to geotechnical problems”, Journal of Mechanics, 29(3), 493-506.

Cheng, Y. M., Li, N., **Li, D. Z**. and Lansivaara, T. (2013). “Relation between bearing capacity, lateral earth pressure and slope stability by the slip-line and extremum limit equilibrium method.” Journal of Geotechnical and Geoenvironmental Engineering-ASCE (under review)

Li, D. Z., Cheng, Y. M. (2013). “Formualtion of the the upper bound axiymmetric problmes as Second Order Programming.” International Journal of Numerical Methods in Geotehcnial engineering (under preparation).

ACKNOWLEDGEMENTS

I would like to express my great gratitude and sincere thanks to my supervisor Dr. Y.M. Cheng. Without his encouragement and guidance, this work would never be near possible. I have been extremely lucky to have Dr. Y.M. Cheng as my supervisor, who is both an excellent scientist and an experienced geotechnical engineer. His constant supports and guidance have kept me updated to the latest science developments and practical difficulties in the field design. I am also more than grateful to his faith in my work, the patience and encouragement during the study. Thanks should also go to the valuable discussions with him in history, economy, career plan etc., which has made working with him inspiring and enjoyable.

I must express my gratitude to the support from The Hong Kong Polytechnic University for this research work and administrative and technical supports from the staff of the Department of Civil and Environmental Engineering.

I should also thank Professor J.A. Wang at the University of Science and Technology, Beijing, for his introduction to the field of geotechnical engineering and continuous encouragement while I am studying in Hong Kong. Many thanks are extended to Dr. K. K. Wong of ENTPE and Dr. J. Yang of The University of Hong Kong for their valuable comments on the thesis and discussions.

All my friends and colleagues, Dr. F. Tong, S.S. Fang, Y. Yang, D.B. Yang, Nancy Li, and others deserve thanks for their efforts of making the my life in Hong Kong filled with warm friendship and love.

At last, I am indebted to my father S.W. Li, my mother C.L, Gao and my brother Dazhi Li for their unconditional support and love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBRIVATIONS

DP	Drucker-Prager
FELA	Finite-Element-based Limit Analysis
FEM	Finite Element Method
HB	Hoek-Brown
KKT	Karush-Kuhn-Tucker condition/ optimality condition for constrained optimization
LEM	Limit Equilibrium Method
LP	Linear Programming
MC	Mohr-Coulomb
MOSEK	A software package for the solution of SOCP
NLP	Nonlinear Programming
NPE	Node per element
SDP	positive Semi-definite Programming
SLM	Slip Line Method
SOCP	Second Order Cone Programming
SQP	Sequential Quadratic Programming

NOTATION

$\alpha, \alpha^{*}, \alpha^{UB}, \alpha^{LB}$	Load multiplier, exact load multiplier, lower bound multiplier, upper bound multiplier
β_{ij}	Direction cosines of local axis with respect to the global axis
B	Set of stress points that are plastically admissible, i.e. $\boldsymbol{\sigma} \in$ B $\Longrightarrow \boldsymbol{\sigma}$PA
C	Hyperplane of velocity field on which the external work of reference external force equal to 1, i.e., $C = \{\hat{\mathbf{u}} \vert P_{ext}(\mathcal{F}^{0}_{ext}, \hat{\mathbf{u}}) = 1 \text{ and } \hat{\mathbf{u}}\text{KA} \}$
$\mathcal{C}$	Set of second order cone
$\mathcal{E}^{\mathcal{O}}$	Set of edges shared by two adjacent elements
$\mathcal{E}^{\mathcal{D}}, \mathcal{E}^{\mathcal{N}}$	Set of edges associated with velocity boundaries and stress boundaries
$\boldsymbol{\varepsilon}, \hat{\boldsymbol{\varepsilon}}$	Strain tensor, virtual stain tensor
$\mathbf{f}(\mathbf{x})$	Vector of the body force
$f(\boldsymbol{\sigma})$	Yield function
F^{0}_{ext}	Reference external load or starting external load that to be multiplied by α
Γ^{N}	Neumann boundary/ stress boundary
Γ^{D}	Dirichlet boundary / velocity boundary
KA	Kinematically admissible defined as KA: $\mathbf{u}\text{KA} \Leftrightarrow \left\{ \boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \mathbf{u}\boldsymbol{\nabla})\ x \in \Omega \text{ and } \mathbf{u} = \bar{\mathbf{u}}\ \ x \in \Gamma^{D} \right\}$

Ω, Ω^e	Domain of the study and domain of and element
$\partial\Omega$, $\partial\Omega^e$	Boundary of the domain and boundary of and element
PA	PA: $\boldsymbol{\sigma}\mathrm{PA} \Leftrightarrow f(\boldsymbol{\sigma}) \leq 0\ \forall \mathbf{x} \in \Omega$, plastically admissible, i.e., stresses nowhere violate yield function.
P_{int}	Internal power dissipation
P_{ext}	External power dissipation
s_{ij}	Deviatoric stresses $\sigma_{ij} - \sigma_m \delta_{ij}$
$\boldsymbol{\sigma}: \boldsymbol{\varepsilon}$	Scalar product of stress tensor and strain tensor
$\boldsymbol{\sigma}, \widehat{\boldsymbol{\sigma}}, \boldsymbol{\sigma}^c, \boldsymbol{\sigma}^*$	Tensor notation for stress field, virtual stress field, true stress field and the stress field that maximise the internal power dissipation
$\mathbf{σ}, \mathbf{σ}_h,\ \mathbf{σ}_h^e$	Vector notation of stress field, stress field approximated in finite element space with typical element size h and discretised stress within a particular element *e*
σ_m	Mean stresses $\frac{1}{D}\sum_{i=1}^{N} \sigma_{ii}$
SA	Statically admissible, i.e., stress field satisfies the equilibrium condition and the stress boundary condition
$\mathbf{t}(\mathbf{x})$	Vector of the surface traction
$\mathbf{u}, \widehat{\mathbf{u}}, \mathbf{u}^{\mathrm{UB}}, \mathbf{u}^{\mathrm{LB}}$	Vector of velocity, vector of virtual velocity, velocity obtained in upper bound analysis, velocity field in lower bound analysis
$\mathbf{x}$	Location vector
X, Y	Infinite dimensional function space for stresses and velocities
$X_h Y_h$	Finite dimensional function space for stresses and velocities
$\xi_e^{e'}, \xi_e^N, \xi_e^D$	An edge shared by elements, on the stress boundary and on the velocity boundary, respectively

$\lVert * \rVert$	Normal of vector *
$[\![*]\!]$	Jump of a quantity *

CHAPTER 1: INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

Stability analysis is one of important areas in geotechnical engineering. Examples of stability problems are the determination of bearing capacity of footings, lateral earth thrust on a retaining wall, the factor of safety a slope, critical height of a vertical wall, etc. Available methods for this purpose can be categorized into four groups, limit equilibrium method, slip line method, limit analysis, and the complete numerical methods. These methods are briefly described in the following sections, which is followed by a literature review on the previous works on limit analysis, in particular the numerical limit analysis based on the finite element discretisation. The objective and the outline of the research are presented at the end of this chapter.

1.1.1 Limit equilibrium method

Up to the present, Limit Equilibrium Method (LEM) is still playing a dominant role in the stability analysis of geotechnical problems mainly due to its simplicity and the capability in obtaining approximate but realistic solutions. With this method, the collapse load is found by applying the force or the moment equilibrium on the failure mechanism consisting of rigid blocks (wedges) and prescribed failure surface (planar, circular, logspiral, curved or combination of these).

The concept of LEM could be traced back to as early as 1773 when Coulomb (1773) determined the earth pressure on a retaining wall by assuming a planer failure surface. The method was later popularized by various engineers and researchers and has now become a widely accepted approach in practice. Detailed discussion of LEM are presented in the classic soil mechanic books by Terzaghi (1943), Taylor (1948) and more recently Murthy (2003) as well as that in rock engineering by Goodman (1989).

With the assumption of simplified failure mechanisms, stability problems in geotechnical engineering will reduce to the determination of the most critical position of the failure surface. Though the shape of the failure surface chosen for the analysis may not be close to the actual situation, acceptable results are usually obtained from the LEM.

The popularity of LEM resides in: (i) simplicity in the solution which includes various loading condition, complex geometries, and soil profile and (ii) ease to code into general engineering software, and this method has been implemented into many commercial codes (Krahn 2004) which makes the technique attractive to the practitioners. LEM is statically indeterminate in nature; hence, it is necessary to make sufficient assumptions regarding the stress distribution along the failure surface or within the failure mass such that an overall equation of equilibrium, in terms of stress resultants, can be written for a given problem. This simplified approach makes it possible to solve various problems with reasonable results by simple statics.

Despite the wide acceptance and the accumulated experience in the applications of LEM, there remain several weaknesses inherited in this approach: (1) none of the equations of solid mechanics is explicitly satisfied everywhere inside or outside the slip surface; (2) equilibrium condition is satisfied only in a limited sense. Cheng et al. (2010) and Cheng et al. (2011a) have however further extended the works in LEM by using an equivalent variational principle or lower bound approach in which the internal forces are taken as control variables, and the most critical solution from the lower bound analysis will then be the rigorous solution of the problem. This approach is based on the concept of ultimate limit state under which the maximum strength of a system will be fully mobilized before failure. Using this extremum principle, the lower bound based approach by Cheng et al. (2010) and Cheng et al. (2011a) can overcome the two weaknesses as mentioned above at the expense of computer time.

1.1.2 Slip line method

Slip line method (SLM), also known as the characteristics method, provides an alternative to assess the ultimate limit state of soil mass. This method was developed in the beginning of 20th century in the theory of plasticity for metal. SLM combines the yield condition with the equilibrium conditions to give a set of differential equations of plastic equilibrium. Given the stress boundary condition, this set of differential equations can be used to investigate the stresses at the ultimate limit state. To solve specific problem, it is convenient to transform the plastic equilibrium equations to curvilinear coordinates where the direction at every point in the yielded region coincide with the direction of the failure or slip plane, which is why it is known as the slip line method (Hill 1950). The slip directions are called the slip lines, and the network of slip lines is called the slip-line field.

The solution of the slip line methods consists of constructing a slip-line field in a limited region that satisfies all the stress boundary conditions at the boundary points of the concerned region, as well as the equilibrium and yield condition at every point inside the region. As a result, the slip line method could be considered as an incomplete "lower bound" technique, as only part of the soil mass beneath a footing or behind a retaining wall is assumed to be in the stage of plastic equilibrium. The stress field so defined is called the partial stress field. There is however difficulty in extending the plastic stress field outside the slip network and satisfying equilibrium, yield condition, and boundary condition at the same time. When the velocity characteristics cannot be determined, which is often the case for many real problems, the corresponding stress characteristic do not represent a complete solution. More detailed discussion on the above mentioned traditional methods is presented in Chen (1975, 1990).

1.1.3 Complete numerical methods

With the development of finite element method and other numerical methods in the 1960s, sophisticated numerical methods have gained increasing acceptance and have been transforming from "a virtual dream to practical reality"(Potts 2003) over the past decades. A variety of numerical procedures are currently available for the solution of initial and boundary value problems arising in the field of geotechnical engineering. Finite element method, boundary element method, finite difference method, and the discrete element method are several among the most commonly used numerical approaches. Commercial software are also developed to ease the applications of the numerical procedures, e.g., ABAQUS, FLAC (Cundall and Board 1988), UDEC and 3-DEC(Lemos et al. 1985) and many others. Each numerical method and program is best suited for particular classes of problems. Strengths and weaknesses of the these methods are discussed in the work of Carter et al. (2000). At present, the use of the finite element method appears to be a popular option to many engineers for complicated problems. Examples in the application of the finite element method in geotechnical engineering can be found in the works of Yu et al. (1993), Potts (2003) , Potts and Zdravkovi (2001), Griffiths (1990), Li (2007), among others.

The most attractive aspects of the full numerical methods are the ability to simulate more realistic soil constitutive relationship and sequences of loadings by means of incremental analysis. This feature enables more realistic simulations of the soil behaviours to be performed, e.g. the hardening and softening behaviour with the consideration of progressive failure and the influence of the construction process etc.

In the application of the ultimate limit analysis, the full numerical methods are found attractive because in addition to the information on the stress state, the numerical methods give deformation (not available from the other three methods) which could serve as the alternative indicators instead of just the stresses in the stability analysis as well as assessment of the ultimate limit state. However, the collapse load is determined from the full numerical analysis by incremental analysis, in which the load-displacement response of the soil mass is traced and the limit load is identified by specifying a chosen criterion (Griffiths 1982; Griffiths and Lane 1999; Griffiths and Marquez 2008) which is different among different computer programs. Such a step-by-step incremental analysis requires a huge amount of computer cost, and it may take days or even weeks to produce a full load-displacement relation for large problems. Moreover, a full numerical analysis requires the specification of the complete constitutive relationship and initial state of stresses which is usually difficult and expensive to define.

1.1.4 Limit analysis

Limit analysis is concerned with the direct computation of the limit load. The underlying idea in limit analysis is to estimate the actual collapse load from two extremes by seeking the highest lower bound and lowest upper bound solutions under the framework of the bound theorems in the theory of plasticity (Drucker 1953; Drucker et al. 1952; Gvozdev 1960; Hill 1950). With the limit analysis, the lengthy elasto-plastic incremental analysis in the full numerical analysis is avoided. In comparison to other simplified methods, solutions by limit analysis are rigorous under the assumptions of (1) perfect plasticity, (2) associated flow rule and (3) small deformation. Similar to LEM, this method is not suitable to assess the deformation as well as the effect of progressive failure as the constitutive relation is not required in the analysis.

The theorem of lower bound (also known as the static principle of the limit analysis) states that any load calculated from a stress field that satisfies the equilibrium condition, boundary condition, and nowhere violates the yield condition would be less than or at most equal to the true collapse load. The stress field so defined is called the statically admissible stress field. On the other hand, the upper bound theorem (also known as the kinematic principle of the limit analysis) states that any load determined by equating the work done by external forces to the internal power dissipation corresponding to a particular kinematically admissible velocity field must be greater than or at least equal to the true collapse load. The kinematically admissible velocity field refers to the velocity field that satisfies the compatibility condition and the velocity boundary condition. With the two principles, the exact collapse load could be bounded from the top and bottom, and the quality of the estimates of the true collapse load is automatically obtained by evaluating the gap between the two bounds. As limit analysis views the problem at the moment of the collapse, the only required inputs are the strength parameters (e.g. cohesion and friction angle for MC material) which are similar to the LEM. This method is therefore easier and simpler to use in practice in comparison to the full numerical methods.

The upper bound approach with simplified failure mechanism has received much attention from many researchers. Works following this line of research include: Chen (1975), Chen and Liu (1990), Michalowski (1989), Donald and Chen (1997), Wang et al. (2001) and many others. Unlike the upper bound analysis, the rigorous lower bound analysis has been lagging behind due to the difficulties in the manual construction of the statically admissible stress field even for a simple problem, particularly when extending the stress field to a semi-infinite domain, which is usually the case in the geotechnical problems.

Pioneer works on the Finite-Element-based Limit Analysis (FELA) were done by Lysmer (1970), Bottero et al. (1980), Sloan and his co-works (Sloan 1988; Sloan 1989; Sloan and Booker 1986; Sloan and Kleeman 1995). These works are restricted to the linear formulation of optimization of the limit analysis. Recent developments by Lyamin and his co-workers (Lyamin 1999; Lyamin and Sloan 2000; Lyamin and Sloan 2002a; Lyamin and Sloan 2002b; Lyamin and Sloan 2003), Krabbenhoft et al. (Krabbenhoft and Damkilde 2003; Krabbenhoft et al. 2005; Krabbenhoft et al. 2007a; Krabbenhoft et al. 2007b; Krabbenhoft et al. 2008) make limit analysis attractive to both geotechnical scientists and practitioners. In the FELA, the stress and velocity fields are approximated with finite element function spaces and extension of the stress field to the semi-infinite domain is performed systematically by using the extension elements (Lyamin 1999). A key advantage of the FELA is that the complex loading, complicated geometries and a variety of yield conditions can be modelled without much difficulty as inherited from the FEM

However, two major difficulties are involved in the FELA. Firstly, the optimization problem arising from the FELA is usually large and poses a computational challenge to the computer capacity and optimization programming algorithms. Secondly, the accuracy of the limit solutions obtained from a finite element discretisation is sensitive (sometimes very sensitive) to the mesh configuration, and it is not an easy task to construct a good finite element mesh for a general problem where the approximate solution is totally unknown. Because of these two limitations, FELA is still mainly limited to research studies and is rarely carried out for practical problems.

1.1.5 Objectives and outline of the thesis

Given the recent developments in the limit analysis and convex optimization, objectives of this thesis are as follows:

(1) To implement the state-of-art lower bound and upper bound formulations of FELA using the nonlinear optimization techniques, particularly making use of the conic programming wherever appropriate and to develop a library code based on the objective-oriented programming language C++ that facilitates the programming with the FELA for stability analysis in geotechnical engineering. The library should be easy for the maintenance and extension and for large practical problems

(2) To develop efficient mesh adaptation strategies that steer the mesh refinement procedure in the FELA so that a good initial mesh is not necessary. A coarse mesh can be used as the initial input and the failure mechanism as part of the solution.

(3) To apply the formulation to study typical stability problems in geotechnical engineering. Interests are given to the performance of the adaptive procedures and the effects of the nonlinearity of the yield criteria on the stability of slopes and earth thrusts on retaining structure.

(4) Three-dimensional stability analyses are seldom considered in the past due to various difficulties. The proposed methodologies from the present study will be extended to three-dimensional considerations.

The outline of this thesis is as follows:

Chapter 2 provides a minimum introduction to the theory of convex optimization that underpins the discussion of limit analysis in subsequent chapters. The terminology and basic concepts of convex programming are covered. The fundamental concept of solution techniques are briefly discussed at the end of the chapter.

Chapter 3 is the literature review of limit analysis and the consideration of the yield criteria. The lower and upper bound formulations as well as the treatment of yield criteria in computation will be discussed.Assumptions of the limit analysis are clarified, validations of the application of limit analysis in geotechnical engineering are discussed and the proofs of the bound theorems are given.

Chapter 4 and Chapter 5 are devoted to the finite element formulations of lower bound and upper bound theorems respectively. The continuous lower bound and upper bound theorems are transformed into a discretised form and the resulting optimization problem is formulated as standard optimization problem with equality and inequality constraints

Chapter 6 targets at investigating efficient strategies for mesh adaptation to improve the solution qualities in the limit analysis. Mesh refinement and element flagging strategies are discussed and the differences due to these strategies are compared.

Chapter 7 discusses the considerations featured in geotechnical engineering and lights are shed on the considerations of presence of water, seismic loading, structure rigidity, and roughness of soil-structure interface, etc.

Chapter 8 presents the numerical illustrations of FELA applied to typical stability problems arising in geotechnical engineering. The performance of the mesh adaptation strategies and the influence of the nonlinearity of the yield criterion are considered.

Chapter 9 presents the conclusions and summarizes the findings. The suggested further works are also proposed and discussed.

1.2 Literature review

The establishment of the bound theorems is believed to be due to Drucker et al. (1951) and Drucker et al. (1952). As the limit theorems can be regarded as the special cases of the shake-down analysis that was developed for the elastic-plastic solid subjected to variable loading, it is fair to consider Melan (1938) as one of the originators of the bound theorems. A historical review of the early development of limit theorems can be found in the works by Hill (1950), Lyamin (1999), Bandini (2003), Davis and Selvadurai (2002), Yu (2006) and others.

1.2.1 Lower bound limit analysis

The lower bound method is a stress field approach for finding a bound solution to the actual limit load, i.e. the load is calculated from a statically admissible stress field which will be a lower bound and on the safe side. This concept is similar to the slip line theory, and the similarities between the two techniques lie in: (1) both methods address the collapse load directly without considering the deformation and (2) the information of the yield criterion is introduced in the formulation of the problem in the first place of the formulation. In this sense, it is reasonable to consider the slip line theory as the early development of the lower bound theorem.

Kotter (1903) is believed to be the first to derive the slip-line equations for the plane deformation. The first analytical closed form solution was obtained by Prandtl (1920) who developed the solution with a singular point with a pencil of straight slip-lines passing through it. These results were later applied by Reissner (1924) and Melan (1938) to the bearing capacity of footings on a weightless soil. Considering the self-weight of the soil would render the solution so complicated that a numerical procedure is required. Numerical solutions to the slip line equations were obtained by finite difference method by Sokolovskii (1960; 1965), who studied a number of interesting geotechnical problems such as the bearing capacity of footings and slopes as well as pressure of a fill on retaining walls. De Jong and Josselin (1957) adopted a different approach and developed a graphical procedure for the solutions of the slip line equations. Other forms of approximate solutions include the application of perturbation methods (Spencer 1961) and series expansion methods. More recently, numerical results considering seismic effects, axisymmetry etc. were given by Cheng (2003; 2005; 2007b) using slip-line method for bearing capacity and lateral earth pressure problems.

Numerical lower bound analysis of the FELA in soil mechanics appears to be first carried out by Lysmer (1970). In Lysmer's formulation, discretisation with three-node triangles is used and normal and shear stresses are taken as the optimal variables. To furnish the problem to be solvable by the linear programming method (LP), the Mohr-Coulomb yield criterion is linearised with an internal polyhedral approximation.

This early formulation of the FELA was later improved by Anderheggen and Knopfel (1972), Pastor and Turgeman (1982) and Bottero et al. (1980). Important modifications include the use of Cartesian stresses to replace the normal and shear stresses as the optimizing variables, the introduction of extension elements to account for the semi-infinite domain and others. Despite these improvements, applications of the numerical lower bound analysis remained limited due to the difficulty in the solution of the large-scale LP optimization. A significant development of the FELA as LP is due to Sloan (1988), who introduced active set algorithm for the solution of the resulting LP in the plane strain analysis. It was demonstrated that the active set algorithm was ideally suited for the optimization problems generated from the finite element formulation in the lower bound analysis.

To consider the axial limit analysis, Pastor and Turgeman (1982) proposed a lower bound formulation in conjunction with the linear finite element for the axisymmetric problems of the material of von Mises or Coulomb yield criterion. Recently, similar problem is re-studied by Khatri and Kumar (2009) who extended the FELA with linear programming based on the assumption that the magnitude of the hoop stress remains close to the least compressive stress σ_3. In this formulation, the Mohr Coulomb yield criterion as well as the formulation is linearised for the plane strain formulation by Sloan (1988). They proposed a formulation that only extra 3 constraints are required instead of 3 times the number of the sides of the polygon in formulation of Pastor and Turgeman (1982). With this formulation, the bearing capacity of circular footing on purely cohesive soils (Khatri and Kumar 2009) and for c-phi soils (Kumar and Khatri 2011) have been studied respectively.

LP formulation of the FELA is simple in the concept and generally efficient in plane strain and axisymmetric analysis. However, extension of these formulations to three-dimensional analysis is almost impractical as the linearzation of the nonlinear yield criteria in the full three-dimensional analysis leads to a huge number of constraints. In order to treat the yield criteria as their natural form, Lyamin (1999) developed a lower bound formulation of the FELA based on the NLP for the MC material for both 2D and 3D analyses. The resulting optimization problems are then solved with a quasi-Newton algorithm that was originally proposed by Zouain et al. (1993). It is shown by comparison studies (Lyamin 1999) that the NLP formulation with a quasi-Newton algorithm is much more efficient and stable than its LP analogue as in Sloan (1988).

When applying the NLP in the FELA, yield criteria are required to be smooth such that gradients with respect to the Cartesian stresses are able to be computed everywhere. The widely accepted yield criteria for geomaterials such as the MC yield criterion and HB yield criterion are notorious for the singularities at the apex and corners, and thus smoothing is required in the application of the optimization algorithm. Lyamin (1999) applied a hyperbolic approximation in the meridional plane, and a round-off technique in the deviatoric plane for three dimensional analysis was first proposed by Sloan and Booker (1986). Three-dimensional problems in the FELA with the MC material have been solved satisfactorily. A similar approach was applied to smooth the HB yield criterion in the limit analysis of rock masses by Merifield et al. (2006) for the plane strain analysis.

It has recently been realized that a class of yield criteria in geomaterials, including the MC yield criterion, could be cast into conic constraints and consequently the FELA could be formulated in the form of conic programming. This concepts had been studied for the von Mises material with lots of studies (Andersen et al. 2000; Ciria 2002). The implication of this manipulation of yield criteria is that the singularities presents in the MC yield criterion will pose no difficulties in the algorithm specialized for the conic programming. Makrodimopoulos and Martin (2006) proposed a conic form of lower bound formulation of the FELA for plane strain analysis for the MC material. The resulting optimization problems are efficiently solved by a primal-dual interior point algorithm with the optimization package MOSEK-Aps (2010), which has been proven to be more robust and efficient than the NLP counterpart. In addition to the numerical benefits of the conic formulation, the yield criteria in a conic form will be treated as its natural form rather than a smoothed approximation in the NLP formulation. Since these works, the solution of plane strain problems of the MC material is much more efficient than they were used to be.

Later, Krabbenhoft et al. (2008) and Martin and Makrodimopoulos (2008) independently proposed the Positive Semidefinite Programming (SDP) formulation of three dimensional limit analysis of the MC material, which is inspired by the fact that the MC yield criterion can naturally be expressed in terms of the linear combination of the principal stresses (eigenvalues of the symmetric stress tensor). Moderate 3D problems with several thousand of tetrahedral elements are solved with encouraging efficiency by algorithms specialized for SDP such as Sedumi (Strum 1999). Currently, no large-scale model analysis has been reported and there are also some convergence problems encountered in the analysis. With the increase in the model size, the optimization problems in 3D would grow appreciably. In spite of the shortcomings of the SDP formulation, it is still attractive in three-dimensional limit analysis as it is a relatively new field of optimization and has numerous applications in disciplines such as engineering, mathematics, computer science, operational research, etc. More powerful code and algorithm will definitely arise in the future to overcome the mathematical difficulties of the current SDP.

Recently, Kammoun et al. (2010) presented a decomposition technique for the lower bound in the FELA. The problem is partitioned into finite element sub problems, and with an auxiliary interface problem, very large-scale problem with millions of variables and constraints are solved using the interior point algorithm. The best lower bound solution $N_c = 3.7752$ is obtained with this technique for the vertical cut of purely cohesive material, and this value is commonly taken as the benchmark for this type of problem.

In the numerical formulation of the lower bound theorem, discretisation using the finite element space is not the only choice. Recently, Chen et al. (2008) presented a lower bound formulation based on the element-free Galerkin (FEG) method. The collapse load is computed iteratively by solving a sequence of sub-problems generated by a reduced-basis technique, and the Complex method is adapted to solve the resulting nonlinear programming problem. Similarly, Le et al. (Le et al. 2010a; Le et al. 2010b) presented a similar lower bound formulation based on element-free Galerkin method for the plates and slabs of von Mises and Nielsen material. The stress/moment field is constructed by using a moving least-square approximation. The main limitation to this approach is the application of the method to nonhomogeneous problems which are commonly found for many geotechnical problems.

1.2.2 Upper bound limit analysis

Generating a kinematically admissible velocity field is comparatively easier than generating a statically admissible stress field. Therefore, in the literature, there are a number of variants of the applications of upper bound approaches for geotechnical problems. Each variant differs from the others primarily in the approximation of the velocity field.

1.2.2.1 **Rigid blocks /Multi-blocks /rigid finite element based upper bound analysis**

A comprehensive study of the conventional upper bound limit analysis in soil mechanics is presented in the work by Chen (1975), in which simplified velocity fields consisting of rigid blocks sliding along assumed failure surface are considered. The collapse load is obtained by equating the external work done to the internal power dissipation that is assumed to occur along the slip surface. Results obtained with the simplified upper bound are found with high accuracy for simple stability problems of homogeneous soil condition if the slip surfaces are carefully chosen. Later, Chen and Liu (1990) extended the technique to take into account of seismic loading and anisotropy soil properties.

Inspired by the slicing techniques in the limit equilibrium method, Donald and Chen (1997) introduced the so called “multi-block” technique in which the sliding soil mass (traditionally treated as a whole rigid block in Chen's work (Chen 1975; Chen and Liu 1990)) is divided into a number of smaller blocks. Velocity discontinuities are permitted between the linear interfaces between those smaller blocks. The collapse load is sought similarly from work balance equation. Following this line of research, Wang (2001) extended the multi-block approach to three-dimensional analysis and applied it to the collapse load of material obeying non-associated flow rule. Michalowski (2001) proposed a three-dimensional failure mechanism using blocks with conical surfaces in the determination of the bearing capacity of square and circular footing. These methods suffer from the inherent limitation as in the conventional upper bound analysis in that the failure mechanism needs to be predefined. To overcome this difficulty, Chen (2004) has employed the finite element discretisation for the generation of the blocks in which each element is viewed as a rigid block. Velocity discontinuities are allowed between the blocks and energy is dissipated only along the velocity discontinuities between two adjoining elements. This general version of the multi-block technique is named as the "rigid-finite-element-based upper bound analysis". The resulting problem in their formulation is then solved by SQP method. (Chen et al. 2005). Using the finite element discretisation for the generation of velocity field is more robust and general; however, the accuracy of the solution obtained is sensitive to the mesh used for the analysis.

To investigate the nonlinearity of the yield criteria in the deviatoric plane, the conventional approach of upper bound analysis was extended by Zhang and Chen (1987) to stability problems with material obeying a general nonlinear yield criterion. In their work, the nonlinearity of the yield envelop in the Mohr plane was analysed with the variational method. Yang et al. (2003b) made a modification to the preceding work and applied a “general tangential line” method which replaced the nonlinear yield criterion with a tangential line associated with a particular point where the position is to be optimized together in the optimization. With this method, a number of geotechnical stability problems are studied (Yang 2007; Yang et al. 2007; Yang and Huang 2009; Yang and Yin 2004; Yang and Yin 2005; Yang et al. 2003b). Similar ideas can also be found in the works by Collins et al. (1988) .

It has long been realized that the limit equilibrium method can, to some extent, be regarded as an upper bound approach. By applying the virtual work principle with the virtual displacements (velocities) that are in consistence with the associated flow rule, the force equilibrium or moment equilibrium equation could be transformed to an equivalent work balance equation. Therefore, the force equilibrium and the rigid block upper bound method could be considered to be equivalent to each other, and the equivalence is proved by Michalowski (1989) and later by Drescher and Detournay (1993), Davis and Selvadurai (2002) and Yu (2006), among others. The most insightful idea from this equivalence is that the upper bound analysis could be formulated with stress (force) variables (as what have been done for LEM) instead of velocity field (rigid block technique). In the conventional upper bound analysis, the power dissipation is expressed in terms of the kinematic variables and the collapse load is obtained from the work balance equation, so it might be misleading that the upper bound solution is a velocity field solution. It is not until quite recently that Krabbenhoft et al. (2005) provided a finite-element-based upper bound formulation in pure stress variables to clarify this situation, and more general discussion on this issue from a variational point of view regarding the “dual formulation” of the upper bound is given in Christiansen (1980), Ciria (2002) and Ciria et al. (2008). It is shown in these works that the dual formulation, i.e.

force or stress formulation of the upper bound analysis is more efficient with the primal-dual solution technique because the duality between the lower and upper bound is explicitly exploited. It is reasonable to believe that the rigid finite element method (Chen 2004; Chen et al. 2004; Chen et al. 2005) might have been more efficient if it is formulated in terms of stresses (as a simple generation of the LEM).

1.2.2.2 **Finite-element-based upper bound limit analysis**

Upper bound formulation of the FELA is pioneered by Anderheggen and Knopfel (1972) and Bottero et al. (1980). In their formulation, three-node triangular elements are utilized to discretize the velocity field and kinematically admissible discontinuities are allowed between two adjacent elements to compensate for the low order of velocity interpolation. To render the resulting optimization problem to be linear programming problem that could be solved by linear programming algorithm, the yield criterion is linearized as an external polygon to ensure the solution to be an upper bound. Shortcomings in these early formulations are: (1) the revised simplex algorithm which is used is generally slow in computation; and (2) the direction of the shearing are required to be specified for each discontinuities a *priori*, making the large number of arbitrary discontinuities impossible. To tackle the first problem, Sloan (1989) improved the solution technique by solving the dual problem of the upper bound formulation with an active set algorithm that is originally developed in his lower bound formulation (Sloan 1988). To address the second shortcoming of the pioneering formulation, Sloan and Kleeman (1995) introduced a method to automatically determine the direction of shearing by describing the velocity jump with an additional set of four unknowns. The amended formulation due to Sloan and

Kleeman (1995) has been proven to be efficient and robust and has been studied in details by Kim et al. (2002) and Bandini (2003) with applications in geotechnical engineering.

Noting the large number of the linear constraints due to the linearization of the yield criterion in the upper bound analysis, Lyamin and Sloan (2002b) proposed general nonlinear formulation in which simplex element is used for the discretisation of the velocity field and each element is associated with constant stresses. A two-stage quasi-Newton algorithm is used to solve the KKT conditions of the optimization problem arising from the discretised upper bound problem. In their formulation, examples in both two-dimensional (2D) and three-dimensional (3D) conditions are presented to demonstrate the superiority of the NLP formulation over the LP analogue. Li and Yu (2005) proposed an NLP formulation for the MC and DP yield criteria using purely the kinematic variables. The MC and DP yield criterion are rewritten in a quadratic form with which the stresses variables are eliminated from the expression of the rate of the power dissipation by applying the normality condition. This formulation is conceptually simpler than the NLP formulation proposed by Lyamin and Sloan (2002b), however, one difficulty of this formulation is that the objective function obtained is not everywhere smooth and is nondifferentiable in the rigid area such that an iterative algorithm based on distinguishing rigid/plastic area was adopted in finding the solution. The algorithm that was originally proposed by Burland et al. (1977b) and Huh and Yang (1991) was adopted in finding the solution.

In finite element formulation of the upper bound analysis, higher order element in addition to the use of constant strain elements has been studied. Linear strain elements with straight edges have been be tailored for the upper bound discretisation formulation by Yu et al.

(1994) and Pastor et al. (2002) for plane strain and axisymmetric analysis respectively. More recently, following similar idea, Makrodimopoulos and Martin (2007) adopted the discretisation using simplex elements for the upper bound discretisation and formulated the upper bound analysis as standard SCOP for the MC material under plane strain condition and Drucker-Prager material under full three-dimensional condition. The optimization problem resulting from this formulation is then solved with primal-dual interior point algorithm specialized for the SOCP.

The significance of the conic formulation of the FELA with the MC material is that the expression of the power dissipation can be derived easily and the singularities of the MC yield criterion will not pose any difficulty at the solution stage. Later, Makrodimopoulos and Martin (2008) extended their formulation to include velocity discontinuities in the discretisation of the velocity field, and the velocity jump is constrained to vary linearly by introducing additional constraints. It has been shown that the introduction of the velocity discontinuities will dramatically increase the size of the resulting problem. However, if the discontinuities are arrange with the knowledge of the slip bands, very impressive results could be obtained.

Formulating the upper bound analysis in velocities require the expression of the rate of power dissipation such that the stress field is not explicitly included in the analysis. However, to express the power dissipation in terms of the velocity variables will become complicated for general yield criterion. A new formulation of the upper bound theorem in terms of the stress variables was proposed by Krabbenhoft et al. (2005), and similar ideas can be found in the works by Ciria (2002). The upper bound theorem is formulated directly from the dual form, i.e. the stresses are used as the optimization variables and the upper bound solution is obtained by maximising the load multiplier instead of minimising the load as in the conventional upper bound analysis. By the duality theory in convex programming, it could be guaranteed that the formulation yields the rigorous upper bound.

More recently, da Silva and Antao (2008) proposed a parallel mixed finite element upper bound, where the velocity and strain field were independently defined and compatibility condition is imposed by augmented Lagrange method. The upper bound FELA has been applied to masonry works by Cavicchi and Gambarotta (2006) and Milani (2008; 2011; 2007; 2010) using the LP formulation.

In recent years, attempts have also been made to seek alternatives of discretisation other than finite element space. For example, Smith and Gilbert (2007) proposed a discontinuity layout optimization procedure to determine the discontinuity directly with interesting and promising applications. Le et al. (2010a) and Le et al. (2010b) presented a meshfree approach to formulate the upper bound theorem where the yield at the interior of the solution may be violated in some cases. In the present research, discussion will be restricted to finite element based limit analysis.

1.2.3 Mesh adaptation in limit analysis

Since the 1970s, error estimators have been developed to assess the discretisation error of the finite element solutions. Posterior error estimators in the literature of the FEM can generally be classified into two groups, namely, residual-based error estimators and the recovery-based error estimators (post processing estimate).

The idea of the residual error is to compute the residual of the equilibrium within an element and jumps at element boundaries to obtain an estimate of the error in the energy norm. Errors are estimated in a sense of how much the discretisation of the continuous field has failed to satisfy the equilibrium condition and the boundary condition for a particular element. Fundamental works regarding the residual error estimate are presented in the works by Babuska and Miller (1978), Babuska and Rheinboldt (1979), Kelly et al. (1983), Babuska and Miller (1987).

Recovery-based error estimate defines the error by evaluating the smoothed derivatives with the original ones inspired by the smoothing procedure to recover more accurate nodal derivatives in the finite element analysis (Zienkiewicz and Zhu 1992a). A good review on the developments of the error estimate in the FEM is presented in the works by Gratsch and Bathe (2005), and more mathematical details on the error estimate are discussed in the works by Brenner and Scott (2008).

Mesh adaptation techniques in the FELA are primarily the extension of the existing procedures in the FEM. Borges et al. (2001) extended a recovery-based error estimator (Borges et al. 1999) to local directional interpolation error in mixed limit analysis formulation which is based on the recovering scheme to compute the second derivatives of the finite element solution. The scalar field of the Lagrangian multiplier is used as the control variables.

The global re-meshing technique by Lyamin et al. (2005) and Sloan et al. (2008) tailored the error estimator (Borges et al. 1999) and applied it to the lower bound formulation of the limit analysis. The advancing front mesh generator with the adaptive mesh refinement has also been studied, and it is shown that when the scheme is coupled with the automatic fan zone general, results with high accuracy can be produced. It is also demonstrated by Lyamin et al. (2005) that the anisotropy of the mesh adaptation does not seem to provide better lower bounds for the unstructured mesh consisting of 3-noded triangles. Both of these methods can be regarded as an extension of the recovery-based error estimate scheme, which implicitly assumes that the smoothness of the solution is equivalent to the accuracy in solution.

Christiansen and Edmund (2001) proposed a mesh adaptation procedure for the von Mises material, which is based on the localized mesh refinement on the unstructured triangular mesh. The mesh update is steered based on the yield slack and equivalent deformation. A computationally economy procedure is tested on plane strain problems with very accurate solutions obtained. The idea of guiding the degree of freedom concentration in the mesh adaptation is simple but adequate in the context of the limit analysis as the FELA generally assumes the rigid plasticity condition, which implies that most of the regions are actually rigid bodies.

An essentially different error estimator was proposed by Ciria et al. (2008). In his work, the total gap between the lower and upper bounds is decomposed into the sum of elemental contributions and elements are marked for refinement based on their contributions to the global error. This procedure has been proven to be successful in applying to 2D analysis. As an extension of the approach by Ciria (Ciria 2002; Ciria et al. 2008), Munoz et al. (2009) further considered the error contribution from internal edges.

CHAPTER 2: MATHEMATICAL PROGRAMMING

2.1 Introduction

Mathematical programming (also known as mathematical optimization) underpins the bound theorems in the theory of the plasticity and the solution techniques for resulting optimization problem. In the development of the limit analysis, the majority of the concepts stem from the theory of convex optimization, particularly in the last three decades when significant advancements have been achieved in this area. This chapter is targeted towards a basic introduction that suffices to clarify the fundamental concepts and terminology in convex programming that will be referred in the later part of this thesis.

2.2 Optimization problems

A physical system tends to attain a minimum energy state, which corresponds to the principle of minimum potential energy. On the other hand, for deformable solid undergoing plastic flows, the power dissipation tends to maximise in order to prevent the failure. Many engineering problem are the results of such optimization processes.

To apply the optimization technique, it is important to identify the quantitative measure of a system or the quantity of interest, i.e., the objective of the analysis. The objective depends on certain characteristics of the system called variables, and usually the variables are restricted or constrained to some extent, e.g., the stress point for a rigid perfectly plastic material cannot lie outside the yield surface and the compatibility condition needs to be satisfied by a continuum requirement such that voids or overlapping will not occur during deformation. Unconstrained variables seldom occur in actual practice due to the physical constraints in a problem.

Mathematically, the standard mathematical programming problem takes the form as below (following the notation of Boyd and Vandenberghe (2004)),

$$\begin{aligned} &\text{minimize} \quad f_0(\mathbf{x}) \\ &\text{subject to} \quad \begin{cases} f_i(\mathbf{x}) \leq 0 \ \ \mathrm{i} = 1, \dots, m \\ \\ h_i(\mathbf{x}) = 0 \ \ \mathrm{i} = 1, \dots, p \end{cases} \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in \mathbf{R}^n$ = optimization variables, $f_0 \colon \mathbf{R}^n \rightarrow \mathbf{R}$ = the objective functions, $f_i \colon \mathbf{R}^n \rightarrow \mathbf{R}$ = the inequality constraints and $h_i \colon \mathbf{R}^n \rightarrow \mathbf{R}$ = the equality constraints. (2.1) defines an optimization problem in finding an optimal point $\mathbf{x}$ that minimizes $f(\mathbf{x})$ and meanwhile satisfies the conditions $f_i(\mathbf{x}) \leq 0$ and $h_i(\mathbf{x}) = 0$.

The set of points that satisfy all constraints is called the domain of the problem that is defined by

$$\mathcal{D} = \bigcap_{i=1}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i \tag{2.2}$$

where **dom** is the short name for domain, and $\mathcal{D}$ = domain of optimization problem (2.1) . A point $\mathbf{x} \in \mathcal{D}$ is called feasible if it satisfies the constraints (2.3).

$$f_i(\mathbf{x}) \le 0 \;\; i = 1, \dots, m$$
$$h_i(\mathbf{x}) = 0 \;\; i = 1, \dots, p \tag{2.3}$$

An optimization problem is called feasible if there exists at least one feasible point in the domain and infeasible otherwise. The set of all feasible points is called the feasible set or the constraint set.

The optimal value $\mathbf{p}^*$ of (2.1) is defined as

$$\mathbf{p}^* = \inf \{f_0(\mathbf{x}) | f_i(\mathbf{x}) \le 0, i = 1, \dots, m, h_i(\mathbf{x}) = 0, i = 1, \dots, p\} \tag{2.4}$$

If the problem is infeasible, $p^* = \infty$ (following the standard convention that infimum of the empty set is ∞). If there are feasible points $\mathbf{x}_k$with $f_0(\mathbf{x}_k) \to -\infty$ as $k \to -\infty$, then $p^* \to -\infty$, and in this case, the problem is said to be unbounded below.

2.3 Convex optimization problems

Most of the robust algorithms seek only a local solution, a point at which the objective function is less than all other feasible points in its vicinity. It is not guaranteed that the solution is the best of all the minima, i.e. the global solution. The concept of local minimum and global minimum must be clearly differentiated, as these two minima may not be equal for general cases. An important subclass of optimization problem is the convex programming, in which all local solutions are automatically global solutions.

By definition, convex programming minimizes convex function over convex set. Throughout this thesis, the discussion will be restricted to convex programming by imposing assumptions on the material behaviour. This issue will be addressed in subsequent chapters.

2.3.1 Definition of Convexity

In (2.1), no constraints have been imposed on the objective functions and constraints. To ensure that an optimization problem is convex, a few additional requirements are necessary to be satisfied.

The word convex means curving out or bulging outward (Figure 2.1). A convex set is defined as that any straight-line segment connecting an arbitrary pair of points within the set again lies entirely within the object.

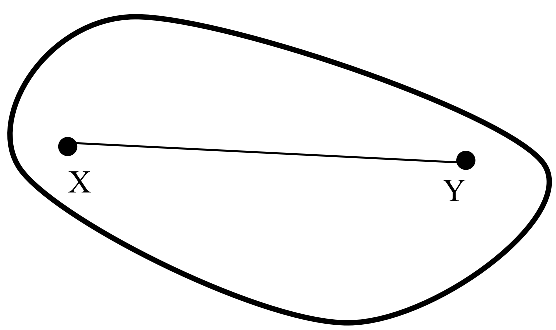


Figure 2.1 A convex set

For functions to be convex, they need to satisfy

$$f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \tag{2.5}$$

for all $\mathbf{x}, \mathbf{y} \in R^n$ and all $\alpha, \beta \in R$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

A mathematical programming is called a convex programming problem if it satisfies the following conditions:

(a) Objective function must be convex,

(b) The inequality constraints must be convex,

(c) The equality constraints must be affine, which mean that the equality constraints need to be stronger than just convex.

Therefore, a convex optimization problem takes the form

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & \begin{cases} f_i(\mathbf{x}) \leq 0 \;\; i = 1, \dots, m \\ \\ \mathbf{Ax} = \mathbf{b} \end{cases} \end{array} \tag{2.6}$$

Where f_i= convex function $\forall i = 1, \dots, m$, and $\mathbf{A} \in \mathbf{R}^{p \times n}$

2.3.2 Linear programming

If the objective function and constraint functions of (2.1) are all affine (or linear in a less rigorous sense), the problem is called linear programming (LP) problem. A general LP problem takes the form

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{d} \\ \text{subject to} & \begin{cases} \mathbf{Gx} \leq \mathbf{h} \\ \\ \mathbf{Ax} = \mathbf{b} \end{cases} \end{array} \tag{2.7}$$

Where $\mathbf{G} \in \mathbf{R}^{m \times n}$, $\mathbf{A} \in \mathbf{R}^{p \times n}$

2.3.3 Conic programming

If the constraints consists of cones (defined in eq.(2.8)) and affine constraints, the optimization problem is called conic programming. Conic programming has found wide and growing applications in a variety of fields such as control, economics and physics, and it is the most important generalization of the linear programming in the past three decades, particularly for the Second Order Cone Programming (SOCP) and the positive Sefinite Programming (SDP). Conic programming is concerned with finding the minimum of a convex function over the intersection of affine half space and cones. To clarify the concept, let us define the cone first.

2.3.3.1 **Cones**

A cone is defined as a point set. If $\forall \mathbf{x} \in \mathcal{C}$ and $\theta \geq 0$, we have $\theta \mathbf{x} \in \mathcal{C}$. A set $\mathcal{C}$ is a convex and a cone, which means that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ and $\theta_1, \theta_2 \geq 0$ we have

$$\theta \mathbf{x}_1 + \theta \mathbf{x}_2 \in \mathcal{C} \tag{2.8}$$

2.3.3.2 **Dual cone**

Let $\mathcal{C}$ be a cone, then the dual cone for $\mathcal{C}$ is defined as

$$\mathcal{C}^* = \{\mathbf{y} | \mathbf{x}^T \mathbf{y} \geq 0 \ \forall \mathbf{x} \in \mathcal{C}\} \tag{2.9}$$

From (2.9), the dual cone ensures a nonnegative product of the constraints and variables, which will be important in deriving the dual problem of a conic programming. If $\mathcal{C}^* = \mathcal{C}$, then the cone $\mathcal{C}$ is called self-dual. The second order cone and positive semi-definite cone as will be presented are both self-dual.

2.3.3.3 **Second Order Cones (SOC)**

Sets $\mathcal{C}^2$ are second order cones, also known as quadratic cones if they take the following form

$$\mathcal{C}^2 = \left\{\mathbf{x} \in \mathbf{R}^n : \left\|\mathbf{x}_{\mathbf{2:n}}\right\| \leq \mathbf{x}_1, \mathbf{x}_1 \geq 0\right\} \tag{2.10}$$

where $\mathbf{x}_{\mathbf{2:n}} = [x_2 \ldots x_n]^T$, $\|\cdot\|$ denotes the Euclidean norm.

2.3.3.4 **Positive Semidefinite Cones and SDP**

A positive semi-definite matrices set is defined as

$$\mathcal{S}_+^{(\mathrm{n})} = \left\{\mathbf{X} \in \mathcal{R}^{\mathrm{n}\times\mathrm{n}} \middle| \mathbf{X} \succcurlyeq 0\,, \mathbf{X}_{ij} = \mathbf{X}_{ji}\right\} \tag{2.11}$$

where $\succcurlyeq$ denotes that a matrix is positive semidefinite.

Following the similar definite as a point cone, $\mathbf{X}_1, \mathbf{X}_2 \in \mathcal{S}_+^{(\mathrm{n})}$ and $\theta_1, \theta_2 \geq 0$ we have

$$\theta_1 \mathbf{X}_1 + \theta_2 \mathbf{X}_2 \succcurlyeq 0 \tag{2.12}$$

A semidefinite program (SDP) has the form of (2.13).

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \begin{cases} x_1 \mathbf{F}_1 + \cdots + x_n \mathbf{F}_n + \mathbf{G} \preccurlyeq 0 \\ \\ \mathbf{Ax} = \mathbf{b} \end{cases} \end{array} \tag{2.13}$$

where $G, F_1, \ldots, F_n \in \mathcal{S}^{(k)}$, and $A \in \mathbf{R}^{p\times n}$**.** It can be seen when $G, F_1, \ldots, F_n$ are diagonal matrices, (2.13) will reduce to the linear programming form.

2.4 Duality

In the constrained optimization, it is possible to convert the original problem (primal) to a dual problem by making use of the Lagrangian. This concept is of great importance in the theory of the optimization. Let us define the Lagrangian $\mathcal{L} \coloneqq \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ associated with problem (2.1) as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{v}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \tag{2.14}$$

where, $\lambda_i \geq 0$ is the Lagrangian multiplier corresponding to i-th inequality $f_i(\mathbf{x}) \leq \mathbf{0}$, ν_i is the Lagrange multiplier corresponding to i-th equality constraints $h_i(\mathbf{x}) = 0$. The vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are called dual variables or Lagrange multiplier vectors associated with (2.1).

Now we introduce the concepts of Lagrange dual function (or simply dual function) $\mathrm{g} \coloneqq \mathbf{R}^{\mathrm{m}} \times \mathbf{R}^p \to \mathbf{R}$ as the minimum of the Lagrangian $\mathcal{L}$ over $\mathbf{x}$: of r $\boldsymbol{\lambda} \in \mathrm{R}^m, \mathbf{v} \in R^p$,

$$g(\boldsymbol{\lambda}, \mathbf{v}) = \inf_{\mathbf{x} \in D} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \right) \tag{2.15}$$

The dual function (2.15) yields a lower bound on the optimal value p^* of the problem (2.1). For any $\lambda_i > 0$ and any ν_i, we have

$$g(\boldsymbol{\lambda}, \mathbf{v}) \leq p^* \tag{2.16}$$

It is not difficult to prove the property of (2.16) . Suppose $\tilde{\boldsymbol{x}}$ is a feasible point (i.e. $f_i(\tilde{\mathbf{x}}) < 0 \; and \; h_i(\tilde{\mathbf{x}}) = 0$) and $\boldsymbol{\lambda} > 0$, then we have

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^{p} v_i h_i(\tilde{\mathbf{x}}) \leq 0 \tag{2.17}$$

Since each term in the first sum of (2.17) is non-positive, and each term in the second sum is zero, therefore

$$L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}, \mathbf{v}) = f_0(\tilde{\mathbf{x}}) + \sum_{(i=1)}^{m} \lambda_i f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^{p} v_i h_i(\tilde{\mathbf{x}}) \leq f_0(\tilde{\mathbf{x}}) \tag{2.18}$$

Hence

$$g(\boldsymbol{\lambda}, \mathbf{v}) = \inf_{x \in D} L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{v}) \leq L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}, \mathbf{v}) \leq f_0(\tilde{\mathbf{x}}) \tag{2.19}$$

Thus the dual problem to (2.1) is in the form

$$\begin{array}{ll} \text{maximize} & g(\boldsymbol{\lambda}, \mathbf{v}) \\ \text{subject to} & \boldsymbol{\lambda} \geq 0 \end{array} \tag{2.20}$$

Let d^* denote the solution of (2.20) and according to (2.16) we have

$$d^* \leq p^* \tag{2.21}$$

When the equal sign of (2.21) occurs, the strong duality holds for optimization problem, i.e. the solution of the problem can be found either from (2.20) or from (2.1). It should be noted that the lower bound and upper bound theorems in the limit analysis is primal and dual to each other and the strong duality holds (see Christiansen (1996) for details).

2.5 Solution techniques for optimization problems

This section will cover basic concepts in the solution of the optimizing problems using the nonlinear programming method. The solution procedure will only be outlined here, as the procedures are generally complicated and lengthy for a robust algorithm.

2.5.1 KKT condition

For the unconstrained optimization, it is well known that the optimization condition is that the gradient with respect to the optimizing variables vanish at the optimal point. In the case of optimization with constraints, the optimal value of a convex programming must satisfy the following conditions, which are known as KKT condition (Kuhn and Tucker 1951).

(1) Primal constraints $f_i(x^*) \leq 0, i = 1, \ldots, m, h_i(x^*) = 0, i = 1 \ldots, m$;

(2) Dual constraints $\lambda_i^* \geqslant 0$;

(3) The complementary slackness $\lambda_i^* f_i(x^*) = 0$;

(4) The gradient of the Lagrangian with respect to x^* vanishes, i.e.

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} v_i^* \nabla h_i(x^*) = 0 \tag{2.22}$$

where x^*, λ^*, ν^* are optimal values of primal and dual variable.

Solution of the original problem is equivalent to the solving system of equations arising from the KKT conditions. To illustrate this, let us consider a quadratic optimization problem *w*ith only equality constraints.

$$\begin{aligned} &\text{minimize} \quad && \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^{\boldsymbol{T}}\mathbf{x} + \mathbf{r} \\ &\text{subject to} \quad && \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{2.23}$$

where $\mathbf{Q} \in \mathcal{S}_+^{(\mathrm{n})}$, then the KKT condition reads

$$\mathbf{A}\mathbf{x}^* = \mathbf{b}; \mathbf{Q}\mathbf{x} + \mathbf{q} + \mathbf{A}^T\mathbf{v}^* = \mathbf{0} \tag{2.24}$$

or in matrix form

$$\begin{bmatrix} \mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{v}^* \end{bmatrix} = \begin{bmatrix} -\mathbf{q} \\ \mathbf{b} \end{bmatrix} \tag{2.25}$$

Finding the optimal value of (2.23) is equivalent to the solution of the linear system (2.25).

The system of equations similar to the form in (2.25) will appear frequently in solution algorithms of a nonlinear programming problem, mainly because Newton method is equivalent to solving a quadratic Taylor expansion at each iteration x^k.

Solution of an optimization problem is to produce a sequence of point $x^{(k)} \in dom\ f, k = 1, \ldots$ with

$$f(x^k) \to p^*$$

Two of the most commonly used methods are described in the following sections.

2.5.2 SQP method

Sequential Quadratic Programming (SQP) defines the search direction d^k as the solution of quadratic sub-problem at the current iterate x^k.

$$\begin{aligned} &\text{minimize} && \mathcal{L} + (\nabla_x \mathcal{L})^T \cdot d^k + \frac{1}{2} d^T (\nabla_{xx}^2 \mathcal{L}) d \\ &\text{subject to} && \begin{cases} \mathbf{f}(x^k) + \nabla \mathbf{f}^T \cdot d^k \le \mathbf{0} \\ \mathbf{A}x^k + A^T \cdot d^k = \mathbf{b} \end{cases} \end{aligned} \tag{2.26}$$

2.5.3 Interior point method

The underlying idea of introducing the logarithmic barrier function is to reduce the inequality constrained problem to the equality constrained problem by incorporating the inequality constraints in the objective function as

$$\begin{aligned} &\text{minimize} && f_0(\mathbf{x}) + \sum_{i=1}^{m} -\left(\frac{1}{t}\right)\log\bigl(-f_i(\mathbf{x})\bigr) \\ &\text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{2.27}$$

where $\sum_{i=1}^{m} -\left(\frac{1}{t}\right)\log\bigl(-f_i(x)\bigr)$ which is denoted as $\phi(x)$ is called the barrier function, and t is a positive number called the "barrier parameter".

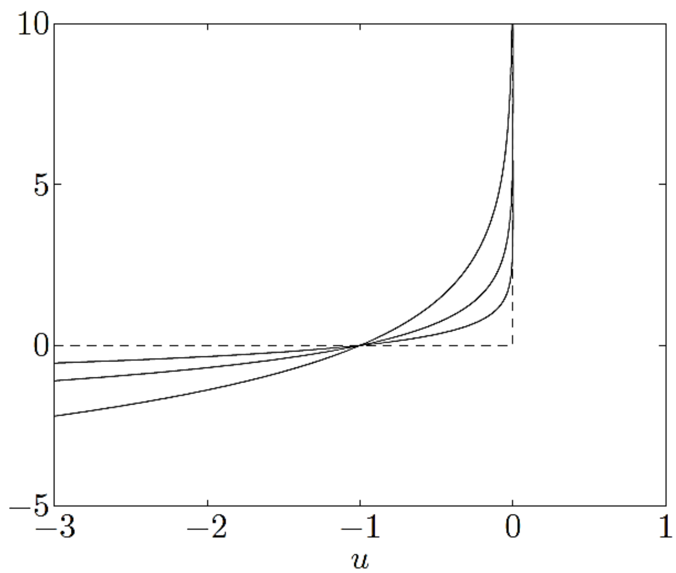


Figure 2.2 Barrier functions corresponding to different values of t

It could be observed that when $f_i(\mathbf{x})$ approaches 0, the value of the log barrier function tends to infinity and minimization of the problems ensures the feasibility contributed by the inequalities. It is easy to verify that the solution of (2.27) approach the original formulation (2.6) as t grows to a large number. The optimization problem as given by (2.27) can be rewritten as follows

$$\begin{aligned} &\text{minimize} \quad && t\, f_0(\mathbf{x}) + \phi(\mathbf{x}) \\ &\text{subject to} \quad && \mathbf{Ax} = \mathbf{b} \end{aligned} \tag{2.28}$$

Consider a LP for example,

$$\begin{aligned} &\text{minimize} \quad && \mathbf{c}^T\mathbf{x} \\ &\text{subject to} \quad && \mathbf{a}_i^T\mathbf{x} \leq \mathbf{b}_i \end{aligned} \tag{2.29}$$

For $n = 2, m = 6$ as shown in Figure 2.3, when $t = 0$ the optimal stay at the analytical center of the domain, as $t \to \infty$, the optimal value $\mathbf{x}^*(t)$ approach $\mathbf{x}^*$. When the optimimal value hits the wall of the doamin, the objective function will go infinity, therefore, by minimization, all points generated by various t will be confined in the interior of the feasible domain, which is the reason for the name interior point method.

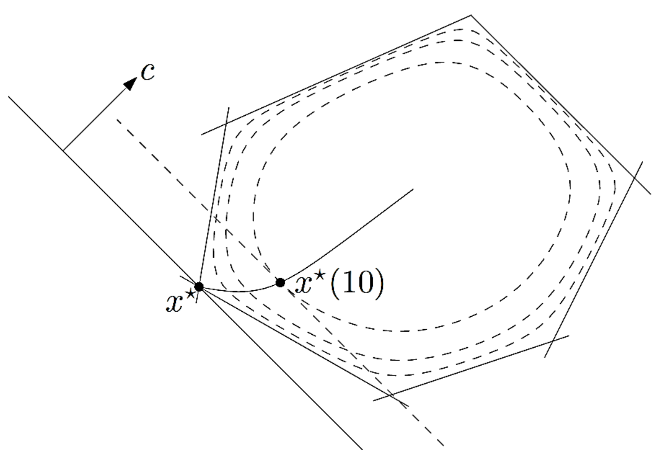


Figure 2.3 Central path for a LP with 2 variables and 6 linear inequality constraints (after Boyd and Vandenberghe (2004))

The barrier method requires a feasible starting point $x^{(0)}$, thus the initial point needs to satisfy the following condition:

$$f_i(\mathbf{x}) \leq 0, i = 1, \dots, m, \qquad and \quad \mathbf{Ax} = \mathbf{b} \tag{2.30}$$

Therefore, the barrier method is preceded by a "Phase I" sub-problem to generate a feasible initial input. To this end, we introduce a slack variable s and formulate the following optimization problem as:

$$\begin{aligned} &\text{minimize} && s \\ &\text{subject to} && f_i(\mathbf{x}) < s, \qquad \forall i = 1, \dots, m \\ &&& \mathbf{Ax} = \mathbf{b} \end{aligned} \tag{2.31}$$

where $\mathbf{x} \in \mathbf{R}^n, and\ s \in \mathbf{R}$. Problem (2.31) is often called the phase I optimization problem. Let s^* denote the optimal value of the optimization problems (2.31). When $s^* < 0$, $f_i(\mathbf{x}) < s^* \Rightarrow f_i(\mathbf{x}) < 0$. In the application, there is no need to solve (2.31) to very high accuracy, and the optimization can terminate once the condition $s^* < 0$ is satisfied. On the other hand, if $s^* > 0$, then (2.30) is infeasible. By evaluating the dual objective function, whenever the dual objective is detected with positive value, according to the lower bound property, i.e. the dual objective is always the lower bound to the primal objective function, then we can guarantee that the primal $s^* > 0$. However, it should be noted that there are other ways to treat the infeasible (Boyd and Vandenberghe 2004) .

2.5.4 A generic algorithm

For the solution algorithm of a nonlinear programming problem, let us assume a feasible initial input (could be generated in the "Phase I" problem), the solution process of an optimization problem can be described as follows:

1. Start with a feasible input, $(\mathbf{x}^0, \boldsymbol{\lambda}^0, \mathbf{v}^0)$;

2. Check the termination criterion, if yes, stop;

3. Compute the Newton's direction, $(\Delta x, \Delta\lambda, \Delta\nu)$ by solving the quadratic sub-problem;

4. Compute the step size by line search;

5. Update the solution $(x^{k+1}, \lambda^{k+1}, \nu^{k+1}) = (x^{k+1} + \Delta x, \lambda^{k+1} + \Delta\lambda, \nu^{k+1} + \Delta\nu)$

For a good nonlinear solver, the algorithm is actually much more complicated than that as described above, as many special situations have to be considered for a general problem.

2.6 Optimization algorithms in this research

Nonlinear programming technique will be applied to the solution of the optimization problems arising from the FELA. The limit analysis problem will be formulated as the second order cone programming for material obeying Mohr-Coulomb yield criterion under plane strain analysis and Drucker Prager yield criterion for full three-dimensional analysis. The primal-dual interior point algorithm specialized for this category of the problem will be applied. The SOCP problem will be solved by the third-party solver MOSEK-Aps (2010) with great efficiency However, it should be noted that there are plenty of solvers for the solution of the standard conic programming such as SDPA (Yamashita et al. 2003), Sedumi (Labit et al. 2002), etc. A benchmarking work among the existing conic programming solvers has been carried out by Mittelman (2003), and it can be expected that more will emerge in the future considering the rapid developments in the area of conic programming.

Regarding the algorithms for the general nonlinear programming, there are a large number of variants of the implementations of the solution techniques as described in the preceding sections. Differences among these algorithms primarily lie in the treatment of the inequality constraints, line search scheme, or the way to rewrite the augmented complementary slackness condition, etc.

In this research, the interior point algorithm for the general nonlinear programming algorithm is applied for the 3D formulation of the Mohr Coulomb material. Third-party solver will be used for this purpose, and IPOPT (Burland et al. 1977a) and KNITRO (Byrd et al. 2006) will be adopted for the present work.

2.7 Summary

A mathematical programming minimizes or maximizes the objective function that is subjected a number of constraints. Efficient and robust algorithms exist only for convex programming for which the local minimum is guaranteed to be the global minimum. For an optimization problem to be convex, the objective and inequality constraints are required to be convex and the equality constraints are required to be affined. Finding the optimal value of the optimization problem is equivalent to the solution of the optimality condition, i.e. KKT condition. Various solution techniques, essentially based on the Newton's method have been developed in the literature. Algorithms used in this research are mostly tailored from third-party solver, but a bespoke solver, particularly for the large-scale limit analysis problem would of great benefit.

CHAPTER 3: FUNDEMENTALS OF LIMIT ANALYSIS FOR GEOTECHNICAL MATERIALS

3.1 Introduction

In this chapter, basic concepts of limit analysis are presented. As the classical theory of limit analysis originated from the plasticity theory for metal, the assumptions in the application in soil mechanics should be clearly understood. The following discussion covers the lower bound and upper bound theorems, and assumptions need to be ensured in the application in geotechnical materials and validations of limit analysis in stability analysis in the geotechnical engineering. Various yield criteria developed and widely accepted in the field of geotechnical engineering are discussed at the end of this chapter.

3.2 Requirements for a general solution for boundary value problems

According to the theory of static behaviour of deformable solid, the below conditions must be satisfied for a solution to be rigorous:

(1) Equilibrium conditions

(2) Compatibility conditions

(3) Boundary conditions

To write the above conditions in mathematical terms, we first introduce some notations for the convenience of expression. Let Ω denotes the domain of study and $\partial\Omega$ denotes its boundary that comprises of a Neumann portion (surface traction boundary) Γ^N and a Dirichlet part (velocity boundary) so that $\partial\Omega = \Gamma^N \cup \Gamma^D$. On Γ^N , the prescribed surface traction $\mathbf{g}(\mathbf{x})$ is specified, $\mathbf{x} \in \Gamma^N$, and the body force is prescribed over the domain under consideration $\mathbf{f}(\mathbf{x})$ $\mathbf{x} \in \Omega$.

The stress field that satisfied the static equilibrium is called *statically admissible* or $\boldsymbol{\sigma}$SA, which is defined in eq.(3.1).

$$\text{SA}: \boldsymbol{\sigma}\text{SA} \Leftrightarrow \{-\boldsymbol{\sigma} \cdot \nabla = \mathbf{f}\ x \in \Omega \text{ and } \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}\ \ x \in \Gamma^N\} \tag{3.1}$$

where $\boldsymbol{\sigma} =$ stress tensor and $\mathbf{n} =$ normal vector of the surface where traction acts on at a given point.

The stress field that satisfied the static equilibrium is called *statically admissible* or $\mathbf{u}$KA, which is defined in eq.(3.2).

$$\text{KA}: \mathbf{u}\text{KA} \Leftrightarrow \left\{\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \mathbf{u}\boldsymbol{\nabla})\ x \in \Omega \text{ and } \mathbf{u} = \bar{\mathbf{u}}\ \ x \in \Gamma^D\right\} \tag{3.2}$$

where $\boldsymbol{\varepsilon} =$ strain tensor and u= displacement field.

For a three dimensional problem, there are in total 9 equations (3 equilibrium equations from (3.1) and 6 compatibility equations from (3.2) and 15 variables (6 stresses, 6 strains, and 3 displacements). The indeterminacy can be eliminated by additional 6 equations from the constitutive relations that relate the stresses with strains. The additional equations are contributed by the plastic flow rule that can either be associated (the potential function being identical to the yield function) or non-associated otherwise. Any rigorous solutions must fulfil all the conditions arising from the equilibrium conditions and constitutive relations, in other words, the equilibrium set $\{\boldsymbol{\sigma}, \mathbf{f}, \mathbf{t}\}$ and the deformation set $\{\mathbf{u}, \boldsymbol{\varepsilon}\}$ must exist simultaneously.

3.3 Principle of virtual work

The alternative to the equilibrium formulation of a boundary value problem is to use the energy formulation that relates the internal power dissipation to the external work done, which yields a variational form of the partial differential equations or the principle of virtual work which is given eq.(3.3).

$$\hat{\boldsymbol{\sigma}}\mathrm{SA}\mathcal{F}_{ext} \Leftrightarrow P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}}) = P_{ext}(\mathcal{F}_{ext}, \hat{\mathbf{u}})\ \forall \hat{\mathbf{u}}\mathrm{KA} \tag{3.3}$$

where $P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}})$ = internal power dissipation and $P_{ext}(\mathcal{F}_{ext}, \hat{\mathbf{u}})$ = work rate of the external loads $\mathcal{F}_{ext} = (\mathbf{f},\ \mathbf{t})$ associated with the velocity field or plastic flow $\hat{\mathbf{u}} = \mathbf{u}(\mathbf{x})$. The hat on top of variable denotes the virtual quantities. $P_{ext}(\mathcal{F}_{ext}, \hat{\mathbf{u}})$ is given by the linear functional eq. (3.4).

$$P_{ext}(\mathcal{F}_{ext}, \hat{\mathbf{u}}) = \int_{\Omega} \mathbf{f} \cdot \hat{\mathbf{u}}\, d\Omega + \int_{\Gamma^N} \mathbf{t} \cdot \hat{\mathbf{u}}\, dS \tag{3.4}$$

$\mathbf{f}$ = body force and $\mathbf{t}$ = surface traction on Γ^N.

$P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}})$ is the internal work rate and is given by eq.(3.5) for continuous field.

$$P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}}) = \int_{\Omega} \hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{\varepsilon}}\, d\Omega = \int_{\Omega} \quad d\Omega \tag{3.5}$$

It should be noted that (3.5) is valid only for the continuous velocity field, and if the discontinuous velocity field is considered, energy dissipation will also take place along discontinuities, which will result in an additional term in P_{int} as (3.6).

$$P_{int}(\widehat{\boldsymbol{\sigma}},\widehat{\mathbf{u}}) = \int_{\Omega} \widehat{\boldsymbol{\sigma}}:\widehat{\boldsymbol{\varepsilon}} d\Omega + \int_{\Sigma} \widehat{\boldsymbol{\sigma}}:\widehat{\boldsymbol{\chi}} d\Sigma \tag{3.6}$$

$\widehat{\boldsymbol{\chi}} = \frac{1}{2}([\![\widehat{\mathbf{u}}]\!]\mathbf{n} + \mathbf{n}[\![\widehat{\mathbf{u}}]\!])$ and $[\![*]\!]$ denotes the jump of a quantity across Σ.

3.4 Assumptions in limit analysis

The following assumptions are required to ensure the validity of theorems of the lower bound and upper bound.

(1) Small change in the geometry such that the principal virtual work applies.

(2) Perfectly plastic behaviour, which implies that work hardening and work softening behaviour are ignored.

(3) Associated flow rule or the normality condition.

For the majority of stability problems in geotechnical engineering, it is certain that a measurable amount of the deformation will accumulate before the plastic flow because soils are relatively compressible. However, in geotechnical engineering for which the domain is usually of relative large in comparison to such deformation, the assumption of small deformation is valid in most cases. The centre of the dispute for the applicability of the limit analysis to geotechnical materials is the assumption of the associated flow rule, because it predicts unreasonably larger dilation than can be measured in laboratory or field tests. However, significance of the dilation effects on the collapse loads has been shown to be not prominent for many problems.

3.4.1 Drucker's stability postulate

The Drucker's stability postulate is equivalent to the requirement of the one-to-one relationship between stresses and strains. In other words, Drucker's stability postulate is a requirement of the uniqueness of the solution for the system. In mathematical programming, it is the requirement of convexity of the problem (convexity of the yield criteria) under which the local minima are automatically guaranteed to be the global minima.

Consider the stress-strain curves in a uniaxial compression test (Figure 3.1), curves (a), (b), and (c) are defined as stable because a one-to-one relationship between stress and strain is observed. While for curve (d), it corresponds to an unstable material as the strains are not uniquely defined by a particular stress state. To distinguish two types of materials, a simple indicator $\Delta\sigma\Delta\varepsilon$ can be designed, i.e., if $\Delta\sigma\Delta\varepsilon > 0$, the material is stable and unstable otherwise.

Drucker (1953) extended the positive incremental work done concept $\Delta\sigma\Delta\varepsilon$ under the uniaxial condition and proposed a more general criterion for cases with general stress state. According to Drucker (1953), stable material should be that "(a) positive work is done by the external agency during the application of the added set of stresses on the changes in strains and (b) nonnegative network is done by the external agency over the cycle of application and removal."

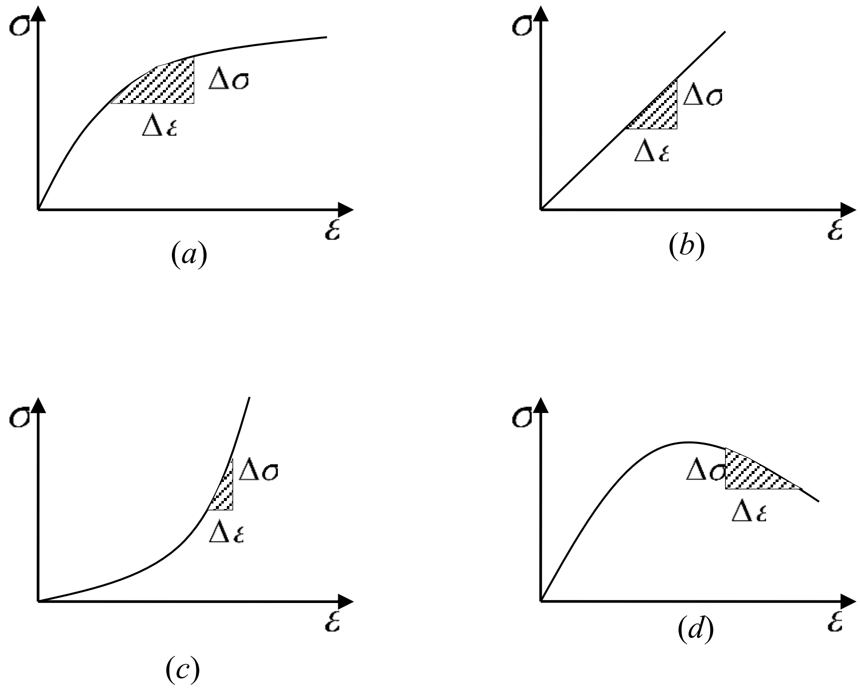


Figure 3.1 Stable and unstable stress-strain curves:(a),(b),and (c) Stable materials, $\Delta\sigma\Delta\varepsilon > 0$. (d) unstable material , $\Delta\sigma\Delta\varepsilon < 0$

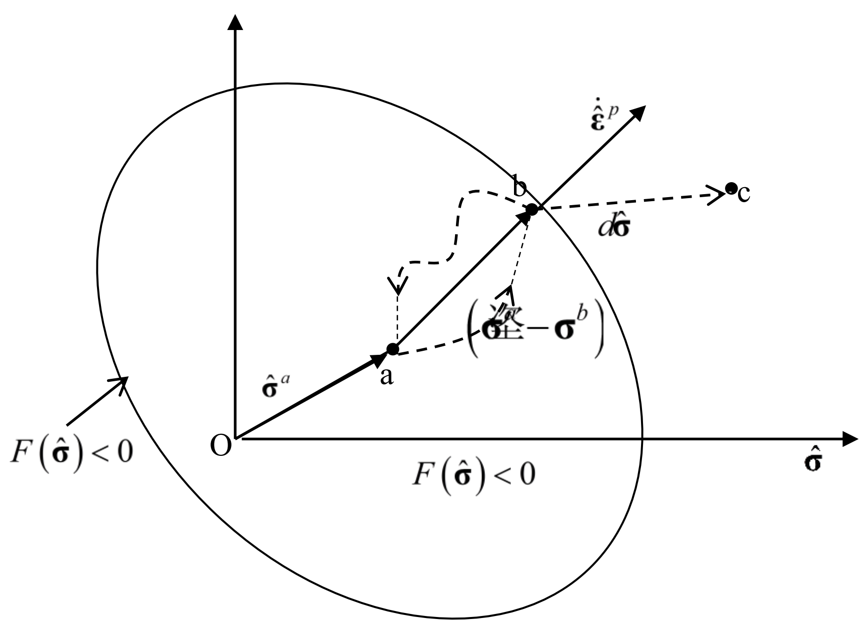


Figure 3.2 A graphical representation of the loading-unloading cycle in the stress space

Consider a complete loading cycle $a \rightarrow b \rightarrow c \rightarrow b \rightarrow a$ for a work hardening material (Figure 3.2), the work done by the external agency is nonnegative, therefore

$$\oint (\boldsymbol{\sigma} - \boldsymbol{\sigma}^a): \dot{\boldsymbol{\varepsilon}} dt = \oint (\boldsymbol{\sigma} - \boldsymbol{\sigma}^a): \dot{\boldsymbol{\varepsilon}}^e dt + \oint (\boldsymbol{\sigma} - \boldsymbol{\sigma}^a): \dot{\boldsymbol{\varepsilon}}^p dt \geq 0 \tag{3.7}$$

where $\dot{\boldsymbol{\varepsilon}}^e, \dot{\boldsymbol{\varepsilon}}^p$= elastic and plastic components of the strain rate respectively. Since the elastic work in the load cycle is zero, eq.(3.7) reduces to

$$\oint (\boldsymbol{\sigma} - \boldsymbol{\sigma}^a): \dot{\boldsymbol{\varepsilon}}^p dt \equiv \oint (\boldsymbol{\sigma}^b + \mathrm{d}\boldsymbol{\sigma} - \boldsymbol{\sigma}^a): \dot{\boldsymbol{\varepsilon}}^p dt \geq 0 \tag{3.8}$$

Taking the limit of (3.8) by setting $dt \rightarrow 0$ yields the expression for eq.(3.9)

$$(\boldsymbol{\sigma}^b - \boldsymbol{\sigma}^a): \dot{\boldsymbol{\varepsilon}}^p \geq 0 \tag{3.9}$$

Eq.(3.9) is of great significance in the proof of the lower and upper bound theorems in limit analysis.

In addition, it is due to the Drucker's stability postulate that the convexity of the yield function is guaranteed, which enables the robust solution of the resulting optimization as discussed in Chapter 2 (Page 25).

3.4.2 Perfectly plastic material

Perfectly plastic material implies that after attaining a certain stress state, the material undergoes unconstrained plastic flow without causing any increase in the limit load.

Apparently, this is a radical simplification of behaviours of real soils for which a certain degree of hardening and softening is always observed. For normally consolidated soil, the soil strength associated with the ultimate strength used in the analysis is practically acceptable as long as sufficient deformation is allowed for the ultimate state to develop without significantly altering the geometry of the problem under consideration (Figure 3.3). For the soils exhibiting post-peaking strain-softening, overconsolidated clay for instance, simplifying the soil behaviour as a perfect plastic material is justified by considering the effect of progressive failure and choosing an average mobilized strength. However, it should be noted that the assumption of the perfect plasticity of material could be removed by performing a series of limit analysis or the so-call incremental limit analysis (Leu 2005; Leu 2007; Yang 1993).

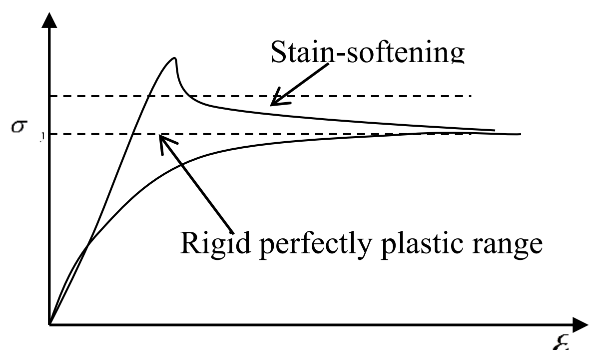


Figure 3.3 Stress –strain relationship for ideal and real soils

For materials assuming perfect plastic behaviour, the yield criterion does not change during the plastic flow and thus the plastic work increment must be equal to zero

$$d\boldsymbol{\varepsilon}^p : d\boldsymbol{\sigma} = 0 \tag{3.10}$$

Eq.(3.10) indicates that the plastic increment vector $d\boldsymbol{\varepsilon}^p$ is perpendicular to the direction of $d\boldsymbol{\sigma}$, that is tangential to the yield envelope.

Since the plastic strain rate is not affected by the elastic behaviour of the material and the only information that is of interests in limit analysis is the ultimate limit load, the entire elastic behaviour can be neglected if small deformation is preserved, consequently, the rigid-perfectly plasticity is assumed throughout the thesis and rigid-perfectly material will referred to as perfectly plastic material for short unless otherwise noted.

3.4.3 Associated flow rule

At collapse, the plastic flow is unconstrained for the perfectly plastic materials, thus the magnitude of the total plastic strain ε_{ij}^{p} cannot be determined. It is more convenient to adopt a concept of the plastic strain rate $\dot{\boldsymbol{\varepsilon}}^{\boldsymbol{p}}$ that relates to the plastic potential function $G(\sigma_{ij})$ as eq.(3.11).

$$\dot{\varepsilon}_{ij}^{p} = \lambda \frac{\partial G(\sigma_{ij})}{\partial \sigma_{ij}} \tag{3.11}$$

where $\lambda =$ nonnegative Lagrange multiplier and G = a scalar function of the stress tensor σ_{ij}. From eq. (3.11), the rate of the plastic strain is independent of the loading history.

Flow rule is of great significance in modeling the plastic behavior of a soil, as it governs the dilatancy effects which in turn determine the volume changes and the strength. To simplify the discussion, the potential function $G(\sigma_{ij})$ is assumed to be the same as the yield function, which leads to associated flow rule. eq. (3.11) becomes

$$\dot{\varepsilon}_{ij}^{p} = \lambda \frac{\partial f(\sigma_{ij})}{\partial \sigma_{ij}} \tag{3.12}$$

Eq.(3.12) is also known as the normality condition (optimality condition), i.e. the plastic strain rate is normal to the yield surface. The principle of the maximum power dissipation forms the basis of limit analysis and offers a systematic mathematical tool for the study of the plastic behaviours. However, the associated flow as given by (3.12) is the centre of dispute for the applicability of limit analysis for granular materials, since (3.12) leads to larger dilation than the observed values in experiments. Consideration of the influence of the different dilatancy of the soil is beyond the scope of this study, but it has been shown by many researchers that the effects of the dilatancy are not sensitive to the collapse load in many cases (Cheng et al. 2007).

3.5 Lower bound and upper bound theorems

The objective of the limit analysis is to find the maximum load that a system can bear without collapse. For the ease of the discussion, let us assume that load will be increased from a reference loading $\mathcal{F}_{ext}^{0}$ by a multiplier α according to

$$\mathcal{F}_{ext} = \alpha \mathcal{F}_{ext}^{0} = \alpha(\mathbf{f^0}, \mathbf{t^0}) \tag{3.13}$$

The linearity of the $P_{ext}(\mathcal{F}_{ext}, \hat{u})$ implies $P_{ext}(\mathcal{F}_{ext}, \hat{u}) = \alpha P_{ext}(\mathcal{F}_{ext}^{0}, \hat{u})$

For a stress field to be *plastic admissible*,

$$\mathrm{PA}: \boldsymbol{\sigma}\mathrm{PA} \Leftrightarrow f(\boldsymbol{\sigma}) \leq 0 \; \forall \mathbf{x} \in \Omega \tag{3.14}$$

Therefore the stress is confined to a set $\mathbf{B} = \{\boldsymbol{\sigma} | \boldsymbol{\sigma}\mathrm{PA} \; \forall \mathbf{x} \in \Omega\}$

The maximum multiplier can be obtained by (3.15).

$$\alpha^* = \quad \sup \alpha \quad \text{subject to} \begin{cases} \exists \hat{\boldsymbol{\sigma}} \in \mathrm{B} \\ P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}}) = \alpha P_{ext}(\mathcal{F}_{ext}^0, \hat{\mathbf{u}}) \quad \forall \hat{\mathbf{u}} \mathrm{KA} \end{cases} \tag{3.15}$$

Noting that the velocity $\mathbf{u}$ is linear in $P_{ext}(\mathcal{F}_{ext}, \hat{\mathbf{u}})$ and bilinear in $P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}})$, it is convenient to choose a reference loading such that $P_{ext}(\mathcal{F}_{ext}^0, \hat{\mathbf{u}}) = 1$, i.e., $\hat{\mathbf{u}}$ is confined to a hyperplane $C = \{\hat{\mathbf{u}}\mathrm{KA} | P_{ext}(\mathcal{F}_{ext}^0, \hat{\mathbf{u}}) = 1 \text{ and } \hat{\mathbf{u}}\mathrm{KA}\}$. (3.15) is then equivalent to (3.16) and (3.17).

$$\alpha^* = \sup_{\hat{\boldsymbol{\sigma}} \in B} \inf_{\hat{\mathbf{u}} \in C} \frac{P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}})}{P_{ext}(\mathcal{F}_{ext}^0, \hat{\mathbf{u}})} = \sup_{\hat{\boldsymbol{\sigma}} \in B} \inf_{\hat{\mathbf{u}} \in C} P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}}) \tag{3.16}$$

$$\alpha^* = \inf_{\hat{\mathbf{u}} \in C} \sup_{\hat{\boldsymbol{\sigma}} \in B} \frac{P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}})}{P_{ext}(\mathcal{F}_{ext}^0, \hat{\mathbf{u}})} = \inf_{\hat{\mathbf{u}} \in C} \sup_{\hat{\boldsymbol{\sigma}} \in B} P_{int}(\hat{\boldsymbol{\sigma}}, \hat{\mathbf{u}}) \tag{3.17}$$

Noting from (3.16), if the inter infimum is exactly calculated, it will lead to a lower bound solution the exact load

$$\alpha^{LB} = P_{int}(\sigma, u^*) \leq P_{int}(\hat{\sigma}^*, \hat{u}^*) \tag{3.18}$$

As the inner infimum (minimum potential energy) implies the equilibrium condition, the lower bound theorem can be expressed as *any load that is in equilibrium with the external loading (σSA) and nowhere violates the yield condition will be less than or at most equal to the actual collapse load.*

On the other hand, if the internal supremum is calculated exactly, i.e., $\alpha = P_{int}(\sigma^*, u)$, it will serve as an upper bound to the exactly collapse load.

In accordance to the lower bound and upper bound theorems, the intuitive idea of the lower bound is to construct various statically admissible stress fields and seek the one corresponding to the largest load. On the other hand, upper bound analysis is to construct kinematically admissible velocity fields and find the one that gives smallest collapse load.

3.5.1 Illustration of lower bound limit theorem

It has been shown in the previous section that the exact solution could be found from a saddle point problem and the lower and upper bounds are obvious from (3.18) . However, this interpretation is slightly mathematical and it is worthy going through a more physical proof to illustrate the importance of the assumptions that are required to be made for the material behaviour in the sections to follow.

The virtual work equation for the actual plastic collapse of a body can be expressed as

$$P_{int}(\boldsymbol{\sigma}^{\boldsymbol{c}}, \mathbf{u}^{\mathbf{c}}) = P_{ext}(\mathcal{F}_{ext}^{c}, \mathbf{u}^{\mathbf{c}}) \tag{3.19}$$

where the superscript c denotes the actual state at collapse. The virtual work equation for the statically admissible stress state with actual kinematical field is written as eq.(3.20).

$$P_{int}(\boldsymbol{\sigma}^{\boldsymbol{L}}, \mathbf{u}^{\mathbf{c}}) = P_{ext}(\mathcal{F}_{ext}^{L}, \mathbf{u}^{\mathbf{c}}) \tag{3.20}$$

Subtracting (3.20) from (3.19) yields

$$P_{ext}(\mathcal{F}_{ext}^{c} - \mathcal{F}_{ext}^{L}, \mathbf{u^c}) = P_{int}(\boldsymbol{\sigma^c} - \boldsymbol{\sigma^L}, \mathbf{u^c}) = \int_{\Omega} (\boldsymbol{\sigma^c} - \boldsymbol{\sigma^L}): \boldsymbol{\varepsilon} d\Omega \tag{3.21}$$

With (3.9), we have

$$P_{ext}(\mathcal{F}_{ext}^{c} - \mathcal{F}_{ext}^{L}, \mathbf{u^c}) \geq 0 \Leftrightarrow \int_{\Gamma^{\mathrm{N}}} (\mathbf{t}^c - \mathbf{t}^L) \cdot \mathbf{u}^U \, dS + \int_{\Omega} (\mathbf{f}^c - \mathbf{f}^L) \cdot \mathbf{u}^U d\Omega \geq 0 \tag{3.22}$$

Considering an analysis with fixed body force, i.e. $\mathbf{f}^c = \mathbf{f}^L$, term 2 in (3.22) will vanish and consequently, it is easy to verify that (3.22) leads to $\mathbf{t}^L < \mathbf{t}^c$. Similar argument can be made for $\mathbf{f}^L$ when the body force is to be optimized.

3.5.2 Illustration of the upper bound theorem

The virtual work equation for the actual stress field with kinematic velocity field can be expressed as

$$P_{ext}(\alpha^c \mathcal{F}_{ext}^{0}, \mathbf{u}^U) = P_{int}(\boldsymbol{\sigma}^c, \mathbf{u}^U) \tag{3.23}$$

In the upper bound analysis, the external power is equated to the maximum power dissipation associated with a particular velocity field, i.e.,

$$\begin{aligned} P_{ext}(\alpha^{UB} \mathcal{F}_{ext}^{0}, \mathbf{u}^U) &= P_{res}(\boldsymbol{\sigma}^{UB}, \mathbf{u}^U) \\ &= \sup_{\sigma \in B} P_{int}(\boldsymbol{\sigma}, \mathbf{u}^U) \\ &\geq P_{int}(\sigma^c, u^U) \end{aligned} \tag{3.24}$$

Subtracting(3.23) from (3.24) yields

$$\begin{aligned} & P_{ext}(\alpha^{UB}\mathcal{F}^0_{ext}, \mathbf{u}^U) - P_{ext}(\alpha^{*}\mathcal{F}^0_{ext}, \mathbf{u}^U) \geq 0 \\ \Longrightarrow \quad & (\alpha^{UB} - \alpha^{*})P_{ext}(\mathcal{F}^0_{ext}, \mathbf{u}^U) \geq 0 \end{aligned} \tag{3.25}$$

Because the work done due to the external force is nonnegative, i.e., $P_{ext}(\mathcal{F}^0_{ext}, \mathbf{u}^U) \geq 0$, we have $\alpha^{UB} - \alpha^{c} \geq 0$.

3.6 Requirements of the discretisation in the lower and upper bound formulations

To ensure the calculated limit solutions are the rigorous bounds for the exact collapse load, the continuous problem needs to be discretised such that the discretisation does not compromise the nature of the bounds. To this end, specialised discretisation will be used for the FELA in this work, which will be restricted to the finite element discretisation only. It should be noted that the finite element discretisation is not the only option for such purposes (Chen et al. 2008; Le et al. 2010a; Le et al. 2010b) .

3.6.1 Exact bounds

A rigorous upper bound on the exact solution α^* can be obtained provided that the inner supremum in eq. (3.26) is computed exactly,

$$\alpha^* = \inf_{\mathbf{u}\in C} \sup_{\boldsymbol{\sigma}\in B} P_{int}(\boldsymbol{\sigma}, \mathbf{u}) = \inf_{\mathbf{u}\in C} P_{int}(\boldsymbol{\sigma}^*, \mathbf{u}) \leq P_{int}(\boldsymbol{\sigma}^*, \mathbf{u}) = \alpha^{UB}, \forall \mathbf{u} \in C \tag{3.26}$$

Likewise, a rigorous lower bound is obtained if the inner infimum is exactly calculated.

$$\alpha^* = \sup_{\boldsymbol{\sigma}\in B} \inf_{\mathbf{u}\in C} P_{int}(\boldsymbol{\sigma}, \mathbf{u}) = \sup_{\boldsymbol{\sigma}\in B} P_{int}(\boldsymbol{\sigma}, \mathbf{u}^*) \geq P_{int}(\boldsymbol{\sigma}, \mathbf{u}^*) = \alpha^{LB}, \forall \boldsymbol{\sigma} \in B \tag{3.27}$$

where $\boldsymbol{\sigma}^*$ and $\mathbf{u}^*$ are the optimals to obtain the inner extrema:

$$P_{int}(\boldsymbol{\sigma}^*, \mathbf{u}) = \sup_{\boldsymbol{\sigma} \in B} P_{int}(\boldsymbol{\sigma}, \mathbf{u}) \ \ \forall \mathbf{u} \ \in C$$
$$P_{int}(\boldsymbol{\sigma}, \mathbf{u}^*) = \inf_{\mathbf{u} \in C} P_{int}(\boldsymbol{\sigma}, \mathbf{u}) \ \ \forall \boldsymbol{\sigma} \in B \tag{3.28}$$

3.6.2 Purely static discretisation and purely kinematic discretisation

Consider finite element function spaces X_h for $\boldsymbol{\sigma}$ and Y_h for $\mathbf{u}$ associated with the domain of study Ω, where the subscript h denotes the typical size of the element in the mesh. Accordingly, the discrete e convex sets of B_h must be such that $B_h \subset B \cap X_h$ and the affine hyperplane to which the u_h is restricted becomes $C_h = \{\mathbf{u}_h \in Y_h | F(\mathbf{u}_h) = 1\}$.

To ensure the rigorous lower and upper bounds on the exact solution, it is of interests to consider the particular properties of the interpolation space $X_h \times Y_h$. Regarding the lower bound analysis, the discretisation must satisfy the following conditions (Ciria 2002):

(1) Satisfaction of the discretize equilibrium equation on X_h implies the continuous equilibrium equation, mathematically, i.e. eq.(3.27)

$$P_{int}(\boldsymbol{\sigma}_h, \mathbf{u}_h) = \alpha P_{ext}(\mathcal{F}^0_{ext}, \mathbf{u}_h) \ \forall \mathbf{u}_h \in Y_h \Rightarrow P_{int}(\boldsymbol{\sigma}_h, \mathbf{u}) = \alpha P_{ext}(\mathcal{F}^0_{ext}, \mathbf{u}), \forall \mathbf{u} \in Y \tag{3.29}$$

(2) The discretised stress field is admissible, i.e. $\boldsymbol{\sigma}_h \in B_h$ at some points implies $\boldsymbol{\sigma}_h \in B_h$ over the entire domain.

A discretisation satisfying the conditions listed above is called *purely static* discretisation (Ciria 2002) and is denoted as $X_h^{LB} \times {Y_h}^{LB}$.

On the other hand, an upper bound solution requires the exact computation of the power dissipation rate, or mathematically

$$\max_{\boldsymbol{\sigma}_h \in B_h} P_{int}(\boldsymbol{\sigma}_h, \mathbf{u}_h) = \max_{\boldsymbol{\sigma} \in B} P_{int}(\boldsymbol{\sigma}, \mathbf{u}_h)\,, \forall \mathbf{u}_h \in Y_h \tag{3.30}$$

A discretisation that satisfies (3.30) is called the *purely kinematic* discretisation and is denoted as $X_h^{UB} \times Y_h^{UB}$.

3.7 Yield criteria

Yield criteria are the most important information in the stability analysis and contribute to the major difficulty in the formulation of the FELA as a standard solvable mathematical optimization. A great portion of the recent advances in the field of numerical FELA is about the manipulation of the yield criteria. Some of the popular criteria developed for the yielding of soils and rock masses are discussed in this section.

3.7.1 Isotropic materials

Isotropic materials are assumed in the present work. The yield criteria are more conveniently represented in the stress invariants for better physical interpretation and removal on the dependency of the choice of the coordinates. Stresses are defined as a tensor σ_{ij}, and it takes the form

$$\sigma_{ij} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \tag{3.31}$$

Before heading further, it is worthwhile to review the various stress variants, in terms of which the yield criteria will be expresses depending on the convenience.

The spherical stress σ_m and the deviatoric stresses $s_{ij}$are given in eq.(3.32).

$$\sigma_m = \frac{1}{D}\sum_{i=1}^{N}\sigma_{ii}\,, \qquad s_{ij} = \sigma_{ij} - \sigma_m\delta_{ij} \tag{3.32}$$

where D = the dimension of the tensors, δ_{ij} = Kronecker's δ. σ_m = the mean stresses or the hydrostatic pressure and s_{ij}= deviatoric stress tensor.

The stress invariants of the stress tensor (3.31) are defined as follows

$$\begin{aligned} I_1 &= \sigma_{ii} = \sigma_{11} + \sigma_{22} + \sigma_{33} \\ I_2 &= \frac{1}{2}\left(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ji}\right) = \sigma_{11}\sigma_{22} + \sigma_{22}\sigma_{33} + \sigma_{11}\sigma_{33} - \sigma_{12}^2 - \sigma_{23}^2 - \sigma_{13}^2 \\ I_3 &= \det\sigma_{ij} \end{aligned} \tag{3.33}$$

For deviatoric stress tensor, we have

$$\begin{aligned} J_1 &= s_{ii} = 0 \\ J_2 &= \frac{1}{2}s_{ij}s_{ji} \\ J_3 &= \det s_{ij} \end{aligned} \tag{3.34}$$

The Lode angle is defined in terms of principal stresses eq.(3.35).

$$\theta = \tan^{-1}\left(\frac{1}{\sqrt{3}}\frac{\sigma_1 - 2\sigma_2 + \sigma_3}{\sigma_1 - \sigma_3}\right) \tag{3.35}$$

3.7.2 Mohr-Coulomb (MC) criterion

Mohr-Coulomb stress envelope is the most widely accepted criteria for determining the yielding for soils and rocks owing to its simplicity and fair accuracy. It postulates a linear relationship between the normal stress and shear stress at failure according to (3.36).

$$|\tau| = c' + \sigma_n' \tan\phi' \tag{3.36}$$

where c'=cohesion and $\phi' =$ friction angle.

Taking tensile stresses as positive, (3.36) can be written as (3.37) under plane strain condition:

$$\sqrt{\frac{\left(\sigma_x - \sigma_y\right)^2}{4} + \tau_{xy}^2} \leq c'\cos\phi' \tag{3.37}$$

Under three-dimensional condition, (3.36) can be written in terms of the principal stresses:

$$\sigma'_1 - a\sigma'_3 \leq k \tag{3.38}$$

where

$$a = \frac{1 - \sin\phi'}{1 + \sin\phi'} \quad \text{and} \quad k = \frac{2c'\cos\phi'}{1 + \sin\phi'}.$$

In the principal stress space, (3.38) is shaped like an irregular hexagonal pyramid as shown in Figure 3.4. Traces of MC in the meridional plane and deviatoric plane are shown in Figure 3.5, in which the larger and smaller magnitude σ_c and σ_t on the trace correspond to the triaxial compression and triaxial tension respectively.

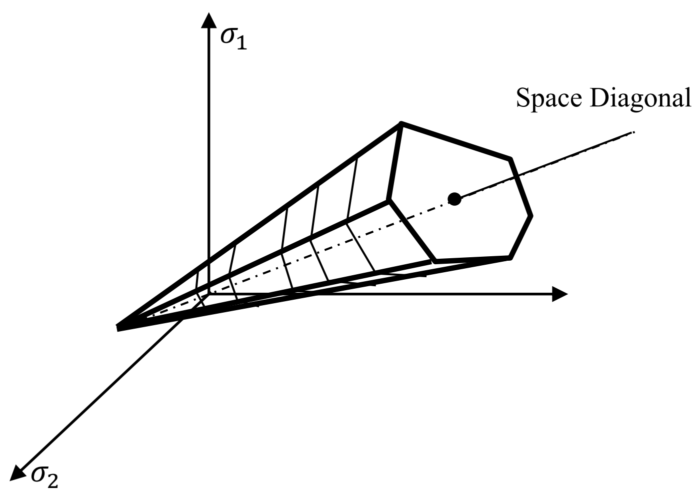


Figure 3.4 Mohr Coulomb yield criterion in the principal stress space

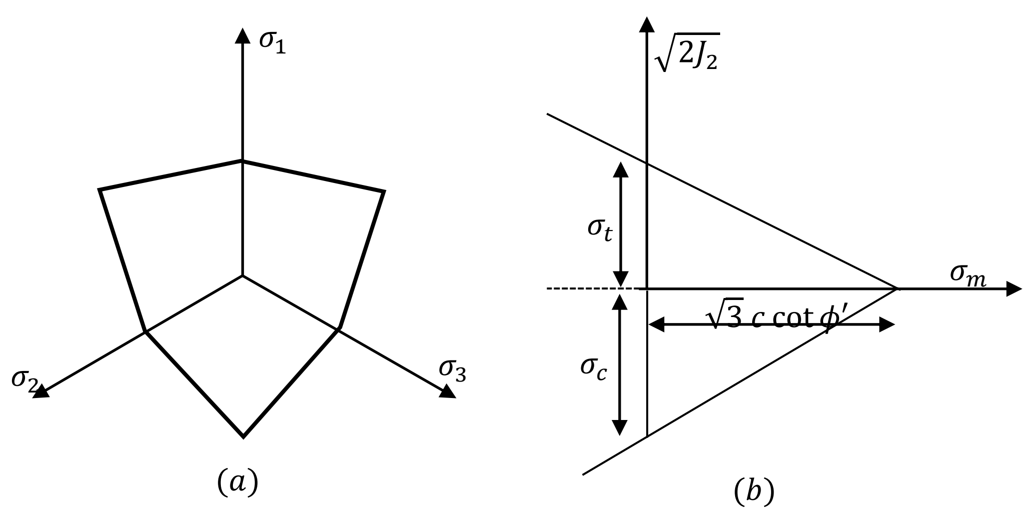


Figure 3.5 Traces of the Mohr-Coulomb yield function on the deviatoric plane and meridional plane, respectively (a) deviatoric plane and (b) meridional plane

In numerical applications, it is more convenient to express the yield criterion under a cylindrical coordinate system $\{\sigma_m, \theta, \bar{\sigma}\}$. Taking tensile stresses as positive, the MC yield function can be expressed as (3.39).

$$f = \sigma_m \sin\phi' + K(\theta)\bar{\sigma} - c\cos\phi' = 0 \tag{3.39}$$

where

$$K(\theta) = \cos\theta + \frac{1}{\sqrt{3}}\sin\theta\sin\phi'$$
$$\bar{\sigma} = \sqrt{J_2}$$

Accumulated empirical experiences and laboratory tests have shown that the MC provides a good model for the strength of most of the geomaterials. However, there are some shortcomings inherited in this failure model. Firstly, the intermediate principal stress σ_2 is not considered which is not complying with the experimental results (see (3.38)). Secondly, the curvature of the yield envelope in the meridional plane or Mohr plane is not considered, so the yield envelope is only valid over a limited range of stress level (see Figure 3.5). Lastly, corners and apex (see Figure 3.4) give rise to singularities of the gradient and Hessian at these points in the numerical application.

3.7.3 Drucker Prager (DP) criterion

Drucker Prager (DP) yield criterion (Drucker and Prager 1952) is an extension of the von Mises model to account for effects of the hydrostatic pressure. It can also be regarded as the simplest smoothing of the Mohr-Coulomb model to remove the singularities at corners.

The DP model can be written as

$$f^{DP} = a\sigma_m + \bar{\sigma} = k \tag{3.40}$$

where a and k are two material constants. Collecting the terms in (3.39), the MC can be rewritten into a similar form as (3.41).

$$f^{MC} = \frac{\sin\phi'}{K(\theta)}\sigma_m + \bar{\sigma} = \frac{c\cos\phi'}{K(\theta)} \tag{3.41}$$

Comparing (3.39) and (3.41), relations as given in eq.(3.42) are obtained when matching the parameters in the MC yield criterion to the DP yield criterion.

$$\begin{aligned} a &= \frac{\sin\phi'}{K(\theta)} \\ k &= \frac{c\cos\phi'}{K(\theta)} \end{aligned} \tag{3.42}$$

From (3.42), the relation between $\{a, k\}$ and $\{c', \phi'\}$ depends on the Lode angle θ for which the DP criterion is matched. Different Drucker Prager yield criteria on the deviatoric plane are shown in Figure 3.6. Parameters of a and k fitted with different situations are tabulated in Table 3.1.

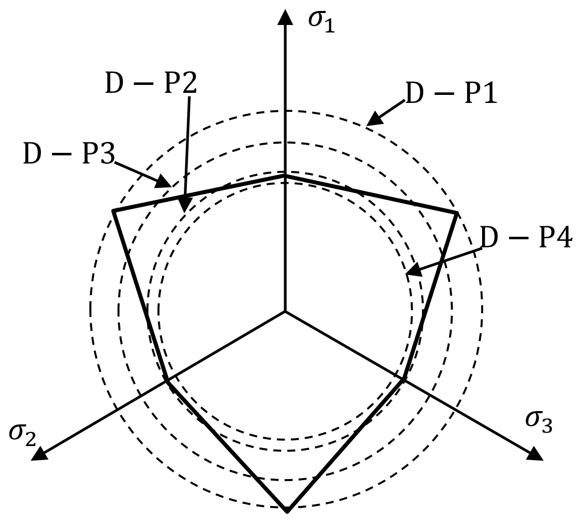


Figure 3.6 Mohr-Coulomb and Drucker Prager criteria in the deviatoric plane

Table 3.1 Relation of constants between Mohr-Coulomb and Drucker Prager model fitted under different circumstances

No.	Description	a	k
D-P1	The out most circle fit in deviatoric plane $\left(\theta = \frac{\pi}{6}\right)$	$\frac{6\sin\phi'}{\sqrt{3}(3-\sin\phi')}$	$\frac{6c\cos\phi'}{\sqrt{3}(3-\sin\phi')}$
D-P2	The circle matched at the triaxial extension point $\left(\theta = -\frac{\pi}{6}\right)$	$\frac{6\sin\phi'}{\sqrt{3}(3+\sin\phi')}$	$\frac{6c\cos\phi'}{\sqrt{3}(3+\sin\phi')}$
D-P3	The circle with equal area with that for the Mohr-Coulomb trace on the deviatoric plane	$\frac{6\sqrt{3}\sin\phi'}{\sqrt{2\sqrt{3}\pi(9-\sin^2\phi')}}$	$\frac{6\sqrt{3}\,\mathrm{c}\cos\phi'}{\sqrt{2\sqrt{3}\pi(9-\sin^2\phi')}}$
D-P4	Inscribe circle	$\frac{3\sin\phi'}{\sqrt{3}\sqrt{(3+\sin^2\phi'}}$	$\frac{3c\cos\phi'}{\sqrt{3}\sqrt{(3+\sin^2\phi'}}$

As noted by Chen and Liu (1990), constants in (3.40) should not be treated as fixed expressions. Rather, they depend on the type of the problems to be solved. Further discussions regarding this issue are given in Chen and Mizuno (1979).

The applicability of the DP yield criterion in the application to geomaterials as an approximation of the MC yield criterion was discussed in various works (Clausen et al. 2010; Schweiger 1994). Results in their researches are against the use of the DP yield criterion and suggested that the DP model was more justified as an educational tool rather than a practical yield model. In the work by Clausen et al. (2010), only three types of the matching were used and numerical experiments were restricted to the bearing capacity problems of rectangular footings. There have also been proponents who advocate the use of the Drucker Prager model with the finite element methods (Yang et al. 2009; Zhao et al. 2006), where more refined parameter matching was considered and conclusions in favour of the use of the Drucker Prager model in slope stability analysis were made.

In the application of the FELA, the most attractive feature of the DP model is that the DP yield criterion can be expressed as a second order conic constraint. Graphically as shown in Figure 3.7, it is obvious that DP model is a cone in the principal stress space. This feature enables the FELA for DP material to be formulated as an SOCP that can usually be solved much faster with specialised algorithms than a general nonlinear formulation with nonlinear programming solver.

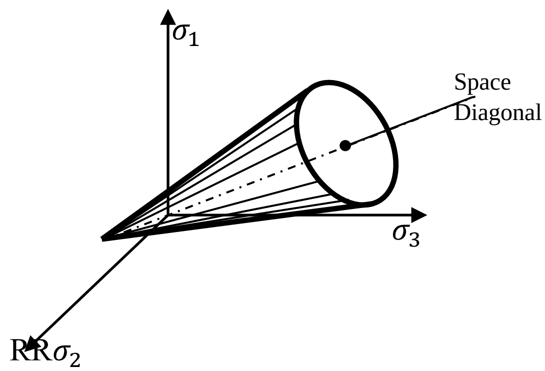


Figure 3.7 Drucker Prager yield criterion in the principal stress space

3.7.4 A family of yield criteria in conic form

It has been shown by many researchers (Krabbenhoft et al. 2007a; Krabbenhoft et al. 2008; Makrodimopoulos 2010; Makrodimopoulos and Martin 2006; Makrodimopoulos and Martin 2007) that a large class of commonly adopted yield criteria developed for geomaterials can be cast into the form of conic constraints by rotation and introduction of auxiliary variables. The significance of this is that such a manipulation of the yield criteria will lead to a conic formulation that is usually easier to solve and bypass the difficulty of the singularity arising from direct formulation.

MC yield criterion under plane strain condition and DP yield criterion can be cast into the second order cone constraints. The MC yield criterion in full three-dimensional analysis can also be cast into the intersection of semi-definite cones.

Using stress variables defined in (3.32), MC criterion (3.37) can be rewritten as

$$\sqrt{s_{11}^2 + s_{12}^2} + \sigma_m \sin\phi' - c\cos\phi' \leq 0 \qquad (3.43)$$

By introducing an auxiliary variable z, (3.43) can be replaced by a second order cone constraint and an equality constraint (Makrodimopoulos and Martin 2006) as

$$\begin{aligned} & z + \sin\phi'\,\sigma_m = c'\cos\phi' \\ & \sqrt{s_{11}^2 + s_{12}^2} \leq z \end{aligned} \tag{3.44}$$

Similarly, the DP yield criterion (3.40) can be treated in the similar fashion and equivalently expressed as

$$\begin{aligned} & z + a\sigma_m = k \\ & \|\mathbf{y}\| \leq z \end{aligned} \tag{3.45}$$

where $\mathbf{y} \in \mathcal{R}^5$ is given (non-uniquely) by

$$\mathbf{y} = \begin{pmatrix} s'_{11} \\ s'_{22} \\ s'_{12} \\ s'_{23} \\ s'_{13} \end{pmatrix} = \begin{pmatrix} 1 & 1/2 & 0 & 0 & 0 \\ 0 & \sqrt{3}/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_{11} \\ s_{22} \\ s_{12} \\ s_{23} \\ s_{13} \end{pmatrix} \tag{3.46}$$

Note that the inequality in the relation (3.44) and (3.45) are the standard form of the second order cone (see the definition (2.10)) and can be readily incorporated in most of the existing convex solvers.

MC yield criterion (3.38) in three-dimensional analysis can be cast into the intersection of two semi-definite constraints, thanks to the fact that the principal stresses are essentially the eigenvalues of the stress tensor (Krabbenhoft et al. 2008; Makrodimopoulos 2010; Martin and Makrodimopoulos 2008).

$$k\mathbf{I} + a\mathbf{I}t - \boldsymbol{\sigma} \succcurlyeq 0$$
$$\boldsymbol{\sigma} - \mathbf{I}t \succcurlyeq 0 \tag{3.47}$$

where $\mathbf{I}$ is the identity matrix, $\boldsymbol{\sigma}$ denotes the matrix of the stress tensor. a and k are material constants defined in (3.38) and t is a scalar auxiliary variable.

Eq.(3.47) can be written in the form of vector as

$$k\mathbf{p} - \boldsymbol{\sigma} + at\mathbf{p} \in \mathcal{S}_{+}^{(3)}$$
$$\boldsymbol{\sigma} - t\mathbf{p} \in \mathcal{S}_{+}^{(3)} \tag{3.48}$$

where $\mathbf{p} = (1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0)^T$.

Makrodimopoulos (2010) has proposed a more compact form of (3.44), (3.45), and (3.47) as

$$\mathbf{b} + \mathbf{Q}\tilde{\boldsymbol{\sigma}} + \mathbf{R}t \in \mathcal{C} \tag{3.49}$$

where $\mathbf{b}, \mathbf{Q}$ and $\mathbf{R}$ depend on the respective yield criterion, $\tilde{\boldsymbol{\sigma}} =$ stress variables defined in (3.50)

$$\tilde{\boldsymbol{\sigma}} = \begin{cases} (\sigma_m \quad s_{11} \quad s_{12})^{\mathrm{T}} & \text{for } 2D \\ (\sigma_m \quad s_{11} \quad s_{22} \quad s_{12} \quad s_{23} \quad s_{13})^T & \text{for } 3D \end{cases} \tag{3.50}$$

For Mohr-Coulomb model under plane strain conditions,

$$\mathbf{b} = \begin{pmatrix} c' \cos\phi' \\ 0 \\ 0 \end{pmatrix}, \text{and } \mathbf{Q} = \begin{pmatrix} -\sin\phi' & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.51}$$

For the DP model

$$\begin{gathered} \mathbf{b} = (k \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^T \\ \mathbf{Q} = \begin{pmatrix} -a & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{3}/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{gathered} \tag{3.52}$$

To write (3.48) in the form of (3.49), we have

$$\begin{gathered} \mathbf{Q_1} = -\mathbf{I}, \mathbf{R_1} = a\mathbf{p}, \mathbf{b_1} = k\mathbf{p} \\ \mathbf{Q_2} = \mathbf{I}, \mathbf{R_2} = -\mathbf{p}, \mathbf{b_2} = \mathbf{0} \end{gathered} \tag{3.53}$$

3.7.5 Nonlinear yield criteria

Neither DP yield criterion nor MC yield criterion considers the nonlinearity of the trace of yield envelop in the meridional plane. They are hence only valid in a limited range of stress level. Experiments have clearly shown that yield envelopes for almost all types of geomaterials are curved to some extent, particularly at the lower level of confining pressure. In some geotechnical problems where wide stress range will be encountered or low level of stress will play a dominant role, the nonlinearity of the trace of the yield criteria should be considered. Some nonlinear yield criteria will be reviewed in following sections.

3.7.5.1 **Nonlinear yield criterion in the Mohr plane**

Yield criteria expressed in the Mohr plane are of interests due to the popularity of the LEM. In addition, yield criteria expressed in the Mohr plane are usually interpreted with rich physical meaning. For instance, the two materials constants in the Mohr-Coulomb model are interpreted as the “cohesion” and “friction angle”.

Nonlinear failure envelopes on the Mohr plane have gained great interests in the stability analysis of slopes as failure surfaces for most of cohesive slopes are shallow, consequently stress level at failure is relatively low. A power-type yield criterion in terms of the normal and shear stresses is given as eq.(3.54) (Drescher and Christopoulos 1988; Zhang and Chen 1987).

$$\tau = c_0\left(1+\frac{\sigma_n}{\sigma_t}\right)^{\frac{1}{m}} \tag{3.54}$$

where τ and σ_n=shear and normal stresses at failure, respectively, and $\{c_0, \sigma_{\mathrm{t}}, m\}$ are determined by experimental tests. Eq.(3.54) reduces to the Mohr-Coulomb yield criterion when $m = 1$. A similar but more generalized nonlinear relation is proposed by Baker (2004) in the form

$$\tau = P_a A\left(\frac{\sigma_n}{P_a}+T\right)^{n} \tag{3.55}$$

where P_a=atmospheric pressure; $\{A, n, T\}$=nondimensional strength parameters.

Maksimovic (1989) has introduced a family of the yield criteria based on the physical model, and the idea is to use explicit expression to reflect the variation of the angle of resistance

$$\phi' = \phi'_B + \frac{\Delta\phi'}{1 + \frac{\sigma_n}{P_N}} \tag{3.56}$$

where

$$\phi' = tan^{-1}\left(\frac{\tau_f}{\sigma_n}\right)$$

where ϕ'_B= the basic angel of friction; $\Delta\phi'$= the maximum angle difference; and P_N= the median angle normal stress.

In the FELA, yield criteria expressed in the Mohr plane are difficult to work with, and they are required to be transformed to equivalent representations in principal stress or stress invariants such that the derivation of the gradient and Hessian with respect to the Cartesian stresses can be simplified. This aspect will be elaborated in chapter 7 (page 183).

3.7.5.2 **Hoek-Brown yield criterion**

Hoek-Brown yield criterion was originally proposed for underground opening works, and it has received wide acceptance over the last two decades in the analyses of highly jointed rock mass. A historical review of the development of the Hoek-Brown yield criterion can be found in the work of Hoek and Marinos (2007). When applying the Hoek-Brown yield criterion, it is important to note that the Hoek-Brown yield criterion implicitly assumes isotropic material because the yield criterion is expressed in terms of the stress variants. In other words, Hoek-Brown criterion is only valid in applications in the rock masses where there is sufficient number of closely spaced discontinuities with similar surface characteristics and randomly oriented such that the failure will not develop along any particular discontinuities or when the structure under study is large and the block size of the rock mass is small in comparison.

The latest version of Hoek-Brown yield criterion (Hoek et al. 2002) is given as

$$\sigma_1' = \sigma_3' + \sigma_{ci}\left(m_b \frac{\sigma_3'}{\sigma_{ci}} + s\right)^{\alpha} \tag{3.57}$$

Where

$$\begin{aligned} m_b &= m_i \exp\left(\frac{GSI-100}{28-14D}\right) \\ s &= \exp\left(\frac{GSI-100}{9-3D}\right) \\ \alpha &= \frac{1}{2}+\frac{1}{6}\left(e^{-\frac{GSI}{15}} - e^{\frac{20}{3}}\right) \end{aligned}$$

where GSI=Geological Strength Index, ranging from 10 for very poor rock mass to 100 for intact rock. A descriptive chart for determining the value of GSI is given by Hoek and Marinos (2007). Parameter D is to take account for the disturbance, being 0 for undisturbed rock mass and 1 for disturbed rock mass

The trace of the Hoek-Brown yield envelope in the deviatoric plane is shown in Figure 3.8.

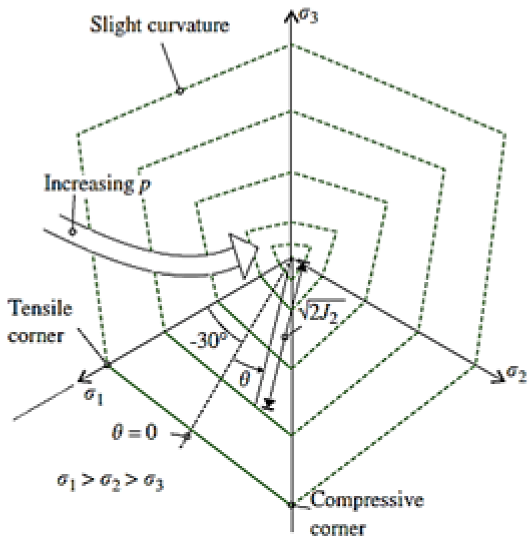


Figure 3.8 Traces of the Hoek-Brown yield criterion in the deviatoric plane with various hydrostatic pressure $p = \sigma_m$(after Clausen and Damkilde (2008))

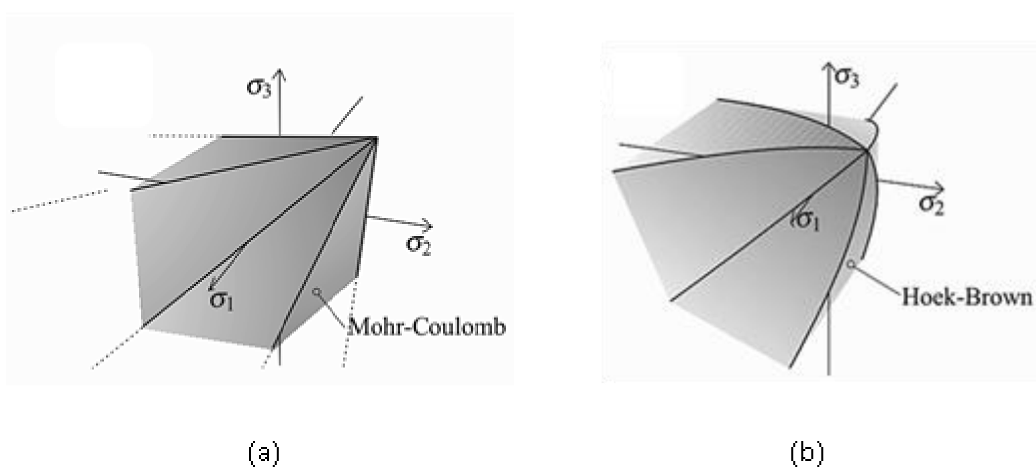


Figure 3.9 Mohr-Coulomb and Hoek-Brown yield criteria in principal stress space (a) Mohr-Coulomb and (b) Hoek-Brown (after Clausen and Damkilde (2008))

By setting $\sigma_3 = 0$ in (3.57), the uniaxial compressive strength can be obtained which is given as

$$\sigma_c = \sigma_{ci} s^{\alpha} \tag{3.58}$$

In terms of the stress variants, Hoek-Brown yield criterion can be written as eq.(3.59).

$$f = \bar{\sigma} g(\theta) + (\bar{\sigma}\, h(\theta) + \beta \sigma_m + \xi)^{\alpha} \tag{3.59}$$

where

$$\begin{aligned} g(\theta) &= -2\cos\theta \\ h(\theta) &= -m_b \sigma_{ci}^{\frac{1-\alpha}{\alpha}} \left(\cos\theta + \frac{\sin\theta}{\sqrt{3}}\right) \\ \beta &= 3 m_b \sigma_{ci}^{\frac{1-\alpha}{\alpha}} \\ \xi &= s \sigma_{ci}^{\frac{1}{\alpha}} \end{aligned} \tag{3.60}$$

Hoek Brown yield criterion is essentially a power type yield criterion and can be used as the yield criterion for soil mass as well. In this case, the material constants can be determined from the experimental tests as suggested by Hoek and Brown (1988). Similar to the MC yield function, Hoek-Brown yield criterion also suffers from singularities in numerical analyses (see Figure 3.8 and Figure 3.9).

3.7.5.3 **Lade yield criterion**

To take into account of the curvature of the failure envelope, Lade (1977) proposed a yield criterion expressed in terms of the stress invariants I_1 and I_3, as

$$f = \left(\frac{I_1^3}{I_3} - 27\right)\left(\frac{I_1}{P_a}\right)^m - k = 0 \tag{3.61}$$

where m, k=two material constants; and P_a=atmospheric pressure in the same unit as I_1.

The value of m and k in (3.61) are determined by plotting experimental data $\left(\frac{I_1^3}{I_3} - 27\right)$ vs $\left(\frac{I_1}{P_a}\right)$ in a log-log diagram and locating the best fit line. k is the intercept of the best fit line with $\frac{P_a}{I_1} = 1$ and m is the slope of the linear fit. In the principal stress space, the shape of the Lade yield criterion is an axisymmetric bullet with the pointed axis at the origin at the stress space.

It has been shown that (3.61) predicts failure stress in cohesionless soils (Lade 1977) and normally consolidated clays (Lade and Musante 1977) with reasonably good agreement with the experimental results at various levels of confining pressure. The problem of the failure surface of (3.61) is that it does not have an asymptote that is observed for most of geomaterials at high hydrostatic pressure, therefore (3.61) is valid only in the range of hydrostatic stresses where the failure surface is not straight.

3.8 Summary

Validity of the limit analysis requires the assumption of rigid perfectly plastic material that obeys the associated flow rule. These assumptions may appear to be radical at the first glance, but they are reasonable assumptions in many cases. Firstly, that there are other sources of uncertainties involved in geotechnical engineering, e.g., data obtained from soil samples may not be representative of the actual property of soil mass in the field, errors due to the measuring technique in the field or the laboratory tests and the understanding of the geomaterial behaviour is not perfect. Secondly, simplifications introduced in the limit analysis are also implicitly assumed in the conventional stability analysis techniques, such as LEM, SLM. The applicability of the limit analysis in geotechnical engineering is not restricted to MC yield criterion as in classic limit analysis. More sophisticated convex or semi-convex yield functions can be coupled with the FELA if the significance of such incorporation overweighs the accompanied complexity. Non-associated flow rule is problematic in the optimization process, as the normality condition does not hold under such condition. As discussed in section 3.4.3, non-associated flow rule is generally not critical for the collapse load determination and is not considered in the present study.

CHAPTER 4: FORMULATION OF THE LOWER BOUND LIMIT ANALYSIS

4.1 Introduction

According to the lower bound theorem, a lower bound solution is sought from statically admissible stress fields that satisfy three sets of constraints: (1) stress equilibrium conditions; (2) yield conditions and (3) stress boundary conditions. Numerical implementation of the lower bound limit analysis is concerned with the construction of discretised form of equilibrium condition and the yield criteria.

4.2 Discretisation of the stress field

In order for a solution to be a lower bound to the exact solution, the stress field associated with the solution must be constructed such that it is statically admissible everywhere over the domain. However, practically it is only possible to impose the constraints on a finite number of discrete points. The domain is discretised into an assembly of finite elements with assumed interpolation function. Therefore, the choice of the type of the element should be made based on the following criteria: (1) the discretisation needs to be purely static; and (2) such types of element should be easy to implement and adequate for practical applications.

4.2.1 Description of the finite element space

The notation for the description of the finite element space follows that by Ciria (2002) for its conciseness. Let $\mathcal{T}_h$ denotes a triangulation consisting of E elements Ω^e. The boundary of an element is denoted by $\partial\Omega^e$. Let $\mathcal{E}$ be the set of all the edges in the mesh that are made up by three disjoint sets: $\mathcal{E} = \mathcal{E}^{\mathcal{O}} \cup \mathcal{E}^{\mathcal{N}} \cup \mathcal{E}^{\mathcal{D}}$, $\mathcal{E}^{\mathcal{O}} = \{\xi_e^{e'} | \xi_e^{e'} = \partial\Omega^e \cap \partial\Omega^{e'}; \forall e, e' \in \mathcal{T}_h\}$(set of discontinuities/inter-element edges), $\mathcal{E}^{\mathcal{D}} = \{\xi_e^D | \xi_e^D = \partial\Omega^e \cap \Gamma^D; \forall e \in \mathcal{T}_h\}$ (set of edges associated with velocity boundaries,), $\mathcal{E}^{\mathcal{N}} = \{\xi_e^N | \xi_e^N = \partial\Omega^e \cap \Gamma^N; \forall e \in \mathcal{T}_h\}$ (set of edges associated with stress boundaries).

Some other notations for the numbering purpose in the formulation are as follows (4.1)

$$\begin{aligned} &NPE - \text{Node Per Element} \\ &VPE - \text{Variables Per Element} \\ &NPS - \text{Variables Per Side} \\ &NYP - \text{Number of Yield Point} \end{aligned} \tag{4.1}$$

For example, the NPE in 3-noded triangular element is 3 and the corresponding VPE in plane strain analysis equals $2 \times 3 = 6$.

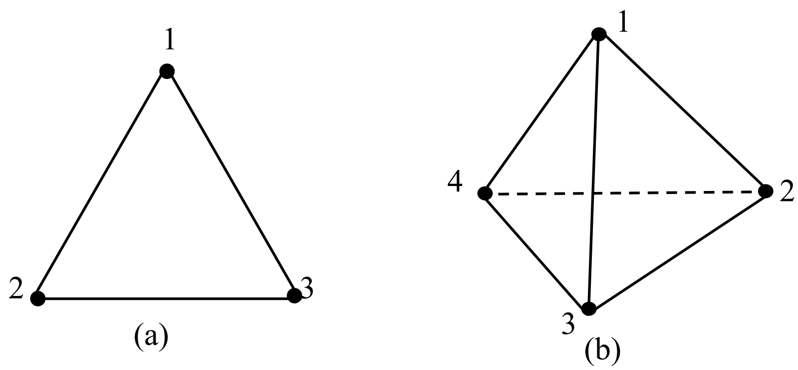


Figure 4.1 Elements used for two dimensional and the three dimensional FELA (a) 3-noded triangular element and (b) 4-noded tetrahedral element

Figure 4.1 shows the linear elements to be adopted in the finite element discretisation. Elements of higher order interpolation lead to difficulties in satisfying the yield criterion and will not be considered in this work. The low degree of interpolation will be partially compensated by introducing discontinuities into the field. Consequently, each node in the mesh will be uniquely possessed by one particular element as shown in Figure 4.2. Along the stress discontinuities, the tangential stresses are permitted to be discontinuous but continuity is required to be preserved for the normal and shear stresses, i.e., the traction along the discontinuities are required to be in equilibrium.

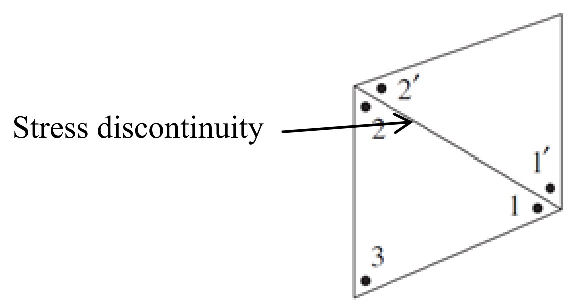


Figure 4.2 Stress discontinuity between elements

4.2.2 Discretised lower bound analysis

Given the nodal stresses, stresses at any point within an element are interpolated according to the relation (4.2).

$$\hat{\boldsymbol{\sigma}}_h^e(\mathbf{x}) = \sum_{a=1}^{NPE} N_a^e(\mathbf{x}) \hat{\boldsymbol{\sigma}}_h^{a,e} \tag{4.2}$$

where a = local numbering of the element and $N_a^e(\mathbf{x})$ = the interpolation function. The discrete form of the lower bound analysis is given in (4.3)

$$
\begin{aligned}
&\text{maximize} \qquad \alpha \\
&\text{subject to}
\begin{cases}
\nabla \cdot \hat{\boldsymbol{\sigma}}_h^e = \alpha \mathbf{f} & in\ \Omega^e, \forall e \in \mathcal{T}_h \\
\left(\hat{\boldsymbol{\sigma}}_h^e - \hat{\boldsymbol{\sigma}}_h^{e'}\right) \cdot \mathbf{n}^{\xi_e^{e'}} = 0, & \forall \xi_e^{e'} \in \mathcal{E}^O \\
\hat{\boldsymbol{\sigma}}_h^e \cdot \mathbf{n}^{\xi_e^{e'}} = \alpha \mathbf{g} & \forall \xi_e^N \in \mathcal{E}^{\mathcal{N}} \\
f(\hat{\boldsymbol{\sigma}}_h) \leq 0 & in\ \Omega^e, \forall e \in \mathcal{T}_h
\end{cases}
\end{aligned}
\tag{4.3}
$$

where $\hat{\boldsymbol{\sigma}}_h^e$ = stress field approximated with finite element space and $\mathbf{n}$ = unit normal vector.

4.3 Stress equilibrium within elements

In the implementation of the lower bound FELA in (4.3), it is convenient to adopt the vector notation that explicitly exploits the symmetry of the stress tensor. The ordering of components in a stress vector is shown in Figure 4.3. The arrow in the figure shows how the independent components are ordered and stored in a stress vector.

$$
\begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1D} \\ \vdots & \ddots & \vdots \\ \sigma_{D1} & \cdots & \sigma_{DD} \end{bmatrix} \Longrightarrow \boldsymbol{\sigma} = \begin{cases} (\sigma_{11} \quad \sigma_{22} \quad \sigma_{12})^T & \text{for } 2D \\ (\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{23} \quad \sigma_{13})^T & \text{for } 3D \end{cases}
$$

Figure 4.3 Mapping from a stress tensor to a stress vector

Substituting (4.2) into (4.3) and replacing the stresses in the tensor notation with vector notation, the discrete elemental equilibrium condition takes the form of (4.4)

$$\begin{pmatrix}\mathbf{B}_1^{\mathbf{T}} & \mathbf{B}_{\mathbf{a}}^{\mathbf{T}} & \cdots & \mathbf{B}_{\mathbf{NPE}}^{\mathbf{T}}\end{pmatrix}\begin{pmatrix}\boldsymbol{\sigma}^{1,e}\\ \boldsymbol{\sigma}^{a,e}\\ \vdots\\ \boldsymbol{\sigma}^{NPE,e}\end{pmatrix} = \alpha\begin{pmatrix}f_1\\ f_i\\ \vdots\\ f_D\end{pmatrix} \tag{4.4}$$

where in 2D case,

$$\boldsymbol{B}_a^T = \begin{pmatrix} N_{a,1}^e & 0 & N_{a,2}^e \\ 0 & N_{a,2}^e & N_{a,1}^e \end{pmatrix};$$

and in 3D case,

$$\boldsymbol{B}_a^T = \begin{pmatrix} N_{a,1}^e & 0 & 0 & N_{a,2}^e & 0 & N_{a,3}^e \\ 0 & N_{a,2}^e & 0 & N_{a,1}^e & N_{a,3}^e & 0 \\ 0 & 0 & N_{a,3}^e & 0 & N_{a,2}^e & N_{a,1}^e \end{pmatrix}.$$

and $N_{a,i} = \frac{\partial N_a}{\partial x_i}$ denotes the first derivative of the a-th shape function with respect to the i-th coordinate.

Writing (4.4) in a more compact form, we have,

$$\mathbf{A}^e \boldsymbol{\sigma}_h^e = \alpha \mathbf{F}^e \;\forall e = 1, \ldots, E \tag{4.5}$$

where $\mathbf{A}^e = \begin{pmatrix}\mathbf{B}_1^{\mathbf{T}} & \mathbf{B}_{\mathbf{a}}^{\mathbf{T}} & \cdots & \mathbf{B}_{\mathbf{NPE}}^{\mathbf{T}}\end{pmatrix}$, $(\boldsymbol{\sigma}_h^e)^T = \begin{pmatrix}\boldsymbol{\sigma}^{1,e}\\ \boldsymbol{\sigma}^{a,e}\\ \vdots\\ \boldsymbol{\sigma}^{NPE,e}\end{pmatrix}$; and $\mathbf{F}^e = \begin{pmatrix}f_1\\ f_i\\ \vdots\\ f_D\end{pmatrix}$

After assembling (4.5) over the mesh, a global matrix of stress equilibrium will be obtained and denoted as $\mathbf{A}^{eq1}$. Let $\boldsymbol{\sigma}_h$ be the assembly of $\boldsymbol{\sigma}_h^e$ and $\mathbf{F}^{\mathbf{eq}}$ be the assembly of

$\mathbf{F}^e$. Noting that each element owns its copy of variables due to the introduction of the stress discontinuities, $\boldsymbol{\sigma}_h$ has a dimension of $VPE \times E$.

As linear elements are used in the discretisation, the first derivative of interpolation function of stresses within each element is consequently constant. The elemental equilibrium condition will lead to D linear equality constraints on the stress variables. The $\mathbf{F}^{eq}$ has a dimension of $D \times E$ and $\mathbf{A}^{eq1}$ is a sparse matrix of dimensions $(D \times E, NPE \times E)$. The global constraints due the static stress equilibrium is in the form of

$$\mathbf{A}^{eq1}\boldsymbol{\sigma}_h + \alpha\mathbf{F}^{eq1} = 0 \tag{4.6}$$

where

$$\mathbf{A}^{eq1} = \begin{pmatrix} \mathbf{B}^1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ & \cdots & \cdots & \cdots & \mathbf{B}^E \end{pmatrix} \tag{4.7}$$

4.4 Stress equilibrium along discontinuities

To allow for stress discontinuities along interfaces shared by two adjacent elements, it is convenient to introduce a local coordinate system that is related to the global one by (Figure 4.4)

$$x'_k = \beta_{kj}x_j; k = 1, \dots, D \tag{4.8}$$

where β_{kj} are the direction cosines of the $x'_k -$ axes with respect to $x_j -$ axes. The traction on the plane where the normal is parallel to one of the axes x'_k is given by the vector $\mathbf{t}^k$ with component

$$t_i^k = \sigma_{ij}\beta_{jk} \tag{4.9}$$

The stress tensor represented in the local coordinate is given by

$$\sigma'_{km} = \sigma_{ij}\beta_{ki}\beta_{jm} \tag{4.10}$$

If the local coordinate system is constructed such that the normal of the discontinuity is parallel to one of its axis, then the constraints on the nodal variable due to discontinuity condition is given as

$$\sigma'^{\,e}_{km} = \sigma'^{\,e'}_{km} \tag{4.11}$$

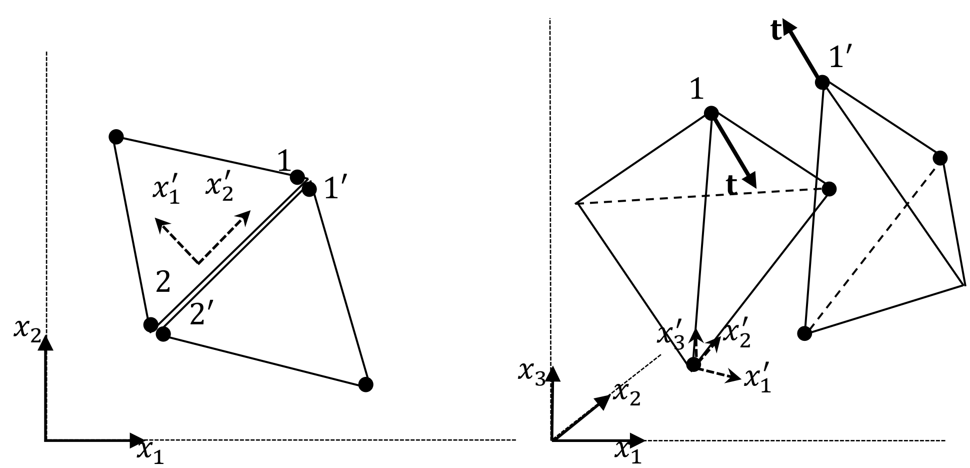


Figure 4.4 Local coordinates for stress discontinuities between two adjacent elements

Since the local coordinate is the same for both elements that share the discontinuities, (4.11) is equivalent to requirement that the tractions on the discontinuities be equal for the neighbouring as given in (4.3).

Assuming that x_1' is the axis that is parallel to the normal of the plane under consideration, the stress equilibrium along the discontinuities will give rise to the following equality constraints,

$$\mathbf{A}^{\xi_e^{e'}} \boldsymbol{\sigma}^{\xi_e^{e'}} = \mathbf{0}\ \forall \boldsymbol{\xi}_e^{e'} \in \boldsymbol{\mathcal{E}}^{\boldsymbol{O}} \tag{4.12}$$

where

$$\mathbf{A}^{\xi_e^{e'}} = \begin{pmatrix} \mathbf{T}^{1,node} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \mathbf{T}^{NPS,node} \end{pmatrix}$$

and

$$\mathbf{T}^{\text{node}} = [\mathbf{T} \quad -\mathbf{T}]$$

$$\left(\boldsymbol{\sigma}^{\xi_e^{e'}}\right)^T$$

$$= \left(\left(\boldsymbol{\sigma}^{1,\xi_e^{e'}}\right)^T \left(\boldsymbol{\sigma}^{1',\xi_e^{e'}}\right)^T \quad \left(\boldsymbol{\sigma}^{i,\xi_e^{e'}}\right)^T \left(\boldsymbol{\sigma}^{i',\xi_e^{e'}}\right)^T \quad \cdots \quad \left(\boldsymbol{\sigma}^{NPS,\xi_e^{e'}}\right)^T \left(\boldsymbol{\sigma}^{NPS',\xi_e^{e'}}\right)^T\right)$$

and

$$\mathbf{T}$$

$$= \begin{pmatrix} \beta_{11}\beta_{11} & \cdots & \beta_{1D}\beta_{D1} & \beta_{11}\beta_{21} + \beta_{12}\beta_{11} & \cdots & \beta_{11}\beta_{D1} + \beta_{1D}\beta_{11} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{11}\beta_{1D} & \cdots & \beta_{1D}\beta_{DD} & \beta_{11}\beta_{2D} + \beta_{12}\beta_{1D} & \cdots & \beta_{11}\beta_{DD} + \beta_{1D}\beta_{1D} \end{pmatrix} \tag{4.13}$$

To build the matrix **T,** the normal vector $\mathbf{n}^{\xi_e^{e'}}$ of a discontinuity (an edge for a triangular element and a triangular face for a tetrahedral element) is computed first, and the coordinate transform matrix β_{ij} is then formed.

After assembling (4.12) over all discontinuities, the global linear constraints as the result of the discontinuity equilibrium across the share boundary of adjacent elements is as follows

$$\mathbf{A}^{\mathrm{eq2}}\boldsymbol{\sigma}_h = 0 \tag{4.14}$$

where $\mathbf{A}^{\mathrm{eq2}}$ is the assembly of $\mathbf{A}^{\xi_e^{e'}}$ and has the dimensions of $(D \times VPS \times 2, VPE \times E)$.

4.5 Boundary equilibrium constraints at external stress boundary

Enforcing the stress boundary conditions is quite similar to the procedure taken for the equilibrium along stress discontinuities except that chances are boundary conditions may only be pre-specified on only one or several boundaries. For example, the symmetric condition only requires the shear stresses be equal to zero etc. Let t_p be the component of prescribed traction, $p \in P$ where P is the set of the N_p indices.

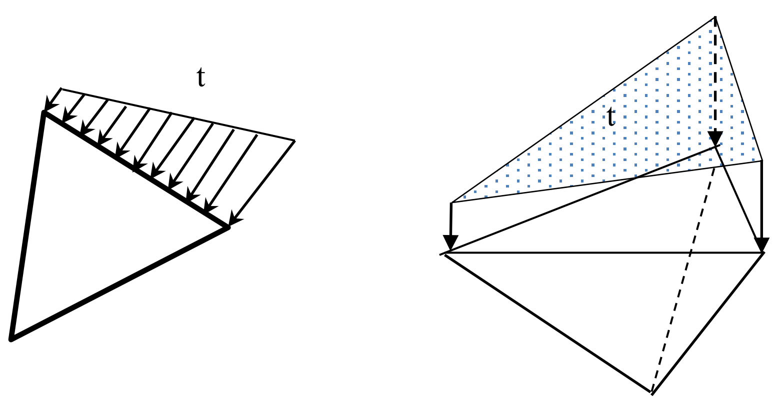


Figure 4.5 Stress boundary conditions for linear elements

Let us assume that the surface traction is applied in the local coordinate, and then the equality constraints due to stress boundary condition can be written as

$$\sigma_{ij}^{a}\beta_{ki}\beta_{jp} = t_{p}^{a} \tag{4.15}$$

The linear constraints due to the stress boundary condition are given as

$$\mathbf{A}^{\xi_e^{\mathcal{N}}}\boldsymbol{\sigma}^{\xi_e^{\mathcal{N}}} = \mathbf{b}_{bound}^{b} \tag{4.16}$$

where

$$\boldsymbol{A}^{\xi_e^{\mathcal{N}}} = \begin{pmatrix} \boldsymbol{T}^{1,p} & 0 & \cdots & 0 \\ 0 & \boldsymbol{T}^{\boldsymbol{i},p} & 0 & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & \cdots & \boldsymbol{T}^{NPS,p} \end{pmatrix},$$

and

$$\left(\boldsymbol{\sigma}^{\xi_e^{\mathcal{N}}}\right)^T = \left(\left(\boldsymbol{\sigma}^{1,\xi_e^{\mathcal{N}}}\right)^T \quad \left(\boldsymbol{\sigma}^{i,\xi_e^{\mathcal{N}}}\right)^T \quad \cdots \quad \left(\boldsymbol{\sigma}^{NPS,\xi_e^{\mathcal{N}}}\right)^T\right)$$

and $\mathbf{T}^{p}$ is the matrix extracted from (4.13) for corresponding rows, i.e. $Row(\mathbf{T^p})_i = Row(T)_{p(i)}, i = 1, \ldots N_p$, where the index function $p(i)$ returns the index of i-th prescribed component. To collect (4.16) into the global matrix, we have

$$\mathbf{A}^{\mathrm{eq3}}\boldsymbol{\sigma}_h = \mathbf{b}^{\mathrm{eq3}} \tag{4.17}$$

The total number of equations in (4.17) is $\left(NPS \times N_p \times \left|\mathcal{E}^{\mathcal{O}}\right|\right)$.

4.6 Yield function

Lower bound analysis works with a safe stress field, i.e., the stress state at any point within the domain cannot violate the yield function. Because the yield functions for the stable material are convex and only linear elements will be applied in the formulation, it is guaranteed that once the nodal stresses are constrained within the yield surface, the stress interpolated within an element will lie within the yield surface as well.

$$\boldsymbol{\sigma}_{a,h}^{e} \in B_{a\,h}^{e} \; \forall a = 1, \dots, NPE; e = 1, \dots, E \tag{4.18}$$

where $\boldsymbol{\sigma}_{a,h}^{e}$denotes the stress point at the a-th node of e-th element. It can be cast in a more compact form as

$$\boldsymbol{\sigma}_{h} \in \mathbf{B}_{h} \tag{4.19}$$

Eq.(4.18) represents a nonlinear constraints $f\left(\boldsymbol{\sigma}_{a,h}^{e}\right) \leq 0$. At the solution stage, most of the general nonlinear algorithms require the computation of the gradient and Hessian. Yield functions contribute the most complicated part of the FELA as they are nonlinear in terms of the stresses, which either need to be linearised in order to apply a linear programming algorithm or be treated as nonlinear function in their own nature to apply a nonlinear programming technique. As the nonlinear formulation has been proven to be more robust, the necessary manipulation of the yield criterion as a nonlinear programming problem will be discussed below.

4.6.1 Gradient of isotropic yield functions

For most of the nonlinear programming algorithms, computation of the gradient and Hessian of the yield function are required. One convenient method in obtaining the gradient ∇f of isotropic yield criteria is to write the gradient in the form of (4.20) (Abbo et al. 2011; Abbo and Sloan 1995; Lyamin 1999; Zienkiewicz and Pande 1997).

$$\nabla f = \frac{\partial f}{\partial \boldsymbol{\sigma}} = C_1 \frac{\partial \sigma_m}{\partial \boldsymbol{\sigma}} + C_2 \frac{\partial \bar{\sigma}}{\partial \boldsymbol{\sigma}} + C_3 \frac{\partial J_3}{\partial \boldsymbol{\sigma}} \tag{4.20}$$

where

$$\begin{aligned} C_1 &= \frac{\partial f}{\partial \sigma_m} \\ C_2 &= \frac{\partial f}{\partial \bar{\sigma}} - \frac{\tan 3\theta}{\bar{\sigma}} \frac{\partial f}{\partial \theta} \\ C_3 &= -\frac{\sqrt{3}}{2\bar{\sigma}^3 \cos 3\theta} \frac{\partial f}{\partial \theta} \\ \bar{\sigma} &= \sqrt{J_2} \end{aligned} .$$

and

$$\frac{\partial \sigma_m}{\partial \boldsymbol{\sigma}} = \frac{1}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \frac{\partial \bar{\sigma}}{\partial \boldsymbol{\sigma}} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ 2s_{12} \\ 2s_{23} \\ 2s_{13} \end{pmatrix}, \frac{\partial J_3}{\partial \boldsymbol{\sigma}} = \begin{pmatrix} s_y s_z - s_{23}^2 \\ s_x s_z - s_{13}^2 \\ s_x s_y - s_{12}^2 \\ 2(s_{23}s_{13} - s_3 s_{12}) \\ 2(s_{13}s_{12} - s_1 s_{23}) \\ 2(s_{12}s_{23} - s_2 s_{12}) \end{pmatrix} + \frac{\bar{\sigma}^2}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This treatment of isotropic yield criteria provides a standard template for the implementation. Different isotropic yield criteria can be implemented by supplying particular expressions for the three constants C_1, C_2, C_3.

4.6.2 Hessians of the yield criterion

With the expression of ∇f, the Hessian can be calculated by differentiating (4.20) and is given by

$$\nabla^2 f = \frac{\partial^2 f}{\partial \sigma^2} = \frac{\partial C_2}{\partial \sigma}\frac{\partial \bar{\sigma}}{\partial \boldsymbol{\sigma}} + C_2 \frac{\partial^2 \bar{\sigma}}{\partial \boldsymbol{\sigma}^2} + \frac{\partial C_3}{\partial \boldsymbol{\sigma}}\frac{\partial J_3}{\partial \boldsymbol{\sigma}} + C_3 \frac{\partial^2 J_3}{\partial \boldsymbol{\sigma}^2} \tag{4.21}$$

where

$$\frac{\partial^2 J_3}{\partial \boldsymbol{\sigma}} = \frac{1}{3}\begin{pmatrix} s_1 - s_2 - s_3 & & & & & \\ 2s_3 & s_1 - s_2 - s_3 & & & symmetric & \\ 2s_2 & 2s_1 & s_1 - s_2 - s_3 & & & \\ 2s_{12} & 2s_{12} & -4s_{12} & -6s_3 & & \\ -4s_{23} & 2s_{23} & 2s_{23} & 6s_{13} & -6s_1 & \\ 2s_{13} & -4s_{13} & 2s_{13} & 6s_{23} & 6s_{12} & -6s_2 \end{pmatrix}$$

and

$$\frac{\partial^2\bar{\sigma}}{\partial\boldsymbol{\sigma}^2} = \frac{1}{\bar{\sigma}}\begin{pmatrix} \frac{1}{3}-\frac{s_1s_1}{4\bar{\sigma}^2} & & & & & \\ -\frac{1}{6}-\frac{s_1s_2}{4\bar{\sigma}^2} & \frac{1}{3}-\frac{s_1s_2}{4\bar{\sigma}^2} & & & \text{symmetric} & \\ -\frac{1}{6}-\frac{s_1s_3}{4\bar{\sigma}^2} & -\frac{1}{6}-\frac{s_2s_3}{4\bar{\sigma}^2} & \frac{1}{3}-\frac{s_3s_3}{4\bar{\sigma}^2} & & & \\ -\frac{s_1s_{12}}{2\bar{\sigma}^2} & -\frac{s_2s_{12}}{2\bar{\sigma}^2} & -\frac{s_3s_{12}}{2\bar{\sigma}^2} & 1-\frac{s_{12}s_{12}}{\bar{\sigma}^2} & & \\ -\frac{s_1s_{23}}{2\bar{\sigma}^2} & -\frac{s_2s_{23}}{2\bar{\sigma}^2} & -\frac{s_3s_{23}}{2\bar{\sigma}^2} & -\frac{s_{23}s_{12}}{2\bar{\sigma}^2} & 1-\frac{s_{23}s_{23}}{\bar{\sigma}^2} & \\ -\frac{s_1s_{13}}{2\bar{\sigma}^2} & -\frac{s_2s_{13}}{2\bar{\sigma}^2} & -\frac{s_3s_{13}}{2\bar{\sigma}^2} & -\frac{s_{13}s_{12}}{2\bar{\sigma}^2} & -\frac{s_{13}s_{23}}{2\bar{\sigma}^2} & 1-\frac{s_{13}s_{13}}{\bar{\sigma}^2} \end{pmatrix}$$

4.6.3 Smoothing of the Mohr-Coulomb envelope

Many failure criteria used in geotechnical engineering are featured with corners and the apex which unfortunately give rise to singularities in the computation of derivatives with respect to the stresses, for example the well-known Mohr-Coulomb and Hoek-Brown yield criterion (Hoek and Brown 1988; Hoek et al. 2002). Singularities are required to be addressed because the stresses state lying at or near to these discontinuities is not uncommonly encountered. Generally two measures can be taken to tackle this problem: (1) treat the yield function as a multi-surface yield criterion using the formulation of Koiter (1953) and; (2) use a global or a local smoothing technique to round off the corners and apex (Abbo and Sloan 1995; Hassiotis and Xiong 2007; Sloan and Booker 1986; Zienkiewicz and Pande 1997). In this research, the smoothing techniques are adopted to remove the singularities resulting from the corners and apex the MC yield criterion.

For the MC yield criterion, the removal of the singularities arising from corners and the apex can be carried out independently in the meridional plane and octahedral plane respectively. The singularity due to the apex can be rounded off by introducing the approximation to the MC yield trace in the meridional plane. To this end, various approximations have been discussed in the works by Zienkiewicz and Pande (1997). A hyperbolic approximation that has been widely adopted (Abbo and Sloan 1995; Hassiotis and Xiong 2007; Lyamin and Sloan 2002a) is shown in Figure 4.6. The advantage in using a hyperbolic approximation is that the approximation can be improved as good as possible by adjusting the parameter a in eq.(4.22), and the approximation asymptotes rapidly to the original MC yield envelope as the hydrostatic stresses increases.

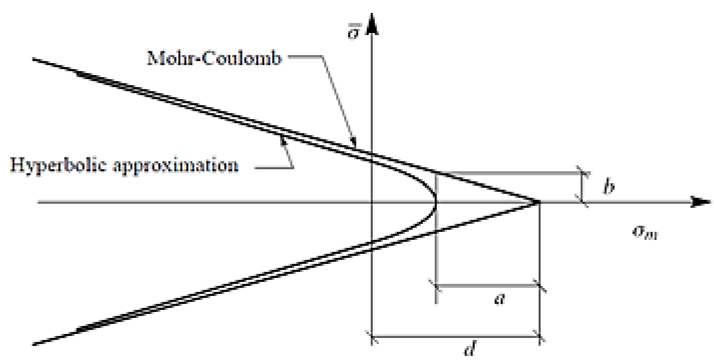


Figure 4.6 Hyperbolic smoothing of Mohr-Coulomb yield criterion in the meridional plane

In the meridional plane, the general expression of a hyperbolic shape takes the form

$$\frac{(\sigma_m - d)^2}{a^2} - \frac{\bar{\sigma}^2}{b^2} = 1 \tag{4.22}$$

Equating the slope and intercept of the Mohr Coulomb trace to those of the hyperbolic approximation leads to the following two relations

$$\frac{b}{a} = \frac{\sin \phi'}{K(\theta)}, d = c \cot \phi' \tag{4.23}$$

Note that (4.22) has two branches and only one of them will be used for the approximation. Substituting the expressions (4.23) in (4.22) gives the yield surface

$$f = \sigma_m + \sqrt{\bar{\sigma}^2 K^2(\theta) + a^2 \sin \phi'} - \mathrm{c} \cos \phi' \tag{4.24}$$

The degree of the accuracy of (4.24) is controlled by the parameter a. Lyamin (1999) suggested that the parameter a can be related to two parameters as

$$a = \beta c_{min} \cot \phi' \tag{4.25}$$

where $\beta \in (0,1)$ is a contracting parameter that ensures that the hyperbola lies inside the Mohr Coulomb envelope and c_{min} is a minimum threshold value for computing the actual cohesion according to $c = \max[c\ c_{min}]$. $\beta = 0.5$ and $c_{min} = 10^{-3}$ are found suitable for most of the applications. Figure 4.7 gives MC yield functions and the corresponding hyperbolic approximation with various values of cohesions.

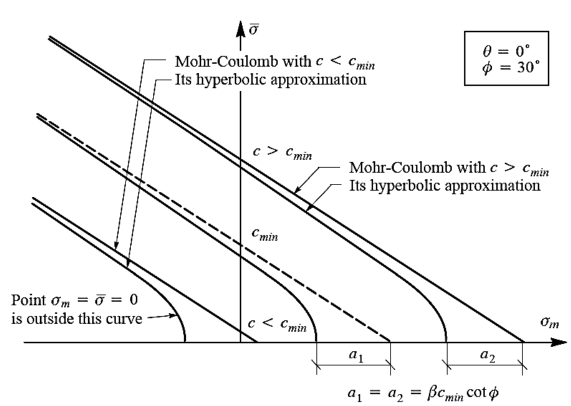


Figure 4.7 Hyperbolic approximation to the MC yield function in the meridional plane

4.6.3.1 **Remove the corners in the $\boldsymbol{\pi}-$ plane**

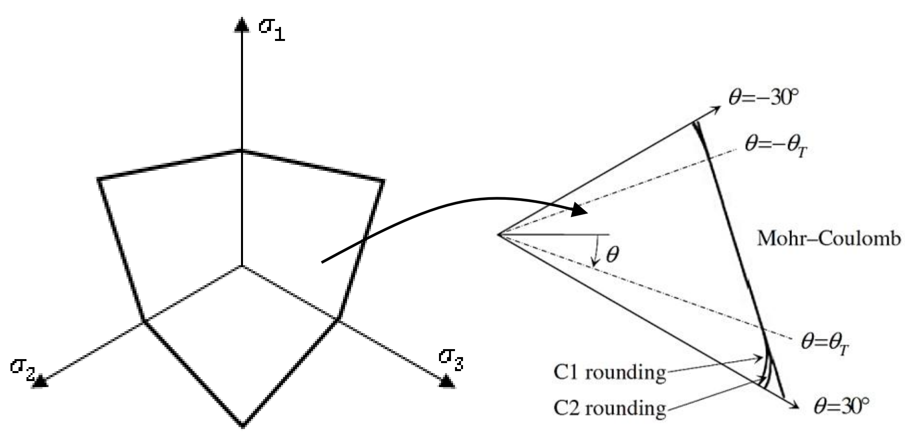
To remove the corners of the MC envelope in the π-plane, the procedure proposed by Sloan and Booker (1986) is adopted in that $K(\theta)$ is replaced when the stress state lies in the vicinity of the corner (see Figure 4.8). C^1 function that can be used to replace $K(\theta)$ is required to satisfy the following properties:

(1) At a transition point $\theta = \pm\theta_T$, the function value is identical to the original.

(2) At a transition point, $\frac{\partial\bar{\sigma}}{\partial\theta}$ should be identical to the original.

(3) Most importantly, the modified yield surface most be convex and $\frac{\partial\bar{\sigma}}{\partial\theta} = 0$ at the corners, i.e. $\theta = \pm\frac{\pi}{6}$

In addition to the three requirements for the first order continuous smoothing technique proposed by Sloan (1986), more recently Abbo et al. (2011) has extended the discussion and considered a C^2 continuous approximation to the MC yield criterion in the deviatoric plane, i.e. continuity of the second derivative is required to be preserved at the transition point.



(a) (b)

Figure 4.8 (a) MC yield criterion on the deviatoric plane and (b) rounding of MC yield criteiron on deviatoric plane ($\phi' = 30^0, \theta_T = 20^0$) (after Abbo et al. (2011))

A suitable function that potentially meets all the requirements as stated above is given by

$$K(\theta) = A + B\sin 3\theta + C\sin^2 3\theta \tag{4.26}$$

It could be verified that (4.26) satisfies the condition (3) automatically. The three unknowns could be solved from three equations resulting from equality conditions at the transition point for function values, first derivative, and second derivative.

The function $K(\theta)$ is defined piecewise as

$$K(\theta) = \begin{cases} A + B\sin 3\theta + C\sin^2 3\theta & |\theta| > \theta_T \\ \cos\theta - \dfrac{1}{\sqrt{3}}\sin\phi\sin\theta & |\theta| \le \theta_T \end{cases} \tag{4.27}$$

The first derivative of (4.27) with respect to θ is given by

$$\frac{\partial K}{\partial \theta} = \begin{cases} 3B\cos 3\theta + 3C\sin 6\theta & |\theta| > \theta_T \\ -\sin\theta - \dfrac{1}{\sqrt{3}}\sin\phi'\cos\theta & |\theta| \le \theta_T \end{cases} \tag{4.28}$$

The second derivative of (4.27) is given by

$$\frac{\partial^2 K}{\partial \theta^2} = \begin{cases} -9B\sin 3\theta + 18C\cos 6\theta & |\theta| > \theta_T \\ -\cos\theta + \dfrac{1}{\sqrt{3}}\sin\phi'\sin\theta & |\theta| \le \theta_T \end{cases} \tag{4.29}$$

By equating the expressions in (4.27), (4.28) and (4.29) at the transition point $|\theta| = \theta_T$, constants A, B and C can be solved easily. C in (4.26) is given as

$$C = C_1 + C_2\langle\theta\rangle \tag{4.30}$$

where

$$C_1 = \frac{-\cos 3\theta_T \cos\theta_T - 3\sin 3\theta_T \sin\theta_T}{18\cos^3 3\theta_T}$$
$$C_2 = \frac{1}{\sqrt{3}}\left(\frac{\cos 3\theta_T \sin\theta_T - 3\sin 3\theta_T \cos\theta_T}{18\cos^3 3\theta_T}\right)$$

and

$$B = B_1\langle\theta\rangle + B_2 \sin\phi' \tag{4.31}$$

where

$$B_1 = \frac{\cos\theta_T \sin 6\theta_T - 6\cos 6\theta_T \sin\theta_T}{18\cos^3 3\theta_T}$$
$$B_2 = \frac{-(\sin\theta_T \sin 6\theta_t + 6\cos 6\theta_T \cos\theta_T)}{18\sqrt{3}\cos^3 3\theta_T}$$

Substituting (4.30) and (4.31) back to (4.26) gives an expression of A with the form

$$A = A_1 + A_2\langle\theta\rangle \tag{4.32}$$

where

$$A_1 = \cos\theta_T - B_1 \sin 3\theta_T - C_1 \sin^2 3\theta_T$$
$$A_2 = -\frac{1}{\sqrt{3}}\sin\theta_T - B_2 \sin 3\theta_T - C_2 \sin^2 3\theta_T$$

It should be noted that in numerical implementations, θ_T can be treated as constant, e.g. $\theta_T = 29.5^0$. $A_1, A_2, B_1, B_2, C_1,$ and C_2 will then become constant and the expressions can be greatly simplified.

4.6.3.2 **Rounded Mohr Coulomb yield function**

The coefficients in (4.20) corresponding to the MC yield criterion are obtained by differentiating (3.39) with respect to the three stress variants as,

$$C_1^{rmc} = sin\,\phi', C_2^{rmc} = K - tan\,3\theta \frac{\partial K}{\partial \theta}, C_3^{rmc} = -\frac{\sqrt{3}}{2\bar{\sigma}\,cos\,3\theta}\frac{dK}{d\theta} \tag{4.33}$$

As $\theta \to 30^0$, $\tan 3\theta \to \infty$ and $1/\cos 3\theta \to \infty$, calculation of the constants C_2^{rmc} and C_3^{rmc}in (4.33) would encounter numerical problems under this case. The yield function and the gradient $K(\theta)$ and $dK/d\theta$ will be replace by (4.27) and (4.28). The constants are now given as

$$C_1^{rmc} = \sin\phi' \tag{4.34}$$

$$C_2^{rmc} = \begin{cases} A - 2B\sin 3\theta - 5C\sin^2 3\theta & |\theta| > \theta_T \\ K - \frac{dK}{d\theta}\tan 3\theta & |\theta| \le \theta_T \end{cases} \tag{4.35}$$

$$C_3^{rmc} = \begin{cases} -\frac{3\sqrt{3}}{2\bar{\sigma}}(B + 2C\sin 3\theta) & |\theta| > \theta_T \\ -\frac{\sqrt{3}}{2\bar{\sigma}\cos 3\theta}\frac{dK}{d\theta} & |\theta| \le \theta_T \end{cases} \tag{4.36}$$

4.6.3.3 **Hyperbolic approximation to Mohr Coulomb criterion**

Constants in (4.20) corresponding to the hyperbolic approximation can be obtained by differentiating (4.24), and they can be expressed in a concise way by introducing an auxiliary parameter defined in (4.37).

$$\alpha = \frac{\bar{\sigma}K}{\sqrt{\bar{\sigma}^2 K^2 + a^2 \sin^2 \phi'}} \tag{4.37}$$

Coefficients for the hyperbolic smoothed yield function are then given as

$$C_1^h = C_1^{rmc},\ C_2^h = \alpha C_2^{rmc},\ C_3^h = \alpha C_3^{rmc} \tag{4.38}$$

4.6.3.4 **Rounded coefficients for Hessian**

$$\begin{aligned} \frac{\partial C_2^{rmc}}{\partial \sigma} &= \frac{\partial \theta}{\partial \sigma}\left(\frac{dK}{d\theta} - \frac{d^2K}{d\theta^2}\tan 3\theta - 3\frac{dK}{d\theta}\sec^2 3\theta\right) \\ \frac{\partial C_3^{rmc}}{\partial \sigma} &= -\frac{\sqrt{3}}{2\bar{\sigma}\cos 3\theta}\left[\frac{\partial \theta}{\partial \sigma}\left(\frac{d^2K}{d\theta^2} + 3\frac{dK}{d\theta}\tan 3\theta\right) - \frac{2}{\bar{\sigma}}\frac{dK}{d\theta}\frac{\partial \bar{\sigma}}{\partial \sigma}\right] \end{aligned} \tag{4.39}$$

where

$$\frac{\partial \theta}{\partial \sigma} = -\frac{\sqrt{3}}{2\bar{\sigma}^3\cos 3\theta}\left(\frac{\partial J_3}{\partial \sigma} - \frac{3J_3}{\bar{\sigma}}\frac{\partial \bar{\sigma}}{\partial \sigma}\right)$$

By substituting the expression for K, the derivatives of the constants are given as

$$\frac{\partial C_2^{mc}}{\partial \sigma} = \begin{cases} -6\cos 3\theta\,(B + 5C\sin 3\theta)\dfrac{\partial\theta}{\partial\sigma} & |\theta| > \theta_T \\ \dfrac{\partial\theta}{\partial\sigma}\left(\dfrac{dK}{d\theta} - \dfrac{d^2K}{d\theta^2}\tan 3\theta - 3\dfrac{dK}{d\theta}\sec^2 3\theta\right) & |\theta| \le \theta_T \end{cases}$$
$$\frac{\partial C_3^{mc}}{\partial \sigma} = \begin{cases} \dfrac{3\sqrt{3}}{\bar{\sigma}^3}\left[-3C\,\bar{\sigma}\cos 3\theta\dfrac{\partial\theta}{\partial\sigma} + (B + 2C\sin 3\theta)\dfrac{\partial\bar{\sigma}}{\partial\sigma}\right] & |\theta| > \theta_T \\ -\dfrac{\sqrt{3}}{2\bar{\sigma}\cos 3\theta}\left[\dfrac{\partial\theta}{\partial\sigma}\left(\dfrac{d^2K}{d\theta^2} + \dfrac{3dK}{d\theta}\tan 3\theta\right) - \dfrac{2}{\bar{\sigma}}\dfrac{dK}{d\theta}\dfrac{\partial\bar{\sigma}}{\partial\sigma}\right] & |\theta| \le \theta_T \end{cases} \tag{4.40}$$

4.6.3.5 **Hyperbolic yield criterion**

The derivative of the coefficients for the hyperbolic yield surface can be expressed conveniently in terms of the Mohr Coulomb coefficients and their derivatives according to

$$\begin{aligned} \frac{\partial C_2^h}{\partial\sigma} &= \alpha\frac{\partial C_2^{rmc}}{\partial\sigma} + C_2^{rmc}\frac{\partial\alpha}{\partial\sigma} \\ \frac{\partial C_3^h}{\partial\sigma} &= \alpha\frac{\partial C_3^{rmc}}{\partial\sigma} + C_3^{rmc}\frac{\partial\alpha}{\partial\sigma} \end{aligned} \tag{4.41}$$

where (Krabbenhoft et al. 2007a; Makrodimopoulos 2010)

$$\frac{\partial\alpha}{\partial\sigma} = \frac{1-\alpha^2}{\sqrt{\sigma^2K^2 + a^2\sin^2\theta}}\left(\frac{\partial\bar{\sigma}}{\partial\sigma}K + \bar{\sigma}\frac{dK}{d\theta}\frac{\partial\theta}{\partial\sigma}\right)$$

4.6.4 Variable transformation in conic programming

It has been shown (Krabbenhoft et al. 2007a; Makrodimopoulos 2010) that by casting the MC yield criterion in the conic form, the singularities resulting from the corners and apex will not pose any difficulties in the solution of the resulting optimization problem by algorithms specialized for the conic programming, and smoothing techniques as described in the previous section will not be necessary. The MC yield criterion for plane strain analysis and the full 3D analysis can be formulated as SOCP and SDP programming respectively. However the SDP formulation for 3D does not seem to outperform the NLP formulation as the number of elements is limited.

To formulate the problem as standard SOCP formulations, new rotated stress variables **s** will be used, which are related to the Cartesian stresses $\boldsymbol{\sigma}$ by

$$\boldsymbol{\sigma} = \mathbf{T}^{\mathbf{SOCP}} \cdot \mathbf{s} \tag{4.42}$$

where **T**= transforming matrix and **s**= rotated variables used in the standard second order cone formulation that forms a second order cone and $\boldsymbol{\sigma}$ =stress vectors

Under the plane strain condition, the following relations can be obtained by expanding (3.32) stress variables

$$\begin{aligned} \sigma_m &= \frac{1}{2}(\sigma_{11} + \sigma_{22}) \\ s_{11} &= \frac{1}{2}(\sigma_{11} - \sigma_{22}) \\ s_{12} &= \sigma_{12} \end{aligned} \tag{4.43}$$

Writing (4.43) in the matrix form yields

$$\boldsymbol{\sigma} = \mathbf{C} \cdot \tilde{\boldsymbol{\sigma}} \tag{4.44}$$

where

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.45}$$

and $\tilde{\boldsymbol{\sigma}} = (\sigma_m \quad s_{11} \quad s_{12})^{\mathrm{T}}$.

It has been shown in (3.44) that by using the rotated variables, the yield criterion could be formed as the second order cone constraint simply by introducing a new variable. $\tilde{\boldsymbol{\sigma}}$ will be used under the plane strain condition in the present study.

Likewise, according to (3.32), the transformation matrix in (4.44) for 3D analysis is given as

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.46}$$

It is more convenient to apply the stress vector defined in (4.47) where $(s'_{11} \quad s'_{22} \quad s'_{12} \quad s'_{23} \quad s'_{13})$ are deviatoric stress variables. A rotated stress vector is adopted as given in (4.47).

$$\mathbf{s} = (\sigma_m \quad s'_{11} \quad s'_{22} \quad s'_{12} \quad s'_{23} \quad s'_{13})^{\mathrm{T}} \tag{4.47}$$

$\mathbf{s}$ is related to the $\tilde{\boldsymbol{\sigma}} = [\sigma_m \quad s_{11} \quad s_{22} \quad s_{12} \quad s_{23} \quad s_{13}]^{\mathrm{T}}$ by (4.48)

$$\mathbf{s} = \mathbf{R} \cdot \tilde{\boldsymbol{\sigma}} \tag{4.48}$$

Where

$$\boldsymbol{R} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{3}}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Using relations (4.44) and (4.48), the transform matrix in (4.42) associated with DP yield criterion is given by

$$\mathbf{T}^{\mathbf{SOCP}} = \mathbf{C} \cdot \mathbf{R}^{-\mathbf{1}} = \begin{pmatrix} 1 & 1 & -\frac{1}{3}\sqrt{3} & 0 & 0 & 0 \\ 1 & 0 & \frac{2}{3}\sqrt{3} & 0 & 0 & 0 \\ 1 & -1 & -\frac{1}{3}\sqrt{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.49}$$

4.7 The global optimization problem

The global optimization problem can be written in a compact form for the lower bound analysis as

$$\alpha^{LB} = \text{maximize} \quad \alpha$$
$$\text{subject to} \begin{cases} \begin{pmatrix} \mathbf{A}^{\mathrm{eq1}} \cdot \mathbf{T}^{\mathrm{SOCP}} & \vdots & \mathbf{b}^{\mathrm{eq1}} & \vdots & \mathbf{0} \\ \mathbf{A}^{\mathrm{eq2}} \cdot \mathbf{T}^{\mathrm{SOCP}} & \vdots & 0 & \vdots & \mathbf{0} \\ \mathbf{A}^{\mathrm{eq3}} \cdot \mathbf{T}^{\mathrm{SOCP}} & \vdots & \mathbf{b}^{\mathrm{eq3}} & \vdots & \mathbf{0} \\ \mathbf{A}^{\mathrm{auxi}} & \vdots & \mathbf{0} & \vdots & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{s}_h \\ \alpha \\ \boldsymbol{z_h} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{k} \end{pmatrix} \\ \boldsymbol{\sigma}_h \; free, \alpha, free, \\ \begin{cases} (z_i, \mathbf{s}_{h_{2:NS}}) \in \mathcal{C}_i \; i = 1, \dots, \mathrm{NYP} \;\; \text{if (SOCP)} \\ f(\boldsymbol{\sigma}_\mathbf{i}) < 0, i = 1, \dots NYP \;\text{if (NLP)} \end{cases} \end{cases} \tag{4.50}$$

where

$$\widetilde{\boldsymbol{T}}^{SOCP} = \begin{pmatrix} \boldsymbol{T}_1^{SOCP} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{T}_2^{SOCP} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ & \cdots & \cdots & \cdots & \boldsymbol{T}_{NY}^{SOCP} \end{pmatrix}$$
$$\boldsymbol{A}^{auxi} = \begin{pmatrix} (\boldsymbol{a}, \mathbf{0}, \mathbf{0}, \dots) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{a}, \mathbf{0}, \mathbf{0}, \dots) & \mathbf{0} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ & \cdots & (\boldsymbol{a}, \mathbf{0}, \mathbf{0}, \dots) \end{pmatrix} \tag{4.51}$$

where NYP = the number of yield points that are needed in the mesh, e.g. in the 3-node triangular mesh, $NYP = 3 \times E$.

For numerical implementation, $\mathbf{b}^{\mathrm{eq1}}$ and $\mathbf{b}^{\mathrm{eq3}}$ are subjected to the load multiplier α depending on the type of the problem under consideration. For instance, in case of the slope stability analysis where the gravity of soil mass is treated as the cause of the failure; $\mathbf{b}^{\mathrm{eq1}}$ will be acted upon by the multiplier α, while $\mathbf{b}^{\mathrm{eq3}}$ will be regarded as the dead load. In the case of the bearing capacity for a foundation or earth pressure on the retaining wall, the traction specified on the stress boundary is of interests and will be subjected to the multiplier. Furthermore, in (4.50), the objective function comprises only the load multiplier, which is in some cases not adequate for the analysis. In determining the bearing capacity of rigid strip footing for which the pressure under the footing is not uniformly distributed, it is difficult to prescribe a load profile and apply a load multiplier as an objective function. Instead, it is more convenient to optimize the applied load. This aspect will be elaborated more in chapter 7 (Page 169).

In (4.50), auxiliary variables z_i are introduced only in the formulation of the SOCP; therefore, the last row in the matrix of (4.50) need to be eliminated when the lower bound analysis is formulated as a general NLP, for which the Cartesian stresses are used directly and the transformation matrix $\mathbf{T}^{\mathrm{SOCP}}$ will be set as the identity matrix.

4.8 Extension elements

Many geotechnical stability problems are featured with a semi-infinite domain, which requires the stress field to be extended to the semi-infinite space to ensure a rigorous lower bound solution. This can be achieved by using extension elements in numerical analysis. Detailed discussion regarding the extension elements is presented in the works by (Lyamin 1999; Sloan 1988) and more recently by Makrodimopoulos and Martin (2006). In the present study, the formulation by Makrodimopoulos and Martin (Makrodimopoulos and Martin 2006)2006) will be adopted, in which slight modifications to Sloan's (1988) formulation are made to guarantee the rigorous lower bound.

As suggested by Lyamin (1999), a model with extension elements should be performed at least twice, with and without extension elements to confirm the accuracy of the results. An inadequate model, particularly when the discretised domain does not cover the whole plastic zone, may appreciably underestimate the true collapse load. It has been noted in this work that when the model is sufficiently large (similar argument for constructing numerical models for finite element method applies), the introduction of the extension element does not noticeably affect the results.

It is considered that the introduction of the extension elements is of importance in a sense to guarantee the rigorousness of the solution. However, in this work, it has been found that the effects of the extension elements are minimal when the model has been reasonably prepared to span the potential plastic zone. In practical application, it is preferable to consider the zone of influence *a priori* instead of relying on extension elements for obtaining a lower bound solution for practical purposes (to avoid the extra efforts required in the analysis).

4.9 Summary

Lower bound formulation of the FELA as general NLP and SOCP are discussed in this chapter. With the assumptions of finite element interpolation and the extension elements, the stress field throughout the whole domain is guaranteed to be statically admissible. Stress equilibrium conditions within elements and along discontinuities need to carefully dealt with to obtain a purely static discretisation.

If formulated as a general nonlinear programming problem, yield functions are required to be smoothed so that the derivatives with respect to the stresses are well defined, otherwise smoothing/ rounding techniques are necessary to obtain an approximate solution. The SOCP formulation for plane strain analysis for Mohr Coulomb material and Drucker-Prager material in three-dimensional analysis have been used to bypass the difficulties arising from the discontinuities and to improve the solution efficiency

CHAPTER 5: FORMULATION OF THE UPPER BOUND LIMIT ANALYSIS

5.1 Introduction

According to the upper bound theorem in the theory of limit analysis, any load calculated by equating the internal power dissipation to the external work done associated with a kinematically admissible velocity field is greater or at least equal to the exact collapse load. Supposing that the load multiplier only applies to the surface traction $\mathbf{t}$,

$$P_{res}(\mathbf{u}) = P_{ext}\big((\mathbf{f}^0, \alpha\mathbf{t}^0), \mathbf{u}\big) = \alpha P_{ext}\big((0, \mathbf{t}), \mathbf{u}\big) + P_{ext}\big((\mathbf{f}^0, 0), \mathbf{u}\big) \tag{5.1}$$

Assuming the loading is proportional, the upper bound analysis could be written as:

$$\begin{aligned} \alpha^{UB} = \min \quad & \Big(P_{res}(\mathbf{u}) - P_{ext}\big((\mathbf{f}^0, 0), \mathbf{u}\big)\Big) \\ \text{subject to} & \begin{cases} P_{ext}\big((0, \mathbf{t}), \mathbf{u}\big) = 1 \\ \quad \boldsymbol{\varepsilon} = \dfrac{1}{2}(\nabla\mathbf{u} + \mathbf{u}\nabla) \qquad in\ \Omega \\ \quad \mathbf{u} = 0 \qquad\qquad\qquad\quad on\ \Gamma^D \end{cases} \end{aligned} \tag{5.2}$$

The formulation in the form of (5.2) is expressed in terms of pure kinematical variables and forms the basis for most of the conventional upper bound analysis. Recall that $P_{res}(\mathbf{u}) = \int_\Omega \sup_{\boldsymbol{\sigma}\in \mathrm{B}} \boldsymbol{\sigma}\colon\boldsymbol{\varepsilon}\ d\Omega + \int_\Sigma \sup_{\boldsymbol{\sigma}\in \mathrm{B}} \boldsymbol{\sigma}\colon\boldsymbol{\chi}\ d\Sigma$, formulation (5.2) yields an upper bound only when the supremum is computed exactly and the stress variables are eliminated with the normality condition.

5.2 Rate of power dissipation

Formulation of the upper bound analysis in velocity field requires the calculation of the exact rate of power dissipation. For general nonlinear yield functions, to express the power dissipation $P_{res}(\mathbf{u})$ in terms of the velocities is not trivial, and explicit expressions exist only for particular classes of yield criteria. Before heading for the upper bound formulation of the FELA, it is worthwhile to review the conventional procedures in obtaining $P_{res}(\mathbf{u})$ as they offer intuitive insights into the pure velocity formulation of the upper bound.

5.2.1 Power dissipation in conventional upper bound limit analyses with MC yield criterion

Conventional upper bound analysis assumes a simplified failure mechanism consisting of a rigid mass moving along a prescribed slip surface (velocity discontinuity). Under such simplifications, internal power dissipation occurs along the slip surface. Given the linear nature of MC yield criterion, stress variables in the expression of the internal power dissipation can be easily eliminated. To illustrate, consider the determination of the critical height H_{cr} of a vertical cut in a cohesive soil (Figure 5.1). MC criterion in terms of the normal and shear stresses takes the form of eq.(5.3).

$$\tau = c' + \sigma_n \tan\phi' \tag{5.3}$$

where σ_n and τ = the normal and shear stresses along the failure surface and c' and ϕ' are the two material constants.

Assuming a translational failure motion, for a velocity field **u** to be kinematically admissible, the flow rule or the normality condition requires that the tangential component u_t and the normal component of u_n be related by

$$u_n = u_t \cdot \tan\phi' \tag{5.4}$$

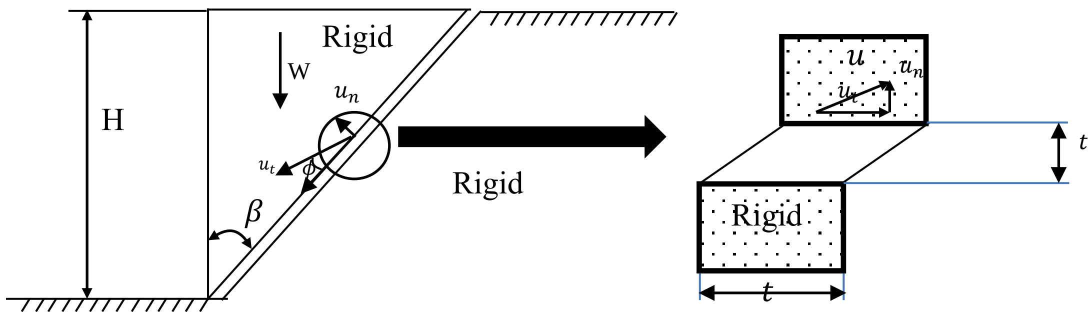


Figure 5.1 Critical height of a vertical cut

Regarding the discontinuity as a transitional layer with thickness t, (see Figure 5.1) the shear strain rate $\dot{\gamma}$ is equal to u_t/t and the normal strain rate $\dot{\varepsilon}$ is equal to u_n/t. The rate of the energy dissipation density $\pi(\mathbf{u})$ can be expressed as (compressions are taken as positive)

$$\pi(\mathbf{u}) = (\tau\dot{\gamma} - \sigma\dot{\varepsilon})t = (\tau u_t - \sigma u_n) \tag{5.5}$$

Using relation (5.4), we have $D = u_t(\tau - \sigma\tan\phi)$. Since the shear strength is assumed to be fully mobilised along the failure surface, with (5.3), it follows from (5.5) that

$$\pi(\mathrm{u}) = \mathrm{c} \cdot \mathrm{u_t} \tag{5.6}$$

It is noted that stress variables vanish in the expression in (5.6). The cancelation of the stress variables is attributed to the fact that the failure criterion (5.3) is the linear with respect to stress components for which the two material constants c' and ϕ' are independent of the stress level. This conclusion holds for the multi-blocks upper bound formulation and rigid finite element upper bound formulation. For more general classes of the yield criteria in which the nonlinearity of the failure envelope is considered, as will be shown in the section 7.6, the elimination of the stress variables will be considerably complicated..

The finite element formulation of the upper bound limit analysis using the linearised failure criterion leads to the same conclusion, i.e. the stress variables will be eliminated from the expression of the rate of power dissipation. A Lagrangian multiplier field will be needed for the fulfilment of the normality condition.

5.2.2 Power dissipation in linear programming formulation with linearised yield criteria

In the linear programming formulation of the upper bound limit analysis, the yield criterion is approximated by an external polygon (Figure 5.2) to ensure that the solutions obtained are the rigorous upper bounds. In the linearization, the MC yield function is approximated by a set of linear functions in the form of (5.7), each representing a side of the polygon.

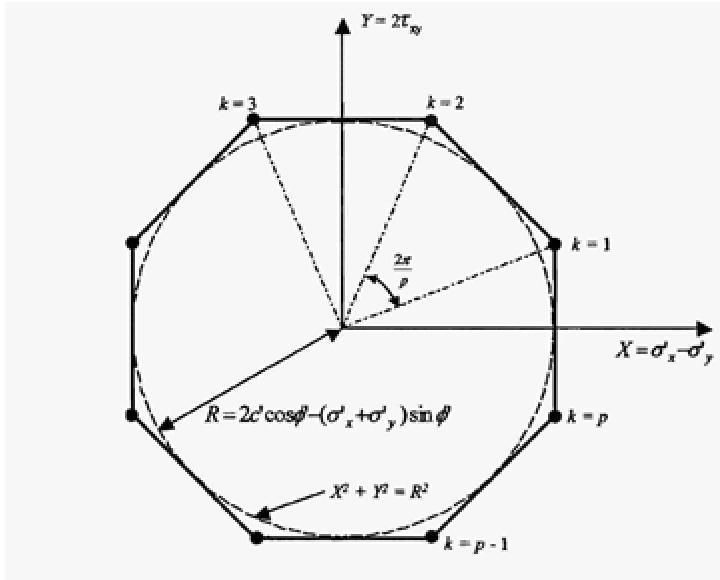


Figure 5.2 External approximation of the MC yield criterion using p sides polygon (after (Bandini 2003))

$$F_k = A_k\sigma'_k + B_k\sigma'_y + C_k\tau_{xy} - D = 0, k = 1, \cdots, p \tag{5.7}$$

where

$A_k = \cos\alpha_k + \sin\phi'$, $B_k = \sin\phi' - \cos\alpha_k$, $C_k = 2\sin\alpha_k$, $D = 2c'\cos\phi'\cos\left(\frac{\pi}{p}\right)$ and $p =$ number of sides of the polygon. This treatment of the MC yield criterion was adopted by Sloan (1989) and has become the most popular manipulation of the MC yield criterion in the LP formulation of the FELA. To simplify the discussion, we only consider the continuous velocity field. The power dissipation within an element is given as

$$P^e_{res}(\mathrm{u}) = \sup_{\boldsymbol{\sigma}\in\mathrm{B}} \int_{\Omega^e} \left(\sigma_x\dot{\varepsilon}_x + \sigma_y\dot{\varepsilon}_y + \tau_{xy}\dot{\gamma}_{xy}\right) d\Omega \tag{5.8}$$

With the normality condition, we have

$$\begin{aligned}
\dot{\varepsilon_x} &= \sum_{k=1}^{p} \lambda_k \frac{\partial F_k}{\partial \sigma_x} = \sum_{k=1}^{p} \lambda_k A_k \\
\dot{\varepsilon_y} &= \sum_{k=1}^{p} \lambda_k \frac{\partial F_k}{\partial \sigma_x} = \sum_{k=1}^{p} \lambda_k B_k \\
\dot{\gamma}_{xy} &= \sum_{k=1}^{p} \lambda_k \frac{\partial F_k}{\partial \tau_{xy}} = \sum_{k=1}^{p} \lambda_k C_k
\end{aligned} \tag{5.9}$$

where λ_k = the nonnegative Lagrange multipliers.

Substituting the linearised strain rate (5.9) into (5.8) yields

$$P_{res}^{e}(\mathbf{u}) = \int_{\Omega^{\mathrm{e}}} \sum_{k=1}^{p} \lambda_k \left(A_k \sigma_k + B_k \sigma_y + C_k \tau_{xy} \right) d\Omega \tag{5.10}$$

Using the relation (5.7), and assuming a constant strain element, the dissipated power can be written as eq.(5.11).

$$P_{res}^{e}(\mathbf{u}) = 2c \cos\phi \, A^e \sum_{k=1}^{p} \lambda_k \tag{5.11}$$

where A^e = area of an element. Unlike conventional upper bound analyses for which the power dissipation is expressed in purely kinematic variable, the Lagrangian multiplier field λ needs to be constructed in the discretisation in addition to the velocity field.

5.2.3 Power dissipation function in NLP formulation with a quadratic yield funcion

Linearzation of the yield criterion will give rise to a huge number of linear constraints (see (5.7)), which precludes its applicability to large-scale numerical analyses. A nonlinear upper bound formulation for the MC and DP materials was proposed by Li and Yu (2005) who utilized the properties of the quadratic form of yield criteria and eliminated the stress variables with the help of the associated flow rule so that the formulation is eventually represented in terms of the pure kinematic variables. A quadratic form of the yield criterion could be expressed (5.12).

$$f = \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{P}\boldsymbol{\sigma} + \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{Q} - 1 = \mathbf{0} \tag{5.12}$$

where Q =coefficient matrix, and q =coefficient vector. For Mohr-Coulomb material in plane strain condition, P and Q are given by:

$$\boldsymbol{P} = \begin{pmatrix} \dfrac{1}{4c^2} & \dfrac{-1-\sin^2\phi}{4c^2\cos^2\phi} & 0 \\ \dfrac{-1-\sin^2\phi}{4c^2\cos^2\phi} & \dfrac{1}{4c^2} & 0 \\ 0 & 0 & \dfrac{1}{c^2\cos^2\phi} \end{pmatrix} \tag{5.13}$$

$$\mathbf{Q} = \begin{pmatrix} \dfrac{\sin\phi}{c\cos\phi} & \dfrac{\sin\phi}{c\cos\phi} & 0 \end{pmatrix}^{\mathrm{T}} \tag{5.14}$$

For the Drucker Prager yield criterion, the corresponding expressions for **P** and **Q** are:

$$P = \begin{pmatrix} \frac{1-a^2}{9k^2} & -\frac{1+2a^2}{18k^2} & -\frac{1+2a^2}{18k^2} & 0 & 0 & 0 \\ -\frac{1+2a^2}{18k^2} & \frac{1-a^2}{9k^2} & -\frac{1+2a^2}{18k^2} & 0 & 0 & 0 \\ -\frac{1+2a^2}{18k^2} & -\frac{1+2a^2}{18k^2} & \frac{1-a^2}{9k^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{k^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{k^2} \end{pmatrix} \tag{5.15}$$

$$\mathbf{Q} = \begin{pmatrix} \frac{2a}{3k} & \frac{2a}{3k} & \frac{2a}{3k} & 0 & 0 & 0 \end{pmatrix}^{\mathrm{T}} \tag{5.16}$$

Now we can relate the rate of strain to the stresses by the normality condition to give

$$\boldsymbol{\varepsilon} = 2\lambda \mathbf{P}\boldsymbol{\sigma}^* + \lambda \mathbf{Q} \tag{5.17}$$

Solving (5.17) for $\boldsymbol{\sigma}$ yields,

$$\boldsymbol{\sigma} = \frac{1}{2\lambda}\mathbf{P}^{-1}\boldsymbol{\varepsilon} - \frac{1}{2}\mathbf{P}^{-1}\mathbf{Q} \tag{5.18}$$

At the instant of plastic flow, the yield criterion (5.12) must be satisfied, hence the Lagrangian multipliers λ can further be eliminated and expressed in terms of the strains. Substituting (5.18) into (5.12) and solving for λ, we have (noting that λ is nonnegative)

$$\lambda = \sqrt{\frac{\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{P}^{-1}\boldsymbol{\varepsilon}}{4 + \mathbf{Q}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{Q}}} \tag{5.19}$$

Then the dissipation density in terms of the pure kinematical variables is expressed as:

$$\begin{aligned}\pi(\boldsymbol{\varepsilon}) = \sigma_{ij}^{*}\varepsilon_{ij} &= \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon} = \left(\frac{1}{2\lambda}\mathbf{P}^{-1}\boldsymbol{\varepsilon} - \frac{1}{2}\mathbf{P}^{-1}\mathbf{Q}\right)^{\mathbf{T}}\boldsymbol{\varepsilon} \\ &= \frac{1}{2}\sqrt{(\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{P}^{-1}\boldsymbol{\varepsilon})\cdot(4+\mathbf{Q}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{Q})} - \frac{1}{2}\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{Q}\end{aligned} \tag{5.20}$$

The expression of (5.20) is a function of strains rate that can in turn be expressed in velocities. Thus a pure velocity field is sufficient for the upper bound analysis. However, power dissipation function (5.20) is not smooth and not differentiable everywhere (Li and Yu 2005). The resulting optimization problem is difficult to solve and a specialized optimization algorithm is required.

A more efficient method to formulate the quadratic form of yield criteria is to rewrite the dissipation function in a conic form and consequently the upper bound analysis is then able to be cast as the SOCP for which a lot of standard and efficient solvers are available.

5.2.4 Power dissipation in conic formulation

The intuitive idea of the conic form of yield criteria comes from the fact that they shaped like a cone. However, to apply the conic programming, it is necessary to perform rotation of variables to formulate the problem in a standard conic program. Let us define the deviatoric and spherical components of the strain tensor as:

$$\varepsilon_v = \varepsilon_{ii} \; and \; e_{ij} = \varepsilon_{ij} - \frac{\varepsilon_v}{D}\delta_{ij} \tag{5.21}$$

In numerical implementations, it is more convenient to introduce new variables in the SOCP formulation such that

$$\tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\tilde{\boldsymbol{\varepsilon}} = \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon} \tag{5.22}$$

where $\tilde{\boldsymbol{\sigma}}$ is defined in (3.50).

In plane strain analysis,

$$\tilde{\boldsymbol{\varepsilon}} = (\, \varepsilon_v \quad 2e_{11} \quad 2e_{12})^{\mathrm{T}} \tag{5.23}$$

and in the three-dimensional analysis,

$$\tilde{\boldsymbol{\varepsilon}} = (\varepsilon_v \quad (2e_{11} + e_{22}) \quad (e_{11} + 2e_{22}) \quad 2e_{12} \quad 2e_{23} \quad 2e_{13})^{\mathrm{T}} \tag{5.24}$$

To rewrite (5.23) and (5.24) in matrix form, we have

$$\tilde{\boldsymbol{\varepsilon}} = \mathbf{R} \cdot \boldsymbol{\varepsilon} \tag{5.25}$$

where in plane strain analysis

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \tag{5.26}$$

and in three dimensional analysis

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 2/3 & -1/3 & -1/3 & 0 & 0 & 0 \\ -1/3 & 2/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.27}$$

The power dissipation density is found from the solution to the maximization problem (5.28).

$$\begin{aligned} \pi(\mathbf{u}) = \sup \quad & \tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}} \\ \text{s.t.} \quad & \mathbf{b} + \mathbf{Q}\tilde{\boldsymbol{\sigma}} \in \mathcal{C}, \end{aligned} \tag{5.28}$$

where $\mathbf{b}$ and $\mathbf{Q}$ are material-dependent and are given in (3.49) and $\mathcal{C}$ = second order cone.

The dual problem of (5.28) is given as (5.29)

$$\begin{aligned} \pi(\mathbf{u}) = \inf \quad & \mathbf{b}^{\mathrm{T}}\mathbf{y} \\ \text{s.t.} \quad & \begin{cases} \tilde{\boldsymbol{\varepsilon}} + \mathbf{Q}^{\mathbf{T}}\mathbf{y} = 0 \\ \mathbf{y} \in \mathcal{C}^* \end{cases} \end{aligned} \tag{5.29}$$

where $\mathbf{y}$=dual variables corresponding to the yield criterion and $\mathcal{C}^*$ = the dual cone to $\mathcal{C}$. Since the second order cone is self-dual, i.e. $\mathcal{C}^* = \mathcal{C}$, therefore, the expression of the dissipation function can be expressed in the following way

$$P_{res}(\mathbf{u}) = \int_{\Omega} \mathbf{b}^{\mathbf{T}}\mathbf{y} \ \text{with} \begin{cases} \mathbf{y} \in \mathcal{C} & in\ \Omega, \\ \tilde{\boldsymbol{\varepsilon}} + \mathbf{Q}^{\mathbf{T}}\mathbf{y} = 0 & in\ \Omega \end{cases} \tag{5.30}$$

5.3 Pure kinematic formulation of the upper bound limit analysis

With the rate of the power dissipation given in (5.30), the generic form of the SOCP formulation of the upper bound analysis takes the form

$$\alpha^{UB} = \min\left(\int_{\Omega} \mathbf{b}^{\mathbf{T}}\mathbf{y}d\Omega - P_{ext}\big((f^{0},0),\mathbf{u}\big)\right)$$
$$\text{subject to} \begin{cases} P_{ext}\big((0,\mathbf{t}),\mathbf{u}\big) = 1 & & (a) \\ \tilde{\boldsymbol{\varepsilon}} + \mathbf{Q}^{\mathbf{T}}\mathbf{y} = \mathbf{0} & in\ \Omega & \text{(b)} \\ \mathbf{u} = 0 & on\ \Gamma^{D} & \text{(c)} \\ \mathbf{y} \in C^{*} & & \end{cases} \tag{5.31}$$

In the following sections, attention will be given to the discrete form of each item in (5.31) based on the finite element discretisation respectively

5.3.1 Discretisation of rate of the power dissipation

As shown in (5.31), the velocity field $\mathbf{u}$ with D components for each node and the Lagrangian multiplier field $\mathbf{y}$ for each flow rule point will need to be constructed. For the moment, discussion will be confined to the continuous velocity field. The velocity at any given point within the domain is given by

$$\mathbf{u}(\mathbf{x}) = \sum_{A=1}^{N} N_{A}^{e}(\mathbf{x})\mathbf{u}^{\mathrm{A}} \tag{5.32}$$

where $\mathbf{u}^{\mathrm{A}}$ =nodal values of the velocity; $N_{A}^{e}(\mathbf{x})$ =global shape function corresponding to the type of interpolation and $N =$ the number of nodes in the mesh. The variation of the equivalent strain variable $\mathbf{y}$ takes the same form as the strain variables (see (5.31) b), and is given by

$$\mathbf{y}(\mathbf{x}) = \sum_{a=1}^{NYP} r_a^e(\mathbf{x})\mathbf{y}^{a,e} \tag{5.33}$$

where NYP =flow rule points per element (or number of the yield point) and $r_a^e(x) =$ the a-th shape function corresponding to the strain rate. It should be noted that the interpolation of the strain rate field is one order lower than the velocity field (strains are the combinations of the first derivatives of displacement), therefore, it is only necessary to impose the flow rule constraints on particular points within the element as shown in Figure 5.3, where ⊗ denotes the flow rule point.

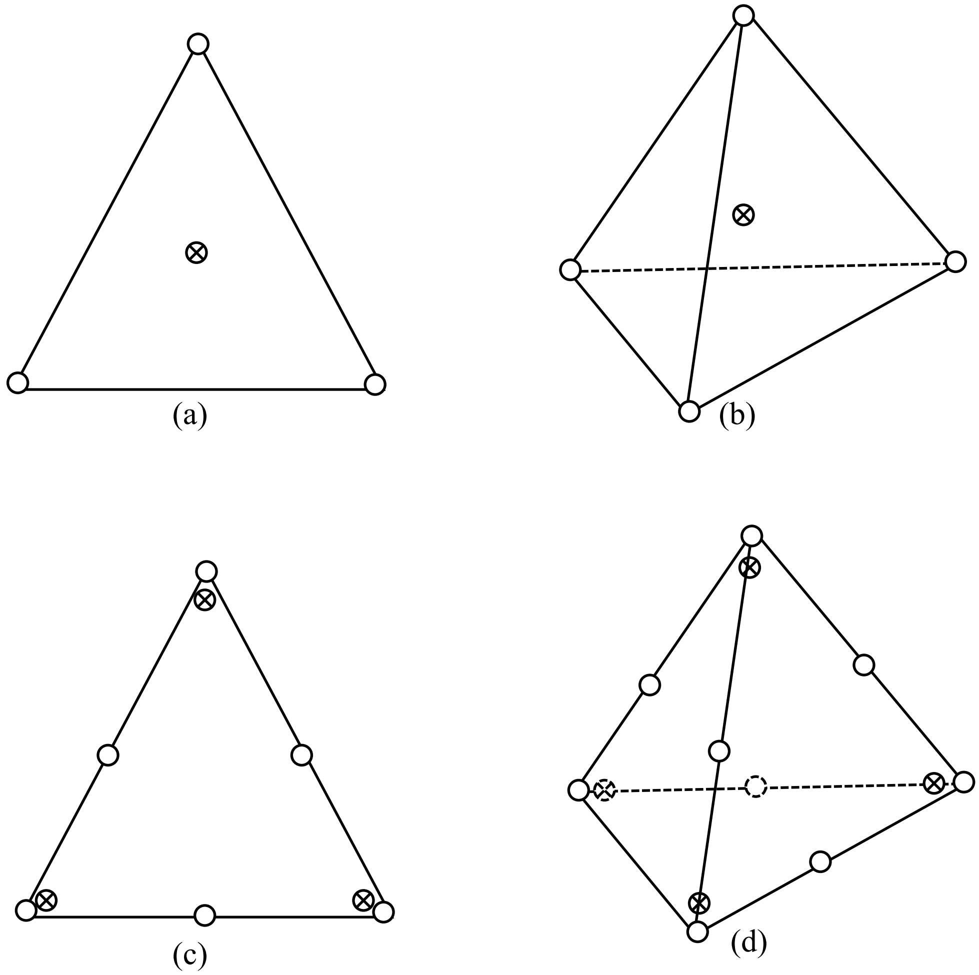


Figure 5.3 Flow rule points for constant strain elements (a, b) and simplex strain elements (c, d)

Assuming that the material vector **b** varies with locations as well, the discretised form of the power dissipation rate is obtained by substituting (5.33) into the objective function of (5.31), which is given by

$$\int_{\Omega}\left(\mathbf{b}(\mathbf{x})\right)^{\mathbf{T}}\mathbf{y}(\mathbf{x})d\Omega = \sum_{e=1}^{E}\sum_{a=1}^{YPE}\int_{\Omega^{\mathrm{e}}}\mathbf{b}^{e}(\mathbf{x})r_a^e(\mathbf{x})\mathbf{y}_a^e d\Omega$$
$$= \sum_{e=1}^{E}\underbrace{\begin{pmatrix}\mathbf{b}_1^{*,e} & \mathbf{b}_2^{*,e} & \cdots & \mathbf{b}_{FPE}^{*,e}\end{pmatrix}}_{\tilde{b}^{*,e}}\underbrace{\begin{pmatrix}\mathbf{y}_1^e\\ \mathbf{y}_2^e\\ \vdots\\ \mathbf{y}_{FPE}^e\end{pmatrix}}_{\tilde{y}^e} = \sum_{e=1}^{E}\mathbf{b}^{*,e}\mathbf{y}^e \tag{5.34}$$
$$= \mathbf{b}^T\mathbf{y}$$

where $\mathbf{b}_1^{*,e} = \int_{\Omega^e}\mathbf{b}^e(\mathbf{x})r_a^e(\mathbf{x})d\Omega$.

Consider the compatibility conditions for three-dimensional analysis, the explicit form of (5.34) is given by (5.35).

$$\boldsymbol{\varepsilon} = \begin{pmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{12}\\ 2\varepsilon_{23}\\ 2\varepsilon_{12}\end{pmatrix} = \begin{pmatrix}\dfrac{\partial u_1}{\partial x_1}\\ \dfrac{\partial u_2}{\partial x_2}\\ \dfrac{\partial u_3}{\partial x_3}\\ \dfrac{\partial u_1}{\partial x_2}+\dfrac{\partial u_2}{\partial x_1}\\ \dfrac{\partial u_1}{\partial x_1}+\dfrac{\partial u_1}{\partial x_1}\\ \dfrac{\partial u_1}{\partial x_1}+\dfrac{\partial u_1}{\partial x_1}\end{pmatrix} = \mathbf{Su} \tag{5.35}$$

With relation (5.35), it is easy to verify that the strains of a particular point in the domain are given by the nodal velocities by (5.36).

$$\boldsymbol{\varepsilon}_i = \sum_{A=1}^{N} \mathbf{B}_{\mathbf{i}}^{A}\, \mathbf{u}^{A} \qquad \forall i = 1 \ldots YP \tag{5.36}$$

where

$$\left(\mathbf{B}_i^A\right)^T = \begin{pmatrix} \frac{\partial N_A}{\partial x_1} & 0 & \frac{\partial N_A}{\partial x_2} & 0 & \frac{\partial N_A}{\partial x_3} & 0 \\ 0 & \frac{\partial N_A}{\partial x_2} & 0 & \frac{\partial N_A}{\partial x_1} & \frac{\partial N_A}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial N_A}{\partial x_3} & 0 & \frac{\partial N_A}{\partial x_2} & \frac{\partial N_A}{\partial x_1} \end{pmatrix} \tag{5.37}$$

Substituting (5.36) into (5.31) and using relation (5.25), we have

$$\mathbf{RBu} - \mathbf{Qy} = 0 \tag{5.38}$$

where

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_1^1 & \boldsymbol{B}_1^2 & \cdots & \cdots & \boldsymbol{B}_1^N \\ \boldsymbol{B}_2^1 & \boldsymbol{B}_2^2 & \cdots & \cdots & \boldsymbol{B}_2^N \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \boldsymbol{B}_{NF}^1 & \cdots & \cdots & \cdots & \boldsymbol{B}_{NF}^E \end{pmatrix}; \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q} & \cdots & \cdots & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \boldsymbol{Q} \end{pmatrix}$$

5.3.2 Discretisation of the rate of work done due to external loading

Assuming $\mathbf{f}^{\mathbf{0}}$ and $\mathbf{g}^0$ are dead loads within the solution domain and along the boundary that are independent of the load multipliers, the rate of the work done takes the form

$$W_{ext}^0(\mathbf{u}) = \int_\Omega \mathbf{f}^{\mathbf{0}} \cdot \mathbf{u}\, d\Omega + \int_{\Gamma^{\mathrm{N}}} \mathbf{g}^0 \cdot \mathbf{u}\, d\Omega \tag{5.39}$$

For simplicity of expression, the superscript o that denotes dead load will be dropped from now. Substituting (5.32) into (5.39), the discretised form of (5.39) is given as,

$$
\begin{aligned}
W_{ext}^{0}(\mathbf{u}) &= \sum_{e=1}^{E}\sum_{a=1}^{NPE} \underbrace{\begin{pmatrix} u_1^{a,e} & u_2^{a,e} & \cdots & u_D^{a,e} \end{pmatrix}}_{(\mathbf{u}^{a,e})^T} \underbrace{\begin{pmatrix} \int_{\Omega^e} f_1^e N_A^e(\mathbf{x}) d\Omega + \int_{\xi_e^N} g_1^{\xi_e^N} N_A^e(\mathbf{x})\, dS \\ \int_{\Omega^e} f_2^e N_A^e(\mathbf{x}) d\Omega + \int_{\xi_e^N} g_2^{\xi_e^N} N_A^e(\mathbf{x})\, dS \\ \vdots \\ \int_{\Omega^e} f_D^e N_A^e(\mathbf{x}) d\Omega + \int_{\xi_e^N} g_D^{\xi_e^N} N_A^e(\mathbf{x})\, dS \end{pmatrix}}_{\mathbf{F}_a^e} \\
&= \sum_{e=1}^{E}\sum_{a=1}^{NPE} ((\mathbf{u}^{a,e})^T \mathbf{F}_a^e) = \sum_{e=1}^{E} \underbrace{\begin{pmatrix} \mathbf{u}^{1,e} & \mathbf{u}^{2,e} & \cdots & \mathbf{u}^{NPE,e} \end{pmatrix}}_{\mathbf{u}^e} \underbrace{\begin{pmatrix} \mathbf{F}_1^e \\ \mathbf{F}_2^e \\ \vdots \\ \mathbf{F}_{NPE}^E \end{pmatrix}}_{\mathbf{F}^E} \\
&= \sum_{e=1}^{E} \mathbf{u}^e \mathbf{F}^E = (\mathbf{F}^0)^{\mathbf{T}} \mathbf{u}
\end{aligned}
\tag{5.40}
$$

Likewise, the work due to the live load takes similar format as (5.40)

$$
w_{ext}(\mathbf{u}) = \mathbf{F}^{\mathbf{T}}\mathbf{u} \tag{5.41}
$$

5.3.3 The global optimization problem

Now we are ready to collect (5.31) as the standard SOCP which is given as

$$
\begin{aligned}
&\min && \mathbf{b}_h^{*T}\mathbf{y}_{\boldsymbol{h}} + (\mathbf{F}_h^0)^{\mathbf{T}}\mathbf{u}_{\boldsymbol{h}} \\
&\text{subject to} && \begin{cases} \mathbf{F}_{\boldsymbol{h}}{}^{\mathbf{T}}\mathbf{u}_{\boldsymbol{h}} = \mathbf{1} \\ \mathbf{RBu_h} - \mathbf{Qy}_{\boldsymbol{h}} = 0 \\ \mathbf{y}_{\boldsymbol{h}_i} \in \mathcal{C}_i \quad \forall i = \{1 \dots NF\} \end{cases}
\end{aligned}
\tag{5.42}
$$

The dual problem to (5.42) is

$$\begin{aligned} &\max\ \alpha \\ &\text{suject to } \begin{cases} \mathbf{B}^{\mathrm{T}}\bar{\boldsymbol{\sigma}} - \alpha\mathbf{F} = \mathbf{F}^{0} \\ \\ \tilde{\mathbf{b}}_i^{*} + \mathbf{Q}\bar{\boldsymbol{\sigma}}_i \in \mathcal{C}^{*} \quad \forall i = 1, \dots, NF \end{cases} \end{aligned} \tag{5.43}$$

where $\bar{\boldsymbol{\sigma}}_i$ = the dual variables corresponding to $\mathbf{y}_i$ for one particular flow rule point. It is noted by many researchers (Krabbenhoft et al. 2005; Makrodimopoulos and Martin 2007) that solving (5.43) is computationally more efficient than the original form (5.42) when using the primal-dual interior point algorithm.

It is worth noting that (5.43) resembles the form of the lower bound analysis in that the equality constraints can be regarded as the stress equilibrium and the conic constraints impose the yield criterion. Inspired by this, it may be worthwhile and more efficient that the upper bound problem be formulated directly from a dual point of view. More importantly, the tedious derivation for the expression of the rate of the power dissipation is avoided. This will be elaborated in the following section.

Up to this point, the velocity field has been assumed continuous. Velocities are of interests for the following two reasons. Firstly, in the discretisation with constant strain elements, the interpolation cannot provide sufficient degrees of freedom to satisfy the incompressibility condition. Elements require a special arrangement, in which four triangles forms a quadrilateral with the central node lying at the intersection of the diagonals. Secondly, the introduction of velocity discontinuities gives rise to additional degrees of freedom, and this can compensate for the low order of interpolation to a certain extent. If prior knowledge is known for a particular problem, inclusion of the discontinuities can be of great benefit in the computation. However it should be noted that the use of discontinuities will complicate the formulation and increase the number of variables in the resulting optimization problem.

5.4 Formulation of the upper bound limit analysis with stress variables

The departure point of the conventional velocity formulations of upper bound limit analysis as discussed is to eliminate stress variables and express the maximum power dissipation in pure velocities. This procedure is simple in concept and has been well developed for the classic upper bound analysis, particularly in the method with the rigid blocks discretisation. However, removing stress variables in the expression of rate of power dissipation is limited to particular classes of yield criteria. Generalization of this technique to more sophisticated yield envelopes is difficult and direct formulations with velocities are usually found computationally inefficient. For this reason, instead of solving directly the resulting optimization problem, the optimization problems are transformed into its dual problem and solved thereafter. For example, Sloan (1989) and Makrodimopoulos and Martin (2007) solved the dual problem arising from the velocity formulations with the active set algorithm and the interior point algorithm respectively.

An intuitive question then arises: is it possible to formulate the upper bound problem directly from the dual perspective? Regarding this aspect, Christiansen (1980) and later Ciria (Ciria 2002; Ciria et al. 2008) provides a theoretical discussion on the duality of the lower and upper bound analysis. By constructing a purely kinematical discretisation, the resulting optimization from the upper bound formulation will be a mathematical programming analysis of stress variables only. A similar formulation was proposed by Krabbenhoft et al. (2005).

5.4.1 Pure kinematically admissible finite element discretisation

The upper bound solution is guaranteed only when the pure kinematically admissible discretisation is applied to the work equation (3.3). In the following discussion, the discussion is restricted to the discretisation with piecewise constant stress field and discontinuous linear velocity as it forms the pure kinematic discretisation (Ciria 2002).

5.4.2 Discretisation without velocity discontinuities

In the discretisation, all that is required is to replace the continuum field $\boldsymbol{\sigma}$ and $\mathbf{u}$ with the kinematically admissible discretisation $\boldsymbol{\sigma}_h$ and $\mathbf{u}_h$ of finite element space $X_h^{UB} \times Y_h^{UB}$.

The piecewise constant stress field in the global interpolation is given by

$$\boldsymbol{\sigma}(\mathbf{x}) = \sum_{e=1}^{E} \sigma^e \varphi_e(\mathbf{x}) \tag{5.44}$$

where $\varphi_e(\mathbf{x})$ is defined as

$$\varphi_e(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega^e \\ 0 & otherwise \end{cases} \tag{5.45}$$

The interpolated velocity field can be expressed in either a global or a local interpolation form as follows

$$u_i(\mathbf{x}) = \sum_{A=1}^{N} u_i^A \phi_A(\mathbf{x}) \tag{5.46}$$

$$u_i^e(\mathbf{x}) = \sum_{\mathrm{a}=1}^{NPE} u_i^{a,e} N_a^e(\mathbf{x}) \tag{5.47}$$

Substituting (5.46) and (5.44) into (3.3), we have

$$\begin{aligned}
&\int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \alpha \left(\int_{\Omega} \mathbf{f} \cdot \mathbf{u} d\Omega + \int_{\Gamma^N} \mathbf{t} \cdot \mathbf{u} \, dS \right) \\
&\Leftrightarrow \sum_{e=1}^{E} \sum_{A=1}^{N} \sum_{i,j=1}^{D} \int_{\Omega} \sigma_{ij}^e \varphi_e(\mathbf{x}) u_i^A \frac{\partial \phi_A(\mathbf{x})}{\partial x_j} d\Omega \\
&= \alpha \sum_{A=1}^{N} \sum_{i=1}^{D} u_i^A \left(\int_{\Omega} f_i \phi_A(x) d\Omega + \int_{\Gamma^{\mathrm{N}}} t_i \phi_A(s) dS \right)
\end{aligned} \tag{5.48}$$

where $\phi_A(s)$ = restriction of $\phi_A(\mathbf{x})$ to the boundary of Γ^N. Taking into account that $\varphi_e(\mathbf{x})$ equals to 1 inside the element Ω^e and vanishes outside, (5.48) can be written elementwise as follows,

$$\begin{aligned}
&\sum_{e=1}^{e} \sum_{A=1}^{N} \sum_{i,j=1}^{D} \underbrace{\int_{\Omega^e} \sigma_{ij}^e u_i^A \frac{\partial N_A^e(\mathbf{x})}{\partial x_j} d\Omega}_{1} \\
&= \alpha \sum_{A=1}^{N} \sum_{i=1}^{D} u_i^A \sum_{e=1}^{E} \left(\int_{\Omega^e} f_i N_A^e(\mathbf{x}) d\Omega + \int_{\xi_e^N} g_i N_A^e(s) dS \right)
\end{aligned} \tag{5.49}$$

Since stresses are constant over each element, it can be moved to the outside of the integral and using the vector notation, it is obtained as

$$\sum_{e=1}^{E}\sum_{A=1}^{N}(\mathbf{u}_h^A)^{\boldsymbol{T}}\overline{\mathbf{B}}_{\mathbf{a}}^{e}(\boldsymbol{\sigma}_h^e) = \alpha\sum_{A=1}^{N}(\mathbf{u}_h^A)^T\mathbf{F}_h^A \tag{5.50}$$

where

$$(\mathbf{u}^a)^{\boldsymbol{T}} = (u_1^a \quad u_2^a \quad \dots \quad u_D^a) \tag{5.51}$$

and $\overline{\mathbf{B}}_a^e$ is the matrix that relates velocities to the rate of strain which is similar to $\mathbf{B}$ matrix defined in (5.37), but each components are integrated over the element. In the case of three-dimension analysis, $\overline{\mathbf{B}}_a^e$ is given as follows

$$\overline{\mathbf{B}}_a^e = \begin{pmatrix} \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_1}\mathrm{d}\Omega & 0 & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_2}\mathrm{d}\Omega & 0 & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_2}\mathrm{d}\Omega & 0 \\ 0 & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_2}\mathrm{d}\Omega & 0 & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_1}\mathrm{d}\Omega & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_2}\mathrm{d}\Omega & 0 \\ 0 & 0 & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_2}\mathrm{d}\Omega & 0 & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_2}\mathrm{d}\Omega & \int_{\Omega^e}\frac{\partial N_A^e}{\partial x_1}\mathrm{d}\Omega \end{pmatrix} \tag{5.52}$$

and $\mathbf{F}_h^A$ resembles the form in (5.40).

Writing (5.50) in a more compact form and using matrix notation, we have

$$\mathbf{u}_h^{\mathrm{T}}\mathbf{A}^{eq}\boldsymbol{\sigma}_{\mathbf{h}} = \alpha\mathbf{u}_h^{\mathrm{T}}\mathbf{F}_h^{eq} \tag{5.53}$$

where

$$\mathbf{A}^{eq} = \begin{pmatrix} \mathbf{B}_1^1 & \mathbf{B}_1^2 & \cdots & \cdots & \mathbf{B}_1^E \\ \mathbf{B}_2^1 & \mathbf{B}_2^2 & \cdots & \cdots & \mathbf{B}_2^E \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \vdots \\ \mathbf{B}_A^1 & \cdots & \cdots & \cdots & \mathbf{B}_{\mathrm{A}}^{\mathrm{E}} \end{pmatrix} \mathbf{u} = \begin{pmatrix} \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^A \\ \vdots \\ \mathbf{u}^N \end{pmatrix}; \boldsymbol{\sigma} = \begin{pmatrix} \boldsymbol{\sigma}^1 \\ \vdots \\ \boldsymbol{\sigma}^e \\ \vdots \\ \boldsymbol{\sigma}^E \end{pmatrix}; \mathbf{F}^{eq} = \begin{pmatrix} \mathbf{F}_h^1 \\ \vdots \\ \mathbf{F}_h^A \\ \vdots \\ \mathbf{F}_h^N \end{pmatrix}$$

Note that (5.53) holds for arbitrary $\mathbf{u_h} \in \boldsymbol{Y^{UB}}$, hence the velocity could be cancelled. (5.53) is equivalent to

$$\mathbf{A}^{eq}\boldsymbol{\sigma}_{\mathbf{h}} = \alpha\mathbf{F}_h^{eq}$$

Thus the global optimization problem is given as

$$\begin{array}{ll} \text{maximize} & \alpha \\ \text{subject to} & \begin{cases} \mathbf{A}^{eq}\boldsymbol{\sigma}_{\mathbf{h}} = \alpha\mathbf{F}^{eq} \\ f(\boldsymbol{\sigma}_{\mathbf{h}}^{e}) \leq 0 \ \ \forall e = 1, \dots, E \end{cases} \end{array} \tag{5.54}$$

5.4.3 Discretisation with velocity discontinuities

The virtual work equation of the form (3.3) does not hold any longer for a discontinuous velocity field. The internal power is dissipated not only within the continuum but also along the discontinuities. To incorporate the discontinuities in the upper bound formulation, the traction field $\mathbf{t}$ is required in addition to the stress $\boldsymbol{\sigma}_h$ to formulate the discrete form of work equation. For the convenience of manipulation, the traction field $\mathbf{t}$ will be defined in the local coordinates x' and varies linearly along the edges shared by neighbouring elements. The traction is given by

$$\mathbf{t}^{\xi_e^{e'}}(s) = \sum_{\alpha=1}^{NPS} \mathbf{t}^{\alpha,\xi_e^{e'}} N_\alpha^{\xi_e^{e'}}(s) \tag{5.55}$$

where α= numbering of the nodes on the side and $NPS =$ the Node Per side of the element.

In fact, it is easier to comprehend if the discontinuities are viewed as the collapse of elements e and e' as shown in Figure 5.4 for 2D case and collapse of tetrahedral for 3D case. This interpretation was provided by Krabbenhoft et al. (2005). The velocity discontinuity is achieved when δ approaches zero. The benefit of such interpretation is that the yield constraint on the traction field will be greatly simplified as the discontinuities will be treated in the same manner as continuous elements.

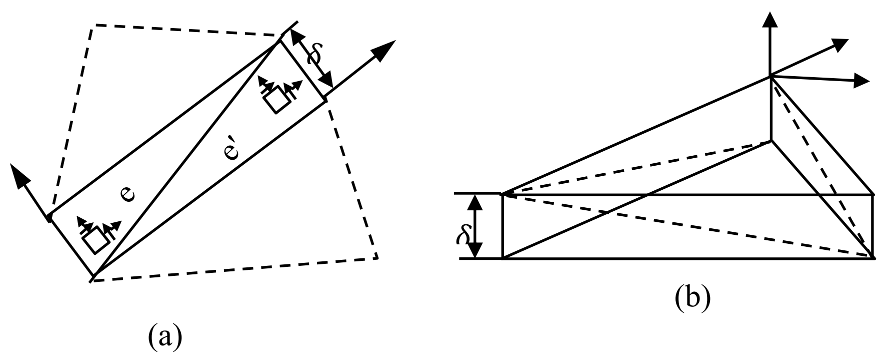


Figure 5.4 Discontinuities as the collapse of elements (a) discontinuitty in triangular mesh and (b) discontinuity in tetrahedral mesh

The virtual work equation with the weak form of equilibrium reads:

$$\underbrace{\sum_{i,j}^{D}\int_{\Omega}\sigma_{ij}\frac{\partial u_i}{\partial x_j}d\Omega}_{1}+\underbrace{\sum_{\xi_e^{e'}\in\mathcal{E}^O}\sum_{i'=1}^{D}\int_{\xi_e^{e'}}t_{i'}^{\xi_e^{e'}}\left(u_{i'}^{e'}-u_{i'}^{e}\right)dS}_{2}$$
$$=\underbrace{\alpha\sum_{i=1}^{D}\left(\int_{\Omega}f_iu_id\Omega+\int_{\Gamma^{\mathrm{N}}}g_iu_idS\right)}_{3},\forall u\in Y \tag{5.56}$$

Discretisation of term 1 and term 3 is similar to that for (5.49), therefore

$$\sum_{i,j}^{D}\int_{\Omega}\sigma_{ij}\frac{\partial u_i}{\partial x_j}d\Omega=\sum_{e=1}^{E}\left(\sum_{a=1}^{NPE}\mathbf{u}^{a,e}\overline{\mathbf{B}}_{\boldsymbol{a}}^{e}\right)\sigma^{e}=\sum_{e=1}^{E}\mathbf{u}^{e}\mathbf{B}^{e}\boldsymbol{\sigma}_{h}^{e}=\mathbf{u}_{h}\mathbf{A}^{eq1}\boldsymbol{\sigma}_{h} \tag{5.57}$$

For term 3

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{a=1}^{NPE}\begin{pmatrix}u_1^{a,e} & u_2^{a,e} & \cdots & u_D^{a,e}\end{pmatrix}\begin{pmatrix}\int_{\Omega^e}f_1^e N_A^e(\mathbf{x})d\Omega+\int_{\xi_e^N}g_1^{\xi_e^N}N_A^e(s)dS\\ \int_{\Omega^e}f_2^e N_A^e(\mathbf{x})d\Omega+\int_{\xi_e^N}g_2^{\xi_e^N}N_A^e(s)dS\\ \vdots\\ \int_{\Omega^e}f_D^e N_A^e(\mathbf{x})d\Omega+\int_{\xi_e^N}g_D^{\xi_e^N}N_A^e(s)dS\end{pmatrix}\\
&=\sum_{e=1}^{E}\sum_{a=1}^{NPE}\left(\mathbf{u}_h^{a,e}\right)^T\mathbf{F}_{h_a}^{\ e}\\
&=\sum_{e=1}^{E}\begin{pmatrix}\mathbf{u}^{1,e} & \mathbf{u}^{2,e} & \cdots & \mathbf{u}^{NPE,e}\end{pmatrix}\begin{pmatrix}\mathbf{F}_{h_1}^{\ e}\\ \mathbf{F}_{h_2}^{\ e}\\ \vdots\\ \mathbf{F}_{h_{NPE}}^{\ e}\end{pmatrix}=(\mathbf{u}_h)^T\mathbf{F}_h^{eq}
\end{aligned} \tag{5.58}$$

For term 2, the velocity jump along the edges in the vector form is written as

$$\Delta\mathbf{u}(s)=\sum_{a=1}^{NPS}\Delta\mathbf{u}_a N_a(s)\,; \tag{5.59}$$

where $\Delta\mathbf{u}_a=\left(\mathbf{u}_a^{e}{}'-\mathbf{u}_a^e\right)$, then term 2 in (5.56) can be written as

$$\sum_{\xi_e^{e'}\in\mathcal{E}^O}\sum_{i'=1}^{D}\int_{\xi_e^{e'}}t_{i'}^{\xi_e^{e'}}\left(u_{i'}^{e'}-u_{i'}^{e}\right)dS=\sum_{\xi_e^{e'}\in\mathcal{E}^O}\int_{\xi}\mathbf{t}^{\mathrm{T}}\mathbf{M}\Delta\mathbf{u}dS \tag{5.60}$$

Note that $\left(u_{i'}^{e'} - u_{i'}^{e}\right)$ in (5.60) is the velocity jump in the local coordinate denoted by the prime. The $\mathbf{M}$ is the transform matrix that is used to construct the local coordinate system. We can focus power dissipation on one particular discontinuity as

$$\begin{aligned}
&\int_{\xi} \mathbf{t}^{\mathrm{T}} \mathbf{M} \Delta \mathbf{u} dS \\
&= \int_{\xi} \begin{pmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_{NPS} \end{pmatrix} \begin{pmatrix} N_1^{\xi} \\ N_2^{\xi} \\ \vdots \\ N_{NPS}^{\xi} \end{pmatrix} \mathbf{M}^{\xi} \begin{pmatrix} \Delta \mathbf{u}_1 & \Delta \mathbf{u}_2 & \cdots & \Delta \mathbf{u}_{NPS} \end{pmatrix} \begin{pmatrix} N_1^{\xi} \\ N_2^{\xi} \\ \vdots \\ N_{NPS}^{\xi} \end{pmatrix} \\
&= \sum_{i,j=1}^{NPS} \mathbf{t}_i \mathbf{B}_{ij}^{\xi} \Delta \mathbf{u}_j
\end{aligned} \tag{5.61}$$

where $\mathbf{B}_{ij}^{\xi} = \mathbf{M}^{\xi} \int_{\xi} N_i^{\xi} N_j^{\xi} \, dS$. We can relate the velocity jump to global velocity vector by (5.62)

$$\Delta \mathbf{u}_i = (\mathbf{u}_i^{\beta} - \mathbf{u}_i^{\alpha}) = \mathbf{A}_i^{\xi} \mathbf{u}_{\boldsymbol{h}} \tag{5.62}$$

where (α, β) denotes a pair of nodes on the interface. After further manipulation of (5.61) , we can obtain the following relation

$$\begin{aligned}
&\mathbf{u}^{\mathbf{T}} \underbrace{\left(\begin{pmatrix} \left(\mathbf{A}_1^{\xi}\right)^T & \left(\mathbf{A}_2^{\xi}\right)^T & \cdots & \left(\mathbf{A}_{NPS}^{\xi}\right)^T \end{pmatrix} \begin{pmatrix} \left(B_{11}^{\xi}\right)^{\mathrm{T}} & \left(B_{12}^{\xi}\right)^{\mathrm{T}} & \cdots & \left(B_{1NPS}^{\xi}\right)^{\mathrm{T}} \\ \left(B_{21}^{\xi}\right)^{\mathrm{T}} & \left(B_{22}^{\xi}\right)^{\mathrm{T}} & \cdots & \left(B_{2NPS}^{\xi}\right)^{\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots \\ \left(B_{NPS1}^{\xi}\right)^{\mathrm{T}} & \left(B_{NPS2}^{\xi}\right)^{\mathrm{T}} & \dots & \left(B_{NPS\,NPS}^{\xi}\right)^{\mathrm{T}} \end{pmatrix} \right)}_{\mathbf{A}_{\xi}^{eq2}} \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_{NPS} \end{pmatrix} \\
&= \mathbf{u}^{\mathbf{T}} \mathbf{A}_{\xi}^{eq2} \mathbf{t}^{\xi}
\end{aligned} \tag{5.63}$$

Summing up the contributions from each discontinuity (5.63), we have

$$terms2 = \sum_{\xi \in \mathcal{E}^O} \mathbf{u}^{\mathbf{T}} \mathbf{A}_{\xi}^{eq2} \mathbf{t}^{\xi} = \mathbf{u}^{\mathbf{T}} \mathbf{A}^{\mathrm{eq2}} \mathbf{t} \tag{5.64}$$

where $\mathbf{A}^{eq2}$=global matrix having dimensions $(2 \times N, 3 \times E)$; $\mathbf{F}_h$=$D \times N$ vector of nodal variables

5.4.4 The global optimization problem

Equation (5.64) holds for all $\mathbf{u}_{\mathrm{h}} \in \mathrm{Y}_h^{UB}$, and thus the velocity variable can be cancelled out from the equation. Tractions on the discontinuities need to be constrained by the yield criterion as well. As discussed earlier, this could be achieved more easily by viewing the discontinuities as collapse of elements (see Figure 5.4). The fulfilment of the yield criterion will then be similar to a general element, let $\boldsymbol{\sigma}^{\xi_e^{e'}}$ denote the nodal stresses of the discontinuities (or of infinitely narrow elements) and **t** is related to $\boldsymbol{\sigma}^{\xi_e^{e'}}$ by the a matrix T, **t**= **T** $\boldsymbol{\sigma}^{\xi_e^{e'}}$

$$\begin{array}{ll} \max & \alpha \\ \text{subject to} & \begin{cases} \mathbf{A}^{\mathrm{eq1}}\boldsymbol{\sigma} + \mathbf{A}^{\mathrm{eq2}}\mathbf{T}\,\boldsymbol{\sigma}^{\xi_e^{e'}} = \alpha \mathbf{F}^{eq} \\ f(\boldsymbol{\sigma}^e) \leq 0 \quad \forall e = 1 \ldots E \\ f\left(\boldsymbol{\sigma}_i^{\xi_e^{e'}}\right) \leq 0 \ \forall \xi_e^{e'} \in \mathcal{E}^O, i = 1, \ldots, NPS \end{cases} \end{array} \tag{5.65}$$

Note that in the upper bound limit analysis, it is likely that the displacement boundary condition $u(x) = 0, x \in \Gamma^D$ will hold, therefore, the rows associated with the prescribed boundary conditions need to be removed in the calculation.

5.5 Summary

There are two directions in formulating the upper bound limit analysis. Following the conventional upper bound technique, the power dissipation is expressed in pure kinematic terms (velocities and strain rate), and the problem is constructed by fulfilling the flow rule condition. The other way is to build the problem directly from the dual problem, in which the dual variables can be considered as the averaged stresses. Formulating the upper bound theorem in terms of the stress variables may not be physically obvious; however, it takes the advantages of the dual relationship between lower bound and upper bound and consequently the optimization problem resulting from the dual formulation can be more efficiently solved. If primal-dual interior point algorithms are to be applied in the solution, the dual variables, i.e., the velocity variables can be obtained automatically.

CHAPTER 6: MESH ADAPTATION IN LIMIT ANALYSIS

6.1 Introduction

The quality of the numerical solution depends on the intensity of the discretisation when the order of the interpolation is held constant. A finer mesh will normally result in solutions with higher accuracy. As uniform refinement is generally not realistic in practice due to the limitation of the computer resources and the solution time, there have been increasing interests in designing adaptive schemes that can concentrate the degrees of freedom to the appropriate region automatically. For most of the stability problems in geotechnical engineering, the slip bands are highly localized and thus a properly arranged mesh considerably improves the accuracy of the solution (Lyamin and Sloan 2002a; Lyamin and Sloan 2002b; Makrodimopoulos and Martin 2006; Makrodimopoulos and Martin 2007)..

The failure mechanism is only known for simple problems with homogeneous soil profile, regular geometries and simple failure models. The location and the shape of the slip bands are usually not easy to be reliably predicted for general condition. Instead of constructing a specialised mesh incorporating the failure mechanism, it is desirable to tackle this problem under a general framework to optimally distribute the grids into the proper region on an adaptive basis with the posterior error estimators. It is important that the reliance on the engineer's experience should be reduced to a minimum or even zero for a useful and practical adaptive mesh refinement scheme.

Two issues need to be addressed for the mesh adaptation: (1) the method in determining which elements to be updated and (2) the method of how the marked elements are to be updated.

Study of the mesh adaptation has been a line of research since the development of the finite element method in which the elements are marked according to the indicators calculated from the discretisation errors. The discretisation error itself also serves as an evaluation of the quality of calculated solutions.

6.2 Error estimate

The error of a discrete solution due to the discretisation is defined by the difference between the numerical approximate and the exact solution. When the exact solution of a system is not known, which is often the case; the accuracy of the numerical approximate could be assessed by performing a sequence of analyses with meshes of increasing intensity. The exact solution is then predicted by the Richardson’s extrapolation and thus the error is calculated. However, this process is overly computationally involved. Various error estimators that are exclusively determined based on the current solution and the problem data have been proposed in the literature of the FEM. Comprehensive reviews regarding error estimate for the FEM are provided in works by Gratsch and Bathe (2005) and Zienkiewicz and Taylor (2005).

Error estimators in the FEM can be classified into three groups (Gratsch and Bathe 2005):

(1) Explicit error estimator that directly makes use of the finite element interpolation and the data of the problem;

(2) Implicit error estimator where an auxiliary local boundary problem is required to be solved;

(3) Recovery-based error estimators that make use of the difference between the smoothed gradient and unsmoothed gradient.

Since the finite element spaces are used for the discretisation for stress and velocity field in the FELA, it is natural to ask whether similar error estimators can be tailored to steer the mesh adaptation in the FELA.

Firstly, it is worthwhile to review the error estimation in the displacement-based FEM in which displacements are taken as the primal variables. The boundary value problem consists of finding the solution u that satisfies eq.(6.1) will be discussed for one-dimensional case for simplicity.

$$
\begin{aligned}
-\Delta \mathbf{u} &= \mathbf{f} \quad on\ \Omega \\
\mathbf{u} &= 0 \quad on\ \Gamma^D \\
\mathbf{n} \cdot \nabla \mathbf{u} &= \mathbf{t} \quad on\ \Gamma^N
\end{aligned}
\tag{6.1}
$$

Writing (6.1) in the weak form, we have

$$
\boldsymbol{a}(\mathbf{u}, \mathbf{v}) = \boldsymbol{F}(\mathbf{v})\ \forall \mathbf{v} \in Y \tag{6.2}
$$

where $Y = \{\mathbf{v} \in H^1(\Omega) \colon \mathbf{v} = 0\ on\ \Gamma^D\}$.

Discretised form of the solution of a governing equation is to find a function $\mathbf{u}_h \in V_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) = F(\mathbf{v}_h); \forall \mathbf{v}_h \in Y_h \subset Y \tag{6.3}$$

The error of the solution due to interpolation is defined as the difference between the exact solution and the numerical approximation.

$$\mathbf{e}_h = \mathbf{u} - \mathbf{u}_h \tag{6.4}$$

The residual due to the interpolation can be expressed as

$$a(\mathbf{e}_h, \mathbf{v}) = a(\mathbf{u}, \mathbf{v}) - a(\mathbf{u}_h, \mathbf{v}) = F(\mathbf{v}) - a(\mathbf{u}_h, \mathbf{v}) = \boldsymbol{R}_h(\mathbf{v}); \forall \mathbf{v} \in Y \tag{6.5}$$

(6.5) forms the basis of a large class of the energy-norm-based error estimator.

6.2.1 Residual based error estimator

Explicit error estimators are concerned with the direct evaluation of the residual. According to (6.1), we have the following relation

$$a(\mathbf{e}_h, \mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma^N} \mathbf{t} \cdot \mathbf{v}\, dS - \int_{\Omega} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v} d\Omega\,; \forall \mathbf{v} \in \Omega \tag{6.6}$$

Applying integration by parts to the last term in (6.6) yields

$$a(\mathbf{e}_h, \mathbf{v}) = \sum_{e=1}^{E} \int_{\Omega^e} \mathbf{R} \cdot \mathbf{v}\, d\Omega + \sum_{\xi \in (\mathcal{E}^N \cup \mathcal{E}^\xi)} \int_{\xi} \mathbf{J} \cdot \mathbf{v}\, dS\,; \forall \mathbf{v} \in \Omega \tag{6.7}$$

where R = the elemental contribution to the residual and $\mathbf{J}$ = jump of the gradient across the element edge ξ .

$$\mathbf{R} = \mathbf{f} + \Delta\mathbf{u} \qquad in\ \Omega^e \tag{6.8}$$

and

$$\mathbf{J} = \begin{cases} (\mathbf{n}\cdot\nabla\mathbf{u}_h + \mathbf{n}'\nabla\mathbf{u}_h'); & \xi \in \Gamma \\ \mathbf{t} - \mathbf{n}\cdot\nabla\mathbf{u}_h; & \xi \in \mathcal{E}^N \\ 0; & \xi \in \mathcal{E}^D \end{cases} \tag{6.9}$$

The Galerkin orthogonal condition follows (Brenner and Scott 2008)

$$a(\mathbf{e}_h, \mathbf{v}_h) = 0, \forall \mathbf{v}_h \in V_h \tag{6.10}$$

Introducing eq.(6.10) into (6.7), we have

$$\begin{aligned} a(\mathbf{e}_h, \mathbf{v}) &= \sum_{e=1}^{E}\int_{\Omega^e} \mathbf{R}\cdot(\mathbf{v} - \mathbf{J}_h\cdot\mathbf{v})\, d\Omega \\ &+ \sum_{\xi\in(\mathcal{E}^N\cup\mathcal{E}^\xi)}\int_{\xi} \mathbf{J}\cdot(\mathbf{v} - \mathbf{J}_h\cdot\mathbf{v})\, dS \qquad ; \forall v \in V \end{aligned} \tag{6.11}$$

By applying the Cauchy-Schwarz inequality elementwise, we have

$$a(\mathbf{e}_h, \mathbf{v}) \leq \sum_{e=1}^{E} ||\mathbf{R}||_{L^2} ||\mathbf{v} - \mathbf{J}_h\cdot\mathbf{v}||_{L^2} d\Omega + \sum_{e=1}^{E} ||\mathbf{J}||_{L^2} ||\mathbf{v} - \mathbf{J}_h\cdot\mathbf{v}||_{L^2} d\Omega \tag{6.12}$$

With the theory of the interpolation

$$\left|||\mathbf{v} - \mathbf{J}_h\cdot\mathbf{v}||\right|_{L^2} \leq ch_k ||\mathbf{v}||_{H^1(K)} \tag{6.13}$$

$$\left|\left|\mathbf{v}-\mathbf{J}_h\cdot\mathbf{v}\right|\right|_{L^2(\partial K)}\leq c\sqrt{h_K}\left|\left|\mathbf{v}\right|\right|_{H^1(\widetilde{K})} \tag{6.14}$$

Where h_K =diameter of the element Ω^e

Relations in (6.13) and (6.14), (6.12) lead to

$$a(\mathbf{e}_h,\mathbf{v})\leq c\left|\left|\mathbf{v}\right|\right|_{H^1(\Omega)}\left(\sum_{e=1}^{E}h_K^2\left|\left|\mathbf{R}\right|\right|_{L^2(K)}^2+\sum_{\xi\in\mathcal{E}}h_K\left|\left|\mathbf{J}\right|\right|_{L^2(\xi)}^2\right)^{\frac{1}{2}} \tag{6.15}$$

Using the inequality $\left|\left|v\right|\right|_{H^1(\Omega)}\leq c\left|\left|v\right|\right|_E$ and replacing v with e_h yields the final error

$$\left|\left|\mathbf{e}_h\right|\right|_E^2\leq\sum_{e=1}^{E}\left(h_h^2\left|\left|\mathbf{R}\right|\right|_{L^2(\Omega^e)}^2+c_2h_h\left|\left|\mathbf{J}\right|\right|_{L^2(\partial\Omega^e)}^2\right) \tag{6.16}$$

(6.16) directly leads to the error indicator η_h defined by

$$\left|\left|\mathbf{e}_h\right|\right|_E^2\leq(E_h)^2=\sum_{\Omega^e\in\mathcal{T}_h}\eta_K^2 \tag{6.17}$$

with $\eta_{\Omega^e}=h_h^2\left|\left|R\right|\right|_{L^2(\Omega^e)}^2+c_2h_h\left|\left|J\right|\right|_{L^2(\partial\Omega^e)}^2$

6.2.2 Recovery based error estimator

For the displacement-based FEM, the displacement field is continuous while the stress and strain fields calculated by differentiation are discontinuous across the inter-element boundaries. Discontinuities of stresses and strains are the direct results of the introduction of the finite element discretisation and could be viewed as an indicator of error. In the recovery-based error estimate, post-processing of the gradient of the solution is first performed and then the estimate of the error is obtained by comparing the post-processed gradients with the original one. For some particular problems, the recovery can be better justified by the fact that there exist certain points within an element that have higher accuracy of derivatives, the so-called superconvergence point. Let $\mathcal{r}(u_h)$ be the recovered gradient, and the elemental error could be defined by (6.18) as

$$(E_h)^2 = \int_{\Omega^e} |\mathcal{r}(\mathbf{u}_h) - \nabla \mathbf{u}_h|^2 d\Omega \tag{6.18}$$

A well-known patch recovery error estimator (Zienkiewicz and Zhu 1992a; Zienkiewicz and Zhu 1992b) is based on the form as given by (6.19).

$$\nabla \mathbf{u}_h^* = \sum_{i=1}^{n} (\nabla \mathbf{u}_h^*)_i \varphi_i \tag{6.19}$$

The unknown nodal values $(\nabla \mathbf{u}_h^*)_i$ are established by a standard L^2-projection

$$\int_{\Omega} \varphi_i (\nabla \mathbf{u}_h^* - \nabla \mathbf{u}_h) d\Omega = 0; j = 1, \dots, n \tag{6.20}$$

(6.20) will lead to a system of linear equations with nodal values as the unknowns

$$\sum_{i=1}^{n} \int_{\Omega} \varphi_j \varphi_i d\Omega \, (\nabla \mathbf{u}_h^*)_i = \int_{\Omega} \varphi_i \nabla \mathbf{u}_h d\Omega ;\;\; j = 1, \dots, n \tag{6.21}$$

In the expression of the error, the smoothed (improved) gradient is used instead of the exact one

$$\left|\left|\mathbf{e}_h\right|\right|_E^2 \approx a(\mathbf{e}_h^*, \mathbf{e}_h^*) = \int_\Omega (\nabla\mathbf{u}_h^* - \nabla\mathbf{u}_h)^2 d\Omega \tag{6.22}$$

In practical application, the error estimate is calculated elementwise as

$$\left|\left|\mathbf{e}_h\right|\right|_E^2 \approx (E_h)^2 = \sum_{\Omega^e \in \mathcal{T}_h} \eta_k^2 \tag{6.23}$$

with $\eta_k^2 = \left|\left|\nabla\mathbf{u}_h^* - \nabla\mathbf{u}_h\right|\right|_{L^2(\Omega^e)}$

6.2.3 Residual and recovery based error estimate in the limit analysis

The deformation pattern of a solid body in elastoplastic FEM differs from that in the limit analysis. In limit analysis, the domain at the ultimate limit state consists of the plastic deformed region and the plastic rigid region (Christiansen 1996), while the deformation is more continuous in the elastoplastic finite element analysis. Nevertheless, there have been a number of attempts in applying the error estimate concepts developed in FEM in limit analysis (Borges et al. 2001; Lyamin et al. 2005).

In limit analysis, the choice of the control variables in the calculation of the error is not obvious. The scalar field of the Lagrangian multipliers have been adopted by Borges et al. (2001) and Lyamin et al. (2005) based on the patch recovery technique. In this research, we will extend the work and also study the performance of the residual-based error estimators evaluating the jumps of the multiplier field. For the FELA formulated as NLP, the Lagrangian multipliers associated with the yield criteria serve as a suitable option for the control variable, as it indicates which inequalities are active and which point is undergoing plastic flow. Adopting the Lagrangian multipliers as the control variable was adopted by Borges et al. (2001) in the mixed limit analysis based on the recovery Hessian matrix technique. Following similar approach, Lyamin et al. (2005) tailored the error estimator for the lower bound analysis and studied various recovery schemes of the recovery of the Lagrangian multiplier field, recovery of the Lagrangian field, recovery of the gradient of Lagrangian field and Hessian of Lagrangian. From their results, it appears that different recovery techniques do not provide major noticeable differences. In views of the previous findings, the gradient recovery scheme will be adopted in this research as it resembles the procedures in recovery process in the displacement-based FEM. In our lower bound technique, multipliers are discontinuous due to the introduction of the stress discontinuities. A local interpolation will be used for the calculation of strains at the Gaussian points in the recovery process.

$$\eta_k^2 = \left|\left|\nabla\mathbf{L}_h^* - \nabla\mathbf{L}_h\right|\right|_{L^2(\Omega^e)} \tag{6.24}$$

where ∇L_h^* is the recovered gradient of the Lagrangian multiplier based on the recovery scheme previously described. Similarly, we could apply (6.17) as well by applying the residual error estimators from the Lagrangian multiplier field.

For the SOCP formulation of the FELA, the dual variable field is very much like the stress field and belongs to a dual cone. In this case, the dual variable could be regarded as the equivalent strains (Christiansen and Edmund 2001). The recovery based error estimator could be obtained by the error of the sum of n components in the dual cone,

$$\eta_k^2 = \sum_{i=1}^{n} \left| |\nabla e_i^* - \nabla e_i| \right|_{L^2(\Omega^e)} \quad (6.25)$$

Similarly, the residual based error estimator for the FELA could be obtained by replacing the continuous displacement field with the discontinuous dual variable field. We can also apply the residual error estimator to the dual variable field.

In the upper bound FELA, if the continuous velocity field is adopted in the formulation, the recovery-based or the residual-based error estimate could be utilized directly following the procedure in the FEM, as the velocity field is exactly the same as that in the FEM. When the velocity field is discontinuous, the gradients of the velocity field are calculated with the same procedure as that in the multiplier field scheme.

6.2.4 Error estimate in lower and upper bound limit analysis

Due to the absence of the superconvergence in the FELA, it seems difficult to assess the local error from either a lower bound or upper bound analysis alone. However, it can be argued that the error of the solution is contributed exclusively from the plastically deformed region. This point is easier to understand from the viewpoint of an upper bound analysis for which the objective function consists of the terms due to internal power dissipation and external work done by body forces. The contribution to power dissipation comes exclusively from the plastic region and the refinement of the rigid zone does not give any improvement to the bound solution. This argument holds for lower bound as well because the lower bound problem can be transformed into an equivalent kinematic form. This is different from the FEM in which the deformation exhibits in a continuous form. The distinctions between the deformation fields calculated with the FELA and FEM have been discussed in length by (Christiansen 1996). Therefore, refinement of the plastic zone is both a practical and reasonable approach to be adopted.

6.2.5 Error based on the gap between the upper and lower bound

An error estimator, which is conceptually different from the residual or recovery based approaches in the FEM, was proposed by Ciria (2002). The error associated with a given triangulation $\mathcal{T}_h$ is defined by the gap between the upper bound and lower bound solution calculated from the same mesh, i.e.

$$\eta_h = \alpha^{UB} - \alpha^{LB} \tag{6.26}$$

The mesh is updated based on the elemental contribution to the error defined by (Ciria 2002; Ciria et al. 2008)

$$\begin{aligned} \eta_h^e = &\int_{\Omega^e} \boldsymbol{\sigma}_{LB}^e : \boldsymbol{\varepsilon}_{eq}(\mathbf{u}_{UB}^e)\, d\Omega \\ &- \left(\int_{\Omega^e} -(\nabla \cdot \boldsymbol{\sigma}_{LB}^e) \cdot \mathbf{u}_{UB}^e d\Omega + \int_{\partial\Omega^e} \left(\mathbf{n}^{\xi_e} \cdot \boldsymbol{\sigma}_{LB}^e\right) \cdot \mathbf{u}_{UB}^e dS \right) \end{aligned} \tag{6.27}$$

where $\boldsymbol{\sigma}_{LB}^e$= the stress field obtained in the lower bound analysis, $\mathbf{u}_{UB}^e$= velocity field obtained in the upper bound analysis.

Ciria (2002) and Ciria et al. (2008) applied the error estimator of (6.27) to a number of problems of von Mises material under plane stress and plane strain conditions and promising results were obtained. However, it is clear that (6.27) requires that both the stress field from the lower bound analysis and velocity field from the upper bound analysis to be calculated, which implies that the lower- and upper bound analysis need to be performed together. This poses a great challenge to the computer capacity for large models and will thus considerably limit the model size in practical applications. In this research, we will seek computationally cheaper mesh update scheme under which it is possible to compute separately the lower and upper bounds for large scale models.

6.2.6 Mesh adaptation in the limit analysis based on slackness of yield function

It could be argued and will be demonstrated in later sections that the error in the FELA comes exclusively from the slip bands because of the assumption of the rigid plasticity of the material. It is therefore reasonable to distribute the grids on the slip bands. To this end, evaluating the yield function provides the most straightforward approach to identify the yield zone of a domain. After the solution of a lower bound problem, the slackness of the yield functions at each vertex in the mesh can be back calculated. A criterion can then be set to determine the range of the refinement by adjusting the value of R as given by eq. (6.28).

$$f(\boldsymbol{\sigma}) = R \tag{6.28}$$

where $f(\boldsymbol{\sigma}) = 0$ is the yield criterion.

In the application for purely cohesionless soils where $c = 0$, the Mohr Coulomb yield criterion reduces to

$$f(\boldsymbol{\sigma}) = \sqrt{4\,\tau_{xy}^2 + \left(\sigma_x - \sigma_y\right)^2} - \sin\phi'\left(\sigma_x + \sigma_y\right) = 0 \tag{6.29}$$

It is clear that on the free boundary where the shear and normal stresses are zero, it is likely that the stress point $(\sigma_x, \sigma_y, \tau_{xy}) = (0,0,0)$, which implies that the yield criterion would be satisfied and the mesh will be unnecessarily refined. The authors have encountered this problem in plane strain analysis when trying to capture the full collapse mechanism. This could be avoided by using a smaller refine ratio, evaluating the norm of the vector of stress at the stage of marking the elements or simply use a practically small cohesion.

Other than evaluating the yield function, the dual variables corresponding to the yield function offers another feasible indicator of the yield zone. Dual variables associated with the yield function can be regarded as the equivalent strains. As rigid perfectly plasticity is assumed in limit analysis, strain components would become zero in rigid portion of the domain, and only the portion undergoing plastic flow has non-zero strains, thus a criterion could be designed as given by eq.(6.30).

$$\eta_e^2 = \left|\left|\mathbf{e}\right|\right|_{L^2(\Omega)} \tag{6.30}$$

where **e** are the vector of dual variables with respect to stresses with **e** belongs to the dual cone of the yield function. It should be pointed out that local refinement meshing guided by eq.(4.43) and eq.(2.25) has been proposed by Christiansen and Edmund (2001) for the von Mises material.

6.3 Localized mesh refinement with unstructured mesh

Methods for mesh adaptation include remeshing, reorienting or splitting elements. Remeshing generates new meshes based on the computed solution, which is comparatively computationally demanding because the mesh generation subroutine is called in each refinement and vast amount of data are to be transferred from one mesh to another. The merits of this technique are that the decedent meshes are not restricted by the initial mesh. This technique was adopted in the works by Borges et al. (2001) for the mixed limit analysis and Lyamin et al. (2005) for the lower bound limit.

In elements splitting method, elements marked according to the prescribed rules are subdivided into a number of children. This technique is simple in concept but the potential problem inherited is that the subdivision of elements will generate hanging nodes which requires more sophisticated data structure and complicated refinement algorithm to perform the refinement. Various element splitting methods can be designed (see Figure 6.1) to for the purpose of local refinement. The embedded refinement in Figure 6.1(d) is the simplest to implement since no hanging nodes will be generated during the refinement. However, the regularity of the mesh is not maintained, and more significantly, it may experiences “locking” because the discontinuities will be severely restricted by the initial mesh. The regular refinement or red-green refinement (Figure 6.1a)(Bank et al. 1983) and the bisection method (Figure 6.1b) (Rivara 1984; Sewell 1972) have been applied in the FEM and proven to be robust. The regular refinement divides the target element into 4 children (red procedure) and a green procedure is required to eliminate the hanging nodes on the edge. The bisection refinement divides the target element into two children and similarly a recursive algorithm is required to address the hanging nodes. For three-dimensional problem, the bisection divides a tetrahedron into two (see Figure 6.1a).

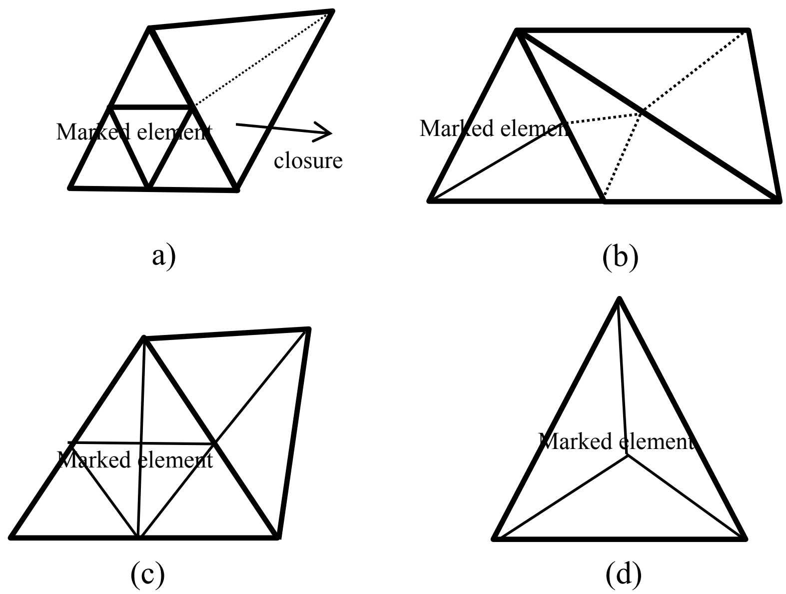


Figure 6.1(a) regular refinement, (b) bisection, (c) directed section and (d) embedment refinement

6.3.1 Bisection based refinement

The basic idea of bisection-based local refinement is to divide the target element into two children and the added extra node is either eliminated with a recursive algorithm or by a closure procedure. Depending on the choices of the edge to split, two variants of element bisection algorithms have been developed in the literature.

6.3.1.1 **Newest vertex bisection**

The newest vertex bisection method was first proposed by Sewell (1972) in which the target element is split into two smaller children by connecting one of the vertexes (called peak) and midpoint of the opposite edge (called base or refinement edge). The newly inserted node is assigned as the peak of the child element, i.e. the edge opposite to the newly-added node in each child will be split.

After the bisection of an element, an extra node (hanging node P) will be generated on the edge of the adjoining element as shown in Figure 6.2. It is necessary to eliminate the hanging node by further bisecting the neighbouring element. It is expected that one step of closure may generate more hanging nodes and thus the closure may propagate. Mitchell (1988) proved that the propagation stops after a finite number of iterations. Four similar classes will be generated as shown in Figure 6.3, thus the regularity of the triangulation is guaranteed.

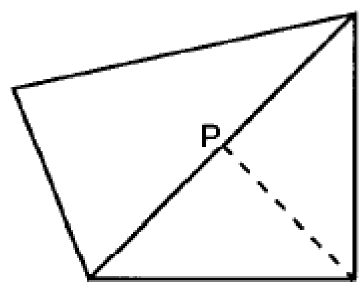


Figure 6.2 Bisection of a triangle

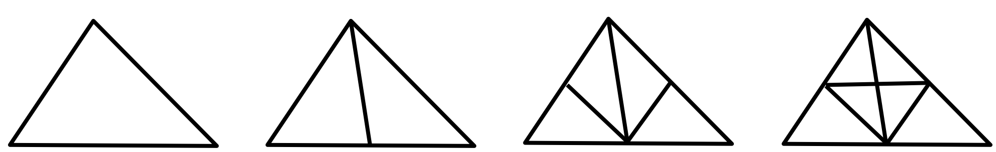


Figure 6.3 Four similar classes generated by the newest-vertex bisection

6.3.1.2 **Longest-edge bisection**

The longest-edge bisection proposed and studied by Rivara's group (Rivara 1984; Rivara 1989; Rivara and Iribarren 1996) always bisects one of the longest edges of the triangle to guarantee the regularity of meshes during the refinement. Indeed, it has been proved by Rosenberg and Strenger (1975) that the smallest angle in the descendants of the original element is bounded below by the half of the smallest angle in the initial triangulation.

When the longest edge is always opposite to the newest vertex, the longest-edge is equivalent to the newest-vertex bisection. One of the examples is for uniform meshes: the mesh obtained by dividing rectangles into triangles using their diagonals. The peaks are always at the right angles and the longest edges are opposite to the peaks.

A recursive algorithm based on the compatible division of an element for the bisection method has also been described in the work by Kossaczky (1994). An element is divided only when it is compatibly divisible. An element is compatibly divisible if the edge marked for division is the refinement edge of the neighbour opposite to the peak or on the boundary of the domain. The refinement process works only on the compatible element and obviously the hanging node is avoided automatically.

6.3.1.3 **Generalization to three dimensional tetrahedral elements**

Extension of the bisection-based algorithm to 3D tetrahedral mesh have been described by Rivara and Levin (1992), Liu and Joe (1995) and Kossaczky (1994) etc. The general idea is similar to what has been described for the triangular element refinement. To facilitate the algorithm, the tetrahedron is considered to be embedded into a parallelepiped M as shown in Figure 6.4 (a). The parallelepiped is subdivided into 8 similar ones by three steps as shown in Figure 6.5. $v_0 - v_1$, $v_0 - v_2$ and $v_o - v_3$ are divided in three steps.

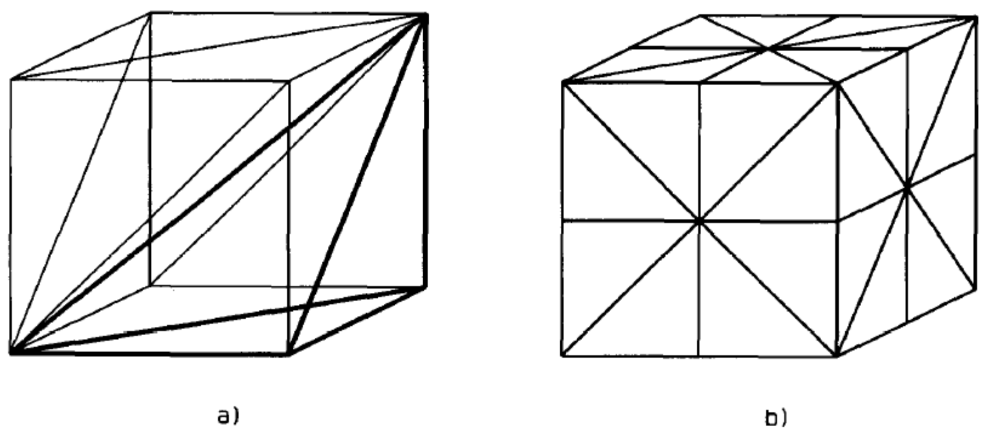


Figure 6.4 Tetrahedrons embedded in the parallelepiped

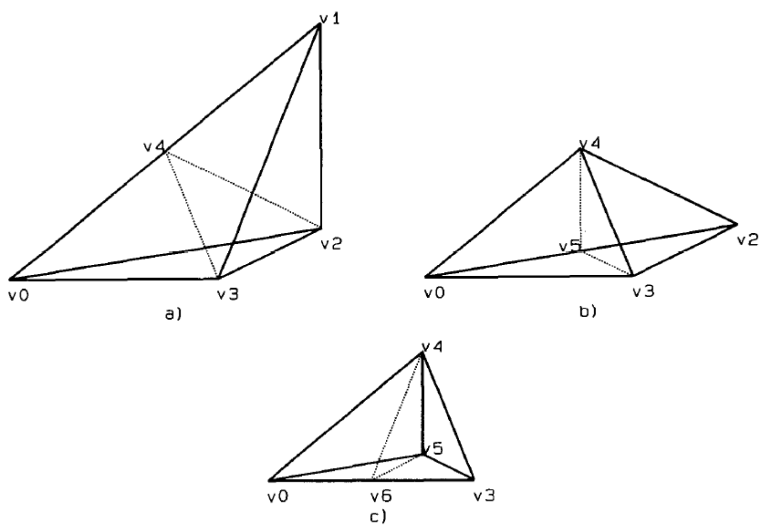


Figure 6.5 Three types of bisection of a tetrahedron (after Kossaczky (1994))

The recursive algorithm of bisection for the triangular mesh can be extended to the tetrahedral grids in a natural way, i.e. tetrahedra sharing a certain edges are divided in one step. Let e be the element and ξ be the refinement edge of e. To make e compatibily divisible, the refinement procedure at first transverse tetrahera around the refinement edge ξ. For the element e' where the refinement edge is not the same as ξ, this element is divided first, which leads to a recursive algorithm.

6.3.2 Red-Green Refinement

6.3.2.1 **Red Green refinement for triangular mesh**

Another method to perform local mesh refinement is the red-green-refinement (Bank et al. 1983) which is commonly applied to two-dimensional applications. During the refinement, a marked triangle is divided into 4 smaller children (called red refinement or regular refinement) as shown in Figure 6.6. The generated hanging nodes on the edges of the neighbouring triangles that are not flagged for the refinement are refined irregularly by bisection (green refinement). If more than two sides are split for a neighbouring element, it should also be marked for the red-refinement. By continuation of these two strategies, a hierarchical mesh is produced.

The pleasant property of this method is that the newly generated children are geometrically similar to the original one and therefore the element quality of the new triangulation is the same as the original ones.

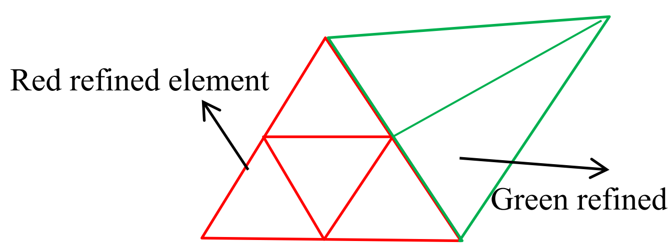
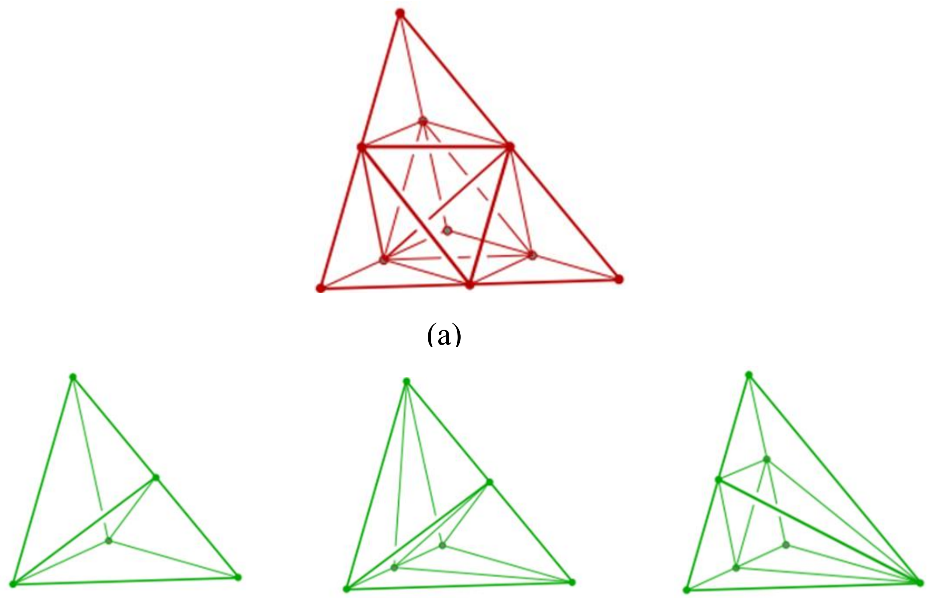


Figure 6.6 A red refined element with a green refined neighbor

6.3.2.2 **Generation to the three dimensional tetrahedral elements**

This strategy could also be generated for the tetrahedral mesh (Bey 1995; Liu and Joe 1996; Molino et al. 2003)，in which the red refinement splits the tetrahedron into 8 sub-tetrahedra (Figure 6.7 a). For the green refinement, it can be proved that three types of green subdivision (Figure 6.7 (b), (c), and (d)) are required to recover the conformity. Algorithms for generating such refinements can be found in Liu and Joe (1996)



(a)

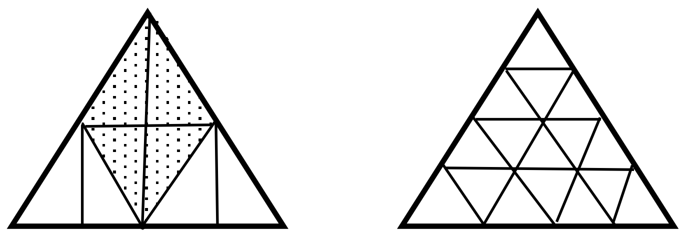
(b) (c) (d)

Figure 6.7 Red and green refinement for tetrahedral elements

6.3.3 Comparisons of the bisecion method and regular method

In the FEM, the choice of the element refinement scheme is primarily a matter of taste of the engineer and the convenience for the computer implementation. Bisection and red-green refinement strategies have both found applications in the FEM. For instance, the bisection algorithm has been implemented in the FEM code ALBERT (Schmidt and Siebert 2000) and the red-green refinement has been adopted in AGM^{3D} (Bey 1995). To investigate the performance of these two categories of the subdivision in the limit analysis for the von Mises material, Christiansen and Edmund (2001) have compared the performances of the two types of refinement procedures in two dimensional cases. From their results, no significant differences are noted except that the bisection refinement appeared more likely to generate ill-conditioned optimization problems in comparison to the regular refinement for limit analysis.

Consider a refinement of one single triangle after three times of refinements by bisection and regular division respectively (Figure 6.8a). A hidden dependence of the linear constraints will be generated due to the elements in the shaded area (Figure 6.8a) (Makrodimopoulos and Martin 2006). This situation is unlikely to occur in the regular refinement scheme (Figure 6.8b). However, it could be avoided locally by traversing the mesh from node to node rather than by edge during the assembly of the inter-element equilibrium equations as suggested by Makrodimopoulos and Martin (2006). The advantage in using the bisection method is that it allows the initial mesh, in particularly a very coarse mesh, to adapt in a moderate rate and distribute the error more optimally as it bisect the target element into two children instead of 4 in the regular refinement. For stability problems with large transitional zone, e.g., the bearing capacity problem with large frictional angle, a huge number of elements can be generated by a regular refinement algorithm in several refinement steps. Based on the current study, it is found that the bisection method can be an attractive approach if the limitation of this method as mentioned is carefully eliminated during adaptive refinement.



(a) (b)

Figure 6.8 (a) mesh after three bisection of a triangle and (b) the mesh after three regular refinements

For the three dimensional FELA, there are 6 independent variables associated for each stress point as opposed to 3 in the displacement-based finite element method. It is found in this research that solving a model with elements more than 30,000 will become difficult. It is preferred that elements are reasonably limited within a moderate number during the refinement. In this sense, the bisection algorithm is a better choice than the regular refinement in terms of model size in the refinement. Furthermore, regarding the implementation of the two algorithms, the green closure for eliminating hanging nodes is tedious and overcomplicated for three dimensional cases.

6.3.4 Consideration of the singularity

It is well known that for problems with singularities, the fan elements would results in much better bound solutions. The local mesh refinement has been studied by Munoz et al. (2009), and two strategies as shown in Figure 6.1(c) and Figure 6.1(d) have been proposed to solve the "locking" phenomenon due to stress singularities. However, the major problem of these two strategies is the regularity of the element not being maintained, and ill-conditioned optimization problem would be generated as the refinement proceeds. To improve the performance of the local mesh refinement in such cases, it is considered more convenient to apply a generic semi-circular zone with fan element at the stage of mesh preparation as shown in Figure 6.9. This zone can generated easily with common mesh generation algorithms and regular refinement can be applied thereafter. Alternatively, using the fan elements generation algorithms (Lyamin and Sloan 2003) for the initial mesh is another approach for simple problems.

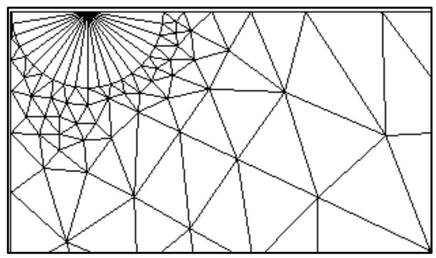


Figure 6.9 Initial mesh with fan elements

6.4 Flagging strategies

When flagging elements for refinement or coarsening, a threshold needs to be set to identify the elements targeted for adaptation. Alternatively, A statistical strategy for marking finite elements for refinement was described by Kirk et al. (2006) while the earlier ideas can be traced back to the work by Carey (1997). The elements are flagged based on the prescribed fraction r_r and c_f as shown in Figure 6.10, in which μ and σ are the mean and standard deviation of the error indicator “population”.

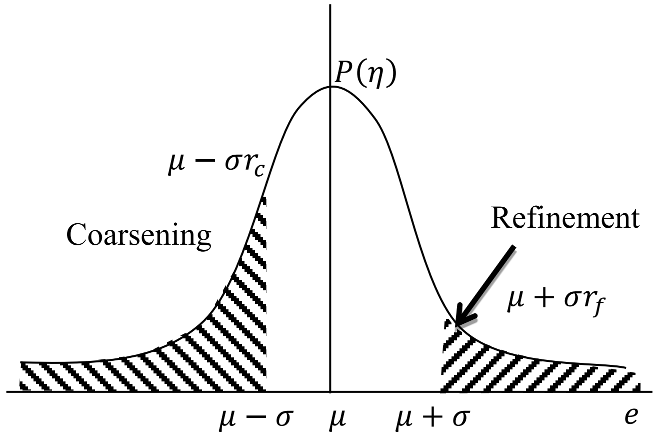


Figure 6.10 Statistical refinement scheme

The element error η is assumed to be assocated with a approximately normal probability density function $P(\eta)$ with mean μ and standard deviation r. Elements with error estimated grater than $\mu + \sigma\, r_f$ are to be flagged for refinement in the subsequent iteration, while those with errors less than $\eta\ < \mu - \sigma\, c_f$ are flagged for coarsening.The majority of the material with assumption of rigid plasticity at collapse is in rigid state and hence the implementation of the mesh coarsening appears more interesting then in the context of elastoplastic fintie element; however, it is not as easy to handle the coarsening process as in finite element method owing to the difficulty in imposing the constraints on hanging nodes. In views of the increased computation power for the modern computer system, coarsening appears to be less important than refining a mesh. Mesh adaptation will be restricted in this work to the mesh refinement only; nevertheless, coarsening could be readily developed when considered necessary.

6.5 Comparison of the various refinement criteria

For the benchmarking purpose, a standard test is to use the rigid footing resting on soil mass for comparison, and the failure mechanism is the well-known Prandtl mechanism which is shown in Figure 6.11.

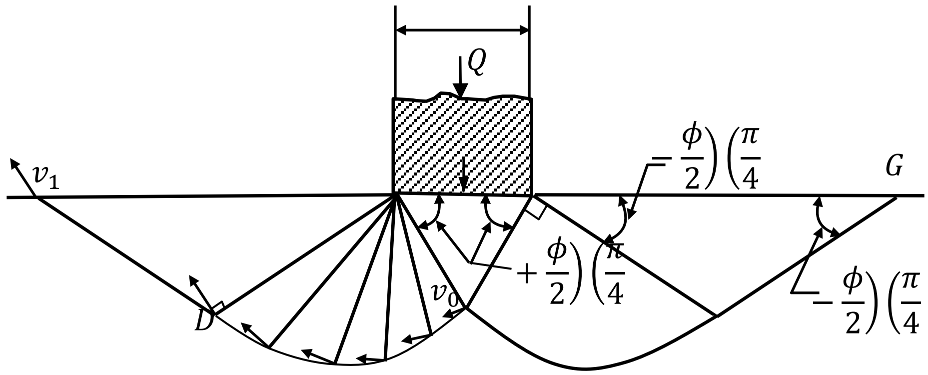
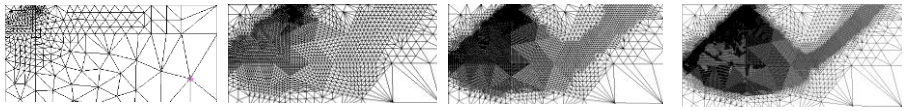


Figure 6.11 Prandtl mechanism

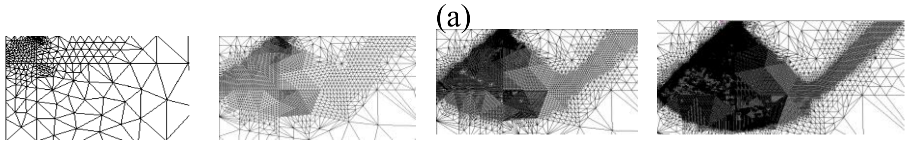
The closed form solution for N_c is given by Prandtl (1920),

$$N_c = \left[e^{\pi \tan \phi} \tan^2\left(\frac{\pi}{4} + \frac{\pi}{2}\right) - 1\right] \cdot \cot \phi \tag{6.31}$$

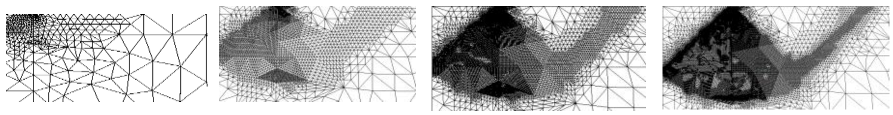
To test the performance of the different steering strategies for the mesh adaptation with an unstructured initial mesh, prior knowledge of the singularity will not be used in the solution of the problem. Figure 6.12 shows the detailed refined meshes with various refinement steering strategies.



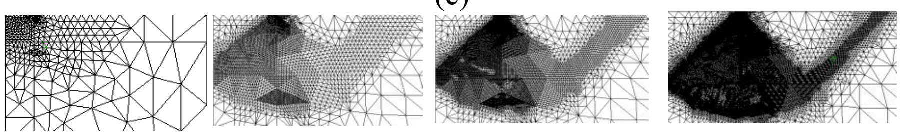
(a)



(b)



(c)



(d)

Figure 6.12 (a) yield function slackness based refinement with $R = 10^{-6}$; (b) equivalent strain (dual variables) based refinement with refinement ratio $r = 0.99$; (c) Jump error estimator with refinement ratio $r = 0.99$ and (d) patch recovery error estimate (gradient recovery) with refinement ratio $r = 0.99$

The author has carried out such tests on many different problems, and has noted the following issues:

1. Different error estimators as discussed earlier do not show obvious superiority over each other in capturing the slip bands, provided that the refinement ratio is carefully selected. It might be concluded that the captured slip bands (or refined region) is more controlled by the refinement ratio than by the indicator itself, *i.e*. the plastic deformed region can theoretically be identified through the yield functions or the dual variables (can be interpreted as equivalent strain) associated with them due to the complimentary principle. The observation (Ciria et al. 2008) that deformation-based error estimator failed to reproduce the failure bands in the plane stress example seems odd, and the cause of the failure in their work appears more likely to be caused the

threshold chosen for the refinement instead of the deformation-based refinement strategy.

2. Patch recovery strategy and Kelly jump error strategy might slightly over-refine the slip bands, because the neighbours for element Ω^e might be assigned the "error" due to the difference even though it might be located in the rigid zone. This is totally unnecessary if only the error of solutions is considered. However, this is less obvious from the refined meshes in Figure 6.12 because we have selected different refinement ratios to take into account this effect in advance with a smaller refinement ratio than the deformation based strategy.

To further investigate the performance of the error estimator in the FELA, we deliberately investigated the formulation for upper bound with no velocity discontinuities allowed between adjacent elements (Makrodimopoulos and Martin 2007). The velocities have been selected as the controlling variables. In this case, the velocity field resembles the displacement field obtained in finite element method. Patch recovery strategy and jump error strategies captured the slip bands as well as those in Figure 6.12, notwithstanding the absence of the theoretical ground for the existence of the superconvergence.

In order to demonstrate that the error is more controlled by the degrees of freedom in the slip band than the grid points over the entire mesh, a comparison study is prepared between the uniform refinement (more grid points) and the adaptive refinement. The bound solutions are found to be almost identical at each refinement step (see Table 6.1), implying that all element that are required to be refined has been refined. It is also noted that the error due to the refinement is not a simple linear relation as shown in Figure 6.13, and extrapolation of the solution can be difficult for general problem as the relation in Figure 6.13 may not be simple for general condition.

Table 6.1 Comparison of Adaptive refinement with uniform refinement for r_f=0.99

Adaptive refinement		Uniform refinement	
Elements	N_c	Elements	N_c
268	4.94534	268	4.94534
621	5.03984	1072	5.03986
1729	5.09062	4288	5.09058
5465	5.11627	17152	5.11628
17562	5.12879	68608	5.12844
47606	5.13476	--	--
78922	5.13724	--	--

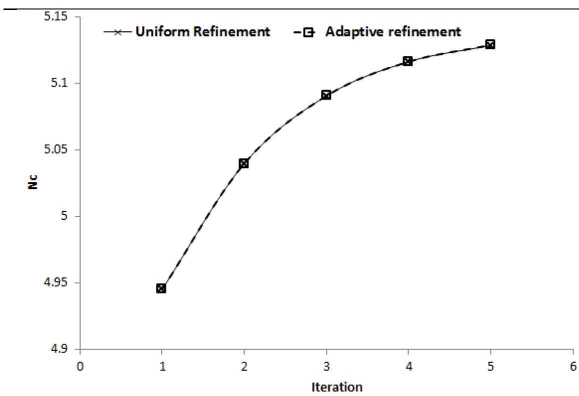


Figure 6.13 N_c vs. iteration

6.6 Comparison of the element splitting methods

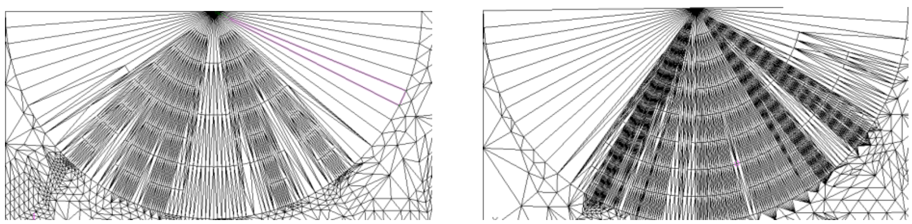
It is commented (Christiansen and Edmund 2001) that the meshes obtained by the bisection algorithm are more likely to lead to ill-conditioned optimization problem. In solving the resulting optimization with MOSEK, this ill-conditioning situation was found in this work for many cases, and “near-optimal” solutions are obtained for the bisection methods for many problems. This problem was overcome by a node-by-node assembly strategy to eliminate the hidden dependency, and it is uncommon for a model with moderate number of elements to come across this problem. The performance of the various splitting methods is given in Table 6.2.

Table 6.2 Comparison of bisection refinement and the regular refinement for 2D case

Bisection			Regular refinement			Fan+Regular refinement		
Iters	Elements	N_c	Iters	Elements	N_c	Iters	Elements	N_c
1	330	4.945	1	441	5.002	1	335	5.040
3	707	5.085	2	1214	5.075	2	809	5.112
5	1799	5.118	3	3551	5.107	3	1354	5.131
7	5335	5.131	4	6701	5.123	4	3211	5.138
9	13445	5.136	5	21244	5.132	5	5536	5.139
11	26613	5.139	6	55875	5.136	6	12302	5.140

Table 6.2 shows that the bisection method generally arrives at better lower bound solutions with fewer elements. This feature is more important in the application for 3D analysis, for which the extension of the regular refinement would generate 8 children as opposed to 2 in the bisection method.

The use and performance of the fan zone is very impressive, and a very accurate solution could be obtained with 1354 elements. The refinement mesh close to the singularity is shown in Figure 6.14 a (not the complete mesh). For the case $\phi' = 0$, the use of semi-circle fan zone in the initial coarse mesh will definitely greatly improve the solution. Even for the case when $\phi' = 30^0$, the semi-circle zone fan elements in the initial coarse mesh can still dramatically improve the solution. It must be pointed out that for the case when $\phi' = 30^0$, the final slip band is practically a log-spiral curve even with an initial semi-circular fan zone.



$(a)\ \phi' = 0$ $(b)\ \phi' = 30$

Figure 6.14 Magnified final mesh generated with fan zone and regular refinement for region close to the singularity

It is evident that the use of fan elements significantly improves the solutions of the problem with discontinuous loading. However, the locking due to the singularity point is attributed to the arrangement of the stress discontinuity rather than the singularity itself, because as shown in Figure 6.14, the point of singularity is not refined at each refinement. It might be argued that the reason for this is because the fan zone with a semi-circle resemble the discontinuities for material with $\phi' = 0$. The numerical example for material with $\phi' = 30^0$ has also been prepared with the refined fans shown in Figure 6.14(b). It is noted that results with a semi-circle regain are still preferable than the unstructured mesh. It is concluded that taking account of the discontinuities in the initial mesh would guide the local refinement and thus could appreciably enhance the quality of the bound solution.

Despite the advantages of the fan elements, preparation for such specialised mesh is still considered demanding, particularly for general problems. It will be demonstrated in the following examples that satisfactory solutions without the fan elements could be obtained with the standard bisection method and jump error estimator, i.e. the locking due to the singularity in the lower bound FELA is not significant for a general unstructured mesh. As discussed earlier, the intention of the bisection method is to avoid huge models generated when a large transitional zone is anticipated and to generate a well graded mesh for good accuracy. Since the choice of the mesh steering strategies practically does not significantly affect the result, the jump error estimator will be used for its simplicity.

6.7 Flow chart of mesh adaptation in the FELA

It is desirable that the mesh refinement is performed automatically and the reliance on the experience of the analyst must be reduced to the minimum. This requires the data structure to be flexible such that the information of the mesh, boundary condition and the soil strengths can be transferred and mapped in the hierarchic mesh.

The procedure of the mesh adaptation in the FELA could be described as (Figure 6.15):

(1) Supply the inputs of the analysis, including the finite element mesh, boundary information, material information (yield criterion), etc. In many cases, we may prefer to include the discontinuities in the analysis, which requires a different meshing such that each node in the mesh is uniquely possessed by one element. This could be achieved by processing the FEM mesh to add the additional nodes. (Bandini, 2003). However, in this research, we will keep the finite element mesh and address the discontinuities in the assemblage stage by renumbering the nodes because all the coordinates are sufficient to define the finite element mesh. In the implementation, a third-party pre-processor is used to generate the finite element mesh, applying the boundary condition and assigning the material information. An

intermediate data file will be generated and will be read by the FELA for the interpretation.

(2) Assemble the upper bound or lower bound problem based on the formulation discussed in the previous chapters for standard convex programming. This process is quite similar to the finite element method. Elemental matrix is formed and assembled into the global matrix guided by a “position vector”. What distinguishes an assemblage process of the FELA from the FEM is that, other than the global matrix (linear constraints); there are objective vectors and matrices for the nonlinear constraints.

(3) Solve the resulting optimization problem with different optimization algorithms. Currently the third-party solver Knitro and IPOPT for the NLP and MOSEK for the SOCP have been incorporated and selected depending on the nature of the optimization.

(4) Compute the indicators for mesh adaptation according to the prescribed rules and flag the elements associated with large error estimators.

(5) Refine the flagged elements. Elements flagged for the refinement are split according the error calculated in (4) with either the bisection algorithm or the red-green algorithm.

(6) Update the system, i.e. to transfer the information from the initial mash to the updated decedent and to form a new limit analysis problem.

(7) Check if the termination criterion is satisfied. In the applications, we may want to refine the mesh for a particular number, say 4 times; or the improvement of the solution is less than a certain ratio or the maximum number of the active element in the mesh should not exceed a certain number.

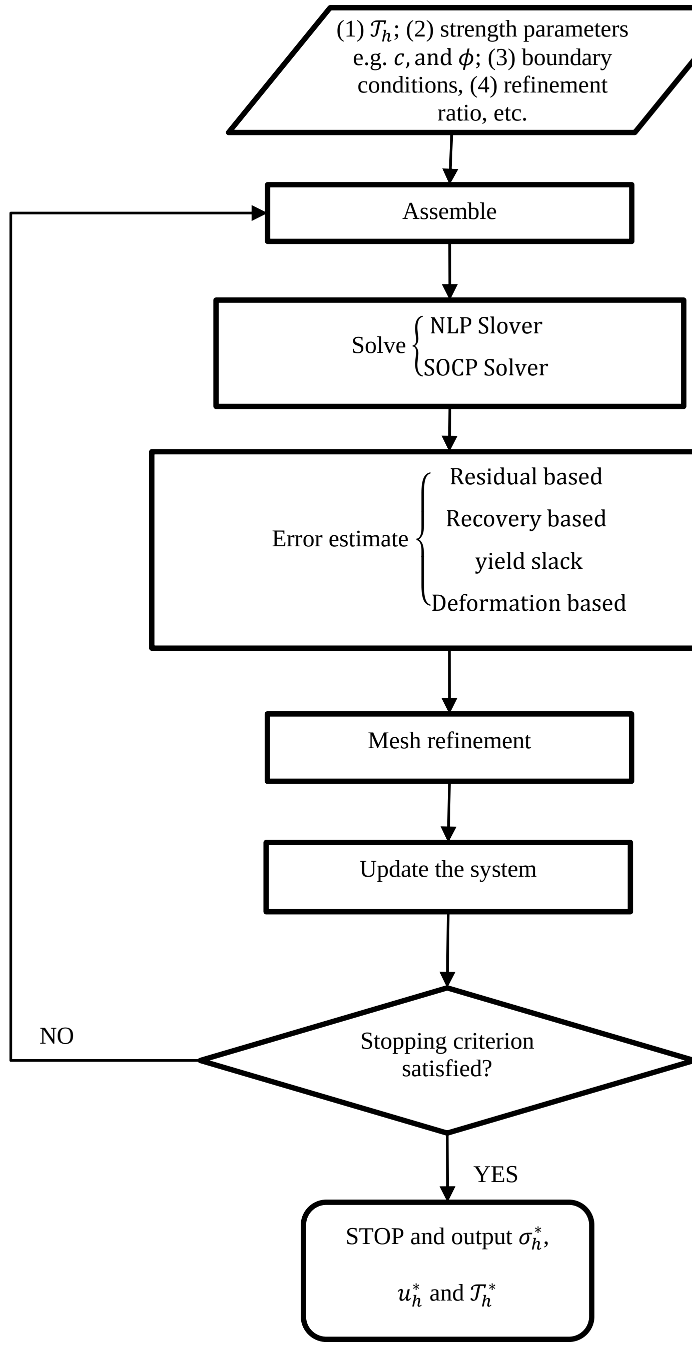


Figure 6.15 Flow chart for the adpative mesh refinement in the FELA

To facilitate the implementation and to improve the efficiency of computation procedure, C++ is adopted as the programming language as opposed to the Matlab code by Ciria (2002) and Fortran code by Lyamin et al. (1998). Large models over 300,000 triangle elements generated by the mesh adaptation procedure has been solved with MOSEK for SOCP (MOSEK-Aps 2010).

The local refinement strategy requires a more sophisticated data structure for the mesh storage in terms of the computer implementation. For example, the binary tree is commonly used to represent the hierarchic mesh. However, it is more powerful to work with if the data structure has been constructed, as the process of mesh adaptation is totally automatic without referring to a general mesh subroutine which is usually expensive and demanding. A fairly coarse mesh can be fed as the input for a general problem. This feature of the current procedure is of great practical significance as it removes the need to prepare a relatively good mesh for complex problems and the development of an appropriate discretisation can be processed under a more systematic framework.

6.8 Summary

Mesh adaptation has been discussed in this chapter. Error estimators based on the residual and recovery developed in the FEM have been tailored for the FELA. The performance of the error estimators using kinematical solution as the control variable shows that these techniques work equally well in comparison to the direct approaches by directly evaluating the yield functions.

The bisection and regular splitting algorithm have received equal acceptance in the FEM, while in the application to the FELA, bisection algorithm seems to be a better choice than the regular refinement to avoid large model generated within a few iteration. It has long been recognised that fan elements provide great benefits for the problem with stress singularities. Generating such specialised mesh depends on the nature of the problem under consideration which requires a case-by-case treatment. As will be shown in the next chapter, good limit solutions can also be obtained under a general mesh adaptation scheme without any special consideration for the initial solution/mesh.

CHAPTER 7: GEOTECHNICAL CONSIDERATIONS IN LIMIT ANALYSIS

7.1 Introduction

In addition to the static and kinematic constraints required in the standard formulation of FELA, this chapter discuss some issues that need to be addressed for the modelling of more realistic stability analysis problems in geotechnical engineering.

7.2 Pore-water pressure

As noted by (Bishop 1966)"all measurable effects of the change of stress, such as the compression, distortion and a change in the shearing resistance are exclusively due to changes in effect stress". The increase in the pore water pressure and the decrease in the effective stress is the main reason for the slope failures in Hong Kong and many other countries. A proper treatment of the pore water pressure is of great significance for the practical applications of FELA.

7.2.1 Consideration of the pore water pressure in the FELA

The effects of the water pressure can be incorporated in the analysis by expressing the formulation in terms of the effective stresses, provided that adequate information about the porewater pressure is available.

The distribution of the porewater pressure can be established by calculating the pore-water pressure at each node from a flow net or a predefined phreatic surface and interpolating the values with the same shape function as those for the stress variables according to eq. (7.1)

$$\sigma_{pp}^{e} = \sum_{a=1}^{NPE} N_a^e(x)\sigma_{pp}^{a,e} \tag{7.1}$$

Expressing the stress equilibrium condition in terms of effective stresses leads to

$$\underbrace{\sum_{j=1}^{NPE} \frac{\partial \sigma_{ij}}{\partial x_j}}_{1} + \underbrace{\frac{\partial \sigma_{pp}}{\partial x_i}}_{2} = f_i \tag{7.2}$$

Assuming that the pore water pressure is independent of the load multiplier, the pore water pressure can be regarded as a body force acting on the soil mass. Moving term 2 in (7.2) to the right hand side yields

$$\underbrace{\sum_{j=1}^{NPE} \frac{\partial \sigma_{ij}}{\partial x_j}}_{1} = \underbrace{f_i}_{2} - \underbrace{\sum_{a=1}^{NPE} \frac{\partial N_a^e(x)}{\partial x_i}\sigma_{pp}^{a,e}}_{3} \tag{7.3}$$

It should be noted that the terms 1 and 2 remain unchanged as in (4.5) and an added term 3 will enter into the vector $\mathbf{F}^e$ in (4.5). Expanding the term 3 in (7.3)

$$\sum_{a=1}^{NPE} \frac{\partial N_a^e(x)}{\partial x_i}\sigma_{pp}^{a,e} = [\partial N_{1,i}^e(x) \quad \partial N_{2,i}^e(x) \quad \dots \quad \partial N_{NPE,i}^e(x)] \begin{bmatrix} \sigma_{pp}^{1,e} \\ \sigma_{pp}^{2,e} \\ \vdots \\ \sigma_{pp}^{NPE,e} \end{bmatrix} = \mathbf{N}_i^e \boldsymbol{\sigma}_{pp}^e$$

Therefore, $\mathbf{F}^e$ in (4.5) will become

$$\mathbf{F}^e = -\left(f_i + \mathbf{N}_i^e \boldsymbol{\sigma}_{pp}^e, \dots, f_D + \mathbf{N}_D^e \boldsymbol{\sigma}_{pp}^e\right)^T$$

Regarding the stress equilibrium along stress discontinuities, the effective tractions on the discontinuities shared by the neighbouring elements are required to be equal, i.e.

$$\left(\sigma_{km}^e - \sigma_{km}^e \delta_{ij}\right) = \left(\sigma_{ij}^e - \sigma_{pp}^{e'} \delta_{ij}\right) \Leftrightarrow {\sigma'_{km}}^{e} = {\sigma'_{km}}^{e'} \tag{7.4}$$

Note that (7.4) is the same as the (4.11) because the pore-water pressures on the adjacent elements are balanced automatically.

For the stress boundary conditions, modifications could be made by expressing (4.16) in terms of the effective stresses.

In the upper bound formulation, the work equation of the form (7.5) are expressed using the effective stresses as well,

$$\underbrace{a(\widehat{\boldsymbol{\sigma}}', \mathbf{u})}_{1} = \underbrace{F(\mathbf{u})}_{2} \tag{7.5}$$

Expanding term 1 in (7.5) leads to

$$\begin{aligned} P_{int}(\boldsymbol{\sigma}', \mathbf{u}) &= \int_{\boldsymbol{\Omega}} \left(\widehat{\boldsymbol{\sigma}} - \sigma_{pp}\mathbf{I}\right) : \hat{\boldsymbol{\varepsilon}}\, d\Omega = \int_{\boldsymbol{\Omega}} \widehat{\boldsymbol{\sigma}} : \hat{\boldsymbol{\varepsilon}} d\Omega - \underbrace{\int_{\boldsymbol{\Omega}} \sigma_{pp} \varepsilon_v \boldsymbol{d\Omega}}_{\boldsymbol{d_w(u)}} \\ &= P_{int}(\widehat{\boldsymbol{\sigma}}, \mathbf{u}) - d_w(\mathbf{u}) \end{aligned} \tag{7.6}$$

As can be seen from (7.6), the consideration of the pore water pressure will give rise to an added term $d_w(\mathbf{u})$ **, i.e.,** work done due to the pore water pressure**.** Since the pore water pressure is predefined, it has been argued by Kim (1998) that $d_w(\mathbf{u})$ should be interpreted as the external work done. It is preferred that a direct interpretation of $d_w(\mathbf{u})$ is expressed as a reduction of the power dissipation because such explanation is in consistence with the concept of effective stresses, i.e., the increase of the pore water pressure would lead to a reduction in the effective stresses. It should be noted that both interpretations give numerically correct solutions, and it is more convenient to handle the pore water pressure as external loading by moving it to the right hand side of (7.5) .

7.2.2 Calculation of the pore water pressure

Realistic analyses of the stability problem with the presence of water should utilize the effective strength parameters, which are only meaningful when they are used in conjunction with the pore-water pressure. In this sense, the pore-water pressure is as important in establishing the correct failure state as the strength parameters themselves.

For the hydrostatic case, the porewater pressure can be calculated from a determined phreatic surface based on observation or using the pore-water pressure ratio R_u as proposed by Bishop (1966). For steady flow case, a flow net can be obtained from the coupled seepage analysis for the evaluation of the pore water pressure.

When the phreatic surface is used to define the pore water pressure, the exact distribution of the pore-water pressure could be determined by an approximate method as given by Krahn (2004). The same technique has been adopted by Kim (1998) and Kim et al. (2002) for slope stability analysis using the LP formulation of FELA. This procedure will also be adopted in the current study.

Figure 7.1 illustrates a simplified phreatic surface. At a given point O, two heads could be estimated, the vertical pressure head H_w and the perpendicular pressure head H_c. The vertical pressure head is simply the vertical distance from the point O to the phreatic surface immediately above it. On the other hand, the perpendicular pressure head is the distance between points O to intersection of the normal to the phreatic surface P . The actual head is determined by the flow net which lies somewhere between these two values. The vertical head, as a conservative estimate of the actual pore-water pressure could overestimate the pore-water pressure by as much as 30% when the slope of the phreatic surface is up to 35%, while the perpendicular estimate may slightly underestimate the actual pore water pressure by approximately 10%. Therefore, a conservative estimate of the pore-water pressure is given by the mean of these two estimates as in eq. (7.7)

$$H = \frac{1}{2}(H_c + H_w) \tag{7.7}$$

(7.7) is a conservative estimate of the porewater pressure defined by a phreatic surface (Achilleos 1988).

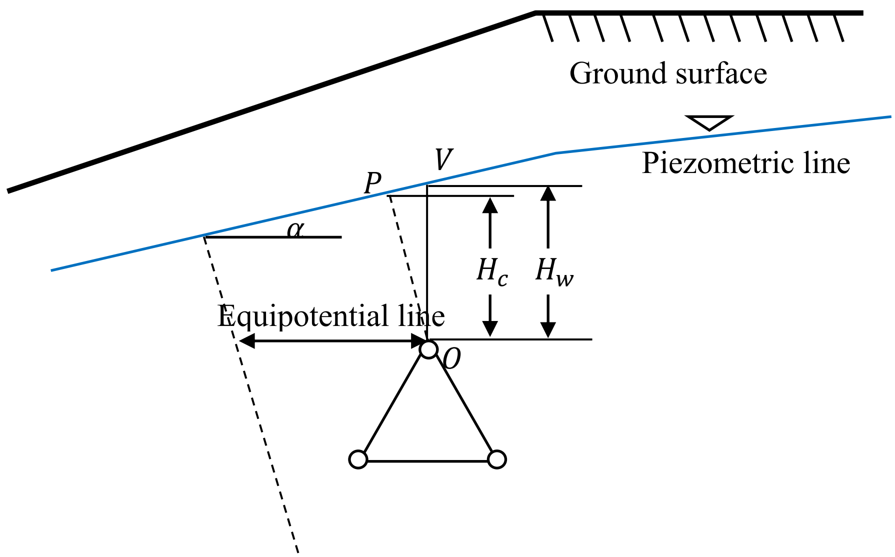


Figure 7.1 Phreatic surface correction

Alternatively, a relatively simple method for describing the distribution of the pore-water pressure is to use a concept of pore water pressure ratio which is defined as

$$R_u = \frac{u}{\gamma z} \tag{7.8}$$

where u = pore-water pressure, γ = total unit weight of the overburden and z = the height of the soil column. When the phreatic surface is not parallel to the ground surface, R_u is not a constant but vary throughout the soil profile, which destroys the simplicity of this concept. This approach is not realistic in practice, but it can however provide a rapid assessment of slope stability and is still available in most of the commercial programs.

7.3 Seismic loading

It is important to consider the effect of earthquake in stability analyses in the seismic zone. Because of the oscillatory nature of the seismic loading, it is certain that a dynamic analysis is more realistic. However, dynamic analysis is seldom carried out in the design considering the seismic effects because there are various uncertainties associated with the earthquake effect. A quasi-static approach, in which the seismic loading is modelled as a permanent inertia force, is commonly adopted for practical purposes. This concept was proposed by Mononobe and Matsuo (1929) in the Coulomb's mechanism and has been widely accepted by Morrison and Ebeling (1995) in the limit equilibrium analysis with a log-spiral surface, by Chen and Liu (1990) with the upper bound approach using log-spiral failure surface and Cheng (2003) with the slip line method. In the FELA, the same approach can be used to incorporate the effects of earthquake loading on the stability problems. It should be noted that the effects of an earthquake is much richer than simply applying an equivalent inertia load, e.g., cyclic softening or liquefaction should be assessed when cohesionless soils are susceptible to seismic effects. However, this topic is beyond the scope of this thesis.

Including the seismic effects by the quasi-static method can be achieved by applying additional body force. In plane strain analysis, the static equilibrium condition for a plane strain analysis will change to (7.9)

$$\begin{aligned}\frac{\partial\sigma_x}{\partial x}+\frac{\partial\tau_{xy}}{\partial y}&=-k_h\gamma\\ \frac{\partial\sigma_y}{\partial y}+\frac{\partial\tau_{xy}}{\partial x}&=-(1+\xi k_h)\gamma\end{aligned}\tag{7.9}$$

where k_h is the horizontal seismic coefficient and ξ is the ratio of the vertical seismic coefficient to the horizontal coefficient. Usually, k_h will take a value between 0.0 and 0.3 and ξ will range 0~0.5. Extension of (7.9) to the full three-dimensional analysis is straightforward and will not be discussed here.

Alternatively, the seismic loading can be considered by modifying the unit gravity of the soil mass as proposed by Cheng (2003). Similar to pore water pressure, the seismic loading which is considered as an inertia force is essentially an additional body force which will enter into the term $\mathbf{F}^e$ in (4.5) in the lower bound formulation. In the upper bound analysis, the rate of external work done due to the modified gravity (or the added work done due to the pure seismic loading) will enter into the objective function in the kinematic formulation. In the dual formulation of the upper bound analysis, the added body force contributes to the elemental equilibrium in an average sense.

7.4 Consideration of the structure elements

The load is transferred to the soil mass by structural elements, footings, piles or retaining structure etc. The behaviour of the soil-structure interface can be crucial to the response of a soil-structure system; in particular, the friction between the structure and the soil can have a major effect on the limit load. In addition to the strength of the soil mass, property of structures and the soil-structure interface may also affect the collapse load that soil can carry. To illustrate, Figure 7.2 and Figure 7.3 show the kinematic results for concentrically loaded footings. For a footing with a perfectly rough base, the velocities of the points on the interface are constrained to be vertical as shown in Figure 7.2 (b), i.e. no relative movement is allowed. If the interface is modelled as smooth as shown in Figure 7.2(a), the velocity will develop freely. These features need to be addressed in the formulation of the FELA.

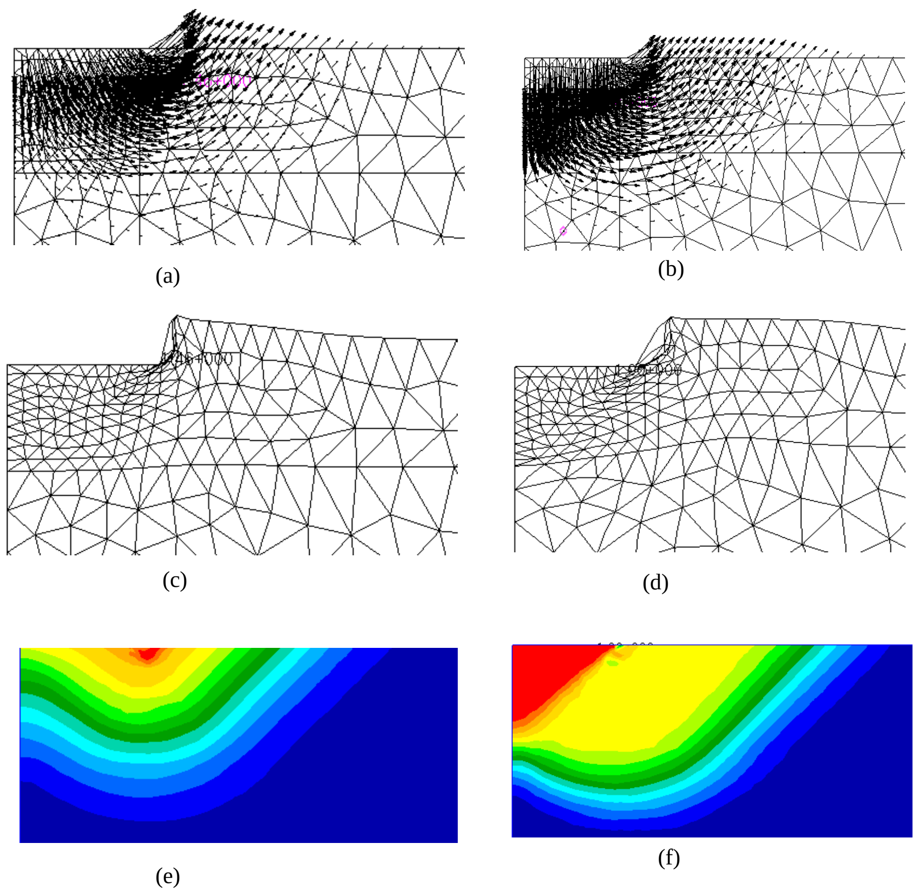


Figure 7.2 Rigid footing with smooth base and rough base, respectively: (a), (b) velocity field for smooth and rough base respectively; (c), (d) deformation for smooth and rough base; (e) and (f) the velocity contour for smooth and rough base.

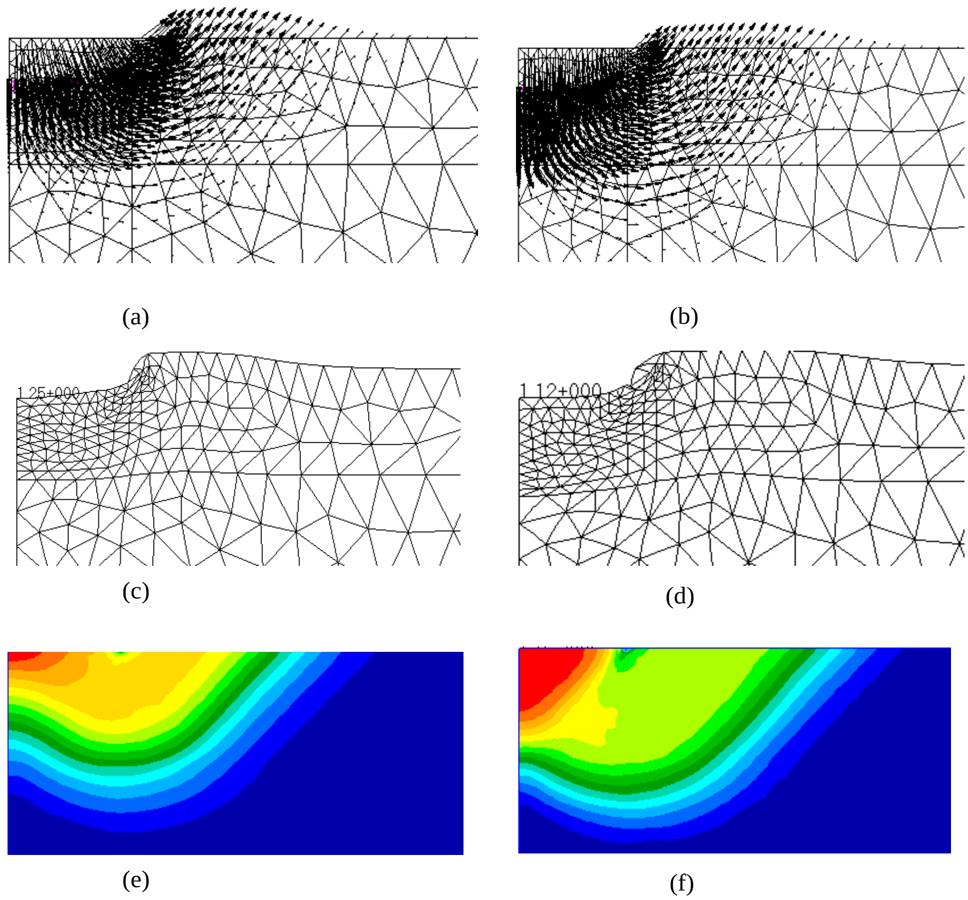


Figure 7.3 Flexible footing with smooth base and rough base, respectively: (a) and (b) velocity field for the smooth and the rough base respectively; (c) and (d) deformation for the smooth and the rough base; (e) and (f) the velocity contour for the smooth and the rough base.

7.4.1 Roughness of the soil-structure interface

The condition has been considered in the work of Bandini (2003), for the bearing capacity problems under plane strain condition. The perfectly smooth or perfectly rough interface condition could be modelled by applying additional constraints to the stresses and velocities respectively. For perfectly rough interface where no slippage is allowed, the velocity tangential to the interface is constrained to be zero and stresses are allowed to develop freely. For a footing with a smooth base, the shear stresses on the interface should be restrained to zero, i.e., additional constraints should be imposed on the optimization problem.

For perfectly rough interface

$$u'_i = \sum_{i=1}^{D} u_j \beta_{ji} = 0, i = 2, \dots, D \tag{7.10}$$

where the prime denotes the expressions in the local system.

Stress constraints for perfectly smooth condition,

$$\sigma'_{1m} = \sigma_{ij} \beta_{1i} \beta_{jm} = 0, m = 2, \dots, D \tag{7.11}$$

Note that in (7.10) and (7.11); we assume that x'_1 is the axis of the local coordinate that is parallel to the normal of the boundary.

7.4.2 Rigidity of the structure

In the FEM, structures could be modelled as rigid, elastic and flexible. However, it is difficult to reflect the varying degrees of rigidity of structures in the FELA as the deformation properties is not included in the FELA, e.g. E, ν etc. However, structures could be simplified either as rigid or flexible by imposing more constraints.

A flexible structure implies that the structure is sufficiently flexible to allow the stress distribution mobilised in the soil mass along the interface to resemble the applied external loading. In such cases, a load profile could be applied in the lower bound analysis and a multiplier can then be optimised, such as the loading on the footing Figure 7.4. For the upper bound formulation with velocities as the optimal variables, the flexible footing does not impose any additional constraints on the velocity field on the interface, since the interface could deform freely.

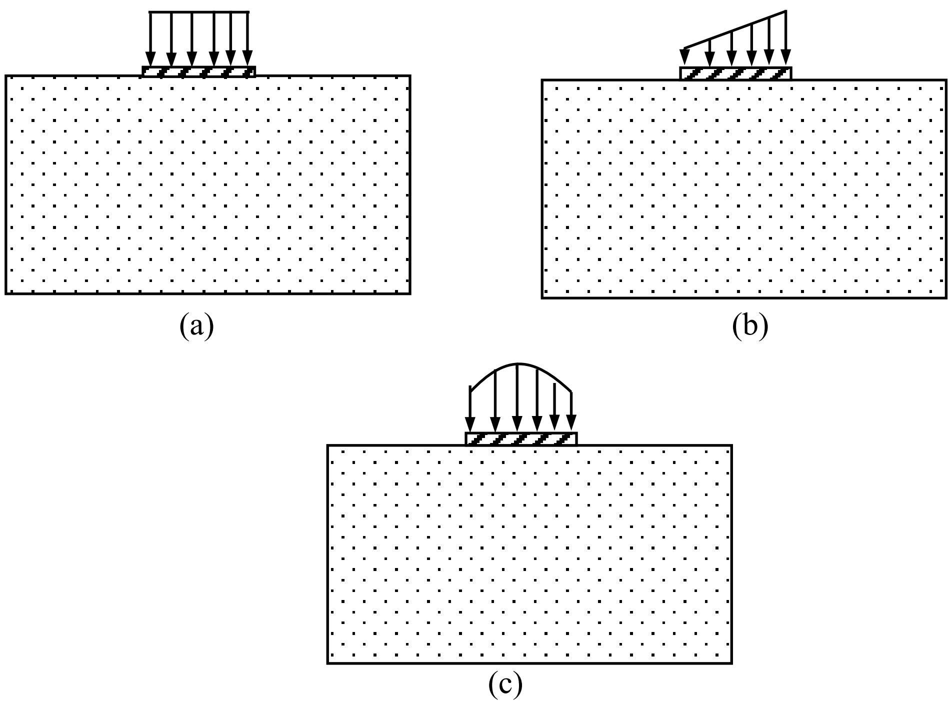


Figure 7.4 Various load profiles for the flexible footing

Rigid structures imply that velocities of the nodes on the soil structure interface need to be restricted while the stress could develop freely.

On the other hand, for the rigid footing, the velocity on the interface needs to be constrained such that the structure moves in a rigid motion. For instance, additional constraints as (7.13) will be required for a translational displaced shallow foundation in an upper bound formulation.

$$(u_1')_n = (u_1')_{n+1} \tag{7.12}$$

where, $u' =$ the vertical component of the velocity in a local coordinate, $n = 1, \dots, N^{\xi_e^N}$ and $N^{\xi_e^N}$ is the number of nodes on the interface.

In the lower bound formulation, the rigid condition can be fulfilled by optimising the resultant force as in (7.13).

$$Q = \int_{\Gamma^N} \sigma_{11}' dS \quad \Gamma^N \text{ refers to the base of the footing} \tag{7.13}$$

where $\sigma_{11}' =$ the stress component in a local coordinate.

7.5 Limit analysis for the rock masses

The most significant feature that distinguishes rock masses from soils is the existence of discontinuities. With the discontinuous field, this feature could be modelled even more easily in the FELA than in the FEM, provided that the strength parameters of the joints, faults, etc. can be obtained. Such analyses resemble the multi-block techniques or the rigid finite element method.

For heavily jointed rock masses, rock mass are basically treated as soils expect that a yield criterion different from the Mohr-Coulomb criterion is used. Hoek-Brown yield criterion is the most widely accepted yield criterion for the study of the rock masses. Incorporating the Hoek Brown yield criterion in the framework of the limit analysis, particularly for the FELA, has been studied by Lyamin et al. (1998) for the original Hoek-Brown yield criterion (Hoek and Brown 1988), Merifield et al. (2006) for the generalized Hoek-Brown yield criterion (Hoek et al. 2002) and more recently by Li et al. (2008), Li et al. (2009a) and Li et al. (2011).

With the FELA, the analysis of the HB material is no more difficult than the MC material in view of convex programming. As discussed on page 71, the HB yield envelope suffers from the similar problem as the MC yield envelope in that singularities exist at the corners and apex. Smoothing technique is required when it is applied in convex programming.

A quasi-hyperbolic smoothing technique for the HB yield envelope under the plane strain condition was proposed by Merifield et al. (2006) in which $\bar{\sigma}$ is permuted with a small value ε

$$\hat{J}_2 = \sqrt{\bar{\sigma}^2 + \varepsilon^2} \tag{7.14}$$

On the condition that ε is related to the strength of the material by

$$\varepsilon = \min(\delta, \mu\rho | \rho g(0) + (\rho h(0) + \chi)^{\alpha} = 0) \tag{7.15}$$

Then the approximated HB yield criterion can be written as

$$f = \hat{J}_2 g(\theta) + \left(\hat{J}_2 h(\theta) + \beta I_1 + \chi\right)^{a} \tag{7.16}$$

where $g(\theta)$ and $h(\theta)$ are defined in (3.60).

Once the yield criterion is smoothed, the technique introduced for the manipulation of the general yield criteria (see page 87) could be applied to obtain the gradient and Hessian of the yield criterion. In our code, a Class “*HoekBrown*” derived from the “*YieldCriterion*” is designed to provide the function value, gradient and Hessian corresponding to a specific stress state.

7.6 Nonlinearity of yield envelopes

Despite the simplicity and wide acceptance in soil mechanics, the MC yield criterion is a linear approximation of the actual yield envelope, which is essentially nonlinear. As illustrated in Figure 7.5, the linear approximation (the MC yield criterion) of the yield criterion vary appreciably depending on the stress range within which the linear regression is preformed, i.e., material parameters c' and ϕ' depend on the stress range and are not real constants throughout the solution domain or the analysis process. If an approximation over the whole range of the stress interval is taken in the lower stress range zone I, c' will be underestimated while the friction angle ϕ' will be overestimated. In zone III, c' will be overestimated while the friction angle ϕ' will be underestimated. To make a better use of the nonlinear yield strength, an iterative procedure can be taken when using a linear approximation of the nonlinear yield criterion. Assuming a stress interval initially, the problem under consideration can then be solved with the linear approximation. After that, the stress level can be back calculated to check whether the assumed stress range is indeed valid. If the back computed stress level is different from the original assumption, the initial stress interval is modified accordingly until the analysis converges. It is therefore preferable that the yield information could be obtained at each point in the natural form without approximation.

The strength at each stress point is determined by a "tangent" technique in slope stability analysis using limit equilibrium method or upper bound analysis (Chen 1975; Collins et al. 1988; Drescher and Christopoulos 1988; Yang and Yin 2004; Zhang and Chen 1987). The strength at a stress point M is replaced by a tangent line with constants c_t and ϕ_t that always serve as an upper bound to the exact strength as shown in Figure 7.6.

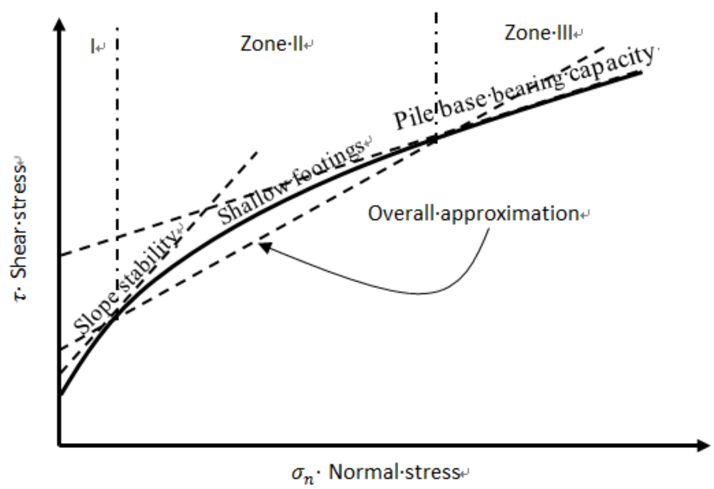


Figure 7.5 Stress range for typical stability problems in geotechnical engineering

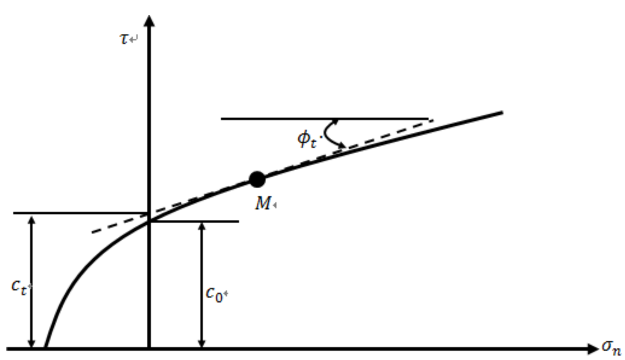


Figure 7.6 Tangential cohesion and tangential friction angle corresponding stress point M

The previous works discuss the nonlinearity of the yield criterion under the Mohr plane (τ, σ_n). It is not convenient to work with in the formulation of the FELA as the current formulation is formulated in terms of Cartesian stresses.

The following will attempt to seek the transformation from the Mohr plane to the triaxial plane $\{p, q\}$ where p and q is defined in (7.18), such that the derivatives with respect to the Cartesian stresses can be obtained easily,

The tangential friction angle associated with (7.17) is defined as

$$\tan\phi_t = \frac{d\tau}{d\sigma_n} = \frac{c_0 m'}{\sigma_t}\left(1+\frac{\sigma_n}{\sigma_t}\right)^{m\prime-1} \tag{7.17}$$

From the geometric relationship, p and q can be represented in terms of the normal and shear stress at failure point M as (7.18).

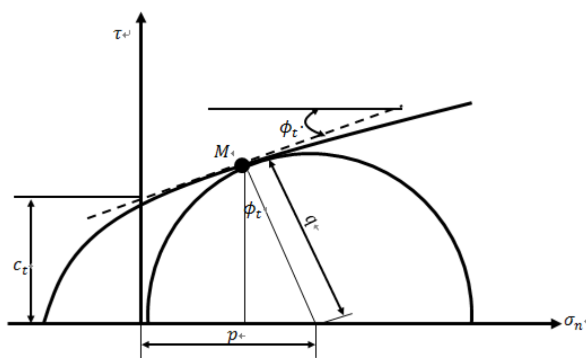


Figure 7.7 Geometric relationship between Mohr stresses and triaxial stresses

$$\begin{aligned} p &= \frac{\sigma_1+\sigma_3}{2} = \sigma_n + \tau\tan\phi_t = \sigma_n + \frac{m' c_0^2}{\sigma_t}\left(1+\frac{\sigma_n}{\sigma_t}\right)^{2m\prime-1} \\ q &= \frac{\sigma_1-\sigma_3}{2} = \tau\sqrt{1+\tan^2\phi_t} = c_0\left(1+\frac{\sigma_n}{\sigma t}\right)^{m\prime}\sqrt{1+\frac{c_0^2 m'^2\left(1+\frac{\sigma_n}{\sigma_t}\right)^{2(m\prime-1)}}{\sigma_t^2}} \end{aligned} \tag{7.18}$$

Equation (7.18) can be regarded as a parametric yield function in which σ_n is the parameter variable. Establishing the explicit representation of the nonlinear yield criterion in triaxial plane involves solving a system of highly nonlinear equations. There is no apparent simple way to achieve this purpose; therefore, a different strategy is proposed and adopted in this research.

By observing the plot of the yield function in the $p-q$ plane, a yield functions in the $p-q$ plane similar to (3.54) is proposed as

$$q = a\left(1+\frac{p}{b}\right)^{c} \tag{7.19}$$

In the present study, the parameters in (7.19) are obtained from a least square curve fit, and data points are sampled from (7.18). Unlike the actual experimental tests, a relatively large number of points can be sampled to ensure the accuracy. The use of the parameters in (7.19) is the solution to the minimisation problem

$$\begin{aligned} &\min \sum_{i=1}^{n}(q(a,b,c,p_i)-q_i) \\ &\text{s.t. } a \geq 0, b \geq 0, 0 \leq c \leq 1.0 \end{aligned} \tag{7.20}$$

where n = the number of the sampled points. In fact, after a close examination of (7.19), the only parameter that needs to be fitted is c, while a and b can be determined directly from (7.18). An optimal solution to the constrained optimization problem given by (7.20) can be found with a high degree of accuracy through heuristic global methods (Cheng et al. 2007a). (7.19) can be rewritten in a general form as shown in (7.21).

$$f(p,q) = q - a\left(1 + \frac{p}{b}\right)^c \tag{7.21}$$

The variables p and q are related to the principal stresses σ_1 and σ_3 as in (7.18), and σ_1 and σ_3 can be expressed in terms of the Cartesian stress as follows:

$$\begin{aligned} \sigma_1 &= \frac{\sigma_x + \sigma_y}{2} + \sqrt{\frac{\left(\sigma_x - \sigma_y\right)^2}{4} + \tau_{xy}^2} \\ \sigma_3 &= \frac{\sigma_x + \sigma_y}{2} - \sqrt{\frac{\left(\sigma_x - \sigma_y\right)^2}{4} + \tau_{xy}^2} \end{aligned} \tag{7.22}$$

The yield function can be expressed in terms of the stress vector $\boldsymbol{\sigma} = \left\{\sigma_x, \sigma_y, \tau_{xy}\right\}$. Once a parameter set $\{a, b, c\}$ is available, the gradient and Hessian of each yield criterion can be calculated.

In plane strain analysis, the gradients and Hessian of the yield function are given by:

$$\text{grad} := \begin{pmatrix} \dfrac{a_3}{a_2} - \dfrac{ac}{b} \cdot a_1^{c-1} \\ -\dfrac{a_3}{a_2} - \dfrac{ac}{b} a_1^{c-1} \\ \dfrac{4\tau_{xy}}{a_2} \end{pmatrix} \tag{7.23}$$

$$\text{Hessian} := \frac{1}{4}\begin{pmatrix} -\frac{2a_3^2}{a_2^3}-\frac{ac^2}{b^2}a_1^{c-2}(c-1)+\frac{2}{a_2} & \frac{2a_3^2}{a_2^3}-\frac{ac^2}{b^2}a_1^{c-2}(c-1)-\frac{2}{a_2} & -\frac{8\tau_{xy}a_3}{a_2^3} \\ & -\frac{2a_3^2}{a_2^3}-\frac{ac^2}{b^2}a_1^{c-2}(c-1)+\frac{2}{a_2} & \frac{8\tau_{xy}a_3}{a_2^3} \\ \text{Symmetric} & & \frac{8a_3^2}{a_2^3} \end{pmatrix} \quad (7.24)$$

where $a_1 = 1+\frac{p}{b}, a_2 = \sqrt{\left(\sigma_x-\sigma_y\right)^2+4\tau_{xy}}$ and $a_3 = \sigma_x - \sigma_y$; grad =gradient of the yield function with respect to the Cartesian stresses; Hessian =Hessian matrix of the yield function with respect to the Cartesian stresses.

In practical applications, the parameters in (7.21) can be obtained directly from the triaxial experimental data regression. Therefore, there is no need to establish a yield criterion in the Mohr plane as given in (7.21) or apply the transformation during analysis.

However, this transformation remains of interest for many cases. Firstly, for applications with the limit equilibrium method or the multi-block method of the upper bound limit analysis, yield criteria in terms of shear and normal stress would be more convenient. Secondly, yield envelopes expressed in the Mohr plane are usually equipped with parameters with physical explanation. For example, in the linear Mohr-Coulomb yield criterion, the two regression constants are interpreted as the “cohesion” and “friction angle” which are superior to the “material constants” from the regression analysis. The transformation proposed in this work renders it possible for comparison of results from different methods.

The fitting technique introduced in this section is simple in concept and can be easily implemented in numerical applications for other type of yield criteria expressed in the

Mohr plane. To illustrate this point, let us examine another nonlinear yield envelope proposed by Maksimovic (1996), which is slightly more sophisticated than the Mohr-Coulomb model. A micro-mechanical model for the soil failure was proposed and the failure criterion is expressed in eq.(7.25).

$$\tau = c' + \sigma_n \tan\left(\phi_B' + \frac{\Delta\phi'}{1 + \frac{\sigma_n}{P_n}}\right) \tag{7.25}$$

Where $\phi_B{}'$ denotes the basic angle of friction

$\Delta\phi$ = the maximum angle of difference

P_n = the median angle of normal stresses

c' = cohesion such that

$c' = 0$ for noncemented soils and rock discontinuities

$c' > 0$ for cemented soils and rocks

The advantage of this model is that all the parameters in this model are associated with physical meanings. Such a yield criterion can be conveniently incorporated in the current formulation with the proposed technique in this section.

7.7 Summary

Additional constraints besides fundamental kinematic or static admissibility conditions are briefly discussed in terms of the implementation in the FELA. Some aspects of the specific features of limit analysis encountered in geotechnical engineering have been considered in the formulation of the FELA. Incorporating the pore water pressure and seismic effects can be considered as the body force in a relatively straightforward fashion.

Interface elements are introduced to model various degrees of roughness and inclined loading. The extension of the MC model to more sophisticated yield envelope is also briefly discussed, e.g., the Hoek-Brown yield criterion proposed for the heavily jointed rock masses and nonlinear yield envelopes expressed on the Mohr plane.

CHAPTER 8: NUMERICAL ILLUSTRATION OF LIMIT ANALYSIS

8.1 Introduction

In this chapter, a number of typical stability problems in geotechnical engineering are solved by FELA in conjunction with mesh adaptation. As the lower bound and upper bound theorems are dual to each other and each one could be transformed into similar form of the other, only the results for the lower bound analyses will be presented. A lower bound FELA using a nonlinear yield criterion of the power type is applied to the slope stability analysis and earth pressure problems, and the results obtained are compared with the various upper bound solutions in the literature.

8.2 Bearing capacity of shallow foundation

8.2.1 Introduction

The widely accepted procedure for estimating the bearing capacity of shallow foundation is the use of the simplified equation (8.1) that assumes simple superposition of the effects from cohesion, surcharge loading and the self-weight.

$$q_u = cN_c + qN_q + \frac{1}{2}\gamma B N_\gamma \tag{8.1}$$

Though (8.1) is simply a result of direct superposition, it has attracted interests of Terzaghi (1943), Sokolovskii (1965), Ukritchon et al. (2003a), Cheng (2004) and many others because of its simplicity and wide acceptance among the engineers. In the literature, coefficients N_c, N_q, N_γ have been determined by various methods including slip line method, limit equilibrium method, and limit analysis. Early contributions to this problem are due to Prandtl (1920) and Reissner (1924) with slip line methods for weightless soils.

For purely cohesive, weightless soils without surcharge, the collapse pressure of rigid strip footing can be expressed as eq. (8.2).

$$q_u = cN_c \tag{8.2}$$

Let the width of footing be B, $(c \geq 0, \phi \geq 0, \gamma = 0)$, concentric loading Q is given by

$$\frac{Q}{B} = q_u \tag{8.3}$$

where Q = the collapse load that will be optimized in the limit analysis. In this numerical example, N_c can be calculated by setting $B = 1.0,\ c' = 1.0,\ \gamma = 0.0$ as the input. The bearing capacity factors N_c has been solved analytically and the well-known closed form expression of N_c is given by eq. (8.4) (Prandtl 1920),

$$N_c = \left[e^{\pi \tan\phi} \tan^2\left(\frac{\pi}{4} + \frac{\pi}{2}\right) - 1\right] \cdot \cot\phi \tag{8.4}$$

(8.4) serves as an ideal tool for benchmarking the performance of our adaptation procedure based on unstructured meshes.

Due to the presence of the singularity at the edge of the footing, the use of fan elements significantly improves the results (Chen 1975; Lyamin et al. 2005; Makrodimopoulos and Martin 2006). However, it will be demonstrated that the procedure as suggested in this study can converge with good accuracy to the rigorous solution even with a poor initial trial mesh. Another important implication is that engineers do not need to spend a lot of time in preparing a good initial mesh for the solution, so that the proposed procedure can be a useful and practical tool to the engineers.

Considering the symmetry of the problem, only half of the problem is analysed. Extension elements are built to extend the stress field to semi-infinite space. Automatic mesh refinement stops if the number of active elements in the triangulation exceeds the specified maximum or no obvious improvement to the bounds are observed. Starting from an arbitrary initial solution (which can be far from the exact solution) is a good method to test for the applicability of the proposed adaptive mesh strategy. A simple initial mesh without taking the advantage of fan elements discretisation is hence used as the input (Figure 8.1). The SOCP formulation will be used in this example.

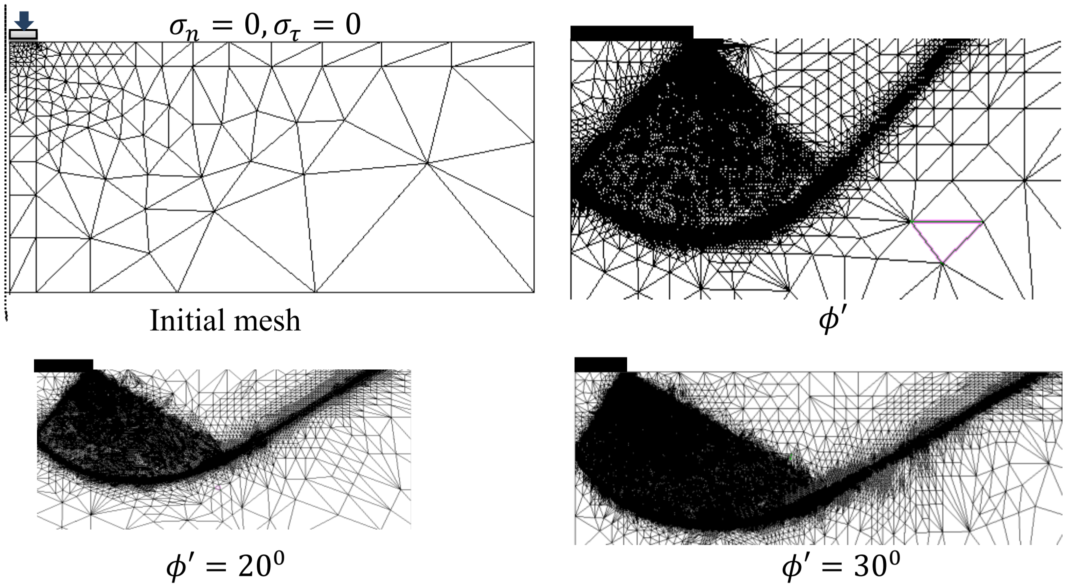


Figure 8.1 Failure mechanisms with different friction angles

It is evident from Table 8.1 that the solution of the collapse load is improved satisfactorily with few steps of iterations in less than a minute, which confirms the efficiency in formulating the Mohr-Coulomb problem as the SOCP. The adaptive refinement procedure in this example gives the best lower bound solution of N_c 5.137 with an error 0.065% calculated from 58349 elements. It has been shown by (Munoz et al. 2009) that N_c converges to the exact value 5.14 satisfactorily by using automatic fan zone generation in their work, but as the eventual mesh and the detailed mesh results are not given, it is not possible to compare these two results. The slip bands at failure captured by the refinement indicators in this numerical example is completely satisfactory as compared with the Prandtl mechanism (Figure 8.2), which is composed of a rigid triangular wedge *ABC* moving downward, a logspiral shear zone, *BCF* with a central angle $\pi/2$ and a rigid wedge *BFG* with base angles $\pi/4 - \phi/2$, as shown in Figure 8.2.

Table 8.1 Results for rough footing with adaptive mesh adaptation

	$\phi' = 0^0$			$\phi' = 30^0$		
Refinement	Elements	N_c error (%)	CPU(s) (iters)	Elements	N_c error(%)	CPU(s) (iters)
0	268	4.94534 (3.787)	0.19 (24)	268	26.3973 (12.418)	0.16 (19)
1	519	5.03321 (1.948)	0.45 (23)	929	28.4838 (5.495)	0.567 (22)
2	1178	5.08078 (0.960)	1.03 (23)	3209	29.3489 (2.625)	2.71 (26)
3	3103	5.10472 (0.461)	3.65 (23)	10257	29.7707 (1.225)	13.95 (33)
4	5543	5.12057 (0.21)	6.49 (24)	29554	29.9667 (0.574)	62.90 (42)
5	11359	5.12983 (0.101)	15.35 (25)	67991	30.0549 (0.282)	167.48 (40)
6	26956	5.13502 (0.0969)	42.53 (22)	92431	30.081 (0.196)	221.99 (38)
7	58349	5.13704 (0.065)	111.24 (22)	244333*	30.120 (0.066)	760.00 (44)

* Not obtained at 7-th iteration, but at 9th iteration.

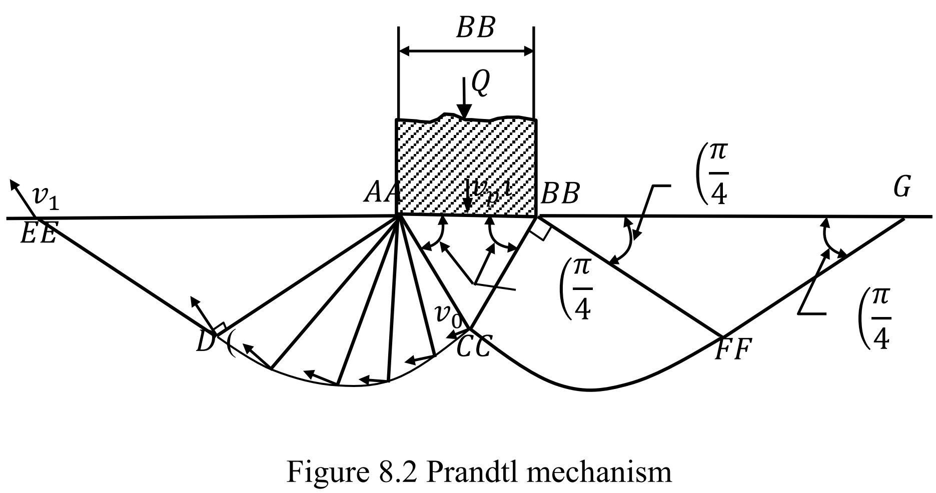


Figure 8.2 Prandtl mechanism

8.2.2 Computation cost in the numerical examples.

In this example, the problem is solved with a Windows7 64 bit Intel Core 2 Quad CPU Q9550 2.83GHz; RAM 8GB desktop computer with 64bit optimization package MOSEK. Generally, the optimization problem arising from the FELA can be solved efficiently with the primal-dual interior point algorithm. In the bearing capacity example for shallow foundation, for the case $\phi' = 0$, MOSEK failed to determine the optimality, and a near-optimal solution was given for models with elements over 60000. Otherwise, it could be expected that the error will be further reduced with more refinements. An optimal solution can be obtained with 92431 elements for the case when $\phi' = 30^0$. Detailed time requirement including the assembly of the problem, file writing, refinement and error estimate for the first example are tabulated in Table 8.2, and the other two examples are solved in similar time scale depending mostly on the number of active elements in the model.

Table 8.2 Time spent on the bearing capacity example.

Example	Itera -tion	Total time (s)	Solution (%)	Assembly (%)	Error estimate (%)	Final elements
Bearing capacity ($\phi' = 0$)	8	601	59.65	12.73	0.46	78922
Bearing capacity ($\phi' = 30^0$)	8	857	62.2	15.38	0.47	92431

It can be observed from Table 8.2 that the examples with adaptive mesh refinement are solved in fairly short time with the MOSEK. Total time required is approximately 10 minutes for 8 iterations in both examples, reducing the error to less than 0.5%. It is evident that the mesh adaptation strategy proposed in the present study is both practical and efficient for the FELA.

8.2.3 Calculation of N_γ with the adaptive procedure

Unlike the terms N_c and N_q, analytical solutions for N_γ cannot be determined explicitly. Various numerical methods have been adopted to obtain the value of N_γ: (1) limit equilibrium method including Meyerhof (1963), Vesic (1973), and others; (2) the multi-upper bound analysis Michalowski (1997), Soubra (1999), Wang et al. (2001) and others; (3) slip line method including Sokolovskii (1965), Booker (1969), and others; (4) full numerical methods including Griffiths (1982), Frydman and Burd (1997), Loukidis and Salgado (2009) and others. In contrast to the N_c and N_q, a relatively large difference exists between the published numerical solutions for N_γ in the literature, particularly when the friction angle is larger than 30^0.

As the lower bound and upper bound methods are rigorous under the framework of theory of plasticity, the two bound theorems provide excellent tools for benchmarking the existing numerical solutions. Results of N_γ based on the lower and upper bound FELA have been reported by Sloan (1988), Sloan and Kleeman (1995) respectively using the FELA in conjunction with the linear formulation. Following these works, Bandini (2003) and Ukritchon et al. (2003b) performed more refined analysis on the benchmarking of the existing numerical solutions. In their work, the numerical results for N_γ found from various methods have been compared and commented. More recently, using the nonlinear formulation of the lower and upper bound analysis respectively developed by Lyamin and Sloan (2002a) and Lyamin and Sloan (2002b), Hjiaj et al. (2005) reported tighter bounds on the exact value of N_γ for a range of friction angles. They bracketed the value of N_γ

with an error at most 3.42% with specially prepared mesh input that takes the advantages of the fan elements and the knowledge from the slip line theory.

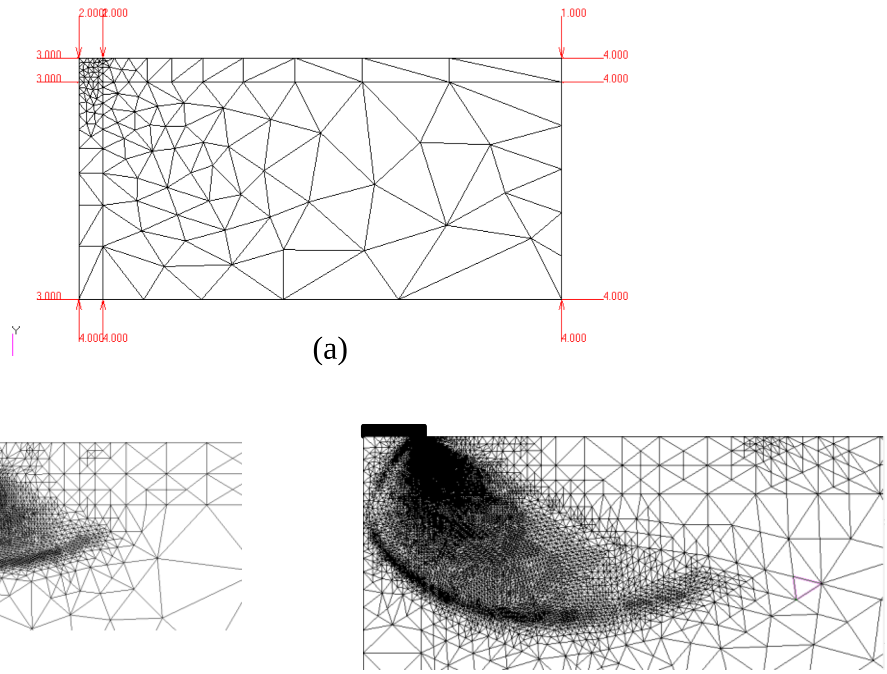
In this example, it is intended to investigate the performance of the current mesh adaptation procedure on this typical difficult problem for the calculation of N_γ. Analyses with $\phi' = 35^0$ and $\phi' = 45^0$ respectively will be carried out. No special prior manipulation of the initial mesh has been adopted, and consequently a relatively coarse unstructured mesh is fed as the input data. The mean of the bounds with errors less than 5% for N_γ produced by Hjiaj et al. (2005) will be used as a reference in this numerical example.

Details of the adapted meshes are given in Figure 8.3 and the stress field obtained at the instant of collapse is shown in Figure 8.4. It is shown that the adaptive refinements appreciably improve N_γ, error being reduced from 60% for the initial mesh to less than 5% in 8 refinements (see Table 8.3). The results are comparable and even better than those by Hjiaj et al. (2005) for which the mesh is specially designed with fan elements.

Table 8.3 N_γ for rough footing with mesh adaptation

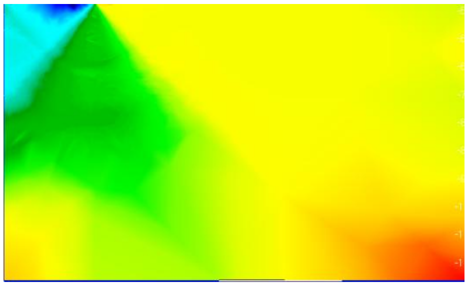
	$\phi' = 35^0$			$\phi' = 45^0$		
Refinement	Elements	N_γ error	CPU(s) (iters)	Elements	N_γ error (%)	CPU(s) (iters)
0	244	14.89 (57%)	0.13 (24)	244	88.37 (62%)	0.17 (26)
2	671	27.00 (22%)	0.39 (23)	883	174.145 (25%)	0.59 (28)
4	2350	31.29 (10%)	1.76 (27)	3067	208.281 (10%)	2.98 (34)

6	7227	33.01 (5%)	8.92 (31)	9507	222.111 (4%)	15.35 (45)
8	19083	33.77 (3%)	45.16 (50)	20541	227.365 (2%)	41.22 (46)

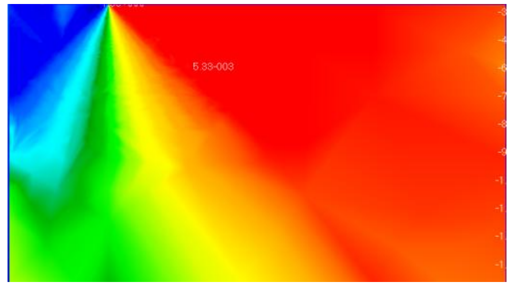


(a) (b) (c)

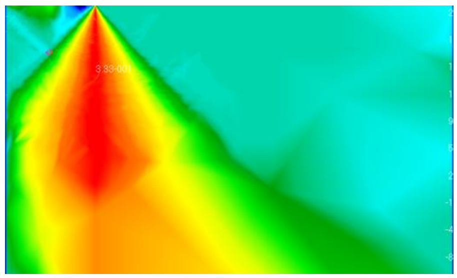
Figure 8.3 Mesh configuration with different friction angles (a) initial mesh, (b) updated mesh for $\phi' = 35^0$ and (c) adapted mesh for $\phi' = 45^0$



(a)



(b)



(c)

Figure 8.4 Stress distribution at the collapse,(a) σ_x, (b) σ_y, and (c) τ_{xy}

The example was solved on 64bit Window 7 with Intel Core 2 Quad CPU Q9550 2.83GHz desktop computer with optimization package 64bit MOSEK. Generally, the optimization problem arising from limit analysis can be solved efficiently with primal-dual interior point algorithm.

Table 8.4 Time spent on the bearing capacity example.

Example	Itera tion	Total time (s)	Solution (%)	Assembly (%)	Error estimate (%)	Final elements
$N_\gamma(\phi' = 35^0)$	8	180	56.86	17.08	2.96	19083
$N_\gamma(\phi' = 45^0)$	8	190	62.20	15.38	2.00	20541

It can be evident from Table 8.4 that the examples with adaptive mesh refinement are solved within fairly short time (acceptable for engineering purpose) with MOSEK. The total time required for the analysis is around 3 minutes for 8 iterations in both cases. It is evident that the mesh adaptation strategy as proposed in the present study is both practical and efficient for the finite element based limit analysis.

8.2.4 Discussion and conclusion

The bearing capacity problem with the MC yield criterion has been investigated with the SOCP formulation. The time of solution for the mesh adaptation procedures suggest that a model with several refinements can be solved in minutes for two-dimensional problem, rendering the adaptive procedure both practical and efficient. The unstructured triangular mesh in conjunction with the localized mesh refinement strategy performed well in the calculation of the dimensionless coefficients for the bearing capacity problem even with the presence of the stress singularities. The slip bands captured by the mesh adaptation procedure could provide a good mesh that accounts for the plastically deformed zone in the solution domain and meanwhile provide a useful knowledge about the ultimate limit state of a general complicated system without prior assumption, which is the advantage of the present study.

8.3 Slope stability with a nonlinear yield criterion

8.3.1 Introduction

Slope stability problem is one of the oldest stability problems in geotechnical engineering. The majority of the slope stability analyses are still performed with LEM. The application of limit analysis in slope stability problem is mainly restricted to the upper bound analysis, and the pioneer works regarding applying the conventional upper bound analysis has been conducted by many researchers, for example, Chen (Chen 1975; Chen and Liu 1990) by assuming different failure mechanisms consisting of rigid blocks and velocity discontinuities. More general limit analysis studies on slopes have been carried out by Kim (Kim 1998; Kim et al. 2002) with the finite element based lower and upper bound analysis under plane strain condition, Chen (2003) and more recently Li et al. (Li et al. 2009a; Li et al. 2008; Li et al. 2009b; Li et al. 2011) used the rigid finite element method on 3D slopes and rock slopes. Most of these researches in this area focus on the ultimate limit state with the linear Mohr-Coulomb material except for works by Li et al. (2011) who addressed slopes in heavily jointed rock mass obeying Hoek Brown yield criterion.

It is well known that for most of the soil, the curvature of the yield surface is more obvious under low confining stress condition. For simplicity, the linear approximation of the failure envelope has been widely accepted over a century in which the shear strength of a material is described by two parameters, namely the cohesion c' and the friction angle ϕ'. Due to the nonlinearity of the failure envelope, the two parameters in the MC yield function are actually dependent on the stress level and are not really constants over a wider stress range. A number of nonlinear failure criteria, mostly in the form of power (Charles 1982; Charles and Watts 1980; Zhang and Chen 1987) and logarithmic expressions have been reported in the literature for geotechnical materials and have been applied with certain degrees of success to some geotechnical problems (such as slope stability analysis and the determination of the lateral earth pressure). The importance of the in-situ stress on slope stability analysis is also discussed by Griffith (in another aspect) who argued that for the soil near to the ground surface, dilative behaviour is more obvious while for deeper soil, the dilative behaviour will be less because of higher confining stress.

The choice of the dilation angle in strength reduction slope stability analysis should consider this phenomenon but is actually helpless in this respect.

Attempts in applying nonlinear yield criteria in an upper bound approach are believed to be first made by Baker and Frydman (1983) who studied the bearing capacity of a slope foundation with variational calculus. Similar variational approach was later proposed by Zhang and Chen (1987) for upper bound slope stability analysis. The resulting equation system was then solved by an inverse solution procedure. However, the applicability of the variational methods is restricted to simple problems with homogeneous soil profile due to the complicated algebraic derivation. In addition, even for simple situations, numerical procedures are required to evaluate the optimal solution. For more complicated analyses, as noted by Cheng et al. (2011b), it might be advisable that variational approach be replaced by heuristic optimization approaches.

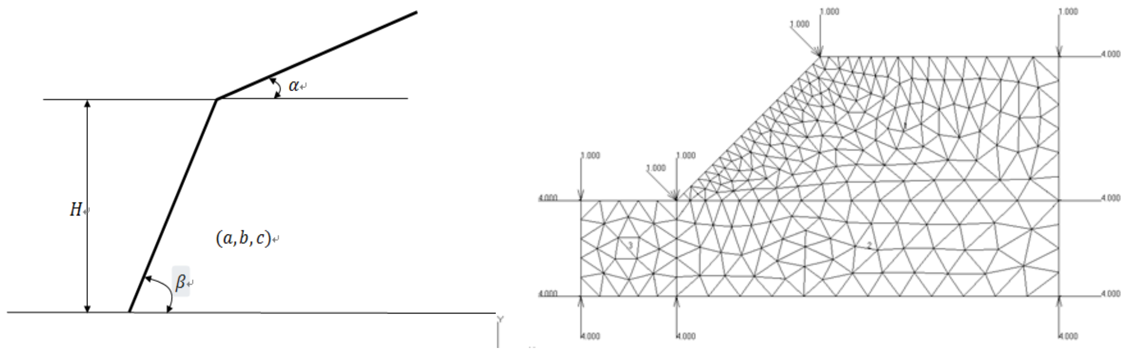
To simplify the procedure, Collins et al. (1988), Drescher and Christopoulos (1988), Yang et al. (Yang 2007; Yang and Yin 2004; Yang and Yin 2005) adopted simplified methods in which the nonlinear yield envelopes are replaced by a series of linear tangential lines or one linear tangential failure criterion at a point M(Figure 7.6).The upper bound solution is then sought by minimising the objective load with respect to the location of M. These “tangent line” methods are shown to yield similar results to those by variational calculus for some specific cases. However, the convergence to the exact load is not possible in nature by this technique, and the difference with the original failure envelope may become large for materials following a highly nonlinear failure criterion, as the collapse load is in fact obtained from a linear tangential yield criterion corresponding to an optimal position of point M. A finite element analysis for slope stability problem with similar power type yield criterion has been performed by Li (2007) who investigated the same problem with the shear strength reduction method. From his results, solutions obtained are shown to be either greater or less than the upper bound solution. It is clear that there is no way to evaluate the quality of the solution from the finite element analysis, and some judgments are required to identify the failure of the system, which may be difficult for the normal engineers to carry out routine design works.

Lower bound solution is attractive in providing safer design. It is however traditionally much more difficult to construct a statically admissible stress field than a kinematically admissible velocity field manually, and there are very few studies on the lower bound analysis using nonlinear yield envelopes.

The FELA eliminates the difficulties in constructing a statically admissible stress field in a systematic manner, particularly when an adaptive procedure is employed. The failure mechanism will form part of the solution. Combining with the state-of-art solution algorithm for mathematical nonlinear programming, limit analysis with nonlinear yield criterion can be adopted to solve more general problems efficiently.

8.3.2 Calculation of the stability number with a power-type yield envelope

The discussion of the MC material will be extended to a power-type yield criterion as discussed in chapter 7 (see page 183). Figure 8.5 presents a simple homogeneous slope with a height $H = 6m$ and slope angle β. The soil mass is assumed to obey a power-type yield criterion in the form of $\tau = c_0\left(1 + \frac{\sigma_n}{\sigma_t}\right)^{\frac{1}{m}}$. Figure 8.5 (b) shows the mesh used for in this analysis. Boundary conditions of the slope consist of two parts: (1) free boundary condition (denoted by ID 1) representing boundary segment free from loading and (2) extension boundary conditions (denoted by ID 4) accounting for the stress field in the semi-infinite space.



(a) (b) 1 for free boundary, 4 for extension boundary condition

Figure 8.5 (a) Diagram of the slope profile and (b) mesh and boundary condition of the numerical model.

Following the example of Zhang and Chen (1987) in which $c_0 = 90\text{kPa}, \sigma_t = 247.3kPa$, and m takes the value from 1.2, 1.4,1.6,1.8, 2.0, 2.5. The corresponding parameters of the yield criterion in (7.19) curve fitted from (7.18) are given in Table 8.5. A typical fitted curve of (7.19) and the original curve plotted from (7.18) are given in Figure 8.6. It is clear that these two curves overlap perfectly with each other

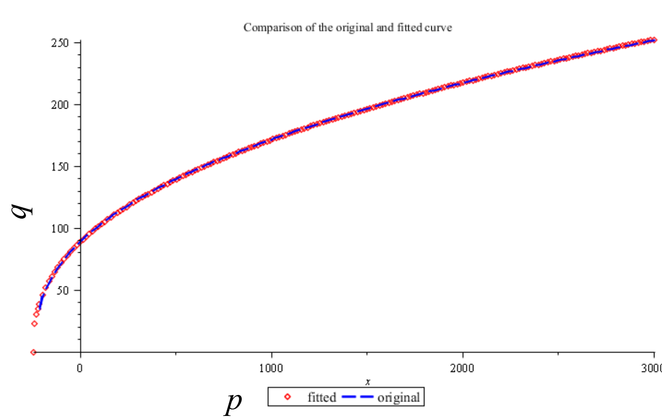


Figure 8.6 Original yield envelop and the fitted yield envelope

Table 8.5 Corresponding material constants transformed from (7.20) with $\mathrm{c_0} =$ $90\mathrm{kPa}, \sigma_\mathrm{t} = 247.3$

m	1.2	1.4	1.6	1.8	2.0	2.5
a	86.067	87.030	87.690	88.157	88.497	89.031
c	0.845	0.727	0.637	0.566	0.509	0.407

For slopes of soil mass obeying a power-type yield criterion, the stability number is defined as

$$N_s = \frac{\gamma H}{c_0} \tag{8.5}$$

N_s is a dimensionless number depending on the geometry of the slope and the properties of the material constituting the slope. c_0 is the initial cohesion, more precisely the interception of the nonlinear yield envelope with the τ-axis.

Let us first consider a limiting case in which $m = 1$. The equivalent material constants for the Mohr-Coulomb yield criterion can be easily obtained as $c' = 90kPa,$ and $\phi' = 20^0$. In the literature, the lower bound FELA has been formulated as a general nonlinear programming by Lyamin and Sloan (2002a) with a hyperbolic smoothing technique and more recently as an SOCP by Makrodimopoulos and Martin (2006). Both of these formulations have been implemented and will be used to verify the NLP formulation using (7.23) and (7.24). The stability numbers calculated for a slope with $\beta = 90^0$ using various methods are given in Table 8.6. It can be observed that the current formulation yields identical stability number to those by the SOCP formulation and the NLP formulation with hyperbolically-smoothed MC yield criterion.

Table 8.6 Stability numbers for linear MC material calculated from different formulations with a model of 738 3-noded triangular elements.

SOCP formulation(2006)	Lyamin's formulation(2002a)	Current formulation (Lower Bound)	Upper bound Result(Chen 1975)
5.35	5.35	5.35	5.51

Table 8.7 gives the results of N_s with different method as coefficient m varies from 1.2 to 2.5. Solutions obtained with the current lower bound formulation are expectedly less than the existing upper bound solutions. It is observed that the gap between the upper bound solutions (Drescher and Christopoulos 1988; Yang and Yin 2004; Zhang and Chen 1987) and the current lower bound solutions are bounded between 2%~11%, and the difference increase with the decrease of the slope angle β.

Table 8.7 N_s with different method as coefficient m from 1.2 to 2.5

m	β	Drescher and Christopoulos (1988)	Zhang and Chen (1987)	Yang and Yin (2004)	Present lower bound solution	Difference (%)
1.2	90	5.15	5.13	5.15	5.00	2.53
	75	6.79	6.77	6.77	6.31	6.79
	60	8.99	8.95	8.95	8.20	8.38
	45	12.60	12.55	12.55	11.06	11.87
1.4	90	4.92	4.89	4.89	4.76	2.66
	75	6.36	6.33	6.33	5.92	6.48
	60	8.18	8.13	8.18	7.48	8.00
	45	10.82	10.82	10.87	10.07	6.93
1.6	90	4.76	4.73	4.76	4.59	2.96
	75	6.07	6.04	6.07	5.66	6.29
	60	7.65	7.61	7.65	7.01	7.88
	45	9.85	9.70	9.84	9.11	6.08
1.8	90	4.64	4.60	4.64	4.47	2.83
	75	5.86	5.82	5.86	5.47	6.01
	60	7.29	7.24	7.29	6.69	7.60

	45	9.26	9.10	9.17	8.48	6.81
2.0	90	4.55	4.52	4.54	4.38	3.10
	75	5.70	5.70	5.70	5.31	6.84
	60	7.02	7.02	7.02	6.41	8.69
	45	8.82	8.78	8.69	8.02	8.66
2.5	90	4.39	4.35	4.38	4.22	2.99
	75	5.43	5.40	5.43	5.09	5.74
	60	6.59	6.54	6.59	6.08	7.03
	45	7.93	7.95	7.94	7.31	8.05

8.3.2.1 **The influence of the parameter m**

Figure 8.7 illustrates the effects of m on the stability number N_s for the problem $\beta = 90^0$. The stability number N_s gradually decreases as m increases, and the lower and upper bound solutions follow almost the identical trends.

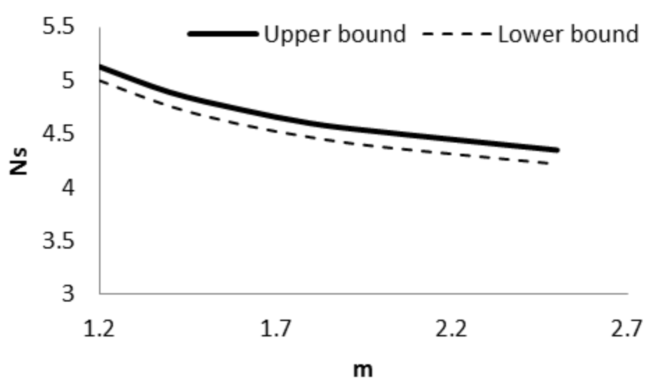


Figure 8.7 Effect of m on the stability number N_s corresponding to $\alpha = 0^0; \beta = 90^0$

Adaptive refinement procedure is adopted in the present study to capture the slip bands as well as to obtain improved lower bounds. From Figure 8.8, it is observed that the assumption that the slip surface pass through the toe of slope in Chen and Liu (1990) and Yang and Yin (2004) for dry and homogeneous soil slope is reasonable and well justified by the failure mechanism obtained with the FELA in this research. This observation is also in consistence with the results for linear MC material. It can be observed in Figure 8.8 that a lower stability number is characterised by a deeper failure surface and larger magnitude of m, implying that keeping the rest of material parameters constant, a larger m corresponds to a weaker material.

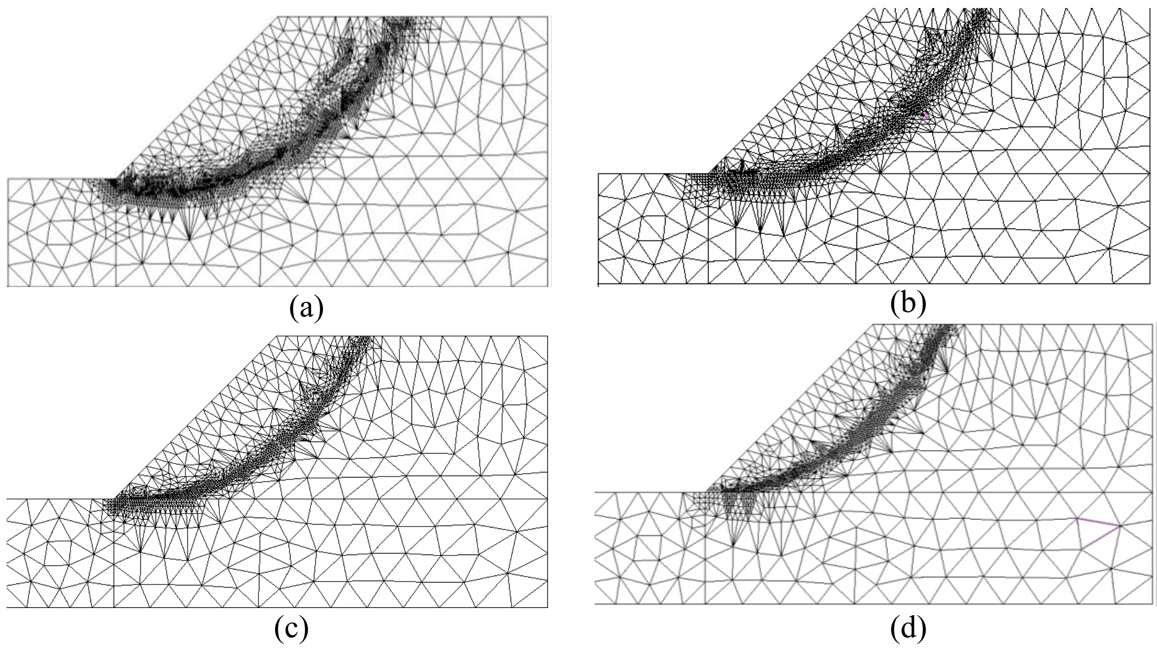
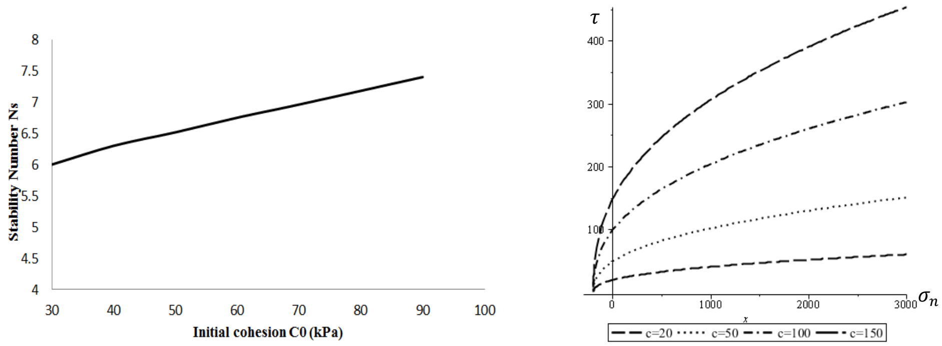


Figure 8.8 Slip surfaces for different m values with $\alpha = 0, \beta = 45^0$, $c_0 = 90kPa, \sigma_t = 247.3kPa$ and (a) $m = 2.5$, (b) $m = 1.8$ (c) $m = 1.4$ (d) $m = 1.2$

8.3.2.2 **Effect of the initial cohesion c_0**

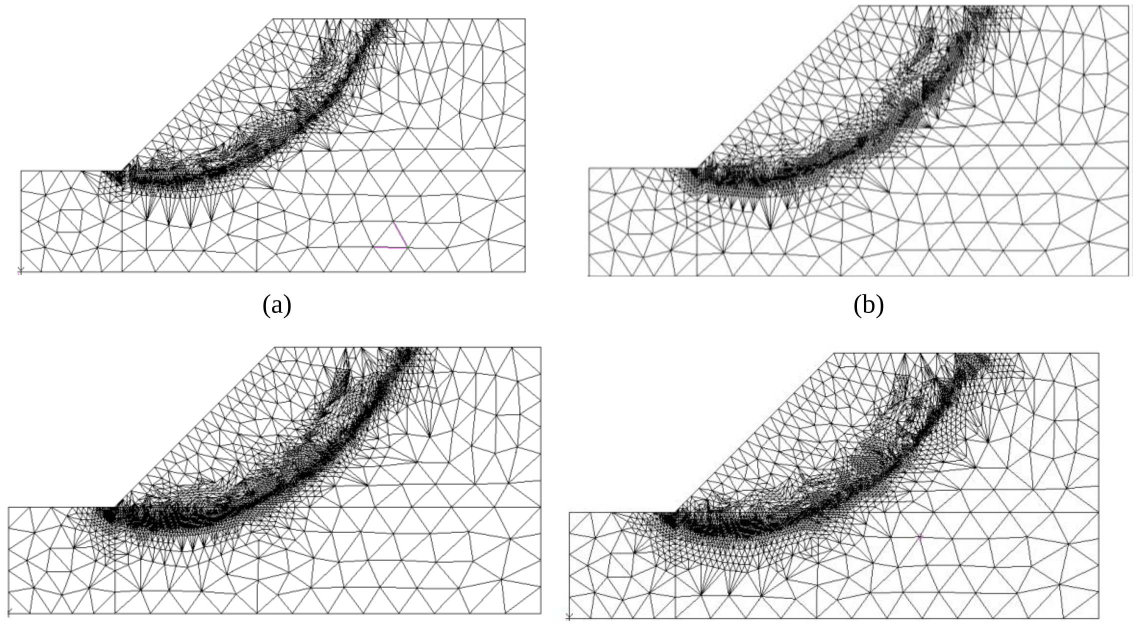
Figure 8.9 shows the effects of c_0 on the stability number. It is clear that the stability number is approximately linearly dependent on c_0, which is not consistent with the observation for MC material. It is well known that for the linear Mohr-Coulomb material, the stability number N_s is independent of the cohesion term c, but this relation does not hold for the case of nonlinear yield criterion with the stability number as defined in (8.5). To explain this, let us consider a series of nonlinear yield criteria with various c_0 as shown in Figure 8.9(b). It is observed that an increase in c_0 implies an increase in ϕ' of its corresponding linear counterpart because for the nonlinear envelope with larger c_0, the curves are generally steeper. Consequently, the increase in the stability number is within the expectation.



(a) (b)

Figure 8.9 Effects of the interception c_0 on the stability factor $\alpha = 0^0, \beta = 45^0, m = 2.5, and\ \sigma_t = 247.3kPa$ (a) N_s vs. c_0, (b) yield envelope in Mohr plane with varying

From Figure 8.10, the slip bands captured for various c_0 suggest that the failure mechanism for slope depends on c_0 as well, which differs from the conclusion for the linear Mohr-Coulomb yield criterion for which the slip bands are almost exclusively controlled by the friction angle. It is also noted that for smaller c_0, the failure surface tends to be more deep-seated.



(a) (b)

(b) (d)

Figure 8.10 Slip surfaces for different m values with $\alpha = 0, \beta = 45^0$, $\sigma_t = 247.3kPa$

and (a) $c_0 = 150kPa$, (b) $c_0 = 90kPa$(c) $c_0 = 70kPa$ (d) $c_0 = 50kPa$

8.3.2.3 **Effect of the tensile strength σ_t**

Figure 8.11 shows the effects of the tensile strength σ_t on the stability number. Keeping the rest of the parameters constant, the increase in σ_t will give rise to an appreciable decrease in the stability number, particularly at the lower range of σ_t. As σ_t approaches zero, the optimization problem is getting more difficult to converge, and very large stability number is to be expected. It can be explained by the yield envelopes in Figure 8.11(b), in which the yield envelopes with varying σ_t are plotted. It can be seen that as σ_t approaches to the origin, the yield curves lying on the positive σ_n region will cover increasing area, and larger limit load is hence expected. This argument is well backed up by the failure surfaces shown in Figure 8.12. It is clear that a reduction in σ_t will strengthen the material and lead to a shallower slip surface.

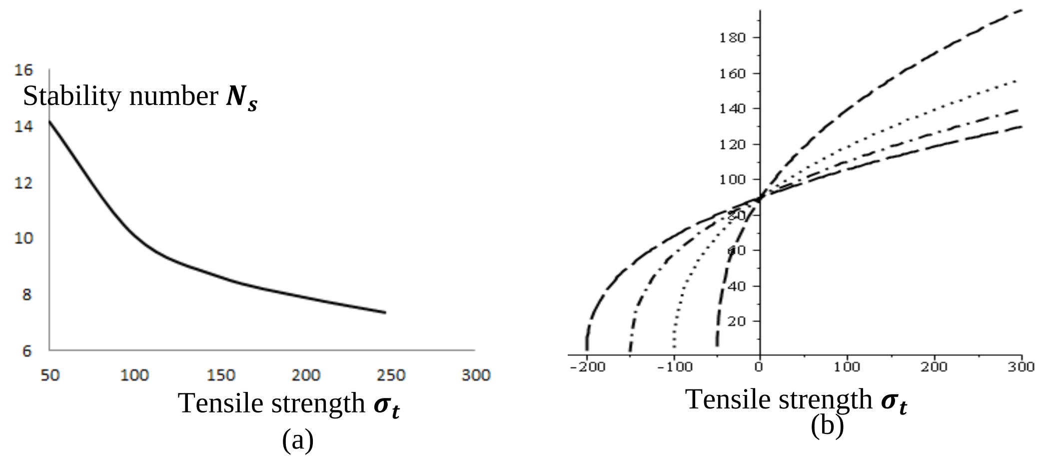


Figure 8.11 Effect of the σ_t ($\alpha = 0, \beta = 45^0, m = 2.5$, $c_0 = 90kPa$) (a) N_s vs. σ_t ;(b) yield envelope in Mohr plane with varying σ_t

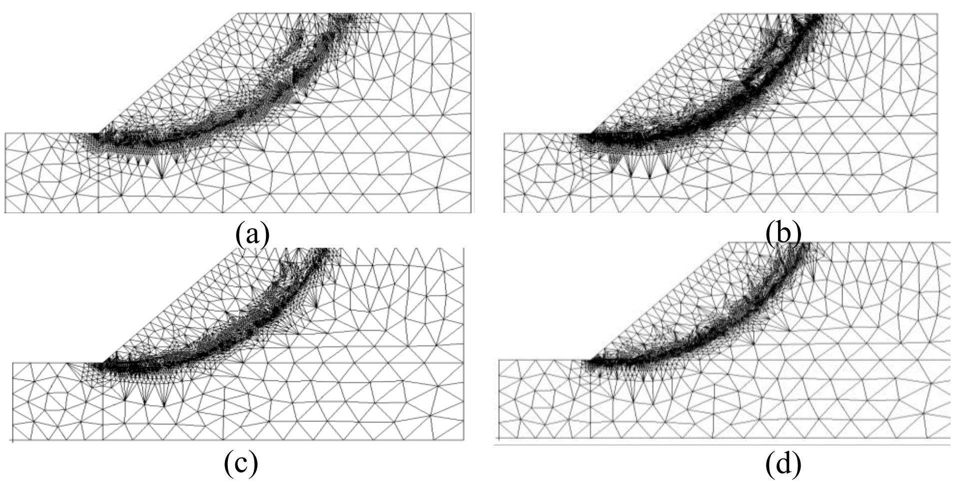


Figure 8.12 slip surface for different σ_t value $\alpha = 0, \beta = 45^0$, $c_0 = 90kPa, m = 2.5$ and (a) $\sigma_t = 247.3$kPa (b) $\sigma_t = 200$kPa (c) $\sigma_t = 150$kPa (d) $\sigma_t = 50$kPa

8.3.2.4 **The effect of slope angle α**

Table 8.8 shows the influences of the angle of the slope crest on the stability number. The upper bound solutions are extracted from the work by Yang and Yin (2004), in which the stability number is calculated with the a generalized tangent line technique. Stability numbers calculated with $\alpha = 5^0$, 10^0, and 15^0 for various slope angles are calculated with the FELA with the transformed yield criterion (7.21). It is found that the stability number decreases slightly with the increase of α, roughly 4% when α is increased from 5 to 15^0 for different β. Therefore, it could be concluded that the stability number N_s is insensitive to the variation of the crest slope angle α in the range of 5^0-15^0.

Table 8.8 Stability number N_s with varying α

m	α	β = 90		75		60		45	
		ub^*	lb	ub^*	lb	ub^*	lb	ub^*	lb
1.2	5	5.10	4.97	6.71	6.22	8.87	8.33	12.46	11.69
	10	5.04	4.92	6.61	6.16	8.73	8.26	12.25	11.40
	15	4.97	4.87	6.48	6.05	8.56	8.22	11.94	11.27
1.4	5	4.87	4.73	6.28	5.83	8.06	7.61	10.71	10.19
	10	4.81	4.64	6.17	5.72	7.91	7.45	10.49	10.03
	15	4.73	4.63	6.04	5.67	7.70	7.40	10.13	9.50
1.6	5	4.71	4.56	5.98	5.56	7.53	7.11	9.68	9.27
	10	4.64	4.44	5.88	5.48	7.37	6.97	9.44	8.83
	15	4.56	4.46	5.73	5.37	7.15	6.87	9.03	8.27
1.8	5	4.59	4.45	5.77	5.36	7.16	6.78	9.00	8.44
	10	4.52	4.33	5.66	5.29	7.00	6.61	8.75	8.11
	15	4.43	4.34	5.51	5.18	6.76	6.40	8.28	7.55
2	5	4.49	4.41	5.61	5.22	6.89	6.56	8.51	8.12
	10	4.42	4.33	5.50	5.14	6.72	6.40	8.25	7.64
	15	4.34	4.24	5.35	5.06	6.47	6.03	-	7.01
2.5	5	4.33	4.21	5.34	5.00	6.45	6.11	7.75	7.20
	10	4.26	4.13	5.22	4.90	6.27	5.89	7.47	6.85
	15	4.17	4.04	5.06	4.76	5.98	5.55	-	6.22

* Taken from Yang and Yin (2004)

8.4 Lateral earth pressure

8.4.1 Introduction

Determination of the lateral earth pressure is a classical problem that has been considered by Coulomb as early as in 1773 (Coulomb 1773) with the limit equilibrium method. Following similar principles, researches have later been carried out to obtain the earth pressure coefficients considering more factors of the earth wall system, among which are the friction between the wall and soil friction, seismic load effect, shape of the failure surface, etc. Methods available for the determination of the active and passive earth pressure fall into four categories: (1) the limit equilibrium method which is still the most popular one, for example (Morrison and Ebeling 1995); (2) slip line method (Cheng 2003) and others; (3) limit analysis (Chen 1975; Shiau et al. 2008); (4) more sophisticated numerical analysis with finite difference and finite element based limit analysis..

Depending on the role that soil play, the earth pressure comprises three types, at rest earth pressure and active/passive earth pressure. The soil state under which the at-rest pressure P_o is applied on the retaining structure is not an ultimate limit state problem and therefore cannot be considered by the limit analysis. If the wall is initially held by a force $P = P_0$, (see Figure 8.13) as the force P is reduced, the wall will be pushed outward due to the weight of the soil mass, and eventually when the pressure is decreased to a critical point $P = P_a$, the backfill will undergo unconfined plastic flow and the whole system fails. The critical force P_a is called the active earth force. On the other hand, if the force P is increased, the wall is pushed inward and a collapse load is obtained at the maximum load $P = P_p$, which is called the passive earth force.

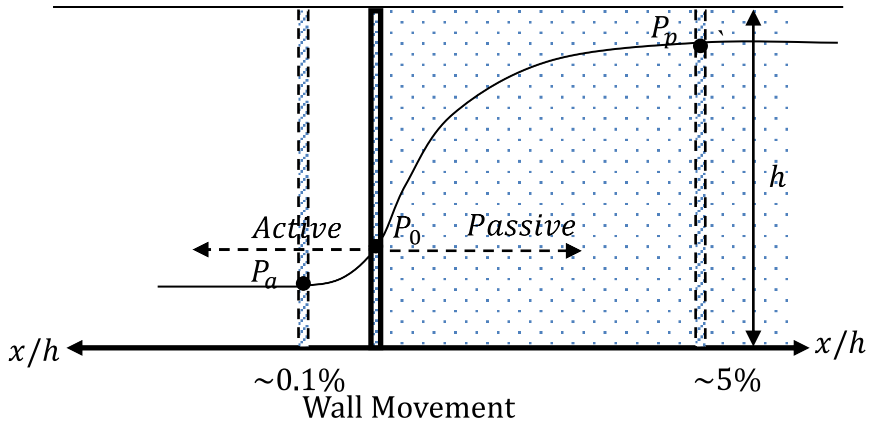


Figure 8.13 Load-displacement relationship for a retaining wall

In other words, the active earth thrust is the minimum force that is required for the backfill to maintain the stability and the passive earth thrust is the maximum force the backfill can hold without failure.

8.4.2 Passive earth pressure obeying MC yield criterion

This example illustrates the application of the adaptive mesh refinement procedure in finding the passive lateral earth thrust on the retaining wall. Figure 8.14 shows the model used in the analysis with back inclination β, wall inclination α and wall friction angle δ. Purely frictional soil is assumed, i.e. $c = 0$, and the passive earth pressure coefficient is defined as

$$K_P = \frac{P_p}{\frac{1}{2}\gamma H^2} \tag{8.6}$$

where P_p is the minimum force required to displace the soil mass inward. For the sake of simplicity, only the case $\beta = 0^0; \alpha = 90^0$ will be considered.

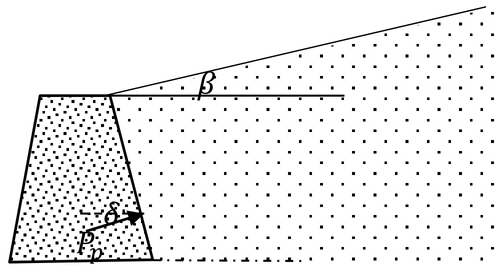


Figure 8.14 Passive lateral earth pressure model

The backfill behind the retaining wall is assumed to be purely frictional i.e., $c = 0$. Recall that $k = c\cos\phi'$, the MC yield criterion is reduced to:

$$R = \sqrt{s_{xx}^2 + s_{xy}^2} + a\sigma_m \tag{8.7}$$

Figure 8.15 shows the failure mechanisms of the passive earth pressure problem for various soil-wall friction angle. It has been noted that K_p can be obtained with fairly coarse mesh when the wall friction is not considered in the analysis, since the failure surface for this case is simply a plane (as shown in Figure 8.15).

When the wall friction is considered, the adaptive procedure does give noticeable improvement in the lower bound solution. From Table 8.9, it is observed that the solutions from lower bound analysis are in good agreement with those by the other methods for frictionless wall. When $\delta = \phi'/2$ is modeled in the analysis, the failure mechanism reflected by the eventual refined mesh is similar to the log-sandwich failure pattern (Chen and Liu 1990). To compare the plastic deformed region with the slip line field, program KP which is developed for the calculation of the passive earth pressure by Cheng (2003) has been used. As shown in Figure 8.16, the slip bands captured by the FELA combined with mesh adaptation are totally satisfactory.

It is also noted that the energy dissipation contributed from the sandwich part is much smaller for purely frictional soil; therefore, a larger refinement ratio is required to reproduce the slip surface in Figure 8.15.

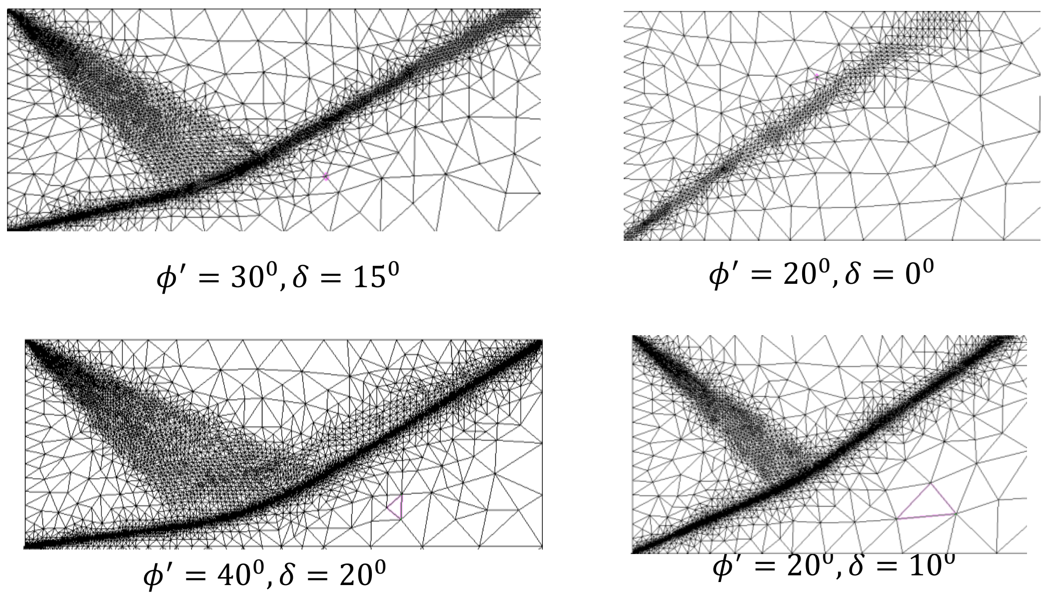


Figure 8.15 Final mesh for the passive lateral earth pressure analysis

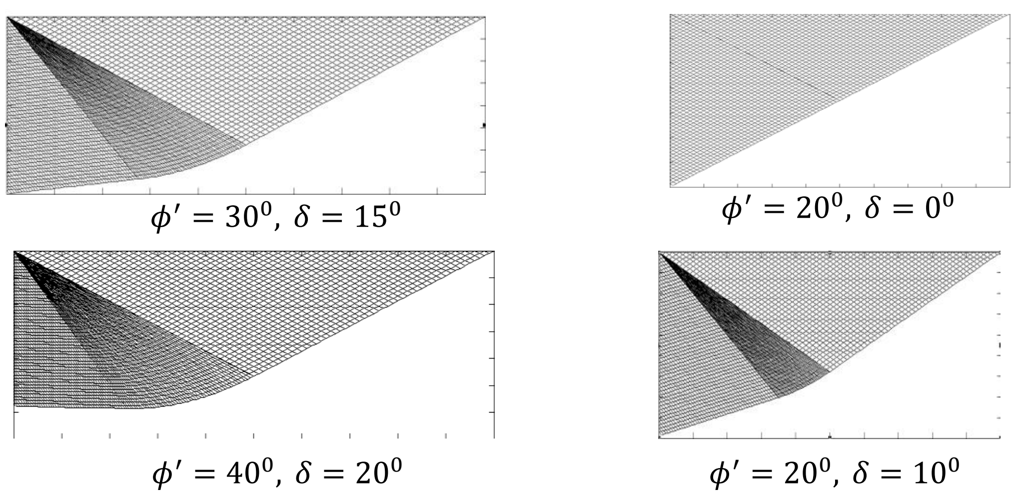


Figure 8.16 Slip network for passive earth pressure from slip line method

Table 8.9 K_p values by various methods ($\beta = 0$)

Methods	$\alpha = 90^0$					
	$\phi = 20^0$		$\phi = 30^0$		$\phi = 40^0$	
	$\delta = 0^0$	$\delta = 10^0$	$\delta = 0^0$	$\delta = 15^0$	$\delta = 0^0$	$\delta = 20^0$
Coulomb	2.04	2.64	3.00	4.98	4.60	11.08
Zero-extension	2.04	2.55	3.00	4.65	4.60	9.95
Slip-line	2.04	2.55	3.00	4.62	4.60	9.69
UB Methods*	2.04	2.58	3.00	4.70	4.60	10.07
Current lower bound value	2.04	2.57	3.00	4.65	4.60	9.80

* the upper bound solutions by Chen and Liu (1990)

8.4.3 Active earth pressure with a nonlinear yield criterion

At present, most of the lateral earth pressure solutions are developed for MC material. However, it has been well-known that the MC criterion is merely a simplified straight-line approximation of the Mohr circle envelope and is only valid in a limited stress range. When the stress range considered lies over a wider interval, the nonlinearity of the failure envelope needs to be considered.

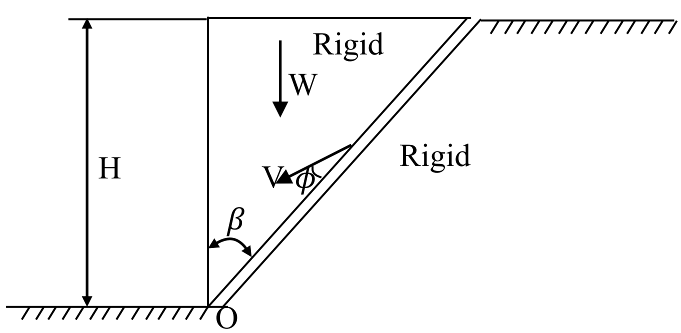


Figure 8.17 Translational failure mode with a block sliding down with a velocity V

Figure 8.17 shows a rigid block sliding down with a velocity V with respect to the fixed soil mass. The slip band for MC material will be planar according to the theory of limit analysis (Chen 1975), and the velocity discontinuity is required to make an angel ϕ to the velocity.

If nonlinear yield criteria are considered, this conclusion will no longer hold. Consider a soil material following a power-type yield criterion (Figure 8.18). The equivalent (tangential) friction angle ϕ is not constant any more, but is a function of the confining pressure as shown in Figure 8.18. The shape of the rupture surface can be approximated using a numerical procedure and an example is presented in Figure 8.19. Figure 8.19(b) predicts a more realistic rupture surface than a plane from inspection.

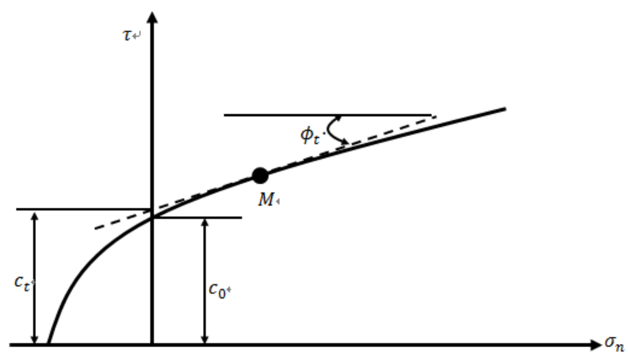
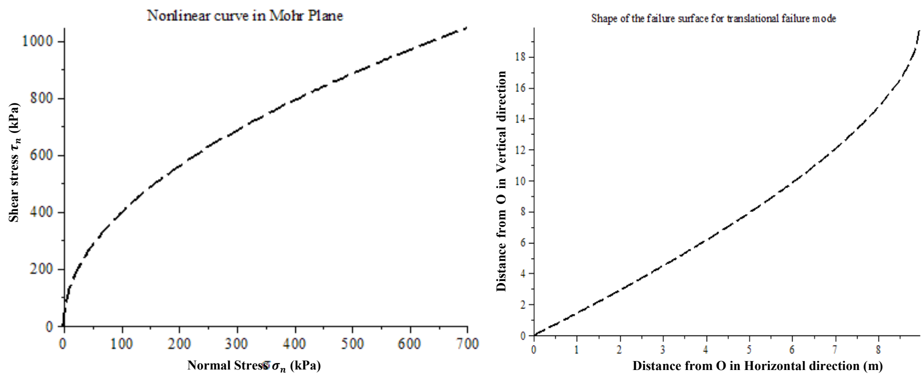


Figure 8.18 Tangential cohesion and tangential friction angle corresponding stress point M



(a) (b)

Figure 8.19 (a) a particular power type yield criterion and (b) the corresponding failure surface for the translational motion

The effects of the nonlinearity of yield criteria on the obtained earth pressure will be illustrated in this section. For simplicity, only the active earth pressure problem will be considered, and the nonlinear yield criterion is used similar to what have been discussed in the slope stability analysis.

In Table 8.10, solutions corresponding to the translation mechanism are obtained with the failure mechanism (Figure 8.17) and the general tangential technique (Yang 2007), which are the strict upper bound solutions. The "Extended Rankine" solutions are the simple extension of Rankine's solution by considering the stress limit equilibrium along the soil-structure interface and the total thrust are obtained by integration along the wall back.

Table 8.10 Static active earth pressure on a smooth and vertical wall with different methods

	Coefficient m				
	1.2	1.4	1.6	1.8	2.0
Translational failure mechanism	20.88	27.01	32.08	36.26	39.72
Extended Rankine solutions	20.88	27.01	32.08	36.26	39.72
Current static approach	22.28	29.35	34.85	39.16	42.65
Difference (%)	6.28	7.97	7.95	7.41	6.87

$k_h = 0, q = 0, \delta = 0^0, P_f = 0, \beta = 90^0, \gamma = 18kN/m^3$, $H = 4m, c_0 = 9kPa, \sigma_t = 20.0kPa$

It is evident from Table 8.10 that the current static approach results in the upper bounds on the active earth thrust. The gap between the static and kinematic solutions ranges between 6%~8%. It should be highlighted that even though solutions from the extended Rankine method are almost identical to those with a translational failure mechanism, it does not by any means imply that the extension of Rankine's method yields the critical lower bound to the active earth thrust, as the extended Rankine solutions are obtained from an incomplete stress field, and there is no way to determine if they are the bound solutions or not.

Table 8.11 Seismic active earth thrusts acting on a smooth and vertical wall with various methods.

$k_h = 0.1, q = 0, \delta = 0^0, P_f = 0, \beta = 90^0, \gamma = 18kN/m^3$, $H = 4m, c_0 = 9kPa, \sigma_t = 20.0kPa$

	Coefficient m							
	1.2	1.4	1.6	1.8	2.0	3.0	4.0	5.0
Extended Rankine solution	35.28	41.41	46.48	50.66	54.12	64.91	70.39	73.66
Log-spiral mechanism	32.66	39.78	45.61	50.39	54.32	66.49	72.64	76.31
Current Static solution	33.67	41.52	47.77	52.57	56.42	67.68	73.25	76.49
Difference (%)	3.00	4.19	4.52	5.14	3.72	1.76	0.83	0.24

Active earth thrusts calculated with different methods for various m values are given in Table 8.11. The gap between the two bound solutions varies between 0.2~5.2%. When m grows larger, the gap reduces to a very small value, indicating that very good solution is achieved. This trend can be explained by the fact that for a large m value, the yield envelope will become level (Figure 8.20) and the influence of the nonlinearity become less dominant. For example, a global linear approximation gives $\phi' = 3.25^0$ for $m = 4.0$

and $\phi' = 2.49^0$ for $\mathrm{m} = 5.0$. The error introduced due to the general tangent technique becomes negligible in comparison to effect of the gravity and seismic loading.

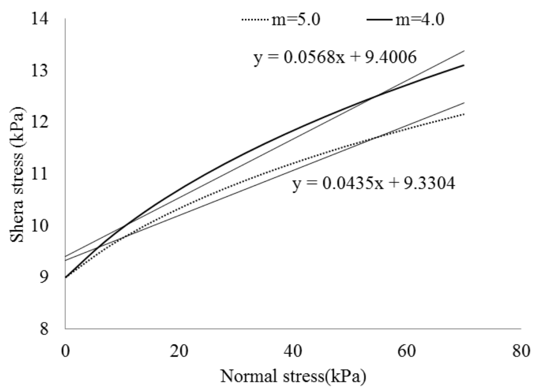


Figure 8.20 Linear approximation of the case $m = 4.0$ and $m = 5.0$

8.4.3.1 **Influence of the nonlinearity on the failure surface**

As explained, the failure surface associated with the soil mass following a nonlinear yield criterion is no longer planar. Figure 8.21 shows the fracture surfaces for various m values captured by the mesh adaptation procedure. Apparently, when the power-type yield criterion is utilized, the failure surface curves up as it approaches the ground surface. It can also be found that the rest of the parameters being constant, an increase in m will give a decrease to the failure angle.

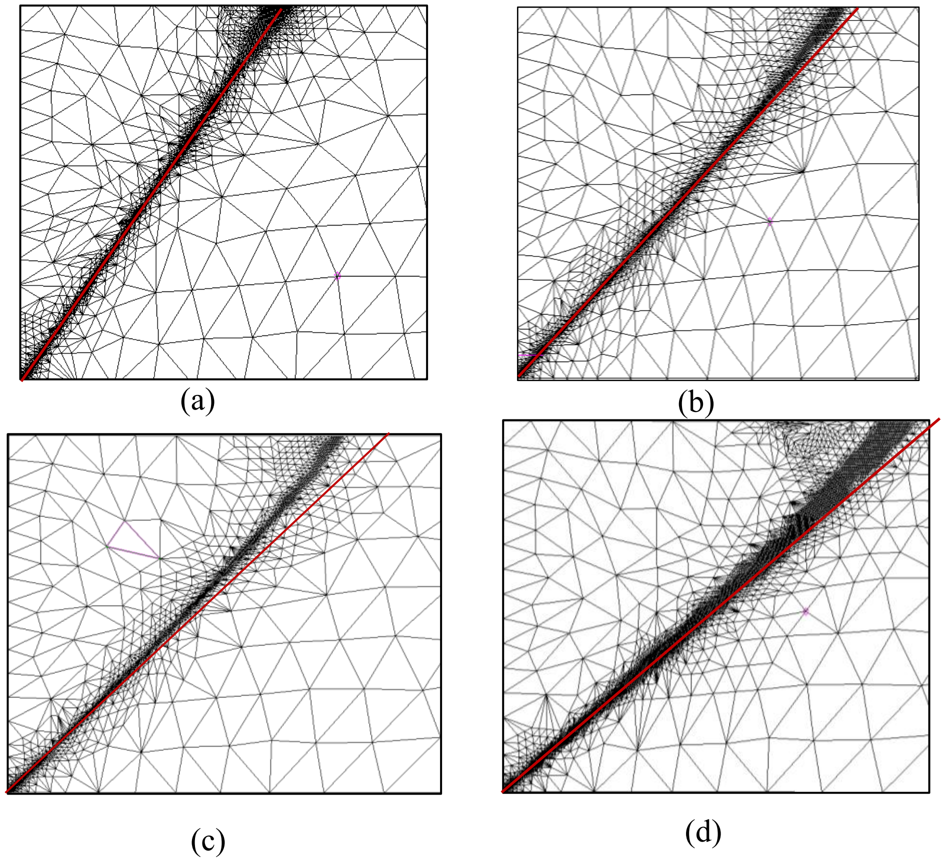


Figure 8.21 Effect of the nonlinear parameter on the fracture surface (a) $m = 1.0$ (b) $m = 1.2$, (c) $m = 1.6$, (d) m=3.0

8.4.3.2 **Influence of the seismic coefficient on the active thrust**

Figure 8.22(a) shows the results of active earth thrusts corresponding to $\gamma = 18\mathrm{kN/m^3}$, $\beta = 0, \mathrm{H} = 4.0\mathrm{m}, \delta = 0.0$ and $\sigma_t = 20\mathrm{kPa}$ and Figure 8.22 (b) illustrates the effect of vertical/horizontal seismic coefficient ratio on the magnitude of active earth thrust for $m = 3.0$.

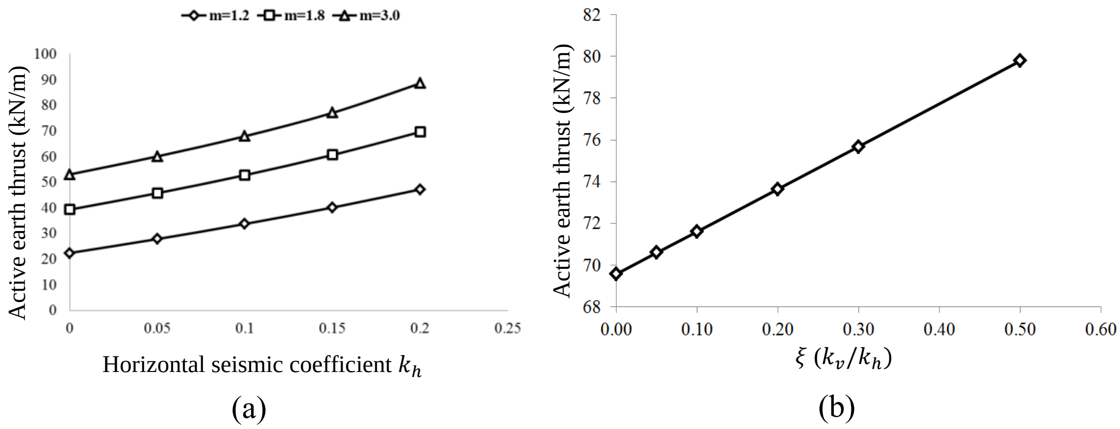


Figure 8.22 Effect of the seismic coefficient on the active earth thrust (a) influence of the horizontal seismic coefficient (b) influence of the vertical to horizontal seismic loading ratio

8.4.3.3 **Influence of the friction angle**

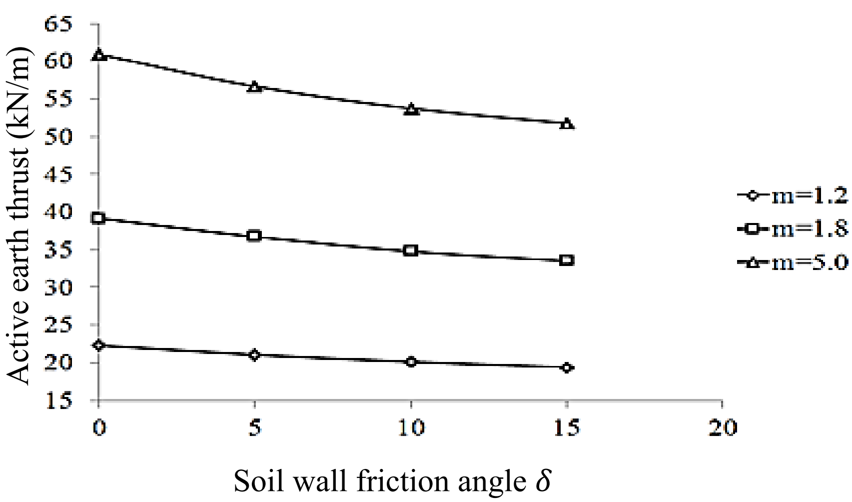


Figure 8.23 Effects of the soil-wall fri ction angle on the active earth thrust

The variation of the active earth thrust with different soil-wall friction angles are plotted in Figure 8.23. The data are calculated from $\beta = 0, c_0 = 3.0kPa, \sigma_t = 20kPa$. It can be observed that the active earth thrust decreases with the rise in the friction angle δ.

The effects of soil-wall friction angle on the active earth thrust become more significant as m increases. A drop of 17% in the active earth pressure is observed as δ increases from 0 to 15, which is almost twice of the corresponding magnitude for $m = 1.2$.

8.5 Stability of vertical cut

This example considers the critical height of an unsupported vertical cut. Unlike the previous two examples where the surface traction is subjected to the loading multiplier, the unit weight of soil mass is to be optimised. For purely cohesive soils, the stability number of a vertical cut is defined as in eq.(8.8)

$$N_s = \frac{\gamma H}{c} \tag{8.8}$$

Since the exact value for N_s is unknown to authors' knowledge, the reference will be made to as the best bound solution of N_s reported by Kammoun et al. (Kammoun et al. 2010) and Pastor et al. (Pastor et al. 2009), computed by solving the limit analysis with large scale model with decomposition method N_s as $3.77522 \le N_s \le 3.77756$.

The intention of this example is not to further tighten bounds, but to demonstrate that a few iterations of refinement could provide a very good estimate of N_s from a general initial mesh. Figure 8.24 shows that four refinements of the initial mesh increase the initial N_s from 3.41 with an error 9.6% (calculated from the bound solutions mentioned above) to 3.767 with an error 0.2%.

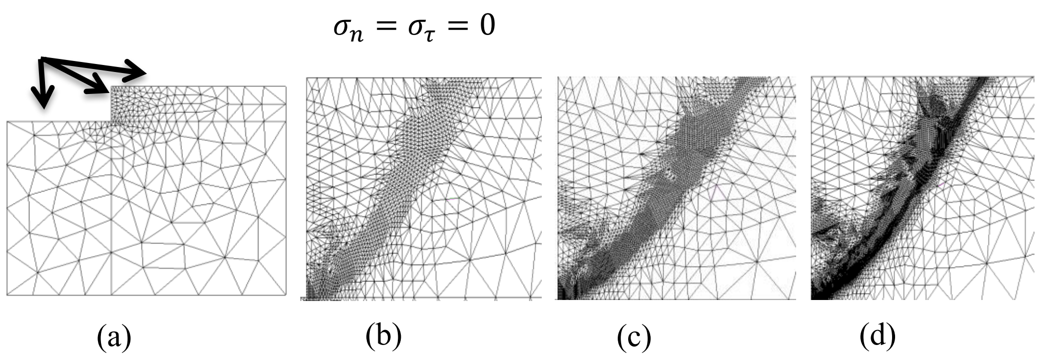


Figure 8.24 (a) initial mesh with 419 elements and $N_s = 3.41$with error 9.6% (b) close view of mesh after two refinements,E =2700, and $N_s = 3.731$ (c) close view of mesh after 3 refinements $E = 6685$, $N_s = 3.754$, (d) close view of mesh after 4 refinement $E = 14635, N_s = 3.767$ with error 0.2%.

8.6 3D lower bound bearing capacity

In applications to the 3D problems, preparing a mesh incorporating the anticipated failure mechanism is very difficult and tedious process whereas a general mesh will considerably underestimate the lower bound solution. Therefore, the significance of the mesh adaptation will become pronounced. We will illustrate this point by considering a 3D bearing capacity problem of material obeying the Druck-Prager yield criterion as shown in eq. (8.9)

$$\mathrm{F}(\sigma) = a\sigma_m + \bar{\sigma} = k \tag{8.9}$$

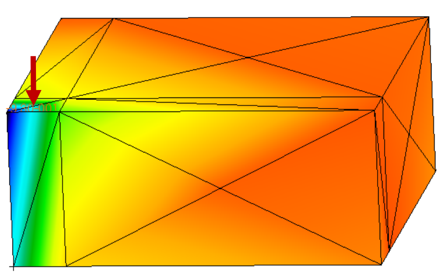
where a and k are two material constants to be determined. They will be back fitted from the MC yield criterion using the equal area method (Yang et al. 2003a).

$$a = \frac{6\sqrt{3}\sin\phi'}{\sqrt{2\sqrt{3}\pi(9-\sin^2\phi')}};\ k = \frac{6\sqrt{3}\,\mathrm{c}\cos\ \phi'}{\sqrt{2\sqrt{3}\pi(9-\sin^2\phi')}} \tag{8.10}$$

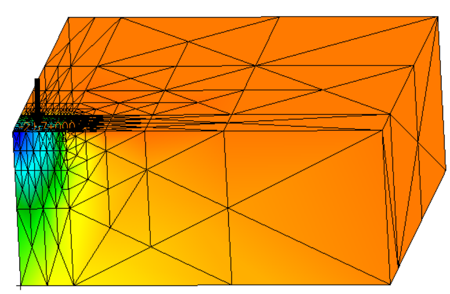
The initial and output final meshes for a square footing (i.e. L/B =1) with the σ_z stress field are shown in Figure 8.25 respectively. As discussed earlier, the bisection method is preferred in the FELA. Similar to the conclusion from the 2D examples, the cost due to the iterative refinement is relatively small in comparison to the solution time, less than 5% of the solution time. It is noted that the initial mesh updates to the output final mesh (Figure 8.25 b) within 12 iterations in two minutes, increasing the initial estimate of N_C from 4.59 to 6.16.

Table 8.12 Comparison of 3D lower bound bearing capacity coefficient N_c for square surface footing on a purely cohesive and weightless soil

Current work		Yang's work (Criterion II)(Yang et al. 2003a)		Michalowski (2001)
Initial mesh	Final mesh	Coarse mesh	Fine mesh	UB formulation
4.59	6.16	5.35	5.57	6.83



(a)



(b)

Figure 8.25 (a) initial mesh and (b) updated mesh

8.7 Summary

A number of the stability problems in geotechnical discipline are considered, and the SOCP and NLP formulation of the FELA combined with mesh adaptation are applied in this study. The procedures have shown great potential of FELA in practical applications in geotechnical engineering. The computing efficiency of the currently developed C++ library is demonstrated to be effective by updating very coarse meshes to over 200,000 elements with optimal concentration of grids in less than half an hour for the MC material under plane strain condition. In addition to the high accuracy of solution, reasonably accurate failure mechanisms have also been obtained with the mesh adaptation.

The nonlinear power type yield criterion which has gained much attention in the literature has been considered in a systematic fashion in this study. The lower bound solutions are calculated for analyses of slope stability and lateral earth pressure problems. The failure mechanisms shows that when the nonlinearity of the yield criterion is considered, the slip surface will not be planar or logspiral as for the MC material.

For problem with the presence of stress discontinuities, i.e. stress point at the edge of the footing convergence to the exact solution sometimes meet difficulties or the so called “locking” phenomenon, but chances are actually not common and the improvement of the solution due to mesh adaptation is dramatic even for this case.

CHAPTER 9: CONCLUSION AND FUTURE WORK

9.1 Summary and conclusions

In the present study, 2D and 3D FELA have been studied with different error estimators and mesh adaptation scheme. Very good solution can now be determined from a general mesh without any consideration to the actual failure mechanism. The practical and theoretical aspects of the present study have been clearly demonstrated through the previous chapters. During the course of study, some conclusions can be drawn:

(1) From the numerical applications in various stability problems in geotechnical engineering, the limit solution is dramatically improved by several mesh adaptations that is totally automatically controlled by the localized mesh adaptation developed in this research. A coarse initial mesh could be supplied as the input while the optimal mesh/ failure mechanism is obtained as part of the solution, which eliminate the need for the preparation of a suitable mesh for stability analysis. If a warm start of solver is used for the optimization solver, the performance of the adaptation could be more impressive in practical applications. The total solution time for adaptive procedures indicates that a model with several iterations can be solved in minutes, rendering the adaptive procedure both practical and effective for complicated large-scale problems. Slip bands captured by the adaptive procedure could give the engineers a good vision of the plastically deformed zone other than the solutions (in terms of pressure or factor of safety). Though the velocity characteristics (actual failure mechanism) are different from

stress characteristics captured by limit analysis when the non-associated flow rule is concerned, these slip bands indeed provide useful knowledge about the ultimate limit state of a system.

(2) The SOCP formulation of the FELA is robust and efficient for both the lower bound and upper bound analyses. The resulting problems are solved with high efficiency even with very large models, e.g., the optimal solutions of the convex optimization problem with millions of the variables generated by the adaptive mesh refinement procedure are found within half an hour on a desktop computer for various types of stability problems. For this aspect, the proposed algorithm and solution technique can be considered as useful to practical engineering besides academic study

(3) Recovery-based and residual-based error estimators originally proposed for FEM have been tailored for the mesh adaptation purpose in the FELA. Results obtained from adaptation analyses are compared with methods using simple back-calculation of the slackness of the yield criterion and the dual variables. These mesh adaptation strategies are found practically equivalent to each other and could be used in practical applications.

(4) The nonlinearity of the yield envelope on the meridional plane could be considered in a systematic manner in the FELA. Nonlinear yield criteria are applied in the natural form without approximation. In the application of the nonlinear yield criterion to the determination of the active earth pressure, the plane failure surface is not kinematically admissible any more for a translational motion as in the case of the MC material, and the failure surface curved up as it approaches the ground surface which is different from the classical MC material.

For the slope stability problem with a power-type yield criterion, the stability number depends on the nonlinearity of the yield envelope, particularly when the stress level is low.

(5) To compare the results obtained from limit equilibrium method or the conventional upper bound method in which the yield function is expressed in terms of the shear and normal stresses on the Mohr plane, a nonlinear yield envelope on the Mohr plane is transformed to an equivalent representation on the triaxial plane leading to a highly nonlinear equation system which is difficult to be solved. The curve fitting procedure proposed in this research is simple in concept and high accuracy can be achieved if a piecewise curve fitting is performed.

9.2 Recommendations for the future work

The FELA developed in the research is not restricted to the MC yield criterion and could be extended to a variety of yield criteria. To facilitate the procedure, an abstract class called "*YieldCriterion*" has been developed in our code. More sophisticated yield criteria could be implemented by overriding the functions value and the information of the gradient and Hessian (optional) in the inherited classes, thus, giving the chance of revealing the richer property of soils. For instance, the Hoek-Brown yield criterion and Lade criterion could be experimented in the stability analysis for full three-dimensional analyses using the NLP.

In the present research, third-party solvers, MOSEK as the SOCP solver and IPOPT and KNITRO as NLP solvers have been adopted to find the solution of the resulting optimization problem. It is expected to be beneficial to develop solvers particularly suited for optimization problems stemming from the FELA that exploits the structure of constraint matrices. Moreover, parallel computing technique and decomposition technique could be included to reduce the time involved in the solution stage and to improve the stability of the convergence of NLP algorithm.

Despite the rapid developments of FELA, applications in full three-dimensional problems are still limited which is in contrast to the finite element method. More efforts should be spent in experimenting this technique in three-dimensional problems. The lack of the practical applications is partially due to the lack of friendly interface software which can handle the pre-and post-process of a general complicated problem.

In this research, perfectly plastic material and small change in geometry are assumed. These assumptions could be further loosened by performing a sequence of limit analyses, in which the geometry and the yield criteria are both updated in every subsequent stability problem. Given the current capability of the computers and the efficiency of the nonlinear algorithm, the sequential limit analysis should be practical, and better insight into the failure mechanisms in the stability analysis could be expected.

The adaptive mesh refinement presented in this research is somewhat inspired by the existing adaptive techniques in the FEM. Since it has been well known that the fan elements would considerably improve the accuracy of the solution of stability problems with the presence of the stress singularities, it might be beneficial to carefully design a local error estimator and mesh refinement/coarsening strategy that are capable of detecting the need for a fan subdivision. Moreover, it would be interesting to introduce the discontinuities on an adaptive basis, because at the ultimate condition, a large portion of the domain is in the rigid plastic state.

REFERENCES

Abbo, A. J., Lyamin, A. V., Sloan, S. W., and Hambleton, J. P. (2011). "A C2 continuous approximation to the Mohr-Coulomb yield surface." *International Journal of Solids and Structures*, 48(21), 3001-3010.

Abbo, A. J., and Sloan, S. W. (1995). "A smooth hyperbolic approximation to the Mohr-Coulomb yield criterion." *Computers & Structures*, 54(3), 427-441.

Anderheggen, E., and Knopfel, H. (1972). "Finite element limit analysis using linear programming." *International Journal of Solids and Structures*, 8(12), 1413-1431.

Andersen, E. D., Roos, C., and Terlaky, T. (2003). "On implementing a primal-dual interior-point method for conic quadratic optimization." *Mathematical Programming*, 95(2), 249-277.

Andersen, K. D., Christiansen, E., Conn, A., and Overton, M. L. (2000). "An efficient primal-dual interior-point method for minimizing a sum of Euclidean norms." *SIAM Journal on Scientifi computing*, 22(1), 243-262.

Babuska, I., and Miller, A. D. (1978). "A posteriori error estimates for the finite element method." *International Journal for Numerical Methods in Engineering*, 12(10), 1597-1615.

Babuska, I., and Miller, A. D. (1987). "A feedback finite-element method with a posteriori error estimation .1. the finite-element method and some basic properties of the a posteriori error estimator." *Computer Methods in Applied Mechanics and Engineering*, 61(1), 1-40.

Babuska, I., and Rheinboldt, W. C. (1979). "Analysis of optimal finite-element meshes in R." *Mathematics of Computation*, 33(146), 435-463.

Baker, R. (2004). "Nonlinear Mohr envelopes based on triaxial data." *Journal of Geotechnical and Geoenvironmental Engineering*, 130(5), 498-506.

Baker, R., and Frydman, S. (1983). "Upper bound limit analysis of soil with non-linear failure criterion." *Soils and Foundations*, 23(4), 34-42.

Bandini, P. (2003). *Numercial limit analysis for slope stability and bearing capacity calculations*, PhD thesis, Purdue University, West Lafayette.

Bank, R. E., Sherman, A. H., and Weiser, A. (1983). "Some refinement algorithms and data structures for regular local mesh refinement." *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences 1*, 1, 3-17.

Bey, J. (1995). "Tetrahedral grid refinement." *Computing*, 55(4), 355-378.

Bishop, A. W. (1966). "The strength of soils as engineering materials, Sixth Rankine Lecture." *Geotechnique*, 16(2), 91-128.

Booker, J. R. (1969). *Application of Theories of plasticity to cohesive fricitonal soils*, PhD thesis, Sydney Universtiy, Sydney.

Borges, L., Feijoo, R., and Zouain, N. (1999). "A directional error estimator for adaptive limit analysis." *Mechanics Research Communications*, 26(5), 555-563.

Borges, L., Zouain, N., Costa, C., and Feijoo, R. (2001). "An adaptive approach to limit analysis." *International Journal of Solids and Structures*, 38(10-13), 1707-1720.

Bottero, A., Negre, R., Pastor, J., and Turgeman, S. (1980). "Finite element method and limit analysis theory for soil mechanics problems." *Computer Methods in Applied Mechanics and Engineering*, 22, 131-149.

Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*, Cambridge: Cambridge University Press.

Brenner, S. C., and Scott, L. R. (2008). *The mathematical theory of finite element methods*, New York: Springer.

Burland, J. B., Broms, B. B., and DEMELLO, V. F. B. (1977a). "Behavior of foundations and structures, State of the art Report, Session 2." *Proceedings of 9th International Conference of SMFE*. City: Tokyo.

Burland, J. B., Longworth, T. I., and Moore, J. F. A. (1977b). "Study of ground movement and progressive failure caused by a deep excavation in Oxford clay." *Geotechnique*, 27(4), 557-591.

Byrd, R. H., Nocedal, J., and Waltz, R. A. (2006). "KNITRO: An Integrated Package for Nonlinear optimization in Large-Scale Nonlinear optimization." *G. di Pillo and M. Roma, eds*, 35-59.

Carey, G. F. (1997). *Computational grids: generation, adaptation,and solution strategies*, London: Taylor & Francis.

Carter, J. P., Desai, C. S., Potts, D. M., Schweiger, H. F., and Sloan, S. W. (2000). "Computing and computer modelling in geotechnical engineering." *ISRM International Symposium. International Society for Rock Mechanics*. City: Melbourne.

Cavicchi, A., and Gambarotta, L. (2006). "Two-dimensional finite element upper bound limit analysis of masonry bridges." *Computers & Structures*, 84(31-32), 2316-2328.

Charles, J. A. (1982). "An Appraisal of the Influence of a Curved Failure Envelope on Slope Stability." *Geotechnique*, 32(4), 389-392.

Charles, J. A., and Watts, K. S. (1980). "The Influence of Confining Pressure on the Shear-Strength of Compacted Rockfill." *Geotechnique*, 30(4), 353-367.

Chen, J. (2004). *Slope stability analysis using rigid elements*, PhD thesis, The Hong Kong Polytechnic Unviersity, Hong Kong.

Chen, J., Yin, J. H., and Lee, C. F. (2003). "Upper bound limit analysis of slope stability using rigid finite elements and nonlinear programming." *Canadian Geotechnical Journal*, 40(4), 742-752.

Chen, J., Yin, J. H., and Lee, C. F. (2004). "Rigid finite element method for upper bound limit analysis of soil slopes subjected to pore water pressure." *Journal of Engineering Mechanics-ASCE*, 130(8), 886-893.

Chen, J., Yin, J. H., and Lee, C. F. (2005). "A three-dimensional upper-bound approach to slope stability analysis based on RFEM." *Geotechnique*, 55(7), 549-556.

Chen, S. S., Liu, Y. H., and Cen, Z. Z. (2008). "Lower-bound limit analysis by using the EFG method and non-linear programming." *International Journal for Numerical Methods in Engineering*, 74(3), 391-415.

Chen, W. F. (1975). *Limit analysis and soil plasticity*, Amsterdam: Elsevier Scientific Publishing Company.

Chen, W. F., and Liu, X. L. (1990). *Limit analysis in soil mechanics*, New York: Elsevier

Chen, W. F., and Mizuno, E. (1979). "On material constants for siol and concrete models." *Proceedings, 3rd ASCE/EMD Specialty conference*. City: Austin, pp. 539-542.

Cheng, Y. M. (2003). "Seismic lateral earth pressure coefficients for c-phi soils by slip line method." *Computers and Geotechnics*, 30(8), 661-670.

Cheng, Y. M. (2004). "N-q factor for pile foundations by Berezantzev." *Geotechnique*, 54(2), 149-150.

Cheng, Y. M., and Au, S. K. (2005). "Bearing capacity problem by slip line method." *Canadian Geotechnical Journal*, 42, 1232-1241.

Cheng, Y. M., LI, D. Z., Li, L., Sun, Y. J., Baker, R., and Yang, Y. (2011a). "Limit equilibrium method based on an approximate lower bound method with a variable factor of safety that can consider residual strength." *Computers and Geotechnics*, 38(5), 623-637.

Cheng, Y. M., Li, L., and Chi, S. C. (2007a). "Performance studies on six heuristic global optimization methods in the location of critical slip surface." *Computers and Geotechnics*, 34(6), 462-484.

Cheng, Y. M., Li, L., and Fang, S. S. (2011b). "Improved harmony search methods to replace variational principle in geotechnical problems." *Journal of Mechanics*, 27(1), 107-119.

Cheng, Y. M., Y.Y., H., and W.B., W. (2007b). "General axisymmetric active earth pressure by method of characteristic-theory and numerical formulation." *Journal of Geomechanics-ASCE*, 7(1), 1-15.

Cheng, Y. M., Zhao, Z. H., and Sun, Y. J. (2010). "Evaluation of interslice force function and discussion on convergence in slope stability analysis by the lower bound method." *Journal of Geotechnical and Geoenvironmental Engineering*, 136(8), 1103-1113.

Christiansen, E. (1980). "Limit analysis in plasticity as a mathematical programming problem." *Calcolo*, 17, 41-65.

Christiansen, E. (1981). "Computation of limit loads." *International Journal for Numerical Methods in Engineering*, 17(10), 1547-1570.

Christiansen, E. (1996). "Limit analysis of collapse states", in Ciarllet P.G. and J. L. Lions, (eds.), *Handbook of Numerical Analysis*. North-Holland: Gulf Professional Publishing, pp. 193-312.

Christiansen, E., and Edmund, O. S. (2001). "Automatic mesh refinement in limit analysis." *International Journal for Numerical Methods in Engineering*, 50(6), 1331-1346.

Ciria, H. (2002). *Computation of upper and lower bounds in limit analysis using second-order cone programming and mesh adaptivity*, Master thesis, Massachusetts Institute of Technology.

Ciria, H., Peraire, J., and Bonet, J. (2008). "Mesh adaptive computation of upper and lower bounds in limit analysis." *International Journal for Numerical Methods in Engineering*, 75(8), 899-944.

Clausen, J., Andersen, L., and Damkilde, L. (2010). "On the differences between the Ducker-Prager criterion and exact implementation of the Mohr-Coulomb criterion in FEM Calculation", in T. Benz and S. Nordal, (eds.), *Numerical methods in Geotechnical Engineering*. London: Taylor & Francis Group, pp. 101-105.

Clausen, J., and Damkilde, L. (2008). "An exact implementation of the Hoek-Brown criterion for elasto-plastic finite element calculations." *International Journal of Rock Mechanics and Mining Sciences*, 45(6), 831-847.

Collins, I. F., Gunn, C. I. M., Pender, M. J., and Wang, Y. (1988). "Slope stability analyses for materials with a non-linear failure envelope." *International Journal for Numerical and Analytical Methods in Geomechanics*, 12(5), 533-550.

Coulomb, C. A. (1773). "Essai sur une application des règles de maximis & minimis àquelques problèmes de statique, relatifs a l'architecture." *Mémoires de*

Mathèmatique & de Physique présentés à L'Academie Royale des Sciences, 7, 343-382.

Cundall, P. A., and Board, M. (1988). "A microcomputer program for modelling large-strain plasticity problems." *Proceeding of 6th Internatinal Conference on Numerical Methods in Geomechanics*. City: Innsbruck, Austria, pp. 2101-2108.

Davis, R. O., and Selvadurai, A. P. S. (2002). *Plasticity and geomechanics*, Cambridge, U.K. ; New York: Cambridge University Press.

De Jong, G., and Josselin, D. (1957). "Graphical method for the determination of slip-line fields in soil mechanics." *De Ingenieur*, 69(29), 61-65.

Donald, I. B., and Chen, Z. Y. (1997). "Slope stability analysis by the upper bound approach: fundamentals and methods." *Canadian Geotechnical Journal*, 34(6), 853-862.

Drescher, A., and Christopoulos, C. (1988). "Limit analysis slope stability with monlinear yield condition." *International Journal for Numerical and Analytical Methods in Geomechanics*, 12(3), 341-345.

Drescher, A., and Detournay, E. (1993). "Limit load in translational failure mechanisms for associative and nonassociative Materials." *Geotechnique*, 43(3), 443-456.

Drucker, D. C. (1953). "Limit analysis of two- and three-dimensional soil mechanics problems." *Journal of the Mechanics and Physics of Solids*, 1(14), 217-226.

Drucker, D. C., Greenberg, H. J., and Prager, W. (1951). "The safety factor of an elastic-plastic body in plane strain." *Journal of Applied Mathematics*, 18, 371.

Drucker, D. C., Greenberg, H. J., and Prager, W. (1952). "Extended limit design theorems for continuous media." *Quarterly of Applied Mathematics*, 9, 381-389.

Drucker, D. C., and Prager, W. (1952). "Soil mechanics and plastic analysis or limit design." *Quarterly of Applied Mathematics*, 10, 157-165.

Frydman, S., and Burd, H. J. (1997). "Numerical studies of bearing-capacity factor N-gamma." *Journal of Geotechnical and Geoenvironmental Engineering*, 123(1), 20-29.

Goodman, R. E. (1989). *Introduction to rock mechanics*, New York ; Chichester: Wiley.

Gratsch, T., and Bathe, K. J. (2005). "A posteriori error estimation techniques in practical finite element analysis." *Computers & Structures*, 83(4-5), 235-265.

Griffiths, D. V. (1982). "Computation of bearing capacity factors using finite elements." *Geotechnique*, 32(3), 195-202.

Griffiths, D. V. (1990). "Failure criteria interpretation based on Mohr-Coulomb friction." *Journal of Geotechnical Engineering-ASCE*, 116(6), 986-999.

Griffiths, D. V., and Lane, P. A. (1999). "Slope stability analysis by finite elements." *Geotechnique*, 49(3), 387-403.

Griffiths, D. V., and Marquez, M. (2008). "Three-dimensional slope stability analysis by elasto-plastic finite elements." *Geotechnique*, 58(8), 683-685.

Gvozdev, A. A. (1960). "The determination of the value of the collapse load for statically indeterminate systems undergoing plastic deformation." *International Journal of Mechanical Sciences*, 1(4), 322-335.

Hassiotis, S., and Xiong, K. (2007). *Deformation of Cohesionless Fill Due to Cyclic Loading*. Stevens Institute of Technology, Hoboken.

Hill, R. (1950). *The mathematical theory of plasticity*, London: Oxford University Press.

Hjiaj, M., Lyamin, A. V., and Sloan, S. W. (2005). "Numerical limit analysis solutions for the bearing capacity factor N-gamma." *International Journal of Solids and Structures*, 42(5-6), 1681-1704.

Hoek, E., and Brown, E. T. (1988). "The Hoek-Brown failure criterion - a 1988 update", in J. H. Curran, (ed.), *Proceedings of 15th Canadian Rock Mechanics Sympodsium*. Toronto: University of Toronto, pp. 31-38.

Hoek, E., Carranza, C., and Corkum, B. (2002). "Hoek-Brown criterion –2002 edition." *Proceedings of NARMS-TAC Conference*. City: Toronto, pp. 267-273.

Hoek, E., and Marinos, P. (2007). "A brief history of the development of the Hoek Brown failure criterion." *Soils and Rocks*, 2, 1-13.

Huh, H., and Yang, W. H. (1991). "A general algorithm for limit solutions of plane-stress problems." *International Journal of Solids and Structures*, 28(6), 727-738.

Kammoun, Z., Pastor, F., Smaoui, H., and Pastor, J. (2010). "Large static problem in numerical limit analysis: A decomposition approach." *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(18), 1960-1980.

Kelly, D. W., Gago, J. P. D. R., Zienkiewicz, O. C., and Babuska, I. (1983). "A-posteriori error analysis and adaptive processes in the finite-element method. 1. error analysis." *International Journal for Numerical Methods in Engineering*, 19(11), 1593-1619.

Khatri, V. N., and Kumar, J. (2009). "Bearing capacity factor N-c under phi=0 condition for piles in clays." *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(9), 1203-1225.

Kim, J. (1998). *Limit analysis of soil slope stability using finite elements and linear programming*, PhD thesis, Purdue University, West Lafayett.

Kim, J., Salgado, R., and Lee, J. (2002). "Stability analysis of complex soil slopes using limit analysis." *Journal of Geotechnical and Geoenvironmental Engineering*, 128(7), 546-557.

Kirk, B. S., Peterson, J. W., Stogner, R. H., and Carey, G. F. (2006). "libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations." *Engineering with Computers*, 22(3-4), 237-254.

Koiter, W. T. (1953). "Stress-strain relations, uniqueness and variational theorems of elasto-plastic material with singular yield surface." *Quarterly of Applied Mathematics*, 11, 350-354.

Kossaczky, I. (1994). "A recursive approach to local mesh refinement in 2 and 3 dimensions." *Journal of Computational and Applied Mathematics*, 55(3), 275-288.

Kötter, F. (1903). " Die bestimmung des druckes an gekrummten gleitflachen, eine Aufgabe aus der Lehre vom eriddruck." *Sizungsberichte der Akademie der Wissenschaften*. City: Berlin, pp. 229-233.

Krabbenhoft, K., and Damkilde, L. (2003). "A general non-linear optimization algorithm for lower bound limit analysis." *International Journal for Numerical Methods in Engineering*, 56(2), 165-184.

Krabbenhoft, K., Lyamin, A. V., Hjiaj, M., and Sloan, S. W. (2005). "A new discontinuous upper bound limit analysis formulation." *International Journal for Numerical Methods in Engineering*, 63(7), 1069-1088.

Krabbenhoft, K., Lyamin, A. V., and Sloan, S. W. (2007a). "Formulation and solution of some plasticity problems as conic programs." *International Journal of Solids and Structures*, 44(5), 1533-1549.

Krabbenhoft, K., Lyamin, A. V., and Sloan, S. W. (2007b). "Shakedown analysis of cohesive-frictional materials using adaptive remeshing." *Numerical Models in Geomechanics: Numog X*, 233-238.

Krabbenhoft, K., Lyamin, A. V., and Sloan, S. W. (2008). "Three-dimensional Mohr-Coulomb limit analysis using semidefinite programming." *Communications in Numerical Method in Engineering*, 24(11), 1107-1119

Krahn, J. (2004). *Stability modeling with SLOPE/W: An engineering methodology*, Calgary, Alta., Canada: GEO-SLOPE/W International Ltd.

Kuhn, H. W., and Tucker, A. W. (1951). "Nonlinear programming." *Proceedings of 2nd Berkeley Symposium*. City: Berkeley, pp. 481–492.

Kumar, J., and Khatri, V. N. (2011). "Bearing capacity factors of circular foundations for a general c-phi soil using lower bound finite elements limit analysis." *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(3), 393-405.

Labit, Y., Peaucelle, D., and Henrion, D. (2002). "SEDUMI interface 1.02 - A tool for solving LMI problems with SEDUMI." *2002 IEEE International Symposium on Computer Aided Control System Design Proceedings*. City: Glasgow, Scotland, pp. 272-277.

Lade, P. V. (1977). "Elastoplastic stress-strain theory for cohesionless soil with curved yield surfaces." *International Journal of Solids and Structures*, 13(11), 1019-1035.

Lade, P. V., and Musante, H. M. (1977). "Failure conditions in sand and remoled clay." *9th Internatinal Conference on Soil Mechanics and Foundation Engineering*. City: Tokyo.

Le, C. V., Askes, H., and Gilbert, M. (2010a). "Adaptive element-free Galerkin method applied to the limit analysis of plates." *Computer Methods in Applied Mechanics and Engineering*, 199(37-40), 2487-2496.

Le, C. V., Gilbert, M., and Askes, H. (2010b). "Limit analysis of plates and slabs using a meshless equilibrium formulation." *International Journal for Numerical Methods in Engineering*, 83(13), 1739-1758.

Lemos, J. V., Hart, R. D., and Cundall, P. A. (1985). "A generalized distinct element programm for modeling jointed rock mass." *Proceedings of the International Symposium on Fundamentals of Rock Joints*. City: Lulea, Sweden, pp. 335-343.

Leu, S. Y. (2005). "Convergence analysis and validation of sequential limit analysis of plane-strain problems of the von Mises model with non-linear isotropic hardening." *International Journal for Numerical Methods in Engineering*, 64(3), 322-334.

Leu, S. Y. (2007). "Analytical and numerical investigation of strain-hardening viscoplastic thick-walled cylinders under internal pressure by using sequential limit analysis." *Computer Methods in Applied Mechanics and Engineering*, 196(25-28), 2713-2722.

Li, A. J., Lyamin, A. V., and Merifield, R. S. (2009a). "Seismic rock slope stability charts based on limit analysis methods." *Computers and Geotechnics*, 36(1-2), 135-148.

Li, A. J., Merifield, R. S., and Lyamin, A. V. (2008). "Stability charts for rock slopes based on the Hoek-Brown failure criterion." *International Journal of Rock Mechanics and Mining Sciences*, 45(5), 689-700.

Li, A. J., Merifield, R. S., and Lyamin, A. V. (2009b). "Limit analysis solutions for three dimensional undrained slopes." *Computers and Geotechnics*, 36(8), 1330-1351.

Li, A. J., Merifield, R. S., and Lyamin, A. V. (2011). "Effect of rock mass disturbance on the stability of rock slopes using the Hoek-Brown failure criterion." *Computers and Geotechnics*, 38(4), 546-558.

Li, H. X., and Yu, H. S. (2005). "Kinematic limit analysis of frictional materials using nonlinear programming." *International Journal of Solids and Structures*, 42(14), 4058-4076.

Li, X. (2007). "Finite element analysis of slope stability using a nonlinear failure criterion." *Computers and Geotechnics*, 34(3), 127-136.

Liu, A. W., and Joe, B. (1995). "Quality Local Refinement of Tetrahedral Meshes Based on Bisection." *Siam Journal on Scientific Computing*, 16(6), 1269-1291.

Liu, A. W., and Joe, B. (1996). "Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision." *Mathematics of Computation*, 65(215), 1183-1200.

Loukidis, D., and Salgado, R. (2009). "Bearing capacity of strip and circular footings in sand using finite elements." *Computers and Geotechnics*, 36(5), 871-879.

Lyamin, A. V. (1999). *Three-dimensional lower bound limit analysis using nonlinear programming*, PhD thesis, University of New Castle, Australia.

Lyamin, A. V., and Sloan, S. W. (2000). "Stability of a plane strain circular tunnel in a cohesive-frictional soil." *Poceedings of the J.R. Booker Memorial Symposium*. City: Sydney, pp. 139-153.

Lyamin, A. V., and Sloan, S. W. (2002a). "Lower bound limit analysis using non-linear programming." *International Journal for Numerical Methods in Engineering*, 55(5), 573-611.

Lyamin, A. V., and Sloan, S. W. (2002b). "Upper bound limit analysis using linear fintie elements and non-linear programming." *International Journal for Numerical and Analytical Methods in Geomechanics*, 26, 165-184.

Lyamin, A. V., and Sloan, S. W. (2003). "Mesh generation for lower bound limit analysis." *Advances in Engineering Software*, 34, 321-338.

Lyamin, A. V., Sloan, S. W., Krabbenhoft, K., and Hjiaj, M. (2005). "Lower bound limit analysis with adaptive remeshing." *International Journal for Numerical Methods in Engineering*, 63(14), 1961-1974.

Lyamin, A. V., Yu, H. S., Sloan, S. W., and Hossain, M. Z. (1998). "Lower bound limit analysis for jointed rockes using the Hoek-Brown yield criterion." *Australian Geomechanics*, 33, 46-62.

Lysmer, J. (1970). "Limit analysis of plane problems in soil mechanics." *Journal of Soil Mechanics and Foundation, Divison, ASCE*, 96(SM4), 1311-1334.

Makrodimopoulos, A. (2010). "Remarks on some properties of conic yield restrictions in limit analysis." *International Journal for Numerical Methods in Biomedical Engineering*, 26(11), 1449-1461.

Makrodimopoulos, A., and Martin, C. M. (2006). "Lower bound limit analysis of cohesive-frictional materials using second-order cone programming." *International Journal for Numerical Methods in Engineering*, 66(4), 604-634.

Makrodimopoulos, A., and Martin, C. M. (2007). "Upper bound limit analysis using simplex strain elements and second-order cone programming." *International Journal for Numerical and Analytical Methods in Geomechanics*, 31(6), 835-865.

Makrodimopoulos, A., and Martin, C. M. (2008). "Upper bound limit analysis using discontinuous quadratic displacement fields." *Communications in Numerical Methods in Engineering*, 24(11), 911-927.

Maksimovic, M. (1989). "Nonlinear failure envelope for soils." *Journal of Geotechnical Engineering-ASCE*, 115(4), 581-586.

Maksimovic, M. (1996). "A family of nonlinear failure envelopes for non-cemented soils and rock discontinuities." *The Electronic Journal of Geotechnical Engineering*, 1(1), 1-36.

Martin, C. M., and Makrodimopoulos, A. (2008). "Finite-element limit analysis of Mohr-Coulomb materials in 3D using semidefinite programming." *Journal of Engineering Mechanics-ASCE*, 134(4), 339-347.

Melan, E. (1938). "Zur plastizitat des raumlichen Kontinuums." *Ingenieur Archiv*, 9, 116-125.

Merifield, R. S., Lyamin, A. V., and Sloan, S. W. (2006). "Limit analysis solutions for the bearing capacity of rock masses using the generalised Hoek-Brown criterion." *International Journal of Rock Mechanics and Mining Sciences*, 43(6), 920-937.

Meyerhof, G. G. (1963). "Some recent research on the bearing capacity of foundations." *Canadian Geotechnical Journal*, 1(1), 16-26.

Michalowski, R. L. (1989). "3-dimensional analysis of locally loaded slopes." *Geotechnique*, 39(1), 27-38.

Michalowski, R. L. (1997). "An estimate of the influence of soil weight on bearing capacity using limit analysis." *Soils and Foundations*, 37(4), 57-64.

Michalowski, R. L. (2001). "Upper-bound load estimates on square and rectangular footings." *Geotechnique*, 51(9), 787-798.

Milani, G. (2008). "3D upper bound limit analysis of multi-leaf masonry walls." *International Journal of Mechanical Sciences*, 50(4), 817-836.

Milani, G. (2011). "Kinematic FE limit analysis homogenization model for masonry walls reinforced with continuous FRP grids." *International Journal of Solids and Structures*, 48(2), 326-345.

Milani, G., Lourenco, P., and Tralli, A. (2007). "3D homogenized limit analysis of masonry buildings under horizontal loads." *Engineering Structures*, 29(11), 3134-3148.

Milani, G., Milani, E., and Tralli, A. (2010). "Approximate limit analysis of full scale FRP-reinforced masonry buildings through a 3D homogenized FE package." *Composite Structures*, 92(4), 918-935.

Mitchell, W. F. (1988). *Unified Multilevel Adaptive Finite Element Methods for Elliptic Problems*, PhD thesis, University of Illinois at Urbana-Champaign.

Mittelman, H. D. (2003). "An independent benchmarking of SDP and SOCP solvers." *Mathematical Programming Series B*, 95, 407-430.

Molino, N., Bridson, R., and Fedkiw, R. (2003). "Tetrahedral mesh generation for deformable bodies"*Proc. Symposium on Computer Animation*. City: San Diego, CA.

Mononobe, N., and Matsuo, H. (1929). "On the determination of earth pressure during earthquakes." *Proceeding of the World Engineering Congress*. City: Tokyo, pp. 179-187.

Morrison, E. E., and Ebeling, R. M. (1995). "Limit equilibrium computation of dynamic passive earth pressure." *Canadian Geotechnical Journal*, 32(3), 481-487.

MOSEK Aps. (2010). "The MOSEK C API manual.Version 6.0 (Revision 103)."*User's Manual and Reference*. www.mosek.com

Munoz, J. J., Bonet, J., Huerta, A., and Peraire, J. (2009). "Upper and lower bounds in limit analysis: Adaptive meshing strategies and discontinuous loading." *International Journal for Numerical Methods in Engineering*, 77(4), 471-501.

Murthy, V. N. S. (2003). *Geotechnical engineering : principles and practices of soil mechanics and foundation engineering*, New York: Marcel Dekker.

Novotortsev, V. I. (1938). "Application of the theroy of plasticity to problems of determining the bearing capacity of building foundation." *Izv., VNIG*, 22.

Pastor, F., Loute, E., and Pastor, J. (2009). "Limit analysis and convex programming: A decomposition approach of the kinematic mixed method." *International Journal for Numerical Methods in Engineering*, 78(3), 254-274.

Pastor, J., Loute, E., and Thai, T. H. (2002). "Limit analysis and new methods of optimization." *Proceedings of the 5th European Conference on Numerical Methods in Engineering*. City: Paris, pp. 211–219.

Pastor, J., and Turgeman, S. (1982). "Limit analysis in axisymmetrical problems: numerical determination of complete statical solutions." *International Journal of Mechanical Sciences*, 24(2), 95-117.

Potts, D. M. (2003). "Numerical analysis: a virtual dream or practical reality?" *Geotechnique*, 53(6), 535-572.

Potts, D. M., and Zdravkovi, L. (2001). *Finite element analysis in geotechnical engineering. Application*, London: Thomas Telford.

Prandtl, L. (1920). "Über die Härte plastischer Körper." *Göttingen Nachr. Math. Phys. K1.*, 12, 74-85.

Reissner, H. (1924). "Zum erddruckproblem", B. a. Burgers, (ed.) *Proc. 1st Int. Congress on Applied Mech.* City: Delft, pp. 295-311.

Rivara, M. C. (1984). "Mesh refinement processes based on the generalized bisection of simplices." *SIAM Journal on Numerical Analysis*, 21(3), 604-613.

Rivara, M. C. (1989). "Selective refinement eerefinement algorithms for sequences of nested triangulations." *International Journal for Numerical Methods in Engineering*, 28(12), 2889-2906.

Rivara, M. C., and Iribarren, G. (1996). "The 4-triangles longest-side partition of triangles and linear refinement algorithms." *Mathematics of Computation*, 65(216), 1485-1502.

Rivara, M. C., and Levin, C. (1992). "A 3-D refinement algorithm suitable for adaptive and multi-grid techniques." *Communications in Applied Numerical Methods*, 8(5), 281-290.

Rosenberg, I. G., and Strenger, F. (1975). "A lower bound on the angles of triangles constructed by bisecting the longest side." *Mathemathics of Computation*, 29(130), 390-395.

Schmidt, A., and Siebert, G. K. (2000). "ALBERT: An adaptive hierarchical finite element toolbox". City: Freiburg,Germany.

Schweiger, H. F. (1994). "On the use of Drucker-Prager failure criteria for earth problems." *Computers and Geotechnics*, 16(3), 223-246.

Sewell, E. G. (1972). *Automatic mesh generation of triangulation for piecewise polynomial approximation*, PhD thesis, Purdue University, West Lafayette.

Shiau, J. S., Augar, C. E., Lyamin, A. V., and Sloan, S. W. (2008). "Finite element limit analysis of passive earth resistance in cohesionless soils." *Soils and Foundations*, 48(6), 843-850.

Sloan, S. W. (1988). "Lower bound limit analysis using finite elements and linear programming." *International Journal for Numerical and Analytical Methods in Geomechanics*, 12(1), 61-77.

Sloan, S. W. (1989). "Upper bound limit analysis using finite-elements and linear-programming." *International Journal for Numerical and Analytical Methods in Geomechanics*, 13(3), 263-282.

Sloan, S. W., and Booker, J. R. (1986). "Removal of singularities in Tresca and Mohr-Coulomb yield functions." *Communications in Applied Numerical Methods*, 2(2), 173-179.

Sloan, S. W., and Kleeman, P. W. (1995). "Upper bound limit analysis using discontinuous velocity-fields." *Computer Methods in Applied Mechanics and Engineering*, 127(1-4), 293-314.

Sloan, S. W., Krabbenhoft, K., and Lyamin, A. V. (2008). "Lower bound shakedown analysis in geotechnics." *International Association for Computer Methods and Advances in Geomechanics (IACMAG)*. City: Goa, India, pp. 328-335.

Smith, C., and Gilbert, M. (2007). "Application of discontinuity layout optimization to plane plasticity problems." *Proceedings of the Royal Society A-Mathematical Physical and Engineering Sciences*, 463(2086), 2461-2484.

Sokolovskii, V. V. (1960). *Statics of soil media*, London: Butterworths Scientific.

Sokolovskii, V. V. (1965). *Statics of granular media.*, New York: Pergamon.

Soubra, A. H. (1999). "Upper-bound solutions for bearing capacity of foundations " *Journal of Geotechnical and Geoenvironmental Engineering*, 125(4), 345-345.

Spencer. (1961). "Perturbation method in plasticity-1 plane strain of nonhomogeneous plastic solids." *Journal of the Mechanics and Physics of Solids*, 9(4), 279-288.

Strum, J. F. (1999). "SeDuMi 1.02, a Matlab toolbox for optimizaing over symmetric cones." *optimization Methods & Software*, 11-12, 625-663.

Taylor, D. W. (1948). *Fundamentals of soil mechanics*, New York: Wiley.

Terzaghi, K. (1943). *Theoretical soil mechanics*, New York: Willey.

Ukritchon, B., Whittle, A. J., and Klangvijit, C. (2003a). "Calculation of bearing capacity factor Nr using numerical limit analysis." *Journal of Geotechnical and Geoenviromental Engineering-ASCE*, 129(6), 468-474.

Ukritchon, B., Whittle, A. J., and Klangvijit, C. (2003b). "Calculations of bearing capacity factor N gamma using numerical limit analyses." *Journal of Geotechnical and Geoenvironmental Engineering*, 129(5), 468-474.

Vesic, A. S. (1973). "Analysis of ultimate loads of shallow foundations." *Journal of Soil Mechanics & Foundations Div*, 99(sm1), ASCE, 45-73.

Wang, Y. J. (2001). *Stability analysis of slopes and footings considering different dilation angles of geomaterial*, PhD thesis, The Hong Kong Polytechnic University, Hong Hong

Wang, Y. J., Yin, J. H., and Chen, Z. Y. (2001). "Calculation of bearing capacity of a strip footing using an upper bound method." *International Journal for Numerical and Analytical Methods in Geomechanics*, 25(8), 841-851.

Yamashita, M., Fujisawa, K., and Kojima, M. (2003). "Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming algorithm 6.0)." *Optimization Methods & Software*, 18(4), 491-505.

Yang, H., Shen, Z., and Wang, J. (2003a). "3D lower bound bearing capacity of smooth rectangular surface footings." *Mechanics Research Communications*, 42(5), 4058-4076.

Yang, W. H. (1993). "Large Deformation of Structures by Sequential Limit Analysis." *International Journal of Solids and Structures*, 30(7), 1001-1013.

Yang, X. L. (2007). "Upper bound limit analysis of active earth pressure with different fracture surface and nonlinear yield criterion." *Theoretical and Applied Fracture Mechanics*, 47(1), 46-56.

Yang, X. L., Guo, N. Z., Zhao, L. H., and Zou, J. F. (2007). "Influences of nonassociated flow rules on seismic bearing capacity factors of strip footing on soil slope by energy dissipation method." *Journal of Central South University of Technology*, 14(6), 842-847.

Yang, X. L., and Huang, F. (2009). "Influences of material dilatancy and pore water pressure on stability factor of shallow tunnels." *Transactions of Nonferrous Metals Society of China*, 19, S819-S823.

Yang, X. L., and Yin, J. H. (2004). "Slope Stability Analysis with Nonlinear Failure Criterion." *Journal of Engineering Mechanics-ASCE*, 130(3), 267-273.

Yang, X. L., and Yin, J. H. (2005). "Upper bound solution for ultimate bearing capacity with a modified Hoek-Brown failure criterion." *International Journal of Rock Mechanics and Mining Sciences*, 42(4), 550-560.

Yang, X. L., Yin, J. H., and Li, L. (2003b). "Influence of a nonlinear failure criterion on the bearing capacity of a strip footing resting on rock mass using a lower bound approach." *Canadian Geotechnical Journal*, 40(3), 702-707.

Yang, X. Q., Lin, P. P., and Xiang, S. H. (2009). "Comments on slope stability based on a series of Drucker-Prager failure criteria." *Rock and Soil Mechanics*, 30(4), 865-870.

Yu, H. S. (2006). *Plasticity and geotechnics*, New York: Springer-Verlag.

Yu, H. S., Houlsby, G. T., and Burd, H. J. (1993). "A novel isoparametric finite-element displacement formulation for axisymmetrical analysis of nearly incompressible materials." *International Journal for Numerical Methods in Engineering*, 36(14), 2453-2472.

Yu, H. S., Sloan, S. W., and Kleeman, P. W. (1994). "A quadratic element for upper bound limit analysis." *Engineering Computations*, 11(3), 195-212.

Zhang, X. J., and Chen, W. F. (1987). "Stability analysis of slopes with general nonlinear failure criterion." *International Journal for Numerical and Analytical Methods in Geomechanics*, 11(1), 33-50.

Zhao, S. Y., Zheng, Y. R., Liu, M. W., and Qian, K. D. (2006). "Definition and transformation of slope safety factor based on Drucker-Prager criterion." *Chinese Journal of Rock Mechanics and Engineering*, 25(S1), 2730-2734.

Zienkiewicz, O. C., and Pande, G. N. (1997). "Some useful forms of isotropic yield surface for soil and rock mechanics", G. Gudehus, (ed.) *in Finite elements in Geomechanics*. City: Chichester, pp. 179-190.

Zienkiewicz, O. C., and Taylor, R. L. (2005). *The finite element method*, Oxford ; New York: Elsevier/Butterworth-Heinemann.

Zienkiewicz, O. C., and Zhu, J. Z. (1992a). "The superconvergent patch recovery and a-posteriori error-estimates .1. the recovery technique." *International Journal for Numerical Methods in Engineering*, 33(7), 1331-1364.

Zienkiewicz, O. C., and Zhu, J. Z. (1992b). "The superconvergent patch recovery and a-posteriori error-estimates .2. error-estimates and adaptivity." *International Journal for Numerical Methods in Engineering*, 33(7), 1365-1382.

Zouain, N., Herskovits, J., Borges, L. A., and Feijoo, R. A. (1993). "An iterative algorithm for limit analysis with nonlinear yield functions." *International Journal of Solids and Structures*, 30(10), 1397-1417.

THE HONG KONG POLYTECHNIQUE UNIVERSITY

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

MANUAL OF LIBRARY OF FELA

LI Dazhong

A document for users and developers of the FELA library

December 2013

TABLE OF CONTENTS

LIST OF CODES

1 INTRODUCTION

This manual documents the programming with the Finite Element based Limit Analysis (FELA). With aims at providing the automatic adaptive mesh adaptation FELA, the code is developed on the basis of the libMesh (Kirk et al. 2006), an object-oriented C++ open source code for the finite element method. It is therefore highly recommended that concepts like mesh, finite element space, systems, etc. in the libMesh are well understood before starting programming with the FELA.

2 STRUCTURE OF THE PROGRAM

The structure of a program of the FELA is similar to that in the traditional finite element method analysis, examples of which can be found in the libMesh documentation.

2.1 Structure of the FELA

Figure 2.1 illustrates the structure of the FELA. The flow of the program starts with the input information of the mesh and material properties followed by the output of the optimized velocity field or the stress field. The following section will address each component in the flow chart that are required in the programming with the FELA.

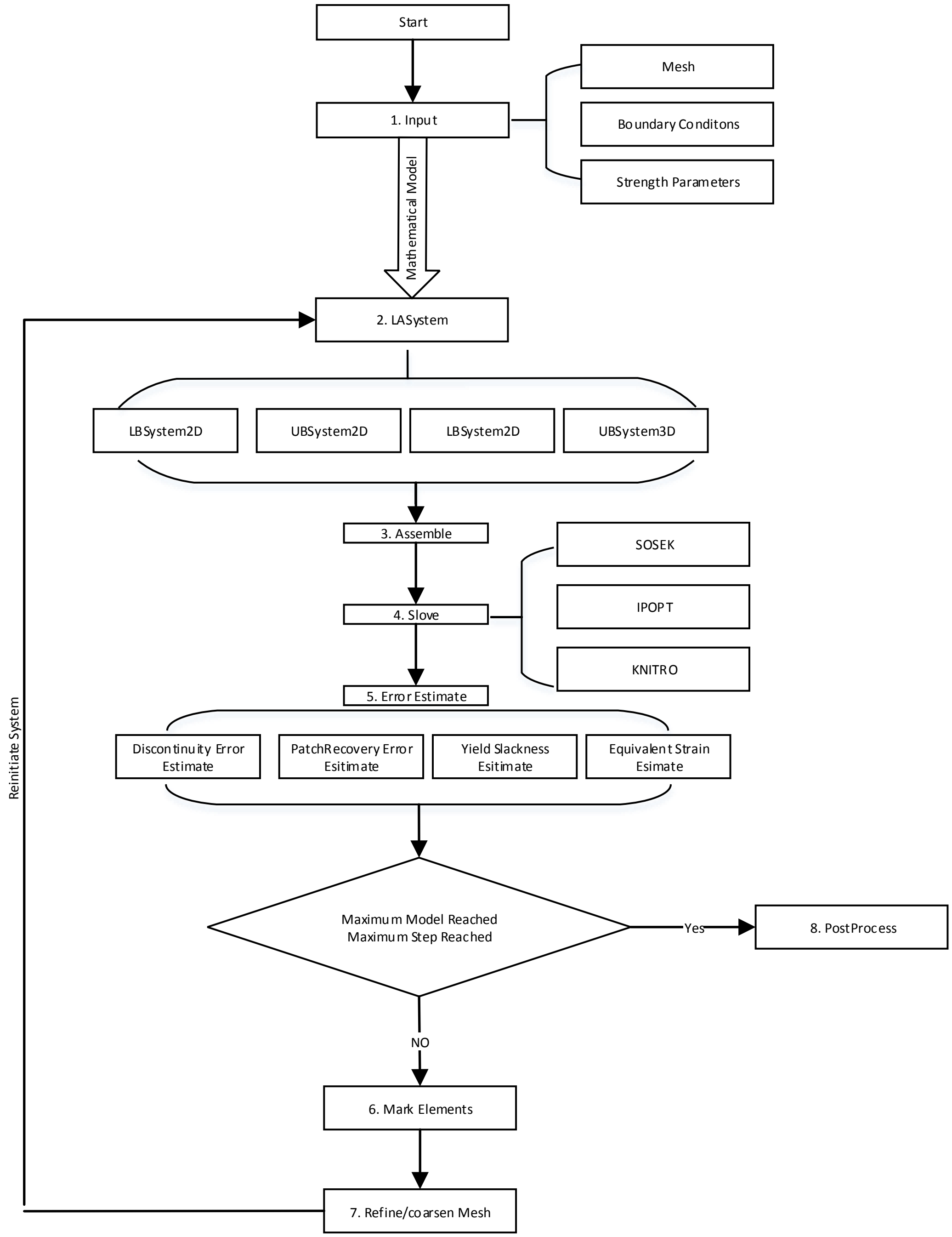


Figure 2.1 Flow Chart of the FELA

2.2 An Example of the Lower bound Plan Strain Analysis

Before heading for the details of the program, let us have a glance of a simple program of the lower bound analysis, which illustrate how the routines are organized to perform a limit analysis as an optimisation. A few classes are used and highlighted in bold in the following.

Code 2.1 Main program of 2D lower bound analysis program

```
int main(int argc, char** argv)
{
Mesh mesh(2);
input_path = ".//input//";      //set up a path of file
output_path = ".\\output\\";
mesh.read(input_path + "model.out");
NeutralIO writer(mesh);
EquationSystems es(mesh);
LBSystem2D &lb_foundation_system = es.add_system<LBSystem2D>("LB
system");
Material &material = lb_foundation_system.get_material();
material.read(input_path + "material.txt");
if(lb_foundation_system.solver_type() == MOSEK_SOLVER)
    {
        lb_foundation_system.add_variable("sm", FIRST);
        lb_foundation_system.add_variable("sx", FIRST);
        lb_foundation_system.add_variable("sy", FIRST);
    }
ErrorVector error;
KellyErrorEstimator error_estimator
error_estimator.controlling_var_type=ErrorEstimator::DISCONTINOUS_VA
RIABLE;
MeshRefinement mesh_refinment(mesh);
mesh_refinment.refine_fraction() = 0.997;
unsigned int step =15;
Optimizing_solver *solver=lb_foundation_system.get_solver();
     solver->set_params("linear_dependence_check","false");
     solver->set_params("objective_sense","min");
     solver->set_params("output_opt_file","true");
mesh_refinment.refine_fraction()=0.99;
for(unsigned int i = 0; i < step; i++)
    {
        error_on_edge.clear();
        if(lb_foundation_system.solver_type() == MOSEK_SOLVER)
        {
            MOSEK_Solver *mosek_solver =
libmesh_cast_ptr<MOSEK_Solver*>(&*(lb_foundation_system.get_solver()
));
            lb_foundation_system.assemble();
```

```
            lb_foundation_system.solve();
            writer.write(output_path + "refined.out", false);
            if(i>=15)
                break;
           NumericVector<double>

*dual_solution=&(*lb_foundation_system.dual_varialbe_yield());
            //error_estimator.use_integration=true;
            error_estimator.estimate_error(lb_foundation_system,
                                           error,

           lb_foundation_system.dual_varialbe_yield().get());

          mesh_refinment.flag_elements_by_error_fraction(error);
           mesh_refinment.embeded_refine();

           if(mesh.n_active_elem() >= 300000)
           {
               out.close();
               break;
           }
           es.reinit();
           lb_foundation_system.reinit();
        }
        else if(lb_foundation_system.solver_type() == IPOPT_SOLVER)
        {
                Optimizing_solver
*solver=lb_foundation_system.get_solver();
            solver->set_params("derivative_check", "none");
                solver->set_params("max_iter",500);
                solver->set_params("mu_strategy","adaptive");
                solver->set_params("tol",1e-9);
                solver->set_params("bound_frac",0.3);
                solver-
>set_params("hessian_approximation","limited-memory");
                //solver->set_params("nlp_scaling_method", "none");
                solver->set_params("mu_init",1e-1);
            lb_foundation_system.assemble();
            lb_foundation_system.solve();

        }
        lb_foundation_system.create_iteration_record(out, i);
    }
}
```

Table 2.1 gives a brief description of the classes used in main function of Code 2.1.

Table 2.1 A simple explanation of the classes used in Code 2.1.

Class	Description
Mesh	A class holding the data structures of nodes, connectivity
NeutralIO	An interfacing class for the input and output with Neutral file[1]
EquationSystems	EquationSystems a class managing system of equations (following the terminology of Libmesh)
LBSystem2D	lower bound 2D system, formulation occurs in the assembly of this system
Material	A material class
ErrorVector	A specialised vector data structure
KellyErrorEstimator	Error Estimator of Kelly's method
MeshRefinement	MeshRefinement
Optimizing_solver	A generalised solver, taking the mathematical problem and return the primal and dual values after the solution

2.3 Mesh

2.3.1 Input of the Mesh

The FELA is based on the finite element discretisation, and theoretically any mesh that works for FEM can theoretically be adopted in the FELA. However, due to the difficulty in dealing with the yield conditions and elimination of the hanging nodes, only 3-node triangular mesh and 4-noded tetrahedral mesh are allowed in the current version of the FELA.

The required mesh can be generated with various algorithms. It would be beneficial to incorporate the mesh generation to the current library. In the application of following examples, the commercial software Patran will be used for the mesh generation. A huge amount of the resources describing how to use this software can be found online. (http://www.mscsoftware.com/products/cae-tools/patran.aspx). However, to read the mesh

[1] Neutral file is an interfacing file from Patron, output information of the FEM including, mesh, material, boundary condition, etc. More detailed information could be accessed from this link: http://www.g-boege.de/english/makrosae/Manual/PATRAN.htm

generated in other format by other pre-process software, an interface module needs to be prepared for the interpretation of the output data. The mesh output from Patran is a Neutral file and the reading the mesh file in the program is simply done as follows.

Code 2.2Read the mesh in

```
mesh.read("model.out");
```

where ".out" is the extension recognised as the Neutral file.

It should be noted that a number of common interfacing classes have been implemented in the libmesh and some other formats are legitimate in the libmesh as shown in Code 2.3. Reference should be made to the document of the mesh generation solvers when independent mesh generators such as Triangle and Tengen are to be incorporated to the library.

Code 2.3 Format implemented

```
*.e     -- Sandia's ExodusII format\n"
*.exd   -- Sandia's ExodusII format\n"
*.gmv   -- LANL's General Mesh Viewer format\n"
*.mat   -- Matlab triangular ASCII file\n"
*.off   -- OOGL OFF surface format\n"
*.ucd   -- AVS's ASCII UCD format\n"
*.unv   -- I-deas Universal format\n"
*.vtu   -- Paraview VTK format\n"
*.xda   -- libMesh ASCII format\n"
*.xdr   -- libMesh binary format\n"
*.out   -- Neutral file format
```

2.4 Neutral file format

The model.out is a standard neutral file following fixed format, a small portion of the file takes the form as in , where the key in the first line

```
25      0      0      1      0      0      0      0      0

P3/PATRAN Neutral File from: D:\Dropbox\model\output\1.db
```

```
26      0      0      1     187     335      0      0      0
03-Jul-12   04:36:43        3.0
 1      1      0      2      0      0      0      0      0
  2.000000000E+0  1.000000000E+1  0.000000000E+0
1G      6      0      0  000000
 1      2      0      2      0      0      0      0      0
  1.000034094E+0  9.000000000E+0  0.000000000E+0
```

The first number of particular line of the file stands for a type of information, for instance, the "25" in the first line indicates the Title card, which store the location of the file; the " 26" stands for a summary card containing the information such as how many nodes and elements in the mesh. A neutral file also stores the information such as boundary condition, material information which are specified with a particular card or packet. Detailed information can be obtained in the Patran document and a small portion is extracted to explain the "model.out" is show in Code 2.4.

Code 2.4 Packet description of the Node Data (extracted form the Patran manual)

Packet 25: Title Card

```
The title card packet contains the following:
        25 ID IV KC
        TITLE

where:  ID = 0 (not applicable)
        IV = 0 (not applicable
        KC = 1
        TITLE = ICEM-PATRAN INTERFACE - Version ... -
```

Packet 26: Summary Data

```
The summary data packet contains the following:
        26 ID IV KC N1 N2 N3 N4 N5
        DATE TIME VERSION
```

```
where:  ID = 0 (not applicable)
        IV = 0 (not applicable)
        KC = 1
        N1 = number of nodes
        N2 = number of elements
        N3 = number of materials
        N4 = number of element properties
        N5 = number of coordinate frames
        DATE = dd-mm-yy
        TIME = hh:mm:ss
        VERSION = 2.5
```

Packet 01: Node Data

```
The node data packet contains the following:
        1 ID IV KC
        X Y Z
        ICF GTYPE NDF CONFIG CID PSP

where:  ID = node ID
        IV = 0 (not applicable)
        KC = number of lines in data card = 2
        X, Y, Z = X, Y, Z Cartesian coordinate of the node
        ICF = 1 (referenced)
        GTYPE = G
        NDF = 2 or 3 for 2D or 3D model respectively
        CONFIG = 0 (not applicable)
        CID = 0 i.e. global Cartesian coordinate system
        PSPC = 000000 (not used)
```

2.4.1 Reading the mesh

Reading a mesh involves interpreting fundamental information of a finite element mesh obtained from mesh generator routine, such as nodes coordinates and the connectivity. We will illustrate how to read the .out file by explaining some member functions implemented in the class `NeutralIO`.

2.4.1.1 The coordinates of the nodes

Note that when reading the information of the mesh, pointer should be passed to the Input Output (IO) class. Based on the pre-defined syntax of the output file, reading coordinates shall read through lines of the ".txt" file and parse it to collect the information. For instance,

Code 2.5 Read the node coordinates

```
void NeutralIO::read_node_data( std::ifstream &in)
{
    MeshBase &mesh = this->mesh(); // Get the mesh reference
    unsigned int node_ID;          // Read the node ID
    in >> node_ID;
    Point xyz;
    skip_lines(in, 1); // go to the next line, nothing intersting in 
the header.
    in >> xyz(0);                 //Read the coordinates x,y,z and
    in >> xyz(1);                 //Store it in a data structure
    in >> xyz(2);                 //xyz
    mesh.add_point(xyz, node_ID - 1); // Add the point to the mesh
    skip_lines(in, 1); // go to data card 2 // skip the line as we do 
not need the information of this line

    unsigned int ICF, NDF, CONFIG, CID; // Not intersted in these data
    char GTYPE, temp;                   // Maybe useful in the future
    char PSPC[7];
    in >> temp;
    ICF = atoi(&temp);
    in >> GTYPE
       >> NDF
       >> CONFIG
       >> CID
       >> PSPC;
    skip_lines(in, 1);

}
```

2.4.1.2 Read the element information and connectivity

Element information consists of the element type and the connectivity of the nodes. The interfacing routine should be able to identify element type and construct the object of element accordingly and then add it to the mesh. As we only use the triangular elements and tetrahedral elements, other types of the element are reserved for the future use if they become necessary.

Code 2.6 Read the element information

```
void NeutralIO::read_element_data( std::ifstream &in)
{
    MeshBase &mesh = this->mesh();
    Elem *elem = NULL;
    //Reading Header Card
    unsigned int element_ID,
             shape_index,   //2-bar, 3-tri, 4-quad,5-tet,7-wedge,8-hex
             KC,
             N1,             //Number of associate data values
             N2;             // ID of node in XY plane(bar only)
    in >> element_ID         // Read in the information
       >> shape_index
       >> KC
       >> N1
       >> N2;
    skip_lines(in, 1);
    //Reading Data Card 1
    int number_nodes,
        element_configureation,
        property_or_material_ID,
        congruent_element_ID;
    float
    theta1,                              //Material Orientation
angle(for bars, these values are the coordinates
    theta2,                              // of a point in X Y plane
    theta3;
    in >> number_nodes
       >> element_configureation
       >> property_or_material_ID
       >> congruent_element_ID
       >> theta1
       >> theta2
       >> theta3;
    skip_lines(in, 1);

    //Reading Data card 2  connectivity
    switch(shape_index) // Switch the shape index to construt the mesh
    {                   // accordingly
    case 3:        /* triangle*/
        switch(number_nodes)
        {
        case 3:
            elem = new Tri3;  // build a 3-noded triangular element
            break;

        case 6:
            elem = new Tri6;  //build a 6-noded triangular element
            break;

        default:
```

```
              libmesh_error();
              break;
          }

          break;

      case 5:      /* Tetrahedron */ //construt the tetrahedral mesh
      {
          switch (number_nodes)
          {
          case 4:
              elem = new Tet4; /* 4-node Tet*/
              break;

          case 10:
              elem = new Tet10; /* 10-node Tet (extra nodes on the
edges*/
              break;

          default:
              libmesh_error();
              break;
          }
      }
      break;

      case 8:
          switch(number_nodes)
          {
          case 8:
              elem = new Hex8;
          }

          break;

      default:
          std::cerr << "Unsupported Element type found!\n";
      }

      /**
       * Read the connectivity
       */
      unsigned int node;

      for (unsigned int i = 0; i < elem->n_nodes(); i++)
      {
          in >> node;
          node -= 1;             // Patran is 1-based, here we use 0-
based.
          libmesh_assert(node < mesh.n_nodes()); //make sure this is
smaller than the total nodes in the mesh
```

```
        elem->set_node(i) = mesh.node_ptr(node);
    }

    /* we use the subdomain ID for the distinction of the materials */

    subdomain_id_type &sbd_type = elem->subdomain_id();
    sbd_type = property_or_material_ID;

    /*****************************************************************/
    skip_lines(in, 1);
    elem->set_id(element_ID - 1);
    mesh.add_elem(elem);

    // Reading Data card 3
    if(N1 == 0)
    {
        return;
    }
    else
    {
        return;  // I am not really sure what can be added here.
    }
}
```

2.4.2 Discontinuities

It is always desirable to model discontinuities in the limit analysis, which is usually not featured in a conventional finite element mesh. Discontinuities can be treated by various methods, for example, the output mesh could be further processed with a stand-alone software to add nodes to the mesh. Considering discontinuities are elements with zero thickness and can be represented by two sets of nodes sharing the same set of coordinates, in this application, the mesh is kept the same as the traditional finite element mesh for the sake of generality of the input, but a different routine designed to compute the steer-vector to cater for the discontinuities at the stage of the assembly is used.

Element, nodes and mesh data structures store the pointer of the actual data rather than IDs in current code and therefore it is more convenient to renumber the nodes based on their element nodes. Elements (active elements) will be renumbered from 0 and the steer-vector will be obtained based on this rule. For instance, Figure 2.2 shows an example of the number of a mesh consisting of two elements.

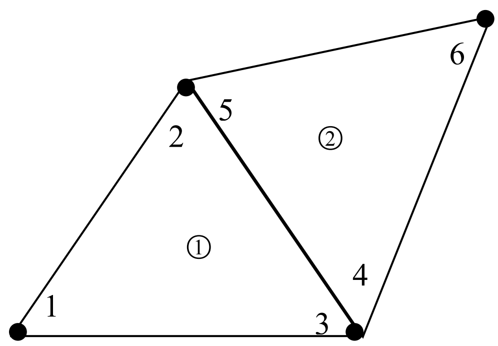


Figure 2.2 Numbering of the nodes to reflect the discontinuity

Code 2.7 Compute the dof maps for a discontinuity

```
dof_map.get_discontinous_dofs_interface(n_elems, elem, side_number,
dofs, false);
```

Within the function Code 2.7, `elem` and `side_number` are adequate to represent a discontinuity and Boolean attribute `false` is used to differentiate edges on the boundary and edge shared with adjacent element.

To navigate among the discontinuities in order to construct the equilibrium constraints along the discontinuities, a data structure has been designed. A map structure (see Code 2.8 and Figure 2.3) is used with the first entry recording the "key" of the edge, a unique number calculated from the edge while the second entry is the pair of the element and the local number that the edge corresponds to. Noting that a discontinuity requires information of the two neighboring elements but only either of the elements will be used to denote the discontinuity. If the other element is required, it could be easily retrieved by calling the module as shown below.

```
elem->neighbor(ns).
```

Code 2.8 Data structure of the discontinuities

```
std::multimap<unsigned int, std::pair<Elem*,unsigned int>>
&discontinuities
```

Key 1	*(Element 1,side_i)*
Key 2	*(Element 2,side_i)*
Key a	*(Element a,side_i)*
. . .	. . .

Figure 2.3 Data structure of discontinuities of a mesh

2.4.3 Command Line File in Patran

It might be worthwhile to prepare a command line file to generate a series of models with different degrees of density or size of a model, and sample code for generating a 2D model is given in Code 2.9.

Code 2.9 Sample code of the command line file in Patran

```
/*
 * This session file is used to create a 2D footing model
  * This model is associated with the example located as in the
filename/ full path
 */
uil_file_close.go()
STRING path[80]="D:\LACode\limitAnalysisCode\2Dfooting_lb\input\"
STRING filename[80],out_filename[80]
filename=path//"model.db"; out_filename=path//"model.out"
dump filename
dump out_filename

IF ( file_exists(filename,"") ) THEN
 file_delete(filename)
ENDIF
uil_file_new.go("",filename)

/*
 * Set the analysis type. For this model Neutral file format is
assumed.
 */
uil_pref_analysis.set_analysis_preference( "PATRAN 2 NF",
"Structural", ".out", ".out", "No Mapping" )

/**
 * Variables declearation used for controlling the geometry and mesh
in the model
 */
REAL   width_of_footing,height_of_model, depth_of_footing,size_ratio,
r_len1, r_len2;
INTEGER i_return_value, seeds_of_L,seeds_of_H,seeds_of_B,
i_seed_option
STRING s_p1[80],
s_p2[80],s_p3[80],s_p4[80],s_p5[80],s_p6[80],s_p7[80],s_p8[80],s_p9[80
], @
       extrude_vector[80]
STRING mesh_type[80], mesh_method[80];

/**
 * Variables required in the function process
 */
STRING s_output_ids[2]
STRING sv_create_grid_xyz_created_ids[VIRTUAL]
STRING sgm_create_surface__created_ids[VIRTUAL]
STRING sgm_sweep_solid_ext_created_ids[VIRTUAL]
STRING fem_create_mesh_s_nodes_created[VIRTUAL]
STRING fem_create_mesh_s_elems_created[VIRTUAL]
STRING fem_renum_node_new_ids[VIRTUAL]
INTEGER fem_create_mesh_surfa_num_elems
INTEGER fem_create_mesh_surfa_num_nodes
```

```
INTEGER max_node_ID
REAL fem_equiv_all_x_equivtol_ab
INTEGER fem_equiv_all_x_segment

/**
 * Initialization of the geometry and seeds of mesh
 * Modification should be made here if any changes in the geometry of
the model
 * or the refinement of the mesh are expected.
 */

width_of_footing=1.0                                    /* width of
the footing     */
height_of_model=10.0*width_of_footing                   /* Total
height of the mdoel*/
depth_of_footing=1.0*width_of_footin                  /* Controls the
region of refinement under the footing*/
mesh_type="Tria3"                                      /* Could be
Tria6 */
mesh_method="Paver"                                    /* or
'IsoMesh'*/

seeds_of_B=2
size_ratio=0.5
i_seed_option=20

r_len1=0.
r_len2=0.

/**
 * A square region are split into 4 subregion by nine points which
will be draw as follows
 */
s_p1="[0,0,0]";
s_p2="["//str_from_real(width_of_footing)//",0,0]"
s_p3="["//str_from_real(20.0*width_of_footing)//",0,0]" /* Total width
of model is 5 times the footing*/
s_p4="[0,"//str_from_real(height_of_model-depth_of_footing)//",0]"
s_p5="["//str_from_real(width_of_footing)//","//str_from_real(height_o
f_model-depth_of_footing)//",0]"
s_p6="["//str_from_real(20.0*width_of_footing)//","//str_from_real(hei
ght_of_model-depth_of_footing)//",0]"
s_p7="[0,"//str_from_real(height_of_model)//",0]"
s_p8="["//str_from_real(width_of_footing)//","//str_from_real(height_o
f_model)//",0]"
s_p9="["//str_from_real(20.0*width_of_footing)//","//str_from_real(hei
ght_of_model)//",0]"


i_return_value=asm_const_grid_xyz("1",s_p1,"Coord
0",sv_create_grid_xyz_created_ids)
```

```
i_return_value=asm_const_grid_xyz("2",s_p2,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("3",s_p3,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("4",s_p4,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("5",s_p5,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("6",s_p6,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("7",s_p7,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("8",s_p8,"Coord
0",sv_create_grid_xyz_created_ids)
i_return_value=asm_const_grid_xyz("9",s_p9,"Coord
0",sv_create_grid_xyz_created_ids)

i_return_value=sgm_const_surface_vertex( "1", "Point 1", "Point 2",
"Point 5", "Point 4", sgm_create_surface__created_ids )
i_return_value=sgm_const_surface_vertex( "2", "Point 2", "Point 3",
"Point 6", "Point 5", sgm_create_surface__created_ids )
i_return_value=sgm_const_surface_vertex( "3", "Point 4", "Point 5",
"Point 8", "Point 7", sgm_create_surface__created_ids )
i_return_value=sgm_const_surface_vertex( "4", "Point 5", "Point 6",
"Point 9", "Point 8", sgm_create_surface__created_ids )
/**
 * Show the labels on the screen
 */
surface_label( TRUE )
point_label( TRUE )
/**
 * Applying the boundary conditions here
 */
 loadsbcs_create2( "Loading", "Pressure", "Element Uniform", "2D",
"Static", ["Surface 3.3"],   "Geometry", "", "1.", [" ", " ", " 2."],
["", "", ""] )
 loadsbcs_create2( "Free", "Pressure", "Element Uniform", "2D",
"Static", ["Surface 4.3"],"Geometry", "", "1.", [" ", " ", " 1."],
["", "", ""] )
 loadsbcs_create2( "Extension", "Pressure", "Element Uniform", "2D",
"Static", ["Surface 2.1 2.2 4.2 1.1"],  "Geometry", "", "1.", [" ", "
", " 4."], ["", "", ""] )
 loadsbcs_create2( "Symmetry", "Pressure", "Element Uniform", "2D",
"Static", ["Surface 1.4 3.4"],"Geometry", "", "1.", [" ", " ", " 3."],
["", "", ""] )
/**
 * Create Mesh seeds
 */
mesh_seed_create( "Surface 3.3", 3, 5, 0.25, 0., 0. )  /* footing
base*/
mesh_seed_create( "Surface 4.3", 3, 10, 10., 0., 0. )  /* right hand
```

```
of footing base*/
mesh_seed_create( "Surface 4.1", 3, 1, 10., 0., 0. )  /* right hand of
footing base*/
mesh_seed_create( "Surface 3.4", 3, 5, 1., 0., 0. )    /* right most
below footing  */
mesh_seed_create( "Surface 3.2", 3, 5, 0.25, 0., 0. )  /* left footing
below        */


mesh_seed_create( "Surface 1.3", 3, 5, 1., 0., 0. )
mesh_seed_create( "Surface 1.4", 3, 10, 0.1, 0., 0. )
mesh_seed_create( "Surface 1.2", 3, 10, 0.1, 0., 0. )
mesh_seed_create( "Surface 2.3", 3, 10, 10., 0., 0. )
mesh_seed_create( "Surface 2.2", 3, 5, 1., 0., 0. )
mesh_seed_create( "Surface 4.2", 3, 1, 1., 0., 0. )
mesh_seed_create( "Surface 2.1", 3, 5, 4., 0., 0. )
mesh_seed_create( "Surface 1.1", 3, 1, 1., 0., 0. )
max_node_ID=0; /* use this to keep track of the maxmum node ID*/

fem_create_mesh_surf_4( "Paver", 49680, "Surface 1", 4, ["0.2", "0.1",
"0.2", "1.0"], mesh_type, "#", "#", "Coord 0", "Coord 0", @

fem_create_mesh_surfa_num_nodes, fem_create_mesh_surfa_num_elems,
fem_create_mesh_s_nodes_created, fem_create_mesh_s_elems_created )
max_node_ID=fem_create_mesh_surfa_num_nodes+max_node_ID
fem_create_mesh_surf_4( "Paver", 49680, "Surface 2", 4, ["0.685714",
"0.1", "0.2", "1.0"], mesh_type, "#", "#", "Coord 0", "Coord 0",@

fem_create_mesh_surfa_num_nodes, fem_create_mesh_surfa_num_elems,
fem_create_mesh_s_nodes_created, fem_create_mesh_s_elems_created )
max_node_ID=fem_create_mesh_surfa_num_nodes+max_node_ID
fem_create_mesh_surf_4( "Paver", 49680, "Surface 3", 4, ["0.2", "0.1",
"0.2", "1.0"], mesh_type, "#", "#", "Coord 0", "Coord 0",@

fem_create_mesh_surfa_num_nodes, fem_create_mesh_surfa_num_elems,
fem_create_mesh_s_nodes_created, fem_create_mesh_s_elems_created )
max_node_ID=fem_create_mesh_surfa_num_nodes+max_node_ID
fem_create_mesh_surf_4( "Paver", 49680, "Surface 4", 4, ["0.32",
"0.1", "0.2", "1.0"], mesh_type, "#", "#", "Coord 0", "Coord 0", @

fem_create_mesh_surfa_num_nodes, fem_create_mesh_surfa_num_elems,
fem_create_mesh_s_nodes_created, fem_create_mesh_s_elems_created )
max_node_ID=fem_create_mesh_surfa_num_nodes+max_node_ID
dump max_node_ID
/*
 * I do not want to see the seeds
 */
mesh_seed_display_mgr.erase(  )
/**
 * Remove duplicate nodes
 */
```

```
fem_equiv_all_group4( [" "], 0, "", 1, 1, 0.0049999999, FALSE,
fem_equiv_all_x_equivtol_ab, fem_equiv_all_x_segment )
/*
 * Renumber the node starting from 1
 */
fem_renum_node_1( "node 1:"//str_from_integer(max_node_ID), "1", 2,
fem_renum_node_new_ids )
/*
 * Export the model as neutral file
 */
IF ( file_exists(out_filename,"") ) THEN
 file_delete(filename)         /* Delete the old one first*/
ENDIF
 neutral_export2( out_filename, "Created By Stone", @
                 [TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
FALSE, TRUE, TRUE, FALSE, FALSE, @
                 TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE, FALSE, FALSE,  @
                 FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE,
FALSE], TRUE, 1, [0] )
ui_write(out_filename//" was created sucessfully!!")
```

2.5 Add explanations to these comments so that next week can understand easily

2.6 Boundary Conditions

2.6.1 Types of Boundary Conditions

Boundary conditions in the limit analysis can be classified into several categories, including

- Free boundary conditions (denoted as ID 1, no external forces applied on this segment of the boundary and it is neither the symmetric boundary nor the extension boundary)
- Boundaries on which interested loading is applied (Boundary ID 2). This boundary is related to the objective function, i.e. the pressure acting on this segment of the boundary will be integrated and subjected to the optimisation
- Symmetric boundaries (Boundary ID 3)

- Fixed or extension boundary condition (Boundary ID 4). This boundary type is required in the semi-infinite problems, for which a fixed boundary need to be specified at the far end for the upper bound problem and extension boundaries conditions for the lower bound analyses.

Each of these boundary types needs to be reflected in either the constraints or the objective functions of the eventual optimisation problem, which will be elaborated on in the following chapter.

2.6.2 Input of Boundary Conditions

Figure 2.4 and Figure 2.5 illustrate the application of the boundary condition described above. These two models are generated with Patran and the boundary ID are specified as surface pressure.

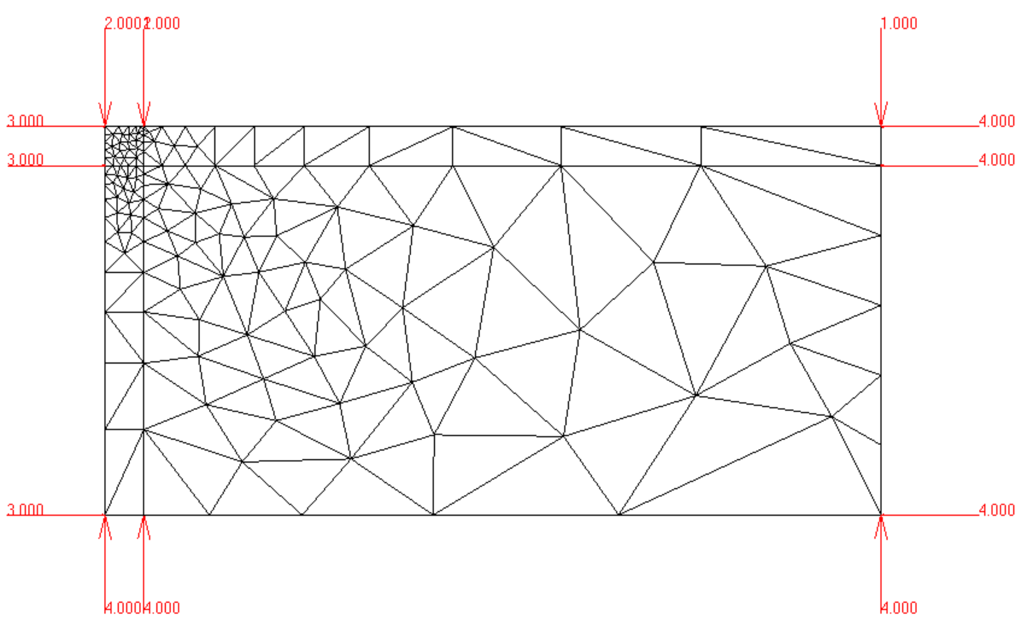


Figure 2.4 Boundary Conditions for 2D problem

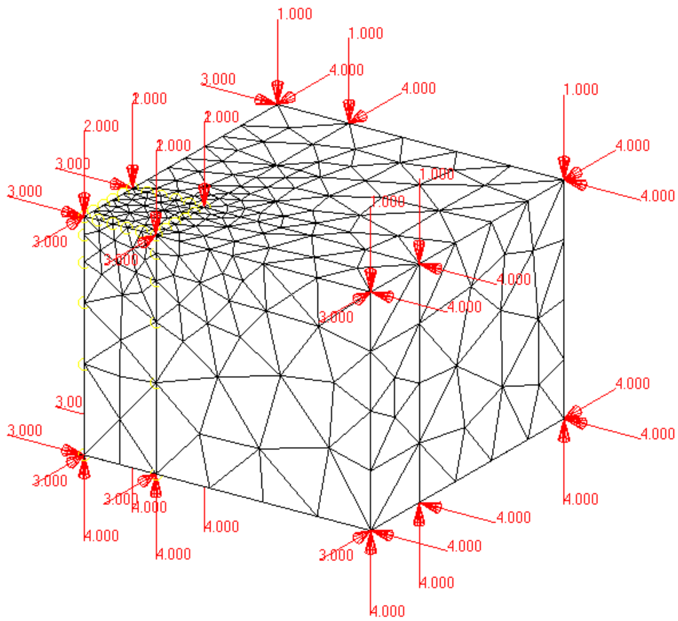


Figure 2.5 Boundary Conditions for 3D problem

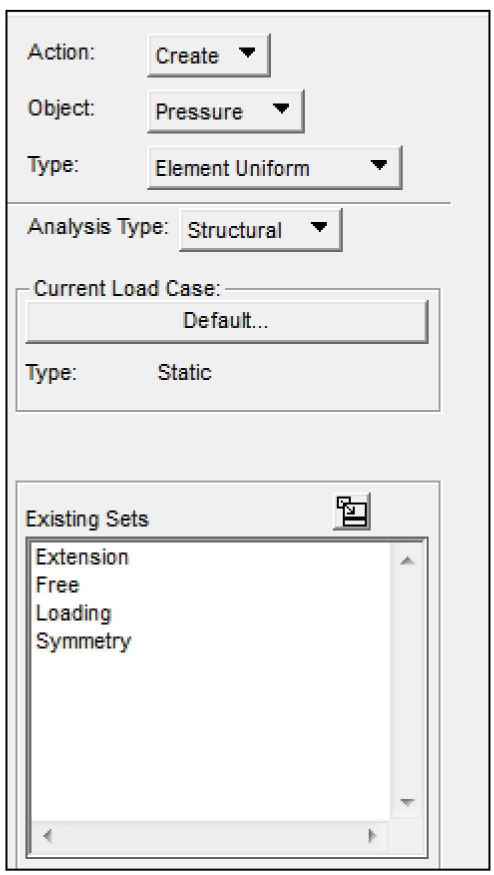


Figure 2.6 Specification of the boundary conditions

The boundary IDs can be specified in the Patran by clicking Load/BCs-> Create->Pressure as shown in Figure 2.6, and the magnitude of the pressure should read “1.0, 2.0,3.0,4.0”. More than 20 IDs are allowed and the input information will later be output in the Neutral file and read in simultaneously with the mesh by calling Code 2.10 Read in a mesh Code 2.10.

Code 2.10 Read in a mesh

```
mesh.read(“mesh.out”)
```

At the assembling stage, the stored boundary conditions will be handled seperately using a switch funcion, e.g for a lower bound analysis, the free boundary will force the normal and shear stresses to be zero on the boundary type ID 1.

Code 2.11 Switch the boundary IDs to apply boundary conditions

```
switch(boundary_ID)
        {
        case 1: //free boundary condition, shear and normal both zero
        {
            this->form_row_index(n_cons, 4, row_indices);
            _opt_task->con_matrix.add_dense_matrix(A, dofs,
row_indices);
            break;
        }
```

2.6.3 Considerations in the Mesh Adaptation

Boundary conditions stored for a problem need to be updated during the mesh adaptation. This will not be required to be considered in the application of the mesh adaptation techniques unless other mesh adaptation algorithms other than those implemented in the library are to be used. For example, combined fan zone generation with a local mesh refinement algorithm. There are two scenarios when stored boundary conditions require an update:

1. Swap nodes. In most of the mesh adaptation algorithms, it is always necessary to swap the nodes for the ease of manipulation of the mesh adaption. This means that once nodes are swapped, the boundary condition read in previously is not valid any more.
2. Mesh adaption. After a mesh adaptation, the element on the boundary might become inactive (either being split or coarsened), and the boundary information needs to be passed to its children for further analysis.

In either of these cases, the command Code 2.12 needs to be called. What is done in the function `update_element_boundary` is to compare the updated mesh with the originally stored boundary information and update the data `boundary_info` accordingly.

Code 2.12 Update boundary condition

```
_mesh.boundary_info->update_element_boundary()
```

The Class `BoundaryInfo` contains all the information relevant to the boundary conditions. It does not hold the actual boundary condition data, but can mark element faces and nodes with the IDs useful for identifying the type of boundary condition. It can also build a mesh that just includes boundary elements/faces.

3 MATERIAL PROPERTIES

In the limit analysis, relatively simple material properties are required. Only the strength parameters defining the yielding properties and the unit weight are required in the current library. In the case of the Mohr-Coulomb material, the input is only the two parameters for the strength properties, i.e. friction angle and the cohesion. For other types of the yield criteria, for example the power type yield criteria, three parameters are needed to describe the behaviour of soil. In order to deal with materials in a general manner, a class Material has been designed which allows a maximum 20 parameters input. The material information is read in via a text file as shown in Code 3.1. Explanations of the function of each code are given in the code.

Code 3.1 Material input file

```
1       // Number of materials to be read in
0 mc 2  // 0- Material ID, mc- Mohr Coulomb material and 2 – two strength parameters
1 0 0   // 1- cohesion =1.0, 0- fricion angle =0 in degree and 0- gravity = 0.0 kN/m3
```

when required, the information can be retrieved by following lines

Code 3.2 Read Material information

```
Material &material = lb_foundation_system.get_material();
material.read(input_path + "material.txt");
```

Code 3.3 Retrieve material information

```
cohesion       = material(material_ID).get_cohesion();
friction_angle = material(material_ID).get_friction_angle_rad();
```

3.1 Mathematical Model

`LASystem` is derived from the `ImplicitSystem` from libMesh (Kirk et al. 2006) corresponds to an optimisation problem associated with the limit analysis in geotechnical engineering. It is therefore highly recommended that the concepts and data structures in the libMesh are understood before coding with the FELA.

`LASystem` stores information of the primal solutions, dual solutions and some of the fundamental functions required in the limit analysis, e.g. compute the transformation matrix for SOCP analysis, finding the equivalent material parameters between different yield criteria. It also holds the information of the solver type and optimisation type.

3.2 Assemble

The purpose of the routine is to construct the global matrices that are capsulated as standard mathematical programming solvable by third-party solvers. Some basic concepts are discussed in this section and details regarding the physical aspects will be provided in the following chapter.

3.2.1 Finite Element Space

The finite element discretization follows the concept of the libMesh for its generality rather than the common form by Sloan and Lyamin(Lyamin 1999; Sloan 1989). A sample code is show in Code 3.4**Error! Reference source not found.** and detailed explanation can be found in the Documentation of libMesh.

Code 3.4 Define a finite element space

```
const DofMap &dof_map = this->get_dof_map(); // a class handling
numbering of degrees
                                             // of freedom in a mesh
    FEType fe_type = dof_map.variable_type(0);// Define a finite
element family
    AutoPtr<FEBase> fe(FEBase::build(dim, fe_type));
    QGauss qrule(dim, SECOND);               // A class handling
Gaussian Integration
    fe->attach_quadrature_rule(&qrule);

    /**
     * for the boundary integration
     */
    AutoPtr<FEBase> fe_face(FEBase::build(dim, fe_type));
    QGauss qface(dim - 1, FIFTH);
    fe_face->attach_quadrature_rule(&qface);
```

3.2.2 The Global Constraint Matrix

The global constraint matrix arising in the limit analysis is a highly sparse matrix. Unfortunately, unlike the finite element method, the matrix does not possess an apparent pattern due to the introduction of the auxiliary variables (e.g. the SOCP formulation), thus a skyline storage technique cannot be obviously applied. A straight forward storing method is adopted in the library as shown in Figure 3.1. The matrix is design as a link or a structure of Map in terms of C++. Each entry stores a pair of the row number `n` the entry value *v*. The column number and row number can be dynamically increased by calling `add_a_dense_matrix` function.

Column1	Column2	Column3	Column4	Column5	Column6	Column7	...	Column N
(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)
(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)
(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	(n,v)	
	(n,v)		(n,v)		(n,v)			
	(n,v)		(n,v)					
			(n,v)					
			(n,v)					

Figure 3.1 Data structure of the global constraint matrix

3.2.3 Objective Function and Nonlinear Constraints

In additional to the constraint matrix, the optimisation process requires provision of the objective function and nonlinear constraints. Because of the convexity of the optimisation problem, the objective function is linear and can be expressed in the similar fashion to the linear constraint matrix.

In the limit analysis, the yield function is nonlinear in terms of the optimizing variables and plays such an important role that it deserves a separate treatment. As each yield function only relates small number of stress variables for a particular yield point, the yield condition can be stored in an ordered vector.

Yield Constraint 1 →	Var11	Var12	Var13	Var14	Var15	Var16
Yield Constraint 2 →	Var21	Var22	Var23	Var24	Var25	Var26
	⋮	⋮	⋮	⋮	⋮	⋮
Yield Constraint n →	Varn1	Varn2	Varn3	Varn4	Varn5	Varn6

Figure 3.2 Data Structure for the nonlinear constraints due to the yield condition

3.2.4 The Optimisation Task

The information for optimisation is stored in a class named `OptimizingTask`, which holds the entire copy of data such as the number of variables, number of constraints, global constraint matrix, constraint on each variables etc. The pointer of this class is later passed to the solver in the function `solve` as shown in Code 3.5.

Code 3.5 Solve function

```
    virtual bool solve(OptimizingTask        *task,           // A
task holding the information of the problem
                          Real                  &obj,            //
The objective value of the problem
                              NumericVector<Number> *primal,
// The vector holding the primal variables
                          NumericVector<Number> *dual_linear,    //
The vector holding the dual variables for linear con
                              NumericVector<Number> *dual_nonlinear)
```

Code 3.6 Class Members and Member functions of an `OptimizingTask`

```
#ifndef __optmizing_task_h__
#define __optmizing_task_h__
#include "PointerSparseMatrix.h"
#include "pointer_vector.h"
#include "optimizing_solver.h"
#include "Material.h"
```

```
#include <dense_vector.h>
#include "optimizing_solver.h"

typedef std::map<unsigned int, std::pair<double, double>>  BoundType;
namespace limitanalysis
{
      /**
       * This defines the types of the yield functions
       */
      enum yield_function_type
      {
            MOHR_COULOMB = 4,
            HOEK_BROWN   = 5,
            POW_LAW      = 6
      };
}
/**
 * This class is the abstract base class for the nonlinear programming
or linear programming
 * The concrete derived class has to be implemented.
 * By default we use the Mohr_Coulomb yield criterion as the yield
function.
*/
class OptimizingTask
{
public:
    OptimizingTask();
    ~OptimizingTask();
    /**
    * Add a cone to the SOCP task, the ||y||<y_1
    * @param v   the vector holding the socp relation, with the first
being y_1, the rest
    * appearing in the norm expression,
    * In the case of the general non-linear constraints, this is just
the gearing vector.
    * the structure of the non-linear constraints is handled by the
yield function.
    */
    void add_a_cone(std::vector<unsigned int > &v);
    /**
     *
     */
    void add_a_cone(unsigned int i,
                    unsigned int j,
                    unsigned int k);
    /**
    * Add a bound to the constrains indexed by c_index, and Duplicate
bounds on the
    * constrains with the same constraint index will be omitted.
    * @param lb      lower bound of the linear constraint
    * @param ub      upper bound of the linear constraint
```

```
    * @param c_index the index number of the linear constraint.
    */
    void add_c_bound(unsigned int c_index, double lb, double ub);
    /**
    * Add a bound to the a variable indexed by v_index, and Duplicate
bounds on the
    * variable will no be stored.
    * @param lb      lower bound of the variable
    * @param ub      upper bound of the variable
    * @param v_index the index number of the variable, the subscript
of x_i for example.
    */
    void add_v_bound(unsigned int v_index, double lb, double ub);
    /**
    * if the task is initialized.
    */
    bool & is_initialized()
    {
        return _is_initialized;
    }

    /**
    * Initiate the problem with number of variables and constraints
    * @param n_vars     number of variables
    * @param v_vars     number of c    onstraints
    */
    void init(unsigned int n_vars, unsigned int n_cons);
     /**
      * Clear the memory
      */
     virtual void clear();
     /**
      * Number of variables
      */
     unsigned int &n_vars(){return _n_vars;}
     /**
      * Number of linear constraints
      */
     unsigned int &n_linear_cons(){return _n_linear_cons;}
     /**
      * Number of nonlinear constraints
      */
     unsigned int &n_nonlinear_cons(){return
_n_nonlinear_constraints;}
     /**
      * Number of total constraints
      */
     unsigned int &n_total_cons(){return _n_cons;}
     /**
      * Build a optimizing task
      */
```

```
      static OptimizingTask * build(Optimizing_solver *solver, NLP_TYPE
task_type);
      /**
       * return the reference to the Yield function type
       */
      yield_function_type &yield_type();
      /**
       * Attach a material to the optimizing task
       */
      void attach_the_material(Material & material);
      /**
       * Get a reference of the objective sense
       * True if it's minimization, false for maximization.
       */
      bool &is_mimization();
      /**
       * Output the problem to diagnose
       */
      bool print_task(std::ostream &out);
      /**
       * Get my solver
       */
      Optimizing_solver *get_solver();
      /**
       * Set the objective sense
       */
      void set_obj_sense(const std::string &objsense);
      /**
       * Return the size of a cone, e.g., the size of cone in plane
strain analysis may return 3;
       */
      unsigned int &size_of_a_cone()
            {
                  return _size_of_a_cone;
            }



public:
      /**
       * A pointer vector holding the nonzero coefficients in the
objective function that is assumed to be linear.
       */
    PointerVector<Real>                    objective;
      /**
       * A matrix holding the linear equality or inequality
constraints.
       */
    PointerSparseMatrix<Real>              con_matrix;
      /**
       * The vector that stores the nonlinear sets
```

```
        */
    std::vector<std::vector<unsigned int>> cones;
      /**
       * Boundary information of the constraints
       */
    BoundType                              c_bound;
      /**
       * Boundary information of the variables.
       */
    BoundType                              v_bound;
      /**
       * Holding the material ID corresponding to each nonlinear
constraints/yield criterion.
       */
      std::vector<unsigned int>            materialIDs;
      /**
       * Return the reference to the material
       */
      Material &material();
protected:
      /**
       * Whether the task is initialized.
       */
    bool _is_initialized;
      /**
       * Number of variables
       */
    unsigned int _n_vars;
      /**
       * Number of the total constraints
       */
    unsigned int _n_cons;
      /**
       * Number of the linear constraints in the task
       */
    unsigned int _n_linear_cons;
      /**
       * Number of the nonlinear constraints in the task.
       */
      unsigned int _n_nonlinear_constraints;
      /**
       * Type of the yield function
       */
      limitanalysis::yield_function_type _yield_function_type;
      /**
      * A pointer to the material
      */
      Material _material;
      /**
       * The optimization direction.
       */
```

```
      bool _minimize;
      /**
       * Solver
       */
      Optimizing_solver *_solver;
      /**
       * Number of variables in a particular cone
       */
      unsigned int _size_of_a_cone;

};
#endif
```

3.3 Solvers

Despite the fact that designing a specialized solver particularly for the large-scale optimisation problem would be beneficial, this requires a profound knowledge of mathematical programming as well as the computer sciences. In this library, the third-party standard convex solver will be adopted for practical problems arising from geotechnical engineering.

The information gathered in the assembling stage will be integrated in a class `OptimizingTask`, which holds the information that is necessary for an optimization process, i.e., the objective function, the global linear constraints and nonlinear constraints.

Three common convex optimisation solvers have been interfaced to the library. Mesh adaptation implies a multiple calling of the outside solver, solver internally linked in through dynamic library. Interfacing Classes `MOSEK_Solver`, `IPOPT_solver` and `Knitro_solver` are designed to interpret the input and retrieve the output. The basic features of the three solvers are presented in Table 3.1.

Table 3.1 Applicability of Solvers

Solvers	Scope	Capacities
MOSEK	☒ LP ☒ SOCP ☒ GNLP	Fast in solving SOCP , academically free
IPOPT(Kawajir et al. 2010)	☒ LP ☐ SOCP ☒ GNLP	Able to solve general nonlinear programming problem, free and open source codes
KNITRO(Byrd et al. 2006)	☒ LP ☐ SOCP ☒ GNLP	Able to solve linear and general nonlinear programming, free for a trial version and commercial codes.

Solver of a problem can be specified by using the similar codes as Code 3.7.

Code 3.7 Set the Solver for an Optimisation

```
lb_foundation_system.set_solver(MOSEK_SOLVER);
```

3.4 Error Estimate

A variety of the error estimate methods are implemented and tested in the library. Some of the methods are tailored from error estimate techniques in the finite element methods. For example, the jump error estimate which computes the jump error along the interface of two adjoining elements and along the boundary. `KellyErrorEstimator` falls into this category. Code 3.8 shows how error estimators are defined and used in the main function.

Code 3.8 Define an ErrorEstimator

```
ErrorVector error;                          // Instantiate a vector
holding the computed
                                               error.
KellyErrorEstimator error_estimator;        // Instantiate an error
estimator
error_estimator.controlling_var_type=ErrorEstimator::DISCONTINOUS_VARI
ABLE
```

Other error estimate methods can also be used; however, different error estimators do not show obvious advantages over each other. Other estimator includes `PatchRecoveryErrorEstimator`, `AdjointResidualErrorEstimator`, `DiscoutinuousErrorEstimator`, etc.

3.5 Elements Mark and Mesh Adaptation

Elements are flagged to be refined according to the error calculated and marked according to the specified rules. Elements can be marked based on the following rules:

- Number of element to be refined at each iteration
- A specified ratio of the refined element to the total elements
- Element featured with an error greater than a threshold.

Mesh refinement are achieved in two levels. One part of the adaptation is accomplished at the elemental level, i.e. a method has been implemented in an element to split itself into children elements. The other is in a class called `MeshRefinement`, which capsules all the methods related to the mesh refinement and should be instantiated if mesh adaptation is to be applied as shown in Code 3.9.

Code 3.9 Instantiate a MeshRefinement Object

```
MeshRefinement mesh_refinement(mesh);
```

3.5.1 Elements Marking

Element marking strategy can be specified by using similar code as Code 3.10. Note that there are a number of different marking strategies which have been implemented in the libmesh and borrowed in our library FELA.

Code 3.10 Selecting the Refinement Strategy

```
mesh_refinment.flag_elements_by_error_fraction(error);
```

The marking routine set the flag of refinement as “flag” and the element will later be split in the local refinement routine.

3.5.2 Mesh Adaption

Mesh coarsening is considered of less significance in comparison to the refinement in the context of the limit analysis and ONLY refinement has been implemented in the current library. It should be noted that there are existing adaptive routines implemented in the libmesh for the finite element analysis application. Unfortunately, these routines developed for the FEM includes hanging nodes that will cause difficulties in satisfying the admissibility conditions for the stress or strain rate field and therefore should not be used in our limit analysis.

Two major classes, bisection and regular refinement of mesh refinement have been implemented for the triangular mesh and tetrahedral mesh respectively. For tetrahedral mesh, regular refinement becomes more complicated and will lead to a dramatic increase in the number of elements within a few iterations and is **NOT** implemented in the current library.

Code 3.11 Refine the mesh

```
// Bisection refinement
mesh_refinement.refine_by_bisection();
// or the calling the following  for uniform refinement
```

```
// mesh_refinement.uniformly_refine();

// mesh_refinement.refine_conformingly(); regular refine.
```

Code 3.12 gives the implementation of the bisection of tetrahedral mesh. Please note that this function is called within the function `refine_by_bisection and` needs not be called by the user. However, it will be a useful sample code if the refinement technique other than bisection is desired to be developed. The algorithm in Code 3.12 bisects all elements that are marked. Element with a hanging nodes are collected and further bisected which will in turn create additional handing nodes, and this process continues until no hanging nodes are found in the mesh.

Code 3.12 Bisection of the tetrahedral mesh

```
bool MeshRefinement::_bisect_the_tetrahedral_mesh()
{
      START_LOG("_bisect_the_tetrahedral_mesh()", "MeshRefinement");
      unsigned int n_elems_flagged = 0;
      this->_initial_labeling(NULL);
      this->update_nodes_map();
      this->update_edge_map();
      MeshBase::element_iterator       it  =
_mesh.active_elements_begin();
      const MeshBase::element_iterator end =
_mesh.active_elements_end();
      for(; it!=end;++it)
      {
            Elem *elem=*it;
            if(elem->refinement_flag()== Elem::REFINE)
                  n_elems_flagged++;
      }
      std::vector<Elem*> local_copy_of_elem;
      local_copy_of_elem.reserve(n_elems_flagged);
      for(it=_mesh.active_elements_begin();it!=end;++it)
      {
            Elem* elem=*it;
            if(elem->refinement_flag()==Elem::REFINE)
            {
                  local_copy_of_elem.push_back(elem);
                  //std::cout<<elem->id()+1<<std::endl;
            }
      }
```

```
      this->_bisect_the_tet_element(local_copy_of_elem);

      this->mark_element_with_hanging_node(this-
>_edge_with_hanging_node);
      //We have handle the first refinement
      //It is required that the mesh be refined to the conformity
      while(_edge_with_hanging_node.size()!=0)
      {
            MeshBase::element_iterator
it=_mesh.active_elements_begin();
            MeshBase::element_iterator end=_mesh.active_elements_end();
            n_elems_flagged=0;
            for (;it!=end;++it)
            {
                  Elem *elem=*it;
                  if(elem->refinement_flag()==Elem::REFINE)
                  {
                        n_elems_flagged++;
                  }
            }
            local_copy_of_elem.clear();
            local_copy_of_elem.reserve(n_elems_flagged);
            for(it=_mesh.active_elements_begin();it!=end;++it)
            {
                  Elem*elem=*it;
                  if(elem->refinement_flag()==Elem::REFINE)
                  {
                        local_copy_of_elem.push_back(elem);
                  }
            }
            this->_bisect_the_tet_element(local_copy_of_elem);
            this->mark_element_with_hanging_node(this-
>_edge_with_hanging_node);

      }
      STOP_LOG("_bisect_the_tetrahedral_mesh()","MeshRefinement");
      return true;
}
```

There are other refinement strategies implemented in the library such as the embedded refinement it proves to lead irregularity of the mesh and should be not used in practice.

4 LOWER BOUND FORMULATION

4.1 Equilibrium Conditions within Elements

The static equilibrium conditions involve the construction of derivatives with respect to the stress variables, and these derivatives can be obtained in a systematic way by using the finite element class as follows.

```
const std::vector<std::vector<RealGradient> >& dphi = fe->get_dphi();
```

Where `dphi` stores the values of derivatives at the Gaussian points and the elemental constraints matrix can be filled in. Constructing the equilibrium constraints is straight forward as the entries in the matrix is only the derivatives of the shape functions, which have been calculated in the function `fe->reinit(elem)`. Code 4.1 shows how the derivatives are placed in the constraint matrix.

Code 4.1 Construct the elemental equilibrium plane strain condition

```
fe->reinit(elem);
for(unsigned int n = 0; n < elem->n_nodes(); n++)
        {
            A_sub.reposition(0, n * elem->n_nodes(), 2, n_vars);
            A_sub(0, 0) = dphi[n][1](0);
            A_sub(0, 2) = dphi[n][1](1);
            A_sub(1, 1) = dphi[n][1](1);
            A_sub(1, 2) = dphi[n][1](0);
            A_sub.right_multiply(Q);
        }
this->form_row_index(n_cons, 2, row_indices);

_opt_task->con_matrix.add_dense_matrix(A, dofs, row_indices);
_opt_task->add_c_bound(row_indices[1], -gamma, -gamma);
```

In Code 4.1, `A_sub` represents the matrix $\boldsymbol{B}_a^T$ in eq.(4.4), and it moves the elemental matrix to fill the entry. After the completion of the elemental matrix, the elemental matrix is then added onto the global matrix, which is held by an object of `_opt_task`. Note that constraint matrix only stores the homogenous part of the constraints, i.e. coefficients of variables; the right-hand side is reflected by a structure called Constraint Bound and stored separately by calling

```
_opt_task->add_c_bound(row_indices[1], -gamma, -gamma);
```

Gamma is the unit weight of the soil mass.

For 3D cases, the only difference as compared with 2D cases is that more entries need to be filled in than the corresponding 2D case (see Eqs(4.4)). Code 4.3 shows how the B matrix is constructed for the case of the 3D condition.

Code 4.2 Construct the elemental equilibrium 3D condition

```
    A_sub(0, 0) = dphi[n][qp](0);
    A_sub(0, 3) = dphi[n][qp](1);
    A_sub(0, 5) = dphi[n][qp](2);
    A_sub(1, 1) = dphi[n][qp](1);
    A_sub(1, 3) = dphi[n][qp](0);
    A_sub(1, 4) = dphi[n][qp](2);
    A_sub(2, 2) = dphi[n][qp](2);
    A_sub(2, 4) = dphi[n][qp](1);
    A_sub(2, 5) = dphi[n][qp](0);
```

4.2 Equilibrium Conditions along Discontinuities

Equilibrium conditions along discontinuities can be collected by looping over all discontinuities and added in the constraint discontinuity by discontinuity. The equilibrium condition along discontinuities virtually requires that the traction on the surface calculated from two neighboring elements equals. The function `form_surf_tra_matrix` is to calculate the transform matrix to obtain the surface traction.

Code 4.3 Construct the discontinuity conditions

```
for(; it != it_end; ++it)
        {
            Elem *elem = (*it).second.first;
            unsigned int side_number = (*it).second.second;
            libmesh_assert(elem->neighbor(side_number) != NULL);
            dof_map.get_discontinous_dofs_interface(n_elems, elem, 
side_number, dofs);
            AutoPtr<Elem> side(elem->build_side(side_number));
            A.resize(dim * side->n_nodes(), 2 * side-
>n_nodes()*n_vars);
            A.zero();
```

```
            for(unsigned int i = 0; i < side->n_nodes(); i++)
            {
                for(unsigned int j = 0; j < 2 * side->n_nodes(); j++)
                {
                    A_sub.reposition(i * dim, j * n_vars, dim,
n_vars);

                    if(i == j)
                    {
                        this->form_surf_tra_matrix(A_sub, *side);
                        A_sub.right_multiply(Q);
                    }

                    if((i + side->n_nodes()) == j)
                    {
                        this->form_surf_tra_matrix(A_sub, *side);
                        A_sub.right_multiply(Q);
                        A_sub *= -1.0;
                    }
                }
            }

this->form_row_index(n_cons, side->n_nodes()*dim, row_indices);
_opt_task->con_matrix.add_dense_matrix(A, dofs, row_indices);

  // now we free the space of discontinuities
discontinuities.clear();
```

In the 3D analysis, the `form_surf_tra_matrix` has the definition of Code 4.4.

Code 4.4 Surface traction matrix

```
void Lb3dSystem::form_surf_tra_matrix( DenseSubMatrix<Real> &A_sub,
Elem &side )
{
    Point normal = side.get_normal();
    libmesh_assert(fabs(normal.size() - 1.0) < 1.0e-6);
    DenseMatrix<Real> beta = this->construct_rotation_matrix(normal,
_dim);

    for(unsigned int i = 0; i < 3; i++)
        for(unsigned int j = 0; j < 6; j++)
        {
            if(j < 3)
                A_sub(i, j) = beta(i, j) * beta(2, j);
            else if(j == 3)
                A_sub(i, j) = beta(2, 1) * beta(i, 0) + beta(2, 0) *
beta(i, 1);
            else if(j == 4)
```

```
                A_sub(i, j) = beta(2, 2) * beta(i, 1) + beta(2, 1) *
beta(i, 2);
            else if(j == 5)
                A_sub(i, j) = beta(2, 2) * beta(i, 0) + beta(2, 0) *
beta(i, 2);
        }

    //      A_sub.p
```

4.3 Boundary Conditions

In the limit analysis, different types of boundary condition need to be addressed in the constraints or the objective functions. The information of the boundary conditions is stored in `mesh.boundary_info`.

```
    BoundarySide_iterator bd_side_iter = mesh.boundary_info-
>boundary_side_id.begin();
    BoundarySide_iterator end_bd_side_iter = mesh.boundary_info-
>boundary_side_id.end();
for(; bd_side_iter != end_bd_side_iter; ++bd_side_iter)
    {
        Elem *elem = (Elem*)bd_side_iter->first;
        unsigned int side_number = bd_side_iter->second.first;
        unsigned int boundary_ID = bd_side_iter->second.second;
        AutoPtr<Elem> side(elem->build_side(side_number));

        unsigned int vars_per_elem = side->n_nodes() * n_vars;
        unsigned int dims_per_elem = side->n_nodes() * dim;
        A.resize(dims_per_elem, vars_per_elem);
        A.zero();
        dof_map.get_discontinous_dofs_interface(n_elems, elem,
side_number, dofs, false);

        for(unsigned int i = 0; i < side->n_nodes(); i++)
        {
            for(unsigned int j = 0; j < side->n_nodes(); j++)
            {
                if(i != j)    // only diagonal has non-zeros
                    continue;

                A_sub.reposition(dim * i, n_vars * j, dim, n_vars);
                this->form_surf_tra_matrix(A_sub, *side);
                A_sub.right_multiply(Q);
                //A_sub.print(std::cout);
            }
```

```
        }
```

4.4 Considerations in the second order cone programming

4.4.1 Additional Linear Constraints for Auxillary Variables

To formulate the Mohr-Coulomb yield criterion in a standard second order cone programming, variables are rotated and new variables need to be introduced. The last row in the formulation Eq. (4.50) can be realised by Code 4.5. The function `add_an_entry` adds a coefficient of the linear constraints in the global matrix and the right-hand side is stored by the function `add_c_bound`.

Code 4.5 Adding intermediate variables and relations in the SOCP formulation

```
for(unsigned int i = 0; i < elem->n_nodes(); i++) /** iterates node to
add intermediate variables and cones***/
        {
                if(this->task_type()==SOCP)
                {
                     _opt_task->con_matrix.add_an_entry(1.0,
num_stress_v_soil + 3 * elem_ID + i, n_cons);
                     _opt_task->con_matrix.add_an_entry(a, elem_ID *
ndof + i * n_vars, n_cons);
                     _opt_task->add_c_bound(n_cons, k, k);
                     n_cons++;
                     _opt_task->add_a_cone((num_stress_v_soil +
elem_ID * elem->n_nodes() + i),   /** z   **/
                                 (ndof * elem_ID + n_vars * i + 1),
/** sxx**/
                                 (ndof * elem_ID + n_vars * i + 2)
/** sxy**/
                                );
                }
                else if(this->task_type()==GNLP)
                {
                     _opt_task->add_a_cone(ndof*elem_ID+n_vars*i,
                                      ndof*elem_ID+n_vars*i+1,
                                      ndof*elem_ID+n_vars*i+2);
                     // In contrast to the SOCP where the material
properties are
                     // addressed already in the formulation, the GNLP
need to pass the material
                     // information to the task and will be retrieved
later in the solution stage.
```

```
                          _opt_task->materialIDs.push_back(material_ID);
                     }

            }
    }
```

4.4.2 Rotation Matrix to Form the SOCP

The tranform matrix $\boldsymbol{T}^{SOCP}$ in the formulation (4.50) is required to cast the optimisation problem in the standard second order cone programming. This matrix is called Q matrix in the library and should be applied. The matrix is obtained by calling the funcion `get_socp_variable_transform_matrix();`

```
Q = this->get_socp_variable_transform_matrix();
```

In the 3D case, matrix Q takes the following value as in Eq. (3.53).

Code 4.6 SOCP variable transform matrix

```
DenseMatrix<Number> Lb3dSystem::get_socp_varialbe_transform_matrix()
{
    DenseMatrix<Number> Q(6, 6);
    Real c0p3 = -1.0 / 3.0;
    Q.zero();
    Q(0, 0) = 1.0;
    Q(0, 1) = 1.0;
    Q(0, 2) = c0p3 * sqrt(3.0);
    Q(1, 0) = 1.0;
    Q(1, 2) = -2.0 * c0p3 * sqrt(3.0);
    Q(2, 0) = 1.0;
    Q(2, 1) = -1.0;
    Q(2, 2) = c0p3 * sqrt(3.0);
    Q(3, 3) = 1.0;
    Q(4, 4) = 1.0;
    Q(5, 5) = 1.0;
    return Q;
}
```

The matrix is applied to the stress variable by being multiplied to the equilibrium equation using the following line like `A_sub.right_multiply(Q)` as show in Code 4.3. If the general nonlinear formulation is to be used, it can be conveniently achieved by setting Q to identity matrix.

5 UPPER BOUND FORMULATION

5.1 Primal Formulation of the Upper Bound Analysis

The primal formulation of the upper bound analysis refers to the direct formulation in terms of the velocities and the Lagrangian multipliers, i.e. the formulation of (5.42). This formulation requires an explicit expression of the energy dissipation function. This may not be an easy task for a general nonlinear yield criterion and we will discuss the SOCP formulation of the Mohr-Coulomb yield criterion for which the expression of the energy dissipation is known.

The major difference between and upper bound formulation and lower bound formulation in terms of coding is the integration. As has been noted in the lower bound formulation, no integration is involved in computing the constraint matrices while in the upper bound formulation, integration is needed to be performed since the energy dissipation over the domain is to be calculated.

5.1.1 Objective Function

Applying the numerical integration using the libMesh is simple. Code 5.1 gives an example of calculating work done due to the weight of an element $W_{ext}^{0}(\mathbf{u})$ in eq.(5.1). `qp` denotes the quadrature points used in the integration.

Code 5.1 Integration over the domain to calculate the weight

```
    elem_energy_vector.resize(n_v_dofs);

        for(unsigned int qp = 0; qp < qrule.n_points(); qp++)
```

```
        {
            for(unsigned int i = 0; i < phi.size(); i++)
            {
                elem_energy_vector(i) += gamma * JxW[qp] * phi[i][qp];
            }
        }

        this->get_opt_task()-
>objective.add_sub_vector(elem_energy_vector, dof_indices_v);

    }
```

5.1.2 Strain Velocity Relationship

Strain rate is the derivatives of the velocity and these two are related by the matrix B calculated in Code 5.2.

Code 5.2 B matrix relating the strain rate to the velocity

```
 Bm_u.reposition(u_var * n_u_dofs, u_var * n_u_dofs, 1, n_u_dofs);
 Bm_v.reposition(0, v_var * n_v_dofs, 1, n_v_dofs);
 Bd_u.reposition(1, 0, 2, n_u_dofs);
 Bd_v.reposition(1, v_var * n_v_dofs, 2, n_v_dofs);
for (unsigned int qp = 0; qp < elem->n_vertices(); qp++)
        {
            //Assemble B  the coefficients of  u and v
            row_indices[0] = n_cons;
            row_indices[1] = n_cons + 1;
            row_indices[2] = n_cons + 2;

            B.zero();


/*********************************************************************
***/
            /* B=[Bm,Bd]^T
            /* B*/

/*********************************************************************
***/
            for(unsigned int j = 0; j < n_u_dofs; j++)
            {
                Bm_u(0, j) = dphi[j][qp](0);
                Bm_v(0, j) = dphi[j][qp](1);

                Bd_u(0, j) = dphi[j][qp](0);
```

```
                    Bd_u(1, j) = dphi[j][qp](1);
                    Bd_v(0, j) = -dphi[j][qp](1);
                    Bd_v(1, j) = dphi[j][qp](0);
                }

                //B.print(B_matrix);
                (this->get_opt_task()->con_matrix).add_dense_matrix(B,
dof_indices, row_indices);
```

5.1.3 Cone Constraints

The conic constraints $\mathbf{y}_{h_i}$??? C_i is added to the optimisation problem by simply recording the ordered sequence of the variable ID, in Code 5.3 the `A_col_index`

Code 5.3 Adding the conic constraints in the plane strain upper bound formulationa

```
    //Assemble A , putting lambda, and strain terms in the matrix.

                std::vector<unsigned int> A_col_index(3);

                A_col_index[0] = n_nodes * 2 + n_cons / 3;
                A_col_index[1] = A_col_index[0] + 3 * n_elem;
                A_col_index[2] = A_col_index[1] + 3 * n_elem;

                this->get_opt_task()->con_matrix.add_dense_matrix(A,
A_col_index, row_indices); /* Lambda, e11,e12*/

                // putting the SOC relation
                this->get_opt_task()->cones.push_back(A_col_index);
```

Implementation of other constraint should be simple and will not be discussed.

5.2 Dual Formulation of the Upper Bound Analysis

The dual formulation resembles the lower bound formulation by observing the constraint matrix (5.52). The only difference is that integration has been applied on the derivatives and it could be achieved by similar technique as Code 5.1.

6 POST PROCESS OF THE PROGRAMS

Independent development of a post-processing of any numerical program, including visualization of the stress and displacement/velocity fields requires a huge amount of graphic programming. There are toolkits for such purpose, for example the open source code VTK (http://www.vtk.org/) can be integrated for the visualization.

A huge amount of work regarding post-processing is expected from the present work. To minimize the work, Patran has been adopted for the visualization of the computed results, which are output in a format as the Neutral file.

As the primal and dual solutions are stored in the System, the current functions only output the Neutral format that can be read by Patran.

Two files need to be provided which are:

Code 6.1 Output the stresses

```
system.write_stress(output_path+"stress");
```

Code 6.2 Output the displacement file

```
system.write_displacement(output_path+"velocity");
```

To output a format the can be read by Patran a template file need also be prepared, and the syntax of such a file have been well-documented in the Patran User's Manual. A stress template used in this library is given as in

Code 6.3 Template file for the stress output

```
/* p2nf_els.res_tmpl1 */
/* PATRAN 2.5 results file template for PATRAN 2 NF .els files */
KEYLOC = 0
TYPE = tensor
COLUMN = 1, 2, 3, 4, 5, 6
```

```
PRI = Stress tensor 1
SEC = Components
CTYPE = elem
TYPE = scalar
COLUMN = 7
PRI = Generic Scalar 1
SEC = Column 7
TYPE = END
```

This process can be simplified by preparing a command line file using Patran, and Code 6.4 gives an example of a command line file for import “text.out” and “stress.els” files into Patran.

Code 6.4 Command line file controls the output

```
/**
 * This session file controls the output of the file
 */
uil_file_close.go(  )
integer flag
string
db_filename[80],out_filename[80],template_name[100],result_filename[10
0],path[100]
path="D:\LACode\limitAnalysisCode\Lb3D_example1\output\"

db_filename="D:\new.db"
out_filename=path//"test.out"
result_filename=path//"stress.els"
flag=0

switch (flag)
      case (0)

      template_name="D:\LACode\limitAnalysisCode\limitAnalysisCode\Patr
an_output_template\limit_analysis.res_tmpl"
      case (1)

      template_name="D:\LACode\limitAnalysisCode\limitAnalysisCode\Patr
an_output_template\limit_analysis_nod.res_tmpl"
end switch



if(file_exits(db_filename,"")) then
      file_delete(db_filename)
end if
uil_file_new.go("",db_filename)
neutold_import_neutral( out_filename,
```

```
@
                        "default_group", [TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, FALSE,           @
                        TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE,
TRUE, TRUE, TRUE, TRUE, FALSE, FALSE,                             @
                        FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE,
FALSE, FALSE],                                                @
                        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],   @
                        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],   @
                        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] )
uil_toolbar.hidden_line(  )
ga_view_aa_set( 0., 0., 0. )
switch(flag)
      case (0)
            resold_import_results(result_filename,"E", 1E-006,
template_name)
  case (1)
      resold_import_results(result_filename,"N", 1E-006, template_name)
end switch
```

7 EXTENSION OF THE LIBRARY

7.1 Extension to Other Isotropic Yield Criteria

As discussed in Section 4.6 of the thesis, isotropic yield criteria can be dealt with by a standard template. Different isotropic yield criteria can be implemented by supplying particular expressions for the three constants C_1, C_2, C_3, for example, the Hoek-Brown yield criterion and the power-type nonlinear yield criterion.

A virtual class `YieldCriterion` is defined to provide basic functions to facilitate the extension to other isotropic yield criteria in the FELA. Code 7.1 shows the Class members and member functions of `YieldCriterion`. Eq.(4.2) and Eq.(4.3) have been implemented leaving the coefficients to be supplied in the pure virtual functions.

Code 7.1 The header file of the yield function

```
#ifndef   __yield_criterion__h__
#define   __yield_criterion__h__

#include <libmesh.h>
#include <vector>
#include <dense_matrix.h>
#include <dense_vector.h>
#include <LASystem.h>

// Some constants to speed up the calculation
const double C00001=1.0;
const double DTINY =1.0e-16;
const double CP3333= 0.3333333333333333;
const double C00IR3= 0.5773502691896258;
const double CP6666= 0.6666666666666667;
const double C1P333= 1.3333333333333333;
const double CP1666= 0.1666666666666667;
const double CP8660= 0.8660254037844386;
const double c00004= 4.0;
const double C00002= 2.0;
const double CP5000= 0.5;
const double CTHETA= 0.2598076211353316;
```

```
namespace limitanalysis
{
      enum ANALYZE_TYPE
      {
            PLANE_STRAIN        =0,
            PLANE_STRESS        =1,
            THREE_DIMENSION     =2
      };
}
using namespace limitanalysis;
class  YieldCriterion
{
public:
      /**
       * Default constructor
       */
      YieldCriterion();
      /**
       * Construct the yield function with stress vector and dimension
       * For 2D case only 3 independent stress components are necessary
       * For 3D case only 6 independent stress components are necessary
       */
      YieldCriterion(std::vector<Real> sigma,unsigned int dim){};
      /**
       * construct the yield function with only the dimension
       */
      YieldCriterion(unsigned int dim,ANALYZE_TYPE my_type);
      /**
       * Deconstructor
       */
      ~YieldCriterion();


public:
      /**
       * Get the value of the yield function
       */
      virtual Real get_function_value()=0;
      /**
       * Compute the Hessian and gradient for the yield criterion
       */
      virtual bool compute_gradient_Hessian()=0;
      /**
       * Compute the first invariant, INV1, of stress deviation tensor
and the
       * deviatoric stresses S, sigma_m
       */
       Real get_inv1();
      /**
       * Compute the second invariant, INV2, of stress deviation tensor
and
```

```
        * _inv2 = sigma_bar*sigma_bar
        * _sqinv2=sigma_bar
        */
        Real get_inv2();
       /**
        * Compute the third invariant, INV3, of the stress deviation
tensor
        * J3
        */
        Real get_inv3();

       /**
        * Return a reference to the parameter
        */
       DenseVector<Real> & param(){return _param;}
       /**
        * Set the stress vector
        */
       virtual void set_stress(DenseVector<Real> &stress);
       /**
        * Update the stress, and calculate the stress invariant used for
the latter calculation
        */
       virtual bool update(DenseVector<Real> sigma);
       /**
        * Return the gradient with respect to the stress vector
        */
       const DenseVector<Real> & get_my_gradient();
       /**
        * Return the Hessian with respect to the stress vector;
        */
       const DenseMatrix<Real> & get_my_hessian();
       /**
        * Change the parameters of the yield function
        * @param i     i-th parameter
        * @val         value of the i-th parameter
        */
       bool change_strength_parameter(unsigned int i,Real val);
       /**
        * Get a reference of the parameter
        */
       DenseVector<Real> & parameter();
       /**
        * Dimension of the yield criterion
        */
       unsigned int dimension(){return _dim;}
       /**
        * Return whether the yield function has been initialized.
        */
       bool is_initialized(){return _is_initialized;}
       /**
```

```
        * Initialize the yield function
        */
       bool init(unsigned int dim, ANALYZE_TYPE my_type);
       /**
        * Gradient calculated
        */
       bool &grad_calculated();
       /**
        * Hessian calculated
        */
       bool &Hess_calculated();



protected:
       /**
        * Number of independent stress components
        */
       unsigned int vnod;
       /**
        * Dimension of the problem
        */
       unsigned int _dim;
       /**
        * first invariant,second, third, and square root of the second
        **/
       Real _inv1,_inv2,_inv3,_sqinv2,_theta;
       /**
        * Stress vector
        */
       DenseVector<Real> X;
       /**
        * devS---------deviatoric stress
        * stscof-------stress coefficient
        */
       DenseVector<Real> devS, stscof;
       /**
        * Parameters used in the yield function, e.g. c and phi for the
mohr_coulomb yield
        * criterion , we set by default in total 10 parameters are
allowed for a
        * yield criterion
        */
       DenseVector<Real> _param;
       /**
        * Index of the non_zero component
        * Index of the independent stress component
        * not clear
        */
       DenseVector<unsigned int> stsnzr,stsind,stscol;
       /**
```

```
       * Analysis type, to distinguish the following case
       * 1------Plane strain
       * 2------Plane stress
       * 3------Full three-dimensional analysis
       */
      ANALYZE_TYPE analysis_type;
      /**
       * Return the sign of a value
       */
      Real sign(Real val);
      /**
       * H   --------     The Hessian Matrix
       * D2IV3-------     second derivative with respect to J3
       * D2IV2-------     second derivative with sigma_bar
       * DD   -------     Intermediate matrix of full Hessian
       */
      DenseMatrix<Real> H, D2INV3, D2INV2,DD;
      /**
       * V ---------     The gradient
       * D ---------     Full gradient(differ V only in 2D case)
       * D1INV1-----     First derivative of sigma_m with respect to
stress vector
       * D1INV2-----     First derivative of sigma_bar with respect to
stress vector
       * D1INV3-----     First derivative of J3 with respect to stress
vector
       */
      DenseVector<Real> D,V, D1INV1,D1INV2,D1INV3,W;
      /**
       * Compute first derivatives, DIINV1,of the first stress
invariant
       */
      const DenseVector<Real> & fd1I1();
      /**
      * Compute first derivatives,  D1INV2,of the second invariant of
the stress deviation tensor
      */
      const DenseVector<Real> & fd1I2();
      /**
      * Compute derivatives of invariant INV2 with respect to stress
and store them in D2INV2
      */
      const DenseVector<Real> & fd1I3();
      /**
      * Compute second derivatives, D2INV2, of the second invariant of
the stress
      * deviation tensor (upper triangle part only)
      */
      const DenseMatrix<Real>& fd2I2();
      /**
      * Compute second derivative, D2INV3, of the third invariant of
```

```
the stress
      * deviation tensor (upper triangle part only)
      */
      const DenseMatrix<Real> & fd2I3();
      /**
      * Fill D1INV, the vector of first derivatives of the invariants
of the
      * stress deviation tensor for VNOD variables, using full vector
of
      * derivatives W and relation between stress components
      * @param  full_vector   the vector with respect to the full
stress vector
      * @param  actual_vector the vector with respect to the actual
number of stress variables
      */
      bool fildvc(DenseVector<Real> &full_vector,DenseVector<Real>
&actual_vector);
      /**
      * Fill out D2INV, the matrix of second derivatives of the
invariants of
      * the stress deviation tensor for VNOD variables, using full
Jacobian matrix
      * MA and relation between stress components
      */
      bool fildma(DenseMatrix<Real> &MA, DenseMatrix<Real> &D2INV);
      /**
      * Set gradient and Hessian for the case of zero deviatoric
stresses
      */
      virtual bool zero_dev();
      /**
       * Update the yield function
       */
      void update_yield_function(DenseVector<Real> &stress);
      /**
       * whether initialized, a yield function must be initialized with
       * the strength parameter before using.
       */
      bool _is_initialized;
      /**
       * gradient calculated
       */
      bool _is_gradient_calculated;
      /**
       * Hessian calculated
       */
      bool _is_hessian_calculated;
      /**
      * Get the Lode angle
      */
      Real get_theta();
```

```
	/**
	 * Hold the string parameters, that might be used for controlling of the
	 */

};
#endif// End of the yield criterion definition.
```

A concrete example of the implementation of the yield criterion is shown in Code 7.2, and some of the notation in the codes follows that by Lyamin (1999).

Code 7.2 Header file of the Mohr-Coulomb yield criterion

```
#ifndef  __mohr_coulomb__h__
#define  __mohr_coulomb__h__

#include "yield_criteria.h"
class MohrCoulomb :public YieldCriterion
{
public:
	MohrCoulomb(){};
	MohrCoulomb(unsigned int dim, ANALYZE_TYPE my_type=PLANE_STRAIN);
	MohrCoulomb(unsigned int dim, Real c,Real phi, ANALYZE_TYPE my_type=PLANE_STRAIN);
	Real get_function_value();  // Provide the function value
	bool compute_gradient_Hessian();//Calculate the Gradient and Hessian.
	typedef YieldCriterion parent;
	Real &cohesion();
	Real &phi();
	bool update(DenseVector<Real> sigma);
	void init(Real c, Real phi,Real attran=29.5, Real beta=0.00005);
private:
	/**
	 * The cutting angle of the hyperbolic approximation to round the
	 * Mohr_Coulomb yield surface in deviatoric plane.
	 * The default value is set to be 29.5;
	 */
	Real theta_T;
	Real _hyper;
	Real A1, A2,B1,B2,C1,C2;
	unsigned int _smoothing_type;
};

#endif
```

7.2 Incorporate Other Optimisation Solvers

Three solvers have been integrated and tested in the current library, and it is admitted that more powerful solvers are surely to come up in the future. We will discuss one interfacing with MOSEK as an example showing how it might be done in the FELA.

It should be noted that the third party solver can be used by output the optimisation problem in certain format and run the optimisation solver outside to find a solution. This approach of interfacing becomes cumbersome when applying the mesh adaption technique as the solver needs to run a number of times during the solution.

Interfacing with a third-party solver requires passing the information collected in the assemble function to API provided by the Solver itself. As most solvers provide APIs in C or C++, it would be not difficult provided that the library and header file are included in the project. A interfacing class `Optimizing_solver` has been prepared and Code 7.3 gives the header of the class. To incorporate a solver in the library, a new class needs to be derived from `Optimizing_solver` and the pure virtual function `solve` should be implemented. Code 7.4 shows and example of interfacing with MOSEK.

Code 7.3 Header of the Class `Optimizing_solver`

```
#ifndef __optmizing_solver_h__
#define __optmizing_solver_h__
#include <libmesh.h>
#include <dense_vector.h>
#include <numeric_vector.h>
class OptimizingTask;
/**
 * This is an interface class of the optimising solver to the
optimization problem arising in the limit analysis
namespace limitanalysis
{
      enum NLP_TYPE
      {
```

```
            LINEAR_PROG    = 0,    /* Linear programming*/
            SOCP = 1,              /* Second order cone programming*/
            QP   = 2,              /* Quadratic Programming*/
            SDP  = 3,              /* Semidefinite programming*/
            GNLP = 4,              /* General nonlinear programming*/
      };
}
using namespace limitanalysis;

class Optimizing_solver
{
public:

    Optimizing_solver();
    ~Optimizing_solver();
    /**
     * Solve the problem
     */
    virtual bool solve(OptimizingTask        *task,           // A
task holding the information of the problem
                          Real                   &obj,             //
The objective value of the problem
                              NumericVector<Number> *primal,
// The vector holding the primal variables
                          NumericVector<Number> *dual_linear,    //
The vector holding the dual variables for linear con
                              NumericVector<Number> *dual_nonlinear) =
0;
    unsigned int n_var()
    {
        return _num_vars;
    }
    /**
     * Build a \p Optimizing solver package specified by \p type
     */
    static Optimizing_solver * build(const SolverPackage type =
MOSEK_SOLVER);
    /**
     * The number of constraints
     */
     unsigned int n_con()
     {
           return _num_cons;
     }
    bool &is_initialized()
    {
        return _is_initialized;
    }
    /**
     * Solver type
     */
```

```
    SolverPackage &solver_type()
    {
        return _solver_type;
    }
    /**
     * Attach a optimizing task to the solver
     */
    void attach_a_task(OptimizingTask *task);
    /**
     * return the reference to the task
     */
    OptimizingTask *get_task()
    {
        return _opt_task;
    }
    /**
     * Close the solver
     */
    virtual void close();
     /**
      * Return the type of the problem
      */
     NLP_TYPE &task_type(){return _task_type;}
     /**
     * Get the primal variables
     */
     DenseVector<Real> &get_primal_variales();
     /**
     * Get the dual variables corresponding to linear constraints
     */
     DenseVector<Real> &get_dual_linear();
     /**
     * Get the dual variables corresponding to the nonlinear
constraints
     */
     DenseVector<Real> &get_dual_nonlinear();
     /**
     * Get the objective value
     */
     Real &get_obj_value();
     /**
      * Set the string parameters
      */
     void set_params(const std::string &str_name,const std::string
&str_var);
     /**
      * Set numerical parameters
      */
     void set_params(const std::string &str_name,double val);
     /**
      * Set Integer value
```

```
         */
        void set_params(const std::string &str_name,int val);
        /**
         * Indicate whether a warm start be used
         */
        bool &warm_start()
        {
                return _warm_start;
        }


protected:
    /**
     * Number of variables in the optimization problem
     */
    unsigned int _num_vars;
    /**
     * Number of constraints in the optimization problem
     */
    unsigned int _num_cons;
    /**
     * Number of linear constraints in the optimization problem
     */
    unsigned int _num_linear_cons;
    bool _is_initialized;
    /**
     * Number of non-linear constraints
     */
    unsigned int _num_nonlinear_con;
    /**
     * Solver type
     */
    SolverPackage _solver_type;
    /**
     * type of the programming
     */
    NLP_TYPE   _task_type;
    /**
     * An optimizing task
     */
    OptimizingTask *_opt_task;
      /**
      * The primal variables
      */
      DenseVector<Real> _primal_solution;
      /**
      * Dual variables corresponding to the linear constraints
      */
      DenseVector<Real> _dual_varaible_linear;
      /**
      * Dual variables corresponding to the nonlinear constraints/yield
```

```
function
      */
      DenseVector<Real> _dual_variable_nonlinear;
      /**
      * The objective value;
      */
      Real _obj;
      std::map<std::string,std::string> _str_param;
      std::map<std::string,double> _numerical_param;
      std::map<std::string,int>     _integer_param;
      /**
       * Indicate whether a user specified initial point will be used
       */
      bool _warm_start;
};
#endif
```

Code 7.4 An interfacing function for MOSEK (2010).

```
bool MOSEK_Solver::solve( OptimizingTask        *task,         /* A
task holding the information of the problem */
                          Real                   &obj,         /* the
objective value of the problem*/
                          NumericVector<Number> *primal,       /* The
vector holding the primal variables */
                          NumericVector<Number> *dual_linear, /* The
vector holding the dual variables for linear con */
                          NumericVector<Number> *dual_nonlinear )
{
    this->_opt_task = task;
    START_LOG("MOSEK_Solution", "Solve"); // A Timer
    this->convert();
    this->set_parameters();
    //assert(_is_initialized==true);
    // we check if the linear dependency check if disabled.
    if(this->use_dependency_check() == false) // if is disabled, we
shut off the dependency check.
    this->set_int_param(MSK_IPAR_PRESOLVE_LINDEP_USE, MSK_OFF);
    if(r == MSK_RES_OK)
        r = MSK_putmaxnumanz(_task, _num_of_nonzero);

    if((_task_type == LINEAR_PROG || _task_type == SOCP) && r ==
MSK_RES_OK)
    {
        for (unsigned int j = 0; j < _num_vars && r == MSK_RES_OK;
++j)
        {
```

```
            /*set the linear term c_j in the objective*/
            if(r == MSK_RES_OK)
                r = MSK_putcj(_task, j, c[j]);

            /*set bounds on the variable j*/
            if(r == MSK_RES_OK)
                r = MSK_putbound(_task,
                                 MSK_ACC_VAR,   /* put bounds on
variable*/
                                 j,
                                 _bkx[j],       /* bound key*/
                                 _blx[j],       /* numeric lower
bounds*/
                                 _bux[j]);      /* upper bounds*/

            /* Input column j of A*/
            if(r == MSK_RES_OK)
                r = MSK_putavec(_task,
                                MSK_ACC_VAR,
                                j,
                                _aptre[j] - _aptrb[j], /* number of
non-zero in j*/
                                _asub + _aptrb[j],    /* pointer to row
indexes of column j*/
                                _aval + _aptrb[j]);   /* pointer to
Values of column j*/
        }

        /* Set bounds on the constraints*/
        for (unsigned int i = 0; i < _num_linear_cons && r ==
MSK_RES_OK; ++i)
        {
            r = MSK_putbound(_task,
                             MSK_ACC_CON,
                             i,
                             _bkc[i],
                             _blc[i],
                             _buc[i]);
        }

        /* set the objective sense of the problem*/
        if(r == MSK_RES_OK)
            r = MSK_putobjsense(_task, _objsense);

        /* Add cones to the task*/
        if(_task_type == SOCP)
            this->add_cones();

        /* Start Solving the problem*/
        if(r == MSK_RES_OK)
        {
```

```
            /** the file to check the problem*/
            if(this->output_opf_file())
                MSK_writedata(_task, "taskdump.opf");

            if(this->output_mbt_file())
                MSK_writedata(_task, "li.mbt");

            MSKrescodee trmcode;
            /* Run the optimizer*/
            r = MSK_optimizetrm(_task, &trmcode);
            /*Print a Summary containing information
            * about the solution for debugging purpose*/
            MSK_solutionsummary(_task, MSK_STREAM_ERR);

            if(r == MSK_RES_OK)
            {
                MSKsolstae solsta;

                if(_task_type == LINEAR_PROG)
                    MSK_getsolutionstatus(_task,
                                          MSK_SOL_BAS,
                                          NULL,
                                          &solsta);
                else if(_task_type == SOCP)
                    MSK_getsolutionstatus(_task,
                                          MSK_SOL_ITR,
                                          NULL,
                                          &solsta);
                else
                    std::cerr << "This is not implemented yet!\n";

                switch (solsta)
                {
                case MSK_SOL_STA_OPTIMAL:
                case MSK_SOL_STA_NEAR_OPTIMAL:
                    std::cout << "We reach a good optimal solution!!"
<< std::endl;
                    break;

                case MSK_SOL_STA_DUAL_INFEAS_CER:
                case MSK_SOL_STA_PRIM_INFEAS_CER:
                case MSK_SOL_STA_NEAR_DUAL_INFEAS_CER:
                case MSK_SOL_STA_NEAR_PRIM_INFEAS_CER:
                    std::cout << "Primal or dual infeasibility
found!.\n";
                    break;

                case MSK_SOL_STA_UNKNOWN:
                    std::cout << "Solution State can not be
determined.\n";
                    break;
```

```
                default:
                    std::cout << "Other solution state" << std::endl;
                    break;
                }
            }

            /**
            *  regardless the solution status , we need the solution
to find out where goes wrong
            */
            if(this->output_solution_file())
                MSK_writesolution(_task, MSK_SOL_ITR, "solution.dat");

            if(_task_type == LINEAR_PROG)
                MSK_getsolutionslice(_task,
                                     MSK_SOL_BAS,      /* request the
basic solution  */
                                     MSK_SOL_ITEM_XX, /* which part of
the solution  */
                                     0,                /* Index of the
first variable */
                                     _num_vars,         /* index of the
last variable+1*/
                                     xx);
            else if(_task_type == SOCP)
                MSK_getsolutionslice(_task,
                                     MSK_SOL_ITR,      /* request the
interior solution*/
                                     MSK_SOL_ITEM_XX, /* which part of
the solution   */
                                     0,                /* Index of the
first variable  */
                                     _num_vars,         /* index of the
last variable+1 */
                                     xx);
            else
                std::cerr << "Not implemented yet!";
        }   //end of solution state.

        if(r != MSK_RES_OK)
        {
            /* In case of an error, print error code and description*/
            char sysname[MSK_MAX_STR_LEN];
            char desc[MSK_MAX_STR_LEN];
            std::cout << "An Error occurred while optimizing" <<
std::endl;
            MSK_getcodedesc(r,
                            sysname,
                            desc);
            printf("Error %s -'%s'\n", sysname, desc);
```

```
        }
    }

    // We put the primal variable in first
    obj = this->get_primal_obj_val();

    libmesh_assert(primal->initialized());

    for(unsigned int i = 0; i < primal->size(); i++)
        primal->set(i, xx[i]);

    this->Get_dual_cone_variables(dual_nonlinear);
    this->get_solution_piece(dual_linear, MSK_SOL_ITEM_Y, 0);
     if(this->solution_good())
            return true;
     else
            return false;
    STOP_LOG("MOSEK_Solution", "Solve");
```

8 COMPLIE OF THE LIBRARY

8.1 Compliers

The finite element library libMesh was developed on Linux and a few modifications have been made to port it to Windows, primarily some APIs to calculate the time of a particular routine. The current library has been developed on Visual Studio 2008; however, codes have been written with best care in standard C++ and should theoretically be able to be compiled using other complier (has not been tested).

8.2 Solvers

8.2.1 MOSEK

MOSEK is academically free software (download from here http://www.mosek.com/), and a license can be applied using a college email for 6 months, after which the license needs to be applied again. To use the library of the MOSEK, the location of the `mosek.lib` and `mosek.h` need to be included in the directory of the library.

8.2.2 IPOPT

IPOPT is an open source solver and has been included and compiled successfully in the current library. However, it is recommended that the latest version of the IPOPT should be tried and complied. The required information in the library is the static library file and the header file. As an open source file, a number of the third-party tools are required. It is necessary that the documentation of the IPOPT is read and well understood (see https://projects.coin-or.org/Ipopt for details).

8.2.3 Knitro

Knitro is a commercial solver (can be downloaded from http://www.ziena.com/knitro.htm) and free version is available for a small number of variables and constraint. This solver has been included and tested (a full version can be applied in person to their sale department for one month trial). According to author's experience, Knitro is generally faster and more efficient than the free solver IPOPT, however, the potential of the IPOPT deserves better exploit as there have been a few successful papers describing using IPOPT.

REFERENCE

Byrd, R. H., Nocedal, J., and Waltz, R. A. (2006). "KNITRO: An Integrated Package for Nonlinear Optimization" in Large-Scale Nonlinear Optimization." *G. di Pillo and M. Roma, eds*, 35-59.

Kawajir, Y., Laird, C., and Wacher, A. (2010). "Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT". http://web.mit.edu/ipopt_v3.8/doc/documentation.pdf.

Kirk, B. S., Peterson, J. W., Stogner, R. H., and Carey, G. F. (2006). "libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations." *Engineering with Computers*, 22(3-4), 237-254.

Lyamin, A. V. (1999). *Three-dimensional lower bound limit analysis using nonlinera programming*, Ph.D Thesis, University of NewCastle, Australia.

MOSEK, A. (2010). "The MOSEK C API manual.Version 6.0 (Revision 103)."*User's Manual and Reference*. http://www.mosek.com.

Sloan, S. W. (1989). "Upper Bound Limit Analysis Using Finite-Elements and Linear-Programming." *International Journal for Numerical and Analytical Methods in Geomechanics*, 13(3), 263-282.