Applications of artificial intelligence in constitutive modelling of soils.

Drakos, Stefanos

Applications of Artificial Intelligence i
Constitutive Modelling of Soils

STEFANOS DRAKOS
B.Sc. PhD
(ARISTOTLE UNIVERSITY OF THSESALLONIKI)

THESIS SUBMITTED TO THE UNIVERSITY OF WALES IN CANDIDATUR
FOR THE DEGREE OF MASTER OF PHILOSOPHY

SUPERVISOR: PROFESSOR G. PANDE

April 2008

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

SUMMARY

ın appropriate constitutive model embedded in a finite element engine is the key to the uccessful prediction of the observed behaviour of geotechnical structures. However, to apture the behaviour of geomaterials accurately, the constitutive models have to be omplex involving a large number of material parameters and constants. This thesis resents a methodology for converting or recasting complex constitutive models for eomaterials developed based on any constitutive theory into a fully trained Artificial leural Network (ANN), which is then embedded in an appropriate solution code. The ngth of strain trajectory traced by a material point, also called 'intrinsic time' is used as n additional input parameter in training. For the purpose of illustration, two constitutive ıodels viz. Hardening Soil Model available in the commercial software, PLAXIS and a vo-surface deviatoric hardening model in the multilaminate framework developed by ee and Pande (2004) have been cast in the form of an ANN.

Acknowledgements

After my PhD thesis I realised that I need to learn more on the Finite Elements methods and more specific on the Constitutive Modelling. Thus, I decide to contact with Professor Pande which is one of the leader in the area. Professor Pande accepts me as his student and I am grateful about that.

I would like to thank Professor G. Pande not only because he gave me the opportunity to work with him but because he become for me a mentor.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

1.1 GENERAL

Appropriate constitutive models of soils are the key to a successful prediction of the behaviour of geotechnical structures. A large number of models based on various constitutive theories have been proposed in the last three decades. All of them assume a-priori, a mathematical framework of the model and the material parameters corresponding to the assumed framework have to be identified from physical material tests. Many material parameters in complex constitutive theories have no physical meaning, are difficult to determine and have to be identified by trial and error from numerical simulations. In spite of this, many features of soil behaviour such as stiffness at small strains, higher stiffness on reversal of stress path, influence of rotation of principal stress axes etc. have not been captured in a single model.

Thus, it is likely that models of greater complexity will have to be developed in the future. In recent years a number of applications of Artificial Neural Networks (ANNs) leading to Neural Network based Constitutive Models (NNCMs) have been proposed by a number of researchers (Ghabbousi, et all 1991, Shin & Pande, 2001, 2002).

The objective of this thesis is to demonstrate how synthetic data from any constitutive model can be used to successfully train a NNCM. We choose the Hardening Soil Model (HSM) of the well-known commercial code PLAXIS and a complex Two-surface Deviatoric Hardening Model (TDHM) in the multilaminate framework for cyclic loading developed by Lee & Pande (2004) for the purpose of illustration.

1.2 LAYOUT OF THESIS

In Chapter 2.0 a general description of ANN and the historical development of this achievement are presented. The relationship of linear regression and ANN are briefly analysed and the conclusions are shown. The Artificial Neural Networks is a methodology which is applied by using computer software,

At the Chapter 3.0 the application of Artificial Neural Networks on the Constitutive Modelling is described. The advantage and the reasons of the NNCM development are given. There are three different ways of using this methodology and these are presented in

Chapter 3.0. At the same Chapter the methodology of implementation of Conventional Constitutive model in NNCM is fully described.

Chapter 4.0 gives brief details of the HSM and generation of 'synthetic data' along triaxial compression and extension paths under loading and unloading conditions. At the same Chapter the response of a trained NNCM with the synthetic data are compared. Chapter 5.0 gives details of the TDHM and generation of 'synthetic data' along one-way cyclic triaxial compression under specified volumetric strains. At the end of the current Chapter the response of a trained NNCM with the synthetic data for a number of cycles of loading and unloading are compared. Chapter 6.0 gives conclusions and indicates the implications of computational efficiency in practical problems.

CHAPTER 2 GENERAL DESCRIPTIONS OF NEURAL NETWORKS

2.1 INTRODUCTION

Over the last years, the use of artificial neural networks (ANNs) has increased in many areas of engineering. Artificial neural network (ANN) is a mathematical model or computational model based on biological neural networks. Actually is the advance of regression analysis. From the one parameter linear regression analysis and later on the first development of Linear Perceptrons now the people are able to solve non-linear multi parameters problem.

Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of real biological system.

2.2 ARTIFICIAL NEURAL NETWORKS

A brain is composed of neurons, cells that receive a stimulus which triggers a response from the neuron. The response from the neuron can trigger other neurons, which trigger other neurons, etc. Eventually, this chain of neural activation results in some response from the body, such as recollection of a memory, identifying a sound, movement of a muscle, etc. More specific the human brain consists of an estimated 10 billion neurons (nerve cells) and 6000 times as many synapses (connections) between them Haykin (1994). All information taken in by a human is processed and assessed in this particular part of the body.

As neurons in the nervous system interconnect, they form large clusters and ultimately form the brain. These large clusters inspired the development of neural networks. These networks model the connectivity of the brain with one major assumption. In neural networks, it is assumed that the interactions between neurons in the brain can be represented mathematically. The connections are like synapses. The connections have different strengths, in terms of numerical values. Depending on the connection "strength," they will transmit stronger or weaker signals to the nodes (neurons). The structure of an artificial neural network is described below. Artificial Neural Networks sometimes be referred to as ANN for brevity.

2.3 THE SINGLE NEURON

Let us begin with a very simplistic description of the working of the human brain. A process begins when stimulus is received from the environment. The receptors transform this information to electrical impulses and transmit them to the neural network (neurons and synapses) (Fig. 2.1). After evaluation inside the network, actions are decided and impulses are sent out to the effectors. Both biological and artificial neurons are elementary information processing units. The artificial neuron is best illustrated by analogy with the biological neuron.Fig. 2.2 depicts an artificial neuron. We see that the connections (synapses) w_i transfer the signals (stimulus) u_i into the neuron. w_i can be interpreted as a weight representing the "importance" of that specific input u_i. Inside the neuron the sum of the weighted inputs $w_i u_i$ is taken. Given that this sum u is greater than an externally applied threshold θ, the neuron emits an output z. z is either continuous or binary valued, depending on the activation function (or squashing function). In most cases an activation function can be chosen in order to leads the neuron's output to a range of the interval [0, 1] or [-1, 1].

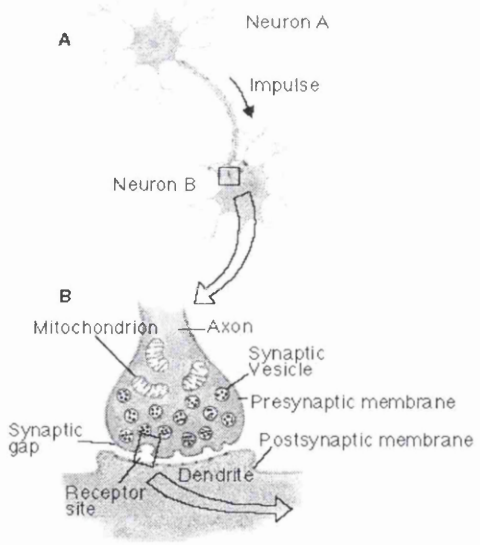


Figure 2.1. Biological neurons
(http://www.alleydog.com/images/2neurons.gif)

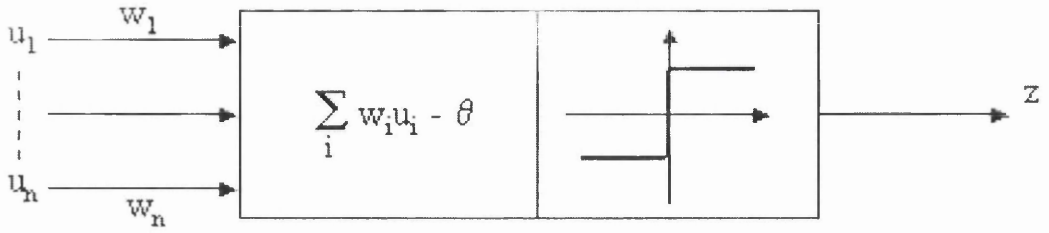


Figure 2.2. The artificial neurons with a threshold function

2.4 BRIEF HISTORY

The branch of artificial intelligence called neural networks dates back to the 1940s, when McCulloch & Pitts (1943) developed the first neural model. They made an attempt to understand and describe the brain functions by mathematical means. McCulloch and Pitts used their neural networks to model logical operators. Contemporary developments in the field of computer science were closely related.

In 1949 Hebb proposed that the synaptic connections inside the brain are constantly changing as a person gains experience. In other words, synapses are either strengthened or weakened depending on whether neurons on either side of the synapse are activated simultaneously or not. Among psychologists Hebb made an instant impact but network modellers have generally shown little interest in his work. In the late fifties Rosenblatt introduced the concept of the perceptron. Basically, the perceptron, which works as a pattern-classifier, is a more sophisticated model of the neuron developed by McCulloch and Pitts. Depending on the amount of neurons incorporated, the perceptron can solve classification problems with various numbers of classes. For a correct classification the classes have to be linearly separable which is a major setback.

Minsky and Papert (1969) also raised the issue of the credit-assignment problem related to the multi-layer perceptron. During the next decade the general interest in neural networks dampened, mainly as a direct consequence of the results reported in the late sixties. Certainly the lack of powerful experimental equipment (computers, work stations etc.) also had an influence on the decline. The interest in neural networks was to be renewed though in eighties.

In 1982 Kohonen introduced the Self-Organising Map (SOM). SOMs use an unsupervised learning algorithm for applications in specifically data mining, image processing and visualisation. As a basic description one can say that high-dimensional

data is transformed and organised in a low-dimensional output space. The same year Hopfield (1982) built a bridge between neural computing and physics. A Hopfield network (consists of symmetric synaptic connections and multiple feedback loops) which is initialised with random weights eventually reaches a final state of stability. From a physicists point of view a Hopfield network corresponds to a dynamical system falling into a state of minimal energy. Two years later the Boltzmann machine was invented. A Boltzmann machine is the name given to a type of simulated annealing stochastic recurrent neural network by Geoffrey Hinton and Terry Sejnowski. Boltzmann machines can be seen as the stochastic, generative counterpart of Hopfield nets.

The discovery of the backpropagation algorithm in 1986 proved crucial for the revival of neural networks. Rummelhart, Hinton and Williams got the credit but it showed that Werbos already in 1974 had introduced the error backpropagation in his PhD thesis. This learning algorithm is unchallenged as the most influential learning algorithm for training of multi-layer perceptrons. We conclude this Chapter with the Radial-Basis Function (RBF) network, which was brought forward by Broomhead and Lowe in 1988. The RBF network emerged as an alternative to the multi-layer perceptron in the search of a solution to the multivariate interpolation problem. By using a set of symmetric non-linear functions in the hidden units of a neural network new properties could be explored. Work by Moody and Darken (presented in 1989) on how to estimate parameters in the basis functions has contributed significantly to the theory of Neural Network.

2.5 THE STRUCTURE OF NEURAL NETWORKS

Preserving the analogy with biological neural systems, an artificial neuron is defined as follows. An artificial neural network consists of several neurons, but they are divided into layers. There is an "input" layer where the initial "stimulus" is received. These neurons are connected to a layer of "hidden" neurons. This hidden layer can be connected to either another hidden layer, or the "output" layer. There can be any number of hidden layers between the input layer and the output layer, but typically the number of hidden layers in any particular ANN is limited. Typically, a neuron of any layer is connected to each other neuron in adjoining layers. Thus, each neuron in the input layer is connected to each neuron in the first hidden layer. Each neuron of the first hidden layer is also connected to each neuron in the next layer (either another hidden layer, or the output layer). Each of these connections between neurons has a "weight" associated with it, and each connection

is treated individually – each connection can have a unique weight associated with it. As the network "learns," these weights are adjusted to represent the strength of connections between neurons. Thus, the network might determine that a particular connection is of no value, and give it a weight of zero. To determine the output from any particular neuron, the neuron must first gather its input. For neurons in the input layer, this is trivial – the input is merely the value given as input to the network. For each other neuron, however, this is more complicated. The input for neurons not in the input layer is a function of the output from each neuron in the previous layer, and the weight of the connection to that neuron. Using Figure 2.4, the input to neuron H_0 in the hidden layer is based upon the output of each neuron in the input layer, multiplied by the connection between that neuron and H_0. To get the total input to H_0, the input values from each connected neuron must be summed. Thus, the input to a neuron j can be written as:

$$\Sigma(O_i * W_{i,j}) \tag{2.1}$$

where O_i is the output from neuron i and $W_{i,j}$ is the weight of the connection between neuron i and neuron j. The total input to the neuron is then used to determine an output value from the neuron. Activations or transfer functions are used for this and introduce the nonlinearity into the network. Almost any nonlinear function can be used, although for learning procedure it must be differentiable and it is preferable if the function is bounded; the sigmoidal functions such as logistic and tanh and the Gaussian function are some common choices. These functions typically fall into one of three categories:

1. linear (or ramp)
2. threshold
3. sigmoid

For linear Function, the output activity is proportional to the total weighted output.

$$F(x)=x \tag{2.2}$$

For threshold Function which, takes on a value of 0 if the summed input is less than a certain threshold value θ, and the value 1 if the summed input is greater than or equal to the threshold value

$$f(x) = \begin{cases} 1 & \text{if } (x \geq \theta) \\ 0 & \text{if } (x < \theta) \end{cases}$$

(2.3)

For sigmoid Function, the output varies continuously but not linearly as the input changes. The sigmoid function is the most common used transfer function, and the output value from a neuron will be between 0 and 1. When a neuron is "inactive" its output will be close to 0. When a neuron is triggered or "active" then its value will be close to 1.

$$f(x) = \frac{1}{1 + e^{X}}$$

(2.4)

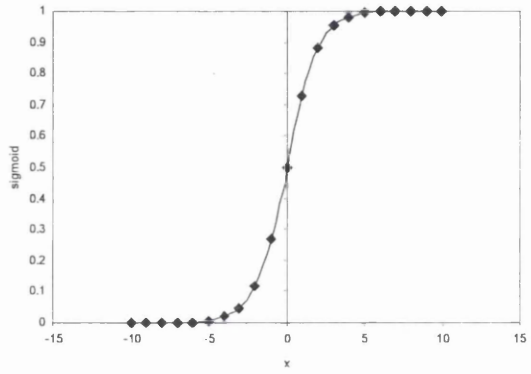


Figure 2.3. Sigmoid function

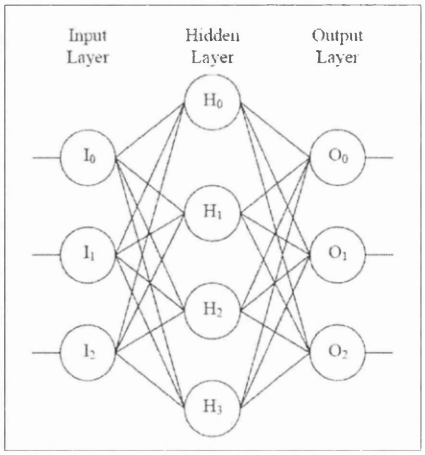


Figure 2.4. A neural network with 3 input neurons, one hidden layer with 4 hidden neurons, and 3 output neurons.

2.6 TRAINING OF NEURAL NETWORK

In order to get an Artificial Neural Network to successfully learn a relationship between input and output variables, it must first be trained. This requires that some training data be gathered. The training data must include the inputs to each input neuron, and the anticipated result from each output neuron.

Finding the right set of weights to accomplish a given task is the central goal. Many learning algorithms have been devised that can calculate the right weights for carrying out many tasks. One of the most widely used of these training methods is called backpropagation. To use this method one needs a training set consisting of many examples of inputs and their desired outputs for a given task.

Backpropagation (BPNN) is a supervised learning technique used for training artificial neural networks.

The BPNN operates in two different phases: one for learning and the other for prediction. The learning phase is composed of forward and reverse paths. In the forward path, a set of learning parameters is presented to the system and the system calculates the output values at output units (outputs) from the input cases given. In the reverse path, the system calculates the sum of the error squared the 'error' being the difference between the ANN output and the known target. The system follows a backpropagation process where the calculated error signals are propagated backward through the network and used to adjust the connection weights. If the error summed over all training cases converges within a certain limit, the learning phase is terminated. On completion of this learning procedure, the weights are stored in the NN for prediction phase. In the prediction phase, the network produces appropriate outputs for new input cases given. The symbolic description of the backpropagation procedure is given in Figure 2.5.

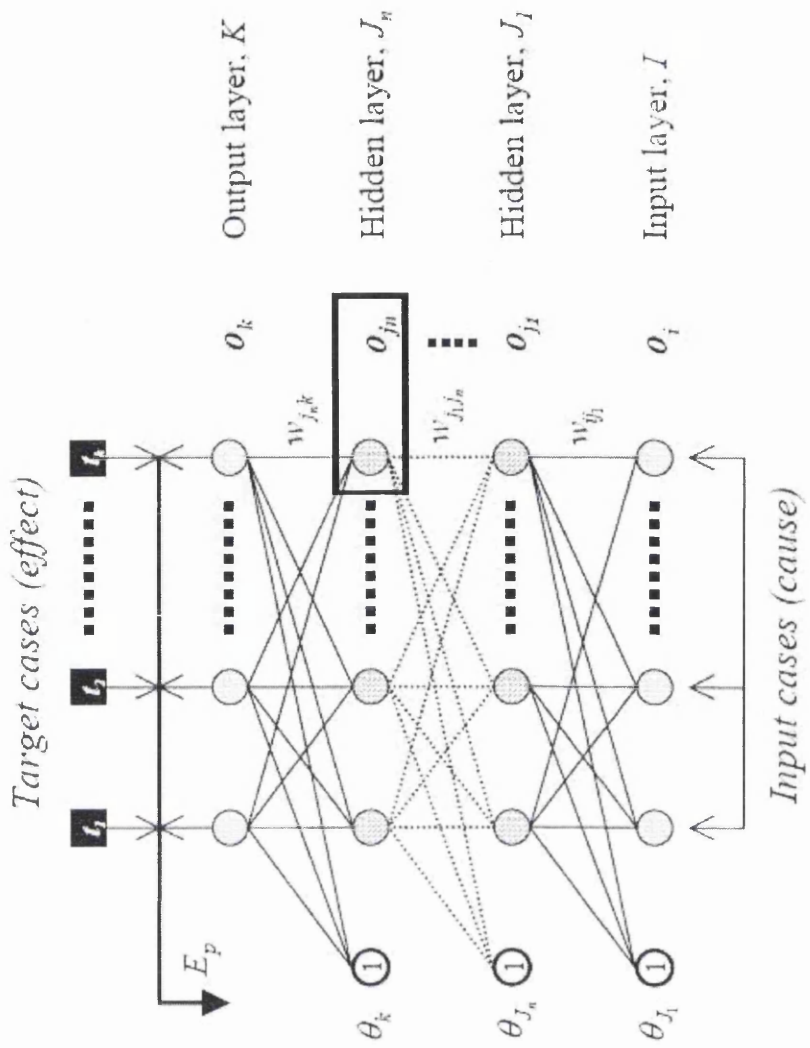


Figure 2.5. A schematic diagram of a BPNN in learning phase. (Shin 2001)

2.7 AVERAGE SYSTEM ERROR, MAXIMUM SYSTEM ERROR AND GENERALISATION ERROR

During training of an ANN, input variables in training cases are allocated to input units, while target variables in the training cases are compared with output values (calculated) at output units. Therefore, the difference between the target and output values for training cases is called the 'residual' or 'error'. This is different from the value of an error function. Note that the residual can be either positive or negative, and negative residuals with large absolute values are typically considered just as bad as large positive residuals. On the other hand, error functions are defined and their absolute values are used to correct the weight value allocated in each connection of a network. The bigger the absolute value of error function is, the greater the error becomes. Usually, an error function is applied to each case and is defined in terms of the target and output values. In this study, the maximum value of the error functions at output units for all the cases is called 'maximum system error' (MSE). In addition, the error function for an entire training set is usually defined as the sum of the case-wise error functions for all the cases in the training set. The value of the sum divided by the number of the cases and output units is called 'average system error' (ASE). The ASE calculated for a test set is called the 'generalisation error' (GE). The error measures (ASE and MSE) are defined as following:

$$\text{Average System Error: } (ASE) = \frac{1}{Pk_{max}} \sum_{p} E_p = \frac{E_t}{Pk_{max}} \tag{2.5}$$

$$\text{Maximum System Error: } (MSE) = MAX\left(\tfrac{1}{2}(t_k - o_k)^2\right)_{p,k} \tag{2.6}$$

Where:

P: is the total number of training cases

k_{max}: is the total number of input units

k: range from 1 to k_{max}

p: range from 1 to P

E_p: is the pattern error:

$$E_p = \tfrac{1}{2} \sum_{k} (t_k - o_k)^2$$

(2.7)

And

E_t: is the total system error:

$$E_t = \sum_p E_p = \frac{1}{2}\sum_p \sum_k (t_k - o_k)^2$$

(2.8)

Where o_k is the output from the output layer K and t_k is the target case from the same layer.

2.8 GENERALISATION OF A NEURAL NETWORK

During learning, the outputs from a supervised ANN are adapted to approximate the target values corresponding to the inputs in the training set. This adaptation is the main concept of training of an ANN. However, the more important purpose of using an ANN is to make it more general, i.e., to have the outputs of the ANN approximate target values for inputs that are not in the training set. Such a generalisation of an ANN may be more important than the training itself, because a trained ANN should be able to produce output acceptable within its population, otherwise output from the ANN has no meaning.

CHAPTER 3 NEURAL NETWORKS BASED CONSTITUTIVE MODEL

3.1 INTRODUCTION

Numerical methods such as the finite element method play an important role as a tool for analysing a variety of problems in engineering. One of the most crucial components of the finite element analyses is the constitutive model used for representing the mechanical response of the material(s) involved. Phenomenological models of complex materials such as soils, concrete, composites etc. are formulated within an assumed mathematical framework and involve determination of a multitude of material parameters. It is generally admitted that in spite of considerable complexities of constitutive theories proposed, it has not been possible to capture the material response universally along all complex stress paths under a wide range of confining pressures. Furthermore, the complexities of constitutive models, in many cases, have inhibited their incorporation in general purpose finite element codes; thus restricting their usefulness in engineering practice.

3.2 WHY APPLY TO MATERIAL MODELLING

Artificial Neural Networks (ANNs) are pattern recognition algorithms using which relationship between a set of 'causes' and 'effects' can be captured. Any set of numeric data can be used to discover the pattern in it, if it exists. In the past two decades, a large number of applications of this methodology in almost all disciples of physical and biological sciences have been reported. This thesis concentrates on developing nonlinear stress- strain relations for geo-materials.

The main advantage of the Neural Network based Constitutive Models (NNCMs) are the following:

- Do not require a-priori stipulation of any mathematical framework
- Do not require checking for yielding, computation of flow vector, updating and reconstitution tangential stress integration matrix
- Do not require identification of any material parameters, although desired material parameters can be identified Shin & Pande (2003).

- NNCMs simply learn the complex relationship between stresses and strains from the physical test data presented to it for training.
- NNCMs enable us to develop FE programs with artificial intelligence.
- NNCMs can be trained from monitored data (stress and displacement profile) of structures.
- A triaxial test with glued platens is a structure (non-uniform stress & strain field) and it provides richer data along diverse stress paths.

3.3 WAYS OF USING

There are three different ways in which NNCMs can possibly be used by engineers with considerable advantage. These are:

1. Firstly, NNCMs can be developed for any material from the raw test data without invoking any constitutive theory Shin & Pande (2002, 2001). This approach has many advantages, the most important being that one does not necessarily have to identify material parameters of the model. However, if one does need to identify them in order to have the knowledge of the conventional engineering parameters, they can be identified by carrying out what is known as 'virtual tests'.

2. Secondly, NNCMs can be trained from incremental load and displacement data of 'structures'. Here, by the term 'structures', we mean solids of arbitrary shape subjected to monotonically increasing loads having a non-uniform states of strain and stress. Thus, a cylindrical specimen of a geo-material having glued rigid platens, subjected to uniaxial load is a structure. This application of NNCMs leads to 'intelligent finite elements' as described by Shin & Pande (2001a) and is available for condition monitoring of real structures. It can also be used to identify material parameters for complex materials such as masonry from structural tests, Shin & Pande (2003).

3. Thirdly, since many constitutive models are very complex, incorporating them in a finite element code and using them for solving real life problems may not be a trivial task. Here, using 'synthetic data' generated from systematic exploration of

strain space and corresponding stress response, a NNCM can be trained and plugged in a Finite Element (FE) code. This will certainly leads to computational efficiency since the computation of stress increment for a strain increment from a trained NNCM is almost instantaneous. Whilst the first two categories of applications have been reported by a number of researchers, there appear to have been no applications in this category.

This thesis belongs to the third category of applications. The objective is to demonstrate how synthetic data from any constitutive model can be used to successfully train a NNCM. We choose two constitutive models for illustration. The first is a well-known model viz. the Hardening Soil Model (HSM) available in the commercial code PLAXIS. This model uses parameters obtained from established engineering practice and is suitable for situations of monotonic loading. The second model is a complex Two-surface Deviatoric Hardening Model (TDHM) in the multilaminate framework for cyclic loading developed by Lee and Pande (2004).

3.4. IMPLEMENTATION CONSTITUTIVE MODELS IN NNCM

3.4.1 METHODOLOGY AND ARCHITECTURE

ANNs can be used to simulate stress-strain response of any material by using appropriate data. In this case, the components of strain rates become the causes whilst the resulting stress rates are the effects. Description of stress-strain behaviour by a NNCM does not require checking for plastic flow, computation of flow vector, updating and reconstitution tangential stress integration matrix. Full details are contained in Shin & Pande (2001, 2002 and 2003). Here a brief description is given for completeness and continuity.

Incremental (as distinct from total) stress vector can be computed from the corresponding incremental strain vector as follow:

$$d\sigma = \text{NNCM}(d\varepsilon) + (\sigma)_{n-1} \qquad (3.4.1)$$

Where:

$$d\varepsilon=\{d\varepsilon_x, d\varepsilon_y, d\varepsilon_z, d\gamma_{xy}, d\gamma_{yz}, d\gamma_{xz}\} \quad (3.4.2)$$

$$d\sigma=\{d\sigma_x, d\sigma_y, d\sigma_z, d\tau_{xy}, d\tau_{yz}, d\tau_{xz}\} \quad (3.4.3a)$$

$$(\sigma)_{n-1} : \text{current components of stress } (\sigma) \quad (3.4.3b)$$

Here, in the terminology of ANNs, NNCM stands for a neural network based constitutive model trained from appropriate incremental stress-strain data at various strain/stress levels. Thus, the input parameters must include current components of stress (σ) and increments of strain ($d\varepsilon$) whilst the required outputs are increment of stress ($d\sigma$). In many geotechnical problems, soils are subjected to cyclic and transient loads. Even in quasi-static problems many elements are subjected to unloading and reloading. For constitutive models to be valid, in such situations, it is proposed to adopt 'intrinsic time' (Bazant, 1982, Valanis, 1982) or the current length of strain trajectory, ξ, as an independent input parameter. Mathematically, intrinsic time, ξ, is defined as follows:

$$\xi = \int d\xi \quad (3.4.4)$$

Where, $d\xi$ is an increment of deviatoric strain defined as

Where, $d\xi$ is an increment of deviatoric strain defined as

$$d\xi = \| d\varepsilon_{ij} - \tfrac{1}{3}\delta_{ij}\varepsilon_{kk} \| \quad (3.4.5)$$

ξ, is a monotonically increasing positive parameter. The above definition can be changed to include volumetric strain as well as real time as is the case in 'endrochronic' theories of constitutive behaviour.

Keeping the above points in mind, the NNCM adopted in this thesis has strain increments, stresses and 'intrinsic time' are the input variables whilst increments of stress are the output variables. For two-dimensional analyses, the workable architecture of the NN has been obtained by trial and error and is presented in Figure 3.1. It is constituted by 9 input nodes two hidden layer of 18 and 8 nodes respectively and 4 output nodes. In order to train the neural network the resilient back-propagation (RPROP) algorithm is used of GDAP program Shin (2001). RPROP was first proposed by Reidmiller (1993)

and it is a local adaptive learning scheme based on the standard back-propagation framework.

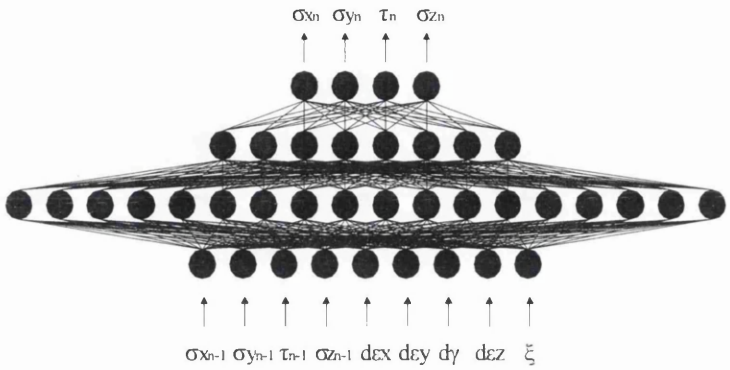


Figure 3.1: Architecture of NNCM [9-18-8-4] for two-dimensional analysis

3.4.2 DATA ENRICHMENT

The strain-stress pairs from the triaxial tests are actually principal stresses and strains since no shear stresses/strains are involved. If such data were used for training, NNCM would have to extrapolate wherever shear stress/strain components are involved. This would lead to large inaccuracies in stress-strain response of the NNCM.

To overcome this limitation, Shin & Pande (2002) proposed a data enrichment strategy. They created additional data by transformation of stress and strain (figure 3.2) in which the shear components will naturally be non-zero. In the two dimensional case, the transformation of a principal stress vector by an angle θ measured anticlockwise from the X axis is as follows:

$$\sigma_x = A + B\cos(2\theta) \tag{3.4.6}$$

$$\sigma_y = A - B\cos(2\theta) \tag{3.4.7}$$

$$\tau_{xy} = B\sin(2\theta) \tag{3.4.8}$$

Where:

$$A = 0.5(\sigma_1 + \sigma_2) \tag{3.4.9}$$

$$B=0.5(\sigma_1-\sigma_2) \tag{3.4.10}$$

For strains

$$\varepsilon_x=C+D\cos(2\theta) \tag{3.4.11}$$

$$\varepsilon_y=C-D\cos(2\theta) \tag{3.4.12}$$

$$\gamma_{xy}=2D\sin(2\theta) \tag{3.4.13}$$

Where:

$$C=0.5(\varepsilon_1+\varepsilon_2) \tag{3.4.14}$$

$$D=0.5(\varepsilon_1-\varepsilon_2) \tag{3.4.15}$$

This method produces a large amount of training data depending on the number of transformations chosen to generate data. Among the expanded data there are many duplicated strain-stress pairs so an additional process of 'data pruning' is adopted. A special algorithm has been developed in this thesis in order to apply the strategy and an incremental angle, $\Delta\theta$ equal to 5° was used in order to rotate the strain-stress axes from -45° to + 45°.

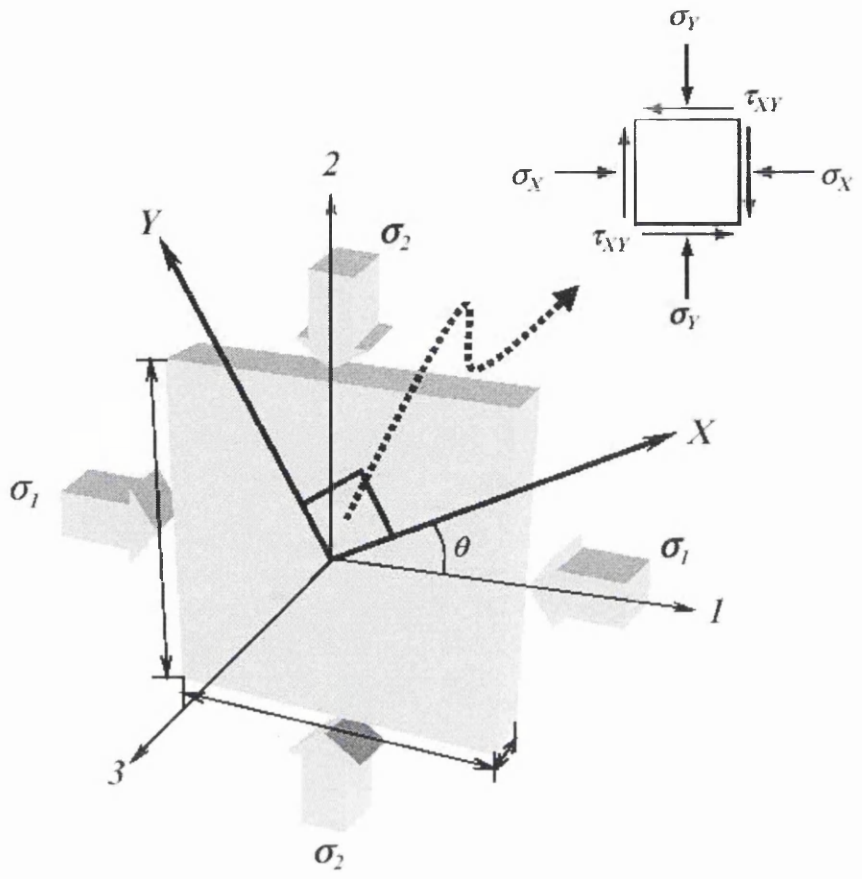


Figure 3.2: Transformation of stress components in two-dimensional domain.
(Shin 2001c)

CHAPTER 4. IMPLEMENTATION OF HARDENING SOIL MODEL IN A NNCM

4.1 INTRODUCTION

At the current paragraph the objective is to demonstrate how synthetic data from the Hardening Soil model can be used in order to train a NNCM. According that validation test has been carried out in order to check the accuracy of the NNCM using the strains results of a conventional Finite elements analysis. Two boundary problems have been solved and the results were compared with the original constitutive soil model.

4.2 HARDENING SOIL MODEL (HSM)

The Hardening Soil Model is described in PLAXIS^{TM} Manual is a nonlinear elastic-plastic model with Mohr Coulomb failure criterion. It is an enhanced version of the nonlinear elastic hyperbolic model of Duncan & Chang (1970) with deviatoric hardening operating on Mohr Coulomb yield surface. A non-associated flow rule defined by a dilatancy angle smaller than the peak friction angle is adopted. It is generally applicable to loose to medium dense sands and normally consolidated to lightly overconsolidated soils. The model captures apparently strong nonlinearity prior to failure, which is a drawback of conventional linear elastic-plastic models. A cap in deviatoric stress – mean effective stress space is also included. Figures 4.1, 4.2 and Table 4.1 illustrate the HS model.

Table 4.1.Description of Input Parameter for HS Model

Stress dependent stiffness according to power law	m
Plastic straining due to primary deviatoric loading	E_{50}^{ref}
Plastic straining due to primary compression	E_{oed}^{ref}
Elastic unloading/reloading	E_{ur}^{ref} , v_{ur}
Failure according to Mohr-Coulomb model	c, φ,ψ

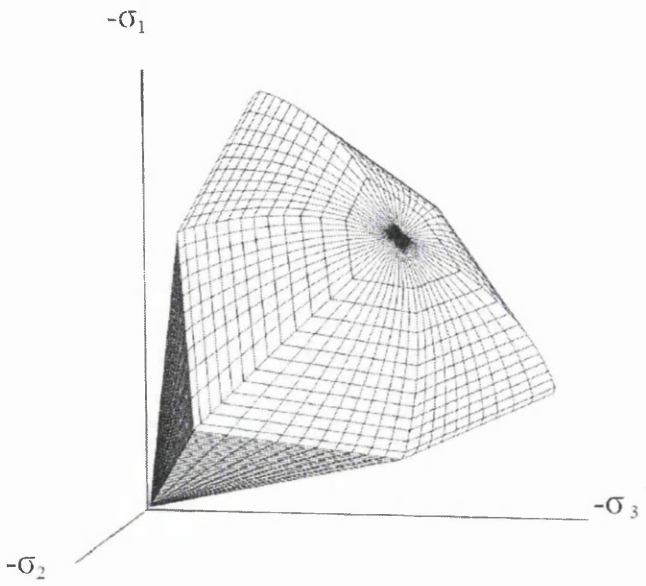


Figure 4.1 HS model in stress space (Plaxis Manual)

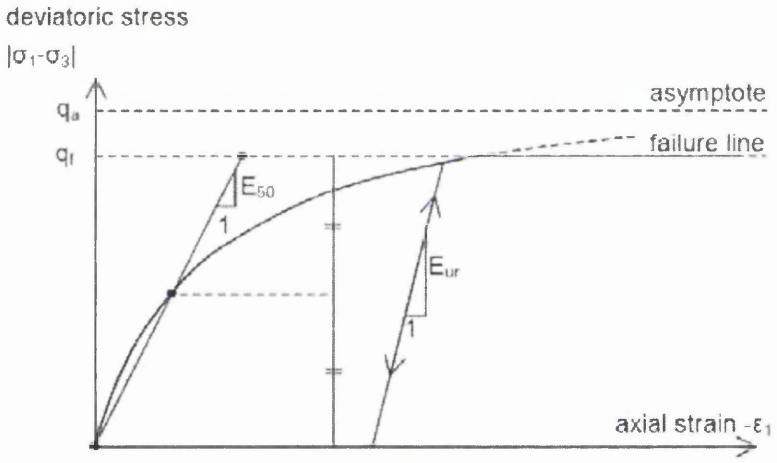


Figure 4.2 Hyperbolic law for deviatoric stress and axial strain in HS Model (Plaxis Manual)

4.3 SYNTHETIC DATA GENERATION

For heuristic purpose, we have chosen typical parameters for medium dense as given in Table 4.2. Though many parameters will be familiar to engineers, reader should refer to the PLAXIS^{TM} software manuals for full details.

Table4.2. Chosen parameters for sand for HSM

E_{50}^{ref} (for p_{ref}=100kPa), kPa	20000
E_{ur}^{ref} (for p_{ref}=100 kPa), kPa	60000
E_{oed}^{ref} (for p_{ref}=100 kPa), kPa	20000
Cohesion c′, kPa	0.0
Friction angle φ′, degrees	30
Dilatancy angle ψ, degrees	0
Poisson ratio v	0.2
Power m	0.5
K_o^{nc}	0.5
Tensile strength, kPa	0
Failure ratio	0.9

The data in Table 4.2 are used to generate stress-strain response of the sand under various experimental configurations, viz. Triaxial Loading in Compression (LC), Triaxial Loading in Extension (LE), Unloading in Compression (UC) and Unloading in Extension (UE). Data have been generated for stress controlled drained conditions. A single finite element subjected to uniform stress conditions was used in analysis with PLAXIS^{TM} software (figure) with HSM model with parameters given in Table 4.2. The left hand side and the bottom of the geometry are axes of symmetry. At these boundaries the displacements normal to the boundary are fixed and the tangential displacements are kept free to allow for 'smooth' movements (figure 4.3). The remaining boundaries are fully free to move. The stress paths, in deviatoric stress, q and effective mean stress, p', space are shown in figure 4.4.

Three different confining pressures of 50 kPa, 100 kPa and 150 kPa were used for each of the above stress paths. In engineering practice, if such test data were available for a soil, one would perhaps term them as 'extensive'. These stress-strain data obtained from these simulations were then used for training the NNCM.

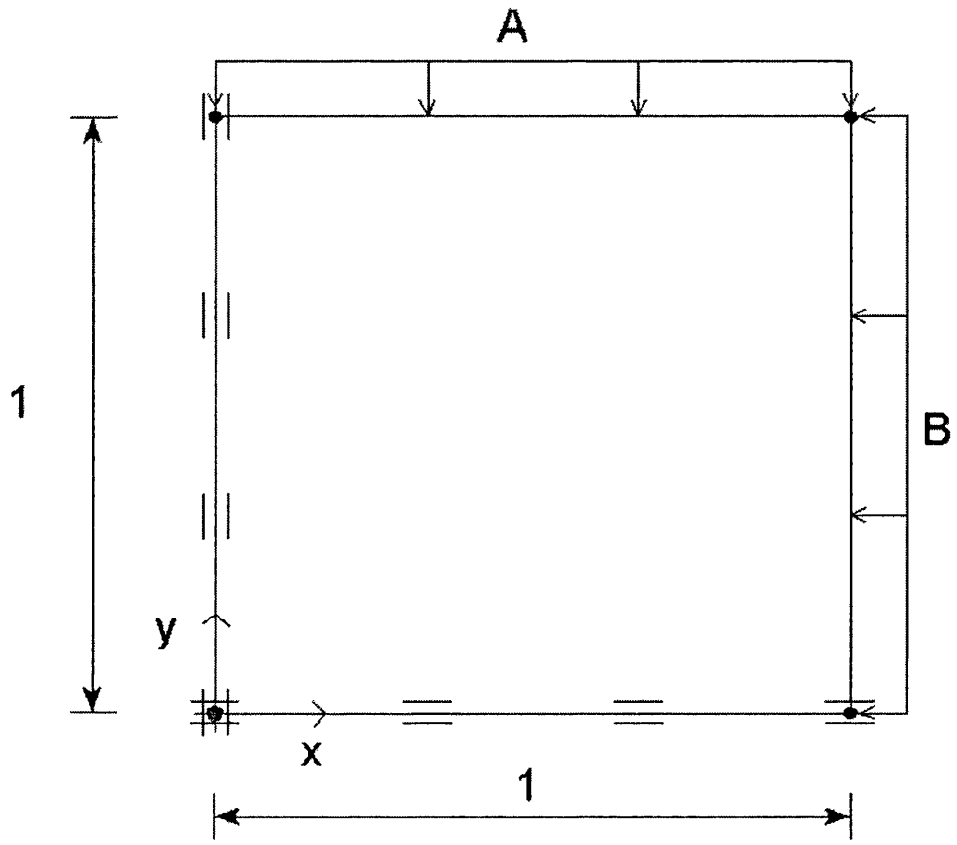


Figure 4.3 Simplified configuration of triaxial test. (Plaxis Manual)

Plots of q versus axial strain, ε_{yy}, as predicted by the trained NNCM and the original HSM data used for training, for various stress paths are shown in Figures 4.5 - 4.8. In figures 4.5 and 4.6 the results from the analysis are presented without considering the strain trajectory. Poor accuracy was presented in that case. As following the strain trajectory has been included in the training data and an excellent match is seen confirming that NNCM has been adequately trained. This, however, is not surprising since prediction of response is made for the confining pressure which was also used in training data. We need to check if the response at confining pressures within the training range as well as outside

that used in training is also reasonable. This issue is discussed in the next paragraph of the thesis.

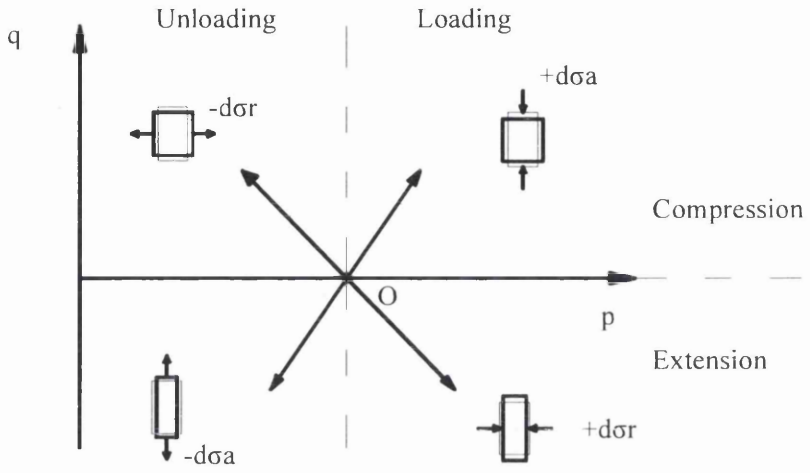


Figure 4.4: Stress paths in q – p' space used for generating synthetic data.

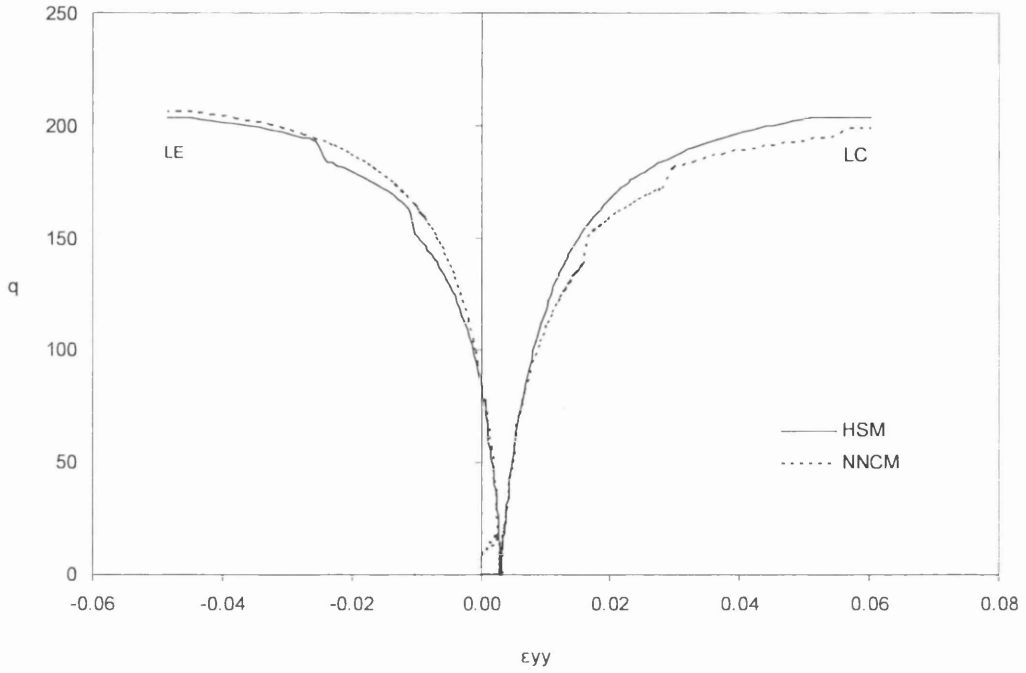


Figure 4.5: Graph of q versus ε_{yy} under LC and LE conditions for a confining pressure of 100 kPa. (No strain trajectory is considered)

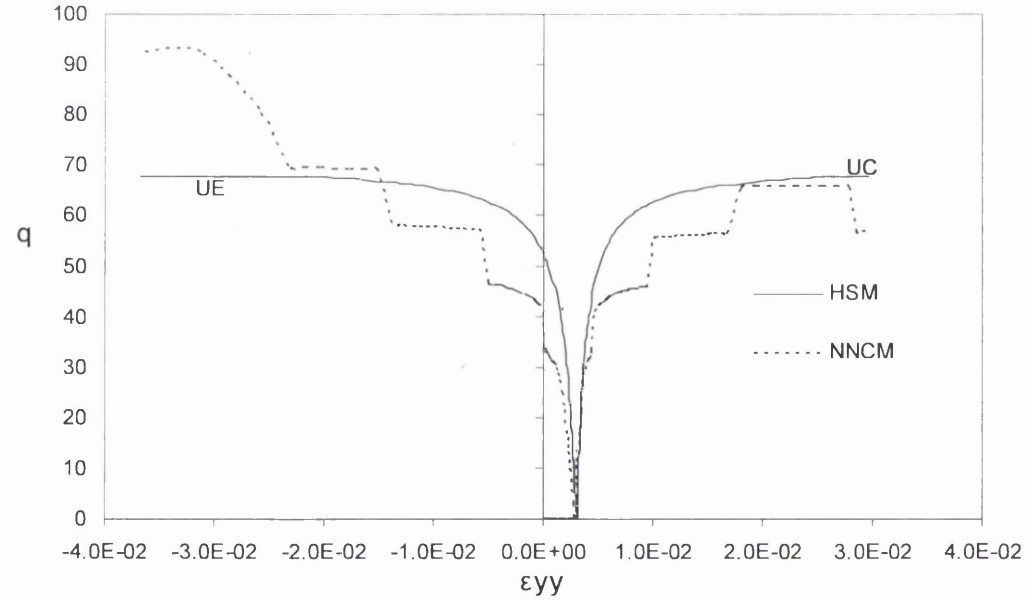


Figure 4.6: Graph of q versus ε_{yy} under UC and UE conditions for a confining pressure of 100 kPa. (No strain trajectory is considered)

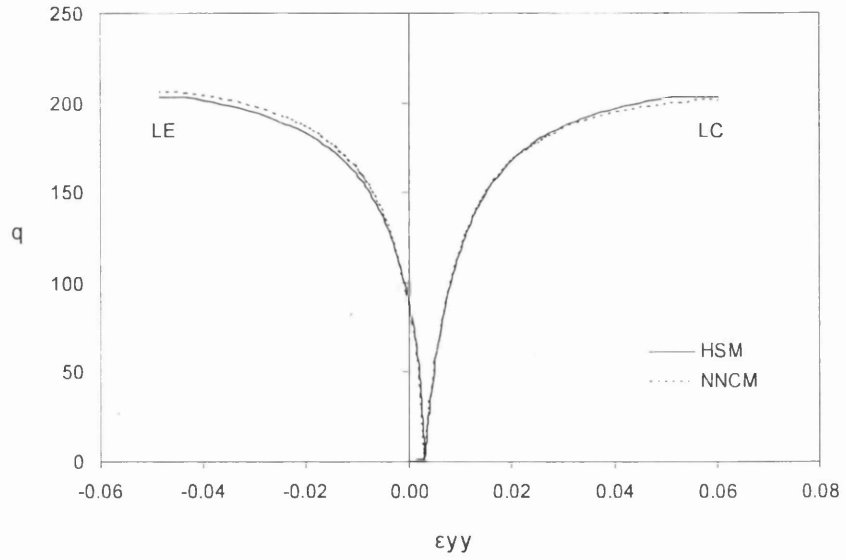


Figure 4.7: Graph of q versus ε_{yy} under LC and LE conditions for a confining pressure of 100 kPa.

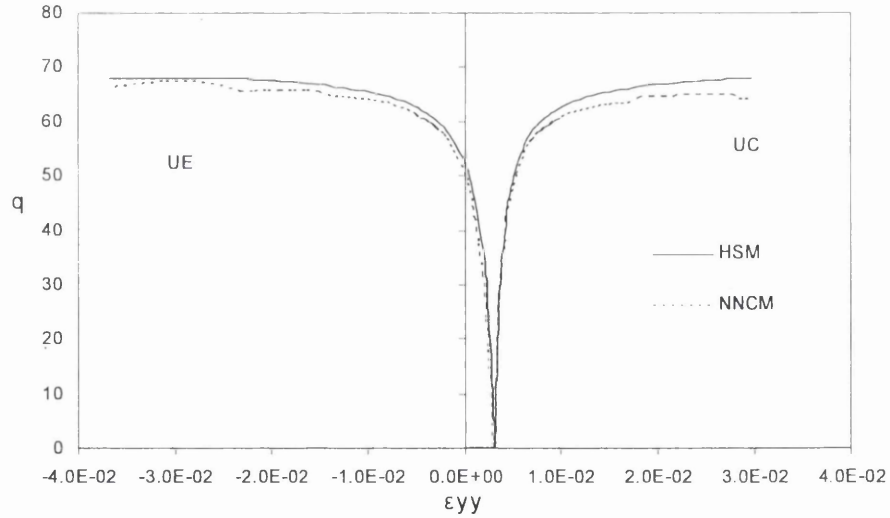


Figure 4. 8: Graph of q versus ε_{yy} under UC and UE conditions for a confining pressure of 100 kPa.

4.4 MODEL VALIDATION.

In this part of the thesis, we present the prediction of the trained NNCM for two additional triaxial tests with confining pressures of 10 kPa & 200 kPa which are out of the range of the training data. For these tests the predictions of the NNCM were poor (figures 4.9 and 4.10). NNCM was then re-trained using additional data generated for the four stress paths described earlier and for confining pressures of 1 kPa and 200 kPa. The graphs of q versus ε_{yy} are presented in the figures 4.9 and 4.10 for the HSM, trained and re-trained NNCMs. This confirms that extrapolation by NNCM is always of poorer quality than interpolation, Pande & Shin (2004).

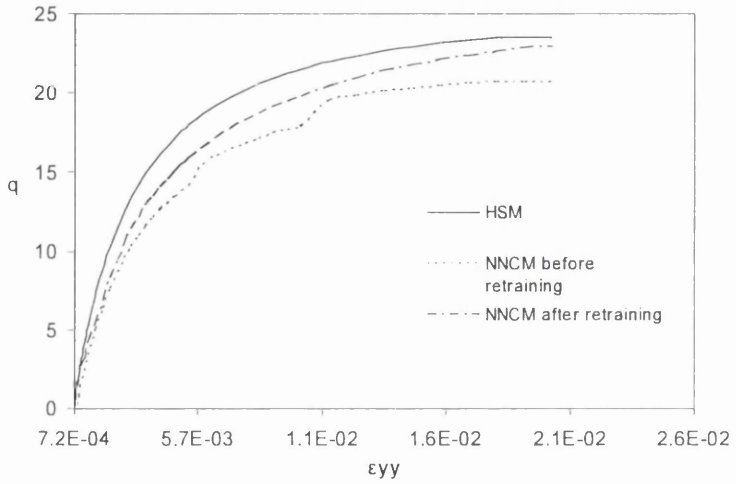


Figure 4.9: Graph of q versus ε_{yy} under LC conditions for a confining pressure of 10 kPa.

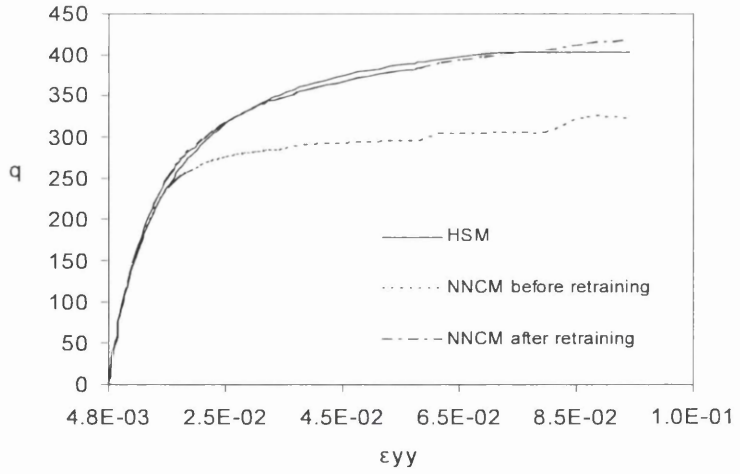


Figure 4.10:Graph of q versus ε_{yy} under LC conditions for a confining pressure of 200 kPa.

4.5 STRESS PATHS IN BOUNDARY VALUE PROBLEMS

The stress paths imposed in real problems are obviously different from those under the mentioned four stress paths. The aim of the following paragraphs is to compare NNMC and HSM predictions in typical geotechnical problems. For this purpose, we have chosen two examples. In the first example a foundation problem has been considered whilst in the second a problem of excavation has been studied. The soil parameters for a both problems are taken from table 4.2. PLAXIS FE code has been used in order to solve the above geotechnical problems. Plane strain and drained condition was assumed for the calculations.

4.5.1 FOUNDATION PROBLEM

In Figure 4.11 the geometry of the foundation problem is presented. Six noded triangular isoparametric elements have been used for the analysis. A uniform load of 150 kPa has been incrementally applied and the resulting strain – stress curves at two monitoring points, A & B, marked on the figure, for NNCM and HSM are shown in Figs. 4.12 and 4.13.

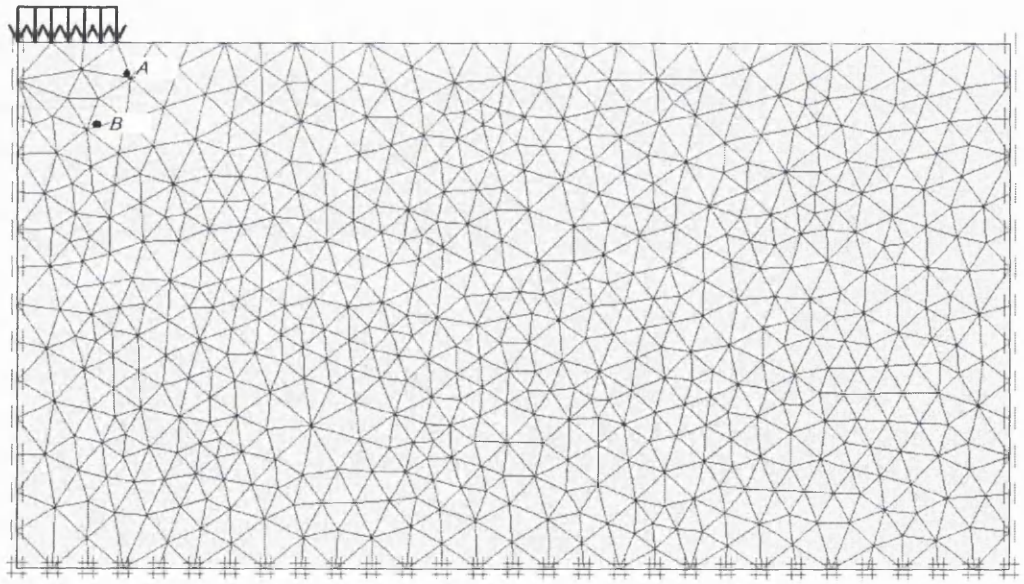
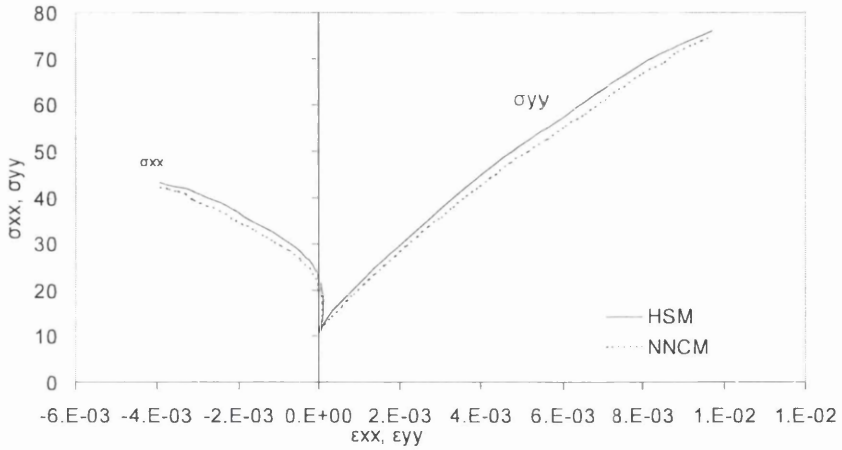
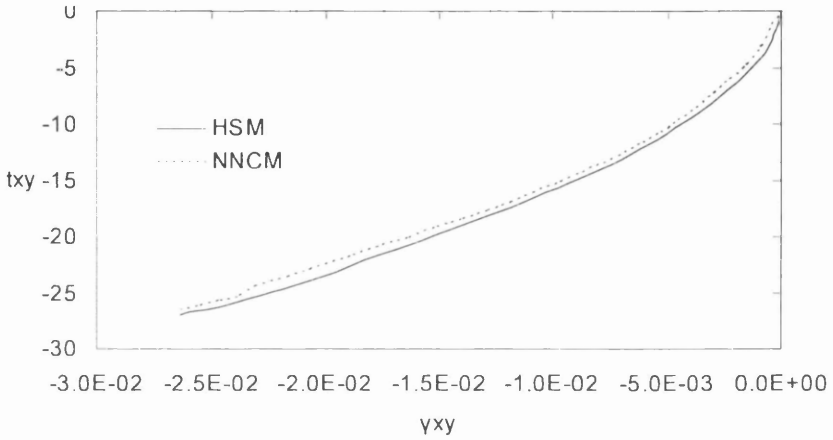


Figure 4.11: Geometry and monitoring points of the foundation problem





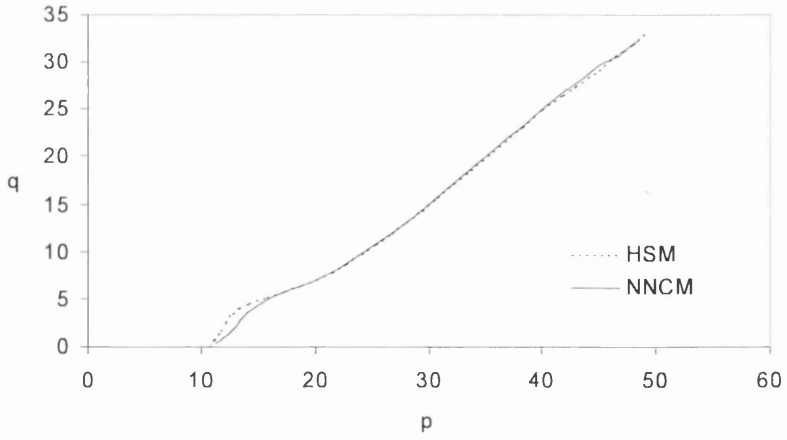
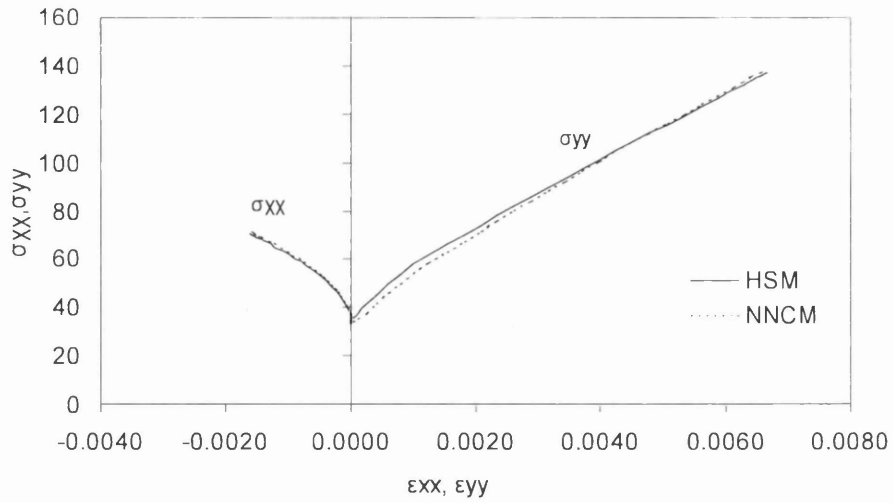
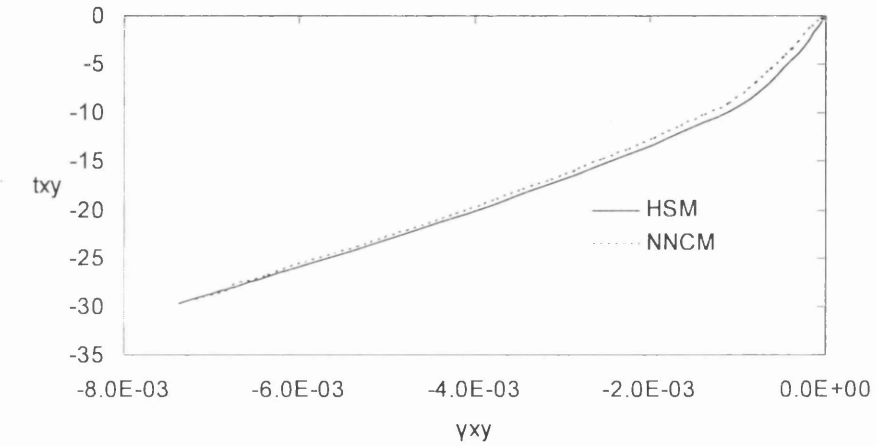


Figure 4.12: Various stress paths at point A in foundation problem





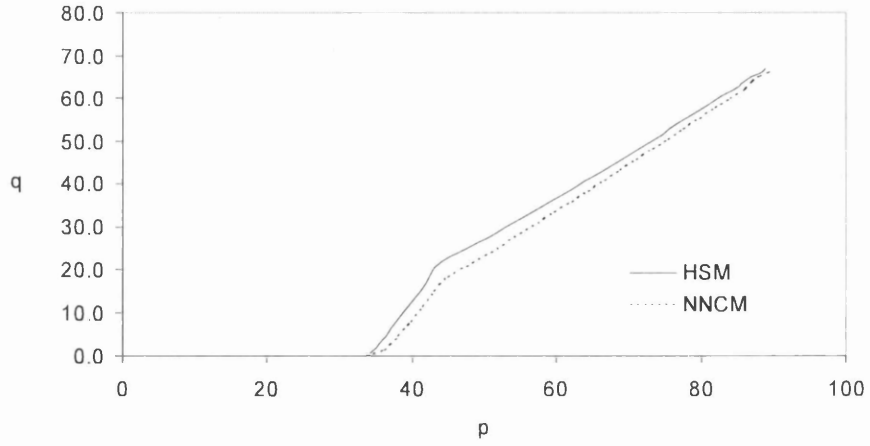


Figure 4.13: Various stress paths at point B in foundation problem

4.5.2 EXCAVATION PROBLEM

In this example, analysis of a 10 m x 4 m deep excavation behind a sheet piled wall was carried out in one stage. After the calculation for the initial conditions of the in situ stress due to the gravity load the excavation load is applied. The geometry of the problem and the locations of monitoring points are presented in figure 4.14. As mentioned earlier, three confinement pressures were initially used to generate data for the training of the NNCM. If this NNCM is used for the prediction of stresses corresponding to the strains at monitoring point A, predictions were found be inaccurate. However, when the NNCM trained with five sets of confining pressure data was used, accurate response corresponding closely to HSM was obtained, see Fig. 4.15 & Fig. 4.16. In the excavation problem, at some points, the confining pressure is small. The initially trained NNCM had to extrapolate the response at low confining pressures whilst after re-training with five sets of pre-consolidation pressures, it had only interpolate or extrapolate only in a small zone beyond the data used in its training.

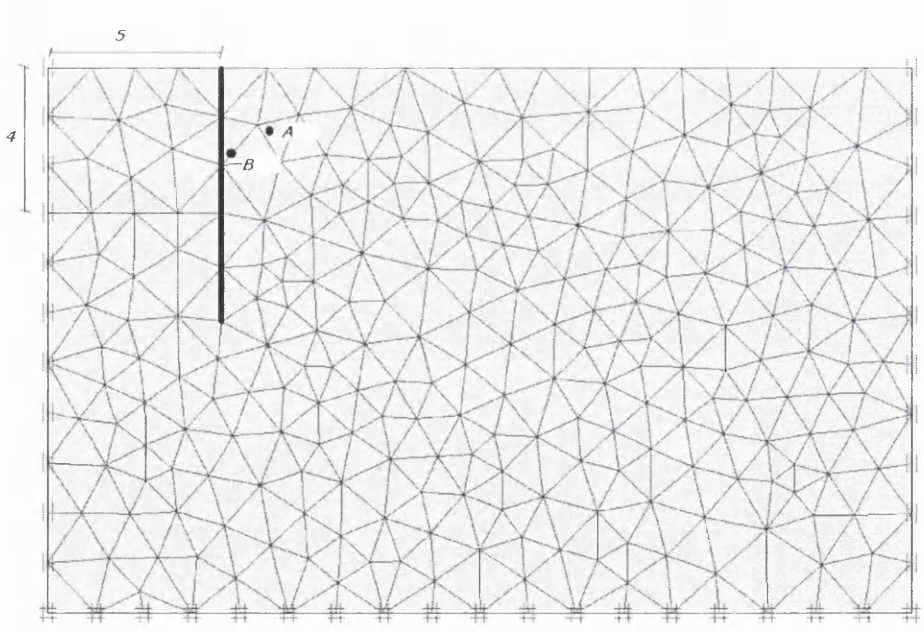


Figure 4.14: Geometry and monitoring points of the retaining problem

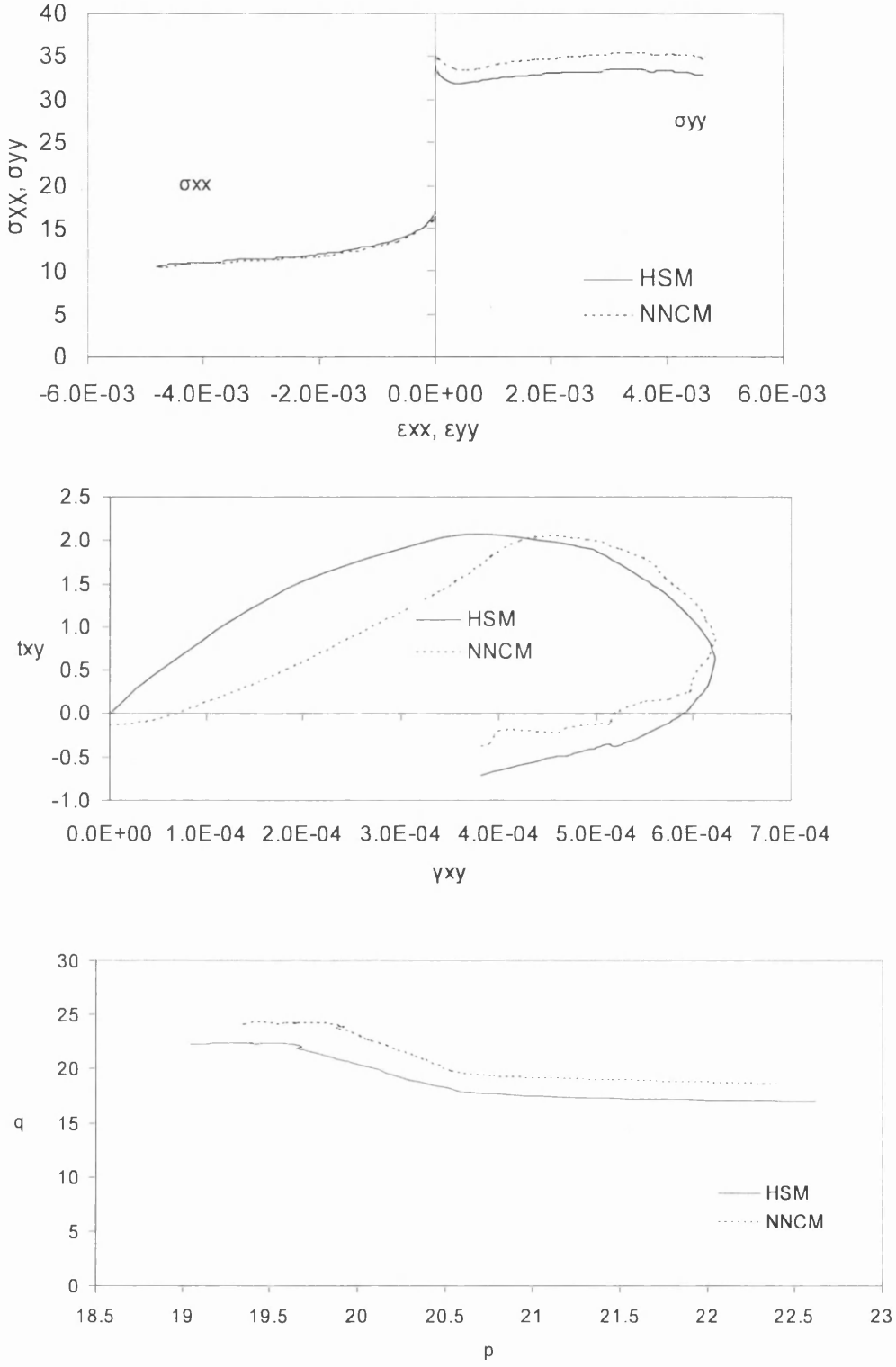


Figure 4.15: Various stress paths at point A in excavation problem

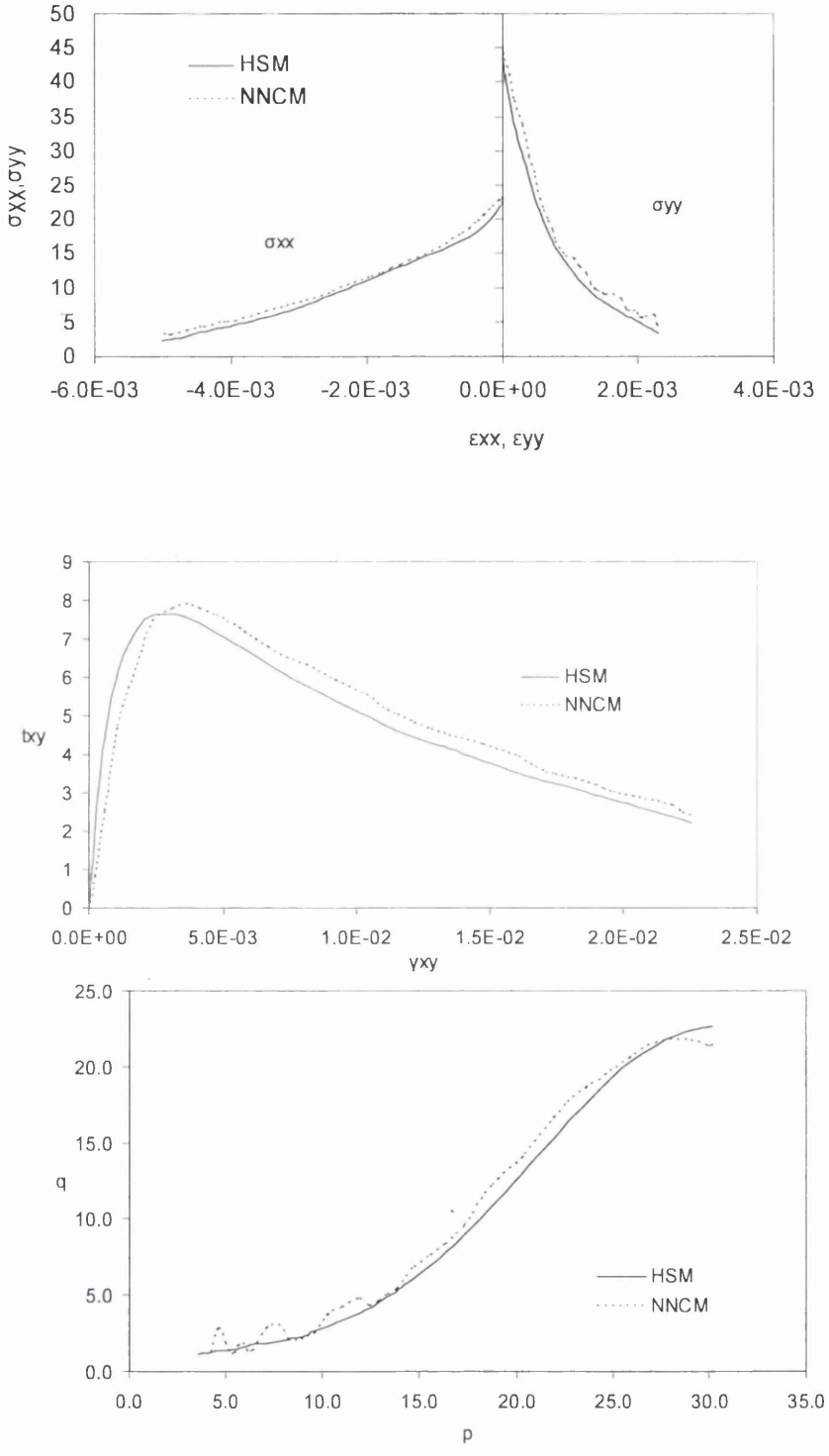


Figure 4.16: Various stress paths at point B in excavation problem

5. IMPLEMENTATION OF TWO-SURFACE MODEL IN MULTILAMINATE FRAMEWORK (TDHM) IN NNCM

5.1 INTRODUCTION

A multilaminate framework for modelling the behaviour of soils was presented by Pande & Sharma [1983] almost two decades ago. Yield and failure criterion for most elasto-plasticity based models are written terms of stress invariants. This type of formulation inhibits development of plastic strains purely due to rotation of principal stresses. Such formulations also exclude plastic flow induced anisotropy. Multilaminate formulation of plasticity based models overcomes both the above mentioned short comings. Here, the yield and failure criteria as well as plastic potential and hardening softening functions are written in shear stress/ normal stress space for randomly oriented micro-planes. A framework similar to multilaminate framework was presented by Bazant under the name 'micro-plane model'. Practical applications of the multilaminate framework have been carried out by Schweiger & his co-workers (2000). In the following, a brief description of the formulation of the new two-surface model in multilaminate framework is presented (Lee & Pande (2004)) which is formulated for cyclic and transient loading. Some details of multilaminate formulation are given for completeness and continuity. It is noted that the objective here is to develop an NNCM equivalent of this complex model.

5.2 MATHEMATICAL FORMULATION

Let us adopt a set of local co-ordinate axes (n, s, t) for each sampling plane. The n-axis is normal to the sampling plane whilst axes s and t are arbitrarily chosen on the sampling plane as shown in Figure 5.1 below

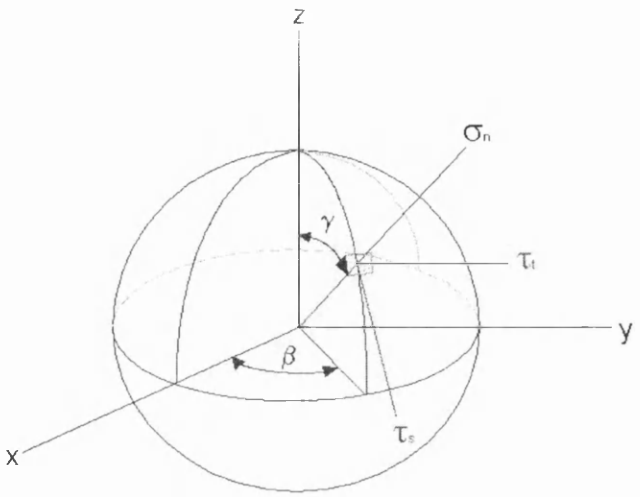


Figure 5.1: Definition of a local system of axes on a typical sampling plane

In the local system of co-ordinates, the normal and shear stress components can be defined as follows :

$$\sigma = \begin{Bmatrix} \sigma'_n \\ \tau_s \\ \tau_n \end{Bmatrix} = [\Gamma] \begin{Bmatrix} px \\ py \\ pz \end{Bmatrix} \tag{5.1}$$

$$\begin{Bmatrix} px \\ py \\ pz \end{Bmatrix} = \begin{Bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{Bmatrix} \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} \tag{5.2}$$

$$\tau = \sqrt{\tau_s^2 + \tau_t^2} \tag{5.3}$$

Where

$$T = \begin{Bmatrix} n_i \\ s_i \\ t_i \end{Bmatrix} = \begin{bmatrix} \sin\gamma\cos\beta & \sin\gamma\sin\beta & \cos\gamma \\ \cos\gamma\cos\beta & \cos\gamma\sin\beta & -\sin\gamma \\ -\sin\beta & \cos\beta & 0 \end{bmatrix} \tag{5.4}$$

If the yield and bounding surfaces for frictional materials are plotted in resultant $\tau - \sigma_n$' (all stresses are effective and prime on the symbols will be omitted henceforth) space, they are represented by a pair of straight lines. The angle between the pair of lines representing elastic domain is assumed as arbitrarily small while the angle between the lines representing bounding surface is related to the friction peak angle φ'. However, when two orthogonal components of shear stress on a sampling plane are considered, the yield and bounding surfaces are represented by two cones in $\tau_s - \tau_t - \sigma_n$ space, the smaller one gyrating inside the larger one, see Fig 5.2.

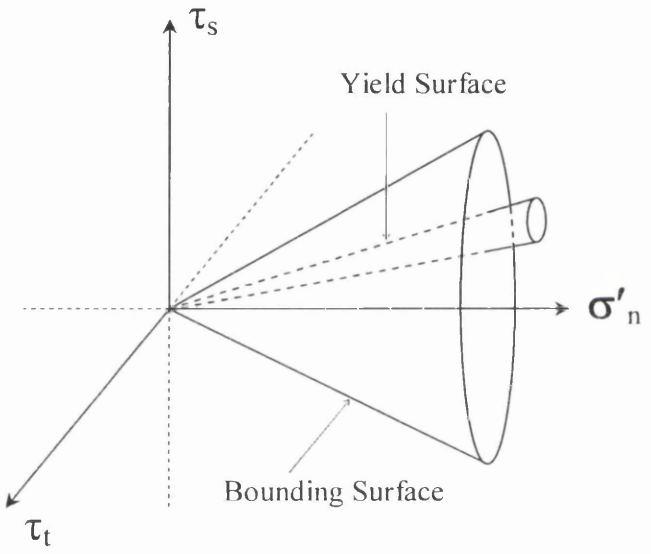


Figure 5.2: Yield and bounding surfaces for a sampling plane in σ'_n - τ_s - τ_t space.
(Lee. K and Pande, G. N. , 2004)

5.3 EQUATIONS OF YIELD AND BOUNDING SURFACES

The equation of the bounding surface is postulated below:

$$F = \tau + \eta\sigma_n^{'} = 0 \Rightarrow F = \sqrt{\tau_s^2 + \tau_t^2} + \eta\sigma_n^{'} = 0 \qquad (5.5)$$

And

$$\eta(\underline{\gamma}^{p}) = \eta_f \frac{\underline{\gamma}^{p}}{A + \underline{\gamma}^{p}} \tag{5.6}$$

The yield surface is represented by the following equation:

$$f = \tau - \eta_l \sigma_n^{'} = 0 \Rightarrow f = \sqrt{\tau_s^2 + \tau_t^2} + \eta_l \sigma_n^{'} = 0 \tag{5.7}$$

Plastic potential surface is given by:

$$\Psi = \tau - \eta_c \sigma_n^{'} \ln\left(\frac{\sigma_n^{'}}{\sigma_{no}^{'}}\right) = \text{const} \tag{5.8}$$

Where,

η_l : is the size of the yield surface, $\eta_l \ll \eta_f$= constant

n_c: is the size of plastic potential surface

And

$$\gamma^{p} = \gamma^{p} - \gamma_0^{p} \tag{5.9}$$

γ^{p} and γ_0^{p} are current and initial values of plastic shear strain on i_{th} plane. It must be noted that at first loading γ_0^{p} is equal to zero and its value is renewed at each change of load increment sign. In the above, A is a soil parameter (positive value), $n_f = \tan\varphi_f$ (φ_f : peak internal friction angle) with

$$\phi_f = \sin^{-1}\left(\frac{3\eta_f}{\eta_f + 2}\right); \text{ the ultimate friction angle} \tag{5.10}$$

And

$$\gamma^{p} = \frac{An}{n_f - n} \tag{5.11}$$

5.4 TRAINING AN EQUIVALENT NNCM OF TDHM

TDHM is for use in the situations of transient and dynamic loading. Here computation of stress increment from stain increment is generally computationally intensive, especially for a complex model such as TDHM. In order to train the NNCM, strain-controlled triaxial test data conforming to seven different specified constant volumetric strains were generated using TDHM. The results of (TDHM) have been used to check the adequacy of the training of the equivalent NNCM. Additionally four new analyses which were not included in the training data have been used to compare and validate the NNCM equivalent to TDHM. The parameters of the analysis were the following:
n_f=0.52, n_l=0.02, n_c=0.43, Cohesion=0, shear modulus=40Mpa, A(soil parameter)=0.012. These parameters are described by Lee(2005).

5.5 SYNTHETIC DATA GENERATION

Synthetic data of strain and stress increments in 7 hypothetical two-way strain-controlled cyclic triaxial configuration ($d\varepsilon_{xx}=d\varepsilon_{zz}$), have been generated using a point integration program of TDHM (DRIVER), described in the previous paragraph. Various ratios of axial to radial strains have been chosen using the following strain paths trajectories:

Table 5.1. Training cases

Strain Path	Vertical strain increment ($d\varepsilon_{yy}$)	Ratio of vertical strain increment to radial strain increment ($d\varepsilon_{yy}/d\varepsilon_{xx}$)	Incremental volumetric strain imposed (%)
SSTD1	-1.75E-4	-2.5	-0.0035
SSTD2	-1.6E-4	-2	0
SSTD3	-1.35E-4	-1.5	0.0045
SSTD4	-9.0E-5	-1	0.0090
SSTD5	-7.0E-5	-0.5	0.013
SSTD6	-1.5E-4	-2.5	-0.003
SSTD7	-6.0E-5	-0.5	0.018

The NNCM was trained using the above data and the architecture shown in Fig. 3.1 and was used to predict the strain-stress response paths. Figs.5.3 - 5.11 show stress paths in q-p' space together with stress paths obtained from the original DRIVER program of TDHM. In figures 5.3 and 5.4 the results from the analysis are presented without considering the strain trajectory. Poor accuracy was presented again in that case as we show in the previous chapter. As following the strain trajectory has been included in the training data and the results were very accurate as we can see in the figures 5.5-5.11.

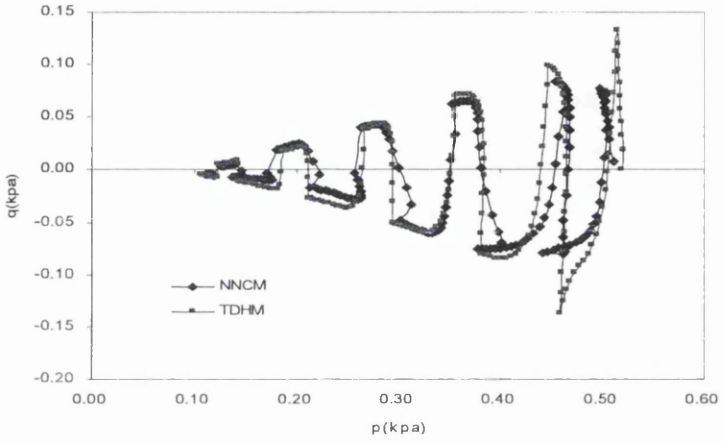


Fig. 5.3: Stress path in q-p' space for the test data SSTD1.
(No strain trajectory is considered)

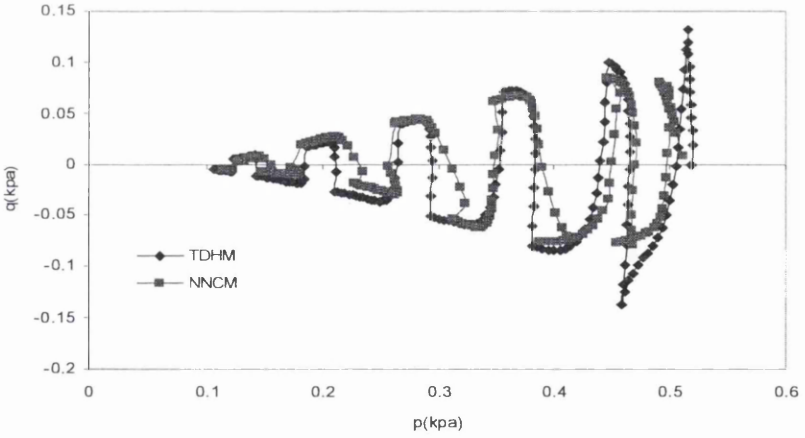


Fig. 5.4: Stress path in q-p' space for the test data SSTD2
(No strain trajectory is considered)

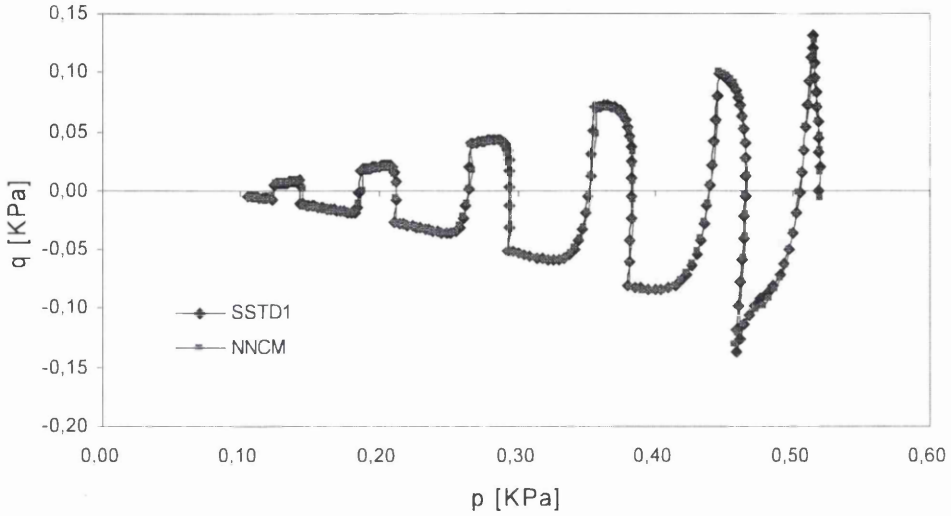


Fig. 5.5: Stress path in q-p' space for the test data SSTD1

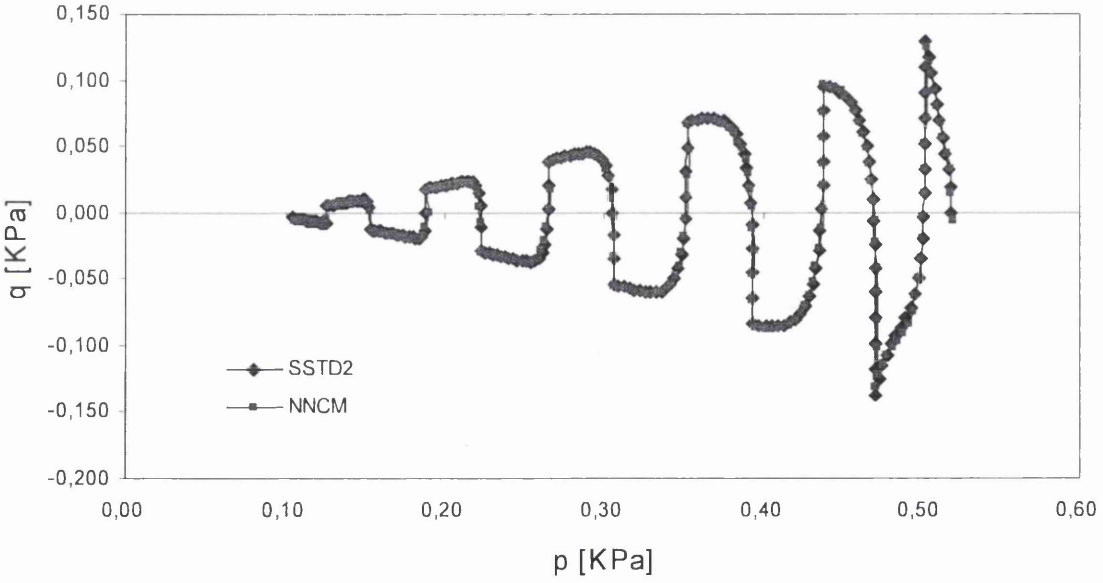


Fig. 5.6: Stress path in q-p' space for the test data SSTD2

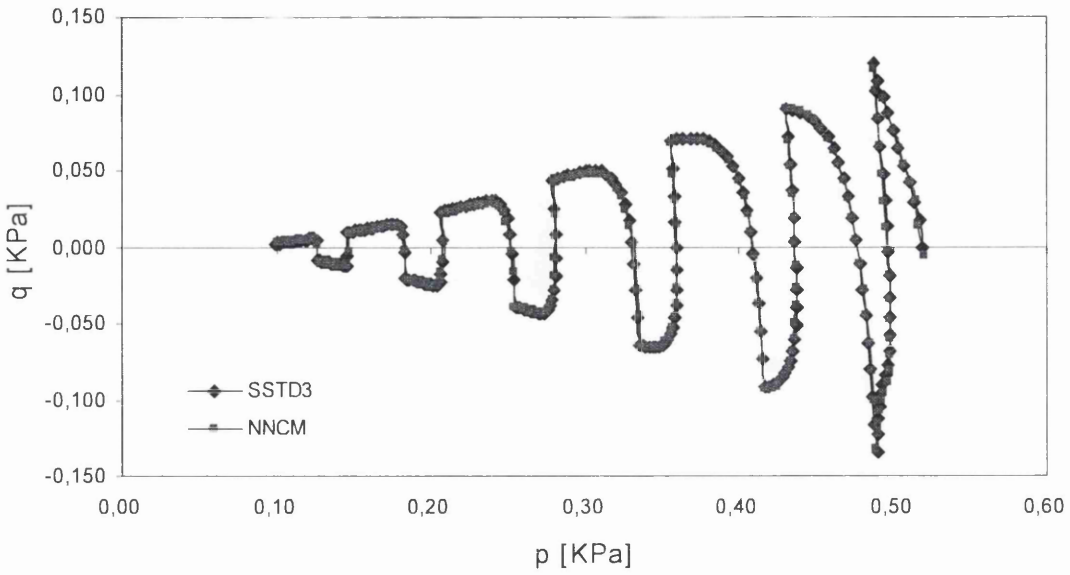


Fig. 5.7: Stress path in q-p' space for the test data SSTD3

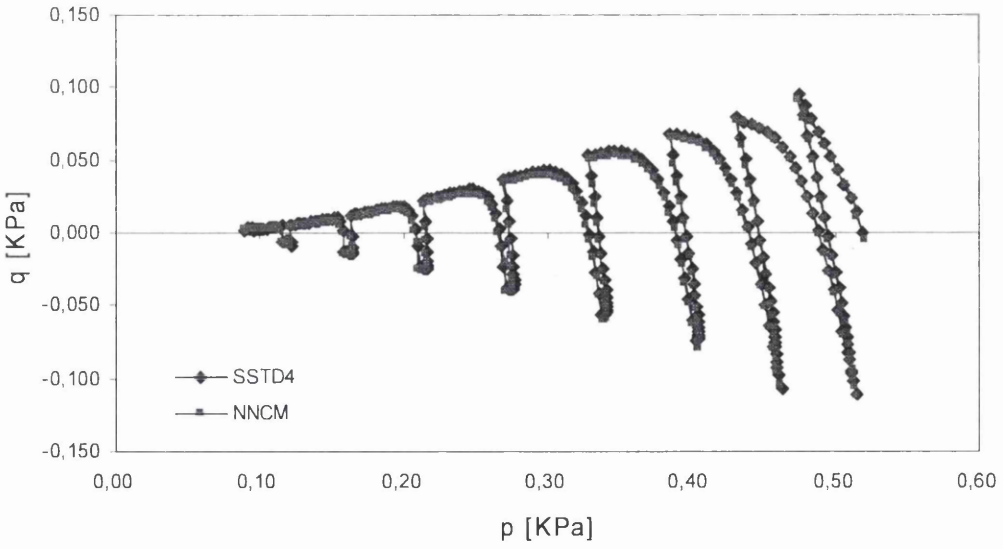


Fig. 5.8: Stress path in q-p' space for the test data SSTD4

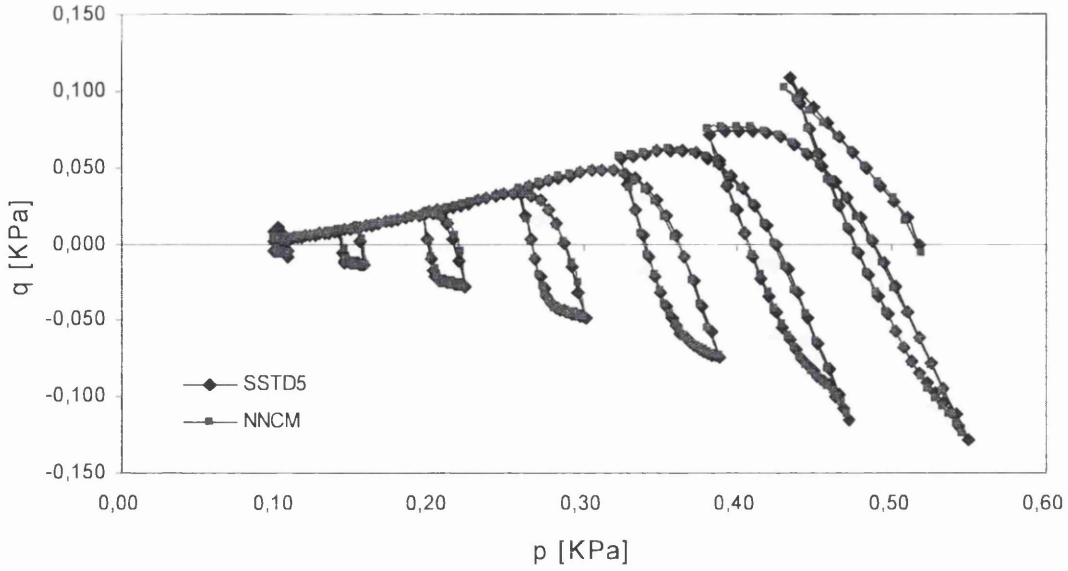


Fig. 5.9: Stress path in q-p' space for the test data SSTD5

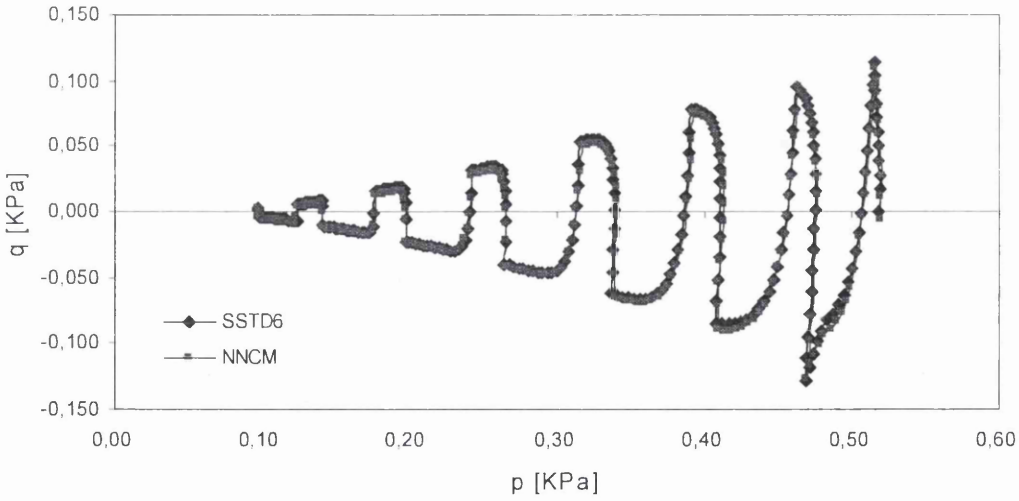


Fig. 5.10: Stress path in q-p' space for the test data SSTD6

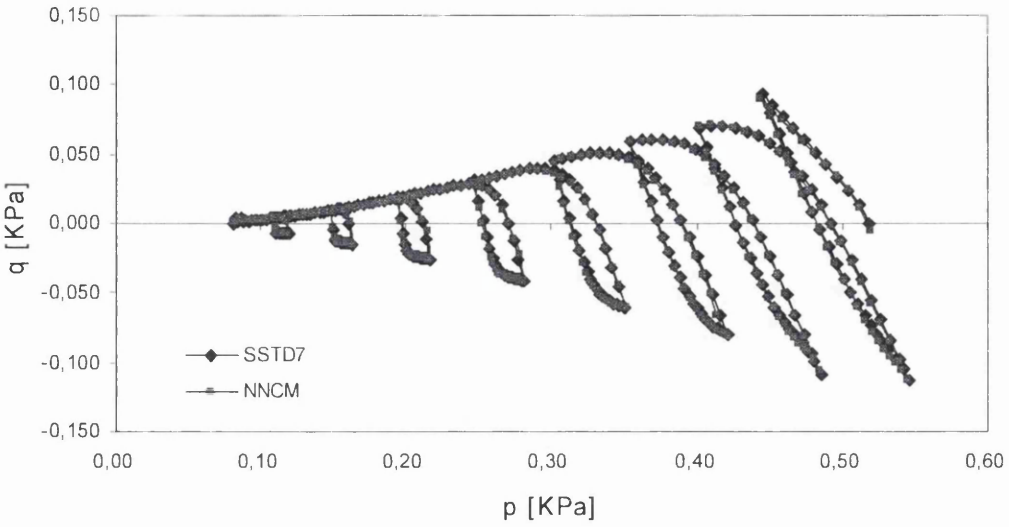


Fig. 5.11: Stress path in q-p' space for the test data SSTD7

It is obvious that the NNCM has learnt the strain-stress response with great accuracy. It may be noted that TDHM model is in a multilaminate framework where plastic strain contributions have to be computed from a large number of sampling planes.

5.6. VALIDATION OF THE TRAINED NNCM

In order to validate the model four new additional strain-stress data were generated using the DRIVER.

Table 5.1. Validation cases

Strain Path	Vertical strain increment ($d\varepsilon_{yy}$)	Ratio of vertical strain increment to radial strain increment ($d\varepsilon_{yy}/d\varepsilon_{xx}$)	Incremental volumetric strain imposed (%)
SSVD1	-1.5E-4	-2.3	-0.002
SSVD2	-1.4E-4	-1.6	0.0035
SSVD3	-1.08E-4	-1.2	0.0072
SSVD4	-0.91E-4	-0.7	0.0169

The results of the equivalent NNCM prediction are compared with the validation data in Figs. 5.12 – 5.15, where stress paths are plotted in q-p' space. A very good agreement is observed.

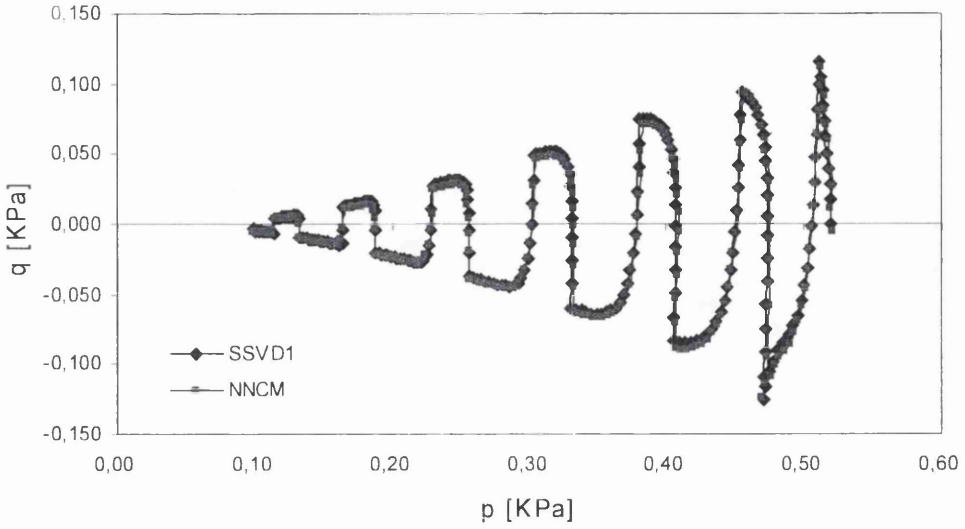


Fig. 5.12: Stress path in q-p' space for the test data SSVD1

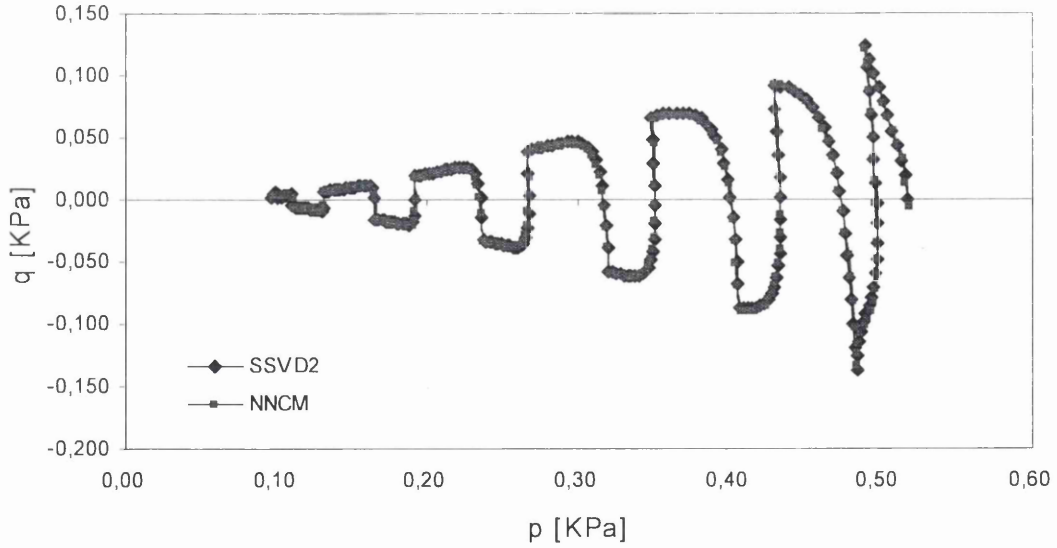


Fig. 5.13: Stress path in q-p' space for the test data SSVD2

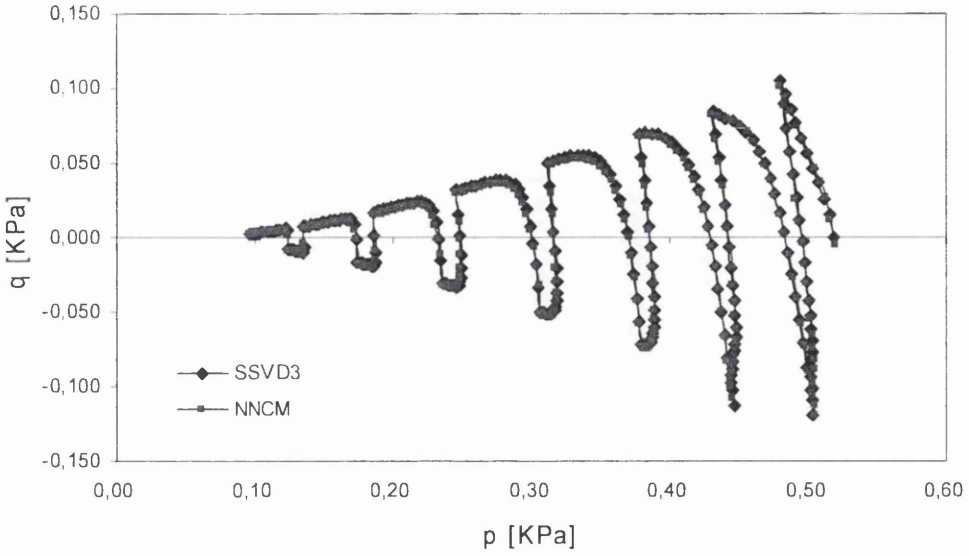


Fig. 5.14: Stress path in q-p' space for the test data SSVD3

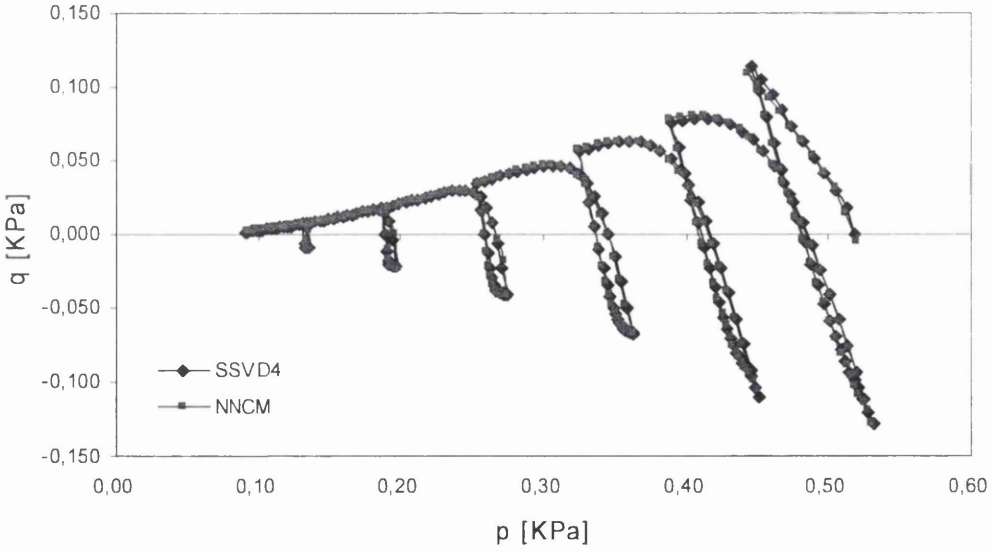


Fig. 5.15: Stress path in q-p' space for the test data SSVD4

CHAPTER 6. CONCLUSIONS AND FURTHER WORK

Appropriate constitutive models of materials are the key to a successful prediction of the behaviour of engineering structures. A vast number of models based on various constitutive theories have been proposed in the last three decades for geomaterials which show a large variation in their properties.

In this thesis a methodology for converting or recasting complex constitutive models for geomaterials developed in any mathematical framework into a fully trained neural network equivalent is proposed without the needs of complex parameters.

The length of strain trajectory traced by a material point, also called 'intrinsic time' is used as an additional new input parameter in training. This is essential for situations of cyclic and transient loading. For the purpose of illustration, two constitutive models viz. Hardening Soil Model (HSM) available in the commercial software, PLAXIS and a Two-surface Deviatoric Hardening Model in the multilaminate framework (TDHM) developed by Lee and Pande (2004) have been cast in the form of an ANN. It is seen that equivalents for both models can be easily trained and produce accurate results in all situations including a large number cycles.

The only limitation of the methodology, as we shown, is that the extrapolation by NNCM is always of poorer quality than interpolation.

This Neural Network based Constitutive Model can be embedded in an appropriate finite element solution code as a further work.

It is perceived that real life problems in future will have to be solved with increasingly more complex constitutive models for geomaterials. This will lead to unacceptable computational processing times. The NNCM which is provided in the current thesis, expected to overcome this problem since response of a trained NNCM equivalent is instantaneous. Computational efficiency could be achieved even for simpler models.

REFERENCES

Bazant, Z. P., Ansal, A. M. and Krizek, R. J. (1982), Endochronic models for soils, Chapter 15, Soil Mechanics - Transient & Cyclic Loads, Edited by G. N. Pande & O. C. Zienkiewicz, J. Wiley & Sons.

Duncan J.M, Chang C.-Y. (1970), Non Linear Analysis of Sress and Srain in Soil. ASCE J. of the Soil Mech. and Found. Div. Vol. 96, pp. 1629-1653.

Ghaboussi, J., Garrett, J. H. & Wu, X. (1991), Knowledge-based modelling of material behavior with neural networks, Journal of Engineering Mechanics, ASCE 117:1, 132-153.

Grothmann R. (2002), Multi-Agent Market Modeli ng based on Neural Networks, Ph.D. thesis, Faculty of Economics, University of Bremen, Germany.

Haykin S. (1994), Neural Networks. A Comprehensive Foundation, Macmillan College Publishing, New York.

Hebb, D.O. (1949), The organization of behavior. New York: Wiley.

Hinton Ackley, D. H., Sejnowski, T. J. (1985), A Learning Algorithm for Boltzmann Machines, Cognitive Science, 9:147-169.

Hopfield, J. J. (1982) "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences of the USA, vol. 79 no. 8 pp. 2554-2558.

Kohonen, T. (1982), Self-organizing formation of topologically correct feature maps. Biological Cybernetics, 43, 59–69.

Lee. K and Pande, G. N. , (2004), Development of a two-surface model in the multilaminate Framework, Proceeding of The Eighth International Symposium on Numerical Models in Geomechanics (NUMOG IX), edited by G.N. Pande & S. Pietruszczak.

Lee.K. (2005). "Development of a two-surface model in the multilaminate framework for soil", PhD Thesis, Swansea University

McCulloch, W. and Pitts, W. (1943), A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7:115 – 133

Minsky M L and Papert S A (1969), Perceptrons (Cambridge, MA: MIT Press)

Pande, G.N. & Sharma, K.G. (1983), Multilaminate model of clays – a numerical Evaluation of the influence of rotation of principal stress axes. Int. J. Numerical and Analytical Methods in Geomechanics: Vol. 7, No. 4, 397-418.

Riedmiller, M. and Braun, H. (1993), A direct adaptive method for faster backpropagation learning: the RPROP algorithm, Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, March 28-April 1.

Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386-408.

Schweiger, H.F. & Schuller, H. (2000). New developments and practical applications of the multilaminate model for soils.In Smith & Carter (Eds.), Proceedings of Developments in Theoretical Geomechanics - John Booker Memorial Symposium,Sydney: 329-350

Shin, H. S. and Pande, G. N. (2000). On self-learning finite element code based on monitored response of structures. Computers and Geotechnics 27, 161-178.

Shin, H. S. and Pande, G. N. (2001a), Intelligent finite elements, Proceeding of Asian-Pacific Conference for Computational Mechanics (APCOM'01), edited by S. Valliappan and N. Khalili, Sydney, Australia, pp. 1301-1310.

Shin, H. S. and Pande, G. N. (2001b), Intelligent finite elements in masonry research, Proceeding of International Symposium on Computer Methods in Structural Masonry (STRUMAS V), edited by T.G. Hughes & G.N. Pande, Rome, Italy, pp. 221-230.

Shin, H.S. (2001c) Neural Network Based Constitutive Models for Finite Element Analysis, Ph.D. thesis, Department of Civil Engineering, University of Wales Swansea.

Shin, H. S. and Pande, G. N. (2002), Enhancement of data for training neural network based constitutive models for geomaterials, Proceeding of The Eighth International Symposium on Numerical Models in Geomechanics (NUMOG VIII), edited by G.N. Pande & S. Pietruszczak, Rome, Italy, pp. 141-146.

Shin, H. S. and Pande, G. N. (2003), Identification of elastic constants for orthotropic materials from a structural test, Computers and Geotechnics (International Journal-SCIE), Vol 30, No. 7, pp. 571-577.

Valanis, K.C. & Read, H. E. (1982), A new endochronic plasticity model for soils, Chapter 14, Soil Mechanics Transient &Cyclic Loads, Edited by G. N.Pande & O. C. Zienkiewicz, J. Wiley & Sons.

Werbos Paul, (1974) Harvard University Ph.D. thesis