NORTHWESTERN UNIVERSITY

An Incrementally Non-linear Model for Clays with Directional Stiffness and a Small Strain Emphasis

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Civil and Environmental Engineering

By

Xuxin Tu

EVANSTON, ILLINOIS

June 2007

ABSTRACT

An Incrementally Non-linear Model for Clays with Directional Stiffness and a Small Strain Emphasis

Xuxin Tu

In response to construction activities and loads from permanent structures, soil generally is subjected to a variety of loading modes varying both in time and location. It also has been increasingly appreciated that the strains around well-designed foundations, excavations and tunnels are mostly small, with soil responses at this strain level generally being non-linear and anisotropic. To make accurate prediction of the performance of a geo-system, it is highly desirable to understand soil behavior at small strains along multiple loading directions, and accordingly to incorporate these responses in an appropriate constitutive model implemented in a finite element analysis.

This dissertation presents a model based on a series of stress probe tests with small strain measurements performed on compressible Chicago glacial clays. The proposed model is formulated in an original constitutive framework, in which the tangent stiffness matrix is constructed in accordance with the mechanical nature of frictional materials and the tangent moduli therein are described explicitly. The stiffness description includes evolution relations with regard to length of stress path, and directionality relations in terms of stress path direction. The former relations provide distinctive definitions for small-strain and large-strain behaviors,

and distinguish soil responses in shearing and compression. The latter relations make this model incrementally non-linear and thus capable of modeling inelastic behavior.

A new algorithm based on a classical substepping scheme is developed to numerically integrate this model. A consistent tangent matrix is derived for the proposed model with the upgraded substepping scheme. The code is written in FORTRAN and implemented in FEM via UMAT of ABAQUS. The model is exercised in a variety of applications ranging from oedometer, triaxial and biaxial test simulations to a C-class prediction for a well-instrumented excavation. The computed results indicate that this model is successful in reproducing soil responses in both laboratory and field situations.

ACKNOWLEDGMENTS

First of all, I sincerely thank my supervisor Prof. Richard Finno for his constant support, warm encouragements and insightful advices, without which I would certainly not have been able to reach where I am. I would like to thank other members on the review committee for my doctoral research – Prof. John Rudnicki, Prof. Raymond Krizek and Prof. Charles Dowding – for their valuable suggestions, comments and good discussions. I would like to extend my gratitude to all those people who aroused and fostered my interest in science and technology at different stages of my development. At this particular point, I would love to show my best respects to all those researchers who have contributed to this school's paramount reputation in mechanics, which makes me truly proud of being a Northwestern graduate.

I have studied in the geotechnical group for five and a half years, during which many fellow students shared time with me and became an indispensable part of the 'recent history' of my personal life. Everyone of them deserves thanks from me for various specific reasons. It is quite worth giving a lengthy but memorable list here, including Alireza Eshtehard, Amanda Morgan, Brandon Hughes, Cecilia Rechea Bernal, Frank Voss, Grigorios Andrianis, Hasan Ozer, Helsin Wang, Hsiao-Chou Chao, Izzat Katkhuda, James Lynch, Jill Roboski, Kirk Ellison, Kristin Molnar, Laura Sullivan, Laureen McKenna, Levasseur Severine, Luke Erickson, Maria Alarcon, Markian Petrina, Matthieu Dussud, Michele Calvello, Mickey Snider, MikeWaldron, Miltiadis Langousis, Remi Baillot, Roea Sabine, Sandra Henning, Sara Knight, Sebastian Bryson, Taesik Kim, Tanner Blackburn, Terence Holman, Wan-Jei Cho and Young-Hoon Jung. Thank you fellows for greetings, smiles, talks, jokes, funs, etc., especially those who were so willing to

come to my help with their expertise, whether it was about Abaqus or Medela, those who frequently enjoyed homemade lunches with me, and those who shared with me their joys and sufferings.

The model presented in this dissertation is based on an extensive experimental program performed on compressible Chicago glacial clays in the Geotechnical laboratory at Northwestern, which has involved a number of institutes, companies and individuals. I would like to particularly acknowledge Terence Holman, Wan-Jei Cho and Luke Erickson for providing high-quality test data and assisting in data interpretations. Prof. Jose Andrade and Dr. Young-Hoon Jung are greatly appreciated for their warmhearted help and many inspirations derived from my conversations with them. Financial support for this work was provided by National Science Foundation grant CMS-0219123 and the Infrastructure Technology Institute (ITI) of Northwestern University. The support of Dr. Richard Fragaszy, program director at NSF, and Mr. David Schulz, ITI's director, is gratefully acknowledged.

Last, but not the least, I would like to thank my mom and dad for the numerous benefits with which they endowed me. A lot of thanks to my parents-in-law for helping me with nursing tasks during my preparation of this dissertation. Finally, I would love to dedicate this dissertation to Mrs. Xin Fang, my beloved wife and best friend, and Miss Vivian Yanyun Tu, my two-month-old lovely daughter. The happiest thing I could ever imagine is making this journey together with you two ladies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Due to conservative codes and standards for design and construction, the strains induced in the soil for a well-designed geotechnical project are usually very small, i.e. the limit state is not crucial in the design of most projects. Rather, accurate prediction of the corresponding small ground movements is the governing factor in design. For instance, the design of an excavation in a crowded urban area must carefully consider the influence of excavation-induced ground movements on adjacent existing buildings. The strain levels of the affected soil in this case are mostly on the order of 0.1% or smaller, a level referred to as *small strains*. To make accurate prediction of the performance of such a geo-system, it is highly desirable to well understand the soil properties at small strains and subsequently incorporate them in an appropriate soil model implemented in a finite element analysis.

For the past twenty years, research concerning compressible Chicago glacial clays at Northwestern University has resulted in a database of stress-strain responses under axisymmetric and, to a lesser extent, plane strain conditions. Recently, an experimental investigation of soil behavior at small strains was started. A series of stress probe tests on high-quality block samples with high resolution strain measurements were conducted, the data from which constitute the main experimental basis for the constitutive study presented in this dissertation.

The research into soil behavior at small strains gained momentum in 1980's. It was found that for a variety of soils there are three notable behaviors at small strain levels – stiffness degradation (e.g. Burland 1989; Atkinson 2000), stiffness directionality (e.g. Burland and Georgiannou 1991; Costanzo et al. 2006) and influence of recent stress history (Atkinson et al. 1990). Stiffness degradation refers to the initial high stiffness at very small strains, with rapidly decreasing values with increasing strains. Stiffness directionality means that soil stiffness has significant path-dependency. Influence of recent stress history refers to the soil property that soil stiffness changes dramatically for any sharp change in loading path, in contrast to the consistent decrease in value if the path is continued with the same direction. Research also showed that these properties play important roles in predicting ground movements accurately at small strains (e.g. Jardine et al. 1986; Burland 1989; Stallebrass and Taylor 1997). Although there are several soil models that attempt to account for some of the previously mentioned behaviors, so far no one has been able to produce satisfactory results in simulating the responses of compressible Chicago glacial clays. However, many components of these models were found to be useful.

This dissertation presents a soil model based on the stress probe tests performed on block specimens of compressible Chicago glacial clay. The theoretical framework of this model is original, in which the tangent stiffness is explicitly described in terms of two basic behaviors, stiffness evolution and stiffness directionality, respectively. Because of the explicitness in the stiffness description, this model is experiment-friendly, for tangent stiffness can be directly measured in most experiments. Because of the description of directionality, this model distinguishes itself from a conventional "variable moduli" model, and provides an alternative and simple approach to incremental non-linearity. Such a framework shows great advantages in

incorporating small strain relations and taking into account other well-known relations for soils or soil properties, such as the critical state, the virgin compression curve, hysteresis in a loading cycle and shear-induced volume change. The model development is achieved with straightforward formulation, easy-to-understand parameters and simple numerical implementation.

In Chapter 2, technical background for the work is provided. Incremental non-linearity and its recent development, the stress probe tests performed on the compressible Chicago glacial clay, and a number of existing models that deal with small strain behaviour of soils are summarized. In addition, a statement of the notation convention used in this dissertation is given.

Chapter 3 presents the mathematical formulation and experimental basis for the proposed model. First, the form of the tangent stiffness matrix is proposed for axisymmetric conditions. The physical nature of the tangent moduli involved in the matrix is discussed. A mathematical mapping from axisymmetric conditions to general conditions is developed. Two basic variables and three characteristic zones are introduced as important features of the proposed model. Next, relations for stiffness evolution are presented with regard to different characteristic zones. The emphasis is placed on the definition of small strain behavior, with elaboration of its relation for compressible Chicago clays with the well-known ageing effect. Relations for stiffness directionality are proposed in terms of each tangent modulus. The mechanism used by the model to handle stress reversals is presented. Relations between directionality and plasticity and recent history effect are discussed. Finally, the material parameters required for the model are summarized, as are recommendations for their experimental determination.

Chapter 4 discusses the numerical implementation of the proposed model in a finite element code. A typical coupling system used to perform a non-linear finite element computation is introduced initially. Details are given of the existing substepping method with error control, and how to improve it to integrate the proposed constitutive equations. Emphasis is placed on deriving the algorithmically-consistent tangent matrix for the improved substepping method. In comparison with other constitutive models, it is shown that the proposed model has remarkable advantages in numerical implementation.

Chapter 5 shows the computed model responses in drained/undrained triaxial tests, drained/undrained biaxial tests, an oedometer test involving an unload-reload cycle, and a well-instrumented deep excavation in downtown Chicago. It is shown that this model is successful in simulating various soil tests and is promising in its ability to predict ground movements due to earth constructions. Suggestions for future improvements of this model are also made.

Chapter 6 presents a summary of this dissertation, conclusions, and recommendations for future research.

2 TECHNICAL BACKGROUND

2.1 INCREMENTAL NON-LINEARITY

Any constitutive relation or material model can be generally expressed by a rate form:

$$\dot{\boldsymbol{\varepsilon}} = \boldsymbol{F}(\dot{\boldsymbol{\sigma}}) \tag{2.1}$$

where $\boldsymbol{\varepsilon}$, $\boldsymbol{\sigma}$ and $\mathbf{F}$ are total strain, effective stress and a tensorial function of second-order, respectively. The mark "$\cdot$" either represents a time rate for a time-dependent material or an infinitesimal increment for a time-independent material. Sometimes Eq. (2.1) is expressed in reverse way, i.e., $\dot{\boldsymbol{\sigma}} = \boldsymbol{F}^{-1}(\dot{\boldsymbol{\varepsilon}})$. For that case, which is merely an issue of preference, the positions of $\dot{\boldsymbol{\varepsilon}}$ and $\dot{\boldsymbol{\sigma}}$ simply need to be exchanged in the subsequent discussions. To be rate-independent, the function $\boldsymbol{F}(\dot{\boldsymbol{\sigma}})$ must be *positively homogeneous* of degree one, i.e.

$$\boldsymbol{F}(\lambda\dot{\boldsymbol{\sigma}}) = \lambda\boldsymbol{F}(\dot{\boldsymbol{\sigma}}) \tag{2.2}$$

where λ is an arbitrary positive real number. For most materials, $\boldsymbol{F}(-\dot{\boldsymbol{\sigma}}) \neq -\boldsymbol{F}(\dot{\boldsymbol{\sigma}})$ due to irreversible or plastic responses, which means λ cannot be negative. Eq. (2.2) imposes a mathematical constraint on developing models for rate-independent materials.

The constitutive relation is so-called *incrementally linear* if $\boldsymbol{F}(\dot{\boldsymbol{\sigma}})$ is a linear function, i.e., $\boldsymbol{F}(\dot{\boldsymbol{\sigma}}_1 + \dot{\boldsymbol{\sigma}}_2) = \boldsymbol{F}(\dot{\boldsymbol{\sigma}}_1) + \boldsymbol{F}(\dot{\boldsymbol{\sigma}}_2)$, with $\dot{\boldsymbol{\sigma}}_1$ and $\dot{\boldsymbol{\sigma}}_2$ arbitrarily given. Otherwise, it is *incrementally non-linear*, corresponding to a non-linear $\boldsymbol{F}(\dot{\boldsymbol{\sigma}})$ that does not satisfy the proceeding equation. Note that homogeneity does not necessarily infer linearity. For instance, the function $\boldsymbol{F}(\dot{\boldsymbol{\sigma}}) = \| \dot{\boldsymbol{\sigma}} \|$ is homogeneous but non-linear, with $\| \ \|$ denoting the Euclidean norm. However, linearity sufficiently infers homogeneity, not only positive homogeneity, which means λ could be negative. Therefore, a linear function $\boldsymbol{F}(\dot{\boldsymbol{\sigma}})$ essentially represents a reversible or elastic relation, which has long been known not to be applicable to most geomaterials. However, plastic responses can be generated using more than one linear functions:

$$\dot{\boldsymbol{\varepsilon}} = \boldsymbol{F}_i(\dot{\boldsymbol{\sigma}}), \quad if \quad \dot{\boldsymbol{\sigma}} \in \Psi_i; \quad i = 1 \cdots n \tag{2.3}$$

where Ψ_i sometimes is referred to as *tensorial zone* (Darve and Labanieh 1982), a subdomain defined in the incremental stress space, for which the linear function $\boldsymbol{F}_i(\dot{\boldsymbol{\sigma}})$ is defined. The term n denotes the total number of the tensorial zones. A constitutive relation in the form of Eq. (2.3) is called *incrementally multi-linear* (Darve et al. 1988) for $n > 1$ in general, and *incrementally bi-linear* for $n = 2$ specifically. For instance, both the Duncan-Chang model (Duncan and Chang 1970) and the Cam-clay model (Schofield and Wroth 1968) are incrementally bilinear, with one tensorial zone defined for *loading* and the other for *unloading*. Generally, elastoplatic models are multi-linear, for yield surfaces are typically used for defining multiple tensorial zones.

Despite a number of significant advances achieved along these lines (Lade 1977; Mroz et al. 1979; Dafalias and Herrmann 1982; Al-Tabbaa and Wood 1989; Whittle and Kavvadas 1994;

Stallebrass and Taylor 1997; Puzrin and Burland 2000), the elastoplastic approach has some noticeable limitations when modeling soils:

i.) Most soils do not exhibit distinct yielding and thus the determination of the yield surface tends to be uncertain (e.g. Smith et al. 1992);

ii.) The decomposition of total strain into elastic and plastic parts is extremely hard to be experimentally determined and often needs assumptive approximations (e.g. Anandarajah et al. 1995);

iii.) Special care must be taken to guarantee the continuity of the incremental response across the boundary between two adjacent tensorial zones (Darve and Labanieh 1982);

iv.) Mathematical structure of this type is relatively complicated, whereas model calibration is often based on a limited variety of soil tests (Tu and Finno 2007).

To overcome these limitations, incrementally non-linear relations have attracted much recent attention. In hypoplastic models, the following rate form has been adopted by different research groups (e.g. Chambon et al. 1994; Tamagnini et al. 1999; e.g. Kolymbas 2000):

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{A} \cdot \dot{\boldsymbol{\varepsilon}} + \boldsymbol{B} \, \| \dot{\boldsymbol{\varepsilon}} \| \tag{2.4}$$

where **A** is a fourth-order tensor, **B** is a second-order tensor, and "·" is the operator of tensor contraction. The non-linearity of Eq. (2.4) comes from $\| \dot{\boldsymbol{\varepsilon}} \|$, due to which model responses vary with strain increment directions. This distinctive feature enables a description of plastic behavior without resorting to strain decomposition and yield surface specification. Furthermore, division into multiple tensorial zones can be avoided, for the dependency of response on path direction

can be continuously defined in the tensorial function **B**. Hence, hypoplastic models possess distinctive advantages for soils in comparison with conventional elastoplastic models.

Though the strain decomposition is not required for hypoplastic models, a decomposition of total response into linear and non-linear parts instead has been imposed by Eq. (2.4), which actually presents another challenge for experimental determination. To be more experiment-friendly, the following form of incremental non-linearity is proposed for soils:

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{E}(\hat{\boldsymbol{\sigma}}) \cdot \dot{\boldsymbol{\varepsilon}} \tag{2.5}$$

where $\hat{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} / \| \dot{\boldsymbol{\sigma}} \|$ represents the path direction of the stress increment. Apparently, Eq. (2.5) meets the mathematical requirement for rate-independency. Darve (1982) suggested a form similar to Eq. (2.5) as a general form of incremental non-linearity for rate-independent materials. However, the equation proposed herein serves as a specific case of the general form suggested by Darve (1982). He proposed $\boldsymbol{E}(\hat{\boldsymbol{\sigma}})$ as a generalized representation of the tangent stiffness matrix. In hypoplastic models, for instance, $\boldsymbol{E} \equiv \boldsymbol{A} + \boldsymbol{B} \otimes \hat{\boldsymbol{\varepsilon}}$, where $\otimes$ is the operator of tensor product and $\hat{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}} / \| \dot{\boldsymbol{\varepsilon}} \|$ represents the path direction of the strain increment. However, $\boldsymbol{E}(\hat{\boldsymbol{\sigma}})$ of Eq. (2.5) corresponds to a direct description of the tangent matrix, without decomposition into multiple parts, as is done in hypoplasticity. Since tangent stiffness is directly measurable in most cases, the setup of an explicit tangent matrix could enable a constitutive relation to be largely experiment-based. Instead of $\hat{\boldsymbol{\varepsilon}}$, the path direction in Eq. (2.5) is solely described by $\hat{\boldsymbol{\sigma}}$, the advantages of which will be detailed later.

The proposed form of Eq. (2.5) appears similar to a "variable moduli" model (e.g. Duncan and Chang 1970; Jardine et al. 1986). However, the proposed model is fundamentally different from the "variable moduli" model, mainly because of the dependency of the tangent matrix on the path direction $\hat{\boldsymbol{\sigma}}$, which, from the author's point of view, is the essence of the incremental non-linearity. The "variable moduli" model is known for two main shortcomings. One limitation is coaxiality between stress and strain increments and a complete volumetric-deviatoric uncoupling (Tamagnini et al. 1999). The other drawback is numerical instabilities due to either the lack of continuity of model response across tensorial zones (Gudehus 1979) or the inconsistency in distinguishing between loading and unloading (Schanz et al. 1999). In the proposed model, the first problem is treated by adopting a cross-anisotropic matrix for $\boldsymbol{E}$, in which mechanisms for stress-strain non-coaxiality and volumetric-deviatoric coupling are naturally included. The second problem is naturally solved using continuous functions in terms of $\hat{\boldsymbol{\sigma}}$.

2.2 STRESS PROBE TEST WITH SMALL STRAIN MEASUREMENTS

In most geotechnical construction, the affected soil generally is subjected to a variety of loading modes varying both in time and location, as has been frequently demonstrated in numerical analysis (e.g. Finno et al. 1991; Whittle et al. 1993; Viggiani and Tamagnini 2000). Hence, it is of practical interest to systematically investigate mechanical properties of soils under various loading modes. Among experimental approaches to this end, a natural one is to perform so-called stress probe tests, in which a number of 'identical' soil specimens are tested with a series of stress increments along different stress path directions. The importance of probe tests

has been increasingly recognized and several large programs have been carried out on different soils, mostly under axisymmetric condition (e.g. Smith et al. 1992; Callisto and Calabresi 1998; Finno and Roboski 2005; Costanzo et al. 2006).

It also has been increasingly appreciated that the strains around well-designed foundations, excavations and tunnels are mostly small, typically on the order of 0.1% (e.g. Jardine et al. 1986; Burland 1989; Atkinson 2000; Clayton and Heymann 2001), with soil responses at this strain level generally being non-linear and anisotropic (e.g. Tamagnini and Pane 1999; Shibuya 2002; Ng et al. 2004). To investigate soil non-linearity and anisotropy at small strain levels, it is important to implement small strain measurements in experimental programs. In the stress probe tests performed on compressible Chicago glacial clay, Holman (2005) used subminiature LVDTs mounted directly on specimens to record local axial and radial strain values.

Fig. 2-1 illustrates the stress probes carried out by Holman (2005). In these tests, triaxial specimens were hand-trimmed from the block samples with a nominal diameter of 71 mm and a height-to-diameter ratio between 2.1 and 2.3. Each specimen was reconsolidated under k_0 conditions to the in-situ vertical effective stress σ_{v0}' of 134 kPa, and then subjected to a 36 hour, drained k_0 creep cycle, wherein lateral restriction was enforced. Following this k_0 creep phase, specimens were subjected to directional stress probes under drained axisymmetric conditions. The internal deformation measurements made by subminiature LVDTs mounted directly on the specimen were used to calculate axial and radial strains using the measured axial gage length and sample diameter, respectively. The axial load was measured using an internal load cell and corresponding axial stress were calculated using the measured axial load and the current sample

area from the measured radial deformation. Cell and pore pressures were measured using external differential pressure transducers. Internal stress and strain measurements were made at 5 to 20 second intervals by an automated data acquisition and control system.

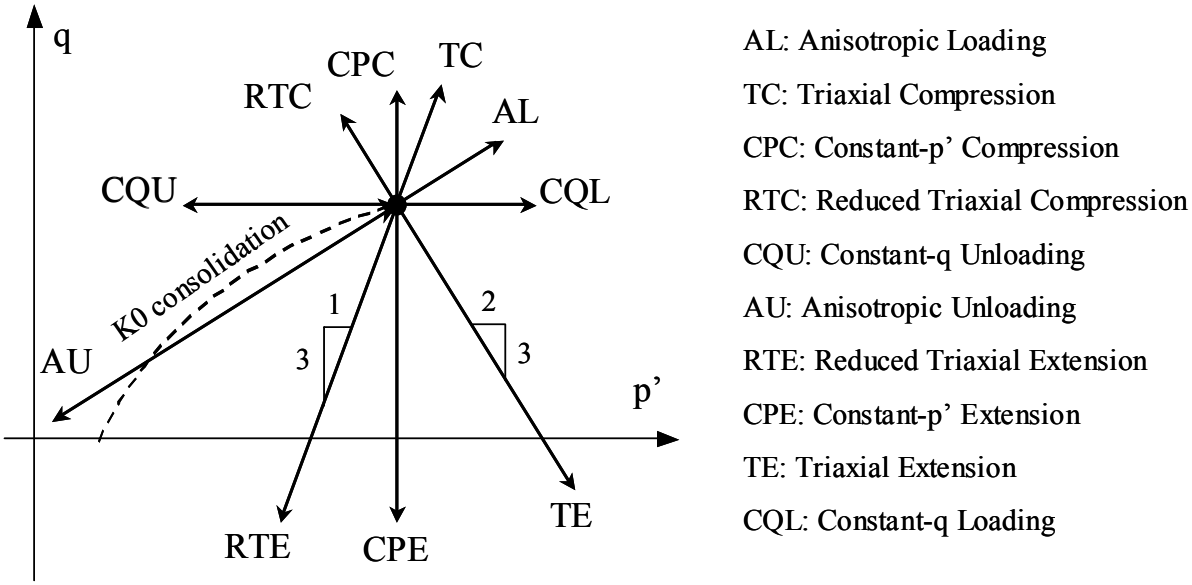

Fig. 2-1. Schematic diagram of a stress probe test program

The readings from the two axial LVDTs were averaged to produce a single axial deformation response, assumed to be representative of the centerline deformations within the zone of local measurement. Smoothed values of data collected by each transducer and load cell were used to calculate the local axial strain, ε_a, local radial strain, ε_r, vertical stress, σ_v', and horizontal stress, σ_h'. All stress probes were carried out at a stress rate of 1.2 kPa/hour to minimize accumulation of excess pore water pressure within a specimen. Duplicate tests were conducted for the majority of the stress probes.

2.3 SMALL STRAIN MODELS

Though a successful numerical analysis is affected by many factors (Finno and Tu 2006), the constitutive model is undoubtedly among the most critical ones. There are a number of models capable of dealing with various aspects of soil behavior at small strains. As a major improvement of the classical critical state model (Schofield and Wroth 1968), the bounding surface model (Dafalias and Herrmann 1982) enabled volumetric-shearing coupling inside a conventional yield surface, which in most cases overlaps the small strain range. On the basis of bounding surface plasticity, MIT-E3 (Whittle and Kavvadas 1994) further introduced a hysteretic elastic relation (Hueckel and Nova 1979) within the inner surface to reproduce the hysteretic response observed in most soil tests. Consistent with a 3-loci hypothesis (Smith et al. 1992), a series of multiple-surface kinematic hardening models (e.g. Al-Tabbaa and Wood 1989; Stallebrass and Taylor 1997; Puzrin and Burland 2000) have been developed in the form of an anisotropic hardening model (Mroz et al. 1979). This type of model provides a conceptually simple way to account for the effect of recent stress history (Atkinson et al. 1990) on directional stiffness at small strains. Another notable method is the hypo-plastic approach (e.g. Niemunis and Herle 1997; Viggiani and Tamagnini 1999; e.g. Kolymbas 2000; Lanier et al. 2004), founded on the theory of hypo-elasticity (Truesdell 1955) and the concept of incremental non-linearity (Darve 1991). Among other advantages, hypo-plastic models need neither strain decomposition nor determination of yield or potential surfaces, which are difficult to define in most soil experiments.

A solid constitutive model demands a solid experimental basis. The stress probe tests, equipped with small strain measurements, can systematically investigate soil responses in the entire axi-symmetric space, thus providing a comprehensive experimental basis for soil modeling. Unfortunately, very few, if any, existing models were developed on the basis of such tests. In fact, it is still an open question that conventional constitutive approaches, typically developed upon relatively limited experimental information, are actually capable of extrapolating correctly soil response upon different path directions (Costanzo et al. 2006). A case in point is that soil responses, especially the tangent stiffness, are significantly dependent on path direction, a material property having been reported by a number of researchers on various soils (e.g. Graham and Houlsby 1983; Callisto and Calabresi 1998; Finno and Roboski 2005; Costanzo et al. 2006) but only considered in very few soil models (Puzrin and Burland 1998). This property has made it difficult for most existing models to use same set of input parameters to simulate soil responses in all stress probes, though simulating one or two probes might not pose a problem.

This dissertation presents a constitutive model mostly based on the stress probe tests performed on compressible Chicago glacial clay (Holman 2005). This work also serves as an example of how to formulate an incrementally non-linear relation based on the conceptual platform laid out by Eq. (2.5). Unlike most existing soil models, the proposed model describes the tangent stiffness explicitly, which facilitates formulating constitutive relations directly based on experimental observations. In the meanwhile, the drawbacks of a "variable moduli" model, wherein tangent stiffness is expressed explicitly as well, are avoided by taking into account incremental non-linearity. Furthermore, it can be shown that the proposed constitutive

framework is fairly suitable for describing small strain behaviours of "unstructured" soils. Though this proposed model does not provide a special mechanism guaranteeing thermo-mechanical correctness as does a hyper-plastic model (e.g. Collins and Houlsby 1997; Houlsby and Puzrin 2000), employing solid experimentally-based relations will effectively minimize possible violations of the fundamental principle, especially in the experimentally-evaluated loading modes. A theoretically rigorous treatment in this aspect remains for future work.

2.4 NOTATION CONVENTION

In this dissertation, the usual sign convention of soil mechanics (compression positive) is adopted throughout. As a default, effective stresses are assumed throughout this dissertation, though its traditional mark "′" sometimes is omitted for simplicity. The usual sign convention of soil mechanics (compression positive) is adopted. In the representation of stress and strain states, use is made of the following invariant quantities: mean normal stress $p' = \mathrm{tr}(\boldsymbol{\sigma}')/3$; deviatoric stress $q = \sqrt{(3/2)}\,\|\mathrm{dev}(\boldsymbol{\sigma}')\|$; volumetric strain $\varepsilon_v = \mathrm{tr}(\boldsymbol{\varepsilon})$; and deviatoric strain $\varepsilon_s = \sqrt{(2/3)}\,\|\mathrm{dev}(\boldsymbol{\varepsilon})\|$. Tensors are represented by bold letters. Unless otherwise stated, summation convention is not employed in equations listed herein.

3 MODEL FORMULATION

This chapter describes the experimental basis and mathematical formulation for the proposed *directional stiffness model*, so-called to emphasize the path-dependency of tangent stiffness, and to distinguish this model from a traditional "variable moduli" model, in which moduli only vary with stress/strain levels.

3.1 TANGENT STIFFNESS MATRIX

As defined in Eq. (2.5), the tangent stiffness $\boldsymbol{E}$ is a 6×6 matrix linking stress and strain increments. Generally, this matrix includes 36 independent components. To be practical, the form of $\boldsymbol{E}$ needs to be prescribed in such a way that matrix components of relative importance should be identified and emphasized. Furthermore, it is noted that the matrix components that can be investigated in conventional soil experiments are quite limited. In developing the proposed model, a basic idea was to formulate experimentally-based relations for these limited components first and then make appropriate extensions to general conditions. Note that these kinds of extensions, essentially due to the limitation of current experimental capability, are not only needed by this specific model but needed by any other model as well.

3.1.1 AXI-SYMMETRIC CONDITION

3.1.1.1 FOUR TANGENT MODULI

In most standard soil experiments, soil specimens are trimmed into a cylindrical shape and tested under axi-symmetric conditions. Under these conditions, the tangent stiffness matrix can be generally expressed as:

$$\begin{Bmatrix} \delta\varepsilon_v \\ \delta\varepsilon_s \end{Bmatrix} = \begin{bmatrix} 1/K & 1/J_v \\ 1/J_s & 1/3G \end{bmatrix} \begin{Bmatrix} \delta p' \\ \delta q \end{Bmatrix} \tag{3.1}$$

where K is the bulk modulus, G is the shear modulus and J_v and J_s are two coupling moduli. These four moduli are all tangent moduli. The infinitesimal mark "δ" is adopted in this chapter for infinitesimal stress/strain increments, indicating that the current version of directional stiffness model is time-independent. This infinitesimal "δ" should be distinguished from the finite mark "Δ" that will be frequently used later in Chapter 4 to denote finite stress/strain increments in numerical schemes.

According to Eq. (3.1), J_v defines shear-induced volume change of the material, a behavior that has been widely observed for many soils. For instance, it is well-known that loose sand or normally consolidated clay tends to contract while dense sand or highly overconsolidated clay tends to dilate, under drained shear conditions. These phenomena can be fully described by devising a proper function for J_v. Specifically, shear-induced contraction can be captured by positive J_v for $\delta q > 0$ or negative J_v for $\delta q < 0$, while shear-induced dilation can be simulated by negative J_v for $\delta q > 0$ or positive J_v for $\delta q < 0$. Note that the shear-dilation response also can be accounted for using dilatancy angle ψ (Rowe 1962), which links volumetric change to shear

strain and typically is implemented in an elasto-plastic model (e.g. Menetrey and Willam 1995; Schanz et al. 1999). In essence, J_v and ψ are two different approaches to the same issue.

In contrast to J_v, J_s describes how the change in mean stress p' contributes to shear strain development, a property that has not received much attention. It is worth having a special discussion on the nature of this unconventional modulus.

3.1.1.2 PHYSICAL NATURE OF J_S

In literature, the two coupling moduli typically are assumed to be identical (e.g. Graham and Houlsby 1983; Puzrin and Burland 1998). Nevertheless, not only their mathematical definitions (cf. Eq. (3.1)), but also experimental observations, indicate that J_v and J_s are different from each other. There are two particular stress probe tests that are especially important for understanding the physical meaning of J_s. Fig. 3–1(a) shows the volumetric strain and the shear strain measured in constant q unloading (CQU) test, wherein p' keeps decreasing while q remains constant, i.e., $\Delta q = 0$. Because shear strains as large as 2% develope in this test, when there is no change in shear stress, Js must play an important role in the response.

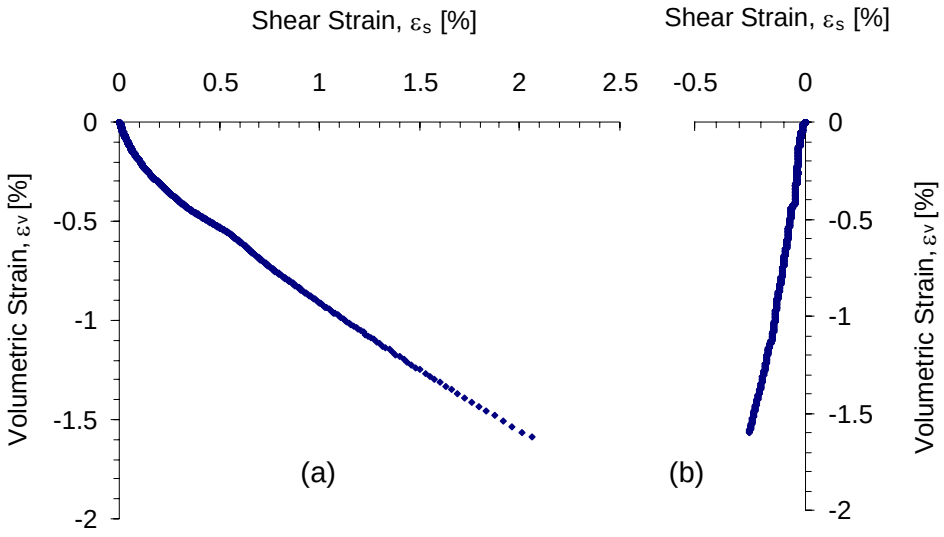

Fig. 3–1. Measured strain components in (a) CQU test and (b) AU test

Another relevant phenomenon is observed in anisotropic unloading (AU) test, wherein the stress path basically points straight back to the origin of the p′-q space, as shown in Fig. 2-1. Though this test involves a significant change in q, the measured shear strain is surprisingly small, nearly negligible as shown in Fig. 3–1(b). These two observations viewed together strongly suggest that shear strain in soil is actually governed by change in the stress ratio q/p′, denoted by η hereinafter, rather than mere change in q. This observation corresponds to the following mathematical form for describing the shear behavior of soils.

$$\delta\varepsilon_s = \frac{\delta\eta}{G^*} \tag{3.2}$$

where G^* is a nondimensional modulus, different from the conventional shear modulus G. According to Eq. (3.2), the observed shear strain in AU test should be relatively small, because η does not change therein. Conversely, in a CQU test, η increases until failure is reached, and a

substantial amount of shear strain should be expected. It can be shown that Eq. (3.2) is suitable for any other stress probe wherein the stress path leads to failure.

A further expansion of the right hand side of Eq. (3.2) yields:

$$\delta\varepsilon_s = \frac{\delta q}{p' G^*} - \frac{q\delta p'}{p'^2 G^*} \quad \textbf{(3.3)}$$

Note that the expression for ε_s implied in Eq. (3.1) is:

$$\delta\varepsilon_s = \frac{\delta q}{3G} + \frac{\delta p'}{J_s} \quad \textbf{(3.4)}$$

By comparing Eq. (3.3) to Eq. (3.4), it is apparent that G^*, G and J_s are related to one another:

$$G = p' G^*/3; \;\; J_s = -p'^2 G^*/q; \;\; J_s = -3p'G/q \quad \textbf{(3.5)}$$

Therefore, both G and J_s essentially originate from G^*, a nondimensional modulus describing the relation between stress ratio η and shear strain.

Note that η is an alternative representation of mobilized friction angle, the peak value of which defines the failure surface for frictional materials. The geometry of a failure surface reveals intrinsic information about the shear behavior of the material. An important characteristic of failure via a critical state definition is the theoretically infinite amount of shear strain. Therefore, the failure surface is also a surface of equal shear strain, an analogy to an equipotential surface. As the normal to the equipotential surface designates the direction of the driving force that leads to the most dramatic change in potential, the normal to failure surface designates the direction in stress space along which the most dramatic change in shear strain will

occur. Thus, the stress quantity measured in this direction represents the stress increment of unit magnitude that produces the largest amount of shear strain.

Fig. 3–2 shows the failure surfaces for non-frictional material and frictional material, respectively. As shown, the failure of non-frictional material is independent of p′. The norm to the failure surface is parallel to the q-axis, which means q is the most critical factor in generating shear strain for non-frictional material. Mathematically, it corresponds to the following.

$$\delta\varepsilon_s = \frac{\delta q}{G} \tag{3.6}$$

Though this equation has been frequently used for describing soil behavior, fundamentally it is applicable only to non-frictional materials, like most metals. As shown in Fig. 3–2(b), an idealized failure surface for frictional material corresponds to a constant stress ratio, and thus the norm to the failure surface can be mathematically represented by a change in η alone. In other words, the most critical factor controlling shear strain development for frictional material is the stress ratio η. Therefore, Eq. (3.2) in essence originates from the mechanical nature of frictional material, as does the coupling modulus J_s.

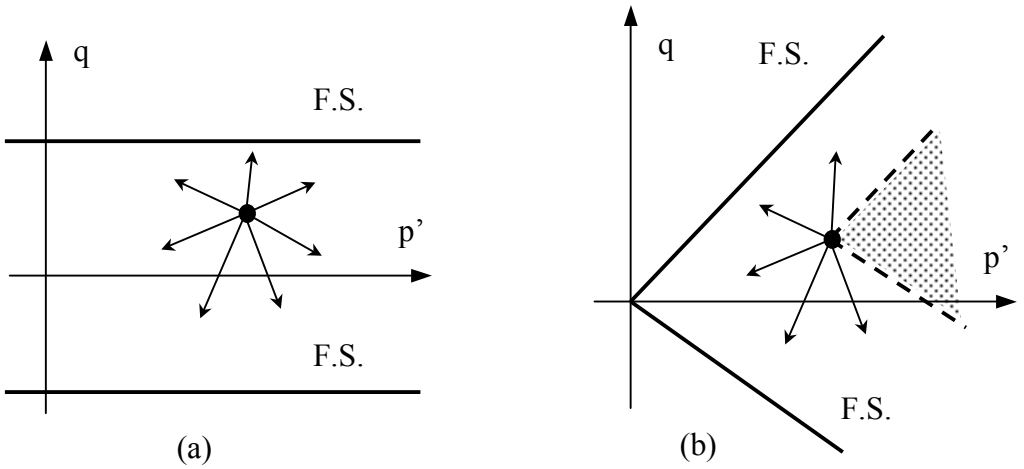

Fig. 3–2. Relation between failure surface (F.S.) and material shear response: (a) non-frictional material and (b) frictional material

Note that the admissible stress space enveloped by the failure surface of frictional material is distorted in comparison to that of non-frictional material. For non-frictional material, nearly all stress paths point to the failure surface, which means Eq. (3.6) is generally applicable. Note that the only two paths not pointing to the failure surface are horizontally oriented in p′-q space, which are still covered by Eq. (3.6) as two special cases in which no shear strain develops. In contrast, for frictional material, a significant percentage of stress paths do not point to the failure surface, as indicated by the hatched sector in Fig. 3–2(b), which is bounded by an envelope parallel to the failure surface. Along any stress path falling in this sector that is oriented to the right with an angle with the p′-axis less than that of the failure surface, the material undergoes a more compressive deformation mode, for which Eq. (3.2) is not necessarily suitable because it essentially describes frictional shearing. In fact, test data from CQL probe, wherein p′ increases with no change in q, exhibits very little shear strain, indicating that Eq. (3.6) is more applicable to the shear response in this sector than Eq. (3.2) and thus J_s is negligible therein. The difference between the hatched sector and the remaining stress space will be detailed later.

3.1.2 GENERAL CONDITION

For in situ soils that have been deposited in horizontal layers, it is appropriate to assume their properties are cross-anisotropic. The following cross-anisotropic matrix has been implemented in a number of soil models (e.g. Lings et al. 2000; Kuwano and Jardine 2002; Jung et al. 2004).

$$\begin{Bmatrix} \delta\varepsilon_x \\ \delta\varepsilon_y \\ \delta\varepsilon_z \\ \delta\gamma_{xy} \\ \delta\gamma_{yz} \\ \delta\gamma_{zx} \end{Bmatrix} = \begin{bmatrix} 1/E_h & -\nu_{hh}/E_h & -\nu_{vh}/E_v & 0 & 0 & 0 \\ -\nu_{hh}/E_h & 1/E_h & -\nu_{vh}/E_v & 0 & 0 & 0 \\ -\nu_{hv}/E_h & -\nu_{hv}/E_h & 1/E_v & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{hh} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{vh} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{vh} \end{bmatrix} \begin{Bmatrix} \delta\sigma'_x \\ \delta\sigma'_y \\ \delta\sigma'_z \\ \delta\tau_{xy} \\ \delta\tau_{yz} \\ \delta\tau_{zx} \end{Bmatrix} \quad \textbf{(3.7)}$$

where subscripts x and y indicate the two horizontal axes and z indicates the vertical axis. There are seven independent indices involved in this matrix. Under axisymmetric conditions in a triaxial cell, only the 3×3 sub-matrix at the top-left corner is applicable, which includes five independent indices. To investigate the relation between these five indices and the four moduli discussed in the previous section, the sub-matrix is extracted and expressed as:

$$\begin{Bmatrix} \delta\varepsilon_x \\ \delta\varepsilon_x \\ \delta\varepsilon_z \end{Bmatrix} = \begin{bmatrix} A & B & C \\ B & A & C \\ D & D & E \end{bmatrix} \begin{Bmatrix} \delta\sigma'_x \\ \delta\sigma'_x \\ \delta\sigma'_z \end{Bmatrix} \quad \textbf{(3.8)}$$

where the five new indices A through E are introduced for convenience, with A = $1/E_h$, B = $-\nu_{hh}/E_h$, C = $-\nu_{vh}/E_v$, D = $-\nu_{hv}/E_h$ and E = $1/Ev$. For axisymmetric conditions, the incremental shear and volumetric strains can be derived from Eq. (3.8):

$$\begin{cases} \delta\varepsilon_s = 2(2D + E - A - B - C)\delta p/3 + 2(-2D + 2E + A + B - 2C)\delta q/9 \\ \delta\varepsilon_v = (2D + E + 2A + 2B + 2C)\delta p + 2(-D + E - A - B + 2C)\delta q/3 \end{cases} \quad \textbf{(3.9)}$$

Comparing Eq. (3.9) with Eq. (3.1), one obtains the following set of linear equations:

$$\begin{cases} 2(A+B)+2C+2D+E=1/K \\ 2[-(A+B)+2C-D+E]/3=1/Jv \\ 2[-(A+B)-C+2D+E]/3=1/Js \\ 2[(A+B)-2C-2D+2E]/9=1/(3G) \end{cases} \quad \textbf{(3.10)}$$

Rearranging Eq. (3.10), the following equations can be established (Finno and Tu 2006):

$$\begin{cases} A+B=(-6/J_s+4/K+3/G-6/J_v)/18 \\ C=(6/J_v-3/J_s+2/K-3/G)/18 \\ D=(2/K-3/G+6/J_s-3/J_v)/18 \\ E=(3/G+1/K+3/J_s+3/J_v)/9 \end{cases} \quad \textbf{(3.11)}$$

Hence, if the four tangent moduli are known, C, D, E and the sum of A and B can be computed through Eq. (3.11). Note that the determination of A and B depends on ν_{hh}, since $B/A = -\nu_{hh}$ while the sum of A and B is known. Experimentally, ν_{hh} can be determined using true triaxial tests, or accordingly changing the specimen orientation in a regular triaxial test. Besides ν_{hh}, the other two unknown indices are G_{hh} and G_{vh}, which can be investigated by properly-oriented bender elements, or in either hollow cylinder torsion tests or direct simple shear tests with specimens appropriately orientated. However, these tests are not common and little test data are available. For simplicity, three hypothetic relations are used in this proposed model.

$$\nu_{hh}=a; \quad G_{hh}=b\cdot G; \quad G_{hv}=c\cdot G \quad \textbf{(3.12)}$$

where a = 0.2 and b = c = 1 in default. Note that both ν_{hh} and G_{hh} are not exercised under plane strain conditions, seemingly the most common case in numerical analysis. Bender element measurements on the compressible Chicago clay have shown that different shear moduli at very small strains are relatively insignificant (Cho 2007), with G_{hh} approximately 20% higher than G_{vh}.

Note that the matrix in (3.7) is a compliance matrix. Its inverse matrix, substituted with Eqs. (3.11) and (3.12), leads to the following stiffness matrix:

$$\begin{bmatrix} \frac{A\mathrm{D}-EC}{\mathrm{D}A^2-2ECA-B^2\mathrm{D}+2EBC} & -\frac{B\mathrm{D}-EC}{\mathrm{D}A^2-2ECA-B^2\mathrm{D}+2EBC} & -\frac{C}{A\mathrm{D}-2EC+B\mathrm{D}} & 0 & 0 & 0 \\ -\frac{B\mathrm{D}-EC}{\mathrm{D}A^2-2ECA-B^2\mathrm{D}+2EBC} & \frac{A\mathrm{D}-EC}{\mathrm{D}A^2-2ECA-B^2\mathrm{D}+2EBC} & -\frac{C}{A\mathrm{D}-2EC+B\mathrm{D}} & 0 & 0 & 0 \\ -\frac{E}{A\mathrm{D}-2EC+B\mathrm{D}} & -\frac{E}{A\mathrm{D}-2EC+B\mathrm{D}} & \frac{A+B}{A\mathrm{D}-2EC+B\mathrm{D}} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad \textbf{(3.13)}$$

Eqs. (3.11)~(3.13) provide a complete mapping of tangent stiffness from an axisymmetric condition to a general condition. The general tangent matrix can be fully obtained, as long as the four tangent moduli under axisymmetry can be identified. The subsequent sections then describe how to define these tangent moduli based on experimental observations.

3.2 GENERAL CONSIDERATIONS IN STIFFNESS DEFINITION

Before going into details of this directional stiffness model, it is worthwhile to briefly discuss several substantial issues related to definition of tangent moduli.

3.2.1 TWO BASIC VARIABLES

The first important issue is how to select the basic variables upon which the moduli will be mathematically described. It is common in literature to report observed soil stiffness variations with regard to a relevant strain. Accordingly, it is convenient to define moduli as functions of

strains. However, strains are mathematically inconvenient since shear strain is theoretically infinite at failure, while failure definition in stress has no such ambiguity. More fundamentally, in many field applications, force/stress is the cause while deformation/strain is its consequence. Therefore, it is rational to use stresses as basic variables when defining stiffness measures.

To this end, it is proposed herein to use two stress-based quantities, *length of stress path*, LSP, and orientation angle, β. As shown in Fig. 3–3, the points O and C represent the initial and current stress states respectively. LSP is the length between O and C along the stress path experienced by the material, which is mathematically defined as:

$$LSP = \int_{\Gamma} \sqrt{(\delta p')^2 + (\delta q)^2} \tag{3.14}$$

where the integration path Γ corresponds to real stress path, which is generally nonlinear. This integration can be easily computed in a numerical scheme by linearizing the stress path in each time step and adding its increment at the end of each step. In Fig. 3–3, the arrow emanating from point C represents current stress increment $\delta\boldsymbol{\sigma}'$. Its inclination with the p′ axis is defined by β as:

$$\beta = \begin{cases} \arctan(\delta q / \delta p') & if \quad \delta p' > 0, \ \ \delta q \geq 0 \\ \pi + \arctan(\delta q / \delta p') & if \quad \delta p' \leq 0 \\ 2\pi + \arctan(\delta q / \delta p') & if \quad \delta p' > 0, \ \ \delta q < 0 \end{cases} \tag{3.15}$$

where $\delta p' = \mathrm{tr}(\delta\boldsymbol{\sigma}')/3$; $\delta q = \sqrt{(3/2)}\,\|\mathrm{dev}(\delta\boldsymbol{\sigma}')\|$. Accordingly, β increases counterclockwise and falls in [0, 2π), with β = 0 parallel to the p′ axis. Note that in a straight stress path, β coincides with the orientation angle of the overall stress path. However, in a general case, such as in Fig. 3–3, the overall stress path direction, which can be represented by the line segment connecting the points O and C, is different from β. β represents the direction of current stress increment,

while LSP accounts for the entire stress history starting from the initial state to the current stress state, as long as changes in path direction, if any, are continuous. For abrupt changes in path direction, e.g., a stress reversal, LSP should be "reset", as will be elaborated in Section 3.3.3. It will be shown later that LSP and β are useful terms to define the stiffness variation with both magnitude and direction of loading.

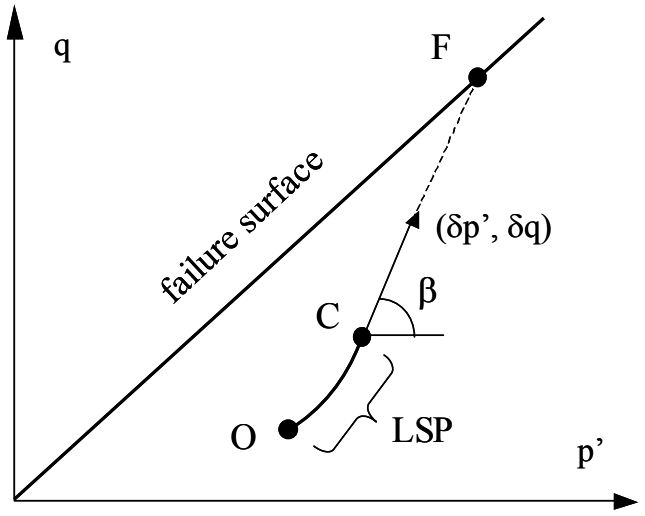

Fig. 3–3. Two basic variables for stiffness definition: LSP and β

3.2.2 SHEAR ZONE & COMPRESSION ZONE

Soil experiments essentially can be categorized into two types – shear and compression tests. As shown in Fig. 3–4, a shear test in the stress space corresponds to a stress path leading to the failure surface, wherein the soil specimen will eventually be failed in shear. All stress paths of this type together form the *shear zone* in Fig. 3–4.

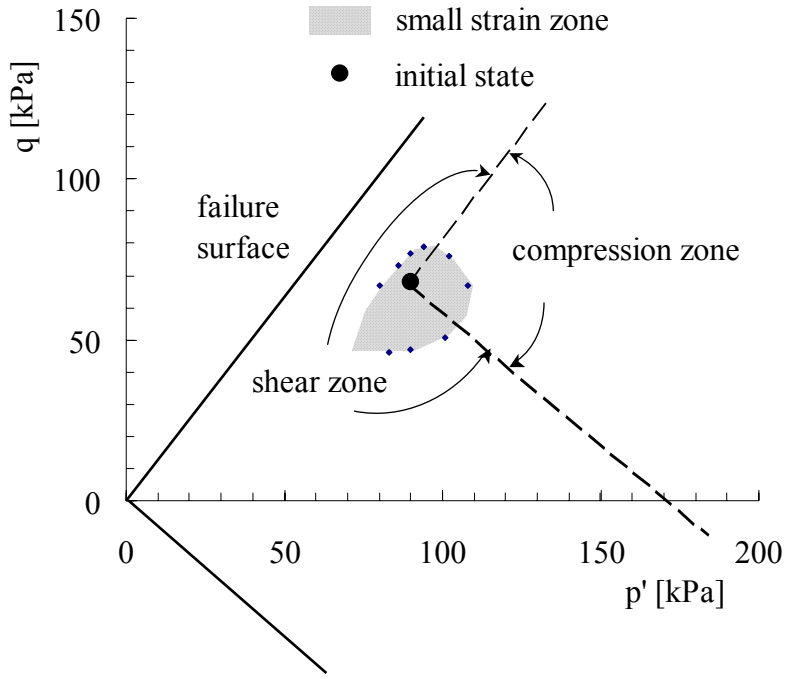

Fig. 3–4. Characteristic zonation: shear zone, compression zone and small strain zone

In contrast, a compression test is characterized by a stress path in which shear failure will never occur and the dominant deformation mode is compression. All paths of this type form a *compression zone* in Fig. 3–4. In a general stress space, the boundary between the shear and compression zones is a conical surface, and can be mathematically defined as:

$$f_{FS}(\boldsymbol{\sigma}'-\boldsymbol{\sigma}'_c) = 0 \qquad \textbf{(3.16)}$$

where $f_{FS}(\boldsymbol{\sigma}') = 0$ is the function for the failure surface, and $\boldsymbol{\sigma}'_c$ is the current stress. Hence, the boundary surface can be obtained by shifting the tip of the failure cone to the current stress point. In the p'-q plane, the boundary manifests itself as two curves that "parallel" the failure state curves. For a Mohr-Coulomb failure criterion, and using the stress path direction β defined in Eq.

(3.15), the two boundary curves can be denoted by two parameters derived from the friction angle ϕ :

$$\beta_{upper} = \arctan(\frac{6\sin\phi}{3-\sin\phi}); \quad \beta_{lower} = 2\pi - \arctan(\frac{6\sin\phi}{3+\sin\phi}) \tag{3.17}$$

where β_{upper} and β_{lower} correspond to the upper and lower boundary curves, respectively.

Generally speaking, the shear zone is dominated by shear response, while the compression zone is dominated by volumetric response. In terms of stiffness definition, it can be expected that tangent moduli will decrease with LSP in the shear zone but increase with LSP in the compression zone. The shear and compression zones constitute two tensorial zones using Darve's terminology (1982). These two tensorial zones, however, fundamentally originate from the frictional nature of soils, rather than a mathematical consideration.

It is worth mentioning here that these two characteristic zones correspond to different yield surfaces if accounted for in an elasto-plastic framework. For instance, the double hardening model (Lade 1977) uses a conical surface and a cap surface accounting for yielding in shear tests and consolidation tests, respectively, which are conceptual counterparts of the shear and compression zones used herein.

Of all stress probe tests performed on the soft Chicago clays (cf. Fig. 2-1), the probes AL and CQL are within compression zone and the others are within shear zone.

The *small strain zone*, the hatched area bounded by experimental data points in Fig. 3–4, will be introduced later as another characteristic zone. This zone is not a tensorial zone, because its boundary is measured in terms of LSP instead of β.

3.3 RELATIONS FOR STIFFNESS DEFINITION

The stiffness definition in this model includes separate relations for stiffness evolution and stiffness directionality. While these two sets of relations are conceptually independent and were developed separately on the basis of test data, they are associated in the sense that the directionality relations are mathematically hosted by the evolution relations.

3.3.1 STIFFNESS EVOLUTION

The evolution relations describe how the tangent moduli vary with LSP. Again, stiffness evolutions in the shear and compression zones are fundamentally different, and thus are treated separately.

3.3.1.1 EVOLUTION IN SHEAR ZONE

A stress path falls in shear zone when it leads toward the failure surface, i.e. the path will intersect the failure surface if extended unlimitedly along its direction. As shown in Fig. 3–3, this intersection, denoted by point F, defines an image point of point C on the failure surface, which serves as a benchmark for measuring the "distance" of the current stress state to possible failure. Note that this mapping approach is a useful technique in developing soil models (e.g.

Dafalias and Herrmann 1982). The stress at this image point, denoted by $\boldsymbol{\sigma}'_f$, can be defined in terms of the current stress $\boldsymbol{\sigma}'_c$ and the stress increment $\Delta\boldsymbol{\sigma}'$.

$$\boldsymbol{\sigma}'_f = \boldsymbol{\sigma}'_c + m\Delta\boldsymbol{\sigma}' \tag{3.18}$$

where m is an unknown scalar. Since the failure surface is defined as $f_{FS}(\boldsymbol{\sigma}'_f) = 0$, m can be obtained by solving:

$$f_{FS}(\boldsymbol{\sigma}'_c + m\Delta\boldsymbol{\sigma}') = 0 \tag{3.19}$$

Note that there could exist two solutions for m of different signs and the desired one is always positive to be consistent with the direction of loading. The "distance" between points C and F then is computed as mΔLSP, where ΔLSP = $\sqrt{\Delta p'^2 + \Delta q^2}$. And the "distance" between points O and F is defined as follows.

$$LSP_f = LSP + m\Delta LSP \tag{3.20}$$

Fig. 3–5 shows tangent shear modulus G versus LSP observed in selected stress probes in the shear zone, with G normalized by its initial value G_0, and LSP normalized by LSP_f, a constant in each individual stress probe. Note that G is computed according to Eq. (3.5). These curves are quite similar to each other, exhibiting the general pattern of stiffness evolution in all stress probes conducted in the shear zone by Holman (2005).

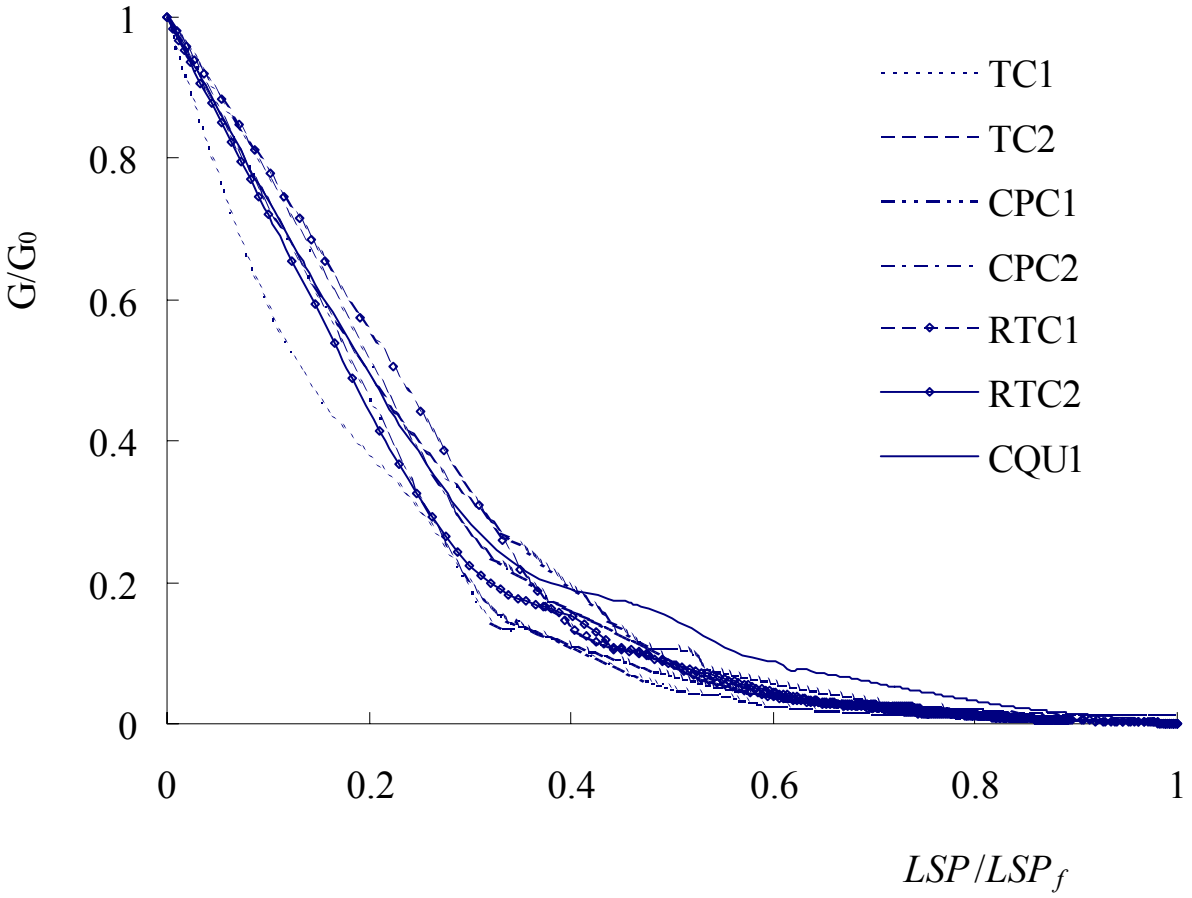

Fig. 3–5. Normalized shear modulus evolution in selected stress probes in the shear zone

As shown in Fig. 3–5, these evolution curves are composed of two distinct stages. Initially, the modulus decreases linearly and rapidly. After passing an easily visible kink in the curves, the degradation becomes nonlinear and much milder. This kink has a clear physical meaning and provides a reasonable criterion for defining the threshold of small strain behavior, which will be discussed later in more detail.

This twofold pattern is observed not only for G, but also for the other tangent moduli. Fig. 3–6 shows that the same pattern is also observed for degradation curves of other moduli. In the shear zone, J_s is defined according to Eq. (3.5) and thus degrades in a way similar to G.

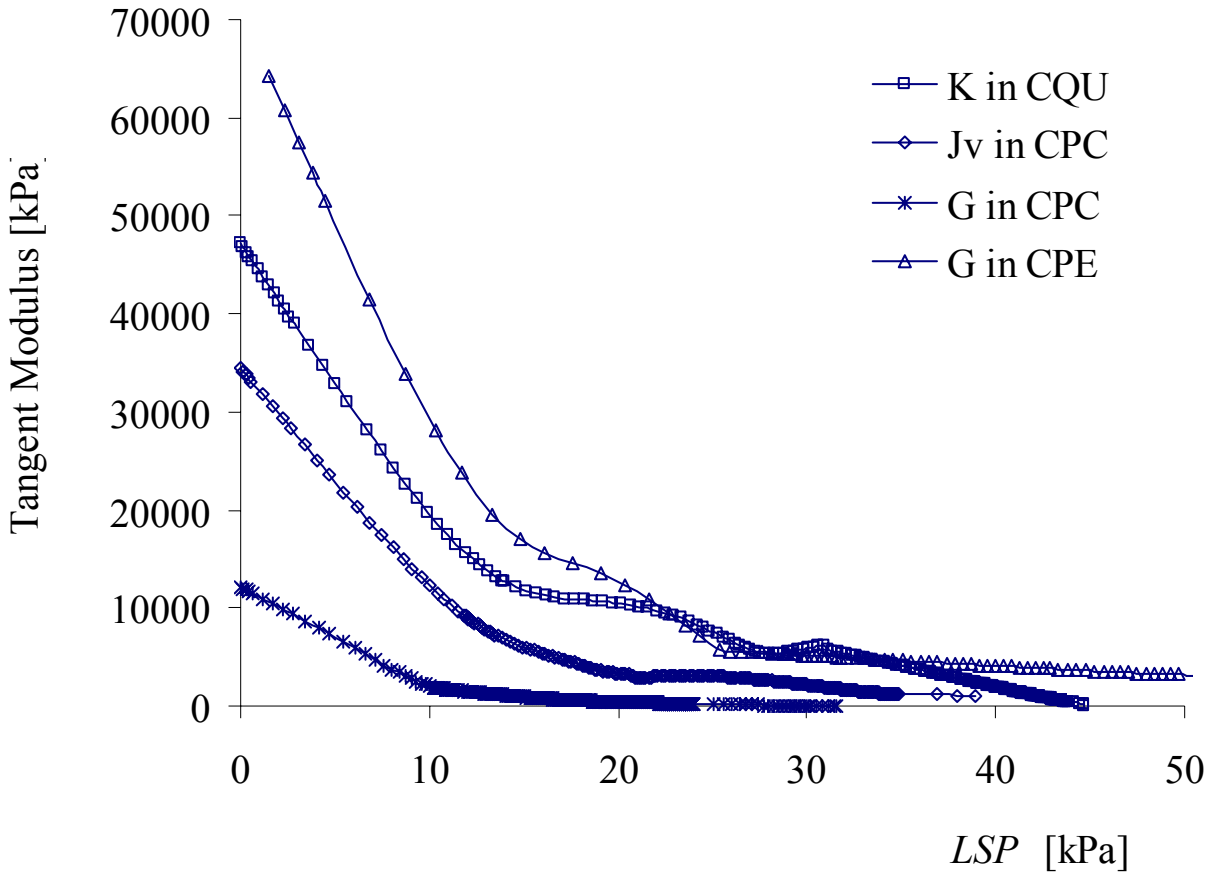

Fig. 3–6. Degradation of various moduli in the shear zone

In summary, evolution relations in the shear zone for the different moduli in Eq. (3.1) have an identical form, as schematically shown in Fig. 3–7.

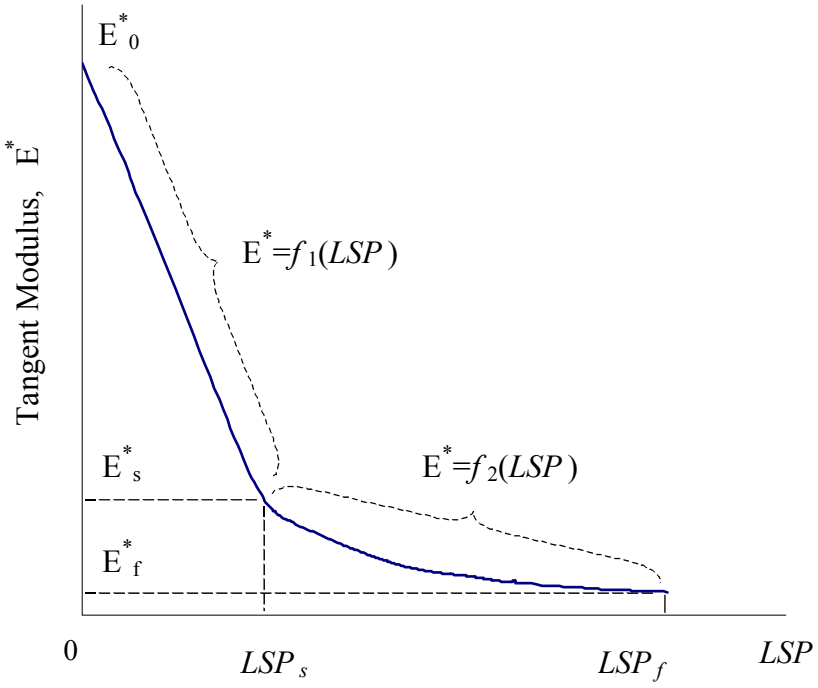

Fig. 3–7. Relation for stiffness evolution in shear zone

E^* represents any of the four tangent moduli and is defined by f_1(LSP) and f_2(LSP):

$$E^* = f_1(LSP), \quad if\ LSP \le LSP_s \qquad (3.21)$$
$$E^* = f_2(LSP), \quad if\ LSP > LSP_s$$

where:

$$f_1(LSP) = E_0^* - LSP \times (E_0^* - E_s^*)/LSP_s \qquad (3.22)$$

$$f_2(LSP) = \frac{(\mu - 1)LSP_s E_f^* - LSP_f E_s^* + (E_s^* + E_f^* - \mu E_f^*)LSP}{(\mu - 1)LSP_s - LSP_f + (2 - \mu)LSP} \qquad (3.23)$$

where E^*_0 is the *initial modulus*, LSP_s is the *threshold LSP*, defining the boundary of the small strain zone (cf. Fig. 3–4), E^*_s is the *threshold modulus*, and μ is the *coefficient of non-linearity* that controls the non-linearity of Eq. (3.23), as shown in Fig. 3–8. E^*_0, LSP_s, E^*_s and μ serve as four parameters in the evolution relations. Instead, LSP_f is a state variable to be computed

according to Eq. (3.20). E^*_f is the *failure modulus*, assumed to be constant. At failure, G_f should be small enough to generate a shear flow, while K_f should be large enough in comparison with G_f, so that a critical state (Schofield and Wroth 1968) can be approximately achieved. Js_f can be derived from G_f according to Eq. (3.5). Jv_f is assumed to be large in comparison with K_f, and thus its effect is neglected in a failure state. In this model, G_f = 1 kPa, K_f = 50 kPa, Jv_f = 1000 kPa for $q_f > 0$ and Jv_f = -10000 kPa for $q_f < 0$, where q_f denotes q at failure.

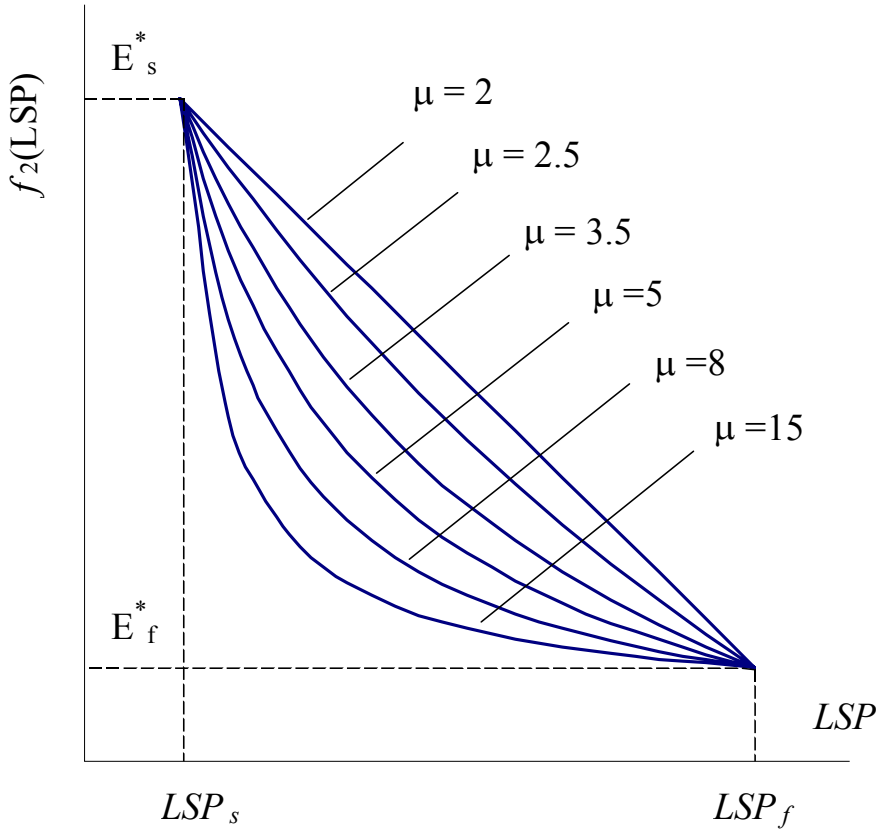

Fig. 3–8. Parameter μ controlling non-linearity of f_2(LSP)

In essence, f_1(LSP) defines the small strain behavior and f_2(LSP) defines the large strain behavior. f_2(LSP) is essentially a hyperbolic function of the following form:

$$E^* = \frac{a}{b + LSP} + c \tag{3.24}$$

The three unknowns, a, b and c, are derived by making Eq. (3.24) satisfy three conditions:

$$\begin{aligned} f_2(LSP_s) &= E_s^* \\ f_2(LSP_f) &= E_f^* \\ f_2(LSP_s/2 + LSP_f/2) &= (E_s^* - E_f^*)/\mu + E_f^* \end{aligned} \tag{3.25}$$

The 3rd condition provides a straightforward way to determine μ from experimental results. Given a stiffness degradation curve E^*~LSP, LSP_s can be determined by identifying the range of the linear degradation portion, while LSP_f basically is the maximum LSP value. Accordingly, E^*_s and E^*_f can be obtained from the curve. To determine μ, one needs to find out on the curve the E^* value at LSP = $(LSP_s+LSP_f)/2$, denoted as E^*_m. Then $\mu = (E^*_s - E^*_f)/ (E^*_m - E^*_f)$, according to Eq. (3.25). Typically, E^*_f is much smaller than E^*_s and E^*_m. Thus, $\mu \approx E^*_s/E^*_m$.

It is worth pointing out that if the twofold relation of Eq. (3.22) and (3.23) is plotted in terms of stiffness, either tangent or secant, versus relevant strain measure in a semi-logarithmic scale, the resulted curve is of a reversed-S shape, as shown in Fig. 3–9. Similar curves for other stress probes in the conventional way of data presentation can be found in Finno et al. (2005) and Holman (2005).

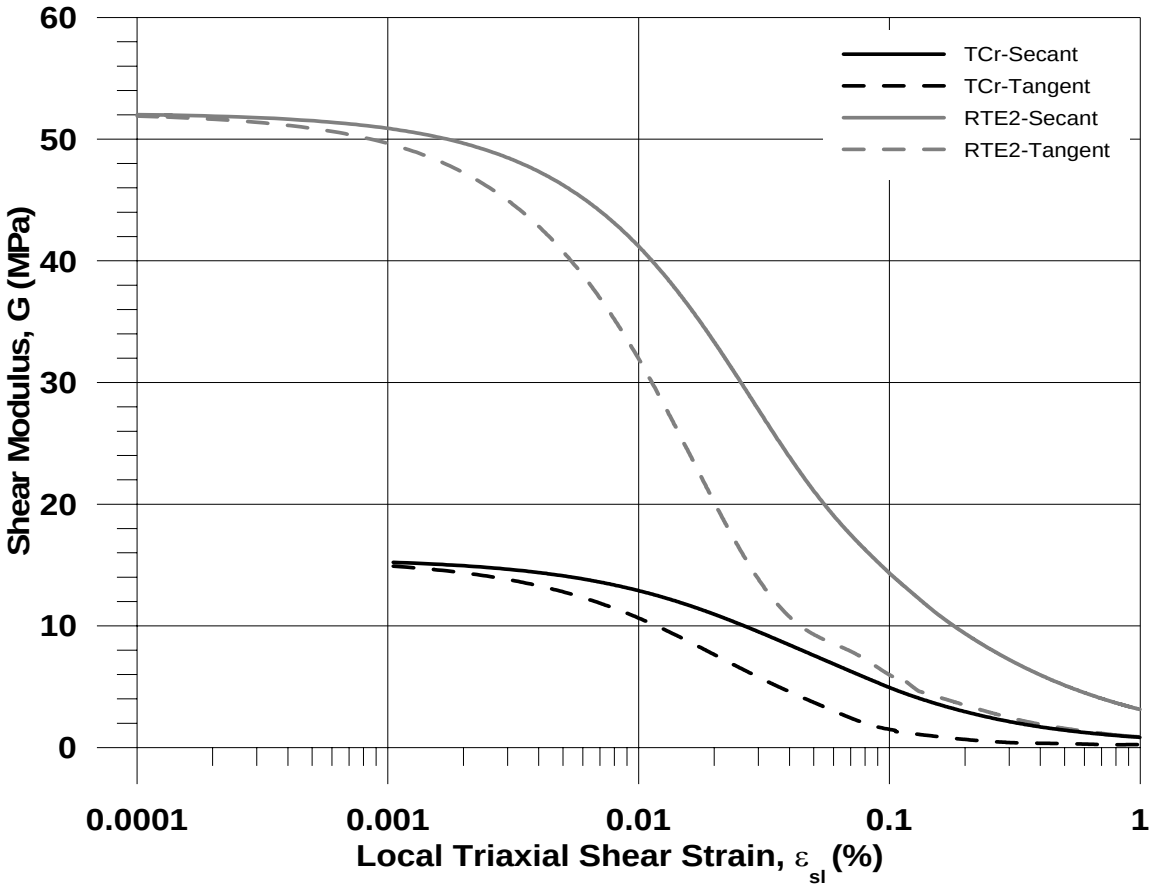

Fig. 3–9. The twofold relation plotted in conventional way

Stiffness degradation curves in reversed-S shape have been widely reported in the literature on numerous soils. As an example, Fig. 3–10 shows an illustration from Atkinson's paper (2000). The point here is that the twofold relation, comprising Eq. (3.22) and (3.23), is consistent with the common characteristic of soils, and thus is expected to be generally applicable.

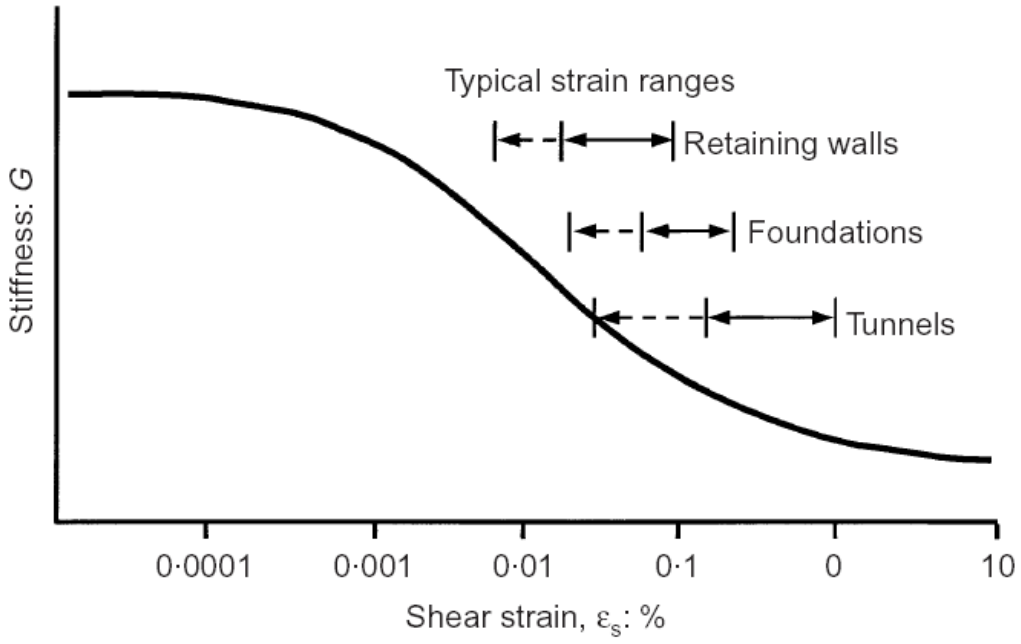

Fig. 3–10. A typical stiffness degradation curve for soils (after Atkinson 2000)

3.3.1.2 EVOLUTION IN COMPRESSION ZONE

Fig. 3–11 shows how the bulk and shear moduli evolve in the two compression tests, based on the raw data given by Holman (2005). Note that the significant jumps, especially on the two K curves, are mainly caused by the applied curve fitting techniques. As explained in Appendix B, it is usually difficult to find a single polynomial function to fit entire stress-strain relation. Hence, the task typically is achieved using different polynomial functions to fit different segments of the curve. As a matter of fact, curve fitting tends to be relatively inaccurate at ends of a segment, especially when the fitted curve is not particularly smooth. Therefore, fluctuations or discontinuities are expected around connections of two adjacent segments. However, these

drawbacks do not prevent the fitting technique from revealing global patterns of tangent stiffness evolution, as shown in Fig. 3–11.

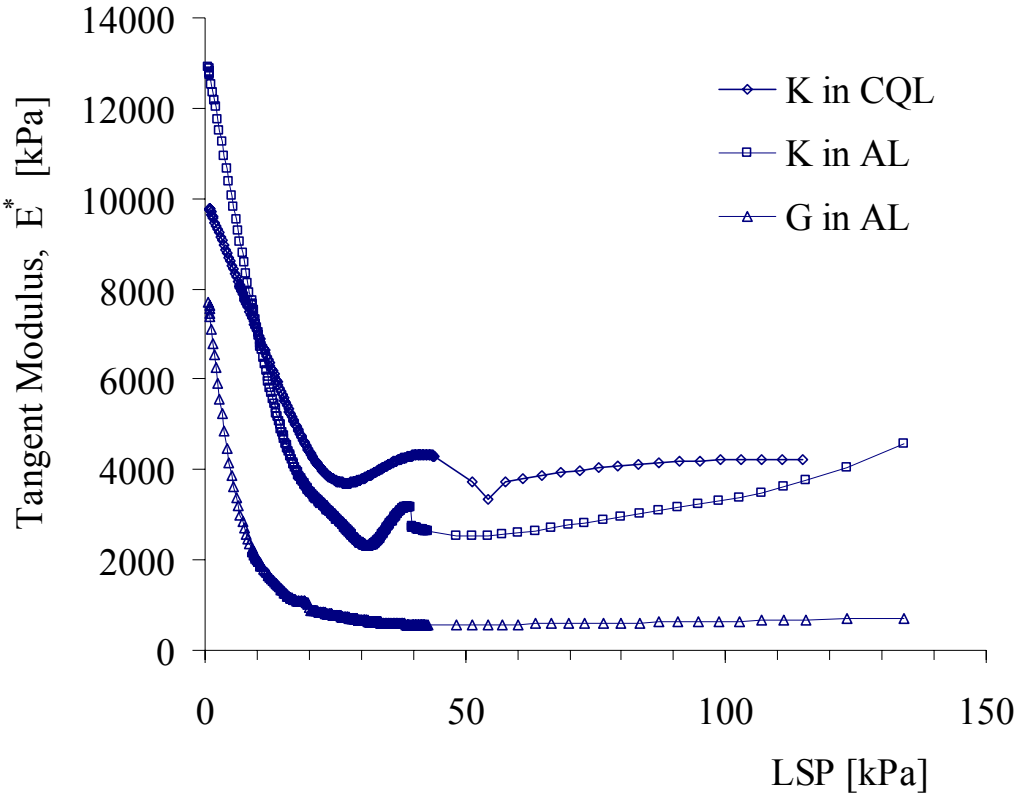

Fig. 3–11 Observed stiffness evolution in the compression zone

Fig. 3–12 shows the relation accordingly proposed for stiffness evolution in the compression zone. As shown, the linear function f_1(LSP) defined in shear zone also applies to the compression zone. LSP_s and E^*_s are again identified by a kink on the global curve. Unlike in shear zone, however, tangent moduli in compression zone eventually increase with LSP, consistent with the well-established equation for normally consolidated soils of a straight line in a $\ln(p')\sim\varepsilon_v$ plot.

$$K = p'/\lambda^* \quad if \quad p' \geq p'_p \tag{3.26}$$

where λ^* is a modified compression index and p'_p is the pre-consolidation mean normal stress, which typically is calculated given the over-consolidation ratio (OCR) and the initial mean normal effective stress p'_0. Accordingly, as a part of f_3(LSP), the proposed evolution relation for K in large strain range is as follows:

$$K = (p'_0 + LSP)/\lambda^* \quad if \quad LSP \geq LSP_p \tag{3.27}$$

where p'_0 is the initial effective pressure and $LSP_p = p'_p - p'_0$.

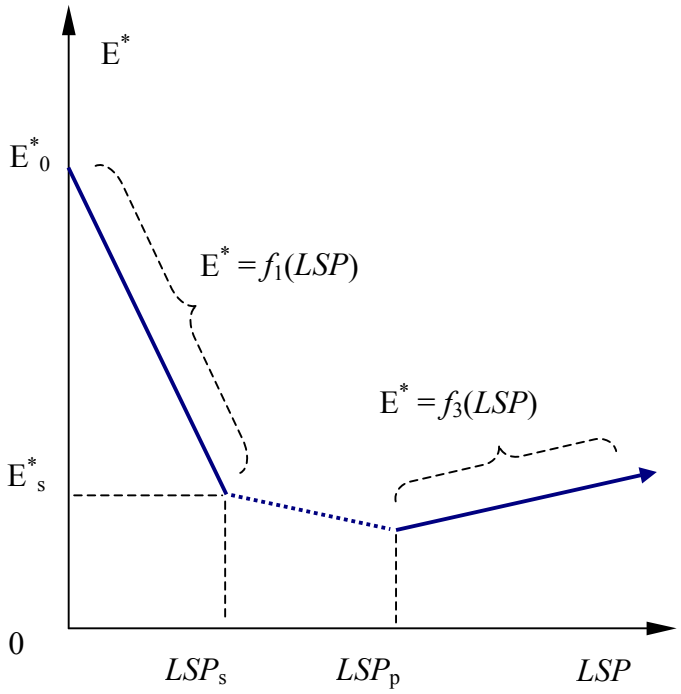

Fig. 3–12 Relation for stiffness evolution in compression zone

The relation for G in this region is derived by assuming a constant lateral stress ratio for normal consolidation:

$$G = \frac{3(1-k_0^{nc})}{2(1+2k_0^{nc})}K \quad if \quad LSP \geq LSP_p \tag{3.28}$$

where k^{nc}_0 is the *lateral stress coefficient* in a normal consolidation. By default, $k^{nc}_0 = 1 - \sin(\phi)$ (Holtz and Kovacs 1981). ϕ is the friction angle.

According to experimental observations, the shearing-volumetric coupling in the compression zone is insignificant. Hence, for convenience, the two coupling moduli J_s and J_v are assumed to be one order of magnitude larger than K:

$$J_v = J_s = 10K \quad if \quad LSP \geq LSP_p \tag{3.29}$$

Eqs. (3.27) to (3.29) constitute a complete representation of f_3(LSP) (cf. Fig. 3–12), the definition of large strain behavior in the compression zone. For over-consolidated soils, LSP_s is smaller than LSP_p, as shown in Fig. 3–12. In this case, a linear interpolation is used to describe the stiffness evolution in the transition.

3.3.1.3 SMALL STRAIN BEHAVIOR

Based on the preceding discussions, the soil response at the beginning of any stress path follows an identical pattern, i.e., as LSP increases, soil stiffness decreases linearly and rapidly from a relatively large initial value. This behavior is mathematically described by Eq. (3.22), and its range is specified by LSP_s. As has been shown in Fig. 3–4, a characteristic zone, herein named the *small strain zone*, is outlined by the dots scattering around the initial state, which correspond to LSP_s values obtained from stress probe tests.

As introduced in Section 2.2, all stress probe tests on the compressible Chicago clay are preceded by a k_0-consolidation stage and a subsequent drained creep stage at constant vertical stress with lateral restriction. It is well-known that most soils get stiffer after creep (e.g. Hueckel and Nova 1979; Rammah et al. 2004), which is commonly referred to as "aging." This aging effect can be well demonstrated in the AL probe test.

As shown in Fig. 2-1, the AL test is essentially a continuation of the k_0-consolidation since these two paths are nearly identical. In Fig. 3–13, the measured volumetric response is plotted for the entire process from the consolidation stage to the end of the AL test.

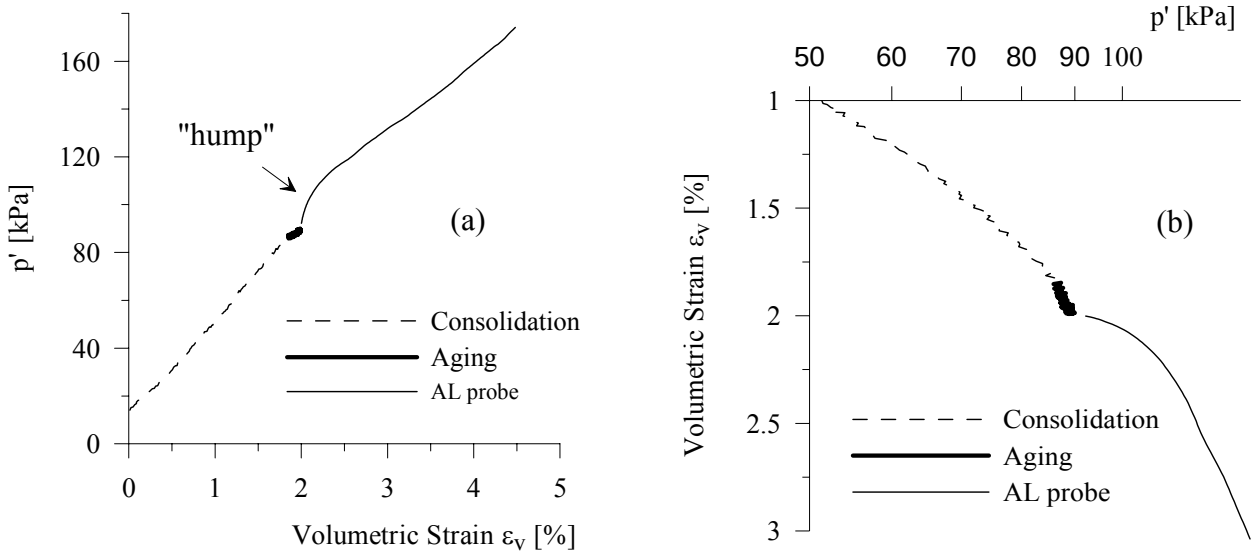

Fig. 3–13. Volumetric response throughout the AL probe test

Fig. 3–13(a) and (b) plot the same data but in different fashions. In Fig. 3–13(a), the significant "hump" at the beginning of the AL probe indicates a jump in soil stiffness, more specifically, the bulk modulus. As the probe continues, however, the bulk modulus gradually decreases, until the

local slope of the stress-strain curve in Fig. 3–13(a) approaches the slope at the end of the consolidation stage.

The ε_v-*log*p′ curve in Fig. 3–13(b) is found quite consistent with Bjerrum's (1967) generalized time-dependent behavior of soils. Bjerrum (1967) postulated the dashed portion of the curve in Fig. 3–14(a) is the soil response without any creep, while its solid counterpart results from certain period of drained creep, or ageing. This response exactly corresponds to what is decribed by Eq. (3.22), with point A corresponding to the LSP_s measured from the AL test. Therefore, the small strain behavior defined by Eq. (3.22) physically is an outcome of ageing, assuming the soil is not cemented in any way. There is no reason that soil stiffness increases only in one path direction, e.g., the direction of AL probe. Instead, soil stiffness increases in all path directions, but only for a limited strain/stress range (e.g. Hueckel and Nova 1979; Rammah et al. 2004). That is the reason why a small strain zone exists, as illustrated in Fig. 3–14(b). Experimentally, LSP_s can be determined for any stress probe by identifying the "kink" on a stiffness evolution curve. The scattering dots outlining the hatched area in Fig. 3–4 correspond to those LSP_s values experimentally obtained for compressible Chicago clay.

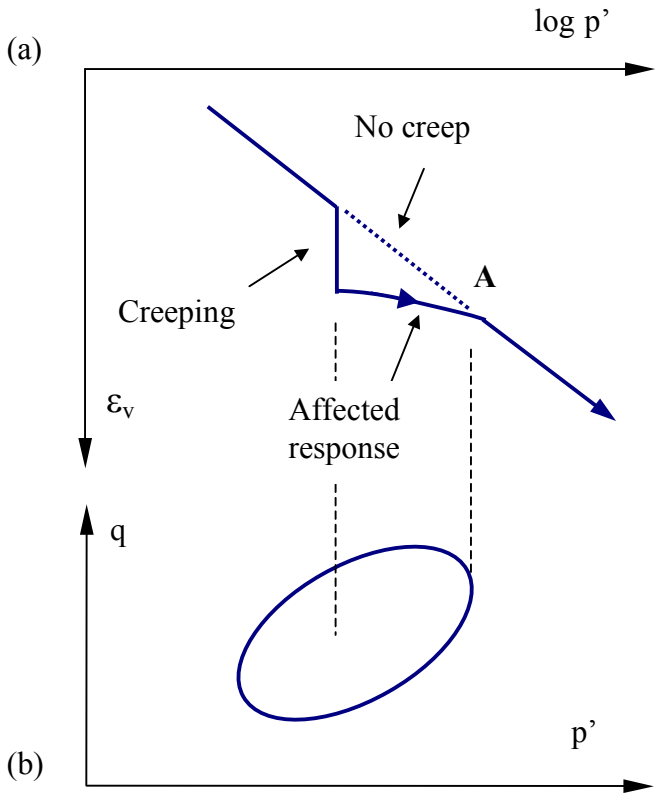

Fig. 3–14. The relation between effect of creeping and small strain behavior

It is expected that the initial stiffness E^*_0 will increase as the length of the creep period increases. However, the form of Eq. (3.22) is not expected to change as creep period changes, i.e., no matter how much increase in E^*_0 occurs, soil stiffness always decreases linearly as LSP increases, until E^*_s is resumed. E^*_s represents the stiffness level unaffected by the aging and thus is independent of the ageing period. To estimate E^*_0 for in situ soils, which typically have experienced hundreds or thousands of years of ageing, it is recommended to use field seismic testing. Though creep periods applied in laboratory are always shorter than in situ geological histories, it is worth noting that both volumetric creep strain and stiffness increase due to creep tend to change quickly at the beginning of creep and approach to asymptotic values as creep

persists (Holtz and Kovacs 1981). Therefore it is still possible to make a close estimate of E^*_0 based on laboratory tests with short but adequate creep periods.

In literature, several criteria have been proposed for defining a range for small strain behavior, including generalized strain (Simpson et al. 1979), major principal strains (Jardine 1985), strain energy (Burland and Georgiannou 1991) and tangent stiffness ratio (Puzrin and Burland 1998). These definitions, however, are somewhat arbitrary and do not involve specifying the mechanical nature of the small strain behavior. In comparison, Eq. (3.22) and the threshold parameter LSP_s provide a new set of definition for small strain behavior and small strain range respectively, with an identified physical meaning and ease of experimental determination. Note that this definition is intended for unstructured soils such as the Chicago glacial clay and does not necessarily apply to cemented or structured soils.

3.3.2 STIFFNESS DIRECTIONALITY

The evolution relations enable a systematic and concise description of stiffness evolution within each characteristic zone using a limited number of parameters, i.e. E^*_0, E^*_s, LSP_s and μ. However, due to *directionality*, i.e., the dependency of soil response on stress path direction, these parameters are actually functions of the path direction β. This section deals with these directionality functions – $E^*_0(\beta)$, $E^*_s(\beta)$, $LSP_s(\beta)$ and $\mu(\beta)$. Based on experimental results, piecewise linear relations are used to construct these functions. Measures are taken to make

smooth transition between the shear and compression zones. Furthermore, discussions are made on the relation of directionality with plasticity and recent history effect.

3.3.2.1 DIRECTIONALITY RELATIONS

Fig. 3–15 shows the directionality relation proposed for the threshold shear modulus, G_s, in comparison with data calculated from Holman's tests (2005). According to the test data, the maximum value G_{max} occurs in the CPE test, where $\beta = 1.5\pi$, while its minimum value G_{min} occurs in the AL test ($\beta = 0.66$). Accordingly, a piecewise linear relation is proposed to represent the observed variation in G_s with direction of loading, as shown in Fig. 3–15.

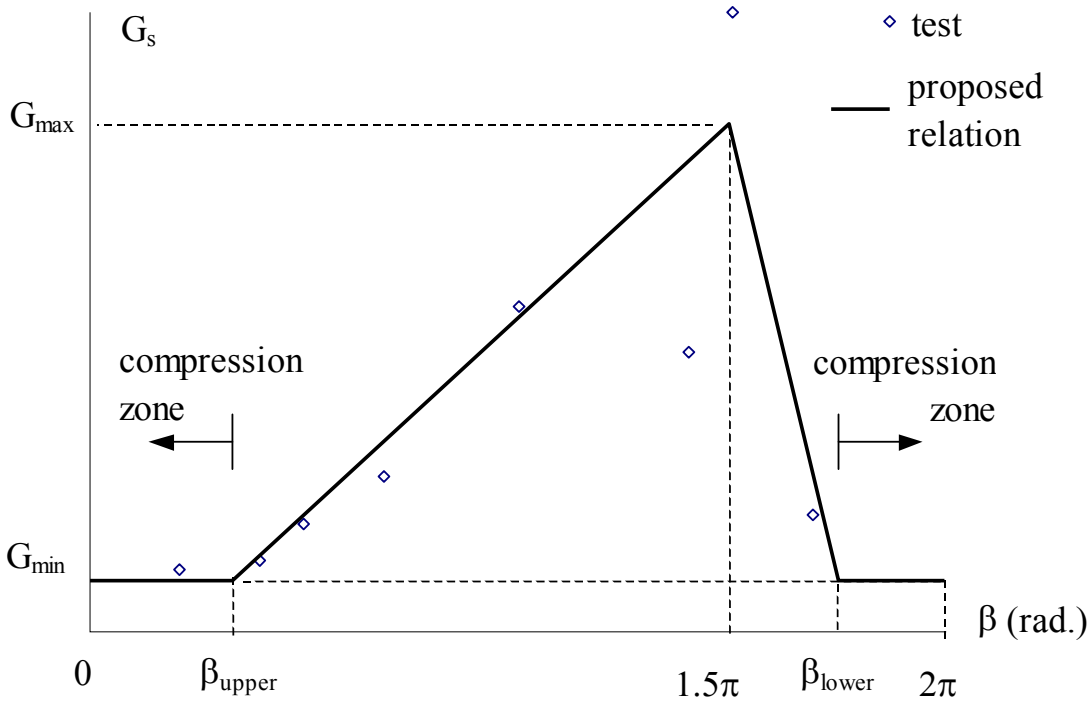

Fig. 3–15. Proposed directionality relation for G_s in comparison with test data

The two constants β_{upper} and β_{lower}, introduced in Eq. (3.17), define the boundaries between the shear and compression zones.

The directionality relation observed for the compressible Chicago clay is very similar to some test data reported on other clays. Fig. 3–16 shows the data from stress probe tests on Pisa clay reported by Callisto and Calabresi (1998), with ω defined in the same way as β, except that the angle is expressed in degree. The Pisa clay had a natural water content around 60%, an OCR varying between 1.5 and 2, and a friction angle of about 26º. As shown, the Pisa clay data bear much resemblance to the Chicago clay data, despite a small rightward shift, with maximum values occurring around the TE test ($\omega = 303.7°$) and minimum values occurring in the CPC test ($\omega = 90°$). Note that the shear modulus in Fig. 3–16 was computed by Callisto and Calabresi (1998) as $G = (\delta q/\delta \varepsilon_s)/3$, which differs from Eq. (3.5) that is used herein with results shown in Fig. 3–15. However, these two computations yield the same results in the CPC and CPE tests, which control the overall shape of the directionality curves. In other words, the difference in computing G does not affect the shape of the observed curves in a significant way. This similarity between Chicago clay and Pisa clay suggests that different soils could have a similar directionality relation for shear modulus.

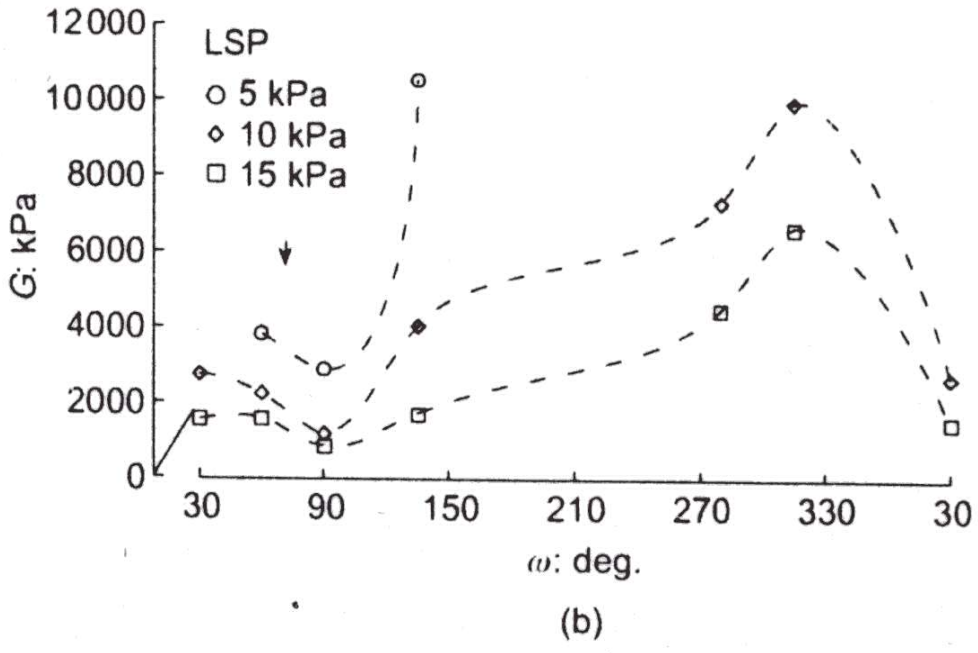

Fig. 3–16. Directionality relation of shear modulus observed on Pisa clay (Callisto and Calabresi 1998)

There is no need to specify any directionality relation for J_s because it is computed from G, in both the shear and compression zones.

Fig. 3–17 shows the directionality relations proposed for K_s and J_{vs}. Unlike G and J_s, K and J_v are mutually independent in most cases. Consequently, K is experimentally obtainable only in CQL ($\beta = 0$) and CQU ($\beta = \pi$) tests, while J_v is obtainable only in CPC ($\beta = 0.5\pi$) and CPE ($\beta = 1.5\pi$) tests. Hence, the relations proposed in Fig. 3–17 are mostly hypothetical. However, these hypotheses can be verified using this model to simulate relevant probe tests, as will be shown in Chapter 5.

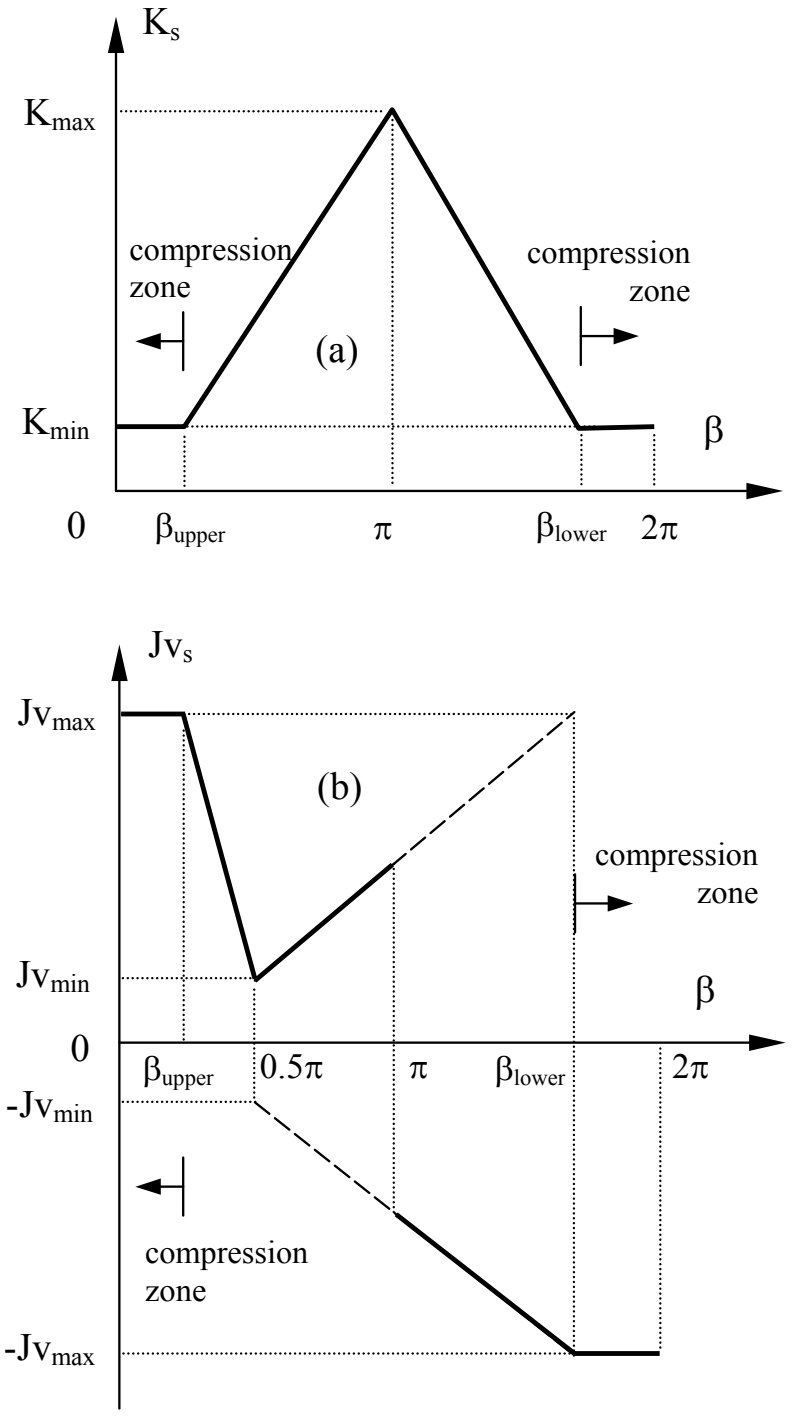

Fig. 3–17. Directionality relations for (a) K_s and (b) Jv_s

As shown in Fig. 3–17(b), the J_v-β relation is discontinuous at β = 0 and β = π. There are two main reasons for this discontinuity. First, Holman's tests (2005) have shown that the compressible Chicago clay undergoes contraction in both CPC and CPE tests, i.e., it is a shear-contractive material. Accordingly, J_v is positive for Δq > 0 and negative for Δq > 0. Suppose that the J_v-β relation is continuous on $\beta \in [0, 2\pi)$. Then, J_v must be zero at certain β, corresponding to an infinite volume change due to an infinitesimal change in q, which is certainly unrealistic. Hence, a discontinuous J_v-β relation is inevitable for materials showing consistent trend in volume change no matter how q changes, which is the case for most soils. The relation proposed in Fig. 3–17(b) basically assumes a consistent sign of J_v for either Δq > 0 (β < π) or Δq < 0 (β > π).

Directionality relations have been constructed for the threshold stiffness E^*_s, mainly based on experimental observations. In a similar way, directionality relations could be developed for the initial stiffness E^*_0, which, however, would double the number of model parameters. Instead, it is found in Holman's tests (2005) that the directionality relations $E^*_s(\beta)$ and $E^*_0(\beta)$ for the compressible Chicago clay differ by an approximate constant. For instance, Fig. 3–18 shows normalized G_0 and G_s curves with regard to β. As shown, G_0 and G_s have similar trend of dependency on β, meaning that the ratio $G_0(\beta)/G_s(\beta)$ maintains constant for $\beta \in [0, 2\pi)$. This similarity between the initial and threshold stiffness relations has been detected on K and J_v too, with similar ratios for different tangent moduli.

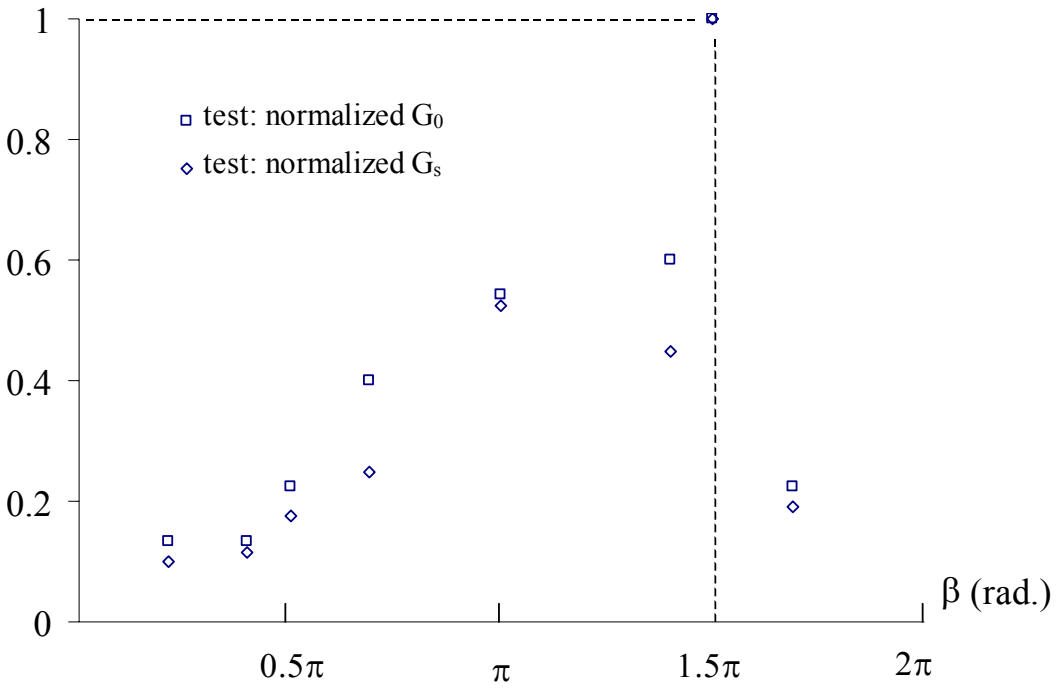

Fig. 3–18. Observed directionality relations for G_0 and G_s

Accordingly, a material parameter is introduced:

$$r_s^0 = \frac{G_0(\beta)}{G_s(\beta)} = \frac{K_0(\beta)}{K_s(\beta)} = \frac{Jv_0(\beta)}{Jv_s(\beta)}; \quad for \quad \beta \in [0, \ 2\pi) \tag{3.30}$$

For the compressible Chicago glacial clay, r^0_s ranges from 3 to 5. With this parameter, there is no need to specify $E^*_0(\beta)$, as long as $E^*_s(\beta)$ is known. It is worth noting that E^*_s values generally are more reliable than E^*_0 values, because the latter demand more accuracy in strain measurement.

Further reduction in parameter number is supported by another experimental observation that the ratio of E^*_{max}/E^*_{min} (cf. Fig. 3–15 and Fig. 3–17) remains an approximate constant for different $E^*_s(\beta)$ relations. Thus, another ratio is introduced as a material parameter:

$$r_x = G_{max} / G_{min} = K_{max} / K_{min} = Jv_{max} / Jv_{min} \quad \textbf{(3.31)}$$

For the compressible Chicago clay, r_x is approximately 5. With r_x, there is no need to specify E^*_{max} as long as E^*_{min} is known.

Fig. 3–19 shows the directionality relation proposed for LSP_s, with its minimum LSP_{min} at β_{RTC} (= 2.16) and its maximum LSP_{min} in a range from β_{RTE} (= 4.39) to 2π. The ratio between two extremes is designated by a constant.

$$r_{LSP} = LSP_{max} / LSP_{min} \quad \textbf{(3.32)}$$

Another directional parameter used in this model is μ, introduced in Eq. (3.23) to define the nonlinearity of stiffness degradation at medium to large strains. According to the test data, the dependence of μ on direction is such that μ equals a constant μ_C for any stress path leading to a failure where σ_{1f} is vertical, and equals another constant μ_E for any stress paths leading to a failure where σ_{1f} is horizontal, with σ_{1f} denoting the maximum principle stress at failure. The ratio of μ_C to μ_E is designated by a constant r_μ:

$$r_\mu = \mu_E / \mu_C \quad \textbf{(3.33)}$$

Note that unlike E^*_0, E^*_s and LSP_s, μ is only used in f_2(LSP) (cf. Eq. (3.23)), which is strictly applicable to the shear zone.

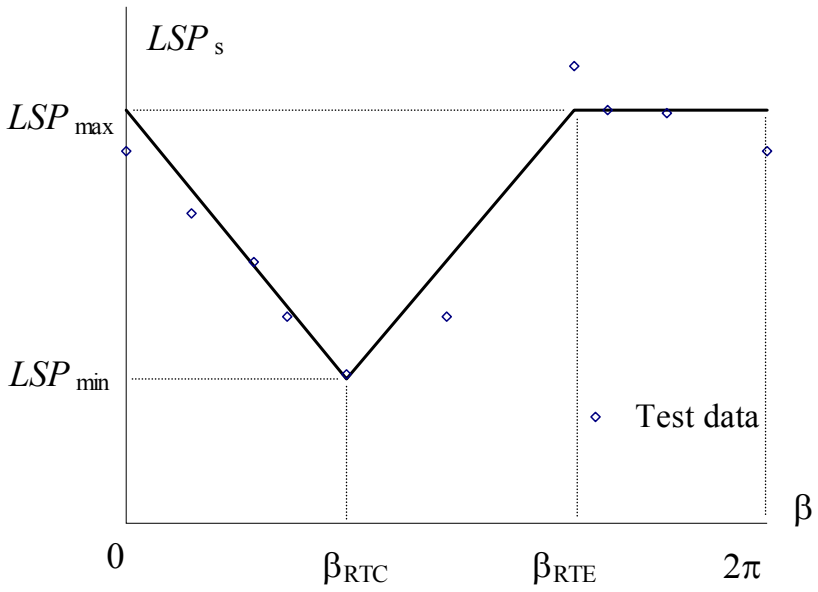

Fig. 3–19. Directionality relation for LSP_s

3.3.2.2 DIRECTIONALITY VS. PLASTICITY

Fig. 3–20 shows a stress-strain relation commonly observed in soils. This figure is a classical representation of plastic response, in the sense that irreversible strain has resulted from a loading-unloading cycle. To account for this phenomenon, the conventional approach of elasto-plasticity uses strain decomposition and provides different relations for elastic and plastic strain development. The directionality approach proposed herein attributes the irreversible strain to the variation of tangent stiffness with loading direction. As shown in Fig. 3–20, the irreversible strain develops because G^+, the shear modulus in the *loading* direction, is smaller than G^-, the shear modulus in the *unloading* direction. In other words, if directionality is properly addressed, the irreversible or plastic response can be reproduced naturally. In this sense,

directionality and plasticity are two different approaches to account for the same material behavior.

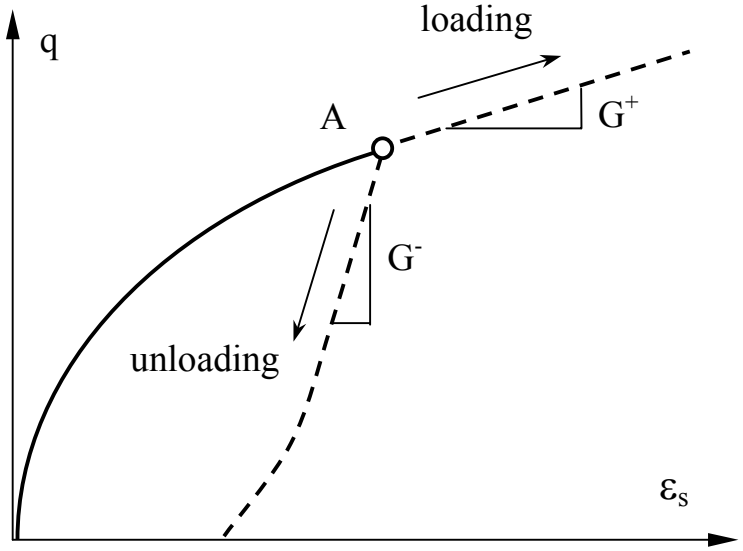

Fig. 3–20. Stiffness directionality vs. plasticity

Note that in situ soils could have an initial state similar to the state represented by the point A in Fig. 3–20. Therefore it is no surprise to observe the dependence of the initial stiffness E^*_0 on path direction. This variation of E^*_0 with path direction does not diminish no matter how low the strain level is reached to measure E^*_0. However, if a soil does exhibit purely elastic responses around initial state, e.g., as postulated by the Y_1 surface of the 3-loci hypothesis (Smith et al. 1992), one can accordingly use a constant function for $E^*_0(\beta)$, as a special case of this directionality approach.

3.3.2.3 DIRECTIONALITY & RECENT HISTORY EFFECT

The effect of recent stress history addresses the issue how the change in the stress path direction affects the subsequent soil stiffness. As shown in Fig. 3–21, the dashed curve represents a stress history up to point O, a current stress state. The arrow denoted by β represents a subsequent stress increment, while the arrow denoted by β_p represents the direction of the recent history. According to the hypothesis made by Atkinson et al.(1990), provided that β_p is fixed while β is allowed to vary in its entire range, soil stiffness tends to reach the maximum value if a complete stress reversal occurs, i.e., $|\beta - \beta_p| = \pi$, while the minimum value is encountered if there is no change in the path direction, i.e., $|\beta - \beta_p| = 0$.

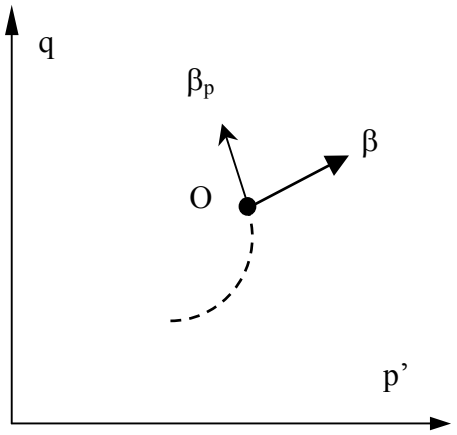

Fig. 3–21. Path direction of recent stress history

Though the recent history effect has not been included in the current model, it could lead to a further generalization of the proposed directionality relations, by taking into account stress history prior to the initial stress state. It is clear that the directionality relations proposed in this model refer to a unique stress history – k_0-consolidation. However, other stress histories also exist in reality, e.g. k_0-unloading due to water table raising in the recent geologic history. In the

probe tests performed on the Pisa clay (Callisto and Calabresi 1998) (cf. Section 3.3.2.1), the initial stress state was preceded by a segmented stress path composed of an isotropic consolidation and a triaxial compression shear (Callisto and Calabresi 1998), i.e., the recent history direction β_p corresponds to the shearing phase of a conventional triaxial compression test. But β_p corresponds to the k_0-consolidation in the tests conducted on the Chicago clay. This difference in the recent stress history might explain the slight but consistent difference in the β values corresponding to the two extreme stiffnesses, as mentioned earlier.

Though there is no other experimental justification, it is possible that the difference in the recent stress history, denoted by $\Delta\beta_r$, only makes the directionality relations shift along the β-axis by a certain amount, denoted by $\Delta\beta_s$. If this is the case, the recent history effect issue can be reduced to defining the relation between $\Delta\beta_r$ and $\Delta\beta_s$, with the directionality relations intact.

For the proposed model, the recent history effect essentially relates to how the directionality relations vary with the recent history. However, it is best to note that there exist contradictive experimental observations on this issue, and no conclusion has been reached to date among researchers (e.g. Clayton and Heymann 2001). More experimental investigation is required to fully understand this specific behavior. When definitive data exists, the current model could be changed to reflect the observations.

3.3.3 STRESS LEVEL DEPENDENCY

In the previous sections, relations for stiffness evolution and stiffness directionality have been proposed based on experimental observations, mainly from Holman's stress probe tests (2005). Note that those tests start from the same stress state. In many cases, one needs to make prediction for the same material, but with a different initial stress state. For instance, in different areas around the Chicago region, the compressible Chicago clay actually finds itself in various depth, corresponding to different in situ stress states. Under such a circumstance, one particularly needs a relation to describe the stiffness dependency upon the initial stress state, so that the material parameters obtained at a specific initial stress state can be employed under general conditions of arbitrary initial states.

Given the constitutive relations developed previously, the aforementioned problem can be mostly solved by making E^*_{min} (cf. Fig. 3–15 & Fig. 3–17) dependent upon the initial mean normal stress p'_0:

$$E^*_{\min} = E^*_{ref} p_a (p'_0 / p_a)^n \tag{3.34}$$

where p_a is atmospheric pressure (≈ 100 kPa). E^*_{ref} and n, both nondimensional, control the dependency of E^*_{min} on p'_0. This equation, with trivial variance in expression, has been commonly used for soils (e.g. Duncan et al. 1980). For most soils, n = 0.5 can be used as a default value (Pestana and Whittle 1999). According to Eq. (3.34), E^*_{ref} serve as material parameters, while E^*_{min} are treated as internal parameters, internally calculated by the model based on E^*_{ref} and p'_0. Ideally, dependency of E^*_{min} on q_0 should be taken into account as well.

However, it seems that no established relation is available in this aspect. Thus, q_0-dependency is left for future improvement.

Unlike E^*_{min}, the parameter LSP_{min} (cf. Eq. (3.32)) mostly depends on the time-dependent response of the material, as discussed in Section 3.3.1.3: There is no direct relation of LSP_{min} to the initial stress state, though this parameter might vary with different in situ conditions.

In this model, LSP_{min} is used as an internal parameter, which is related to a normalized parameter LSP_{ref} via the following definition:

$$LSP_{ref} = LSP_{\min} / p_a \tag{3.35}$$

Again, p_a is atmospheric pressure. The purpose is to achieve a nondimensional material parameter.

3.3.4 CRITERION FOR STRESS REVERSAL

Note that the constitutive equations of the proposed model are essentially founded on experimental observations made in monotonic loading paths, in which no abrupt change in stress path direction is encountered. In contrast, an unload-reload cycle in an oedometer test is made by switching between increasing and decreasing the vertical stress. Such stress reversals take place in cyclic loading and wave propagation. Apparently, the proposed model cannot be directly applied to these conditions, if proper modifications are not made. This section discusses the mechanism the directional stiffness model applies to deal with stress reversals.

To this end, the first issue is to define the stress reversal in a general way. In MIT-S1 model (Pestana and Whittle 1999), a stress reversal point is defined in strain using the following criterion:

$$\chi : \dot{\chi} < 0 \tag{3.36}$$

where the tensor χ is the accumulated strain from the last reversal point, $\dot{\chi}$ is the incremental strain, and the operator ":" represents a double contraction. Eq. (3.36) means a stress reversal occurs when χ and $\dot{\chi}$ are "opposite" in a tensor space. Basically, χ represents the overall strain path direction, which can be described in other ways. In a hypoplastic model (Niemunis and Herle 1997), the incremental strain is compared to the so-called intergranular strain to determine stress reversals. Following the basic idea behind these criteria, this model defines a stress reversal in terms of stresses:

$$(\sigma'_c - \sigma'_{rev}) : \dot{\sigma}' < 0 \tag{3.37}$$

where σ'_c is the current stress, σ'_{rev} corresponds to the last reversal point, and $\dot{\sigma}'$ is the incremental stress. Accordingly, this criterion uses (σ'_c- σ'_{rev}) to represent the overall stress path direction from the last reversal point.

In a perfect hysteretic elastic model (Hueckel and Nova 1979), stiffness evolution is described in terms of some measure of distance from the current state to the reversal point. This distance is zeroed whenever a new reversal point is found. Note that the stiffness evolution in the proposed model is described in terms of LSP, which is a distance between the initial and current stress states. In analogy to the hysteretic model, the definition of LSP can be slightly modified:

$$LSP = \int_{rev}^{c} \sqrt{(dp')^2 + (dq)^2} \tag{3.38}$$

where $\int_{rev}^{c}$ means an integral path identical to the actual stress path from the reversal stress σ'_{rev} to the current stress σ'_c. When a stress reversal is identified using Eq. (3.37), σ'_{rev} is updated by σ'_c and thus LSP is zeroed according to Eq. (3.38). That means small strain response will be invoked upon stress reversal. Though the small strain behaviour in the model is initially defined as a consequence of ageing, it is convenient to use the same definition to describe post-reversal behaviour. Though these two behaviours have different physical natures, they have similar patterns, i.e., an initial high stiffness and subsequent quick degradation.

In accordance with Eq. (3.38), the function of stress level dependency for E^*_{min} (cf. Eq. (3.34)) is modified as follows:

$$E^*_{\min} = E^*_{ref} p_a (p'_{rev} / p_a)^n \tag{3.39}$$

where p'_{rev} corresponds to σ'_{rev}. E^*_{min} will be updated every time that σ'_{rev} is updated, according to Eq. (3.39), and will be hold constant until next stress reversal occurs. Note that the initial tangent moduli E^*_0 can be derived from E^*_{min} using the directionality relations and the r^0_s ratio (cf. Section 3.3.2.1). At an initial state, or in a monotonic loading, $p'_{rev} = p'_0$, where p'_0 corresponds to the initial state. Therefore, Eqs. (3.38) and (3.39) do not alter model responses under monotonic loading conditions, only extend the model to stress reversals.

Note that the stress reversal criterion is different from the loading-unloading criterion in elastoplastic theory, which compares the incremental stress $\dot{\sigma}'$ to the local norm of the yield surface. At same stress point, the direction of the local norm is relatively constant in most

elastoplastic models, while the overall stress path, which is defined by (σ'_c- σ'_{rev}) in this model, might point to any direction in stress space. In fact, material responses upon stress reversals are closely related to the issue of recent history effect (Atkinson et al. 1990). Using Eqs. (3.38) and (3.39), the response generated by the model upon a stress reversal is generally stiffer than that in a monotonic loading, assuming that this comparison is made at same stress points and with identical incremental stresses. Therefore, this proposed model is expected generally to be consistent with the recent history effect suggested by Atkinson et al. (1990), though no special measure has been taken to address this issue.

3.4 MATERIAL PARAMETERS

Table 3-1 lists the thirteen material parameters involved in this model, description for each parameter, recommendations for their experimental determination, and values found from block samples of compressible Chicago clay. The over-consolidation ratio OCR is used to initialize LSP_p (cf. Fig. 3–12), an internal parameter in this model. For the compressible Chicago clay, also for most soils, the friction angle is not a pure constant. Instead, it is notably dependent on p'_f, which will be further discussed in the Chapter 5. The value listed in Table 3-1 is a mean value for the compressible Chicago clay.

The typical conventional properties of the Chicago clay are summarized in Table 3-2, as a comparison with the values listed in Table 3-1.

Table 3-1. Input parameters of the directional stiffness model

Parameters	Description	Tests	Chicago Clay
G_{ref}	Eq. (3.34)	triaxial/bender element	19
Jv_{ref}	Eq. (3.34)	triaxial	116
K_{ref}	Eq. (3.34)	triaxial	48
n	Eq. (3.34)	triaxial/bender element	0.55
r^0_s	Eq. (3.30)	triaxial	4
r_x	Eq. (3.31)	triaxial	5
LSP_{ref}	Eq. (3.35)	triaxial	0.07
r_{LSP}	Eq. (3.32)	triaxial	3
μ_C	Eq. (3.23)	triaxial	8
r_μ	Eq. (3.33)	triaxial	0.5
λ^*	Eq. (3.26)	triaxial/oedometer	0.04
OCR	over-consolidation ratio	triaxial/oedometer	1.5
ϕ	friction angle	triaxial test	28°

Table 3-2. Summary of conventional properties of compressible Chicago clay

Parameter	Value
Water content (%)	28.5
Liquid limit (%)	37
Plasticity index (%)	19
Void ratio	0.79
Unit weight (kN/m^3)	19.0
Limit pressure(kPa)	190
OCR	1.5

Most of these thirteen parameters can be estimated in triaxial probe tests. Using the directionality relations designated in the model, the value of $E^*_s(\beta_1)$ can always be derived from $E^*_s(\beta_2)$, where $\beta_1 \neq \beta_2$. Therefore, the ways to experimentally determine E^*_{ref} are essentially unlimited. As a general rule, it is recommended to choose stress probes approximating as closely as possible the loading conditions expected in the problem of interest. To analyze a supported excavation, for instance, the RTC and RTE tests are particularly desirable for parameter development, which well represent the loading conditions for soils behind the support wall and soils at the bottom of the excavation, respectively. More discussion will be given in Chapter 5.

A brief procedure of parameter development for this model is given here as an illustrative example. Triaxial probe tests with small strain measurements are performed in three path directions – CQL, CPC and CQU. Certainly, these tests are better performed from an initial stress state matching the in situ condition. The two G_s values from CPC and CQU tests are used to compute G_{ref}, G_{max}/G_{min} ratio and thus the r_x ratio. K_{ref} and Jv_{ref} are obtainable from CQL and CPC tests, respectively. The K_{max}/K_{min} ratio from CQL and CQU tests serves as a double check for r_x. If necessary, the mean value of G_{max}/G_{min} and K_{max}/K_{min} ratios should be used to estimate the r_x ratio. Two G_0/G_s values, two K_0/K_s values and one Jv_0/Jv_s value can be obtained from these three tests and the r^0_s ratio can be estimated using their average (cf. Eq. (3.30)). Given the directionality relation designated in Fig. 3–19, the LSP_s values from the three probe tests are more than adequate for estimating both LSP_{ref} and r_{LSP} values. The parameter μ_C can be estimated using the mean μ value obtained from the CQU and CPC tests. λ^* and OCR are

attainable from CQL test and ϕ can be measured in both CPC and CQU tests. Though the power n can be obtained using bender elements, abundant test data reported in literature have shown that n is close to 0.5 for most soils (e.g. Pestana and Whittle 1999).

The ratio r_μ cannot be obtained from the CQL, CPC and CQU tests. In the case that the material response in the extension side ($\Delta q < 0$) is of critical concern, performing an extra CPE test is highly recommended. Otherwise, it is feasible to simply use a default value, e.g. 2, for this parameter. In many cases, not all of the material parameters are critical. For instance, the compression index λ^* is generally irrelevant for simulation of shear tests, while the friction angle ϕ is mostly unimportant for simulating oedometer tests. Similar situations exist for the other parameters listed in Table 3-1. It largely depends on the goal of specific application and users' judgment which parameters deserve experimental exploration and which parameters can be roughly estimated.

The aforementioned procedure employs local small strain measurements to obtain E^*_0 values in triaxial tests, typically requiring mounting electronic units, e.g., LVDTs on the soil sample, which seems too cumbersome to become a common practice in the geotechnical industry. In the author's opinion, the E^*_s value is much more reliable than the E^*_0 value and is much more easily obtained, since detecting the "kink" does not require fine-resolution small strain data. Hence, if small strain behavior is not a big concern, it is recommended to use measured E^*_s values to estimate E^*_0 values, using an assumed r_x value. Alternatively, data from field seismic tests can be used to estimate G_0 value at $\beta = \pi/2$, as suggested by Holman (2005). Hopefully, when more

data become available regarding some of the parameters, e.g., r^0_s, r_x, r_{LSP} and r_μ, or relations between individual parameters, e.g., the G_{ref}/K_{ref} ratio, the required tests for the parameter development is expected to be reduced further. And the demand for an unconventional test technique, e.g., small strain measurement or bender element, is expected to be diminished.

4 NUMERICAL IMPLEMENTATION IN FINITE ELEMENT METHOD

This chapter describes how to implement the proposed model in the finite element method. First, a coupling system typically used to perform a non-linear finite element analysis is briefly explained. This coupling system is composed of two solution schemes – a global scheme and a local scheme. The proposed model is integrated in the local scheme, using a modified substepping method that not only has automatic error control, an advantage of the original substepping method, but also is capable of treating the incremental non-linearity caused by stiffness directionality, a new type of constitutive relation proposed in the preceding chapters. Furthermore, a consistent tangent matrix for the proposed model is derived to guarantee the stability and efficiency of the proposed substepping method. For this part, it is especially important to understand how the coupling system works, i.e., the interactions between the global and local solution schemes.

4.1 GLOBAL AND LOCAL SOLUTION SCHEMES

A static finite element method involves two basic types of equations – equilibrium equations and constitutive equations. For a concise discussion, an equilibrium equation can be expressed in a simplified weak form:

$$R_I(\boldsymbol{\sigma}_J, d_I) = f_I^{int}(\boldsymbol{\sigma}_J, d_I) - f_I^{ext}(d_I) = 0 \tag{4.1}$$

where the subscripts I and J denotes nodal number and Gauss point number, respectively. d_I, the nodal displacement, and $\boldsymbol{\sigma}_J$, the stress at a Gauss point, are unknown variables in this equation. An equilibrium demands the nodal *residual force* R_I to be zero, i.e., the *internal force* f^{int}_I equals the *external force* f^{ext}_I for each finite element node.

At a Gauss integration point, a constitutive equation can be generally expressed in the following form:

$$\dot{\boldsymbol{\sigma}}_J = \boldsymbol{\psi}(\dot{\boldsymbol{\varepsilon}}_J) \tag{4.2}$$

When a finite element analysis only involves simple elastic materials, i.e., an explicit relation between $\boldsymbol{\sigma}_J$ and $\boldsymbol{\varepsilon}_J$ can be obtained, the constitutive equation can be directly substituted into the equilibrium equations. Since $\boldsymbol{\varepsilon}_J$ can be expressed in terms of d_I, the only unknown in Eq. (4.1) becomes d_I. Therefore, the nodal displacements can be solved first and the stresses at Gauss points can be obtained subsequently. However, this approach does not work for general conditions involving non-linear materials. A general approach to solve the finite element equations uses a coupling system, in which a *global solution scheme* and a *local solution scheme* are interactively operated to obtain solutions.

The global solution scheme deals with the equilibrium equations. In this scheme, stresses are treated as constants, while nodal displacements are collectively updated using the global stiffness matrix. In most finite element packages, this updating typically is carried out using Newton method (Belytschko et al. 1999), especially for static analyses. Discussions herein are thus limited to the Newton method serving as the global scheme, though other numerical methods can be used as well (Abbo and Sloan 1996; Belytschko et al. 1999). An illustration of the Newton method is given in Fig. 4-1. In this figure, the superscript denotes the iteration number. $\boldsymbol{d}$ and $\boldsymbol{R}$ in the bold case are equivalent to R_I and d_I in Eq. (4.1), respectively. As shown, the computation starts from $\boldsymbol{d}^0$, and iteratively converges to the wanted solution $\boldsymbol{d}^*$. The iterative procedure essentially depends on the computation of the inverse of the Jacobian $\partial\boldsymbol{R}/\partial\boldsymbol{d}$.

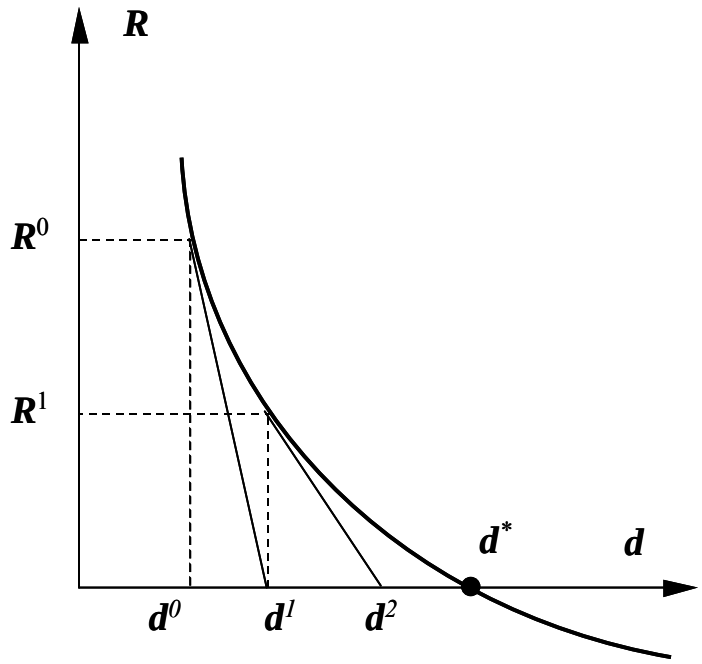

Fig. 4-1. Illustration of the Newton method

In a finite element analysis, numerical computation typically is executed in a number of *steps*, which are usually specified by users in most finite element codes. A finite step typically requires a computation to be executed in a number of small *increments*. In ABAQUS (2003), there is a set of criteria that automatically determines the magnitude of subsequent increment as a computation proceeds. For each increment, the Newton method is applied to obtain converged nodal displacements using a series of *iterations*, as shown in Fig. 4-1.

After the i^{th} Newton iteration is completed within the n^{th} increment, the global solution scheme computes $\Delta\boldsymbol{\varepsilon}^{i}_{n}$, the strain increment at a specific Gauss point, based on the updated nodal displacements. For discussions herein, the Gauss point number, denoted by the subscript J previously, is dropped for simplicity. $\Delta\boldsymbol{\varepsilon}^{i}_{n}$ is then passed to a local solution scheme, which accordingly integrates $\Delta\boldsymbol{\sigma}^{i}_{n}$, the stress increment, and $\Delta\boldsymbol{\xi}^{i}_{n}$, the increments of state variables, based on specified constitutive equations. Note that $\Delta\boldsymbol{\varepsilon}^{i}_{n}$, $\Delta\boldsymbol{\sigma}^{i}_{n}$ and $\Delta\boldsymbol{\xi}^{i}_{n}$ all refer to the same Gauss point. Such an integration is made one-by-one for each Gauss point in the entire finite element mesh. Hence, the local solution scheme is point-dependent. When different materials are involved in the same analysis, different local solution schemes can be used for different points. For a path-dependent material, which is the case for the compressible Chicago clay, the integration of $\Delta\boldsymbol{\sigma}^{i}_{n}$ and $\Delta\boldsymbol{\xi}^{i}_{n}$ should respectively start from $\boldsymbol{\sigma}_{n-1}$ and $\boldsymbol{\xi}_{n-1}$, the states converged in the $(n-1)^{th}$ increment. Accordingly, $\Delta\boldsymbol{\varepsilon}^{i}_{n}$ actually corresponds to the total change in nodal displacements counted from the 1^{st} to the i^{th} iteration within the current increment. This treatment can effectively avoid non-converged values of stress and state variables from

erroneously driving the constitutive equations (Belytschko et al. 1999) and it has been implemented in most commercial codes.

A main output of the local scheme, $\Delta\boldsymbol{\sigma}^{i}_{n}$, is then given back to the global scheme, wherein the corresponding stress is updated to $\boldsymbol{\sigma}^{i}_{n}$ and subsequently will be used as a constant in the $(i+1)^{th}$ Newton iteration. This procedure continues until converged solutions are achieved for the current increment, i.e. the n^{th} increment. Another important output of the local scheme is $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}$, a tangent stiffness matrix for the Newton global scheme to compute the Jacobian matrix $\partial\boldsymbol{R}/\partial\boldsymbol{d}$, which will be discussed in details later. The interaction between the global and local solution schemes is illustrated in Fig. 4-2, wherein the subscripts are consistent with those used in Eq. (4.1).

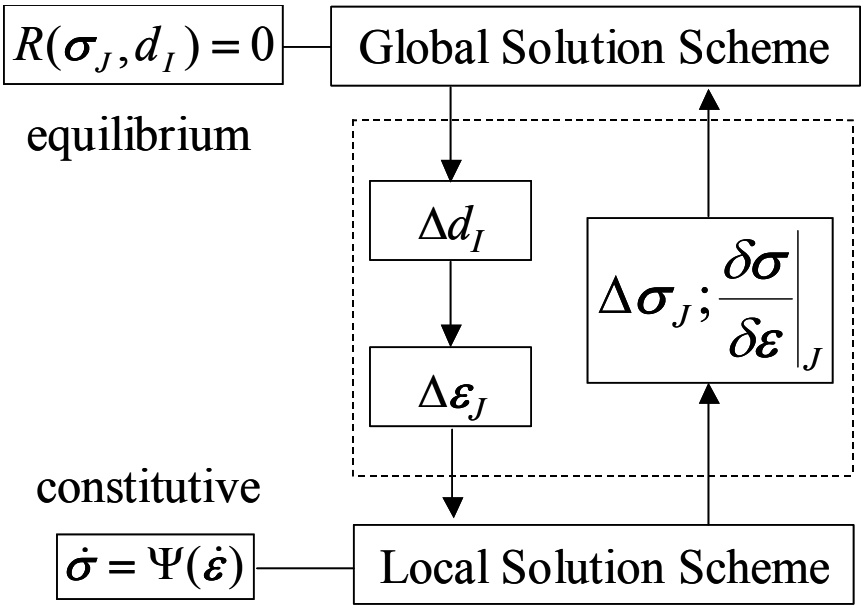

Fig. 4-2. Coupling system of the global and local solution schemes

Both implicit and explicit methods can be used in either global solution scheme or local solution scheme. Basically, an implicit method requires an inverse of Jacobian matrix while an explicit method does not. As stated previously, Newton method, which is an implicit method, is favored for the global scheme. In contrast, using which method for local solution scheme depends on specific model and does not have an established standard. The pros and cons of applying the two different methods to local solution schemes have been discussed in literature (Sloan 1987; Potts and Ganendra 1994; Luccioni et al. 2001; Zhao et al. 2005).

Note that whether an implicit or explicit method is used, the local solution scheme in a coupling solution system has the same computational sequence, i.e., $\boldsymbol{\sigma}_{n-1}$, $\boldsymbol{\xi}_{n-1}$ and $\Delta\boldsymbol{\varepsilon}^{i}_{n}$ are passed in, while $\Delta\boldsymbol{\sigma}^{i}_{n}$ and $\Delta\boldsymbol{\xi}^{i}_{n}$ are to be integrated. For the rest of this chapter, $\Delta\boldsymbol{\varepsilon}^{i}_{n}$ and $\Delta\boldsymbol{\sigma}^{i}_{n}$ are simplified as $\Delta\boldsymbol{\varepsilon}$ and $\Delta\boldsymbol{\sigma}$, respectively, if no explanation is otherwise given.

4.2 MODIFIED SUBSTEPPING SCHEME FOR STRESS INTEGRATION

This section describes a solution scheme developed for integration of the proposed model. This scheme is an improvement of the original substepping scheme with automatic error control, which is briefly introduced for general constitutive relations of a rate-form type. Then, the improved scheme is presented, with regard to the specific relations involved in the directional stiffness model, especially the stiffness directionality, to which the original substepping scheme does not directly apply.

4.2.1 AUTOMATIC SUBSTEPPING WITH ERROR CONTROL

The local solution scheme devised for the directional stiffness model is a modification of the automatic substepping with error control (Sloan 1987), which is an explicit method. To describe this scheme, it is necessary to give a brief description of the substepping method first.

This method divides $\Delta\boldsymbol{\varepsilon}$ into a number of *substeps* $\Delta\boldsymbol{\varepsilon}_k$ and integrates a constitutive relation in each $\Delta\boldsymbol{\varepsilon}_k$:

$$\Delta\boldsymbol{\varepsilon}_k = T_k \Delta\boldsymbol{\varepsilon}, \qquad \sum_1^m T_k = 1 \tag{4.3}$$

where subscript k is the substep number and m is the total number of substeps. T_k is a scalar no more than one. Integrations are made in two rounds within each $\Delta\boldsymbol{\varepsilon}_k$. For the first round:

$$\begin{aligned} \Delta\boldsymbol{\sigma}_k^1 &= \boldsymbol{E}(\boldsymbol{\sigma}_{k-1}, \xi_{k-1})\Delta\boldsymbol{\varepsilon}_k \\ \Delta\xi_k^1 &= \Xi(\boldsymbol{\sigma}_{k-1}, \xi_{k-1}, \Delta\boldsymbol{\varepsilon}_k) \end{aligned} \tag{4.4}$$

In the original substepping method (Sloan 1987), the tangent stiffness matrix **E** is a function of stress $\boldsymbol{\sigma}$ and a set of state variables $\boldsymbol{\xi}$, while Ξ is a known function to calculate increments of state variables. Note that $\boldsymbol{\sigma}_{k-1}$ and $\boldsymbol{\xi}_{k-1}$ are known in the k^{th} substep. For Eq. (4.4) basically is an forward Euler method, $\Delta\boldsymbol{\sigma}^1{}_k$ has a local truncation error of order $O(T_k^2)$. The results of the first round of integration are used in the second round:

$$\begin{aligned} \Delta\boldsymbol{\sigma}_k^2 &= \boldsymbol{E}(\boldsymbol{\sigma}_k^*, \xi_k^*)\Delta\boldsymbol{\varepsilon}_k \\ \Delta\xi_k^2 &= \Xi(\boldsymbol{\sigma}_k^*, \xi_k^*, \Delta\boldsymbol{\varepsilon}_k) \end{aligned} \tag{4.5}$$

where

$$\boldsymbol{\sigma}_k^* = \boldsymbol{\sigma}_{k-1} + \Delta\boldsymbol{\sigma}_k^1 \qquad \textbf{(4.6)}$$
$$\xi_k^* = \xi_{k-1} + \Delta\xi_k^1$$

$\Delta\boldsymbol{\sigma}^1{}_k$ and $\Delta\boldsymbol{\sigma}^2{}_k$ are illustrated in Fig. 4-3, wherein $\Delta\boldsymbol{\sigma}^1{}_k$ is an underestimate of stress increment while $\Delta\boldsymbol{\sigma}^2{}_k$ is an overestimate.

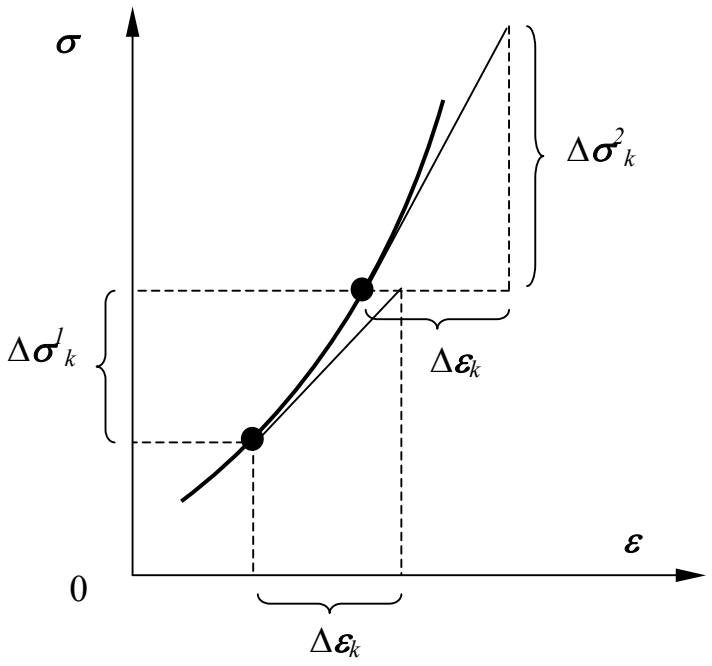

Fig. 4-3. Two stress integrations in each substep

A more accurate integration is thus given by:

$$\Delta\boldsymbol{\sigma}_k = (\Delta\boldsymbol{\sigma}_k^1 + \Delta\boldsymbol{\sigma}_k^2)/2 \qquad \textbf{(4.7)}$$
$$\Delta\boldsymbol{\xi}_k = (\Delta\boldsymbol{\xi}_k^1 + \Delta\boldsymbol{\xi}_k^2)/2$$

Using Taylor expansion, it can be shown that the truncation error of $\Delta\boldsymbol{\sigma}_k$ is $O(T_k^3)$ (e.g. Fellin and Ostermann 2002). This error can be approximated by:

$$\boldsymbol{Err}_k \approx (\Delta\boldsymbol{\sigma}_k^2 - \Delta\boldsymbol{\sigma}_k^1)/2 \tag{4.8}$$

$\boldsymbol{Err}_k$ is of order $O(T_k^2)$, serving as an upper bound estimate. Hence, the global error, which is calculated by a summation of $\boldsymbol{Err}_k$ over k = 1 to m and thus is of order $O(T_k)$, can always be controlled by limiting substep size T_k. To this end, a relative local error RE_k is defined and satisfies:

$$RE_k = \frac{\| \boldsymbol{Err}_k \|}{\| \boldsymbol{\sigma}_k \|} \le TOL_{RE} \tag{4.9}$$

Based on the fact that $\boldsymbol{Err}_k$ is of order $O(T_k^2)$, T_{k+1} can be estimated given T_k and RE_k:

$$T_{k+1} = 0.8\left(\frac{TOL_{RE}}{RE_k}\right)^{1/2} T_k \tag{4.10}$$

where 0.8 is a reduction factor, introduced for predicting T_{k+1} conservatively. To prevent an extrapolation from being carried too far, the T_{k+1} estimated by Eq. (4.10) is further constrained by:

$$0.1T_k \le T_{k+1} \le 2T_k \tag{4.11}$$

Eqs. (4.10) and (4.11) together enable an automatic procedure, in which integrations are made using Eqs. (4.4)~(4.7) while T_k is iteratively updated until Eq. (4.9) is satisfied. Then, Eq. (4.10) and Eq. (4.11) will provide an initial value of T_{k+1}, to initiate the procedure for the next substep. The entire process is started by setting $T_1 = 1$ and continues until $\sum T_k \ge 1$. Note that this error and substep size control is fundamentally similar to the Richardson extrapolation (Fellin and Ostermann 2002), though these two methods appear to have different forms.

Using the substepping method for stress integration has two main advantages. Beside the simplicity of numerical implementation for complex models (Zhao et al. 2005), substepping for the local solution scheme generally can reduce computational time for solving the global problem. In most finite element analyses, computation difficulties, typically caused by severe non-linearity in material or in geometry, usually only occur at a limited number of Gauss points. These difficulties will force the global scheme to use a smaller increment to proceed, if no substepping is allowed in the local scheme. This amounts to letting the most restrictive Gauss point control the global problem. In contrast, the automatic substepping method is able to increase the number of substeps only at those Gauss points with severe situations, without reducing the global time increment too much. In this way, the computational cost of more local substepping at limited Gauss points is marginal in comparison with the cost of more global iterations (Perez-Foguet et al. 2001).

4.2.2 A SUBSTEPPING SCHEME IMPROVED FOR INCREMENTAL NON-LINEARITY

In essence, the proposed directional stiffness model can be expressed in an incremental form:

$$\delta\boldsymbol{\sigma} = \boldsymbol{E}(\boldsymbol{\sigma}, LSP, \beta)\delta\boldsymbol{\varepsilon} \tag{4.12}$$

where $\mathbf{E}(\boldsymbol{\sigma}, \text{LSP}, \beta)$ is a generalized function for the tangent stiffness matrix, including the relations for stiffness evolution with LSP and the relations for dependency of stiffness on the path direction β. The current stress $\boldsymbol{\sigma}$ is needed for computing the internal variable LSP_f given β. An implicit form of $\mathbf{E}(\boldsymbol{\sigma}, \text{LSP}, \beta)$ has been given by Eq. (3.13).

Details of $\mathbf{E}(\boldsymbol{\sigma}, LSP, \beta)$ have been given in Chapter 3. But the discussion therein is not organized in a computational sequence. Before moving forward to the numerical scheme, it is worth giving a sequential outline of the procedure for $\mathbf{E}(\boldsymbol{\sigma}, LSP, \beta)$ calculation:

1.) Calculate E^*_0, E^*_s and LSP_s based on β (cf. **Fig. 3–15**, **Fig. 3–17** and **Fig. 3–19**);

2.) If $LSP < LSP_s$, calculate G, K and J_v using Eq. **(3.22)**, and J_s using Eq. **(3.5)**;

3.) If $LSP \geq LSP_s$, use Eq. **(3.17)** to determine whether the path direction falls in the shear zone or the compression zone (cf. **Fig. 3–4**). For the shear zone, calculate LSP_f using Eqs. **(3.18)~(3.20)**. Then calculate E^*, the four tangent moduli, according to Eqs. **(3.23)** and **(3.5)**. For the compression zone, calculate E^* using Eqs. **(3.27)~(3.29)**;

4.) Calculate variables A~E with the four tangent moduli using Eq. **(3.11)**, and then the tangent stiffness using Eq. **(3.13)**.

Given the form of Eq. (4.12), it is natural to integrate the proposed model using the strategy of the substepping method, which is especially suitable for an explicitly expressed tangent stiffness matrix, as shown in Eqs. (4.4) and (4.5). Substituting Eq. (4.12) in Eq. (4.4), the first round of integration in each substep is as follows:

$$\Delta\boldsymbol{\sigma}_k^1 = \boldsymbol{E}(\boldsymbol{\sigma}_{k-1}, LSP_{k-1}, \beta_k)\Delta\boldsymbol{\varepsilon}_k \qquad \textbf{(4.13)}$$
$$\Delta LSP_k^1 = \Delta LSP(\Delta\boldsymbol{\sigma}_k^1)$$

where the definition of ΔLSP is obtained by replacing each infinitesimal "δ" in (3.14) with finite increment "Δ". Similarly, stress can be integrated for the second round:

$$\Delta\boldsymbol{\sigma}_k^2 = \boldsymbol{E}(\boldsymbol{\sigma}_k^*, LSP_k^*, \beta_k)\Delta\boldsymbol{\varepsilon}_k \qquad \textbf{(4.14)}$$

where

$$\boldsymbol{\sigma}_k^* = \boldsymbol{\sigma}_{k-1} + \Delta\boldsymbol{\sigma}_k^1 \qquad \textbf{(4.15)}$$
$$LSP_k^* = LSP_{k-1} + \Delta LSP_k^1$$

Finally,

$$\Delta\boldsymbol{\sigma}_k = (\Delta\boldsymbol{\sigma}_k^1 + \Delta\boldsymbol{\sigma}_k^2)/2 \qquad \textbf{(4.16)}$$
$$\Delta LSP_k = \Delta LSP(\Delta\boldsymbol{\sigma}_k)$$

The error control and iterative procedure are the same as Eqs. (4.8)~(4.11).

Note that Eq. (4.13) uses β_k instead of β_{k-1}. Unlike $\boldsymbol{\sigma}$ and LSP, which are accumulated from the previous state, β only depends on the current state. If β_{k-1} is used in Eq. (4.13), there will be two main problems. For the first local substep within the first global step, i.e., k = 1 and n = 1, there is no simply way to make an initial guess of β_0. Furthermore, when a stress reversal is applied, i.e., β_k is quite different from β_{k-1}, an integration using β_{k-1} might produce totally erroneous results that lead to serious computational instabilities.

β_k can be expressed in terms of $\Delta\boldsymbol{\sigma}^1{}_k$ according to Eq. (3.15):

$$\beta_k = \beta(\Delta\boldsymbol{\sigma}_k^1) \qquad \textbf{(4.17)}$$

Eqs. (4.17) and (4.13) together form an implicit scheme. Instead of $\Delta\boldsymbol{\sigma}^1{}_k$, β_k is treated as the unknown to be solved, by substituting Eq. (4.13) in Eq. (4.17):

$$H(\beta_k) = \beta_k - \beta(\boldsymbol{E}(\boldsymbol{\sigma}_{k-1}, LSP_{k-1}, \beta_k)\Delta\boldsymbol{\varepsilon}_k) = 0 \qquad \textbf{(4.18)}$$

To solve Eq. (4.18), the Newton method is used. The derivative $\partial H/\partial\beta_k$ is obtained using numerical differentiation, since its analytical derivation is extremely tedious. During updates, it is guaranteed that β_k falls in the range $[0, 2\pi]$, by invoking the periodic relation $\beta_k = \beta_k + 2\pi N$

accordingly, where N is an integer. A flowchart of the subroutine for β calculation is shown in Fig. 4-4. There are two numerical parameters used in this subroutine - TOL_β and r_β. TOL_β, in rad., specifies the convergence criterion for the iterative procedure. A nondimensional quantity, r_β, designates the perturbance for numerically calculating $\partial H/\partial \beta_k$. For computations with the compressive Chicago clay, it is adequate that $TOL_\beta = 0.01$ and $r_\beta = 0.1$. In Fig. 4-4, β_{pre} is the path direction obtained in the previous substep.

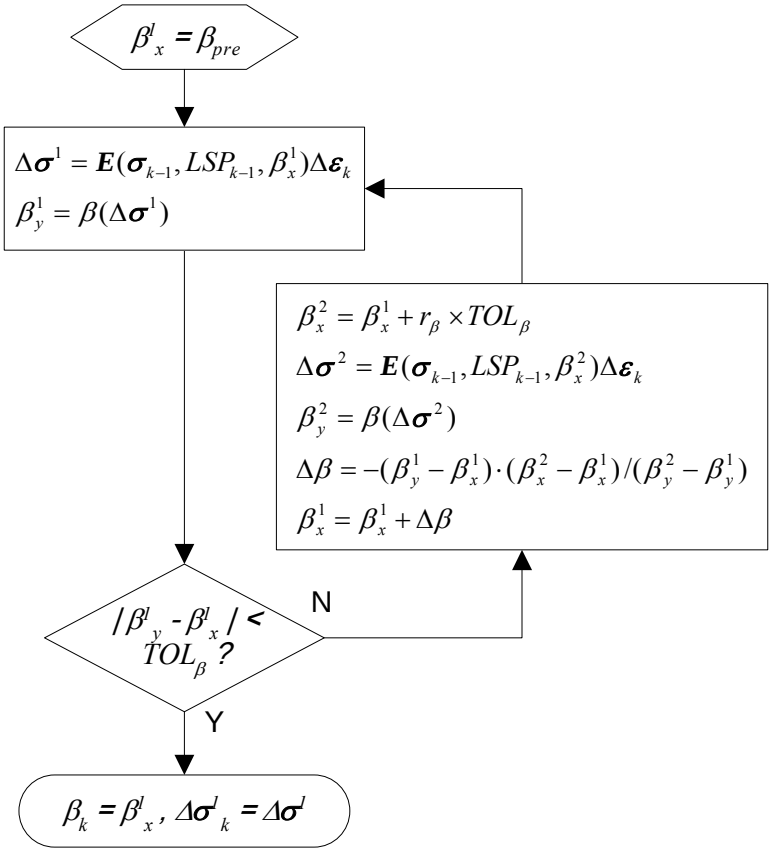

Fig. 4-4. Subroutine for β calculation

Note that β^*_k is approximated by β_k in Eq. (4.14), with $\beta^*_k = \beta(\Delta\boldsymbol{\sigma}^2_k)$. This approximation, made for saving computer time, is justified by the fact that $\Delta\boldsymbol{\sigma}^1_k$ is limited by the error control

and thus the difference between β_k and β^*_k tends to be negligible. Consequently, no extra solution scheme, like that for Eq. (4.18), is needed for the second round of integration.

Hence, the automatic substepping method with error control (Sloan 1987) can be slightly modified to integrate the proposed incrementally non-linear model. Specifically, the incremental non-linearity, or the path dependency, can be treated in the first round of integration in each substep. As a result, the path direction of the stress increment is determined. The remaining part of the solution scheme then iteratively chooses the substep size and computes the magnitude of the stress increment. The flowchart of this modified substepping scheme is shown in Fig. 4-5.

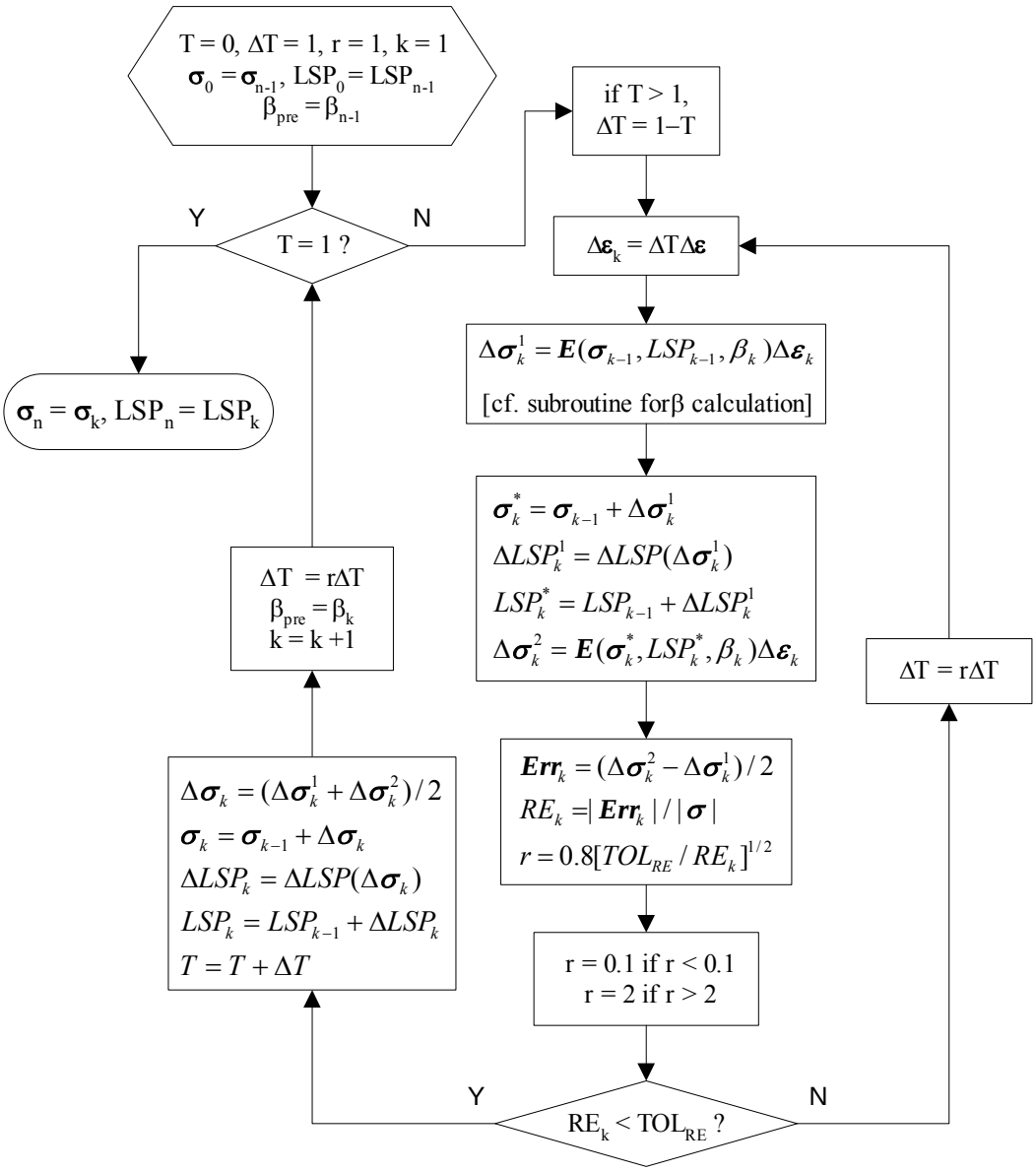

Fig. 4-5. Flowchart of the modified substepping scheme for the proposed model

4.3 ALGORITHMIC TANGENT MATRIX

4.3.1 DEFINITION

Newton method is an iterative solution procedure, the key part of which is a Jacobian matrix computed at the end of each iteration. When Newton method serves as a global solution scheme,

which is the case for discussions herein, computing this Jacobian matrix requires the local solution scheme to provide a tangent stiffness matrix for each Gauss point. This tangent stiffness matrix is a result of consistent linearization of finite element equations (Hughes and Pister 1978). Simo and Taylor (1985) pointed out that computing this matrix should be consistent with the integration algorithm used in the local solution scheme, which can be defined as follows with the notation used herein:

$$\boldsymbol{c}^{ATM} \equiv \left.\frac{\partial(\Delta\boldsymbol{\sigma})}{\partial(\Delta\boldsymbol{\varepsilon})}\right|_i \tag{4.19}$$

Again, $\Delta\boldsymbol{\varepsilon}$ is the strain increment calculated by the global scheme at the end of the i^{th} Newton iteration of the current n^{th} increment, while $\Delta\boldsymbol{\sigma}$ is the stress increment calculated by the local scheme based on $\Delta\boldsymbol{\varepsilon}$. "$\partial$" is differential operator. "$|_i$" means the derivative is evaluated at the end of the i^{th} iteration, because the Jacobian matrix in Newton method is always evaluated at the end of each iteration. $\mathbf{C}^{ATM}$ is the resulting matrix, herein referred to as *algorithmic tangent matrix* (*ATM*). According to Eq. (4.19), the key of computing $\mathbf{c}^{ATM}$ is to find the relation between $\Delta\boldsymbol{\sigma}^i_n$ and $\Delta\boldsymbol{\varepsilon}^i_n$, which not only depends on the constitutive equations but also on the algorithm used to integrate these equations.

The counterpart of *ATM* is a *continuum tangent matrix* (*CTM*), which is computed as:

$$\boldsymbol{c}^{CTM} \equiv \left.\frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\varepsilon}}\right|_i \tag{4.20}$$

wherein the relation between stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$ is described by the original constitutive equations. The difference between *ATM* and *CTM* is graphically demonstrated in Fig. 4-6,

wherein the stress-strain relation $\sigma(\varepsilon)$ is defined by the original constitutive equations, while $\Delta\sigma(\Delta\varepsilon)$ corresponds to the numerical integration of these equations by certain local solution scheme. Though these two relations ideally should be close to each other, their local tangents could be quite different, which means *ATM* and *CTM* could be quite different, as shown in the figure. Simo and Taylor (1985) showed that using *ATM* instead of *CTM* can achieve the expected quadratic convergence rate with a Newton global scheme. In essence, *ATM* complies with a consistent linearization of finite element equations, while *CTM* does not.

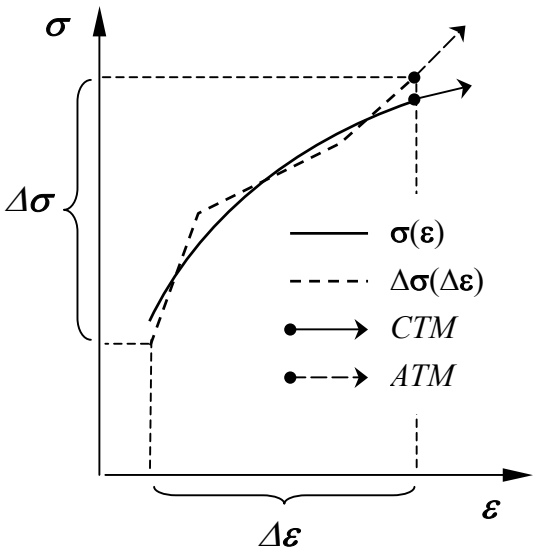

Fig. 4-6. Continuum tangent matrix (*CTM*) and algorithmic tangent matrix (*ATM*)

4.3.2 DERIVATION

Substituting Eqs. (4.13) and (4.14) in Eq. (4.16), the relation between $\Delta\sigma_k$ and $\Delta\varepsilon_k$ can be obtained:

$$\Delta\boldsymbol{\sigma}_k = \boldsymbol{c}_k \Delta\boldsymbol{\varepsilon}_k \tag{4.21}$$
$$\boldsymbol{c}_k = 0.5 \times [\boldsymbol{E}(LSP_k^*(\Delta\boldsymbol{\sigma}_k^1), \beta(\Delta\boldsymbol{\sigma}_k^1)) + \boldsymbol{E}(LSP_{k-1}, \beta(\Delta\boldsymbol{\sigma}_k^1))]$$

$\mathbf{c}_k$ is a secant stiffness matrix over a single substep, which can be easily calculated, for the two required matrices have already been calculated in Eq. (4.13) and Eq. (4.14), respectively. Accordingly, the stress-strain relation $\Delta\boldsymbol{\sigma}(\Delta\boldsymbol{\varepsilon})$ integrated with the automatic substepping can be determined with $\mathbf{c}_k$ known for each substep, as shown in Fig. 4-7.

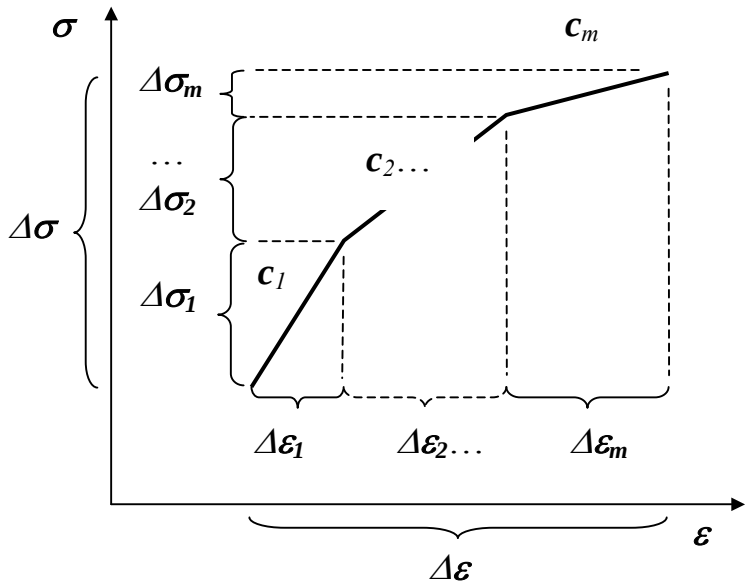

Fig. 4-7. Stress integration with substepping over entire strain increment

According to Eq. (4.19), *ATM* in essence is the local tangent at the end of the $\Delta\boldsymbol{\sigma}(\Delta\boldsymbol{\varepsilon})$ curve. Hence, to calculate the ATM for a model integrated by substepping, what really matters is the relation $\Delta\boldsymbol{\sigma}_m(\Delta\boldsymbol{\varepsilon}_m)$ at the last substep, rather than the entire relation $\Delta\boldsymbol{\sigma}(\Delta\boldsymbol{\varepsilon})$, i.e.

$$\boldsymbol{c}^{ATM} \equiv \left.\frac{\partial(\Delta\boldsymbol{\sigma}_m)}{\partial(\Delta\boldsymbol{\varepsilon}_m)}\right|_m \tag{4.22}$$

where "$|_m$" means the derivative is evaluated at the end of the m^{th} substep, the last one. Note that Eq. (4.22) is applicable to any model integrated with a substepping scheme, not limited to a specific model. In Perez-Foguet et al.(2001), an expression of *ATM* for substepping scheme is derived directly based on Eq. (4.19), which takes every substep into account and involves m times of matrix inverse. In contrast, the point presented herein suggests that only the last substep is relevant to *ATM*, because the differential operator "∂", in either Eq. (4.19) or Eq. (4.20), computes the local tangent at a specified evaluation point. The details of the differentiated function at any point other than the evaluation point basically have nothing to do with the derivative at the evaluation point, as illustrated in Fig. 4-6. Substituting Eq. (4.21) in Eq. (4.22), the expression of the *ATM* for substepping results:

$$\boldsymbol{c}^{ATM} = \boldsymbol{c}_m + \frac{\partial \boldsymbol{c}_m}{\partial(\Delta\boldsymbol{\varepsilon}_m)}\Delta\boldsymbol{\varepsilon}_m \tag{4.23}$$

$\mathbf{c}_m$ is calculated according to Eq. (4.21). The derivative of $\mathbf{c}_m$ with respect to $\Delta\boldsymbol{\varepsilon}_m$ is governed by the local non-linearity of the constitutive relation and is independent of the size of $\Delta\boldsymbol{\varepsilon}_m$. This derivative is difficult to obtain, due to the complexities of the constitutive equations. However, it is noted that $\Delta\boldsymbol{\varepsilon}_m$ generally is quite small, thanks to the error control of the automatic substepping method, i.e., a conservatively small value usually assigned for TOL_{RE} in Eq. (4.9). Consequently, the second term on the right hand side of Eq. (4.23) can be conveniently neglected:

$$\boldsymbol{c}^{ATM} \approx \boldsymbol{c}_m \tag{4.24}$$

Apparently, the smaller the $\Delta\boldsymbol{\varepsilon}_m$ is, the more accurate the computed *ATM* will be. On the one hand, the accuracy is always achievable by controlling the TOL_{RE} value, though a smaller TOL_{RE}

value means more computation time. This possible extra cost in computation time is justified, especially when computing the *ATM* must otherwise resort to numerical differentiation (e.g. Fellin and Ostermann 2002). On the other hand, a "minor" deviation from the true *ATM* is not expected to impair Newton method's efficiency, though a totally irrelevant substitute for *ATM* could make the solution scheme never converge. The well-known quasi-Newton method could serve as an example of what a "minor" deviation could be, in which the *ATM* computed for one iteration is repeatedly used for several subsequent iterations so that the overall time-efficiency can be enhanced by skipping some computations of matrix inverse. That means the Newton method itself allows certain amount of inaccuracy in computing the *ATM*, which serves as another justification for Eq. (4.24).

4.3.3 TESTING OF CONVERGENCE RATE

To evaluate the convergence rate resulting from the algorithmic tangent matrix derived in the previous section, a series of computations with ABAQUS (2003) were performed on single elements under various load conditions. The computations include a uniaxial loading test, a uniaxial unloading test, a biaxial compression test and a reduced biaxial extension test. The boundary conditions and load conditions for each simulated test are illustrated in Fig. 4-8. All of these computations start from the same initial state, with the vertical stress σ'_{v0} = 133 kPa and the horizontal stress σ'_{h0} = 74.5 kPa. In these computations, both TOL_{RE} (Eq. (4.9)) and TOL_{β} (Fig. 4-4) are equal to 0.001.

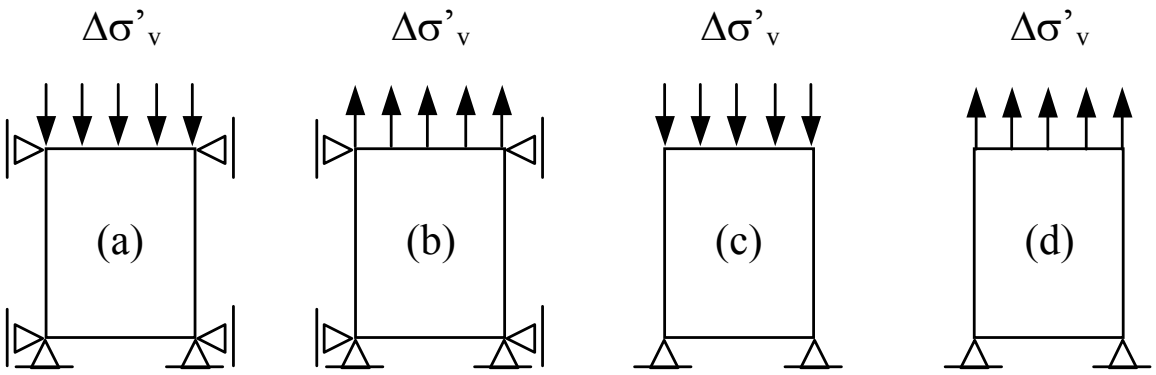

Fig. 4-8. Computations for convergence rate evaluation: (a) uniaxial loading, (b) uniaxial unloading, (c) biaxial compression, and (d) reduced biaxial extension

The convergence information obtained from the computed tests is listed in Table 4-1 to Table 4-4. As introduced in Section 4.1, the residual force is the difference between the internal force and external force at a specific finite element node (cf. Eq. (4.1)). As illustrated in Fig. 4-1, the residual force is evaluated at each iteration in a Newton scheme. The "residual force" columns within the tables show "largest residual force" values recorded by ABAQUS, which can also be obtained in the ABAQUS Message file or via the Visualization module of CAE. The *reduction ratio* is the absolute value of the ratio of the current residual force to the previous residual force. The smaller the reduction ratio, the faster the Newton scheme will converge.

Table 4-1. Convergence of the computed uniaxial loading test

	increment 1		increment 10		increment 20	
iteration #	residual force	reduction rate	residual force	reduction rate	residual force	reduction rate
1	-5.58E+00	--	1.04E-01		-1.18E-01	
2	-6.04E-01	0.108	6.95E-04	0.0067	6.31E-04	0.0053
3	-1.18E-02	0.020	4.67E-06	0.0067	-3.38E-06	0.0054
4	-2.37E-04	0.020	3.14E-08	0.0067	2.25E-08	0.0066
5	-4.74E-06	0.020	--	--	--	--
6	-9.47E-08	0.020	--	--	--	--

Table 4-2. Convergence of the computed uniaxial unloading test

	increment 1		increment 10		increment 20	
iteration #	residual force	reduction rate	residual force	reduction rate	residual force	reduction rate
1	-8.50E-01	--	1.90E-05		2.23E-05	
2	-2.00E-03	0.0024	9.32E-08	0.0049	1.19E-07	0.0053
3	-4.64E-06	0.0023	4.58E-10	0.0049	6.34E-10	0.0053
4	-1.07E-08	0.0023	2.27E-12	0.0050	3.32E-12	0.0052
5	-2.30E-11	0.0021	--	--	--	--

Table 4-3. Convergence of the computed biaxial compression test

	increment 1		increment 10		increment 20	
iteration #	residual force	reduction rate	residual force	reduction rate	residual force	reduction rate
1	5.67E-02	--	1.28E-04		1.50E-04	
2	1.71E-02	0.30	-1.10E-07	0.00086	-1.46E-07	0.00098
3	4.01E-03	0.23	-3.83E-10	0.00349	-5.51E-10	0.00376
4	9.04E-04	0.23	--	--	--	--
5	2.03E-04	0.22	--	--	--	--
6	4.55E-05	0.22	--	--	--	--
7	1.02E-05	0.22	--	--	--	--
8	2.28E-06	0.22	--	--	--	--
9	5.12E-07	0.22	--	--	--	--

Table 4-4. Convergence of the computed reduced biaxial extension test

	increment 1		increment 10		increment 20	
iteration #	residual force	reduction rate	residual force	reduction rate	residual force	reduction rate
1	-0.918	--	1.54E-05		-2.18E-04	
2	3.93E-02	0.04	6.43E-08	0.00419	-7.99E-07	0.00367
3	-1.05E-02	0.27	2.69E-10	0.00419	-3.62E-09	0.00453
4	2.21E-03	0.21	--	--	--	--
5	-4.97E-04	0.22	--	--	--	--
6	1.11E-04	0.22	--	--	--	--
7	-2.47E-05	0.22	--	--	--	--
8	5.50E-06	0.22	--	--	--	--
9	-1.23E-06	0.22	--	--	--	--
10	2.73E-07	0.22	--	--	--	--

Three major observations can be made from Table 4-1 to Table 4-4. First, the reduction ratio tends to be constant for iterations within the same increment, indicating that the proposed algorithmic tangent matrix results in a linear convergence rate. As mentioned previously, a rigorous ATM should lead to a quadratic convergence rate for a Newton global scheme (e.g. Simo and Taylor 1985). If the force residual of the previous iteration is on the order of 1E-n, for instance, the force residual of the current iteration will be on the order of 1E-2n, with n being a positive integer. Apparently, a linear convergence rate is slower than a quadratic convergence rate. This reduction in the order of convergence rate is mainly due to the approximation made in the proposed ATM (cf. Eq. (4.24)).

Secondly, the reduction rate tends to be low in the first increment under the same load condition. Furthermore, the reduction rate for the initial increment is found dependent upon load

conditions, or stress paths. As shown in the tables, the reduction rate of the initial increment is lower in the uniaxial tests than in the biaxial tests.

In most increments other than the first one, the reduction rate tends to be relatively constant and independent of load conditions. As shown in the tables, this rate is about 0.005 for the compressible Chicago clay. Note that ABAQUS recommends using the control parameter R^{α}_{n} = 0.005 as the convergence criterion for most engineering applications, with R^{α}_{n} defined as the ratio of the largest residual force to the average force over a finite element mesh. For the computations presented herein, the average force is on the order of 100 kPa. With the reduction rate of 0.005, a convergence with residual forces less than 1E-6 kPa can be achieved within only two to three iterations, as shown in the tables. With such a small reduction rate, the difference between a linear convergence rate and a quadratic convergence rate becomes insignificant, especially from an engineering point of view.

It is worth noting that a linear convergence rate guarantees computational stability, as long as the reduction rate is smaller than unit, because the residual force can always be reduced to a desired level within a finite number of iterations and thus divergence generally is not expected.

4.3.4 DISCUSSIONS

The previous sections basically state that the *ATM* associated with the substepping scheme has a simple form and an extremely low computational cost, with only a slight approximation. Furthermore, this proposed ATM leads to a linear convergence rate with a mostly small

reduction rate, which sustains an efficient and stable numerical computation. However, the substepping scheme, with the exhibited advantages, cannot be directly applied to elastoplastic models, because the stresses integrated thereby may diverge from the yield surface (Crisfield 1991; Belytschko et al. 1999), a problem commonly known as yield surface "drift". That is the main reason why most elastoplastic models have been integrated using implicit methods (Simo and Hughes 1998; Belytschko et al. 1999). That in turn makes it generally difficult to compute the corresponding *ATM* (Zhao et al. 2005), because an already complex relation $\sigma(\varepsilon)$ and an implicit method blended togather tend to make the resulting relation $\Delta\sigma(\Delta\varepsilon)$ even more complex and implicit. It is not unusual to come across a situation where computing the ATM turns out infeasible. A remedy for such a circumstance is numerical differentiation (Miehe 1996; Perez-Foguet et al. 2000a; Perez-Foguet et al. 2000b), which, however, requires a substantial extra amount of computations and thus is not quite time-efficient.

Rather than resort to implicit methods, the yield surface "drift" can be remedied using correction algorithms in an explicit scheme (Potts and Gens 1985; Sloan 1987), by which diverted stresses are iteratively sent back to the yield surface. These extra iterations alter the relation $\Delta\sigma_k(\Delta\varepsilon_k)$ defined in Eq. (4.21) and the local secant stiffness $\mathbf{c}_k$ is not explicit any more. Consequently, how to derive *ATM* becomes an issue for the corrected substepping method.

Apparently, the difficulties of computing the *ATM* for an elastoplastic model mostly originate from the yield surface. Though defining yield surface is straightforward for metals, it is generally difficult for soils. Essentially, recent research on soil modeling, such as the bounding surface model (Dafalias and Herrmann 1982; Whittle and Kavvadas 1994), the multi-surface

kinematic hardening model (Al-Tabbaa and Wood 1989; Stallebrass and Taylor 1997) and the hypoplastic model (Viggiani and Tamagnini 1999; Kolymbas 2000), also confirmed that classical elastoplasticity theory is not quite suitable for modeling soil behavior. Under this backdrop, the directional stiffness model is proposed as an alternative approach to modeling soils, in which material tangent stiffness is explicitly defined while plastic responses are generated without the need for a yield surface. In addition to other advantages of this model, as elaborated in Chapter 2 and Chapter 3, the numerical implementation of this model, including the stress integration and the *ATM* computation, as discussed previously, is remarkably simply, especially in comparison with an elastoplastic model.

5 MODEL TESTING

The numerical scheme proposed in the preceding chapter has been coded in FORTRAN and employed in a series of computations for testing and verification purposes. In these computations, two forms of this code were used. The first one is an executable file generated by a FORTRAN compiler, using an input file containing incremental strain data prepared by users. The second one has the same numerical algorithm as the first one but was coded in the UMAT file of ABAQUS (ABAQUS_Inc. 2003), with input data passed in from the finite element program. The constitutive relations and numerical algorithms implemented in these two forms are completely identical. Using which code is dependent on convenience.

The model has been tested under various conditions, including drained/undrained triaxial tests, drained/undrained biaxial tests and an oedometer test involving an unload-reload cycle. It was additionally used to compute ground movement at the Lurie site, a well-instrumented deep supported excavation in downtown Chicago. In all cases, the input parameters for the model were developed from drained triaxial probe tests performed on block samples obtained at the Lurie site, as listed in Table 3-1. To simulate the drained triaxial probe tests, the executable version of the model was used for convenience and the simulations were strain-controlled. All

other computations were performed using ABAQUS, wherein the laboratory test simulations were stress-controlled and single elements were used to represent soil specimens.

Due to a limited sample quantity, the soil specimens used in the simulated tests were not obtained from the same site, though all of them are regarded as compressible Chicago glacial clays. Table 5-1 shows where and from which geologic stratum the specimens used in each type of test were obtained. Project descriptions of Lurie Center, Ford Center and Block 37 are given by Rechea-Bernal (2006), Blackburn (2005) and Morgan (2006), respectively. Geologic classifications and material properties of soil strata in Chicago area are discussed by Chung and Finno (1992).

Table 5-1. Laboratory tests used to verify model

Type of test	Site of sampling	Geologic stratum
Drained triaxial tests	Lurie Center	Blodgett
Drained biaxial test	Ford Center	Upper Deerfield
Oedometer test	Lurie Center	Blodgett
Undrained triaxial tests	Block 37	Blodgett
Undrained biaxial tests	Ford Center	Upper Deerfield

Note: All computations based on parameters obtained from drained triaxial probe tests on Lurie block specimens

The main purpose of the laboratory test simulations presented herein is to reveal model responses under various boundary conditions, though computed results and experimental data are compared for most cases. In light of the difference in sample source and the fact that all computations were

carried out with material parameters developed from stress probes on the Lurie block specimens, care must be taken when reviewing and interpreting both the computed and experimental results.

While the soil samples used in the drained triaxial probe tests were obtained from the Lurie Center site, the modeling of this excavation using parameters developed on the triaxial probe tests constituted a class C prediction (Lambe 1973), in light of the fact that the computation was performed after the excavation was completed.

5.1 DRAINED AXISYMMETRIC CONDITIONS

As elaborated in Chapter 3, the development of the directional stiffness model is mostly based on the observations made in the drained triaxial probe tests. Therefore, a first step to verify both the model formulation and implementation is to reproduce soil responses observed in these tests. In this way, the validity of the constitutive equations can be verified and the feasibility of the improved substepping method can be tested. Furthermore, the proposed constitutive relation involves several assumptions, e.g., the directionality relations proposed for K and J_v, and these can be evaluated from the simulations. All simulations were conducted with the parameters shown in Table 3-1.

5.1.1 STRESS PATHS

Fig. 5–1 shows the axial strain, ε_a, and the radial strain, ε_r, measured in each probe test. In the simulation, the numerical model uses these strain data as inputs, and computes stress components σ'_a and σ'_r accordingly.

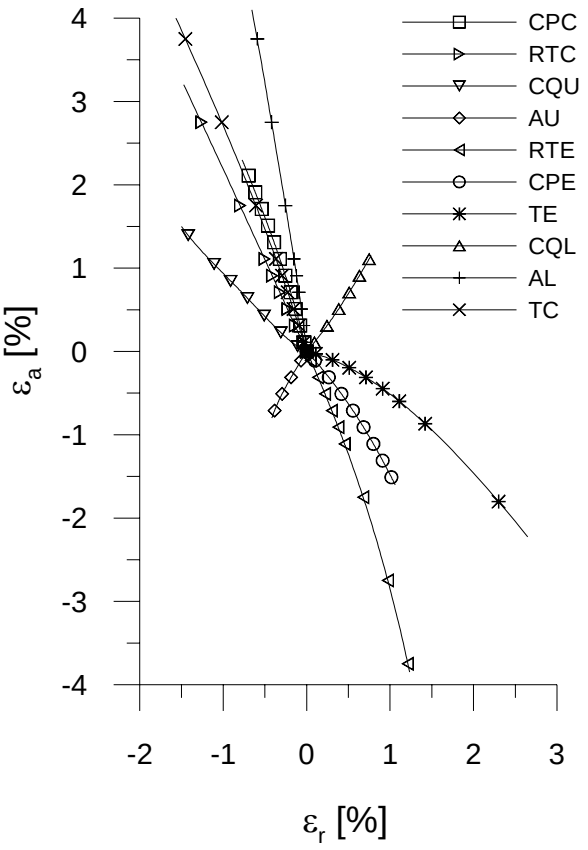

Fig. 5–1. Measured strains input to the numerical model

Fig. 5–2 shows the computed stress paths in comparison with those actually applied in the tests, where $p' = (\sigma'_a+2\sigma'_r)/3$ and $q = \sigma'_a-\sigma'_r$. As shown, the computed path generally matches the experimental one for each probe. The most perceivable difference occurs in the TE test, mainly due to a slight but constant deviation persisting for a significant length of stress path. In

essence, Fig. 5–2 indicates that the numerical model is successful in finding the right path direction in each simulation. Note that in those stress paths wherein both $\Delta p'$ and Δq are nonzero, e.g., TC, RTC, AU, RTE, TE and AL, the directionality relations for K and J_v are essentially assumed without experimental validations. These assumptions are justified partially by the model's success in reproducing the stress paths.

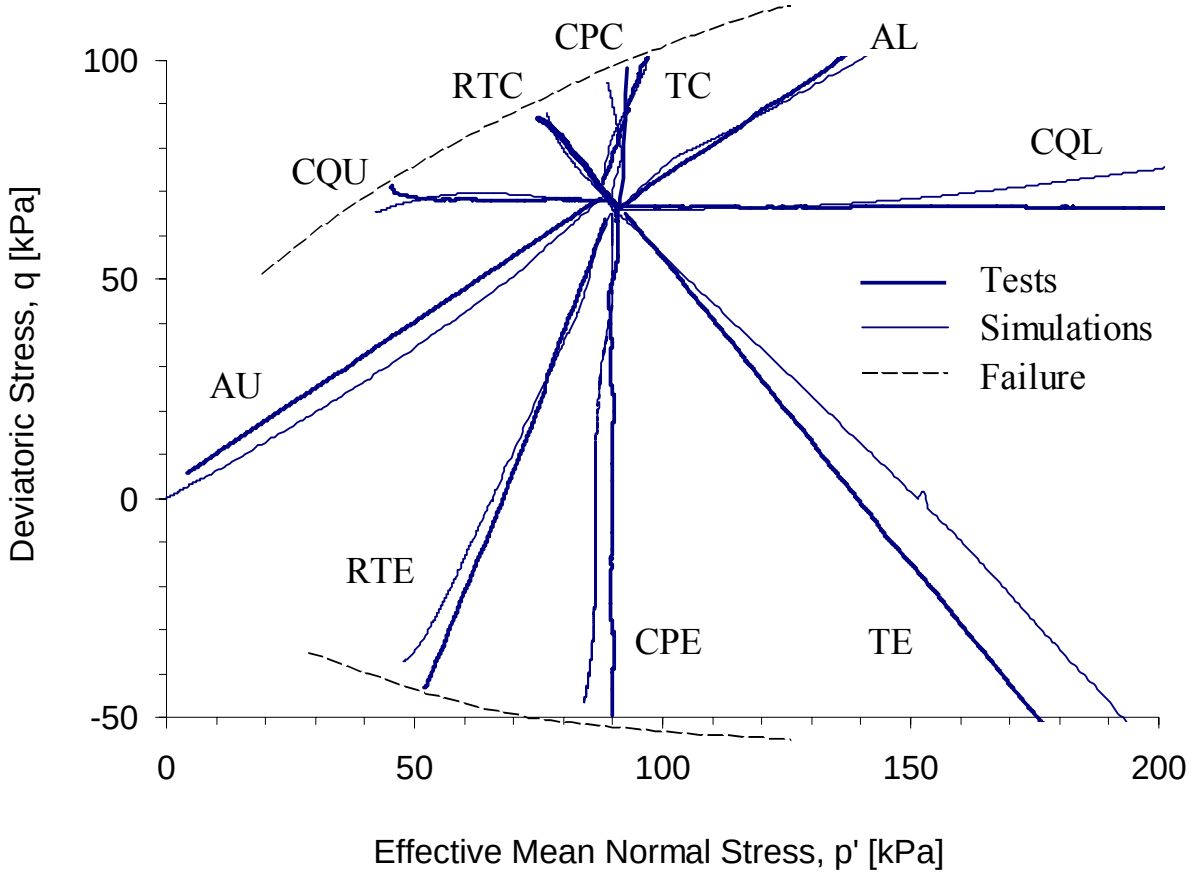

Fig. 5–2. Simulated stress paths of the triaxial probe tests

The proposed model uses one failure parameter, i.e., the friction angle ϕ, to define the failure condition. However, the experimental results show that ϕ varies with the path direction. For

example, ϕ measured in CQU test is 39°, while the value measured in TC is 27°. In fact, most shear tests shown in Fig. 5–2 are close to failure. Therefore, the tips of those stress paths roughly outline a failure envelope for the soil, indicating that the friction angle ϕ tends to decrease with p'_f, the effective mean normal stress at failure. In the simulating these tests, ϕ values are assigned on a case-by-case basis, instead of using a single value, which is a limitation of the current model. However, it is not difficult to incorporate a p'_f-dependent relation for ϕ in this model, as long as it is established on a solid experimental basis.

5.1.2 SHEAR RESPONSES

Fig. 5–3 shows the simulated and observed shear responses in the tests with σ_{1f} vertical and $\Delta q > 0$, and the tests with σ_{1f} horizontal and $\Delta q < 0$. The displayed responses at a strain level less than 1% emphasize the small-intermediate strain behaviours. Note the deviatoric stress change, Δq, conventionally is plotted versus the shear strain ε_s (= $2(\varepsilon'_a - \varepsilon'_r)/3$), which by no means indicates ε_s is solely caused by Δq. As elaborated in Section 3.1.1.2, for frictional materials like soils, changes in the stress ratio η (=q/p'), fundamentally results in shear strains. Accordingly, the slopes of the curves in Fig. 5–3 do not truly reveal the tangent shear modulus G. For instance, although the stress-strain curves in Fig. 5–3(a) are very close to one another in the small strain range less than 0.1%, the corresponding values of the initial shear modulus G_0 are quite different, for example, G_0 for the RTC path is about twice the value for the TC path. In another case, the small strain response in the RTE path seems stiffer than that in the CPE path,

while the G_0 value turns out being larger in the CPE path. Hence, the curves in Fig. 5–3 merely serve as a tool to present and compare the material and model responses. As shown, overall matches between the simulations and observations are satisfactory.

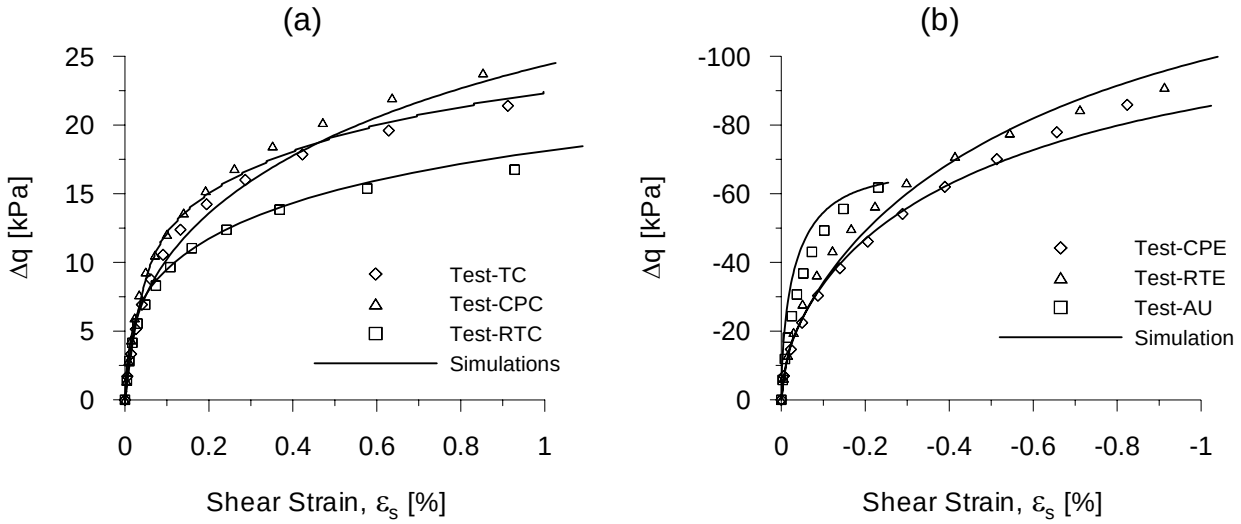

Fig. 5–3. Simulated and observed shear responses in (a) compression tests, and (b) extension tests.

Note the relatively small magnitude of shear strain developed in the AU test (Fig. 5–3b) wherein η is essentially kept constant, and the significant amount of shear strain developed in the CQU test (Fig. 5–5a) wherein q is kept constant. These observations serve as two important pieces of evidence in support of the fundamental notion behind this model – the stress ratio η governs shear responses of frictional materials (cf. Section 3.1.1.2).

5.1.3 VOLUMETRIC RESPONSES

Fig. 5–4 shows the simulated and observed volumetric responses in the *loading tests* wherein $\Delta p' > 0$, and the *unloading tests* wherein $\Delta p' < 0$. As explained in Section 3.1.1, the volumetric strain ε_v is not only caused by $\Delta p'$ via the bulk modulus K, but also by Δq via the coupling modulus J_v. Thus, plotting $\Delta p'$ along with ε_v is merely a form of data presentation, not inferring any exclusive relation between these two quantities. For instance, the RTC curve in Fig. 5–4(b) does not indicate a negative K. In fact, the Δq-induced volume contraction exceeds the $\Delta p'$-induced volume dilation in this probe.

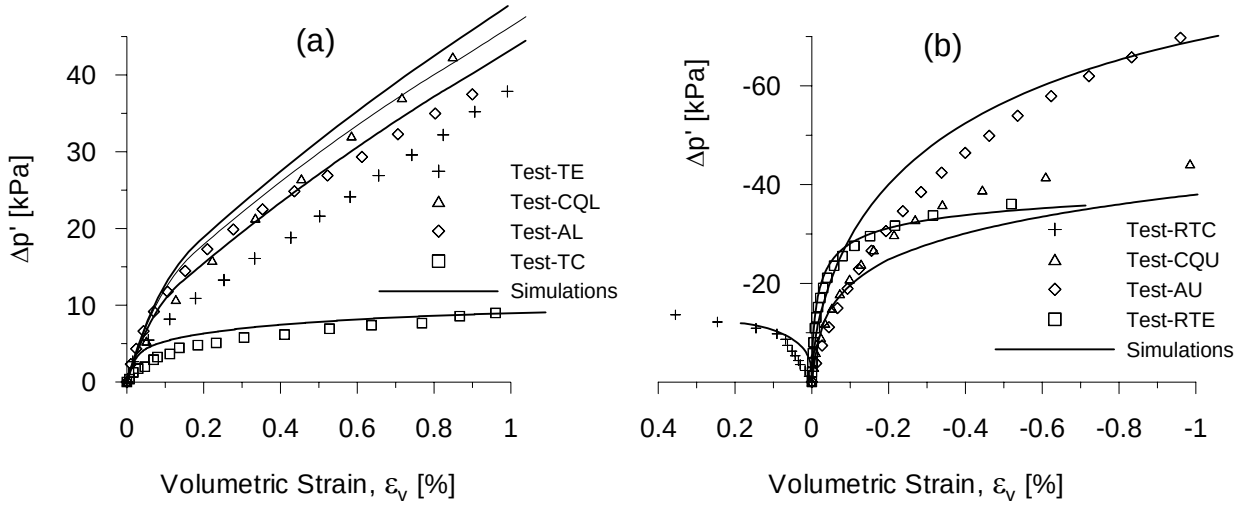

Fig. 5–4. Simulated and observed volumetric responses in (a) loading tests, and (b) unloading tests.

Note that CQL and AL probes are within the compression zone, wherein the stress paths do not lead to the failure surface. As mentioned in Section 3.3.1.2, J_v is assumed to be one order of magnitude larger than K in the compression zone. According to Eq. (3.1), therefore, the volume change due to Δq is much less than that due to $\Delta p'$ in the compression zone, and thus can be

neglected. Hence, the slopes of both CQL and AL curves in Fig. 5–4(a) approximately represent magnitudes of the tangent bulk modulus K. As shown, these curves are quite similar to each other, suggesting that K follows an identical evolution rule in the compression zone, as has been implemented in the proposed model.

Again, in those stress paths wherein both $\Delta p'$ and Δq are nonzero, the directionality relations of both K and J_v use the assumed linear interpolation in Fig. 3–17. The success in reproducing the volumetric responses in these paths, i.e., TC, RTC, AU and RTE in Fig. 5–4, serves as a justification of these assumptions.

5.1.4 COUPLING RESPONSES

Fig. 5–5 shows the simulated and observed coupling responses in constant-q tests wherein $\Delta q = 0$, and constant-p' tests wherein $\Delta p' = 0$. Because one stress component is zero, the effect of the other component is explicitly exhibited herein. As shown, the coupling responses are well reproduced by the model.

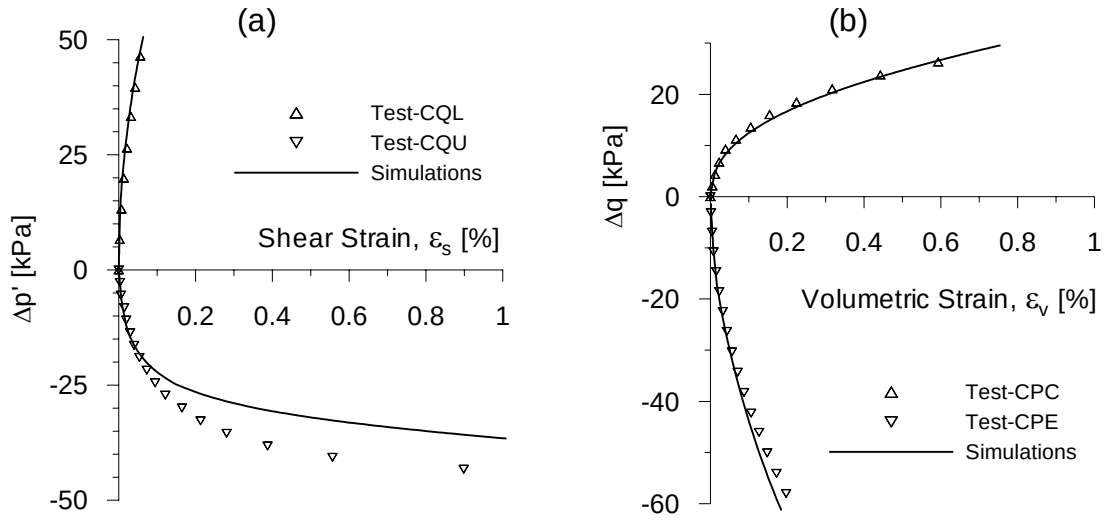

Fig. 5–5. Simulated and observed coupling responses in (a) constant-q tests, and (b) constant-p′ tests.

In the constant-q tests (Fig. 5–5(a)), only $\Delta p'$ contributes to $\Delta\eta$. Therefore, shear strain ε_s is solely caused by $\Delta p'$ and their relation is described by the coupling modulus J_s. In the CQL test, the stiff response justifies the assumption made in the model that the effect of J_s is negligible in the compression zone. In the CQU test, the large shear strain developed in the CQU path serves as another important evidence that shear strains in soils are essentially caused by change in η instead of q alone, as q is constant throughout the test.

In the constant-p′ tests (Fig. 5–5(b)), the relation between Δq and ε_v is described by the coupling modulus J_v. As shown, contractive volume changes are generated, whether q is increased (CPC test) or decreased (CPE test). This is consistent with the fact that the compressible Chicago clay is lightly overconsolidated, because lightly overconsolidated soils tend to be contractive during shearing. In an undrained cyclic shear test, therefore, soil skeleton

is prone to contraction. As a result, positive pore water pressure will be generated in order to satisfy the constant volume condition.

It is worth mentioning that the computed results shown in Fig. 5–5(b) indicate that this model might be applicable to liquefaction analysis. Since the contraction occurs in either way of cyclic shearing, as exhibited in the figure, the positive pore water pressure is expected to accumulate and the effective pressure p′ will keep decreasing as the cyclic loading continues, until the soil is failed, which is the mechanical nature of a liquefaction phenomenon. Hence, the coupling modulus J_v provides a reasonable mechanism to simulate the pore water pressure buildup during a cyclic loading. Furthermore, the material degradation during this process can be taken into account by the mechanism in the model for stress reversals, which updates stiffness parameters based on p' at each stress reversal. Since p' keeps decreasing as explained previously, computed material stiffness will degrade cycle by cycle. Though cyclic or seismic loadings are not major concern of this dissertation, it is worth noting the model's potential in dynamic applications.

By adjusting the sign of J_v accordingly, the shear-induced dilation, which mainly occurs in highly overconsolidated clays or dense sands, can be simulated as well. Ideally, J_v should be a function of OCR for clays and a function of relative density for sands. However, this application has not been evaluated in the current model, due to a lack of test data.

5.2 DRAINED PLANE STRAIN CONDITIONS

A series of biaxial tests have been performed on both natural and reconstituted samples of Chicago clay (Erickson 2006). In a biaxial test, soil responses are probed under plane strain conditions, which are frequently encountered in field applications and numerical analysis. Since this proposed model is based on triaxial tests, performing plane strain computations allows one to check the generalization of the constitutive relation from axisymmetric to plane strain conditions.

As shown in Table 5-1, the samples used in the biaxial tests were obtained from the Ford Center site in Evanston, IL (Erickson 2006), not the same site as the triaxial test samples (Holman 2005). Furthermore, Holman's samples are from the Blodgett stratum while Erickson's samples are from the upper Deerfield stratum. Though both of them are categorized as compressible Chicago clays, it is known that the Deerfield clay is generally stiffer than the Blodgett clay (Chung and Finno 1992).

Only one drained biaxial test has been performed on a natural sample, in which load was added vertically until failure, i.e., a biaxial compression (BC) test. This test was modeled using a stress-controlled finite element procedure. In accordance with the test, the initial condition was specified as σ'_v = 133 kPa and σ'_v = 74.5 kPa. An increment of 60 kPa was then added vertically to the simulated element.

Fig. 5–6 shows the shear and volumetric responses in the BC test. The subscript "1" denotes the in-plane vertical axis, respectively. In either case, the computation is able to capture the

overall trend of the stress-strain relation. It is noted that the observed response is initially stiffer than the computed response, which is consistent with the fact that the Deerfield clay is stiffer than the Blodgett clay, since the parameters used in the computation are based on Blodgett samples.

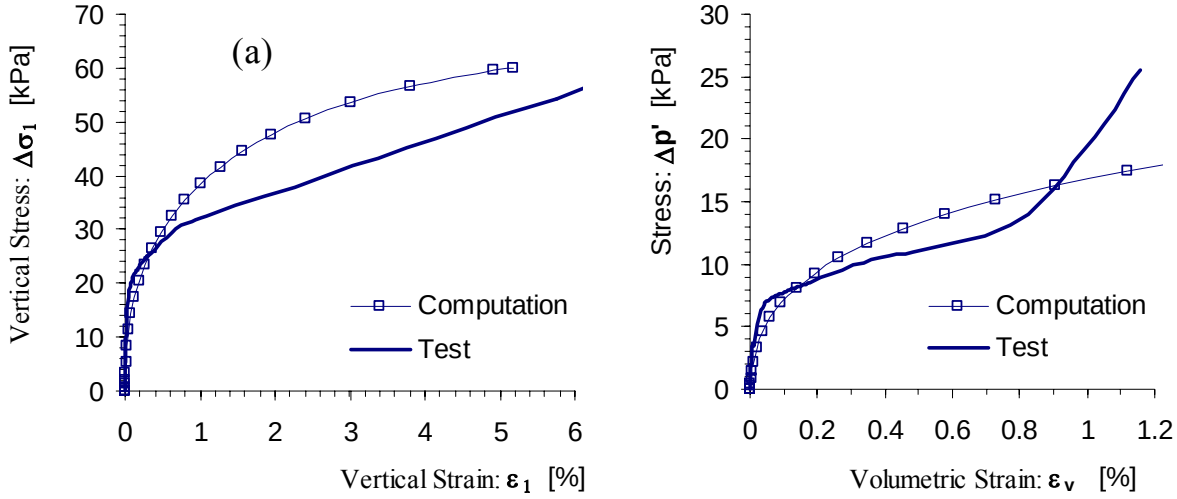

Fig. 5–6. Drained biaxial compression test: (a) shear response, and (b) volumetric response

Fig. 5–7 shows the computed and observed out-of-plane stress development with the vertical load, which exhibits a reasonable match. Though the out-of-plane stress is usually ignored in plane strain computation, it is a useful tool to verify a constitutive relation. Because the core relations of the directional stiffness model mostly rely on experimental observations under axisymmetric conditions, the evaluation of the out-of-plane stress, which is characteristic of plane strain condition, is especially important. The match shown in Fig. 5–7 indicates a reasonable extension of the proposed constitutive relation from axisymmetric conditions to plane strain conditions.

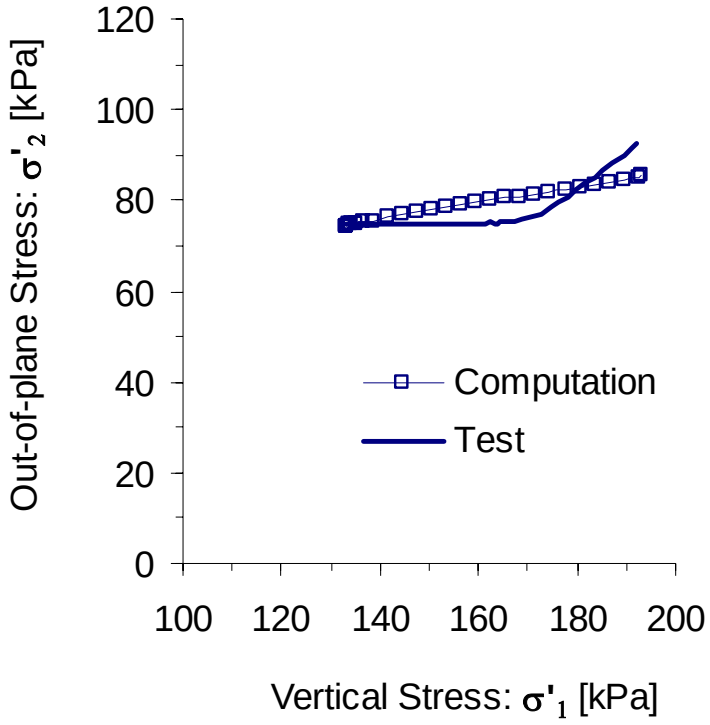

Fig. 5–7. Drained biaxial compression test: Out-of-plane stress

5.3 UNLOAD-RELOAD UNDER ONE-DIMENSIONAL CONDITIONS

In an oedometer test, a soil specimen may be subjected to one or more unload-reload cycles, so that the compression and recompression indices can be obtained for the material. In the unload-reload cycle, the stress path direction is reversed, a situation referred to as *stress reversal* (cf. Section 3.3.3). This section presents computed material responses in an unload-reload cycle in an oedometer test. The computation was made using stress-controlled finite element procedure, in which the material was treated as drained and pore water dissipation was not

actually simulated, since time-dependent hydrodynamic responses were not of concern in this case. The computation was made by starting from the same initial stress state as that used in the stress probe tests, increasing the vertical load by 300 kPa, then decreasing it by 400 kPa, and finally increasing it again by 800 kPa. During the entire computation, the lateral deformation of the element is restricted to enforce zero lateral strain conditions.

Fig. 5–8 shows the computed stress-strain relation in comparison to the curve obtained from an oedometer test conducted on a Lurie site sample. In traditional soil mechanics, the average slope of the unload-reload portions of the curve is represented by the recompression index C_r. However, there is no such parameter in the model to control this slope. As shown, not only the slope but also the hysteretic behaviour are well represented by the model. In the MIT series of models (e.g. Whittle 1990; Pestana and Whittle 1999), hysteretic behaviours are separately considered using a perfect hysteretic relation (Hueckel and Nova 1979), which requires several special parameters. In contrast, Fig. 5–8 has shown that the hysteretic response is naturally taken into account in the directional stiffness model, mainly due to the non-linear relations for stiffness evolution (cf. Section 3.3.1) and the criterion for stress reversals (cf. Section 3.3.3).

As shown in Fig. 5–8, the initial portion of the computed curve is inclined in an angle approximately the same as that of the unload-reload cycle. As discussed in Section 3.3.1.3, this portion corresponds to the small strain behavior and is defined by Eq. (3.22). It is worth pointing out that the proposed model is intended for in situ soils that are subject to ageing. It treats the small strain behavior as an inherent property of in situ soils, which explains the reason why the small strain behavior manifests itself in Fig. 5–8. In conventional elastoplastic theory for soils,

the same behavior is represented by a linear segment in the semi-logarithmic scale (e.g. Bjerrum 1967), similar to the approach of using C_r to describe the unload-reload portions. In comparison, the computed response exhibits a nonlinear curve. The linear relation used in the conventional approach is an approximation, in comparison with the small strain relation defined in this model.

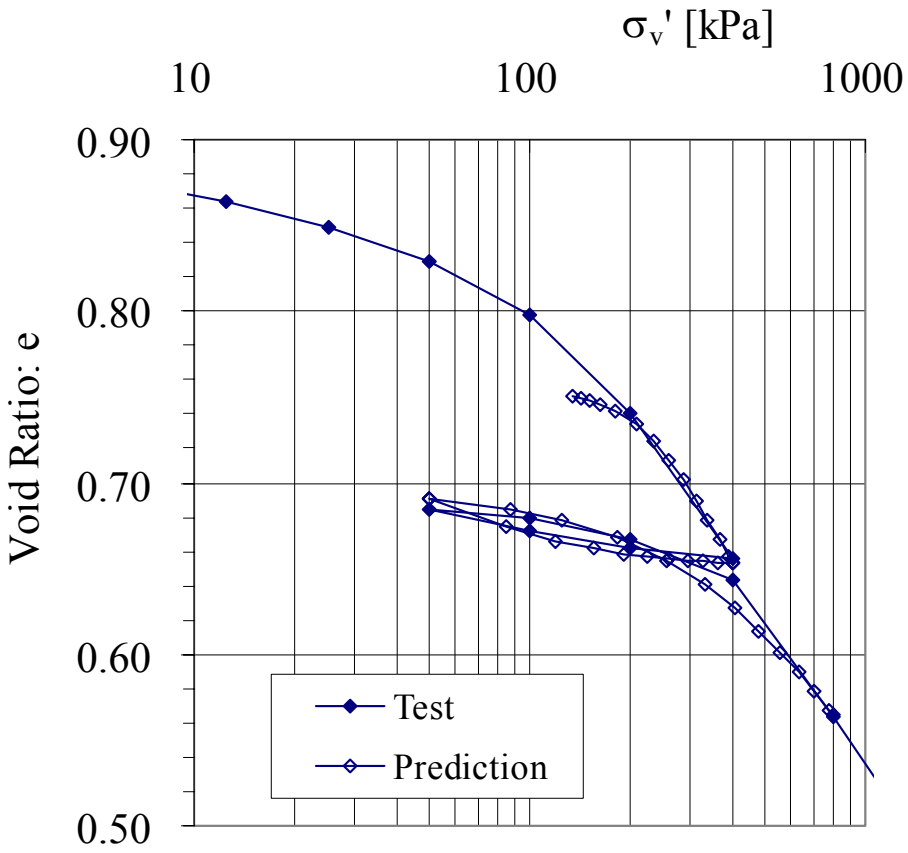

Fig. 5–8. Computed and observed responses in oedometer test

5.4 UNDRAINED CONDITIONS

This section explains how the directional stiffness model is applied to undrained computations, discusses characteristics of effective stress paths expected in undrained tests, and presents computed and observed responses for undrained triaxial and biaxial tests.

5.4.1 UNDRAINED COMPUTATION

With an effective stress constitutive relationship available, undrained computations can be performed by invoking the principle of effective stress.

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' + \boldsymbol{u} \tag{5.1}$$

where $\boldsymbol{u} = [u, u, u, 0, 0, 0]^T$, with u being the pore fluid pressure. The effective stress $\boldsymbol{\sigma}'$, can be computed by the effective model in a rate form.

$$\dot{\boldsymbol{\sigma}}' = \boldsymbol{E}' \dot{\boldsymbol{\varepsilon}} \tag{5.2}$$

where $\mathbf{E}'$ represents the stiffness of soil skeleton. Under undrained conditions, soil skeleton and pore fluid deform together and the strains in each phase can be regarded as the same, if the pore fluid is imaged as a uniform material occupying the total volume of the soil. Thus, the change in the pore fluid pressure can be computed from the macroscopic average strain.

$$\dot{\boldsymbol{u}} = \boldsymbol{E}_u \dot{\boldsymbol{\varepsilon}} \tag{5.3}$$

where $\mathbf{E}_u$ is the stiffness matrix for the imaginary pore fluid. Since a fluid does not transmit shear, this matrix has the following form

$$\boldsymbol{E}_u = K_u \begin{bmatrix} \boldsymbol{1}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{5.4}$$

where $\mathbf{1}_3$ is a 3×3 matrix with all elements equal to 1, $\mathbf{0}_3$ is a 3×3 null matrix, and the constant K_u can be related to K_f, the bulk modulus of the pore fluid, and K_p, the bulk modulus of the soil particles (Naylor 1974). However, since both K_f and K_p are typically much larger than the bulk modulus of the soil skeleton – K_s, the exact value of K_u becomes unimportant, as long as it is a large number. Potts and Zdravkovic (1999) recommend setting K_u equal to nK_s, where n ranges from 100 to 1000.

Note that a total stress instead of an effective stress is eventually required by UMAT in ABAQUS for each integration point. With $\mathbf{E}'$ and $\mathbf{E}_u$ available, the total stress increment can be computed given the strain increment.

$$\dot{\boldsymbol{\sigma}} = (\boldsymbol{E}' + \boldsymbol{E}_u)\dot{\boldsymbol{\varepsilon}} \tag{5.5}$$

where the sum of $\mathbf{E}'$ and $\mathbf{E}_u$ represents the overall stiffness of the undrained soil. A nearly zero volume change condition is enforced by the presence of $\mathbf{E}_u$, with K_u much larger than K_s. For the directional stiffness model, $\mathbf{E}'$ is given by Eq. (3.13). In the model, an undrained condition is activated by making K_u equal to the bulk modulus of water ($\approx 2*10^6$ kPa), while a drained condition results when K_u is set to zero.

5.4.2 EFFECTIVE STRESS PATH (ESP)

5.4.2.1 UNIQUENESS OF ESP

Before presenting computed results for undrained tests, it is beneficial to review the general characteristics of effective stress paths (ESP) in undrained tests. Under either an axisymmetric or a plane strain condition, an undrained test is subject to two possible loading modes only, corresponding to two unique ESPs.

Fig. 5–9 shows any applied stress in an undrained triaxial test can be decomposed into two components – a hydrostatic component and a deviatoric component. The hydrostatic component does not contribute to any change in effective stresses of the tested sample, because K_u, the bulk modulus of water, is typically much larger than K_s, the bulk modulus of the soil skeleton, as explained in the previous section. Thus, nearly all increase in the hydrostatic stress will be sustained by the bulk of pore water. In other words, one unit of change in the hydrostatic stress results in approximately one unit change in the excess pore water pressure, while the effective stress exerted on the soil skeleton virtually does not change. In contrast, variation of the deviatoric component (see Fig. 5–9) leads to change in the effective stress, because water cannot sustain shearing. Apparently, there are only two possibilities of shearing mode, i.e., $\Delta\sigma_v$-$\Delta\sigma_h > 0$, a *compression mode*, or $\Delta\sigma_v$-$\Delta\sigma_h < 0$, an *extension mode*. Accordingly, these two distinct modes will result in two unique ESPs, no matter how the hydrostatic load component varies.

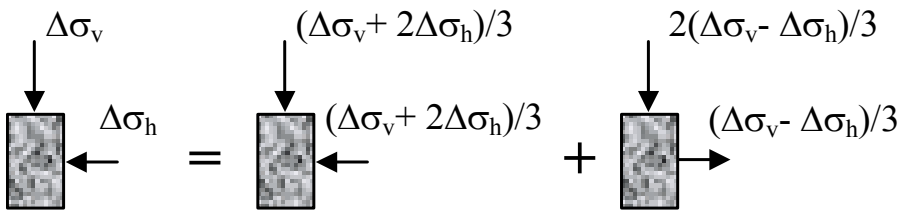

Fig. 5–9. Decomposition of undrained triaxial test into hydrostatic and deviatoric components

For an undrained biaxial test, this approach using stress decomposition to explain unique ESP is not straightforward because the out-of-plane boundary condition is displacement-controlled. However, no matter what in-plane stress is applied to the specimen, the vertical strain ε_1 and horizontal strain ε_3 must maintain the relation $\varepsilon_3 = - \varepsilon_1$ to satisfy the undrained condition. Therefore, all undrained biaxial tests could only have two deformation modes, i.e., $\varepsilon_1 > 0$, a compression mode or $\varepsilon_1 < 0$, an extension mode. Obviously, these two modes correspond to two unique ESPs.

Because of the uniqueness of the ESP, one needs only two different tests to investigate undrained responses of a soil under either axisymmetric or plane-strain conditions, corresponding to the two distinct modes, respectively.

5.4.2.2 ORIENTATION OF ESP

In the foregoing section, it is made clear that ESP is unique in each of the two distinct modes. The next important issue is the orientation in stress space of the unique ESP.

It is well known that the orientation of ESP for an isotropic elastic material is vertically up in p'-q plot for a compression mode ($\Delta q > 0$) and vertically down for an extension mode ($\Delta q < 0$), because there is no shear-volumetric coupling for this material, i.e., both J_v and J_s are infinitely large. Therefore, an isotropic elastic relation is a special case of the relation proposed in this dissertation. In general, the orientation of observed ESP in most soil tests is not vertical. Using β (cf. Eq. (3.17)) to define the orientation of ESP, in a compression mode, β is generally larger than $\pi/2$ for normally consolidated or slightly overconsolidated clays or loose sands, and is generally smaller than $\pi/2$ for highly overconsolidated clays or dense sands.

In the directional stiffness model, the orientation of ESP is governed by the ratio of J_v and K. According to Eq. (3.1), the volumetric strain is calculated as follows,

$$\delta\varepsilon_v = \delta p' / K + \delta q / J_v \tag{5.6}$$

In an undrained computation, the zero volume change condition is enforced by Eq. (5.5). Therefore, $\delta\varepsilon_v = 0$, and

$$\delta q / \delta p' = -J_v / K \tag{5.7}$$

The left hand side basically represents the orientation of ESP (cf. Eq. **(3.17)**). Hence, Eq. (5.7) shows how the ratio of J_v/K affects the orientation of ESP under undrained conditions. Given a constitutive relation with constant J_v and K, one can calculate the orientation directly using Eq. (5.7). With the proposed directional stiffness model, the tangent moduli, including J_v and K, vary with β and LSP. It is not straightforward to determine a priori the orientation of ESP. Consequently, one needs an iterative algorithm (cf. Section 4.2.2) to find the right orientation, in which the path-dependent J_v and K yield zero volume change.

A qualitative determination of the orientation of ESP, however, can be achieved simply by knowing the sign of J_v. Here, attention is paid to which side the ESP is oriented from the vertical direction, i.e., $\beta = \pi/2$ for the compression mode, or $\beta = 3\pi/2$ for the extension mode. These two values correspond to isotropic elastic responses. For conciseness, discussion herein is made for the compression mode. Because K is always positive, the sign of $\delta q/\delta p'$ solely depends on the sign of J_v, according to Eq. (5.7). For a positive J_v, i.e., a shear-contractive material, $\delta q/\delta p' < 0$, $\beta > \pi/2$, and thus the ESP leans to the left of a vertical line in the p'-q plot. According to Eq. (5.7), the smaller the ratio of J_v/K, the more leftward the ESP will track. Apparently, the ESP of a shear-dilative material leans rightward in the compression mode. The deviation from the vertical direction represents the degree of anisotropy (Wood 1990). Note that whether the ESP leans to the left or right hand side is governed by material properties of the soil skeleton, and is independent of the orientation of total stress path (TSP).

Hence, the orientation of ESP can be intrinsically derived from the constitutive relation of the material, and thus is purely material-dependent. In the proposed model, it is controlled by the ratio of J_v/K. In contrast, the excess pore pressure u_e is partially material-dependent, because it results from the difference between ESP (material-dependent) and TSP (material-independent). For this reason, it is undesirable to use any parameter directly describing u_e as a material parameter in a constitutive relation. Using the decomposition illustrated in Fig. 5–9, the change in u_e due to the hydrostatic component of a load is basically material-independent, while the change in u_e due to the deviatoric component is material-dependent. The total change in u_e is the sum of them.

From a physical point of view, an undrained test is an interaction between the pore water and the soil skeleton, in which u_e develops and ESP is mobilized in a way that the condition of no volume change of the soil skeleton is observed. An undrained test can be regarded as a special drained test, in the sense that a drained test conducted along the observed ESP from an undrained test on the same material should yield zero volume change. Therefore, the drained and undrained material responses are intrinsically related to one another. Using this idea, drained and undrained tests can be designed to verify results obtained in either case.

5.4.3 UNDRAINED AXISYMMETRIC CONDITIONS

K_0-consolidated, undrained triaxial compression (CK_0U-TXC) and reduced triaxial extension (CK_0U-RTXE) tests were performed on tube samples of compressible Chicago glacial clay from the Block 37 site in Chicago (Cho 2007). These samples were not obtained from the same site as those used for the stress probe tests (Holman 2005). However, both of them were retrieved from the same geologic stratum and thus were expected to have similar responses despite being extracted from different sites.

The computations were made in four different total stress paths – U-TXC, U-RTXC, U-TXE and U-RTXE, all of which started from same initial stresses, corresponding to the in situ state of the samples used in the undrained tests. Fig. 5–10 shows the computed ESPs in comparison with the observed ones. As expected, the computations for these paths have converged to a unique

ESP for the compression mode ($\Delta q > 0$), and another for the extension mode ($\Delta q < 0$), which is consistent with the conclusion drawn in Section 5.4.2.1.

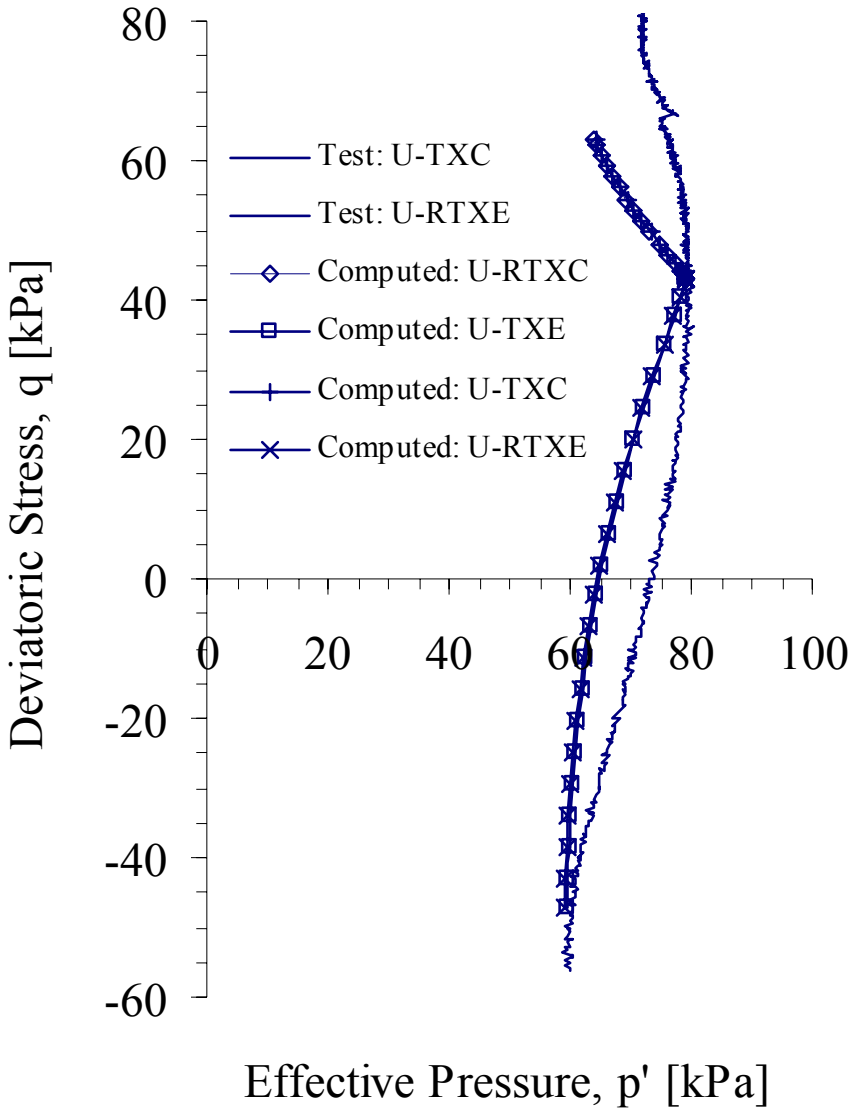

Fig. 5–10. Computed and observed effective stress paths for undrained triaxial tests

Furthermore, in comparison to the drained stress probe tests, it is found that the computed ESP in the compression mode is close to the stress path of the RTC test (cf. Fig. 5–2). And the computed ESP in extension mode is close to the stress path of the RTE test (cf. Fig. 5–2). In fact, it is found that the volume changes in both of these two tests are quite small (cf. Fig. 5–4). In other words, the undrained computations with material parameters obtained from the drained tests have yielded results consistent with the responses observed in the drained tests.

However, the difference is perceivable between the orientations of the observed and computed ESPs, especially in the compression mode, as shown in Fig. 5–10. According to the discussion made in Section 5.4.2.2, the deviation in the compression mode is due to the difference in J_v/K ratio. Specifically, the J_v/K ratio used in the computation, which was obtained from the drained tests , is smaller than the J_v/K ratio observed in the undrained test. That means the drained tests (Holman 2005) are not quite consistent with the undrained test, as one would expect when samples are taken from sites that are about 1 mile apart.

Fig. 5–11 shows that using an increased Jv_{ref} value, the proposed model yields an ESP close to that observed in the undrained test. The computation of U-TXC_1 is the same as the computation of U-TXC shown in Fig. 5–10, using the parameters listed in Table 3-1. Based on those parameters, $Jv_{max}/Jv_{min} = 5$, because $r_x = 5$ (cf. Eq. (3.31)). Note that $Jv_{ref} = Jv_{min}$. The computation of U-TXC_2 using a Jv_{ref} 5 times of Jv_{ref} used in U-TXC_2 basically makes Jv_{min} equal to Jv_{max}. This additional computation serves as an example how the ESP varies with J_v/K ratio with the proposed model. Fig. 5–11 basically shows that the proposed model is capable of accommodating either observed response.

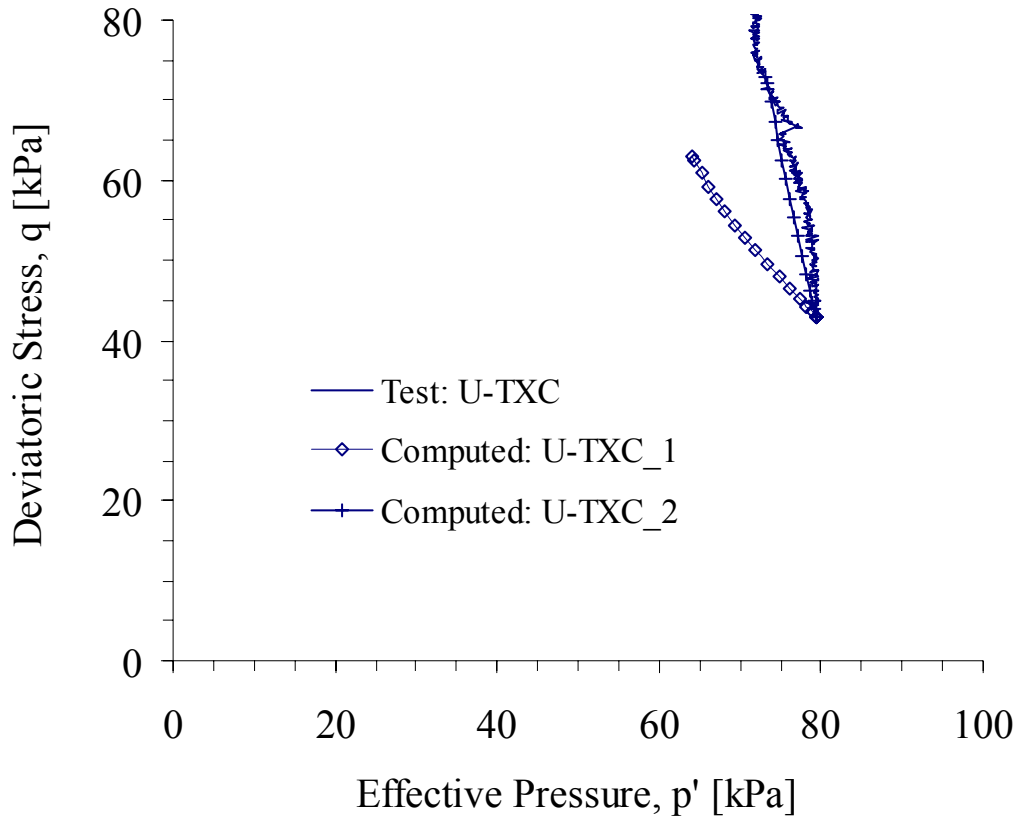

Fig. 5–11. Variation of the orientation of ESP: Jv_{ref} = 116 in U-TXC_1 and Jv_{ref} = 580 in U-TXC_2

Fig. 5–12 shows the computed and measured shear responses in the two undrained triaxial tests. As shown, the computation well matches the response measured in the U-RTXE test. The differences in the responses in the U-TXC test is mainly due to the mismatch in ESP (cf. Fig. 5–10), which is caused by the inconsistency in volumetric response, as discussed before. One can imagine that the computed and observed responses will match by proportionally stretching the computed curve along vertical axis. Therefore, it is fair to say that the overall trend of the material response is captured by the computation. Though there exists inconsistency in the

observed volumetric response, Fig. 5–10 shows that the observed responses generally are consistent under undrained and drained conditions, since the computed shear response is largely derived from the observed drained material response.

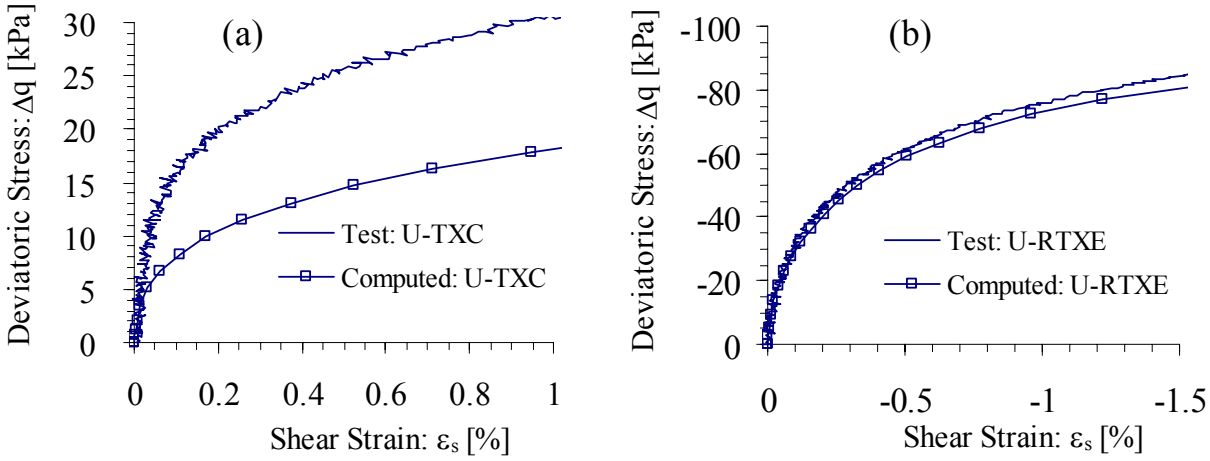

Fig. 5–12. Computed and observed shear responses in undrained triaxial tests

5.4.4 UNDRAINED PLANE STRAIN CONDITIONS

Soil responses under undrained plane strain conditions are critical for analyzing short-term construction activities in many geotechnical applications, which is the case for the computation presented in the next section for a real excavation in downtown Chicago. In order to manifest model responses under undrained plane strain conditions, four different stress paths were computed, including U-BC, U-RBC, U-BE and U-RBE, wherein one of the two in-plane stresses is monotonically increased or decreased while the other remains constant.

Fig. 5–13 shows that the computed effective stress paths have converged in the compression and extension modes, respectively, which is expected according to Section 5.4.2. In comparison with Fig. 5–10, it is found that the computed ESPs lead to similar directions under axisymmetric and plane strain conditions. Furthermore, the computed ESPs are nearly linear, as also observed for the axisymmetric cases. Under undrained conditions, strain paths are linear in either triaxial or biaxial tests, due to the condition of zero volume change. The computed results imply that this model tends to yield linear ESPs given linear strain paths.

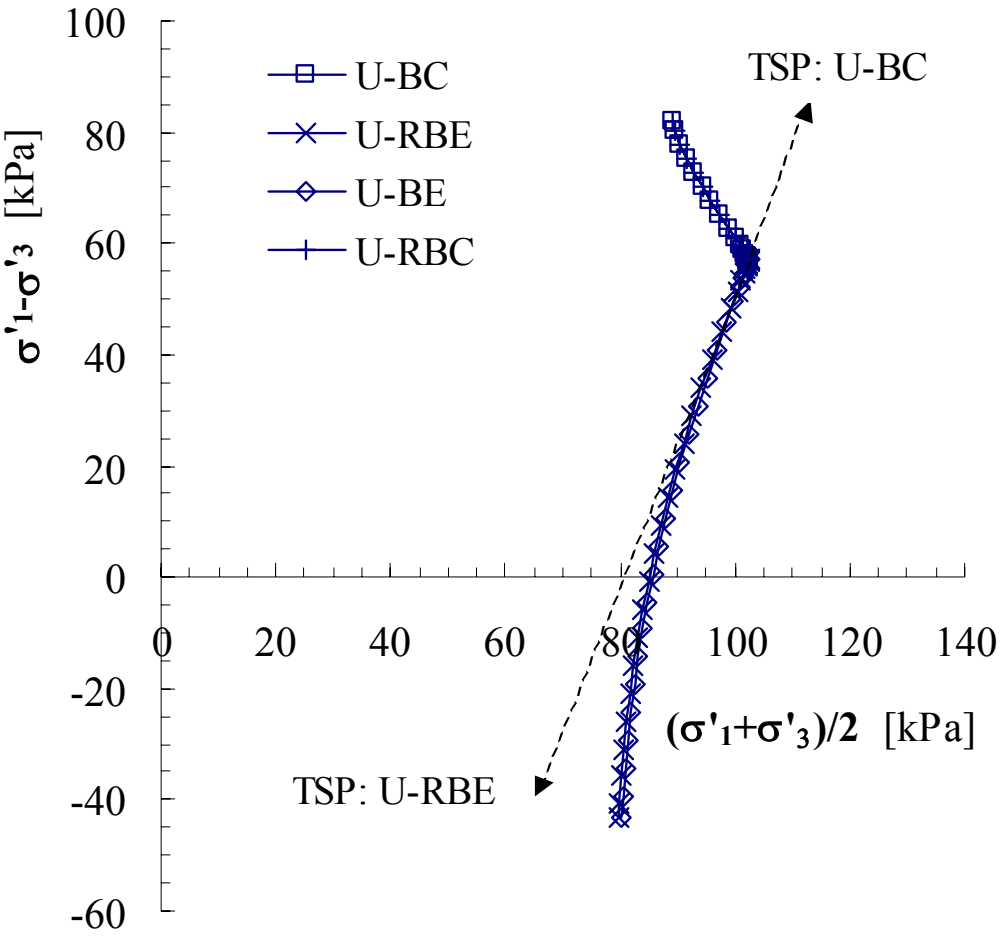

Fig. 5–13. Computed effective stress paths in undrained biaxial tests

Fig. 5–14 shows the computed stress-strain responses in the U-BC and U-RBE paths. Since the other two computations basically result in the same ESPs, they are skipped in Fig. 5–14 for conciseness. As shown, the stress-strain response in the U-RBE path is much stiffer than that in the U-BC path, which is similar to the model response under an undrained axisymmetric condition.

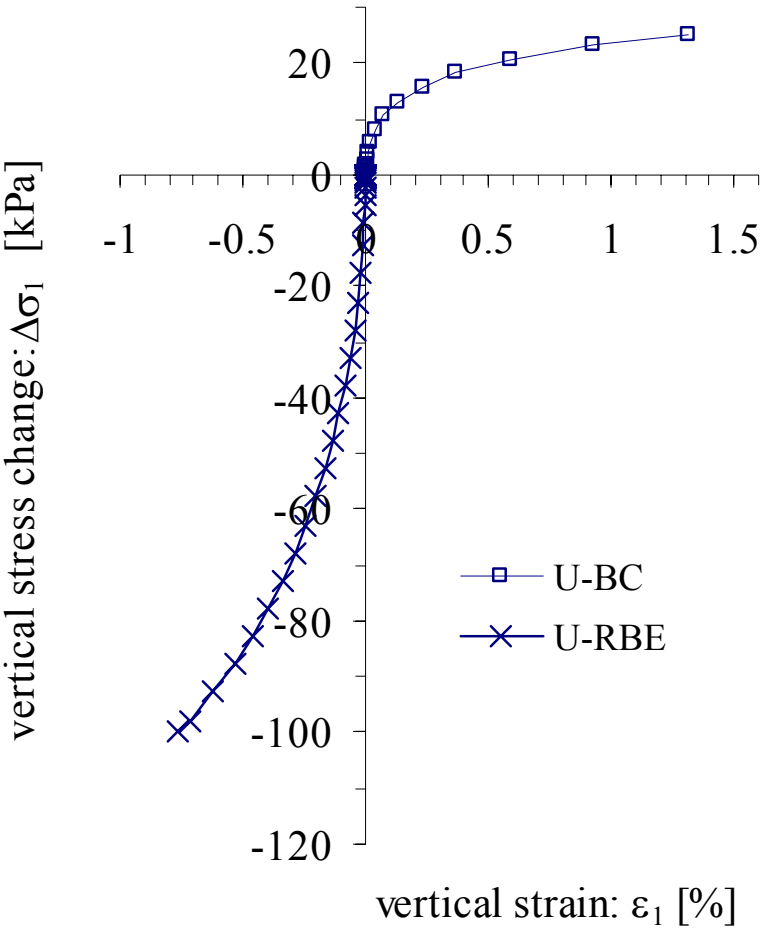

Fig. 5–14. Computed stress-strain relations under undrained plane strain conditions

Fig. 5–15 shows the computed excess pore water pressure developed with vertical stress change. As indicated in Fig. 5–13, the TSP of U-RBE is orientated to the left of the unique ESP in the extension mode, while the TSP of U-BC is to the right of the unique ESP in the compression mode, resulting in the development of negative excess pore pressures in the U-RBE test and positive excess pore pressures in the U-BC test.

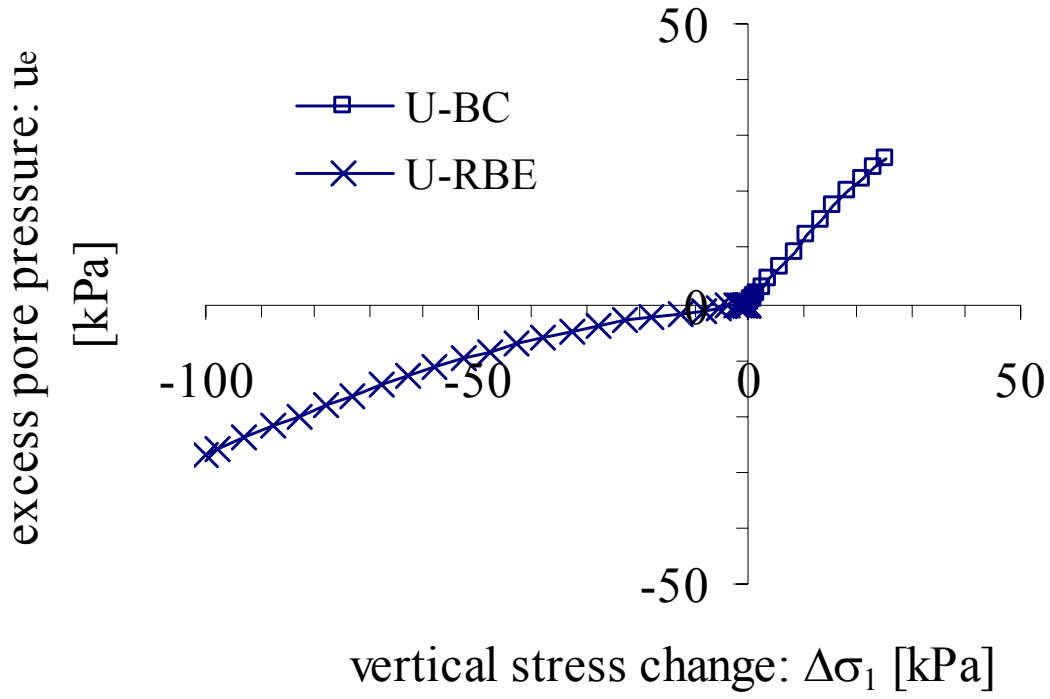

Fig. 5–15. Computed excess pore water pressure under undrained plane strain conditions

When numerical analysis is to be performed under plane strain conditions, it is desirable to test the pertinent soils under the same condition. Experimentally, however, biaxial tests are much more complicated and much less common than triaxial tests. The computed results presented herein suggest that the two unique ESPs tend to have similar directions under

undrained axisymmetric and plane strain conditions. For the compressible Chicago clays tested by Holman and Cho, the two unique ESPs are around RBC and RBE paths under plane strain conditions, and the RTXC and RTXE paths under axisymmetric conditions, respectively. Accordingly, the mobilized soil responses tend to be similar. Hence, it might be justified to use undrained triaxial tests to estimate soil responses under undrained plane strain conditions. For the same purpose, drained tests can be performed in the vicinity of the two unique ESPs.

5.5 PREDICTIONS FOR LURIE CENTER EXCAVATION

5.5.1 LURIE PROJECT DESCRIPTION

The Robert H. Lurie Medical Research Building included a 12.8 m deep cut for two basement levels. Detailed descriptions and ground responses of the excavation are reported by Finno and Roboski (2005). A plan view of the approximately 80 m × 68 m area is shown in Fig. 5–16. To monitor the ground response to excavation activities, 150 surface survey points, 18 embedded settlement points and 30 utility points were installed on three surrounding streets prior to wall installation. Measurements of both lateral and vertical ground surface movements were obtained. In addition to the optical survey data, seven inclinometers were installed at distances from 1 to 2.4 m from the sheet-pile wall.

Fig. 5–17 shows the support system in relation to the stratigraphy. The excavation is supported by a PZ-27 sheet pile wall on all sides. Three levels of tieback anchors provided lateral support for the south wall section analyzed herein. Both the first and second level ground anchors are founded in the beach sand.

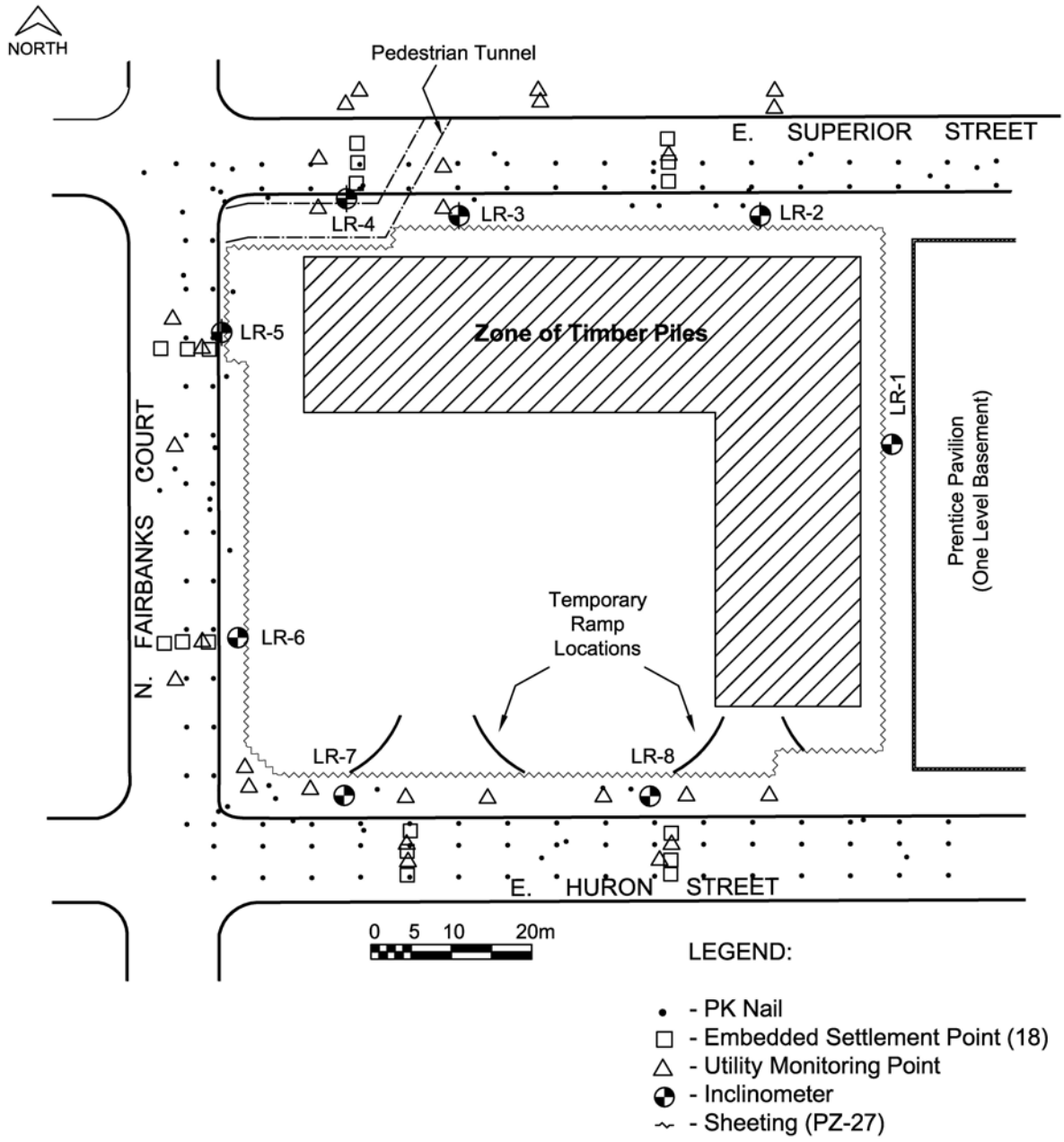

Fig. 5–16. Plan view of Lurie Center excavation

Prior to installation of the sheeting, the excavator "pot-holed" the site to remove large obstructions such as pile caps and building rubble. PZ-27 sheets of length 18.3 meters were

installed by a vibratory hammer. Ground water within the site was removed by dewatering wells. Surface water and leakage through the sheeting into the excavation was controlled by sump pits and pumps. Excavation of the site and tieback installation took place simultaneously within the site. However, four distinct excavation stages were defined: excavation to elevations +1.5 m CCD, -2.5 m CCD, -5.8 m CCD, and –8.5 m CCD, corresponding to levels immediately below tieback elevations and the final excavated grade. Excavation was limited to a distance of 0.6 to 1.2 meters below the tieback installation elevation, depending on the angle of the tieback installation.

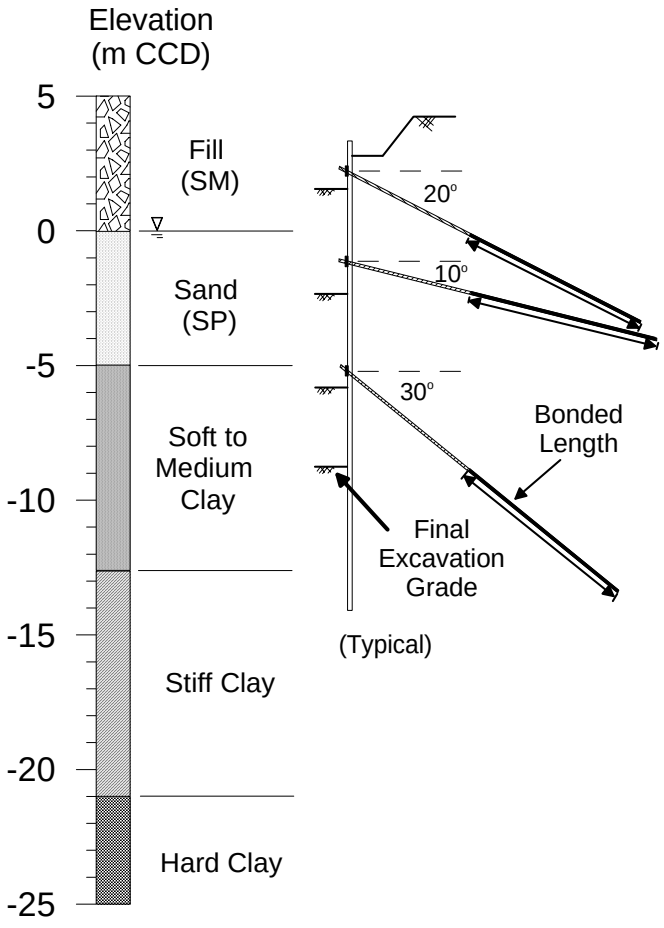

Fig. 5–17. Support system of Lurie Center excavation

5.5.2 FINITE ELEMENT DESCRIPTION

5.5.2.1 F.E. MESH

The finite element simulation of the Lurie Center excavation was made using ABAQUS. The finite element mesh used for the prediction is shown in Fig. 5–18, which corresponds to the final stage of the excavation and thus does not include those elements representing the excavated soils. The simulation was made assuming a plane strain condition with the excavation system being symmetric about the centerline. The entire mesh was fixed at the bottom and allowed to move vertically and freely at both sides. To eliminate influences of the boundary condition on the modeling, the mesh behind the support wall was extended to a distance five times the excavation depth.

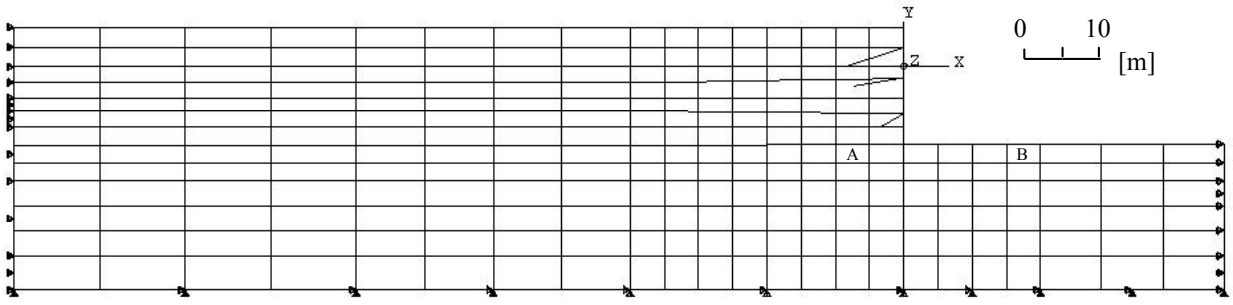

Fig. 5–18. Finite element mesh for Lurie Center excavation

5.5.2.2 ELEMENT TYPES

In the finite element model, the soils were simulated using 8-node biquadratic elements with reduced integration (CPE8R), the sheet pile wall was represented by the 3-node quadratic beam element (B22), and the interfaces between the wall and soils were simulated by the small sliding contact pair provided in ABAQUS. The tiebacks were simulated by 2-D truss elements (T2D2) with axial stiffness determined from performance tests. For each truss, one end was connected to the beam element using connection type JOIN and the other was fixed in space. This approximation is much closer to reality than if the anchor was modeled explicitly, because the proximity of the first two levels of bonded lengths in a 2-D representation (cf. Fig. 5–17) results in an unrealistically flexible support in an explicit simulation (Finno and Tu 2006).

5.5.2.3 MATERIAL MODELS

In terms of material modeling, the fill, sand, stiff clay and hard clay (cf. Fig. 5–17) were simulated by the Mohr-Coulomb (M-C) model given in ABAQUS, while the soft-medium clay was simulated by the directional stiffness model implemented in UMAT. The M-C parameters for each soil stratum are shown in Table 5-2, which were obtained from results of site investigations made in conjunction with foundation design studies for the Lurie Center structure, and past research experience with excavations in the Chicago area (e.g. Finno et al. 1991; Calvello and Finno 2004). Note that the fill and sand are drained, while the stiff and hard clays are undrained, which is approximated by specifying $\nu = 0.49$ for these layers. The parameters used for the directional stiffness model are completely based on the drained stress probe tests on

the block samples obtained from the Lurie site. Their values have been given in Section 3.3.3. Material parameters for the beam, trusses and interfaces can be found in the ABAQUS input file given in Appendix D. The same information is also given by (Rechea-Bernal 2006).

Table 5-2. Parameters for soils using M-C model

Stratum	E (MPa)	υ	ϕ°	c (kPa)	ψ°
Fill	51	0.2	30	0	2
Sand	29	0.39	35	0	5
Stiff clay	171	0.49	0	105	0
Hard clay	677	0.49	0	383	0

5.5.2.4 COMPUTATION STEPS

The finite element simulation represented the construction history near inclinometer LR-8 (cf. Fig. 5–16). The modeling consisted of 12 steps, including stress field initialization using a user-defined "GEOSTATIC" procedure, activating the beam elements and contact pairs, and cycles of element removals and truss element activations. Detailed definitions for these computation steps can be found in Appendix D.

5.5.2.5 STATIC PORE PRESSURE

Ground water is present in the Lurie site (cf. Fig. 5–17). It is not straightforward to model static pore pressure in ABAQUS, though ABAQUS allows analyses that couple effective stresses and excess pore pressures. There are two approaches to this problem, using effective stress and

total stress, respectively. The effective stress approach has been elaborated by Hashash (1992). In this approach, the soil skeleton serves as the object to be analyzed. Accordingly, effective density is used for the soils and the finite element code computes effective stresses. In the computation, pore water pressure is regarded as an external load. Specifically, the unbalanced water pressure on both sides of the wall is treated as lateral traction on the exposed wall. Similarly, the unbalanced water pressure at the bottom of the excavation is treated as uplift force on the exposed soil surface. The exposed part of the wall and the exposed soil surface vary as the excavation proceeds. When a simulation consists of many computation steps, this approach can be somewhat cumbersome, because the unbalanced forces must be defined by manually modifying the input file, instead of being graphically defined in CAE, the GUI preprocessor of ABAQUS.

In contrast, a total stress approach was adopted herein, wherein soil skeleton and pore water together serve as the object to be analyzed. Total density is used for the soil body and total stresses are calculated by the FEM code. Accordingly, the constitutive relation for a soil needs to represent total stress response of the material. How to obtain the total stress relation based on an effective stress model is elaborated in Section 5.4.1. In UMAT, a variable of excess pore water pressure should be included in STATEV, the array containing the "solution-dependent state variables," and another variable computes hydrostatic pressure according to COORDS, the array containing the coordinates of the integration point. More details can be found in the UMAT file given in Appendix C. Therefore, the user-defined soil model is able to update and track pore water pressure, and performs stress integration in terms of effective stresses. To make

this total stress approach work, the initial lateral stress must be specified in terms of total stress and match the expected in situ condition. In this computation, the stress field was initialized using a user-defined "GEOSTATIC" procedure, in which the balance between the internal and external nodal forces was checked by ABAQUS without actual computation.

5.5.3 RESULTS

Fig. 5–19 shows the computed ground movements around the excavation when the final grade is reached. The end of each displacement vector is located at the corresponding finite element node. The vector length corresponds to the displacement magnitude, with the scale shown at the top-right corner. The global pattern of the ground movements caused by the excavation is shown in this figure. Computed ground movements extend to a distance about 2.5 times of the final depth of the excavation, with the maximum settlement occurring at a distance about half of the excavation depth. In terms of lateral movements, there is very little cantilever movement, indicating an effective restriction provided by the upper level tiebacks. The maximum lateral movement occurs at a depth corresponding to the soft to medium clay layer, with the maximum lateral displacement about twice of the maximum settlement. In contrast, the bottom heave due to the excavation is relatively insignificant.

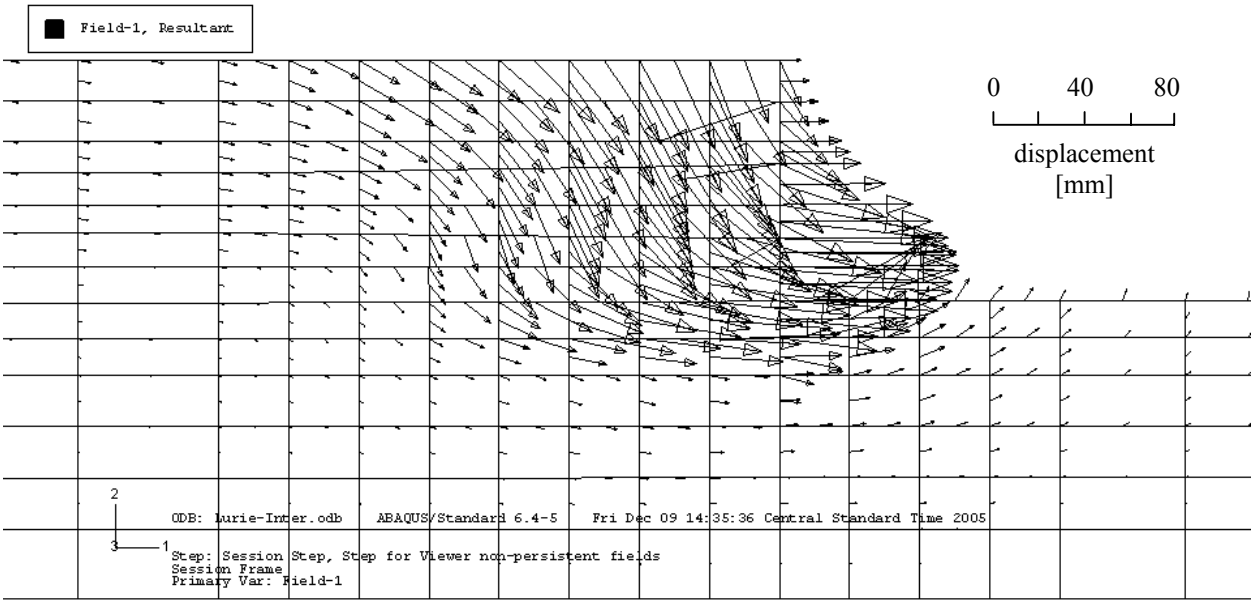

Fig. 5–19. Computed ground movements around the excavation when the final grade is reached

Fig. 5–20 shows the computed and observed lateral movements of the wall at three different excavation levels, including the grades –2.5m CCD, -5.8m CCD and –8.5m CCD, corresponding to the installations of the 2nd and 3rd levels of tiebacks and the final grade of the excavation, respectively. It is found that significant movements start to occur when the excavation proceeds into the soft to medium clay layer, the top of which corresponds to the depth of –5m CCD. This result agrees with the observation that the softest soil layer involved in an excavation governs the magnitude of the resulting ground movement (e.g. Clough et al. 1989). As shown in the figure, the observed deformations were somewhat overpredicted at each stage. However, the overall deformation patterns were well captured by the computation in a consistent manner.

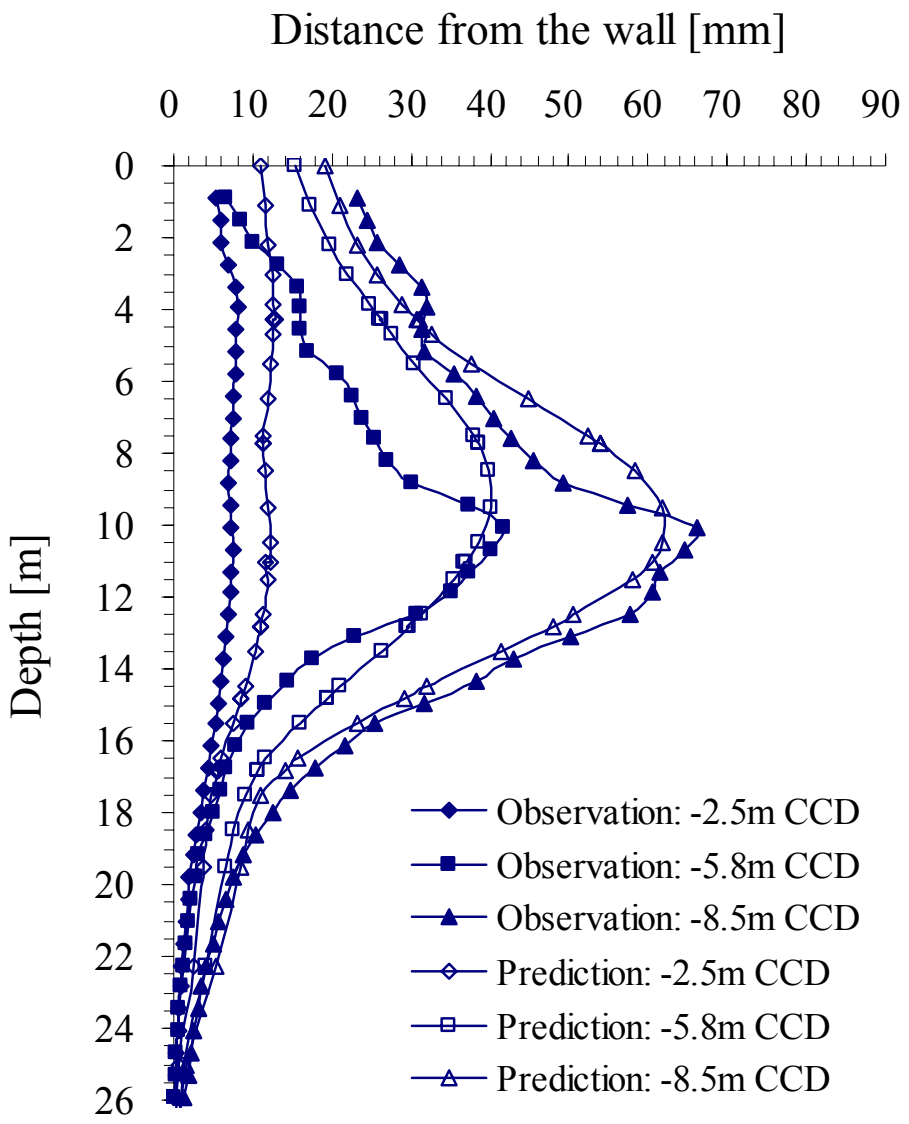

Fig. 5–20. Lateral movements of the sheet pile wall

Fig. 5–21 shows the observed and computed ground settlements behind the wall at the end of the excavation. Settlement observations were made for a distance up to 12m away from the excavation. As shown, the computation was successful in predicting the location of the

maximum settlement, which is around 6m away from the excavation, though the maximum was slightly underpredicted. Note that observed settlement tends to be somewhat larger than the actual settlement caused by the excavation, because both construction vibration and onsite traffic tend to increase the observed settlement. Under a fully undrained condition, the maximum settlement should be approximately the same as the maximum lateral displacement. This is not the case for the Lurie site, because of the 10m thick of sand layers present in the field (cf. Fig. 5–17). The settlement profile beyond the surveyed area is estimated using the empirical relation suggested by Hsieh and Ou (1998). As shown in Fig. 5–21, the predicted settlement profile is consistent with the empirical relation. Both of them indicate that the major impact range of the excavation is about 28 m.

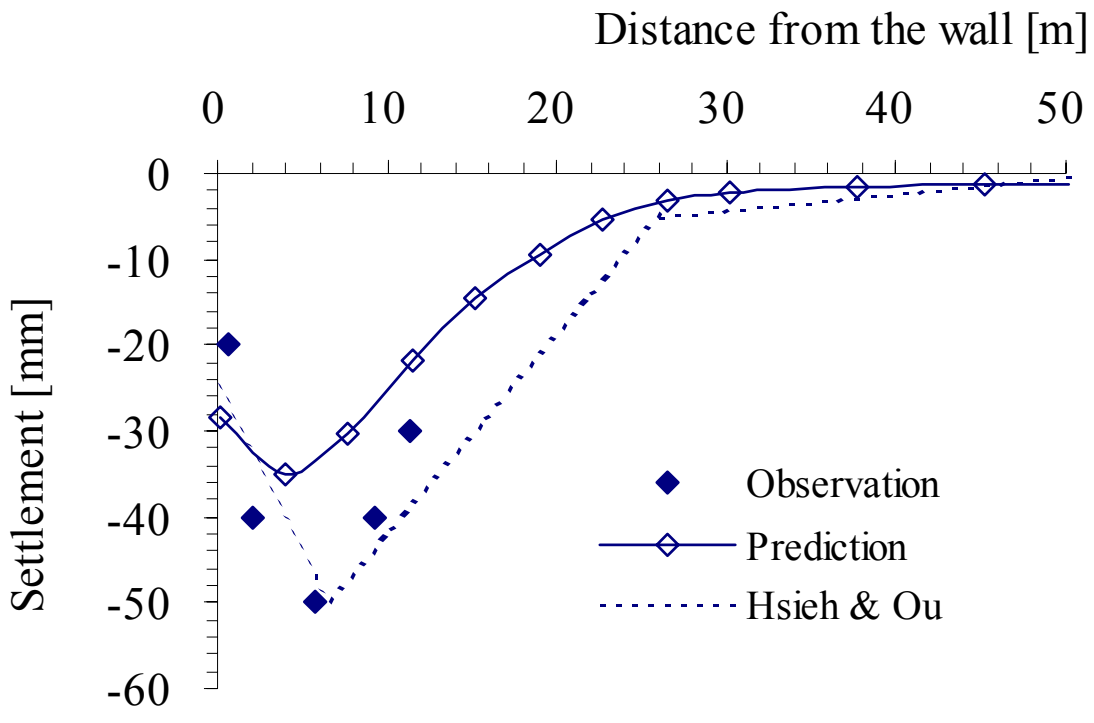

Fig. 5–21. Ground settlements behind the wall when the final grade is reached

Fig. 5–22 shows two effective stress paths at points A and B (Fig. 5–18), respectively. Point A is within the soft to medium clay layer (cf. Fig. 5–17) behind the wall, while point B is within the same layer at the bottom of the excavation and at the same depth as A. It is found that the direction of the ESP for most soils behind the wall is similar to the ESP of point A, while the direction of the ESP for most soils at the bottom of the excavation is similar to the ESP of point B. This is reasonable, because the soils behind wall experience a compression mode of undrained shearing, i.e., $\Delta(\sigma'_v - \sigma'_h) > 0$, while the soils at the bottom of excavation experience an extension mode, i.e., $\Delta(\sigma'_v - \sigma'_h) < 0$. According to the conclusion of Section 5.4.2, there should exist two unique ESPs. Furthermore, the two ESPs shown in Fig. 5–22 are found similar to those computed ESPs shown in Fig. 5–13, as expected.

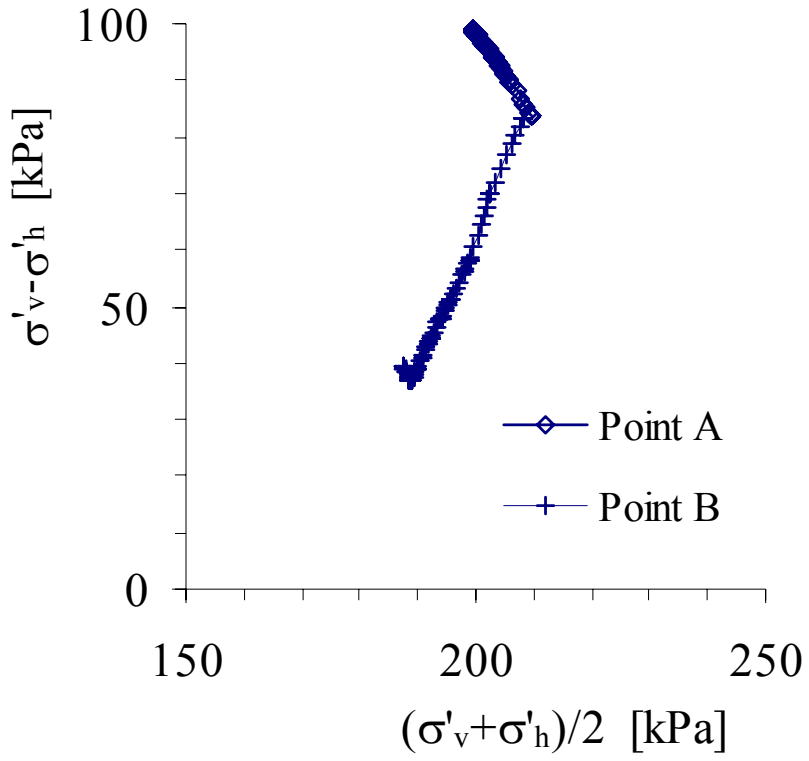

Fig. 5–22. Effective stress paths at two representative points

Note that the loading conditions encountered in such a field computation is somewhat different from those in a biaxial test, in the sense that the former case involves τ_{vh}, the in-plane shear stress. However, the computed results have shown that the development of τ_{vh} is not quite significant in comparison with the change in horizontal and vertical normal stresses. τ_{vh} at point A is less than 7 kPa while τ_{vh} at point B is no more than 10 kPa.

Overall, the computation of the Lurie Center excavation yielded reasonable results, in comparison with filed observations and computed results from numerical testing with single

elements. Note that the soil samples used in the triaxial stress probe tests, from which the material parameters used in this computation were developed, were obtained exactly from the Lurie site. Thus, this computation constituted a successful class-C prediction (Lambe 1973). The computation also showed the importance of the two unique ESPs (cf. Section 5.4.2) for an undrained analysis. To perform a finite element analysis under undrained conditions, it is desirable to used soil parameters developed from tests corresponding to these two ESPs.

In summary, this chapter shows the computed model responses in drained/undrained triaxial tests, drained/undrained biaxial tests, an oedometer test involving an unload-reload cycle, and a well-instrumented deep excavation in downtown Chicago. It is shown that this model is successful in simulating various soil tests and is promising in its ability to predict ground movements due to excavations.

6 CONCLUDING REMARKS

This dissertation presents a constitutive model capable of representing soil response in a wide variety of loading modes and strain levels. The key parts of this model include (a) a cross-anisotropic tangent matrix that is consistent with the mechanical nature of frictional materials and (b) a generalized relation that uses length of stress path and stress path direction as internal variables. Soil stiffness evolutions in stress space are quantified by variations in 4 tangent moduli, K, J_v, G and J_s, in shear, compression and small strain zones. Accordingly, small strain and large strain behaviors are defined. The definition of small strain behavior is hypothesized to be related to ageing effects. This definition is applicable to "unstructured" clays wherein only ageing effects cause large variations in stiffness at very small strains, and is applied herein to freshwater, lightly overconsolidated Chicago glacial clays. The effects of direction of loading on each modulus is included in the proposed evolution relations, which allows this model to simulate irrecoverable deformations, and provides an experimentally-based approach to incremental non-linearity.

Model responses are defined in terms of 13 material parameters, most of which have clearly identified physical meanings and can be determined from triaxial experiments. In particular, it is

recommended that parameters be developed from results of CPC, CPE, CQU and CQL tests. It is noted that not all of the 13 parameters are of critical concern in many cases. Typically, three of those four tests would be adequate and selecting the tests largely depends on the specific goal of the application. More work needs to be done to relate some of the non-traditional parameters to those more easily identified with conventional field and laboratory tests.

The proposed model is implemented in the finite element program ABAQUS via UMAT, the user-defined material file. The numerical scheme for stress integration is based on an existing substepping method with automatic error control, with an improvement made to deal with the complexity caused by the stiffness directionality. Furthermore, a consistent tangent matrix is derived for this improved substepping method with the proposed model. This matrix, which becomes remarkably simple with a reasonable and slight approximation, makes the proposed model and substepping method quite stable and efficient when functioning in a finite element computation.

In this dissertation, model responses are exercised in a variety of applications ranging from laboratory test simulations to a computation of ground deformations from a well-instrumented excavation. The simulations of the drained triaxial probe tests have shown that the numerical model is able to find the right stress path in each probe test, indicating that the numerical scheme devised for stress integration is successful. The model has been able to reproduce shear, volumetric and coupling responses over a wide strain range, providing evidence that those assumptions made in directionality relations are reasonable. The computation for a drained

biaxial test has shown that the model satisfactorily reproduces the out-of-plane stress development, and reasonably reproduces the overall stress-strain responses.

The model computation of an unloading-reloading cycle in an oedometer test has shown that the recompression index and the hysteretic behavior are naturally implied by the model without resorting to extra parameters as a result of the nonlinear evolution relations and the extensions for stress reversals.

Under either undrained axisymmetric or plane strain conditions, there exist two unique effective stress paths for a compression mode and an extension mode, respectively. The computations for the undrained tests converge to these two ESPs successfully. It has been showed that the predicted ESPs for the undrained triaxial tests are consistent with the observations made in the drained triaxial tests. Furthermore, the computed ESPs for the undrained biaxial tests have exhibited similar orientations as those for the undrained triaxial tests.

Using the original model parameters based on results of stress probe tests on block samples obtained from the same site, the results of a simulation of a well-instrumented deep excavation in downtown Chicago have shown that the model is promising in its ability to predict both lateral wall deformations and ground settlements behind the wall. The ESPs computed at various locations around the excavation are found consistent with the computed ESPs under undrained plane strain conditions. For the compressible Chicago clays, it is found that the drained RTC and RTE tests are of particular importance in predicting undrained excavations through these materials.

Among possible improvements of this model for future work, a relation of J_v and OCR is needed to account for the differences between highly overconsolidated clays and slightly overconsolidated to normally consolidated clays. The dependency of friction angle on effective mean normal stress can be incorporated to describe a nonlinear failure surface. It is desirable to further investigate the dependency of stiffness parameters on the deviatoric stress, to complement the dependency on the mean normal stress that has been already incorporated into the model. Within the context of numerical implementation, it is worth exploring the general solvability of a problem with directional stiffness. Given an arbitrary set of directionality relations, for instance, does a solution always exist for an arbitrary strain increment? With a discrete directionality relation, what is the solvability condition? Recall that the directionality relation for J_v is intrinsically discrete. Furthermore, room always exists for optimizing the numerical algorithm in implementing the directional stiffness model.

REFERENCES

ABAQUS_Inc. (2003). "ABAQUS Online Documentation: Version 6.4-1."

Abbo, A. J., and Sloan, S. W. (1996). "An automatic load stepping algorithm with error control." *International Journal for Numerical Methods in Engineering*, 39(10), 1737-1759.

Al-Tabbaa, A. (1987). "Permeability and stress-strain response of Speswhite Kaolin," PhD Thesis, University of Cambridge.

Al-Tabbaa, A., and Wood, D. M. (1989). "An experimentally based "bubble" model." *Numerical Models in Geomechanics NUMOG III*, Niagara Falls, Ont. Canada, 91-99.

Anandarajah, A., Sobhan, K., and Kuganenthira, N. (1995). "Incremental stress-strain behavior of granular soil." *Journal of Geotechnical Engineering - ASCE*, 121(1), 57-68.

Atkinson, J. H. (2000). "Non-linear soil stiffness in routine design." *Geotechnique*, 50(5), 487-507.

Atkinson, J. H., Richardson, D., and Stallebrass, S. E. (1990). "Effect of recent stress history on the stiffness of overconsolidated soil." *Geotechnique*, 40(4), 531-540.

Belytschko, T., Moran, B., and Liu, W. K. (1999). *Nonlinear finite element analysis for continua and structures*, Wiley, Chichester.

Bjerrum, L. (1967). "Engineering geology of Norwegian normally-consolidated marine clays as related to settlements of buildings." *Geotechnique*, 17(2), 83-117.

Blackburn, J. T. (2005). "Automated sensing and three-dimensional analysis of internally braced excavations," PhD thesis, Northwestern University, Evanston, IL.

Burland, J. B. (1989). "Small is beautiful - the stiffness of soils at small strains." *Canadian Geotechnical Journal*, 26(4), 499-516.

Burland, J. B., and Georgiannou, V. N. (1991). "Small strain stiffness under generalised stress changes." *Proceedings of the 10th European Conference on Soil Mechanics and Foundation Engineering*, Florence, Italy, 41-44.

Callisto, L., and Calabresi, G. (1998). "Mechanical behaviour of a natural soft clay." *Geotechnique*, 48(4), 495-513.

Calvello, M., and Finno, R. J. (2004). "Selecting parameters to optimize in model calibration by inverse analysis." *Computers and Geotechnics*, 31(5), 411-425.

Chambon, R., Desrues, J., Hammad, W., and Charlier, R. (1994). "CLoE, a new rate-type constitutive model for geomaterials. Theoretical basis and implementation." *International Journal for Numerical and Analytical Methods in Geomechanics*, 18(4), 253-278.

Cho, W. J. (2007). "Ongoing work toward Ph.D. Dissertation," Northwestern University, Evanston, IL.

Chung, C. K., and Finno, R. J. (1992). "Influence of depositional processes on the geotechnical parameters of Chicago glacial clays." *Engineering Geology*, 32(4), 225-242.

Clayton, C. R. I., and Heymann, G. (2001). "Stiffness of geomaterials at very small strains." *Geotechnique*, 51(3), 245-255.

Clough, G. W., Iwabuchi, J., Rad, N. S., and Kuppusamy, T. (1989). "Influence of cementation on liquefaction of sands." *Journal of Geotechnical Engineering-Asce*, 115(8), 1102-1117.

Collins, I. F., and Houlsby, G. T. (1997). "Application of thermomechanical principles to the modelling of geotechnical materials." *Proceedings of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences*, 453(1964), 1975-2001.

Costanzo, D., Viggiani, G., and Tamagnini, C. (2006). "Directional response of a reconstituted fine-grained soil - Part I: experimental investigation." *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(9).

Crisfield, M. A. (1991). *Non-linear finite element analysis of solids and structures*, John Wiley & Sons, Chichester.

Dafalias, Y. F., and Herrmann, L. R. (1982). "Bounding surface formulation of soil plasticity." *Soil mechanics, transient and cyclic loads: constitutive relations and numerical treatment*, G. N. Pande and O. C. Zienkiewicz, eds., Wiley, Chichester, New York, 253-282.

Darve, F. (1991). "The expression of rheological laws in incremental form and the main classes of constitutive equations." *Geomaterials: Constitutive Equations and Modelling*, F. Darve, ed., Elsevier, London, 123-148.

Darve, F., Chau, B., and Dendani, H. (1988). "Incrementally multi-linear and non-linear constitutive relations: a comparative study for practical use." *Numerical Methods in Geomechanics*, Swoboda, ed., Innsbruck, 37-43.

Darve, F., and Labanieh, S. (1982). "Incremental constitutive law for sands and clays - simulations of monotonic and cyclic tests." *International Journal for Numerical and Analytical Methods in Geomechanics*, 6(2), 243-275.

Duncan, J. M., Byrne, P., Wong, K. S., and Mabry, P. (1980). "Strength, stress-strain and bulk modulus parameters for finite element analyses of stresses and movements in soil masses." *UCB/GT/80-01*, Department of Civil Engineering, University of California, Berkeley.

Duncan, J. M., and Chang, C.-Y. (1970). "Nonlinear analysis of stress and strain in soils." *Journal of Soil Mechanics and Foundations Division, ASCE*, 96(SM5), 1629-1653.

Erickson, L. (2006). "Plane Strain Responses of Compressible Chicago Clays," MS thesis, Northwestern University, Evanston.

Fellin, W., and Ostermann, A. (2002). "Consistent tangent operators for constitutive rate equations." *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(12), 1213-1233.

Finno, R. J., Harahap, I. S., and Sabatini, P. J. (1991). "Analysis of braced excavations with coupled finite element formulations." *Computers and Geotechnics*, 12(2), 91-114.

Finno, R. J., and Roboski, J. F. (2005). "Three-dimensional responses of a tied-back excavation through clay." *Journal of Geotechnical and Geoenvironmental Engineering*, 131(3), 273-282.

Finno, R. J., and Tu, X. X. (2006). "Selected topics in numerical simulation of supported excavations." *International Conference on Numerical Simulation of Construction Processes in Geotechnical Engineering for Urban Environment*, Bochum, Germany, 3-19.

Graham, J., and Houlsby, G. T. (1983). "Anisotropic elasticity of a natural clay." *Geotechnique*, 33(2), 165-180.

Gudehus, G. (1979). "Comparison of some constitutive laws for soils under radially symmetric loading and unloading." *Proceedings of the 3rd International Conference on Numerical Methods in Geomechanics*, Aachen, 1309–1323.

Gudehus, G. (2003). "Short manual: (Visco-)hypoplastic element-test-program."

Gudehus, G., Goldsheider, M., and Winter, H. (1977). "Mechanical properties of sand and clay and numerical integration methods: some sources of errors and bounds of accuracy." *Finite Elements in Geomechanics*, G. Gudehus, ed., Wiley.

Hashash, Y. M. A. (1992). "Analysis of deep excavations in clay," PhD, Massachusetts Institute of Technology, Cambridge, MA.

Holman, T. P. (2005). "Small strain behavior of compressible Chicago glacial clay," PhD thesis, Northwestern University, Evanston, IL.

Holtz, R. D., and Kovacs, W. D. (1981). *An introduction to geotechnical engineering*, Prentice-Hall, Englewood Cliffs, N.J.

Houlsby, G. T., and Puzrin, A. M. (2000). "A thermomechanical framework for constitutive models for rate-independent dissipative materials." *International Journal of Plasticity*, 16(9), 1017-1047.

Hsieh, P. G., and Ou, C. Y. (1998). "Shape of ground surface settlement profiles caused by excavation." *Canadian Geotechnical Journal*, 35(6), 1004-1017.

Hueckel, T., and Nova, R. (1979). "Some hysteresis effects of the behavior of geologic media." *International Journal of Solids and Structures*, 15(8), 625-642.

Hughes, T. J. R., and Pister, K. S. (1978). "Consistent linearization in mechanics of solids and structures." *Computers & Structures*, 8(3-4), 391-397.

Jardine, R. J. (1985). "Investigations of pile-soil behaviour with special reference to the foundations of offshore structures," PhD Thesis, University of London.

Jardine, R. J., Potts, D. M., Fourie, A. B., and Burland, J. B. (1986). "Studies of the influence of nonlinear stress-strain characteristics in soil structure interaction." *Geotechnique*, 36(3), 377-396.

Jung, Y. H., Chung, C. K., and Finno, R. J. (2004). "Development of nonlinear cross-anisotropic model for the pre-failure deformation of geomaterials." *Computers and Geotechnics*, 31(2), 89-102.

Kavvadas, M. (1982). "Non-linear consolidation around driven piles in clays," PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Kolymbas, D. (2000). *Introduction to hypoplasticity*, Brookfiled, Rotterdam.

Kuwano, R., and Jardine, R. J. (2002). "On the applicability of cross-anisotropic elasticity to granular materials at very small strains." *Geotechnique*, 52(10), 727-749.

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.

Lambe, T. W. (1973). "Predictions in soil engineering." *Geotechnique*, 23(2), 149-202.

Lanier, J., Caillerie, D., Chambon, R., Viggiani, G., Besuelle, P., and Desrues, J. (2004). "A general formulation of hypoplasticity." *International Journal for Numerical and Analytical Methods in Geomechanics*, 28(15), 1461-1478.

Lings, M. L., Pennington, D. S., and Nash, D. F. T. (2000). "Anisotropic stiffness parameters and their measurement in a stiff natural clay." *Geotechnique*, 50(2), 109-125.

Luccioni, L. X., Pestana, J. M., and Taylor, R. L. (2001). "Finite element implementation of non-linear elastoplastic constitutive laws using local and global explicit algorithms with automatic error control." *International Journal for Numerical Methods in Engineering*, 50(5), 1191-1212.

Menetrey, P., and Willam, K. J. (1995). "Triaxial Failure Criterion for Concrete and Its Generalization." *Aci Structural Journal*, 92(3), 311-318.

Miehe, C. (1996). "Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity." *Computer Methods in Applied Mechanics and Engineering*, 134(3-4), 223-240.

Morgan, A. (2006). "A parametric study of a supported excavation and tunnel connection through Chicago glacial clays," MS thesis, Northwestern University, Evanston, IL.

Mroz, Z., Norris, V. A., and Zienkiewicz, O. C. (1979). "Application of an anisotropic hardening model in the analysis of elastoplastic deformation of soils." *Geotechnique*, 29(1), 1-34.

Naylor, D. J. (1974). "Stresses in nearly incompressible materials by finite elements with application to the calculation of excess pore pressures." *International Journal for Numerical Methods in Engineering*, 8(3), 443-460.

Ng, C. W. W., Leung, E. H. Y., and Lau, C. K. (2004). "Inherent anisotropic stiffness of weathered geomaterial and its influence on ground deformations around deep excavations." *Canadian Geotechnical Journal*, 41(1), 12-24.

Niemunis, A., and Herle, I. (1997). "Hypoplastic model for cohesionless soils with elastic strain range." *Mechanics of Cohesive-Frictional Materials*, 2(4), 279-299.

Perez-Foguet, A., Rodriguez-Ferran, A., and Huerta, A. (2000a). "Numerical differentiation for local and global tangent operators in computational plasticity." *Computer Methods in Applied Mechanics and Engineering*, 189(1), 277-296.

Perez-Foguet, A., Rodriguez-Ferran, A., and Huerta, A. (2001). "Consistent tangent matrices for substepping schemes." *Computer Methods in Applied Mechanics and Engineering*, 190(35-36), 4627-4647.

Perez-Foguet, A., Rodriquez-Ferran, A., and Huerta, A. (2000b). "Numerical differentiation for non-trivial consistent tangent matrices: an application to the MRS-Lade model." *International Journal for Numerical Methods in Engineering*, 48(2), 159-184.

Pestana, J. M., and Whittle, A. J. (1999). "Formulation of a unified constitutive model for clays and sands." *International Journal for Numerical and Analytical Methods in Geomechanics*, 23(12), 1215-1243.

Potts, D. M., and Ganendra, D. (1994). "An evaluation of substepping and implicit stress point algorithms." *Computer Methods in Applied Mechanics and Engineering*, 119(3-4), 341-354.

Potts, D. M., and Gens, A. (1985). "A critical-assessment of methods of correcting for drift from the yield surface in elasto-plastic finite-element analysis." *International Journal for Numerical and Analytical Methods in Geomechanics*, 9(2), 149-159.

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

Puzrin, A. M., and Burland, J. B. (1998). "Non-linear model of small-strain behaviour of soils." *Geotechnique*, 48(2), 217-233.

Puzrin, A. M., and Burland, J. B. (2000). "Kinematic hardening plasticity formulation of small strain behaviour of soils." *International Journal for Numerical and Analytical Methods in Geomechanics*, 24(9), 753-781.

Rammah, K. I., Val, D. V., and Puzrin, A. M. (2004). "Effects of ageing on small-strain stiffness of overconsolidated clays." *Geotechnique*, 54(5), 319-322.

Rechea-Bernal, C. (2006). "Inverse analysis of excavations in urban environments," PhD thesis, Northwestern University, Evanston, IL.

Rowe, P. W. (1962). "Stress-Dilatancy Relation for Static Equilibrium of an Assembly of Particles in Contact." *Proceedings of the Royal Society of London Series a-Mathematical and Physical Sciences*, 269(1339), 500-527.

Schanz, T., Vermeer, P. A., and Boninier, P. G. (1999). "The hardening soil model: Formulation and verification." *Beyond 2000 in computational geotechnics -10 years of plaxis international*, R. B. J. Brinkgreve, ed., Balkema, Rotterdam, 281–296.

Schofield, A. N., and Wroth, P. (1968). *Critical state soil mechanics*, McGraw-Hill, New York.

Shibuya, S. (2002). "A non-linear stress-stiffness model for geomaterials at small to intermediate strains." *Geotechnical and Geological Engineering*, 4(20), 333-369.

Simo, J. C., and Hughes, T. J. R. (1998). *Computational inelasticity*, Springer, New York.

Simo, J. C., and Taylor, R. L. (1985). "Consistent tangent operators for rate-independent elastoplasticity." *Computer Methods in Applied Mechanics and Engineering*, 48(1), 101-118.

Simpson, B., Oriordan, N. J., and Croft, D. D. (1979). "Computer-model for the analysis of ground movements in London clay." *Geotechnique*, 29(2), 149-175.

Sloan, S. W. (1987). "Substepping schemes for the numerical integration of elastoplastic stress-strain relations." *International Journal for Numerical Methods in Engineering*, 24(5), 893-911.

Smith, P. R., Jardine, R. J., and Hight, D. W. (1992). "The yielding of bothkennar clay." *Geotechnique*, 42(2), 257-274.

Stallebrass, S. E. (1990). "Modelling the effect of recent stress history on the deformation of overconsolidated soils," PhD Thesis, City University, London.

Stallebrass, S. E., and Taylor, R. N. (1997). "The development and evaluation of a constitutive model for the prediction of ground movements in overconsolidated clay." *Geotechnique*, 47(2), 235-253.

Tamagnini, C., and Pane, V. (1999). "On the influence of soil non-linearity on ground movements induced by excavations in cohesive soils." *2nd International Symposium on Pre-Failure Deformation Characteristics of Geomaterials*, Torino, Italy, 573-580.

Tamagnini, C., Viggiani, G., and Chambon, R. (1999). "A review of two different approaches to hypoplasticity." *Constitutive Modelling of Granular Materials including Developments and Perspectives of Hypoplasticity*, D. Kolymbas, ed., Springer, 107-145.

Truesdell, C. (1955). "Hypo-Elasticity." *Journal of Rational Mechanics and Analysis*, 4(1), 83-131.

Tu, X., and Finno, R. J. (2007). "Modeling shear & volumetric responses in stress probe tests." *Geo-Denver*, Denver, CO, in press.

Viggiani, G., and Tamagnini, C. (1999). "Hypoplasticity for modeling soil non-linearity in excavation problems." *Pre-failure Deformation Characteristics of Geomaterials*, M. B. Jamiolkowski, R. Lancellotta, and D. Lo Presti, eds., Balkema, Rotterdam, 581-588.

Viggiani, G., and Tamagnini, C. (2000). "Ground movements around excavations in granular soils: a few remarks on the influence of the constitutive assumptions on FE predictions." *Mechanics of Cohesive-Frictional Materials*, 5(5), 399-423.

vonWolffersdorff, P. A. (1996). "A hypoplastic relation for granular materials with a predefined limit state surface." *Mechanics of Cohesive-frictional Materials*, 1(3), 251-271.

Whittle, A. J. (1987). "A constitutive model for overconsolidated clays with application to the cyclic loading of friction piles," Sc.D., Massachusetts Institute of Technology, Cambridge, MA.

Whittle, A. J. (1990). "A constitutive model for overconsolidated clays." *MITSG 90-15*, Sea Grant College Program, Massachusetts Institute of Technology, Cambridge, MA.

Whittle, A. J., Hashash, Y. M. A., and Whitman, R. V. (1993). "Analysis of Deep Excavation in Boston." *Journal of Geotechnical Engineering - ASCE*, 119(1), 69-90.

Whittle, A. J., and Kavvadas, M. J. (1994). "Formulation of MIT-E3 constitutive model for overconsolidated clays." *Journal of Geotechnical Engineering - ASCE*, 120(1), 173-198.

Wood, D. (1990). *Soil behaviour and critical state soil mechanics*, Cambridge University Press.

Wu, H., Wang, W., and Shi, Q. (1998). "Stress-strain behaviour of soils at small strains." *Yantu Gongcheng Xuebao/Chinese Journal of Geotechnical Engineering*, 20(1), 70-75.

Zhao, J. D., Sheng, D. C., Rouainia, M., and Sloan, S. W. (2005). "Explicit stress integration of complex soil models." *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(12), 1209-1229.

APPENDICES

A. MAJOR SMALL STRAIN MODELS

A.1 MIT-E3 MODEL

MIT-E3 is a successor of MIT-E1 that was developed by Kavvadas (1982) in MIT. To be consistent with results from soil laboratory, Kavvadas modified the Modified Cam-clay model (MCC) and replaced the isotropic yield surface with a distorted or anisotropic ellipsoidal surface in stress space. The associated hardening rule was designed such that with plastic volumetric strain increments, the yield surface rotates as well as expands in stress space. The larger the distance between the axis of the yield surface and the current stress point, the larger the rotation rate of the yield surface during hardening. Such an anisotropic yield surface was designed to simulate the stress-induced anisotropy of natural soil.

Basically, MIT-E1 improved simulations for normally consolidated clays. However, it is not applicable to overconsolidated clays at small strains, because the simulation inside the yield surface was oversimplified by linear isotropic elasticity, as used in MCC. There were three main issues in characterization of overconsolidated clays. First, most overconsolidated clays are highly nonlinear even at small strain levels to which linear elasticity does not apply. Second, it

was noted that when reloaded beyond the maximum past pressure, yielding of overconsolidated clays is rather smooth, not pronounced as predicted by a traditional elasto-plastic model. Lastly, the volumetric and shear behavior needs to be coupled because overconsolidated clays can fail due to excessive pore water pressure accumulated under cyclic loading.

To address these issues, Whittle (1987) updated MIT-E1 to MIT-E3, with focus on model responses inside the traditional yield surface. The major modifications include applying the concept of bounding surface plasticity (Dafalias and Herrmann 1982) and replacing linear isotropic elasticity with perfectly hysteretic elasticity (Hueckel and Nova 1979). The bounding surface plasticity is shown in Fig. A - 1, wherein p' is effective mean normal stress and q is the deviatoric stress. The *bounding surface* in MIT-E3 is exactly the yield surface in MIT-E1, and corresponds to the locus of maximum past stress states. The *load surface* functions as a classical yield surface, i.e. the border between elastic and elasto-plastic regions in stress space. And the load surface is designed such that the current stress point is always on the load surface. As shown in Fig. A - 1, point C represents current stress state, and point I is the image point of point C on the bounding surface. There is a mapping rule in terms of the distance of these two points to determine both magnitude and orientation of plastic strain during loading, e.g. from point C to point L as shown in Fig. A - 1. By exploiting the bounding surface plasticity, the coupling of volumetric and shear behavior can be achieved and smooth yielding can be produced as well.

During unloading such as the stress path from point C to point U, the model response is described by hysteretic elasticity. Basically, the hysteretic elasticity creates nonlinearity in

constitutive relation, which is desirable. However, this elastic relation is isotropic in nature. In other words, it cannot simulate the stiffness anisotropy of Chicago clays observed at very small strains in Holman's tests. Here, it is worthy to mention that this model was initially proposed to study the performance of offshore friction piles supporting tension leg platforms where soil behavior under cyclic loading controls the responses of the system. Since anisotropic stiffness, e.g. cross anisotropy, enables coupling between volumetric and shear responses, the solution for cyclic loading condition can be further facilitated by applying stiffness anisotropy.

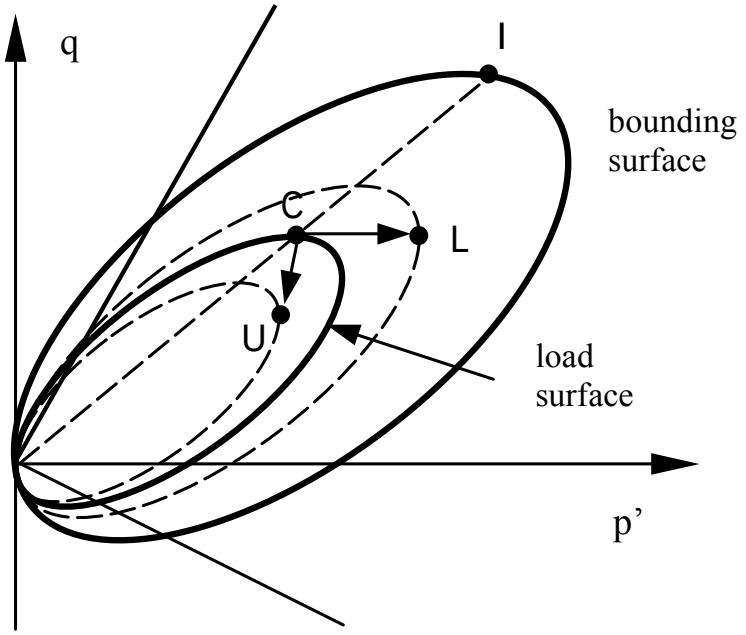

Fig. A - 1. Bounding surface plasticity in MIT-E3

A.2 THREE SURFACE KINEMATIC HARDENING MODEL

Kinematic hardening means that during loading the yield surface moves in stress space without change in size. The concept is based on the experimental evidence that "elastic" or

quasi-elastic range of natural soil is typically very small and its size is relatively independent of the starting stress (Wood 1990). The kinematic nature makes the yield surface behave like a bubble floating in stress space. That is why this class of constitutive relation is also referred to as 'bubble' model.

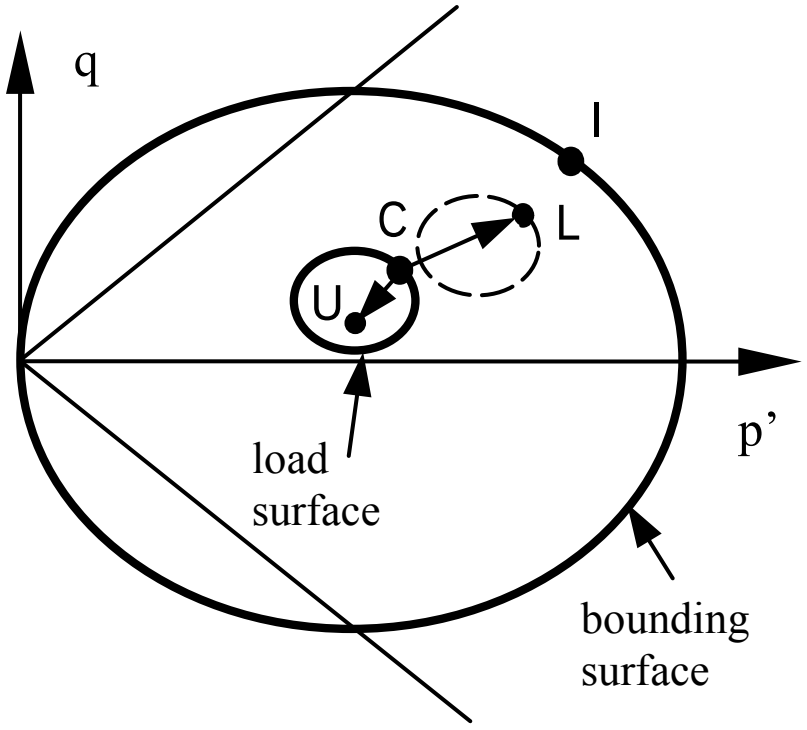

Fig. A - 2. The early version of 'Bubble' model (Al-Tabbaa, 1987)

Similar to MIT-E3, the 'bubble' model was developed in two stages as well. The early version of the model was developed by Al-Tabbaa (1987). As shown in Fig. A - 2, inside a bounding surface of MCC type, one kinematic ellipsoidal surface was introduced as a load surface. As before, the load surface serves as the border between the elastic and elasto-plastic regions in stress space. As shown in Fig. A - 2, for loading stress path, e.g. from point C to point

L, the load surface moves with the current stress point, while for unloading path, e.g. from point C to point U, the surface remains stationary. The point I is the image point of the current stress point C on the bounding surface. The distance between these two points is used as an internal variable in a bounding surface plastic relation to quantify the plastic strain during loading. A transition law of the load surface was made to ensure that it never intersects the bounding surface. Continued loading after the two surface contact will reproduce exactly the same response as MCC.

The current version of the 'bubble' model was presented by Stallebrass (1990) with emphasis on stress history issues. In this model, a third surface, called the history surface, is added between the load and bounding surfaces. Actually, the movement of this history surface is controlled by a translation law of the same form as that used for the load surface. The relative position of the history surface to the load surface is incorporated into the formula for plastic modulus to control the rate of stiffness degradation.

Fig. A - 3 shows how this model works for two different stress paths. In Fig. A - 3 (a), the path is A-B-E while in Fig. A - 3 (b) the path is C-B-E. With respect to the identical portion B-E, portion A-B and C-B are termed 'recent stress history' (Atkinson et al. 1990). As shown, after the recent history of A-B, the model response is elasto-plastic, while after C-B, the response is elastic. The portion B-D shown in Fig. A - 3 (b) is elastic and during it, the load surface does not move, as previously mentioned. That is why there is a plateau of constant stiffness at the initial portion of the solid curve in Fig. A - 3 (c). Thus at the same point B, lower stiffness is produced after A-B while higher stiffness is produced after C-B, as shown in Fig. A - 3 (c). This result is

consistent with the experimental observations that the smaller the deviation from the recent stress path (history), the softer the soil response, and vice versa. Furthermore, in spite of the difference in recent stress history, the model will reach an identical state after continued loading along path B-E. The starting point of the identical state, point E in Fig. A - 3 (c), is called state of 'swept-out memory', beyond which the influence of recent stress history will be eliminated (Gudehus et al. 1977). The history surface is designed to represent the locus of that state.

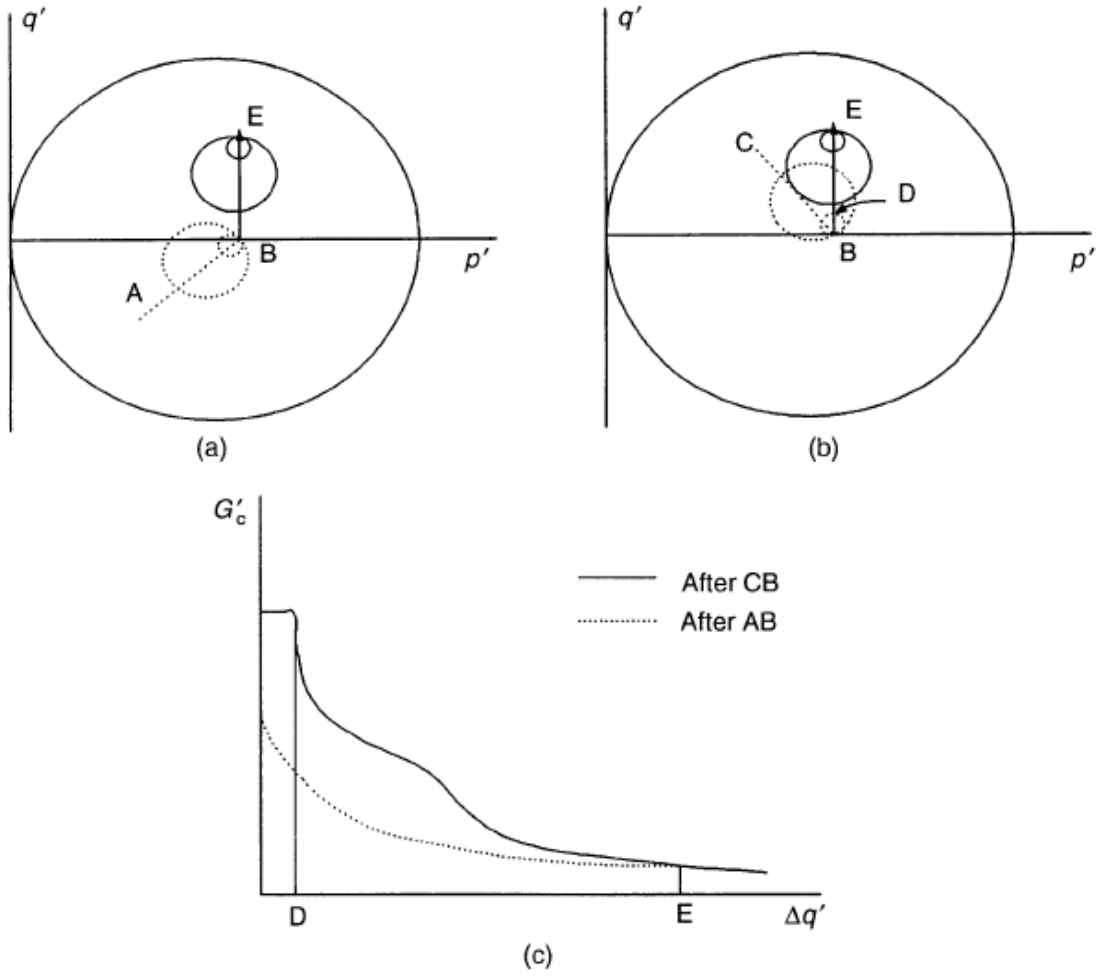

Fig. A - 3. The variation of stiffness with recent stress history (Stallebrass and Taylor 1997)

It must be pointed out that the elastic model response inside the load surface is still isotropic. So, this model can do no better in simulating stiffness anisotropy at very small strains than a traditional elasto-plastic model. However, the 'bubble' model does address an associated and important issue, i.e. stiffness anisotropy is highly dependent upon recent stress history.

A.3 HYPOPLASTIC MODEL

The two models previously reviewed are extensions to a traditional elasto-plastic relationship. The capability of models of elasto-plastic type is typically gained by sacrificing simplicity, which is manifested by incorporating complicated formulations and a large number of material constants. Furthermore, there seem to be difficulties in applying the elasto-plastic relation to natural soils, because most soils do not exhibit an elastic range even at small strain levels. As a consequence, the decomposition of deformation into elastic and plastic parts is questionable, and does not necessarily reflect the reality.

The development of hypoplastic theory was exactly motivated by the above thoughts. The prefix 'hypo-' means non-existence of yield surface (Kolymbas 2000). The work was initiated in Karlsruhe in late 70's by Kolymbus and his associates. The initial model was designed for granular materials (vonWolffersdorff 1996; Wu et al. 1998; Kolymbas 2000), and was extended to cohesive materials later on (Niemunis and Herle 1997; Wu et al. 1998; Gudehus 2003). The constitutive relation of hypoplasticity can be expressed as follows:

$$\overset{\circ}{T}_{ij} = L_{ijkl} D_{kl} + N_{ij} |D| \qquad \textbf{(A-1)}$$

where $\overset{\circ}{T}_{ij}$ is co-rotational or Jaumann stress rate, D_{kl} is Euler's stretching tensor, $|D|$ equals $\sqrt{D_{kl}D_{kl}}$, and L and N are functions of T_{ij} and other state variables such as void ratio. Owing to the second term in Eq. (A-1), the stress rate is a function nonlinear in D. Furthermore, the second nonlinear term leads to the result that $-\overset{\circ}{T}(D) \neq \overset{\circ}{T}(-D)$. If the relation is invertible, then it is equivalent to state that the recovered strain is not equal to the incurred strain in a complete stress cycle. The inelastic behavior is thus achieved without the decomposition of strain measure into elastic and plastic parts, as in elasto-plasticity.

The rate form of its general equation gives hypoplasticity advantages in dealing with the small strain behaviors mentioned before. Stiffness anisotropy is naturally controlled by the tensorial function *L*. Meanwhile, the function *N* can be designed to describe a particular nonlinear relation and stiffness degradation at small strain levels. Furthermore, the influence of recent strain history has also been investigated within hypoplastic framework (Niemunis and Herle 1997). In terms of large strain behavior, it has been shown by Von Wolffersdorff (1996) that appropriate *L* and *N* can lead to implicit failure criteria.

B. CALCULATION OF TANGENT MODULI FROM TEST DATA

It is straightforward matter to calculate the secant moduli from data obtained in triaxial tests. When calculating the tangent moduli, however, one has to smooth the curve first, because of unavoidable fluctuation in the stress-strain curve. Approximating the tangent by directly

connecting two consecutive data points usually yields fluctuating results. Hence, it is necessary in most circumstances to conduct a regression analysis to obtain an analytical function that fits the raw curve. The tangent modulus is then calculated by differentiating the regression function.

An example is given in Fig. B - 1, wherein polynomial function is used to smooth the Δp-ε_v curve. The order of the polynomial function depends on the non-linearity of the curve. Normally, the order is increased until a satisfactory match is reached. If the match cannot be improved by increasing order, the whole curve is divided into several segments and different polynomial function is used to fit different curve segments, as shown in Fig. B - 1. By differentiating these functions, the tangent modulus K can be obtained for the whole stress/strain range.

Note that herein the form of the regression function is not crucial. One can use any type of function for the curve fitting if there is an apparent advantage. In addition, it is unnecessary, and difficult in most cases, to use a single function to fit the entire curve. In most cases, a visual fit is what is needed to make a judgment about a “satisfactory” fit.

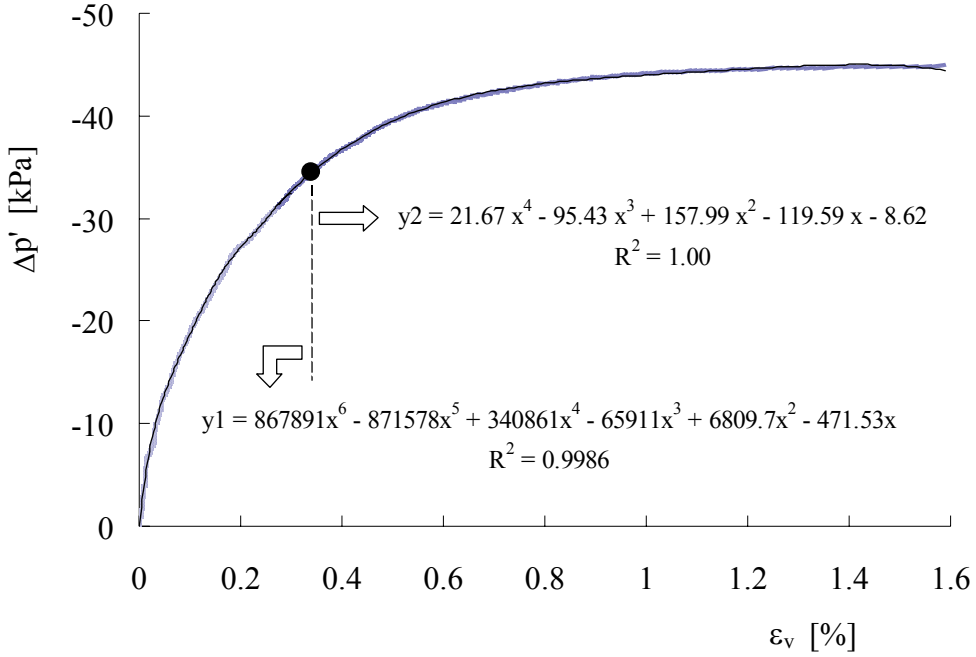

Fig. B - 1. Regression method used to smooth stress-strain curve

C. MODEL CODED IN UMAT FILE (ABAQUS)

```
C
C **----------------------------------------------------------------**
C ||        An Incrementally Non-linear Model for Clays           ||
C ||   with Directional Stiffness and a Small Strain Emphasis     ||
C **----------------------------------------------------------------**
C
C                Copyright by Xuxin Tu & Richard Finno  2006
C                              All Rights Reserved
C
C                contact: Xuxin.Tu@gmail.com
C -------------------------------------------------------------------------------------------
C
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
```

```
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      ! NPROPS = 15; NSTATV = 15
          !
          DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
          REAL*8 STRESS, DSTRAN, STRAN, DDSDDE, PROPS, STATEV
          REAL*8 TIME
C
C
C                                          UMAT Interface
C ***********************************************************************
C                                          Variable Declaration
C
          ! ----------- Material Parameters (External) -----------
          !
          !
          REAL*8 G_ref, Jv_ref, K_ref
                  ! refer to Es_min at p_a
          REAL*8 LSP_ref
                  ! refer to LSP_min at p_a (normalized by p_a already)
          REAL*8 mu_up
                  ! mu for stress path leading to the upper F.S.
          REAL*8 nn
                  ! E/pa=Eref*(p'/pa)^nn          !
          REAL*8 r_0s, r_x, r_LSP, r_mu
                  !         r_0s=E_0/E_s;          r_x=Emax/Emin;          r_LSP=LSP_max/LSP_min;
r_mu=mu_down/mu_up
          REAL*8 faf
                  ! frictonal angle at failure [degree]
          REAL*8 lambda
                  ! Dstrn = lambda*D(ln(strss))
          REAL*8 OCR
                  ! over-condolidated ratio
          REAL*8 k_nc
                  ! lateral stress ratio for normal consolidation
          !
          ! ----------- Material Parameters (Internal) -----------
```

```
      !
      REAL*8 G_max, G_min, K_max, K_min, Jv_max, Jv_min
            ! used in directionality relations for Estar_s
      REAL*8 LSP_max, LSP_min
      REAL*8 Mf, Mf_upper, Mf_lower, beta_upper, beta_lower
            ! Mf_upper --> beta_upper; Mf_lower --> beta_lower
      !
      ! -------------- State Variables -------------
      !
      REAL*8 Beta_pre, LSP_pre
            ! values in the previous increment
      REAL*8 p_0, p0_pre
            ! p_0: p' value at the starting point of the most recent 'monotonic' path
            !          updated upon a path reversal
            ! p0_pre: previous p_0
      REAL*8 s_p
            ! s_p:   preconsolidation stress; from OCR & p0
      REAL*8 Ue
            ! excess pore water pressure
      INTEGER flag_drain
            ! 1: drained; 0: undrained
      REAL*8 WT
            ! vertical coordinate of water table
      REAL*8 Uh
            ! hydrostatic pressure
      !
      ! -------------- Other Global Variables -------------
      !
      REAL*8 K, Jv, G, Js
            ! Tangent moduli under axisymmetric condition
            ! Dev = Dp/K+Dq/Jv; Des = Dq/G+Dp/Js
      REAL*8 stiff(NTENS,NTENS)
            ! Cross-anisotropic matrix for general conditions
      REAL*8 strs(NTENS)
            ! used in MAIN program
C
C                                    Variable Declaration
C ********************************************************************
C                                    Variable Assignment
C
      !
      ! ------------- Numerical Parameters ------------
      !
```

```
REAL*8, PARAMETER :: pi=3.14159265D0
REAL*8, PARAMETER :: p_a= 100.D0
        ! atmospheric pressure [kPa]
REAL*8, PARAMETER :: K_fld = 2.D6      ! Bulk Modulus of water: 2.D6 kPa
        ! "Bulk modulus" for pore water
REAL*8, PARAMETER :: TOL_ZERO = 1.D-18
REAL*8, PARAMETER :: TOL_r=1.0D-3
        ! for substepping
REAL*8, PARAMETER :: TOL_Beta = 1.0D-3
        ! for Beta_Converge
REAL*8, PARAMETER :: K_f = 50.D0, G_f = 1.D0,
1                                       Jvf_up = 1000.D0, Jvf_low = -10000.D0
        ! E_f regarded as independent of materials
INTEGER, PARAMETER :: m_max=12
        ! record size in "Beta_Converge", for
INTEGER, PARAMETER :: KSUB_max = 100000
        ! in case exceptions occur
REAL*8, PARAMETER :: r_JvK = 10.D0, r_JsK = 30.D0
        ! the ratio of Jv/Js to K in Compression Zone
!
!
! ----------- Material Properties -------------
!
G_ref = 19.D0           ! 76.D0 for G0
K_ref = 48.D0           ! 115.D0 for K0
Jv_ref = 116.D0                 ! 462.D0 for Jv0
LSP_ref = 0.07D0
mu_up = 8.D0
r_0s = 4.D0
r_x = 5.D0
r_LSP = 3.D0
r_mu = 0.5D0
nn = 0.55D0
faf = 28.D0    ! RTXC: 29.D0, RTXE: 26.5
OCR = 1.5D0
lambda = 0.04D0
k_nc = 0.7D0
!
Mf = DSIN(faf*pi/180.D0)
        ! Mf = (sigma1f-sigma3f)/((sigma1f+sigma3f)
SELECT CASE (NTENS)
CASE (4)
        Mf_upper = 2.D0* Mf
```

```
            Mf_lower = Mf_upper
        CASE (6)
            Mf_upper = 6.D0*Mf/(3.D0-Mf)
            Mf_lower = 6.D0*Mf/(3.D0+Mf)
        END SELECT
        !
        Beta_upper = DATAN(Mf_upper)
        Beta_lower = 2.D0*pi-DATAN(Mf_lower)
            ! Mf_lower is positive & beta: [0, 2*pi)
        !
        WT = 0.D0
        flag_drain = 0
        !
        !
C                                       Variable Assignment
C ************************************************************************
C                   Main Program
C
        !!! DEBUG
        OPEN(UNIT=16,FILE='E:\DEBUG.txt',POSITION='APPEND')
        !
        IF (NOEL==1 .AND. NPT==1 .AND. KSTEP==1 .AND. KINC==1) THEN
            CALL DEBUG_GENERAL
        END IF
        !
        !
            IF(KSTEP==28 .AND. KINC==28) THEN
            ! (NOEL==122 .AND. NPT==2)
            ! (KSTEP==28 .AND. KINC==28)
                CALL DEBUG_GENERAL
            END IF
            !
            IF (KSTEP == 1) THEN
                CALL ISO_ELA(3.0D4, 0.4D0)              ! nu for s'h/s'v = 0.5
            ELSE IF (KSTEP == 2) THEN
                CALL INI_STATEV
                CALL ISO_ELA(3.0D4, 0.495D0)
            ELSE IF (KSTEP == 4 .OR. KSTEP == 7 .OR. KSTEP == 10) THEN
                CALL ISO_ELA(3.0D4, 0.495D0)
                !
            ELSE IF (Norm(DSTRAN) < TOL_ZERO) THEN
            ! DSTRAN = 0
                CALL READ_STATEV
```

```
                CALL Sign_Change(strs, STRESS)
                CALL Eff_Stress(strs)
                CALL P_Dependency
                !
                CALL Axisymmetry(strs, LSP_pre, beta_pre)
                CALL Cross_Anisotropy
                DDSDDE = stiff
            ELSE
                !
                CALL READ_STATEV
                CALL Sign_Change(Strs, STRESS)
                CALL Eff_Stress(Strs)
                CALL P_Dependency
                !
                CALL Integration_Scheme(strs)
                CALL WRITE_STATEV
            END IF
            !
        CLOSE(16)
        !
        RETURN
C
C           Main Program
C **************************************************************************
C                                   Internal Procedures
C
        CONTAINS
C -------------------------------------------------------------
C       [ Internal Subroutine ]     Integration_Scheme
C -------------------------------------------------------------
        SUBROUTINE Integration_Scheme(strs_t)
        ! An automatic substepping scheme extended for incremental non-linearity
        ! strs_t in geomechanics sign convention
        ! -------------------- Main Components----------------------
        ! Beta_Converge, Dstrs_CALC, OUT_FS,
        !
        REAL*8 strs_t(NTENS)        ! passed in
        REAL*8 Dx(NTENS)            ! strain incrmnt
        REAL*8 y(2,NTENS), Dy(2,NTENS)          ! *(1,:): k; *(2,:): k+1
        REAL*8 stiff1(NTENS,NTENS)              ! record stiff in the 1st guess
        REAL*8 beta, LSP, LSP2, D_LSP, Dp, Dq
        REAL*8 D_beta               ! for stress reversal treatment
        REAL*8 t, Dt, r             ! r: estimated ratio for next strain incrmnt
```

```
REAL*8 Err_y(NTENS), norm_Err, norm_y, Err_Ratio
INTEGER KSUB       ! KSUB:         substep #
INTEGER flag_end   ! for each substep: 1=converged, 0=unconverged
INTEGER flag_fail  ! 1= upper F.S.; 2= lower F.S.; 0= pre-failure
!
! ------------------------------------------------
!
CALL Sign_Change(Dx, DSTRAN)
        ! Using sign convention of geomechanics, i.e. compressive = (+)
y(1,:) = strs_t
!
!  ------------- failure treatment ---------------
flag_fail = Failure(y(1,:))
IF(flag_fail /= 0) THEN
!
        !!! DEBUG
        CALL DEBUG_SUBSTEP(KSUB)
        !
        CALL OUT_FS(Dx, flag_fail, y(1,:), beta)
        ! update y(1,:), beta & stiff
        WRITE (16,'("OUT OF F.S. at beginning")')
        GOTO 828
END IF
! ---- variable initialization
LSP = LSP_pre
beta = Beta_pre
Dt = 1.D0
t = 0.D0
r = 1.D0
KSUB = 0
!
!  ------------ extended substepping -------------
DO WHILE (t < 1.D0 .AND. KSUB < KSUB_max)
        flag_end = 0        ! 0:inaccurate; 1: accurate
        KSUB = KSUB+1
        !
        Dt =r*Dt
        IF (t+Dt > 1.D0) THEN
                Dt = 1.D0 - t
        END IF
        !
        CALL Beta_Converge(y(1,:), LSP, Dt*Dx, Dy(1,:), beta)
                ! update beta, Dy(1,:) & stiff
```

```
!
! ---------- stress reversal treatment ----------
D_beta = beta - beta_pre
CALL Dbeta_Correction(D_beta)
        ! make D_beta fall in [-pi, pi]
D_beta = DABS(D_beta)
!
IF (D_beta > 0.5D0*pi) THEN
        ! stress reversal
        LSP = 0.D0
        p0_pre = p_0
        p_0 = pp(y(1,:))
        CALL P_Dependency
        !
        IF (p_0 > s_p) THEN
                s_p = p_0
        END IF
        !
        CALL Beta_Converge(y(1,:), LSP, Dt*Dx, Dy(1,:), beta)
                ! update beta, Dy(1,:) & stiff
        beta_pre = beta
        LSP_pre = LSP
        !
END IF
!
! ------------ original substepping -------------
stiff1 = stiff
Dp = pp(Dy(1,:))
Dq = qq(Dy(1,:))
D_LSP = DSQRT(Dp*Dp + Dq*Dq)
!
DO WHILE (flag_end == 0)
! in this LOOP, y(1,:) doesn't change
        !
        y(2,:) = y(1,:)+Dy(1,:)
        LSP2 = LSP + D_LSP
        !
        CALL Dstrs_CALC(y(2,:), LSP2, beta, Dt*Dx, Dy(2,:))
        ! calculate Dy(2,:) & stiff
        !
        Err_y = (Dy(2,:)-Dy(1,:))/2.D0
        !
        norm_Err = Norm(Err_y)
```

```
        norm_y = Norm(y(1,:))
        Err_Ratio = DABS(norm_Err/norm_y)
        r = 0.8D0*DSQRT(TOL_r/Err_Ratio)
        !
        IF (Err_Ratio < TOL_r) THEN
        ! converged
                flag_end = 1
                !
                IF (r > 2.D0) THEN
                        r = 2.D0
                END IF
        ELSE
        ! unconverged
                IF (r < 0.1D0) THEN
                        r = 0.1D0
                END IF
                !
                Dy(1,:) = r*Dy(1,:)
                D_LSP = r*D_LSP
                !
                Dt = r*Dt
                !
        END IF
        !
END DO
!
Dy(1,:) = (Dy(1,:)+Dy(2,:))/2.D0
Dp = pp(Dy(1,:))
Dq = qq(Dy(1,:))
D_LSP = DSQRT(Dp*Dp + Dq*Dq)
!
LSP = LSP + D_LSP
t = t+Dt
y(1,:) = y(1,:) + Dy(1,:)
stiff = 0.5D0*(stiff1+stiff)
        ! secant stiffness matrix for this substep
        ! an approximation of ATM
!
! ------------- failure treatment ----------------
flag_fail = Failure(y(1,:))
IF(flag_fail /= 0) THEN
        !!! DEBUG
        CALL DEBUG_SUBSTEP(KSUB)
```

```
                    !
                    CALL OUT_FS((1.D0-t)*Dx, flag_fail, y(1,:), beta)
                    ! update y(1,:), beta & stiff
                    WRITE (16,'("OUT OF F.S. within substepping")')
                    GOTO 828
                END IF
        !
        END DO
        !
        !------------ Final Update -------------
828     Beta_pre = beta
        LSP_pre = LSP
        CALL Final_Update(y(1,:), Dx)
        ! Update Ue, STRESS & DDSDDE
        !
        !!! DEBUG
        IF (NOEL==1 .AND. NPT==1) THEN
                CALL DEBUG_SUBSTEP(KSUB)
        END IF
        !
        END SUBROUTINE Integration_Scheme
C
C -------------------------------------------------------------------
C               [ Internal Subroutine ]         Beta_Converge
C -------------------------------------------------------------------
        SUBROUTINE Beta_Converge(strs_t, LSP_t, Dstrn, Dstrs, beta_t)
        ! calculate Dstrs & beta_t using Newton method and numerical differentiation
        ! update k, Jv, G, Js & stiff
        ! ---------------- main components -----------------
        ! Dstrs_CALC, FN_Beta, Dbeta_Correction, Beta_Period
        !
        REAL*8 strs_t(NTENS), Dstrn(NTENS), Dstrs(NTENS)
        REAL*8 LSP_t, beta_t
        REAL*8 beta_rec(m_max)  ! beta_t
        REAL*8 x1, x2, y1, y2, y_p, Dx, x_disturb
                ! y_p: y prime, numerical differentiation; y_p = (y2-y1)/(x2-x1)
                ! x2 = x1 + x_disturb
                ! Dx = -y1/y_p
        REAL*8 Dp, Dq
        INTEGER flag_end
                ! 1: converged, 0: unconverged
        INTEGER m
                ! m: Newton iteration #
```

```
!
x_disturb = 0.01D0*TOL_Beta
beta_rec(1:m_max) = 0.D0
m = 1
beta_rec(1) = beta_t
!
x1 = beta_t
!
CALL Dstrs_CALC(strs_t, LSP_t, x1, Dstrn, Dstrs)
beta_t = FN_Beta(Dstrs)
!
y1 = beta_t-x1
!
CALL Dbeta_Correction(y1)
        ! makes y1 fall in [-pi, pi]
IF (DABS(y1) < TOL_Beta) THEN
        flag_end = 1
ELSE
        flag_end = 0
END IF
!
!
DO WHILE (flag_end == 0 .AND. m < m_max)
        !
        x2 = x1 + x_disturb
        !
        CALL Dstrs_CALC(strs_t, LSP_t, x2, Dstrn, Dstrs)
        beta_t = FN_Beta(Dstrs)
        !
        y2 = beta_t-x2
        CALL Dbeta_Correction(y2)
                ! makes y1 fall in [-pi, pi]
        !
        y_p = (y2-y1)/(x2-x1)
        Dx = -y1/y_p
        beta_t = x1 + Dx
        !
        CALL Beta_Period(beta_t)
                ! makes beta_t fall in [0, 2pi)
        ! ------------------------------------
        x1 = beta_t
        !
        CALL Dstrs_CALC(strs_t, LSP_t, x1, Dstrn, Dstrs)
```

```
                ! update stiff also
            beta_t = FN_Beta(Dstrs)
            !
            y1 = beta_t-x1
            !
            CALL Dbeta_Correction(y1)
                ! makes y1 fall in [-pi, pi]
            IF (DABS(y1) < TOL_Beta) THEN
                flag_end = 1
            ELSE
                flag_end = 0
            END IF
            !
            m = m + 1
            beta_rec(m) = x1
            !
        END DO
        !
        beta_t = x1
            ! must be consistent with Dstrs
        !
        !!! DEBUG ----------
        !IF(m <= m_max) THEN
            ! WRITE (16,'("Beta:  ", F6.3)') beta_t
        !       WRITE (16,'("beta_rec():  ", 12F6.3)') beta_rec
        !END IF
        !
        !
        END SUBROUTINE Beta_Converge
C
C---------------------------------------------------------------------
C       [ Internal Subroutine ]         Dbeta_Correction
C---------------------------------------------------------------------
        SUBROUTINE Dbeta_Correction(Dbeta)
        ! makes Dbeta fall in [-pi, pi]
        ! Dbeta: the difference between 2 betas
        ! Note: any Dbeta should fall in (-2pi, 2pi)
        ! ------------------------------------
        REAL*8 Dbeta
        !
        IF (Dbeta > pi) THEN
            Dbeta = Dbeta - 2.D0*pi
        ELSE IF (Dbeta < -pi) THEN
```

```
            Dbeta = Dbeta + 2.D0*pi
        END IF
        !
        !
        END SUBROUTINE Dbeta_Correction
C
C---------------------------------------------------------------------
C       [ Internal Subroutine ]          Beta_Period
C---------------------------------------------------------------------

        SUBROUTINE Beta_Period(beta_t)
        ! make beta_t fall in [0, 2*pi)
        ! ------------------------------------
        REAL*8 beta_t
            DO WHILE (beta_t < 0.D0)
                beta_t = beta_t + 2.0D0*pi
            END DO
            !
            DO WHILE (beta_t >= 2.D0*pi)
                beta_t = beta_t - 2.0D0*pi
            END DO
        !
        END SUBROUTINE Beta_Period
C
C ---------------------------------------------------------------------
C           [ Internal Subroutine ]     Dstrs_CALC
C ---------------------------------------------------------------------
        SUBROUTINE Dstrs_CALC(strs_t, LSP_t, beta_t, Dstrn, Dstrs)
        ! calculate Dstrs at strs_t based on LSP_t, beta_t & Dstrn
        ! updated quantities: 'K, Jv, G, Js', stiff(*,*)
        ! ----------------------------------------
        REAL*8 strs_t(NTENS), Dstrs(NTENS), Dstrn(NTENS)
        REAL*8 LSP_t, beta_t
        !
        CALL Axisymmetry(strs_t, LSP_t, beta_t)
            ! calculate K, Jv, G, Js
        CALL Cross_Anisotropy
            ! calculate stiff(*,*) using K, Jv, G, Js
        CALL Dstrs_Dstrn(Dstrs, Dstrn)
            ! calculate Dstrs using stiff(*,*) & Dstrn
        !
        END SUBROUTINE Dstrs_CALC
C
```

```
C -----------------------------------------------------------------
C               [ Internal Subroutine ]                 Axisymmetry
C -----------------------------------------------------------------
      SUBROUTINE Axisymmetry(strs_t, LSP_t, beta_t)
      ! calculate tangent moduli - K,Jv,G,Js - under axisymmetric condition
      ! --------------- main components --------------
      ! Evolution_CZone, Evolution_SZone, LNR
      !
      REAL*8 strs_t(NTENS), LSP_t, beta_t
      REAL*8 beta_min, beta_max
            ! specify transition zones between Shear & Compression Zones
      REAL*8 K_1, K_2, Jv_1, Jv_2, G_1, G_2, Js_1, Js_2
            ! for transition between Shear & Compression Zones
      !
      beta_min = 1.01D0*beta_upper
      beta_max = 0.995D0*beta_lower
      !
      !
      IF (beta_t <= beta_upper .OR. beta_t >= beta_lower) THEN
      ! In Compression Zone
            CALL Evolution_CZone(strs_t, LSP_t, beta_t)
            !
      ELSE IF (beta_t >= beta_min .AND. beta_t <= beta_max) THEN
      ! In "PURE" Shear Zone
            CALL Evolution_SZone(strs_t, LSP_t, beta_t)
            !
      ELSE IF (beta_t > beta_upper .AND. beta_t < beta_min) THEN
      ! upper part of transition
            CALL Evolution_CZone(strs_t, LSP_t, beta_upper)
            K_1 = K
            Jv_1 = Jv
            G_1 = G
            Js_1 = Js
            CALL Evolution_SZone(strs_t, LSP_t, beta_min)
            K_2 = K
            Jv_2 = Jv
            G_2 = G
            Js_2 = Js
            K = LNR(beta_upper, beta_min, K_1, K_2, beta_t)
            Jv = LNR(beta_upper, beta_min, Jv_1, Jv_2, beta_t)
            G = LNR(beta_upper, beta_min, G_1, G_2, beta_t)
            Js = LNR(beta_upper, beta_min, Js_1, Js_2, beta_t)
            !
```

```
        ELSE
        ! lower part of transition
                CALL Evolution_CZone(strs_t, LSP_t, beta_lower)
                K_1 = K
                Jv_1 = Jv
                G_1 = G
                Js_1 = Js
                CALL Evolution_SZone(strs_t, LSP_t, beta_max)
                K_2 = K
                Jv_2 = Jv
                G_2 = G
                Js_2 = Js
                K = LNR(beta_lower, beta_max, K_1, K_2, beta_t)
                Jv = LNR(beta_lower, beta_max, Jv_1, Jv_2, beta_t)
                G = LNR(beta_lower, beta_max, G_1, G_2, beta_t)
                Js = LNR(beta_lower, beta_max, Js_1, Js_2, beta_t)
        END IF
        !
        !
        END SUBROUTINE Axisymmetry
C
C
C-----------------------------------------------------------------------
C       [ Internal Subroutine ]          Evolution_CZone
C-----------------------------------------------------------------------
        SUBROUTINE Evolution_CZone(strs_t, LSP_t, beta_t)
        ! for stiffness evolution in Compression Zone
        ! update E_0, E_s, LSP_s based on beta_t
        ! calculate K,Jv,G,Js
        ! -------------- main components -------------
        ! Ks_Beta, Jvs_Beta, Gs_Beta, LSPs_Beta, LNR
        !
        REAL*8 strs_t(NTENS), LSP_t, beta_t
        REAL*8 p_t, q_t
        REAL*8 K_0, K_s, Jv_0, Jv_s, LSP_s
        REAL*8 G_0, G_s, Js_0, Js_s
        REAL*8 sign_Js, sign_Jv_ref
        REAL*8 LSP_p, K_p, Jv_p, G_p, Js_p
                ! values at beta=0 & LSP=LSP_p
        REAL*8 K_ic, Jv_ic, G_ic, Js_ic
                ! values at beta=0 & LSP>LSP_p
        REAL*8 Js_min
                ! Js_s varies in Compression Zone, but other E_s don't
```

```
!
p_t = pp(strs_t)
q_t = qq(strs_t)
!
!
LSP_s = LSPs_beta(beta_t)
LSP_p = s_p-p_0
!
IF (LSP_t <= LSP_s) THEN
! Small Strain Zone
        K_s = Ks_Beta(beta_t)
        Jv_s = Jvs_Beta(beta_t)
        G_s = Gs_Beta(beta_t)
        Js_s = Jss_Beta(beta_t, p_t, q_t)
        !
        K_0 = r_0s*K_s
        Jv_0 = r_0s*Jv_s
        G_0 = r_0s*G_s
        Js_0 = r_0s*Js_s
        !
        K = LNR(0.D0, LSP_s, K_0, K_s, LSP_t)
        Jv = LNR(0.D0, LSP_s, Jv_0, Jv_s, LSP_t)
        G = LNR(0.D0, LSP_s, G_0, G_s, LSP_t)
        Js = LNR(0.D0, LSP_s, Js_0, Js_s, LSP_t)
                ! ( x1, x2,  y1, y2, x )
ELSE
        sign_Jv_ref = DABS(Jv_ref)/Jv_ref
        sign_Js = DABS(-q_t)/(-q_t)
                ! Js = (-3p/q)*G
        ! ---------- calculate K_ic ----------
        IF (LSP_t >= LSP_p) THEN
                ! Normal Consolidation at beta=0
                K_ic = p_t/lambda
        ELSE
                ! Transition at beta=0
                K_p = s_p/lambda
                K_s = Ks_Beta(0.D0)
                !
                K_ic = LNR(LSP_s, LSP_p, K_s, K_p, LSP_t)
        END IF
        ! ---------- calculate other E_ic ----------
        IF (beta_t < beta_upper) THEN
                Jv_ic = sign_Jv_ref*r_JvK*K_ic
```

```
                ELSE
                        Jv_ic = -sign_Jv_ref*r_JvK*K_ic
                END IF
                G_ic = 1.5D0*(1-k_nc)/(1+2.D0*k_nc)*K_ic
                Js_ic = sign_Js*r_JsK*K_ic
                !
                ! E at the boundaries equal to E_s because LSP_f is infinite
                IF (beta_t < beta_upper) THEN
                        !
                        K_s = Ks_Beta(beta_upper)
                        Jv_s = Jvs_Beta(beta_upper)
                        G_s = Gs_Beta(beta_upper)
                        Js_s = Jss_Beta(beta_upper, p_t, q_t)
                        !
                        K = LNR(0.D0, beta_upper, K_ic, K_s, beta_t)
                        Jv = LNR(0.D0, beta_upper, Jv_ic, Jv_s, beta_t)
                        G = LNR(0.D0, beta_upper, G_ic, G_s, beta_t)
                        Js = LNR(0.D0, beta_upper, Js_ic, Js_s, beta_t)
                ELSE
                        K_s = Ks_Beta(beta_lower)
                        Jv_s = Jvs_Beta(beta_lower)
                        G_s = Gs_Beta(beta_lower)
                        Js_s = Jss_Beta(beta_lower, p_t, q_t)
                        !
                        K = LNR(2.D0*pi, beta_lower, K_ic, K_s, beta_t)
                        Jv = LNR(2.D0*pi, beta_lower, Jv_ic, Jv_s, beta_t)
                        G = LNR(2.D0*pi, beta_lower, G_ic, G_s, beta_t)
                        Js = LNR(2.D0*pi, beta_lower, Js_ic, Js_s, beta_t)
                END IF
        END IF
        !
        !
        END SUBROUTINE Evolution_CZone
C
C
C-----------------------------------------------------------------------
C       [ Internal Subroutine ]          Evolution_SZone
C-----------------------------------------------------------------------
        SUBROUTINE Evolution_SZone(strs_t, LSP_t, beta_t)
        ! for stiffness evolution in Shear Zone
        ! update E_0, E_s, LSP_s based on beta_t
        ! calculate K,Jv,G,Js
        ! -------------- main components -------------
```

```
! Ks_Beta, Jvs_Beta, Gs_Beta, LSPs_Beta,
! LNR, LSP_tf, NONLNR
!
REAL*8 strs_t(NTENS)
REAL*8 LSP_t, beta_t
REAL*8 p_t, q_t, LSP_f
REAL*8 G_0, G_s, K_0, K_s, Jv_0, Jv_s, LSP_s
REAL*8 mu, Jv_f
INTEGER flag_upward
                !       1: Dstrs leads to the upper F.S.
                !       0: Dstrs leads to the lower F.S.
!
K_s = Ks_Beta(beta_t)
Jv_s = Jvs_Beta(beta_t)
G_s = Gs_Beta(beta_t)
LSP_s = LSPs_beta(beta_t)
!
K_0 = r_0s*K_s
Jv_0 = r_0s*Jv_s
G_0 = r_0s*G_s
!
p_t = pp(strs_t)
q_t = qq(strs_t)
!
IF (LSP_t <= LSP_s) THEN
! Small Strain Zone
        K = LNR(0.D0, LSP_s, K_0, K_S, LSP_t)
        G = LNR(0.D0, LSP_s, G_0, G_S, LSP_t)
        Jv = LNR(0.D0, LSP_s, Jv_0, Jv_S, LSP_t)
                ! ( x1, x2,  y1, y2, x )
ELSE IF (p_t <= s_p ) THEN
! Large Strain
        LSP_f = LSP_t + LSP_tf(strs_t, beta_t, flag_upward)
        !
        IF (flag_upward == 1) THEN
                Jv_f = Jvf_up
                mu = mu_up
        ELSE
                Jv_f = Jvf_low
                mu = r_mu*mu_up
        END IF
        !
        K = NONLNR(LSP_s, LSP_f, K_s, K_f, LSP_t, mu)
```

```
            Jv = NONLNR(LSP_s, LSP_f, Jv_s, Jv_f, LSP_t, mu)
            G = NONLNR(LSP_s, LSP_f, G_s, G_f, LSP_t, mu)
                  ! (  x1, x2, y1, y2, x, mid)
      END IF
      !
      IF (DABS(q_t) < TOL_ZERO) THEN
            Js = -3.D0*G*p_t/TOL_ZERO*(q_t/DABS(q_t))
                  ! avoid p_t = 0
      ELSE
            Js = -3.D0*G*p_t/q_t
      END IF
      !
      !
      END SUBROUTINE Evolution_SZone
C
C -----------------------------------------------------------------
C       IN-SUB                  Cross_Anisotropy
C -----------------------------------------------------------------
      SUBROUTINE Cross_Anisotropy
      ! calculate stiff(*,*) using 'K, Jv, G, Js'
      ! a mapping from axisymmetric condition to general conditions
      ! ----------------------------------------
      REAL*8 A, B, C, D, E    ! (/A,B,C,B,A,C,D,D,E/)
      REAL*8 deno1, deno2     ! 2 denominators in stiff matrix
      !
      !-------- Compliance matrix
      A = (-1.D0/(Js*3.D0)+1.D0/(K*4.5D0)+1.D0/(G*6.D0)-1.D0/(Jv*3.D0))
      1                 *1.25D0
      B = -A*0.2D0
      C = 1.D0/(Jv*3.D0)-1.D0/(Js*6.D0)+1.D0/(K*9.D0)-1.D0/(G*6.D0)
      D = 1.D0/(K*9.D0)-1.D0/(G*6.D0)+1.D0/(Js*3.D0)-1.D0/(Jv*6.D0)
      E = 1.D0/(G*3.D0)+1.D0/(K*9.D0)+1.D0/(Js*3.D0)+1.D0/(Jv*3.D0)
      !-------- Stiffness matrix
      deno1 = -A*A*E+2.D0*A*C*D+B*B*E-2.D0*B*C*D
      deno2 = -A*E+2.D0*C*D-B*E
      !
      stiff(1:NTENS,1:NTENS) = 0.D0
      !
      SELECT CASE (NTENS)
      CASE (4)        ! dimension-2 is vertical
            stiff(1,1) = (-A*E+C*D)/deno1
            stiff(1,3) = (B*E-C*D)/deno1
            stiff(1,2) = C/deno2
```

```
                stiff(3,1) = stiff(1,3)
                stiff(3,3) = stiff(1,1)
                stiff(3,2) = stiff(1,2)
                stiff(2,1) = D/deno2
                stiff(2,3) = stiff(2,1)
                stiff(2,2) = -(A+B)/deno2
                stiff(4,4) = G
        CASE (6)        ! dimension-3 is vertical
                stiff(1,1) = (-A*E+C*D)/deno1
                stiff(1,2) = (B*E-C*D)/deno1
                stiff(1,3) = C/deno2
                stiff(2,1) = stiff(1,2)
                stiff(2,2) = stiff(1,1)
                stiff(2,3) = stiff(1,3)
                stiff(3,1) = D/deno2
                stiff(3,2) = stiff(3,1)
                stiff(3,3) = -(A+B)/deno2
                stiff(4,4) = G
                stiff(5,5) = G
                stiff(6,6) = G
        END SELECT
        !
        END SUBROUTINE Cross_Anisotropy
C
C---------------------------------------------------------------------
C               [ Internal Subroutine ]         Dstrs_Dstrn
C---------------------------------------------------------------------
        SUBROUTINE Dstrs_Dstrn(Dstrs, Dstrn)
        ! get Dstrs using stiff(*,*)
        ! ----------------------------------------
        REAL*8 Dstrs(NTENS), Dstrn(NTENS)
        !
        Dstrs(1:NTENS) = 0.D0
        DO i=1,NTENS
                DO j=1,NTENS
                        Dstrs(i) = Dstrs(i)+stiff(i,j)*Dstrn(j)
                END DO
        END DO
        !
        END SUBROUTINE Dstrs_Dstrn
C
C ---------------------------------------------------------------------
C               [ Internal Subroutine ]         READ_PROPS
```

```
C -----------------------------------------------------------------
      SUBROUTINE READ_PROPS       !!!
      ! ------------------------------------------------------
            G_ref =  PROPS(1)
            Jv_ref =  PROPS(2)
            K_ref =  PROPS(3)
            LSP_min = PROPS(4)
            mu_up =  PROPS(5)
            nn =  PROPS(6)
            r_0s = PROPS(7)
            r_x = PROPS(8)
            r_LSP = PROPS(9)
            r_mu =PROPS(10)
            faf = PROPS(11)
            lambda = PROPS(12)
            OCR =PROPS(13)
            k_nc =  PROPS(14)
      !
      END SUBROUTINE READ_PROPS
C
C
C -----------------------------------------------------------------
C           [ Internal Subroutine ]          P_Dependency
C -----------------------------------------------------------------
      SUBROUTINE P_Dependency
      ! calculate internal stiffness parameters based on p_0
      ! E_min = E_ref*p_a*(p_0/p_a)^nn
      ! LSP_min = (LSP_ref*p_a)*(p_0/p_a)
      ! ---------------------------------
      !
      REAL*8 r_t, r_E, sign_Jv_ref
      REAL*8 p_0t
      !
      IF (p0_pre == s_p .AND. p_0 < p0_pre) THEN
            p_0t = p0_pre
      ELSE
            p_0t = p_0
      END IF
      !
      r_t = p_0t/p_a
      r_E = p_a*r_t**nn
      sign_Jv_ref = DABS(Jv_ref)/Jv_ref
      !
```

```
      K_min = K_ref*r_E
            !K_min falls in Compression Zone
      Jv_min = Jv_ref*r_E
            !Beta_Jvmin = 0.5pi
      G_min = G_ref*r_E
            !G_min falls in Compression Zone
      !
      K_max = r_x*K_min
            !Beta_Kmax = pi; K_max falls in Shear Zone
      Jv_max = (-sign_Jv_ref)*r_JvK*K_min
            !Jv_max at 1.5pi
      G_max = r_x*G_min
            !Beta_Gmax = 1.5pi
      !
      LSP_min = p_0*LSP_ref
            !Beta_LSPmin = Beta_RTC
      LSP_max = r_LSP*LSP_min
            !LSPmax falls in [Beta_RTE, 2pi]
      !
      END SUBROUTINE P_Dependency
C
C
C ---------------------------------------------------------------------
C           [ Internal Subroutine ]          INI_STATEV
C ---------------------------------------------------------------------
      SUBROUTINE INI_STATEV
      ! strs_t with general mechanics sign convention (compressive: +)
      ! ------------------------------------
      REAL*8 strs_t(NTENS)
      !
      STATEV(1) = 0.5D0*pi       ! Beta_pre
      STATEV(2) = 0.D0           ! LSP_pre
      !
      CALL Sign_Change(strs_t, STRESS)
      Ue = 0.D0
      CALL Eff_Stress(strs_t)
      !
      STATEV(3) = pp(strs_t)             ! p_0
      STATEV(4) = pp(strs_t)*OCR         ! s_p
      STATEV(5) = Ue                              ! Ue
      !
      ! STATEV(6-9) for K, Jv, G, Js, for debug purpose
      !
```

```
      STATEV(10) = STATEV(3)          ! p0_pre
      !
      END SUBROUTINE INI_STATEV
C
C -------------------------------------------------------------------
C               [ Internal Subroutine ]         READ_STATEV
C -------------------------------------------------------------------
      SUBROUTINE READ_STATEV
      ! ----------------------------------------------------------
      Beta_pre = STATEV(1)
      LSP_pre = STATEV(2)
      !
      p_0 = STATEV(3)
      s_p = STATEV(4)
      Ue = STATEV(5)
      !
      ! state variables K, Jv, G, Js are only for debug purpose
      !
      p0_pre = STATEV(10)
      !
      END SUBROUTINE READ_STATEV
C
C -------------------------------------------------------------------
C               [ Internal Subroutine ]         WRITE_STATEV
C -------------------------------------------------------------------
      SUBROUTINE WRITE_STATEV
      ! ----------------------------------------------------------
      STATEV(1) = Beta_pre
      STATEV(2) = LSP_pre
      !
      STATEV(3) = p_0
      STATEV(4) = s_p
      STATEV(5) = Ue
      !
      STATEV(6) = K
      STATEV(7) = Jv
      STATEV(8) = G
      STATEV(9) = Js
      !
      STATEV(10) = p0_pre
      !
      END SUBROUTINE WRITE_STATEV
C
```

```
C-----------------------------------------------------------------------
C                [ Internal Subroutine ]          Sign_Change()
C-----------------------------------------------------------------------
        SUBROUTINE Sign_Change(strs1, strs2)
        ! switch between General Mechanics & GeoMechanics sign conventions
        ! from strs2 to strs1
        ! ------------------------------------
        REAL*8 strs1(NTENS), strs2(NTENS)
        !
        DO i = 1, NDI
                strs1(i) = -strs2(i)
        END DO
        !
        DO i = NDI+1, NTENS
                strs1(i) = strs2(i)
        END DO
        !
        END SUBROUTINE Sign_Change
C
C ----------------------------------------------------------------------
C                           [ Internal Subroutine ]    OUT_FS
C ----------------------------------------------------------------------
        SUBROUTINE OUT_FS(Dstrn_t, flag_fail, strs_t, beta_t)
        ! update strs_t, beta_t & stiff using failure parameters
        ! ----------------------------------------
        REAL*8 Dstrn_t(NTENS), strs_t(NTENS), Dstrs_t(NTENS)
        INTEGER flag_fail
        REAL*8 beta_t, p_t, q_t
        !
        p_t = pp(strs_t)
        q_t = qq(strs_t)
        !
        K = K_f
        IF (flag_fail == 1) THEN
                Jv = Jvf_up
        ELSE
                Jv = Jvf_low
        END IF
        G = G_f
        Js = -3.D0*G*p_t/q_t
        !
        CALL Cross_Anisotropy
                ! calculate stiff(*,*) using K, Jv, G, Js
```

```
        CALL Dstrs_Dstrn(Dstrs_t, Dstrn_t)
                ! calculate Dstrs using stiff(*,*) & Dstrn
        beta_t = FN_Beta(Dstrs_t)
        !
        strs_t = strs_t + Dstrs_t
        !
        END SUBROUTINE OUT_FS
C
C -----------------------------------------------------------------
C                       [ Internal Subroutine ]     Eff_Stress
C -----------------------------------------------------------------
        SUBROUTINE Eff_Stress(strs_t)
        ! strs in geotechnical sign convention
        ! ------------------------------------------------------
        REAL*8 strs_t(NTENS)
        !
        SELECT CASE (NTENS)
        CASE (4)
                ! dimension-2 is vertical
                IF (WT > COORDS(2)) THEN
                        Uh = 9.8D0*(WT - COORDS(2))
                ELSE
                        Uh = 0.D0
                END IF
        !
        CASE (6)
                ! dimension-2 is vertical
                IF (WT > COORDS(3)) THEN
                        Uh = 9.8D0*(WT - COORDS(3))
                ELSE
                        Uh = 0.D0
                END IF
        !
        END SELECT
        !
        IF (flag_drain == 0) THEN
                ! undrained
                DO i=1,NDI
                        strs_t(i) = strs_t(i) - Ue - Uh
                END DO
        ELSE
                ! drained
                DO i=1,NDI
```

```
                        strs_t(i) = strs_t(i) - Uh
                END DO
        END IF
        !
        END SUBROUTINE Eff_Stress
C
C -------------------------------------------------------------------
C                       [ Internal Subroutine ]     Final_Update
C -------------------------------------------------------------------
        SUBROUTINE Final_Update(strs_t, Dstrn_t)
        ! Update Ue, STRESS & DDSDDE
        ! ----------------------------------
        REAL*8 strs_t(NTENS), Dstrn_t(NTENS)
                ! both in geotechnical sign convention, i.e. compressive=(+)
        !
        IF (flag_drain == 0) THEN
                ! undrained
                Ue = Ue + K_fld*3.D0*pp(Dstrn_t)
                DO i=1,NDI
                        strs_t(i) = strs_t(i) + Ue + Uh
                        !
                        DO j=1,NDI
                                stiff(i,j) = stiff(i,j) + K_fld
                        END DO
                        !
                END DO
        ELSE
                ! drained
                DO i=1,NDI
                        strs_t(i) = strs_t(i) + Uh
                END DO
        END IF
        !
        CALL Sign_Change(STRESS, strs_t)
                ! update in general mechanics sign convention
        DDSDDE = stiff
        !
        !
        END SUBROUTINE Final_Update
C
C
C-------------------------------------------------------------------
C               [ Internal Subroutine ]          ISO_ELA
```

```
C-----------------------------------------------------------------------
      SUBROUTINE ISO_ELA(E, nu)
      ! update DDSDDE & STRESS
      ! ------------------------------------
      REAL*8 E, nu
      INTEGER k1, k2
      REAL*8 GG, c1
      !
      ! Matrix - Drained
      DO k1=1,NTENS
    DO k2=1,NTENS
      DDSDDE(k2,k1) = 0.
    END DO
  END DO
      !
  c1 = (1.D0-2.D0*nu)*(1.D0+nu)
  DO k1=1,NDI
    DO k2 = 1,NDI
        DDSDDE(k1,k2) = nu*E/c1
    END DO
  END DO
      !
  DO k1=1,NDI
    DDSDDE(k1,k1) = (1.D0-nu)*E/c1
  END DO
      !
  GG = 0.5D0*E/(1.D0+nu)
  DO k2=NDI+1, NTENS
    DDSDDE(k2,k2) = GG
  END DO
      !
      DO k1=1,NTENS
            DO k2=1,NTENS
                  STRESS(k1) = STRESS(k1)+DDSDDE(k1,k2)*DSTRAN(k2)
            END DO
      END DO
      !
      !
      END SUBROUTINE ISO_ELA
C
C -------------------------------------------------------------------
C           [ Internal Subroutine ]       DEBUG_GENERAL
C -------------------------------------------------------------------
```

```
      SUBROUTINE DEBUG_GENERAL
      !
      ! ----------------------------------
      WRITE (16,'("------------------ Numerical --------------------")')
      WRITE (16,'("TOL_r = ", F10.9)') TOL_r
      WRITE (16,'("TOL_Beta = ", F10.9)') TOL_Beta
      WRITE (16,'("K_f = ", F5.0)') K_f
      WRITE (16,'("G_f = ", F5.0)') G_f
      WRITE (16,'("Jvf_up = ", F8.0)') Jvf_up
      WRITE (16,'("Jvf_low = ", F8.0)') Jvf_low
      WRITE (16,'("r_JvK = ", F5.0)') r_JvK
      WRITE (16,'("r_JsK = ", F5.0)') r_JsK
      WRITE (16,'("K_fld = ", F8.0)') K_fld
      WRITE (16,'("------------------ Material --------------------")')
      WRITE (16,'("K_ref = ", F5.0)') K_ref
      WRITE (16,'("G_ref = ", F5.0)') G_ref
      WRITE (16,'("Jv_ref = ", F5.0)') Jv_ref
      WRITE (16,'("LSP_ref = ", F5.0)') LSP_ref
      WRITE (16,'("mu_up = ", F5.2)') mu_up
      WRITE (16,'("r_0s = ", F5.2)') r_0s
      WRITE (16,'("r_x = ", F5.2)') r_x
      WRITE (16,'("r_LSP = ", F5.2)') r_LSP
      WRITE (16,'("r_mu = ", F5.2)') r_mu
      WRITE (16,'("nn = ", F5.3)') nn
      WRITE (16,'("faf = ", F6.2)') faf
      WRITE (16,'("OCR = ", F5.3)') OCR
      WRITE (16,'("lambda = ", F8.6)') lambda
      WRITE (16,'("k_nc = ", F5.3)') k_nc
      WRITE (16,'("------------------ Others --------------------")')
      WRITE (16,'("WT = ", F6.3)') WT
      WRITE (16,'("flag_drain = ", I3)') flag_drain
      WRITE (16,'()')
      WRITE (16,'("------------------ Computation -----------------")')
      WRITE (16,'("NOEL,KSTEP,KINC, KSUB,  p,     q,   strs(1), strs(3),
      1  ev,      es,        K,       Jv,      G,      Js,
     2     Beta,   LSP,    Uh,    Ue")')
      WRITE (16,'("-----------------------------------------------")')
      !
      !
      END SUBROUTINE DEBUG_GENERAL
C
C -----------------------------------------------------------------
C                          [ Internal Subroutine ]          DEBUG_SUBSTEP
```

```
C -----------------------------------------------------------------
      SUBROUTINE DEBUG_SUBSTEP(KSUB)
      !
      ! ------------------------------------
      !
      INTEGER KSUB
      REAL*8 strs_t(NTENS)
      REAL*8 p_t, q_t, ev, es
      !
      CALL Sign_Change(strs_t, STRESS)
      p_t = pp(strs_t)
      q_t = qq(strs_t)
      !
      ev = -300.D0*pp(STRAN)
      es = -100.D0*qq(STRAN)
      !
      WRITE (16,'(4I5, 4F8.2, 2F10.7, F8.0, F12.0, F8.0, F12.0, 4F8.2)')
     1 NOEL, KSTEP, KINC, KSUB, p_t, q_t, strs_t(1), strs_t(3),
     2 ev, es, K, Jv, G, Js, Beta_pre, LSP_pre, Uh, Ue
      WRITE (16,'(4F12.8)') DSTRAN(1:4)
      !
      !
      END SUBROUTINE DEBUG_SUBSTEP
C
C -----------------------------------------------------------------
C                        [ Internal Subroutine ]          DEBUG_Beta
C -----------------------------------------------------------------
      SUBROUTINE DEBUG_Beta
      ! record computational info in Beta_Converge
      ! ------------------------------------
      !
      !WRITE (16,'("-------------------------------------------------")')
      !
      END SUBROUTINE DEBUG_Beta
C
C
C                                                SUBROUTINES
C ********************************************************************
C                                                 FUNCTIONS
C
C
C -----------------------------------------------------------------
C                [ Internal Function ]             LSP_tf
```

```
C ---------------------------------------------------------------------
      REAL*8 FUNCTION LSP_tf(strs_t, beta_t, flag_upward)
      ! evaluate distance in LSP from strs_t to its image point on F.S.
      ! return flag_upward for determining mu
      ! ---------------------------------
      REAL*8 strs_t(NTENS)
      REAL*8 beta_t
      INTEGER flag_upward
            ! 1: path direction is flag_upward; 0: downward
      REAL*8 Beta_crit
            ! boundary for Mf_upper and Mf_lower
      REAL*8 Dp, Dq, p, q
      REAL*8 D_LSP, mm
            ! mm = LSP_tf/D_LSP
      !
      Dp = DCOS(beta_t)
      Dq = DSIN(beta_t)
      D_LSP = DSQRT(Dp*Dp + Dq*Dq)
            ! Magnitudes of D_LSP does not matter here
      p = pp(strs_t)
      q = qq(strs_t)
      !
      !
      Beta_crit = FN_Beta(strs_t) + pi
      CALL Beta_Period(Beta_crit)
            !
      !
      IF (beta_t < Beta_crit .AND. beta_t > Beta_upper) THEN
            ! FS(p+mm*Dp, q+mm*Dq) = 0
            mm = (-Mf_upper*p+q)/(Mf_upper*Dp-Dq)
            flag_upward = 1
      ELSE
            mm = (Mf_lower*p+q)/(-Mf_lower*Dp-Dq)
            ! note: Mf_lower is positive
            flag_upward = 0
      END IF
      !
      LSP_tf = mm*D_LSP
      !
      !
      END FUNCTION LSP_tf
C
C---------------------------------------------------------------------
```

```
C                       [ Internal Function ]           Failure
C-----------------------------------------------------------------
      INTEGER FUNCTION Failure(strs_t)
      ! evaluate failure state
      ! 1: out of upper F.S.; 2: out of lower F.S.; 0: allowable
      ! strs in geotechnical sign convention
      ! ------------------------------------------------------
      REAL*8 strs_t(NTENS)
      REAL*8 p_t, q_t
      !
      p_t = pp(strs_t)
      q_t = qq(strs_t)
      !
      IF(Mf_upper*p_t < q_t) THEN
            Failure = 1
      ELSE IF(q_t < -Mf_lower*p_t) THEN
                  ! Note: Mf_lower is positive
            Failure = 2
      ELSE
            Failure = 0
      END IF
      !
      !
      END FUNCTION Failure
C
C-----------------------------------------------------------------
C                       [ Internal Function ]           FS_MN
C-----------------------------------------------------------------
      REAL*8 FUNCTION FS_MN(strs)
      ! Matsuoka-Nakai failure surface
      ! NOTE: M-N function represents multiple surfaces,
      !       only one of which is intended for Failure Surface.
      ! Extra numerical measures must be taken if one has to use M-N for F.S.
      ! ---------------------------------------
      REAL*8 strs(NTENS)
      REAL*8 a, I_1, I_2, I_3
      !
      a = faf*pi/180.D0
      !
      I_1 = strs(1)+strs(2)+strs(3)
      !
      SELECT CASE (NTENS)
      CASE (4)
```

```
            I_2 = strs(4)*strs(4)-strs(1)*strs(2)
     1                      -strs(2)*strs(3)-strs(3)*strs(1)
            ! (s1,s3:horizontal; s2:vertical; s4:12)
            I_3 = strs(1)*strs(2)*strs(3)-strs(3)*strs(4)*strs(4)
     !
     CASE (6)
            I_2 = strs(4)*strs(4)+strs(5)*strs(5)+strs(6)*strs(6)
     1              -strs(1)*strs(2)-strs(2)*strs(3)-strs(3)*strs(1)
            ! (s1,s2:horizontal; s3:vertical; s4:12; s5:13; s6:23)
            I_3 = strs(1)*strs(2)*strs(3)+2*strs(4)*strs(5)*strs(6)
     1              -strs(1)*strs(6)*strs(6)-strs(2)*strs(5)*strs(5)
   2        -strs(3)*strs(4)*strs(4)
     END SELECT
     !
     FS_MN = DCOS(a)*DCOS(a)/(9-DSIN(a)*DSIN(a))
     FS_MN = I_3 + FS_MN*I_1*I_2
     !
     !
     END FUNCTION FS_MN
C
C---------------------------------------------------------------------
C                       [ Internal Function ]          FN_eta
C---------------------------------------------------------------------
     REAL*8 FUNCTION FN_eta(strs_t)
     ! for Matsuoka-Nakai Surface
     ! --------------------------------
     REAL*8 strs_t(NTENS)
     REAL*8 a, I_1, I_2, I_3
     !
     I_1 = strs_t(1)+strs_t(2)+strs_t(3)
     !
     SELECT CASE (NTENS)
     CASE (4)
            I_2 = strs_t(4)*strs_t(4)-strs_t(1)*strs_t(2)
     1                      -strs_t(2)*strs_t(3)-strs_t(3)*strs_t(1)
            ! (s1,s3:horizontal; s2:vertical; s4:12)
     I_3 = strs_t(1)*strs_t(2)*strs_t(3)-strs_t(3)*strs_t(4)*strs_t(4)
     !
     CASE (6)
     I_2 = strs_t(4)*strs_t(4)+strs_t(5)*strs_t(5)+strs_t(6)*strs_t(6)
     1      -strs_t(1)*strs_t(2)-strs_t(2)*strs_t(3)-strs_t(3)*strs_t(1)
            ! (s1,s2:horizontal; s3:vertical; s4:12; s5:13; s6:23)
     I_3 = strs_t(1)*strs_t(2)*strs_t(3)+2*strs_t(4)*strs_t(5)*strs_t(6)
```

```
        1       -strs_t(1)*strs_t(6)*strs_t(6)-strs_t(2)*strs_t(5)*strs_t(5)
   2            -strs_t(3)*strs_t(4)*strs_t(4)
        END SELECT
        !
        a = -I_3/(I_1*I_2)
        a = (1.D0-9.D0*a)/(1.D0-a)
        FN_eta = DSQRT(a)
        !
        END FUNCTION FN_eta
C
C
C-----------------------------------------------------------------------
C                       [ Internal Function ]           Norm
C-----------------------------------------------------------------------
        REAL*8 FUNCTION Norm(strs_t)
        ! Euclidean norm
        ! --------------------------
        REAL*8 strs_t(NTENS), nm
        !
        nm = 0.D0
        DO i=1,NTENS
                nm = nm + strs_t(i)*strs_t(i)
        END DO
        Norm = DSQRT(nm)
        !
        END FUNCTION Norm
C
C-----------------------------------------------------------------------
C                       [ Internal Function ]           Dot
C-----------------------------------------------------------------------
        REAL*8 FUNCTION Dot(Dstrs1, Dstrs2)
        ! evaluate the difference of the two path directions
        ! Dot=0: perpendicular; Dot>0: "small"; Dot<0: "large"
        ! ---------------------------------
        REAL*8 Dstrs1(NTENS), Dstrs2(NTENS), dt
        !
        dt = 0.D0
        DO i=1,NTENS
                dt = dt + Dstrs1(i)*Dstrs2(i)
        END DO
        Dot = dt/(Norm(Dstrs1)*Norm(Dstrs2))
        !
        END FUNCTION Dot
```

```
C
C-----------------------------------------------------------------
C                       [ Internal Function ]           pp
C-----------------------------------------------------------------
      REAL*8 FUNCTION pp(strs_t)
      ! ---------------------------------------------------------
      REAL*8 strs_t(NTENS)
      !
      SELECT CASE (NTENS)
      CASE (4)
      ! dimension-2 is vertical, dimension-1 is in-plane horizontal
            pp = (strs_t(1)+strs_t(2))*0.5D0
            ! consistent with simplified definition for q
            !pp = (strs_t(1)+strs_t(2)+strs_t(3))/3.D0
      CASE (6)
            pp = (strs_t(1)+strs_t(2)+strs_t(3))/3.D0
      END SELECT
      !
      END FUNCTION pp
C
C-----------------------------------------------------------------
C                       [ Internal Function ]           qq
C-----------------------------------------------------------------
      REAL*8 FUNCTION qq(strs_t)
      ! for simplicity, q = sigma_v - sigma_h
      ! note: if one uses general definition for q,
      !       measures must be taken to distinguish different quadrants
      !       in normal stress space (sigma1, sigma2, sigma3).
      ! ---------------------------------------------------------
      !
      REAL*8 strs_t(NTENS)
      !
      SELECT CASE (NTENS)
      CASE (4)
      ! dimension-2 is vertical
            qq = strs_t(2)-strs_t(1)
      !     qq = DSQRT(((strs_t(1)-strs_t(2))*(strs_t(1)-strs_t(2))+
      1!          (strs_t(2)-strs_t(3))*(strs_t(2)-strs_t(3))+
      2!          (strs_t(3)-strs_t(1))*(strs_t(3)-strs_t(1)))*0.5D0+
   3!       3.D0*strs_t(4)*strs_t(4))
      CASE (6)
      ! dimension-3 is vertical
            qq = strs_t(3)-strs_t(1)
```

```
      !         qq = DSQRT(((strs_t(1)-strs_t(2))*(strs_t(1)-strs_t(2))+
      1!                   (strs_t(2)-strs_t(3))*(strs_t(2)-strs_t(3))+
      2!                   (strs_t(3)-strs_t(1))*(strs_t(3)-strs_t(1)))*0.5D0+
     3!  3.D0*(strs_t(4)*strs_t(4)+strs_t(5)*strs_t(5)+strs_t(6)*strs_t(6)))
      !
      END SELECT
      !
      END FUNCTION qq
C
C-----------------------------------------------------------------
C     [ Internal Function ]          NONLNR (Nonlinear Degradation)
C-----------------------------------------------------------------
      REAL*8 FUNCTION NONLNR(x1, x2, y1, y2, xx, mu)
      ! eqn1: a/(x1+b)+c = y1
      ! eqn2: a/(x2+b)+c = y2
      ! eqn3: a/(1/2*x1+1/2*x2+b)+c = (y1-y2)/mu+y2
      ! eqn4: a/(xx+b)+c = y
      ! calculate y
      ! ------------------------------------------------------
      REAL*8 x1, x2, y1, y2, xx, mu
      REAL*8 nume, deno
      !
      nume = -y2*x1 + y2*mu*x1 - y1*x2 + xx*y1 - xx*y2*mu + xx*y2
      deno = 2.D0*xx - xx*mu - x1 - x2 + mu*x1
      NONLNR = nume/deno
      !
      END FUNCTION NONLNR
C
C-----------------------------------------------------------------
C     [ Internal Function ]          LNR (Linear Interpolation)
C-----------------------------------------------------------------
      REAL*8 FUNCTION LNR(x1, x2, y1, y2, xx)
      ! eqn1: a*x1+b=y1
      ! eqn2: a*x2+b=y2
      ! eqn3: a*xx+b=y
      ! calculate y
      ! ------------------------------------------------------
      REAL*8 x1, x2, y1, y2, xx
      !
      LNR = (xx*y1-xx*y2+x1*y2-x2*y1)/(-x2+x1)
      !
      END FUNCTION LNR
C
```

```
C
C-----------------------------------------------------------------
C       [ Internal Function ]           FN_Beta
C-----------------------------------------------------------------
        REAL*8 FUNCTION FN_Beta(Dstrs)
        ! ---------------------------------------------------
        ! Beta: [0, 2*pi)
        !
        REAL*8 Dstrs(NTENS)
        REAL*8 Dp, Dq
        !
        Dp = pp(Dstrs)
        Dq = qq(Dstrs)
        !
        !
        IF (Dp > 0.D0 .AND. Dq > 0.D0) THEN             !quadrant 1
                FN_Beta = DATAN(Dq/Dp)
                !
        ELSE IF(Dp < 0.D0 .AND. Dq > 0.D0) THEN         !quadrant 2
                FN_Beta = DATAN(Dq/Dp) + pi
                !
        ELSE IF(Dp < 0.D0 .AND. Dq < 0.D0) THEN         !quadrant 3
                FN_Beta = DATAN(Dq/Dp) + pi
                !
        ELSE IF(Dp > 0.D0 .AND. Dq < 0.D0) THEN         !quadrant 4
                FN_Beta = DATAN(Dq/Dp) + 2.D0*pi
                !
        ELSE IF(Dp > 0.D0 .AND. Dq == 0.D0) THEN !p-axis (+)
                FN_Beta = 0.D0
                !
        ELSE IF(Dp < 0.D0 .AND. Dq == 0.D0) THEN !p-axis (-)
                FN_Beta = pi
                !
        ELSE IF(Dp == 0.D0 .AND. Dq > 0.D0) THEN !q-axis (+)
                FN_Beta = 0.5D0*pi
                !
        ELSE IF(Dp == 0.D0 .AND. Dq < 0.D0) THEN !q-axis (-)
                FN_Beta = 1.5D0*pi
        ELSE                                                    ! Dp=Dq=0
                !!! exception
        END IF
        !
        END FUNCTION FN_Beta
```

```
C
C-----------------------------------------------------------------
C               [ Internal Function ]           Ks_beta
C-----------------------------------------------------------------
        REAL*8 FUNCTION Ks_beta(beta_t)
        ! For K_s
        ! Ks_beta(beta_upper) = Ks_beta(beta_lower) = K_min;
        ! Ks_beta(pi) = K_max
        ! beta: [0, 2*pi)
        ! ----------------------------------------
        REAL*8 beta_t
        !
        IF(beta_t >= Beta_lower .OR. beta_t <= Beta_upper) THEN
                ! compression zone
                Ks_beta = K_min
        ELSE IF(beta_t > Beta_upper .AND. beta_t <= pi) THEN
                ! shear zone - 1
                Ks_beta = LNR(Beta_upper, pi, K_min, K_max, Beta_t)
                ! LNR(x1, x2, y1, y2, x)
        ELSE
                ! shear zone - 2
                Ks_beta = LNR(Beta_lower, pi, K_min, K_max, Beta_t)
        END IF
        !
        !
        END FUNCTION Ks_beta
        !
C
C
C-----------------------------------------------------------------
C               [ Internal Function ]           Jvs_beta
C-----------------------------------------------------------------
        REAL*8 FUNCTION Jvs_beta(beta_t)
        ! For Jv_s
        ! -Jvs_beta(beta_upper) = Jvs_beta(beta_lower) = Jv_max;
        ! Jvs_beta(0.5*pi) = Jv_min
        ! beta: [0, 2*pi)
        ! ----------------------------------------
        REAL*8 beta_t
        !
        IF(beta_t <= Beta_upper) THEN
                ! upper compression zone
                Jvs_beta = -Jv_max
```

```
                ! Jv_max refering to beta=1.5pi
        ELSE IF(beta_t >= Beta_lower) THEN
                ! lower compression zone
                Jvs_beta = Jv_max
                        ! sign of Jv changes with Dq
        ELSE IF(beta_t > Beta_upper .AND. beta_t <= 0.5D0*pi) THEN
                ! shear zone - 1 (upper)
                Jvs_beta = LNR(Beta_upper, 0.5D0*pi, -Jv_max, Jv_min, Beta_t)
                                        ! LNR(x1, x2, y1, y2, x)
        ELSE IF(beta_t > 0.5D0*pi .AND. beta_t <= pi) THEN
                ! shear zone - 2 (upper)
                Jvs_beta = LNR(0.5D0*pi, Beta_lower, Jv_min, -Jv_max, Beta_t)
        ELSE
                ! shear zone - 3 (lower)
                Jvs_beta = LNR(0.5D0*pi, Beta_lower, -Jv_min, Jv_max, Beta_t)
                        ! sign of Jv changes with Dq
        END IF
        !
        !
        END FUNCTION Jvs_beta
        !
        !
C
C---------------------------------------------------------------------
C               [ Internal Function ]           Gs_beta
C---------------------------------------------------------------------
        REAL*8 FUNCTION Gs_beta(beta_t)
        ! For G_s
        ! Gs_beta(beta_upper)=Gs_beta(beta_lower)=G_min;
        ! Gs_beta(1.5*pi)=G_max
        ! beta: [0, 2*pi)
        ! ----------------------------------------
        REAL*8 beta_t
        REAL*8 Gs_0         ! Gs value at beta=0
        !
        Gs_0 = 1.5D0*(1-k_nc)/(1+2.D0*k_nc)*K_min
        !
        IF(beta_t <= Beta_upper) THEN
                ! compression zone - upper
                Gs_beta = LNR(Beta_upper, 0.D0, G_min, Gs_0, Beta_t)
                ! LNR(x1, x2, y1, y2, x)
        !ELSE IF(beta_t >= Beta_lower) THEN
                ! compression zone - lower
```

```
!       Gs_beta = LNR(Beta_lower, 2.D0*pi, G_min, Gs_0, Beta_t)
        !
ELSE IF(beta_t > Beta_upper .AND. beta_t <= 1.5D0*pi) THEN
        ! shear zone - 1
        Gs_beta = LNR(Beta_upper, 1.5D0*pi, G_min, G_max, Beta_t)
        !
ELSE
        ! shear zone - 2
        Gs_beta = LNR(2.D0*pi, 1.5D0*pi, Gs_0, G_max, Beta_t)
!       Gs_beta = LNR(Beta_lower, 1.5D0*pi, G_min, G_max, Beta_t)
END IF
!
!
END FUNCTION Gs_beta
!
C
C-----------------------------------------------------------------
C               [ Internal Function ]           Jss_beta
C-----------------------------------------------------------------
        REAL*8 FUNCTION Jss_beta(beta_t, p_t, q_t)
        ! For Js_s in compression zone
        ! Jss_beta(beta_upper)=(-3p/q)*Gs_beta(beta_upper)
        ! Jss_beta(beta_lower)=(-3p/q)*Gs_beta(beta_lower)
        ! Jss_beta(0) = (sign)*r_JsK*K_min
        !
        ! ------------ involved procedure ---------------
        !               Gs_beta
        !
        REAL*8 beta_t, p_t, q_t
        REAL*8 Jss_upper, Jss_lower, Jss_0
                ! Jss values at Beta_upper, Beta_lower & Beta=0
        REAL*8 sign_Js
        !
        Jss_upper = (-3.D0*p_t/q_t)*Gs_Beta(Beta_upper)
        Jss_lower = (-3.D0*p_t/q_t)*Gs_Beta(Beta_lower)
        sign_Js = DABS(-q_t)/(-q_t)
        Jss_0 = sign_Js*r_JsK*K_min
        !
        IF(beta_t <= Beta_upper) THEN
                !
                Jss_beta = LNR(0.D0, Beta_upper, Jss_0, Jss_upper, Beta_t)
                ! LNR(x1, x2, y1, y2, x)
        ELSE IF(beta_t >= Beta_lower) THEN
```

```
                !
                Jss_beta = LNR(2.D0*pi, Beta_lower, Jss_0, Jss_lower, Beta_t)
        END IF
        !
        !
        END FUNCTION Jss_beta
        !
C
C
C-----------------------------------------------------------------
C               [ Internal Function ]           LSPs_beta
C-----------------------------------------------------------------
        REAL*8 FUNCTION LSPs_beta(beta_t)
        ! For LSP_s
        ! LSPs_beta(0) = LSPs_beta(beta_RTE) = LSP_max;
        ! LSPs_beta(beta_RTC) = LSP_min
        ! beta: [0, 2*pi)
        ! ----------------------------------------
        !
        REAL*8 beta_t, beta_RTC, beta_RTE
        beta_RTC=2.16D0
        beta_RTE=4.39D0
        !
        IF(beta_t >= 0.D0 .AND. beta_t <= beta_RTC) THEN
                !
                LSPs_beta = LNR(0.D0, beta_RTC, LSP_max, LSP_min, Beta_t)
                ! LNR(x1, x2, y1, y2, x)
                !
        ELSE IF(beta_t > beta_RTC .AND. beta_t <= beta_RTE) THEN
                !
                LSPs_beta = LNR(beta_RTC, beta_RTE, LSP_min, LSP_max, Beta_t)
        ELSE
                !
                LSPs_beta = LSP_max
        END IF
        !
        !
        END FUNCTION LSPs_beta
        !
C
C
C
        END
```

```
C                                                       END of UMAT
C ***************************************************************
C
C
      SUBROUTINE UVARM(UVAR,DIRECT,T,TIME,DTIME,CMNAME,ORNAME,
     1 NUVARM,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,NDI,NSHR,COORD,
     2 JMAC,JMATYP,MATLAYO,LACCFLA)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME,ORNAME
      CHARACTER*3 FLGRAY(15)
      DIMENSION UVAR(NUVARM),DIRECT(3,3),T(3,3),TIME(2)
      DIMENSION ARRAY(22),JARRAY(22),JMAC(*),JMATYP(*),COORD(*)
          !
          INTEGER NTENS
          NTENS = NDI + NSHR
C
C Error counter:
      JERROR = 0
C Stress tensor:
      CALL GETVRM('E',ARRAY,JARRAY,FLGRAY,JRCD,JMAC,JMATYP,
     1 MATLAYO,LACCFLA)
      ! 'SDV': The 'Output Variable Identifier' for 'Solution-dependent State Variables'
          JERROR = JERROR + JRCD
          !
          SELECT CASE (NTENS)
          CASE (4)
          ! dimension-2 is vertical, dimension-1 is in-plane horizontal
                    UVAR(1) = -(ARRAY(1)+ARRAY(2))*100.D0               ! ev
                    UVAR(2) = -100.D0*(ARRAY(2)-ARRAY(1))               ! es
                    ! consistent with simplified definition for q
          CASE (6)
                    UVAR(1) = -(ARRAY(1)+ARRAY(2)+ARRAY(3))*100.D0
                    UVAR(2) = -200.D0*(ARRAY(3)-ARRAY(1))/3.D0
          END SELECT
          !
          IF(JERROR.NE.0)THEN
        WRITE(6,*) 'REQUEST ERROR IN UVARM FOR ELEMENT NUMBER ',
     1      NOEL,'INTEGRATION POINT NUMBER ',NPT
      ENDIF
          !
          ! ------------------------------------------------------------
```

```
      CALL GETVRM('S',ARRAY,JARRAY,FLGRAY,JRCD,JMAC,JMATYP,
     1 MATLAYO,LACCFLA)
      ! 'SDV': The 'Output Variable Identifier' for 'Solution-dependent State Variables'
      JERROR = JERROR + JRCD
      !
      SELECT CASE (NTENS)
      CASE (4)
      ! dimension-2 is vertical, dimension-1 is in-plane horizontal
            UVAR(3) = -(ARRAY(1)+ARRAY(2))*0.5D0                    ! p'
            UVAR(4) = -(ARRAY(2)-ARRAY(1))                                ! q
            ! consistent with simplified definition for q
      CASE (6)
            UVAR(3) = -(ARRAY(1)+ARRAY(2)+ARRAY(3))/3.D0
            UVAR(4) = -(ARRAY(3)-ARRAY(1))
      END SELECT
C If error, write comment to .DAT file:
      IF(JERROR.NE.0)THEN
        WRITE(6,*) 'REQUEST ERROR IN UVARM FOR ELEMENT NUMBER ',
     1      NOEL,'INTEGRATION POINT NUMBER ',NPT
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE SIGINI(SIGMA,COORDS,NTENS,NCRDS,NOEL,NPT,LAYER,
     1 KSPT,LREBAR,REBARN)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION SIGMA(NTENS),COORDS(NCRDS)
      CHARACTER*80 REBARN

      SIGMA(2) = -317.5*(4.3-COORDS(2))/(4.3+12.5)
      SIGMA(1) = 0.667*SIGMA(2)
      SIGMA(3) = SIGMA(1)
      SIGMA(4) = 0

      RETURN
      END
```

D. INPUT FILE (ABAQUS) FOR LURIE CENTER PREDICTION

```
*Heading
** Job name: Lurie-CAE Model name: Lurie
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Soil-Left
*End Part
*Part, name=Soil-Right
*End Part
*Part, name=Strut-1
*End Part
*Part, name=Strut-2
*End Part
*Part, name=Strut-3
*End Part
*Part, name=Wall
*End Part
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Strut-1, part=Strut-1
*Node
      1,          0.,          2.1
      2,        -6.4,           0.
*Element, type=T2D2
1, 1, 2
** Region: (Strut-1:Picked), (Beam Orientation:Picked)
*Elset, elset=_PickedSet6, internal
 1,
** Section: Strut-1
*Solid Section, elset=_PickedSet6, material=Strut-1
1.,
*End Instance
**
*Instance, name=Strut-2, part=Strut-2
```

```
*Node
      1,          0.,         -1.2
      2,        -5.5,         -2.1
*Element, type=T2D2
1, 1, 2
** Region: (Strut-2:Picked), (Beam Orientation:Picked)
*Elset, elset=_PickedSet6, internal
 1,
** Section: Strut-2
*Solid Section, elset=_PickedSet6, material=Strut-2
1.,
*End Instance
**
*Instance, name=Strut-3, part=Strut-3
*Node
      1,          0.,         -5.2
      2,        -2.6,         -6.7
*Element, type=T2D2
1, 1, 2
** Region: (Strut-3:Picked), (Beam Orientation:Picked)
*Elset, elset=_PickedSet4, internal
 1,
** Section: Strut-3
*Solid Section, elset=_PickedSet4, material=Strut-3
1.,
*End Instance
**
*Instance, name=Soil-Left, part=Soil-Left
*Node
      1,          15.,        -24.4
      2,          15.,        -20.7
      3,          0.,         -20.7
   ……
   skipped
   ……
    758,        -52.5,    -2.549932
    759,        -52.5,   -0.8499314
    760,       -56.25,    -1.699931
*Element, type=CPE8R
1,   1,   2,  43,  48, 267, 268, 269, 270
2,  48,  43,  44,  47, 269, 271, 272, 273
3,  47,  44,  45,  46, 272, 274, 275, 276
……
```

skipped
……
227, 265, 159, 158, 266, 756, 638, 759, 757
228, 183, 266, 104, 25, 758, 760, 410, 745
229, 266, 158, 16, 104, 759, 642, 414, 760
*Nset, nset=Fill
 16, 17, 18, 19, 20, 21, 22, 23, 24, 81, 82, 83, 84, 85, 86, 87
 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103
155, 156, 157, 158, 159, 160, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195
238, 239, 240, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358
359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374
375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390
391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406
407, 408, 409, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638
639, 640, 641, 642, 643
*Elset, elset=Fill
 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
 43, 44, 45, 46, 47, 48, 49, 50, 155, 156, 157, 158, 159, 160, 161, 162
*Nset, nset=Sand-Up
 16, 19, 21, 23, 25, 26, 27, 28, 29, 86, 87, 88, 90, 91, 92, 93
 94, 95, 96, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116
158, 159, 160, 181, 182, 183, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205
264, 265, 266, 349, 355, 360, 365, 373, 378, 383, 388, 393, 398, 403, 408, 410
411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426
427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442
443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458
459, 460, 461, 628, 633, 638, 642, 745, 747, 749, 750, 751, 752, 753, 754, 755
756, 757, 758, 759, 760
*Elset, elset=Sand-Up
 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66
 67, 68, 69, 70, 71, 72, 73, 74, 222, 223, 224, 225, 226, 227, 228, 229
*Nset, nset=Sand-Down
  8,   9, 12, 14, 25, 26, 27, 29, 55, 56, 57, 58, 59, 60, 61, 69
 70, 71, 75, 76, 77, 106, 107, 108, 110, 111, 112, 113, 114, 115, 116, 181
182, 183, 297, 301, 304, 307, 310, 313, 316, 319, 322, 325, 328, 331, 334, 337
340, 343, 413, 418, 422, 426, 432, 436, 440, 444, 448, 452, 456, 460, 462, 463
464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 744, 745, 746, 747
748, 749, 750
*Elset, elset=Sand-Down
 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 218, 219, 220, 221
*Nset, nset=Clay-Soft
  8,   9, 10, 11, 12, 13, 14, 15, 36, 37, 38, 39, 40, 55, 56, 57
 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73

```
 74,  75,  76,  77,  78,  79,  80, 129, 130, 131, 132, 133, 134, 135, 136, 137
138, 139, 140, 141, 142, 143, 144, 145, 152, 153, 154, 206, 207, 208, 209, 210
211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 258
259, 260, 261, 262, 263, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307
308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323
324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339
340, 341, 342, 343, 344, 345, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510
511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526
527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542
543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558
559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574
575, 576, 577, 578, 609, 615, 621, 625, 727, 728, 729, 730, 731, 732, 733, 734
735, 736, 737, 738, 739, 740, 741, 742, 743
*Elset, elset=Clay-Soft
 11,  12,  13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26
 95,  96,  97,  98,  99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126
127, 128, 129, 130, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217
*Nset, nset=Clay-Stiff
  2,   3,   5,   6,   7,  31,  32,  35,  36,  37,  38,  40,  42,  43,  44,  45
 49,  50,  51,  52,  53,  54, 117, 118, 119, 123, 124, 125, 133, 134, 135, 138
139, 140, 141, 142, 143, 144, 146, 147, 148, 149, 150, 151, 152, 153, 154, 163
164, 165, 166, 167, 168, 169, 170, 178, 179, 180, 184, 185, 226, 227, 228, 229
230, 231, 232, 233, 234, 235, 236, 237, 241, 242, 243, 244, 245, 246, 247, 248
249, 250, 251, 252, 253, 254, 255, 256, 257, 268, 271, 274, 277, 280, 281, 282
283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 477, 480
483, 486, 490, 493, 496, 499, 504, 512, 518, 524, 535, 541, 547, 553, 559, 565
571, 577, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592
593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608
609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624
625, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664
665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680
681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696
697, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726
*Elset, elset=Clay-Stiff
  5,   6,   7,   8,   9,  10, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140
141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 166, 167
168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183
184, 185, 186, 187, 188, 189, 198, 199, 200, 201, 202, 203, 204, 205
*Nset, nset=Hard-Pan
  1,   2,   3,   4,   7,  30,  31,  32,  33,  34,  35,  41,  43,  44,  45,  46
 47,  48,  52,  53, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128
161, 162, 163, 164, 165, 166, 167, 168, 169, 171, 172, 173, 174, 175, 176, 177
```

```
 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 291, 294, 296
 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491
 492, 493, 494, 495, 496, 497, 498, 499, 500, 644, 645, 646, 647, 648, 649, 654
 660, 666, 672, 678, 684, 690, 696, 698, 699, 700, 701, 702, 703, 704, 705, 706
 707, 708, 709, 710, 711, 712
*Elset, elset=Hard-Pan
   1,   2,   3,   4,  87,  88,  89,  90,  91,  92,  93,  94, 163, 164, 165, 190
 191, 192, 193, 194, 195, 196, 197
** Region: (Hard-Pan:Hard-Pan)
** Section: Hard-Pan
*Solid Section, elset=Hard-Pan, material=Hard-Pan
1.,
** Region: (Clay-Stiff:Clay-Stiff)
** Section: Clay-Stiff
*Solid Section, elset=Clay-Stiff, material=Clay-Stiff
1.,
** Region: (Clay-Soft:Clay-Soft)
** Section: Clay-Soft
*Solid Section, elset=Clay-Soft, material=Clay-Soft
1.,
** Region: (Fill:Fill)
** Section: Fill
*Solid Section, elset=Fill, material=Fill
1.,
** Region: (Sand-Up:Sand-Up)
** Section: Sand-Up
*Solid Section, elset=Sand-Up, material=Sand-Up
1.,
** Region: (Sand-Down:Sand-Down)
** Section: Sand-Down
*Solid Section, elset=Sand-Down, material=Sand-Down
1.,
*End Instance
**
*Instance, name=Soil-Right, part=Soil-Right
*Node
      1,        35.1,         -8.5
      2,         15.,         -8.5
      3,         15.,        -12.5
   ……
   skipped
   ……
```

```
   243,         3.75,          3.2
   244,        1.875,          4.3
   245,           0.,          3.2
*Element, type=CPE8R
1,  1,  31,  84,  36,  89,  90,  91,  92
2,  31,  32,  85,  84,  93,  94,  95,  90
3,  32,   2,  33,  85,  96,  97,  98,  94
……
skipped
……
68,  73,  81,  82,  74, 239, 240, 241, 212
69,  74,  82,  83,  75, 241, 242, 243, 215
70,  75,  83,  30,  27, 243, 244, 245, 218
*Nset, nset=Fill
   7,   8,   9,  10,  11,  12,  27,  28,  30,  44,  45,  46,  47,  48,  49,  73
  74,  75,  76,  77,  78,  81,  82,  83, 126, 127, 128, 129, 130, 131, 132, 133
 134, 135, 136, 137, 138, 139, 140, 141, 142, 209, 210, 211, 212, 213, 214, 215
 216, 217, 218, 219, 220, 238, 239, 240, 241, 242, 243, 244, 245
*Elset, elset=Fill
 15, 16, 17, 18, 19, 20, 49, 50, 51, 52, 67, 68, 69, 70
*Nset, nset=Sand-Up
  11,  12,  16,  17,  18,  19,  25,  26,  28,  48,  49,  55,  56,  57,  58,  67
  68,  69,  70,  71,  72,  76,  77,  78, 137, 139, 141, 158, 159, 160, 161, 162
 163, 164, 165, 166, 167, 168, 169, 170, 171, 197, 198, 199, 200, 201, 202, 203
 204, 205, 206, 207, 208, 211, 214, 217, 220, 234, 235, 236, 237
*Elset, elset=Sand-Up
 28, 29, 30, 31, 32, 33, 45, 46, 47, 48, 63, 64, 65, 66
*Nset, nset=Sand-Down
  18,  19,  22,  23,  24,  26,  57,  58,  62,  63,  64,  65,  66,  70,  71,  72
 166, 168, 170, 181, 182, 183, 184, 185, 186, 187, 189, 191, 193, 195, 199, 202
 205, 208, 230, 231, 232, 233
*Elset, elset=Sand-Down
 38, 39, 40, 59, 60, 61, 62
*Nset, nset=Clay-Soft
   1,   2,   3,   4,   5,   6,  20,  21,  22,  23,  24,  29,  31,  32,  33,  34
  35,  36,  37,  38,  39,  40,  41,  42,  43,  59,  60,  61,  62,  63,  64,  65
  66,  79,  80,  84,  85,  86,  87,  88,  89,  90,  91,  92,  93,  94,  95,  96
  97,  98,  99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112
 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 172, 173, 174
 175, 176, 177, 178, 179, 180, 182, 184, 186, 188, 189, 190, 191, 192, 193, 194
 195, 196, 221, 222, 223, 224, 225, 226, 227, 228, 229
*Elset, elset=Clay-Soft
  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 34, 35
```

```
 36, 37, 41, 42, 43, 44, 53, 54, 55, 56, 57, 58
*Nset, nset=Clay-Stiff
   3,   4,   6,  13,  14,  15,  34,  35,  41,  42,  43,  50,  51,  52,  53,  54
 100, 103, 105, 119, 121, 123, 125, 143, 144, 145, 146, 147, 148, 149, 150, 151
 152, 153, 154, 155, 156, 157
*Elset, elset=Clay-Stiff, generate
 21,  27,   1
** Region: (Clay-Soft:Clay-Soft)
** Section: Clay-Soft
*Solid Section, elset=Clay-Soft, material=Clay-Soft
1.,
** Region: (Fill:Fill)
** Section: Fill
*Solid Section, elset=Fill, material=Fill
1.,
** Region: (Clay-Stiff:Clay-Stiff)
** Section: Clay-Stiff
*Solid Section, elset=Clay-Stiff, material=Clay-Stiff
1.,
** Region: (Sand-Up:Sand-Up)
** Section: Sand-Up
*Solid Section, elset=Sand-Up, material=Sand-Up
1.,
** Region: (Sand-Down:Sand-Down)
** Section: Sand-Down
*Solid Section, elset=Sand-Down, material=Sand-Down
1.,
*End Instance
**
*Instance, name=Wall, part=Wall
         0.,       -15.2,          0.
*Node
      1,          0.,          0.
      2,          0.,         10.
      3,          0.,         14.
      4,          0.,       17.3
      5,          0.,       19.5
      6,          0.,          2.
      7,          0.,          4.
      8,          0.,          6.
      9,          0.,          8.
     10,          0.,         12.
     11,          0.,       15.65
```

```
    12,          0.,          1.
    13,          0.,          3.
    14,          0.,          5.
    15,          0.,          7.
    16,          0.,          9.
    17,          0.,         11.
    18,          0.,         13.
    19,          0.,       14.825
    20,          0.,       16.475
    21,          0.,         18.4
*Element, type=B22
1,  1, 12,  6
2,  6, 13,  7
3,  7, 14,  8
4,  8, 15,  9
5,  9, 16,  2
6,  2, 17, 10
7, 10, 18,  3
 8,  3, 19, 11
 9, 11, 20,  4
10,  4, 21,  5
*Nset, nset=Wall, generate
  1,  21,   1
*Elset, elset=Wall, generate
 1, 10,  1
** Region: (Wall:Wall), (Beam Orientation:Picked)
** Section: Wall  Profile: Wall
*Beam General Section, elset=Wall, poisson = 0.3, section=RECT
1., 0.424
0.,0.,-1.
8.068e+06, 3.103e+06
*End Instance
*Nset, nset=Strut-1, instance=Strut-1
 1, 2
*Elset, elset=Strut-1, instance=Strut-1
 1,
*Nset, nset=Strut-2, instance=Strut-2
 1, 2
*Elset, elset=Strut-2, instance=Strut-2
 1,
*Nset, nset=Strut-3, instance=Strut-3
 1, 2
*Elset, elset=Strut-3, instance=Strut-3
```

```
 1,
*Nset, nset=BC-Bottom, instance=Soil-Left
   1,   4,  30,  33,  34,  41,  46,  47,  48, 120, 121, 122, 126, 127, 128, 161
 162, 171, 172, 173, 174, 175, 176, 177, 270, 273, 276, 279, 479, 482, 485, 488
 492, 495, 498, 500, 644, 646, 648, 698, 700, 702, 704, 706, 708, 710, 712
*Elset, elset=BC-Bottom, instance=Soil-Left
   1,   2,   3,   4,  87,  88,  89,  90,  91,  92,  93,  94, 163, 164, 165, 190
 191, 192, 193, 194, 195, 196, 197
*Nset, nset=BC-Sides, instance=Soil-Left
   5,   7,  14,  15,  18,  19,  26,  32,  33,  37,  41,  54,  85, 105, 131, 132
 148, 149, 283, 292, 344, 364, 367, 425, 427, 463, 487, 523, 526, 527, 598, 600
 601, 649
*Nset, nset=BC-Sides, instance=Soil-Right
   1,   4,   9,  10,  12,  14,  17,  19,  23,  29,  36,  92, 101, 134, 142, 149
 164, 171, 187, 226, 229
*Elset, elset=BC-Sides, instance=Soil-Left
   5,   8,  26,  33,  34,  57,  58,  75,  90, 104, 105, 106, 140, 141, 142, 165
*Elset, elset=BC-Sides, instance=Soil-Right
  1,  4, 17, 20, 23, 30, 33, 40, 55, 58
*Nset, nset=Removal-1, instance=Soil-Right
   7,   8,   9,  10,  27,  30,  44,  45,  46,  47,  73,  74,  75,  81,  82,  83
 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 209, 212, 215, 218, 238, 239
 240, 241, 242, 243, 244, 245
*Elset, elset=Removal-1, instance=Soil-Right
 15, 16, 17, 67, 68, 69, 70
*Nset, nset=Removal-2, instance=Soil-Right
   8,   9,  11,  12,  27,  28,  44,  45,  48,  49,  73,  74,  75,  76,  77,  78
 127, 130, 133, 136, 137, 138, 139, 140, 141, 142, 209, 210, 211, 212, 213, 214
 215, 216, 217, 218, 219, 220
*Elset, elset=Removal-2, instance=Soil-Right
 18, 19, 20, 49, 50, 51, 52
*Nset, nset=Removal-3, instance=Soil-Right
  11,  12,  16,  17,  25,  28,  48,  49,  55,  56,  67,  68,  69,  76,  77,  78
 137, 139, 141, 158, 159, 160, 161, 162, 163, 164, 197, 200, 203, 206, 211, 214
 217, 220, 234, 235, 236, 237
*Elset, elset=Removal-3, instance=Soil-Right
 28, 29, 30, 63, 64, 65, 66
*Nset, nset=Removal-4, instance=Soil-Right
  16,  17,  18,  19,  25,  26,  55,  56,  57,  58,  67,  68,  69,  70,  71,  72
 159, 161, 163, 165, 166, 167, 168, 169, 170, 171, 197, 198, 199, 200, 201, 202
 203, 204, 205, 206, 207, 208
*Elset, elset=Removal-4, instance=Soil-Right
 31, 32, 33, 45, 46, 47, 48
```

```
*Nset, nset=Removal-5, instance=Soil-Right
 18,  19,  22,  23,  24,  26,  57,  58,  62,  63,  64,  65,  66,  70,  71,  72
166, 168, 170, 181, 182, 183, 184, 185, 186, 187, 189, 191, 193, 195, 199, 202
205, 208, 230, 231, 232, 233
*Elset, elset=Removal-5, instance=Soil-Right
38, 39, 40, 59, 60, 61, 62
*Nset, nset=Removal-6, instance=Soil-Right
 20,  21,  22,  23,  24,  29,  59,  60,  61,  62,  63,  64,  65,  66,  79,  80
173, 176, 178, 180, 182, 184, 186, 188, 189, 190, 191, 192, 193, 194, 195, 196
221, 222, 223, 224, 225, 226
*Elset, elset=Removal-6, instance=Soil-Right
41, 42, 43, 44, 53, 54, 55
*Nset, nset=Removal-7, instance=Soil-Right
  1,   2,   5,  20,  21,  29,  31,  32,  37,  38,  39,  59,  60,  61,  79,  80
 89,  93,  96, 106, 109, 112, 115, 172, 173, 174, 175, 176, 177, 178, 179, 180
221, 223, 225, 227, 228, 229
*Elset, elset=Removal-7, instance=Soil-Right
34, 35, 36, 37, 56, 57, 58
*Nset, nset=Soil-All, instance=Soil-Left, generate
  1,  760,    1
*Nset, nset=Soil-All, instance=Soil-Right, generate
  1,  245,    1
*Elset, elset=Soil-All, instance=Soil-Left, generate
  1,  229,    1
*Elset, elset=Soil-All, instance=Soil-Right, generate
 1,  70,   1
*Nset, nset=Wall, instance=Wall, generate
 1,  21,   1
*Elset, elset=Wall, instance=Wall, generate
1, 10,  1
*Elset, elset=_Surf-Soil-L_S4, internal, instance=Soil-Left
11,
*Elset, elset=_Surf-Soil-L_S3, internal, instance=Soil-Left
 49,  50,  73,  74, 128, 129, 130, 187
*Elset, elset=_Surf-Soil-L_S2, internal, instance=Soil-Left
86,
*Surface, type=ELEMENT, name=Surf-Soil-L
_Surf-Soil-L_S4, S4
_Surf-Soil-L_S3, S3
_Surf-Soil-L_S2, S2
*Elset, elset=_Surf-Soil-Dn_S1, internal, instance=Soil-Left
  5,   6,   7, 198, 199, 200, 201
*Surface, type=ELEMENT, name=Surf-Soil-Dn
```

```
_Surf-Soil-Dn_S1, S1
*Elset, elset=_Surf-Soil-R_S2, internal, instance=Soil-Right
 10, 14
*Elset, elset=_Surf-Soil-R_S4, internal, instance=Soil-Right
 24, 34
*Elset, elset=_Surf-Soil-R_S3, internal, instance=Soil-Right
 44, 48, 52, 62, 66, 70
*Surface, type=ELEMENT, name=Surf-Soil-R
_Surf-Soil-R_S2, S2
_Surf-Soil-R_S4, S4
_Surf-Soil-R_S3, S3
*Elset, elset=_Surf-Soil-Up_S1, internal, instance=Soil-Right, generate
 21,  27,   1
*Surface, type=ELEMENT, name=Surf-Soil-Up
_Surf-Soil-Up_S1, S1
*Elset, elset=_Surf-Wall-L_SPOS, internal, instance=Wall, generate
 1, 10,  1
*Surface, type=ELEMENT, name=Surf-Wall-L
_Surf-Wall-L_SPOS, SPOS
*Elset, elset=_Surf-Wall-R_SNEG, internal, instance=Wall, generate
 1, 10,  1
*Surface, type=ELEMENT, name=Surf-Wall-R
_Surf-Wall-R_SNEG, SNEG
** Constraint: Tie-SoilSoil
*Tie, name=Tie-SoilSoil, adjust=yes
Surf-Soil-Up, Surf-Soil-Dn
** Constraint: Tie-WallSoil
*Tie, name=Tie-WallSoil, adjust=yes
Surf-Wall-R, Surf-Soil-R
**
** CONNECTORS
**
*Element, type=CONN2D2, elset=_Strut1-1_CnSet_
1, Wall.4, Strut-1.1
*Connector Section, elset=_Strut1-1_CnSet_
Join,
*Element, type=CONN2D2, elset=_Strut1-2_CnSet_
2,  , Strut-1.2
*Connector Section, elset=_Strut1-2_CnSet_
Join,
*Element, type=CONN2D2, elset=_Strut2-1_CnSet_
3, Wall.3, Strut-2.1
*Connector Section, elset=_Strut2-1_CnSet_
```

```
Join,
*Element, type=CONN2D2, elset=_Strut2-2_CnSet_
4,  , Strut-2.2
*Connector Section, elset=_Strut2-2_CnSet_
Join,
*Element, type=CONN2D2, elset=_Strut3-1_CnSet_
5, Wall.2, Strut-3.1
*Connector Section, elset=_Strut3-1_CnSet_
Join,
*Element, type=CONN2D2, elset=_Strut3-2_CnSet_
6,  , Strut-3.2
*Connector Section, elset=_Strut3-2_CnSet_
Join,
*End Assembly
**
** MATERIALS
**
*Material, name=Clay-Soft
*Density
 1.89,
*Depvar
     10,
*User Material, constants=2, unsymm
17000., 0.333
*User Output Variables
      5,
*Material, name=Clay-Stiff
*Density
 1.89,
*Elastic
171000., 0.49
*Material, name=Fill
*Density
 1.89,
*Elastic
51000., 0.2
*Mohr Coulomb
30.,2.
*Mohr Coulomb Hardening
20.,0.
*Material, name=Grout
*Elastic
 3e+06, 0.2
```

```
*Material, name=Hard-Pan
*Density
 1.89,
*Elastic
677000., 0.49
*Material, name=Sand-Down
*Density
 1.89,
*Elastic
175000., 0.365
*Mohr Coulomb
40.,8.
*Mohr Coulomb Hardening
 0.1,0.
*Material, name=Sand-Up
*Density
 1.89,
*Elastic
79000., 0.391
*Mohr Coulomb
35.,5.
*Mohr Coulomb Hardening
 0.1,0.
*Material, name=Strut-1
*Elastic
69600., 0.3
*Material, name=Strut-2
*Elastic
57500., 0.3
*Material, name=Strut-3
*Elastic
17200., 0.3
*Material, name=Tieback
*Elastic
 2.05e+08, 0.3
**
** INTERACTION PROPERTIES
**
*Surface Interaction, name=IntProp-L
1.,
*Friction, slip tolerance=0.005
 0.3,
*Surface Behavior, no separation, pressure-overclosure=HARD
```

```
*Surface Interaction, name=IntProp-R
1.,
*Friction, slip tolerance=0.005
 0.3,
*Surface Behavior, no separation, pressure-overclosure=HARD
**
** BOUNDARY CONDITIONS
**
** Name: BC-Bottom Type: Displacement/Rotation
*Boundary
BC-Bottom, 1, 1
BC-Bottom, 2, 2
** Name: BC-Sides Type: Displacement/Rotation
*Boundary
BC-Sides, 1, 1
**
** INTERACTIONS
**
** Interaction: Int-SoilSoil
*Contact Pair, interaction=IntProp-L, small sliding
Surf-Soil-R, Surf-Soil-L
** Interaction: Int-SoilWall
*Contact Pair, interaction=IntProp-L, small sliding
Surf-Soil-L, Surf-Wall-L
*INITIAL CONDITIONS, TYPE=STRESS, USER
** ----------------------------------------------------------------
**
** STEP: Step-1
**
*Step, name=Step-1
*Geostatic
*MODEL CHANGE, REMOVE
Wall
*MODEL CHANGE, REMOVE
Strut-1, Strut-2, Strut-3
*MODEL CHANGE, TYPE=CONTACT PAIR, REMOVE
Surf-Soil-L, Surf-Wall-L
**
** LOADS
**
** Name: gravity   Type: Gravity
*Dload
Soil-All, GRAV, 10., 0., -1.
```

```
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*El Print, freq=999999
*Node Print, freq=999999
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-2
**
*Step, name=Step-2
*Static
1., 1., 1e-05, 1.
*MODEL CHANGE, ADD
Wall
*MODEL CHANGE, TYPE=CONTACT PAIR, ADD
Surf-Soil-L, Surf-Wall-L
*MODEL CHANGE, TYPE=CONTACT PAIR, REMOVE
Surf-Soil-R, Surf-Soil-L
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
```

```
** ----------------------------------------------------------------
**
** STEP: Step-3
**
*Step, name=Step-3
*Static
0.1, 1., 1e-05, 0.2
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
, 0.03,
*CONTROLS, PARAMETERS=LINE SEARCH
4,
**
*MODEL CHANGE, REMOVE
Removal-1
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-4
**
*Step, name=Step-4
*Static
1., 1., 1e-05, 1.
*MODEL CHANGE, ADD
Strut-1
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
```

```
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-5
**
*Step, name=Step-5
*Static
0.1, 1., 1e-05, 0.15
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
, 0.03,
*CONTROLS, PARAMETERS=LINE SEARCH
4,
**
*MODEL CHANGE, REMOVE
Removal-2
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-6
**
*Step, name=Step-6
*Static
0.1, 1., 1e-05, 0.15
```

```
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
, 0.05,
*CONTROLS, PARAMETERS=LINE SEARCH
4,
**
*MODEL CHANGE, REMOVE
Removal-3
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-7
**
*Step, name=Step-7
*Static
1., 1., 1e-05, 1.
*MODEL CHANGE, ADD
Strut-2
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
```

```
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-8
**
*Step, name=Step-8
*Static
0.05, 1., 1e-05, 0.1
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
0.01, 0.1,
*CONTROLS, PARAMETERS=LINE SEARCH
4,
*CONTROLS, PARAMETERS=TIME INCREMENTATION
8, 10,
0.5, 0.8,
**
*MODEL CHANGE, REMOVE
Removal-4
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-9
**
*Step, name=Step-9
*Static
0.05, 1., 1e-05, 0.1
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
0.015, 0.1, , , 0.02
*CONTROLS, PARAMETERS=LINE SEARCH
4,
```

```
*CONTROLS, PARAMETERS=TIME INCREMENTATION
8, 10,
0.5, 0.8,
**
*MODEL CHANGE, REMOVE
Removal-5
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-10
**
*Step, name=Step-10
*Static
1., 1., 1e-05, 1.
*MODEL CHANGE, ADD
Strut-3
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
```

```
** ----------------------------------------------------------------
**
** STEP: Step-11
**
*Step, name=Step-11
*Static
0.05, 1., 1e-05, 0.1
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
0.01, 0.1,
*CONTROLS, PARAMETERS=LINE SEARCH
4,
*CONTROLS, PARAMETERS=TIME INCREMENTATION
8, 10,
0.5, 0.8,
**
*MODEL CHANGE, REMOVE
Removal-6
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
** ----------------------------------------------------------------
**
** STEP: Step-12
**
*Step, name=Step-12
*Static
0.05, 1., 1e-05, 0.1
*CONTROLS, PARAMETERS=FIELD, FIELD=DISPLACEMENT
0.01, 0.1
*CONTROLS, PARAMETERS=LINE SEARCH
4,
*CONTROLS, PARAMETERS=TIME INCREMENTATION
```

```
8, 10,
0.5, 0.8,
**
*MODEL CHANGE, REMOVE
Removal-7
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
U,
*Element Output
E, LE, S, SDV, SF, UVARM
*Output, history, frequency=0
*End Step
```