

**Evolutionary Polynomial Regression Based  
Constitutive Modelling and Incorporation  
in Finite Element Analysis**

**Mohammad Rezaia**

Ph.D. in Geotechnical Engineering

**University of Exeter  
August 2008**



School of Engineering, Computing and Mathematics

**EVOLUTIONARY POLYNOMIAL REGRESSION BASED  
CONSTITUTIVE MODELLING AND INCORPORATION  
IN FINITE ELEMENT ANALYSIS**

Submitted by

**Mohammad Rezaia**

to the University of Exeter as a thesis for the degree of  
Doctor of Philosophy in Geotechnical Engineering

**August 2008**

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

.....

To my parents whom I owe my whole life.

## **ACKNOWLEDGEMENTS**

I would like to express my sincere appreciation and my deepest gratitude to my supervisor Dr Akbar A. Javadi for his invaluable guidance, support and enthusiasm throughout the past four years of my study at University of Exeter. His patience and encouragement have led me through the hard time of my life and study during last four years, which I will never forget.

I also want to thank the past and current members of Computational Geomechanics Group who have been very supportive during my study. I would like to extend my appreciation to all the staff of and the services provided by the School of Engineering, Computing and Mathematics in the University of Exeter.

I wish to express my boundless gratefulness to my mother, Mehrangiz Solooki, and my father, Ali Rezania, for their invaluable love and support which have always given me the strength, wisdom and courage to overcome obstacles of my life. Together, my parents have prepared me for my life over the years, by teaching me how to set goals, make plans and then concentrate my efforts to reach my ambitions. I would like to thank my father for always inspiring me to take an interest in science, as he has always impressed me with his common sense and bright talent for science. I have my mother to thank for teaching me self discipline in all my works and to take pride in all that I have accomplished. I also wish to gratefully thank my older brother, Reza, and my sisters, Rita and Fatemeh, who have always stood by me, for their endless support and encouragement that have always helped me through the hard decisions I have made in my life. In addition, I would like to thank my brother in law, Siamak Safaei who is the one that talking with him has always inspired me to pursue my goals. I am also deeply indebted to my wife, Mohaddeseh Mousavi Nezhad, who deserves more praise than I can put down in words for her love, prayers, patience, and encouragement during my study.

Finally I thank my mighty God for the many blessings he has bestowed on me during my life and for giving me the strength and endurance to complete this research.

## **ABSTRACT**

Numerical analyses have, in recent years, been widely used as a powerful tool in the analysis of engineering problems. Conventionally, in numerical analysis, the behaviour of the actual material is approximated with that of an idealised material that deforms in accordance with some constitutive relationships. Therefore, the choice of an appropriate constitutive model that adequately describes the behaviour of the material plays an important role in the accuracy and reliability of the numerical predictions.

In this thesis a new evolutionary polynomial regression (EPR) based approach is presented for constitutive modelling of soils. EPR is an evolutionary computing method that generates a transparent and structured representation of the data provided. EPR can operate on large quantities of data in order to capture the complex interaction between the variables of the system. Furthermore it allows the user to gain additional insight into the constitutive behaviour of the material by providing a structured representation of the data. Capabilities of the EPR methodology in constitutive modelling of soils are illustrated by application to modelling of the behaviour of soils under drained and undrained loading conditions. In addition, an algorithm, so called feed-forward algorithm, has been developed to show that the proposed EPR based constitutive model is capable of reproducing the behaviour of soil over an entire stress path.

Moreover, a new approach is presented, for the first time, for constitutive modelling of materials in finite element analysis, with potential applications in different engineering disciplines. The proposed approach provides a unified framework for modelling of complex materials, using evolutionary polynomial regression based constitutive model (EPRCM), integrated in finite element analysis. The main advantage of EPRCM over conventional constitutive models is that it provides the optimum structure for the material constitutive model representation as well as its parameters, directly from raw experimental (or field) data. The proposed algorithm provides a transparent relationship for the constitutive material model that can easily be incorporated in a finite element model. The application of the EPRCM for material modelling in finite element analysis will be illustrated through a number of simple and complex examples.



## CONGRATULATIONS

**Mohammad Rezania**

on obtaining an

**Effective Researcher Award for Excellence 2008**

We are delighted that you have obtained this prestigious and highly competitive award whilst studying for your PhD at the University of Exeter.

We will do our best to ensure you have the support and training you need to succeed in your studies and in the future, and I hope that you will make full use of the resources we have on offer and will play a full part in the life of the postgraduate community at Exeter.

We wish you every good fortune for your future.

A handwritten signature in black ink that reads "Hilary Lappin-Scott".

Professor Hilary Lappin-Scott  
Dean of Postgraduate Studies  
July 2008

---

## TABLE OF CONTENTS

<b>Chapter 1. INTRODUCTION</b>	<b>1</b>
1.1 General background.....	1
1.2 Objectives.....	3
1.3 Structure of the thesis.....	4
<b>Chapter 2. REVIEW ON NEURAL NETWORK BASED CONSTITUTIVE MODELLING</b>	<b>6</b>
1.1 Introduction.....	6
2.2 Organisation of literature review.....	7
2.3 Neural network based constitutive models.....	7
2.4 NNCM implementation into finite element method.....	28
<b>Chapter 3. FINITE ELEMENT METHOD</b>	<b>31</b>
3.1 Introduction.....	31
3.2 Finite element procedure.....	32
3.3 General formulation.....	33
3.3.1 Basics of FE Formulation.....	33
3.3.2 Determination of the local element characteristics.....	39
3.3.3 Transformation of the element characteristics.....	39
3.3.4 Assemblage of the global element characteristics.....	39
3.3.5 Imposition of the boundary conditions.....	40
3.3.6 Solution.....	40
3.3.6.1 Gaussian elimination and back substitution.....	40
3.3.6.2 Iterative solution for nonlinear finite element.....	43
3.4 Constitutive relationships.....	46
3.5 Coupled stress-strain-flow analysis in saturated porous media.....	47
3.5.1 Principle of effective stress.....	47
3.5.2 Background.....	48
3.5.3 Finite element formulation.....	49

---

3.5.4 Discretisation in time domain.....	51
<b>Chapter 4. DATA MINING TECHNIQUES</b>	<b>53</b>
4.1 Introduction.....	53
4.2 Artificial neural network.....	54
4.2.1 General theory.....	54
4.2.2 The architecture of artificial neural network.....	54
4.2.2.1 ANN function.....	54
4.2.2.2 Layers.....	55
4.2.2.3 Neurons, connections, weights and biases.....	56
4.2.3 Learning rule.....	57
4.2.4 Training.....	58
4.2.5 The application of ANN in geomechanics.....	59
4.3 Genetic programming.....	60
4.3.1 General theory and background.....	60
4.3.2 Overview of genetic programming process.....	62
4.3.3 Initial population.....	63
4.3.4 Reproduction.....	64
4.3.5 Mutation.....	64
4.3.6 Crossover.....	65
4.3.7 The application of GP in geomechanics.....	66
4.4 Evolutionary polynomial regression.....	66
4.4.1 Overview.....	60
4.4.2 EPR model construction.....	62
4.4.3 Least squares solution by singular value decomposition.....	63
4.4.4 Model selection.....	64
4.4.4.1 Single objective approach.....	74
4.4.4.2 Multi objective approach.....	78
4.4.5 Extension of evolutionary polynomial regression.....	80
4.4.6 The application of EPR in geomechanics.....	81
<b>Chapter 5. CONSTITUTIVE MODELLING OF SOILS USING EVOLUTIONARY POLYNOMIAL REGRESSION</b>	<b>82</b>
5.1 Introduction.....	82
5.2 Constitutive modelling of soils using EPR.....	84

---

5.2.1 Drained condition.....	84
5.2.1.1 Input and output parameters and data preparation.....	85
5.2.1.2 Training and validation subsets.....	87
5.2.1.3 EPR procedure and results.....	87
5.2.2 Undrained condition.....	96
5.2.2.1 Input and output parameters.....	97
5.2.2.2 Training and validation subsets.....	97
5.2.2.3 EPR procedure and results.....	98
5.3 Application of EPR based material models for triaxial test simulation.....	106
5.3.1 Drained triaxial test simulator.....	106
5.3.2 Undrained triaxial test simulator.....	110
5.4 Discussion.....	116
<b>Chapter 6. INTELLIGENT FINITE ELEMENT METHOD AND NUMERICAL EXAMPLES</b>	<b>118</b>
6.1 Introduction.....	118
6.2 Background.....	119
6.3 Intelligent finite element method.....	122
6.3.1 Numerical examples.....	129
6.3.1.1 Example 1.....	129
6.3.1.2 Example 2.....	132
6.3.1.3 Example 3.....	138
6.3.1.4 Example 4.....	144
6.3.1.5 Example 5.....	147
6.3.2 Coupled stress-strain-flow analysis.....	150
6.3.2.1 Verification Example.....	150
6.3.2.2 Intelligent coupled analysis.....	153
6.4 Conclusions.....	159
<b>Chapter 7. CONCLUSIONS AND RECOMMENDATIONS</b>	<b>161</b>
7.1 Concluding remarks.....	161
7.2 Recommendations for future work.....	164
<b>REFERENCES</b>	<b>165</b>

## LIST OF FIGURES

<b>Figure 2.1</b>	A simple neural network based constitutive model and the adaptive determination of its hidden layers during training process, showing (a) initial neural network based constitutive model at the beginning of training and (b) final neural network based constitutive model at the end of training.....	9
<b>Figure 2.2</b>	Architecture of a typical sequential NN.....	11
<b>Figure 2.3</b>	Neural network predictions on training data for different sands showing (a) deviator stress predictions and (b) pore water pressure predictions (Ellis et al. 1995).....	13
<b>Figure 2.4</b>	Neural network prediction of stress-strain relationship of clay for (a) training and (b) testing (Penumadu and Chameau 1997).....	17
<b>Figure 2.5</b>	A typical recurrent neural network.....	18
<b>Figure 2.6</b>	The autoprogressively trained neural network material models showing (a) neural network material model for loose sand (b) neural network material model for medium dense sand (c) neural network material model for dense sand (Sidarta and Ghaboussi 1998).....	21
<b>Figure 2.7</b>	Symbolic representation of a typical nested adaptive neural network (Ghaboussi and Sidarta 1998).....	23
<b>Figure 2.8</b>	Results of trained NANN on drained triaxial compression tests (Ghaboussi and Sidarta 1998).....	25
<b>Figure 3.1</b>	The rectangular element (quadratic order).....	34
<b>Figure 3.2</b>	Graphical representation of the modified Newton-Raphson scheme.....	45
<b>Figure 4.1</b>	Neural network function.....	55
<b>Figure 4.2</b>	General architecture of a neural network with one hidden layer.....	55
<b>Figure 4.3</b>	A detailed schematic view of different layers and processing units in a neural network.....	56
<b>Figure 4.4</b>	The back-propagation processing unit.....	57
<b>Figure 4.5</b>	Typical GP tree representing function $\left(\frac{2}{x_1} + x_2\right)^2$ .....	61
<b>Figure 4.6</b>	Typical flow diagram for a genetic programming procedure where $M$ is the number of generations, $Pm$ is the probability of mutation; $Pc$ is the probability of crossover and $Ps$ is the probability of reproduction (Rezania and Javadi 2007).....	63
<b>Figure 4.7</b>	Typical mutation operation in GP.....	64
<b>Figure 4.8</b>	Typical cross over operation in GP.....	65

<b>Figure 4.9</b>	EPR classification among modelling techniques.....	67
<b>Figure 4.10</b>	Typical flow diagram for the EPR procedure (Rezania et al. 2008).....	71
<b>Figure 4.11</b>	Overview on main objective functions/strategies in EPR.....	73
<b>Figure 4.12</b>	SSE variations vs. $px$ .....	76
<b>Figure 5.1</b>	A schematic view of conventional triaxial experiment.....	84
<b>Figure 5.2</b>	Increase in COD value at early generations up to 25 seconds after starting the EPR procedure on drained triaxial data.....	90
<b>Figure 5.3</b>	Results of training the EPR models using (a) Equation (5.4) and (b) Equation (5.15).....	94
<b>Figure 5.4</b>	Results of testing the EPR models using (a) Equation (5.4) and (b) Equation (5.15).....	95
<b>Figure 5.5</b>	The results of undrained triaxial test on Nevada sand in terms of (a) deviatoric stress versus axial strain and (b) pore pressure change versus axial strain.....	96
<b>Figure 5.6</b>	Results of training the EPR models using (a) Equation (5.19) and (b) Equation (5.30)	103
<b>Figure 5.7</b>	Results of testing the EPR models using (a) Equation (5.19) and (b) Equation (5.30).....	104
<b>Figure 5.8</b>	Results of training the EPR models using Equation (5.44).....	105
<b>Figure 5.9</b>	Results of testing the EPR models using Equation (5.44).....	105
<b>Figure 5.10</b>	The schematic sketch of feed-forward algorithm for drained triaxial test simulator.....	108
<b>Figure 5.11</b>	The comparison between the results of drained triaxial test simulator and the tests used for training the EPR model.....	109
<b>Figure 5.12</b>	The comparison between the results of drained triaxial test simulator and two real tests which have not been part of EPR model construction.....	109
<b>Figure 5.13</b>	The schematic sketch of feed-forward algorithm for undrained triaxial test simulator.....	113
<b>Figure 5.14</b>	The comparison between the results of undrained triaxial test simulator and the tests used for training the EPR model.....	114
<b>Figure 5.15</b>	The comparison between the results of undrained triaxial test simulator and the real test which has not been part of EPR model construction.....	114
<b>Figure 6.1</b>	The incorporation of EPR based material model in a finite element code.....	123
<b>Figure 6.2</b>	Finite element solution strategy in nonlinear analysis using EPRCM...	126
<b>Figure 6.3</b>	Comparison of conventional FEM and EPR-FEM.....	128
<b>Figure 6.4</b>	Finite element mesh in symmetric quadrant of a thick cylinder with the boundary conditions.....	130

---

<b>Figure 6.5</b>	(a) Linear stress-strain relationship used for training, (b) the results of EPR predictions for stress-strain values.....	130
<b>Figure 6.6</b>	Comparison of the results of the EPR-FEM, standard FEM and theoretical solution in terms of (a) radial displacement, (b) radial stress and (c) tangential stress.....	132
<b>Figure 6.7</b>	Geometry of the bridge and the FE mesh showing boundary conditions.....	133
<b>Figure 6.8</b>	EPR based stress-strain curve passing through experimental stress-strain data points (adopted from Marsono 2000).....	134
<b>Figure 6.9</b>	Comparison of the results obtained from conventional linear elastic analysis and intelligent nonlinear EPR based analysis.....	136
<b>Figure 6.10</b>	Comparison between the values of deflection at the centre of bridge calculated from conventional linear and EPR based nonlinear FE models.....	137
<b>Figure 6.11</b>	Geometry of the embankment and the FE mesh.....	138
<b>Figure 6.12</b>	Comparison of the computed results for the surface settlement of embankment at different increments using standard FEM and EPR-FEM.....	139
<b>Figure 6.13</b>	Comparison of the results for displacement in Ydirection showing (a) displacement contours from standard FE analysis and (b) displacement contours from EPR based FE analysis.....	140
<b>Figure 6.14</b>	Comparison of the results for displacement in Xdirection showing (a) displacement contours from standard FE analysis and (b) displacement contours from EPR based FE analysis.....	141
<b>Figure 6.15</b>	Comparison of the results for major principal stress showing (a) stress contours from standard FE analysis and (b) stress contours from EPR based FE analysis.....	142
<b>Figure 6.16</b>	Comparison of the results for major principal strain showing (a) strain contours from standard FE analysis and (b) strain contours from EPR based FE analysis.....	143
<b>Figure 6.17</b>	Geometry of the tunnel and the FE mesh.....	144
<b>Figure 6.18</b>	Comparison of the results of the intelligent FEM and conventional FE analyses.....	145
<b>Figure 6.19</b>	Comparison between deformation results of the tunnel wall using two different EPRCMs.....	146
<b>Figure 6.20</b>	Geometry of the footing and the FE mesh.....	147
<b>Figure 6.21</b>	Contours of vertical stress obtained from (a) charts presented in Lambe and Whitman (1979) and (b) results of EPR-FEM.....	148
<b>Figure 6.22</b>	Comparison of the results for surface settlement of the footing obtained from the intelligent FEM and conventional FE analyses.....	150
<b>Figure 6.23</b>	Geometry of the verification problem and the FE mesh showing boundary conditions.....	152
<b>Figure 6.24</b>	Pore water pressure changes for the verification problem computed using analytical solution and finite element analysis.....	153

- 
- Figure 6.25** A schematic view of the finite element model used to generate the required data for development of EPR based constitutive relationship. For each particular cell pressure the variations of different parameters including stresses, strains, pore water pressure, velocity, hydraulic gradient and displacement at the specific integration point A are monitored and used for EPRCM development..... 154
- Figure 6.26** The results of the numerical simulation of isotropic consolidation test showing (a) the variation of effective stress with time at different applied pressures, (b) the variation of pore water pressure with time at different applied pressures, and (c) the variation of velocity of flow with time at different applied pressures..... 156
- Figure 6.27** The variation of surface settlement of the specimen (normalised) with square root of time at different applied pressures..... 156
- Figure 6.28** Comparison of the numerical results for the variations of the surface settlement with square root of time using intelligent EPR based and conventional FE models..... 158
- Figure 6.29** Pore water pressure changes with time along the specimen computed using coupled conventional FE model and coupled intelligent EPR based FE model..... 158

**LIST OF TABLES**

<b>Table 5.1</b>	The summary of results obtained for EPR based constitutive models using EPR structure type 1.....	88
<b>Table 5.2</b>	The summary of results obtained for EPR based constitutive models using EPR structure type 2.....	88
<b>Table 5.3</b>	The summary of results obtained for EPR based constitutive models using EPR structure type 3.....	89
<b>Table 5.4</b>	The summary of results obtained for EPR based constitutive models using EPR structure type 4.....	89
<b>Table 6.1</b>	Material parameters for Mohr-Coulomb model.....	139
<b>Table 6.2</b>	Material parameters for Duncan-Chang (1970) and Mohr-Coulomb models.....	145
<b>Table 6.3</b>	Comparison of EPR-FEM and closed form results for circular footing.	149

## LIST OF PUBLICATIONS

The publications listed here were completed during the author's period of registration to the higher degree program. Other publications by author are not included.

### Refereed Journals

- **Rezania, M.**, Javadi, A.A. and Giustolisi, O. (2008). An Evolutionary-Based Data Mining Technique for Assessment of Civil Engineering Systems. *Journal of Engineering Computations* (accepted, in print).
- **Rezania, M.** and Javadi, A.A. (2007). A New Genetic Programming Model for Predicting the Settlement of Shallow Foundations. *Canadian Geotechnical Journal*, 44(12), pp 1462-1473.
- Javadi, A.A., **Rezania, M.** and Nezhad, M.M. (2006). Evaluation of Liquefaction Induced Lateral Displacements using Genetic Programming. *Journal of Computers and Geotechnics*, 33(4-5), pp 222-233.
- Javadi, A.A. and **Rezania, M.** (2008). Intelligent Finite Element Method. *International Journal for Numerical Methods in Engineering*, (under review).
- **Rezania, M.**, Javadi, A.A. and Faramarzi, A. (2008). A 3-D Limit State Boundary Surface for Assessment of Liquefaction Potential. *Geotechnique*, (under review).
- **Rezania, M.**, Javadi, A.A. and Giustolisi, O. (2008). Evaluation of Liquefaction Potential Based on CPT Results using Evolutionary Polynomial Regression, *Journal of Computers and Geotechnics*, (under review).
- **Rezania, M.** and Javadi, A.A. (2008). Predicting Settlement of Shallow Foundations using Evolutionary Polynomial Regression. *Journal of Computer-Aided Civil and Infrastructure Engineering* (under review).
- **Rezania, M.** and Javadi, A.A. (2008). A New Approach to Data Driven Modelling in Geotechnical Engineering. *Journal of Geomechanics and Geoengineering* (under review).

### Book Chapter

- **Rezania, M.**, Faramarzi, A. and Javadi, A.A. (2008). An evolutionary based approach for assessment of earthquake-induced soil liquefaction and lateral displacement. Chapter 74, to be published in "Foundation on Computational Intelligence", special volume on the series "Studies in Computational Intelligence", published by Springer Verlag, Germany.

**Refereed Conferences**

- Javadi, A.A. and **Rezania, M.** (2008). Intelligent Finite Element Modelling. Proceeding of the 15th Annual Workshop of the European Group for Intelligent Computing in Engineering, Y. Rafiq, P. de Wilde and M. Borthwick (eds.), 2-4 July 2008, Plymouth, England, UK, pp. 147-156.
- **Rezania, M.** and Javadi, A.A. (2008). Settlement Prediction of Shallow Foundations; A New Approach. Proceeding of the 2nd British Geotechnical Association International Conference on Foundations, M. Brown (ed.), 24-27 June 2008, Dundee, Scotland, UK, pp. 1593-1603.
- Javadi, A.A., Nezhad, M.M., **Rezania, M.** and Al-Najjar, M.M. (2008). Modelling of Contaminant Transport in Soils Considering the Effects of Bioremediation. Proceeding of the International Conference on Impacts of Pollution in a Changing Urban Environment, 17-19 September 2008, Manchester, England, UK, paper ID. 1332.
- Javadi, A.A., **Rezania, M.** and Nezhad, M.M. (2007). A New Approach to Data-Driven Modelling in Civil Engineering. Proceeding of the 1st International Conference on Digital Communications and Computer Applications, 19-22 March, 2007, Amman, Jordan, pp. 17-24.
- **Rezania, M.** and Javadi, A.A. (2006). Application of Evolutionary Programming Techniques in Geotechnical Engineering. Proceeding of the 6th European Conference on Numerical Methods in Geotechnical Engineering, H. F. Schweiger (ed.), 6-8 September, 2006, Graz, Austria, pp. 677-682.
- Javadi, A.A. and **Rezania, M.** (2006). A New Genetic Programming-Based Evolutionary Approach for Constitutive Modelling of Soils. Proceeding of the 7th World Congress on Computational Mechanics, 16-22 July, 2006, Los Angeles, CA.

**LIST OF ABBREVIATIONS**

ANN	Artificial Neural Network
BPNN	Back-Propagation Neural Network
COD	Coefficient of Determination
EPR	Evolutionary Polynomial Regression
EPRCM	Evolutionary Polynomial Regression based Constitutive Model
EPR-FEM	Evolutionary Polynomial Regression based Finite Element Method
FE	Finite Element
GA	Genetic Algorithm
GP	Genetic Programming
HSM	Hardening Soil Model
LS	Least Square
MO-EPR	Multi-Objective approach in Evolutionary Polynomial Regression
NANN	Nested Adaptive Neural Network
NN	Neural Network
NNCM	Neural Network based Constitutive Model
OCR	Over Consolidation Ratio
PCS	Penalisation of Complex Structures
RNN	Recurrent Neural Network
SSE	Sum of Squared Errors
StD	Standard Deviation
SVD	Singular Value Decomposition

---

**LIST OF SYMBOLS**

$\sigma_1$	major principal stress
$\sigma_3$	confining pressure
$p'$	mean effective stress
$\varepsilon_1$	major principal strain
$\varepsilon_3$	minor principal strain
$\varepsilon_v$	volumetric strain
$\varepsilon_d$	deviatoric strain
$\dot{\varepsilon}$	rate of strain increment
$u$	pore water pressure
$D_r$	initial relative density
$C_u$	coefficient of uniformity
$\Delta\varepsilon_1$	increment of major principal strain
$\Delta\varepsilon_3$	increment of minor principal strain
$\sigma'$	effective stress
$\sigma'_3$	effective confining stress
${}^i\tau$	shear stress
$w_0$	natural water content
$w_l$	liquid limit
$w_p$	plastic limit
$e$	void ratio
$e_0$	initial void ratio
$h$	hardness of the mineral
$n_s$	shape factor for granular material
$\theta$	soil water content
$\rho_d$	soil dry density
$S_r$	degree of saturation of soil
$\Delta d$	vector of displacements
$\psi$	out of balance force
$\Omega$	element domain

---

$\Gamma$	element boundary
$\hat{t}$	vector of surface tractions
$\phi$	porosity of the soil
$\alpha$	time interpolation parameter
$\kappa$	slope of the unloading/reloading line
$B$	strain-nodal displacement matrix
$b$	vector of body forces
$c'$	cohesion
$c_v$	coefficient of consolidation
$D$	elastic constitutive matrix
$D_{50}$	average mean grain size for granular materials
$D_{ep}$	elasto-plastic constitutive matrix
$E$	Young's modulus
$e$	error vector
$E_{EPR}$	EPR based elastic modulus
$F$	vector of applied loads
$g$	gravitational constant
$H$	permeability matrix of the porous skeleton
$I$	unity matrix
$J$	Jacobian matrix
$k$	absolute permeability matrix
$K_E$	element stiffness matrix
$K_S$	bulk modulus of the soil grains
$K_W$	bulk modulus of the pore water
$L$	coupling matrix between the solid and fluid phases
$m$	identity vector
$M$	slope of the critical state line
$N$	intercept of the virgin consolidation line with vertical axis
$P_0$	isotropic preconsolidation pressure
$q$	deviatoric stress
$Q$	vector of boundary flows
$q$	vector of applied fluid flux
$S$	compressibility matrix
$t$	time
$T$	time factor
$Z$	depth of soil sample from ground surface

$\gamma$	unit weight
$\lambda$	slope of the virgin consolidation line
$\mu$	dynamic viscosity of the pore water pressure
$\rho$	water density
$\nu$	Poisson's ratio
$\varphi'$	friction angle

# CHAPTER 1

## INTRODUCTION

### 1.1 General background

The modelling of material behaviour is of vital importance in numerical analysis of various structures. With advancing developments in numerical methods such as finite element (FE), analysing a wide variety of problems in different engineering disciplines has become viable. One of the most fundamental aspects of FE analysis is the constitutive model used to represent the mechanical response of the material involved. It is generally accepted that the reliability of the results of FE analysis is largely dependant on the suitability of the constitutive model utilised within the FE procedure.

Traditionally, constitutive modelling aim to construct mathematical models to describe the relationship between stresses and strains in general. Conventional constitutive models for determination of behaviour of different materials, have been developed more or less in the same manner

- i) A material is tested to extract its behaviour.
- ii) A mathematically formulated model is postulated to represent the observed behaviour of the material.

- iii) The performance of the developed model is examined by its application to predict unseen stress paths and compared against additional experiments.
- iv) The mathematical model is modified to extend its application for behaviours observed but not explained by the model.

Most of the mathematically developed constitutive models for various materials such as soil, rock, concrete, etc., involve the determination of various material parameters, some of which have a little physical meaning. Where the requirements for accuracy in predicting the behaviours of materials become more crucial, the constitutive models need to be more sophisticated. As the complexity of the material model increases, so does the number of parameters to be identified. However, despite considerable complexity of the conventional constitutive models, capturing the real material response, along all the stress paths and over a wide range of confining pressures, has not been possible so far. In addition the increased complexity of some constitutive theories, makes them difficult to implement in FE codes and therefore restrict their functionality in engineering practice.

Following the successful application of artificial intelligent techniques, particularly neural network (NN), (as a paradigm for computational knowledge representation and a pattern recognition device) in modelling nonlinear processes in various engineering fields, researchers have been encouraged to investigate the potential of NN in constitutive modelling. Research on the application of neural networks to the problem of material modelling was first carried out by Ghaboussi et al. (1990). Since then a large number of intelligent NN based constitutive models (NNCMs) for different materials has been introduced. From these works, it has been shown that a well trained NN can be used to model the constitutive behaviour of an engineering material. However, as a so called black box method, NN has its own drawbacks. Learning in NN involves tuning the parameters (weights) of interconnections in a highly parameterised system. This requires that the structure of the neural network to be identified *a priori*, which can be a very time consuming procedure (i.e., determination of network type, kernel type, transfer functions, number of hidden layers, etc.). Because the output of NNCM is not a structured relationship, after the development of a NNCM, each time for its application it has to be reproduced. However, the reproduced NNCM is not necessarily the same as the initially developed network. This is due to the high sensitivity of NN to the values of its parameters. Although one of the main purposes of constitutive modelling is to be

incorporated in numerical analysis, to date very few researchers have dealt with the implementation of NNCM in numerical procedures such as the FE method. The main reason for this appears to be the fact that there are considerable difficulties, like the one described above, in incorporating a NNCM in FE analysis. Also the robustness of a NNCM in FE analysis is a vital issue; that is, it is essential that a FE code incorporating a NNCM converges at each load increment otherwise the NNCM is worthless.

Thus, the application of an alternative intelligent based methodology which can overcome the shortcomings of using NNCM, would be of great advantage. Accordingly, for the development of the new methodology, at least the following three issues should be addressed

- i) Obtaining a structured and transparent relationship that represents the material behaviour.
- ii) Training the intelligent based material model in such a way that it can predict the material behaviour for a wide range of unseen stress paths objectively.
- iii) Incorporation of such intelligent constitutive model in a FE code in an easy and flawless manner.

In order to deal with the concerns listed above, for the first time in this thesis, a new evolutionary based data driven technique called evolutionary polynomial regression (EPR) has been utilised for the purpose of constitutive modelling of materials in general and geomaterials in particular.

## 1.2 Objectives

Evolutionary polynomial regression is a new hybrid paradigm which has been developed and tested for data mining purposes in environmental related issues (Doglioni 2004, Giustolisi and Savic 2006). Further investigation of the methodology showed its great performance in other fields of civil engineering as well (Rezania et al. 2008a). Motivated by the capability of EPR in pattern recognition and data mining, the objectives of this work are

- The full description and development of EPR based material modelling, particularly aimed at modelling the behaviour of geomaterial; to investigate the optimal learning algorithm for EPR for incorporation in material modelling; and to verify the approach through modelling constitutive relations of soils in different stress states (i.e., normal stress-strain and coupled stress-strain-flow conditions).
- To set up a strategy for using the developed EPR based constitutive model (EPRCM) of soils as a stand alone simulator and triaxial data generator. This can help to maximise the value of information embedded in the EPRCM especially in cases where results from material tests are only available for a limited number of stress paths.
- To develop and introduce the new EPR based finite element methodology in computational mechanics and investigate technical aspects involved in the incorporation of EPRCM in the FE code.
- To validate the proposed intelligent FE model through some illustrative structural and geotechnical examples.

### **1.3 Structure of the thesis**

With the above objectives, this thesis is organised in 7 chapters. The main text of each chapter is intentionally kept as short as possible in favour of easy reading and is written to include only the fundamental concepts and the new ideas.

In chapter 2, a literature review, of the efforts that have been made on neural network based constitutive modelling, is provided. This chapter begins with the background information relating to NNCM and follows by an overview of main published works in this field. The way that the data is treated at each work is studied and the merits and deficiencies of each work are discussed in detail.

In chapter 3, the general concept of finite element method is illustrated. The architecture of the method itself is looked at in some details and the common solution procedures for dealing with linear and nonlinear problems in FEA are discussed. Also the mathematical formulations and governing equations which are used in the development of the required uncoupled and coupled FE programs for this thesis are provided.

Chapter 4 gives an insight into the most commonly used data mining techniques (i.e., neural network and genetic programming). The conceptual description of the terminologies related to the operation of each of these techniques is provided in separate sections. Furthermore the main shortcomings of these approaches which have prevented them from achieving their full potential are highlighted. Also the new EPR technique is introduced in this chapter.

In chapter 5, which is one of the main chapters of this thesis, the EPR based constitutive modelling of soils is introduced. Two different conditions (drained and undrained) are considered for EPRCM and the way that the data for each case is managed is discussed. In addition, the results of the investigation which has been made to find the optimal EPR structure for the purpose of constitutive modelling are presented in this chapter. Furthermore, two separate feed-forward algorithms are introduced for simulating triaxial experiments, from the already developed EPRCMs, in drained and undrained conditions.

Chapter 6 is the second main chapter of this thesis, in which the EPR based intelligent FE method is described in detail. The methodology in which the EPRCM is incorporated in FE code is described and the procedure to identify the nonlinear elastic constant from the trained EPRCM is depicted. Several different examples are analysed using the proposed intelligent FE model, and for each example comparisons are made between the results obtained from intelligent FE and those of a standard FEA using conventional constitutive models.

Finally in chapter 7, the main conclusion of the thesis and recommendations for further research are presented.

# CHAPTER 2

## REVIEW ON NEURAL NETWORK BASED CONSTITUTIVE MODELLING

### 2.1 Introduction

The engineering properties of soil and rock exhibit varied and uncertain behaviour due to the complex and imprecise physical processes associated with the formation of these materials. Recognition of nonlinear behaviours of soils and rocks becomes increasingly important in design, stability analysis, prediction and control of failure for geotechnical engineering projects. Conventionally in numerical analysis like finite element method (FEM) the behaviour of the actual material is approximated with that of an idealised material that deforms in accordance with some constitutive relationships. Therefore the choice of an appropriate constitutive model which adequately describes the behaviour of the material plays a significant role in the accuracy and reliability of the numerical predictions.

During the past few decades several constitutive models have been developed for different geomaterials based on mechanics (e.g., Desai et al. 1986; Duncan and Chang 1970; Einstein and Hirschfeld 1973; Kawamoto et al. 1988; Lade and Duncan 1975; Roscoe and Schofield 1963). Most of these models involve determination of material parameters, many of which have no physical meaning (Shin and Pande 2000). For

example in spite of considerable complexities of constitutive theories, due to the erratic nature of soils, rocks, composites, etc. none of the existing constitutive models can completely describe the real behaviour of various types of these materials under various stress paths and loading conditions. In conventional constitutive material modelling, an appropriate mathematical model is initially selected and the parameters of the model (material parameters) are then identified from appropriate physical tests on representative samples to capture the material behaviour. When these constitutive models are used in numerical analysis (e.g., FEM), the accuracy with which the selected material model represents the various aspects of the actual material behaviour and also the accuracy of the identified material parameters affect the accuracy of the finite element prediction.

Development of a unified approach for constitutive modelling that can be applied to any type of soil would be extremely beneficial in the finite element analysis of geotechnical problems. Recently some researchers attempted to build nonlinear constitutive models based on computer-aided pattern recognition methods. Artificial neural network (ANN) has been the most widely used pattern recognition procedure to model the constitutive material behaviour.

## **2.2 Organisation of literature review**

The references reviewed for this evaluation of published literature are covered in two categories. The first is the state of the art papers on modelling of soil behaviour with artificial neural network. The second category is the papers describing the performance of the implementation of neural network based constitutive models (NNCMs) in finite element analysis.

## **2.3 Neural network based constitutive models**

Modelling of soil behaviour plays an important role in dealing with issues related to soil mechanics and foundation engineering. In addition to conventional constitutive

modelling, the application of artificial intelligence methods offers an alternative means for the modelling of soil behaviour.

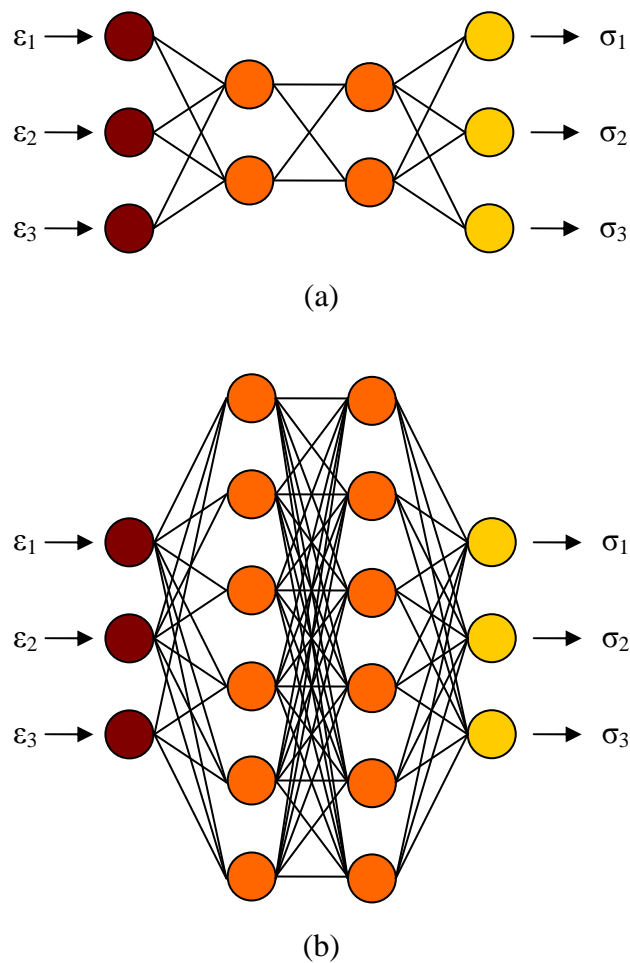
A neural network based constitutive model is fundamentally different from the conventional constitutive model (Zhu et al. 1998a). One of its distinctive features is that it is based on experimental data rather than on assumptions made in developing a constitutive model. Also in an NNCM the material constants are not needed. These features ascertain the NNCM to be an objective model that can truly represent the natural connections among variables, rather than a subjective model which assumes the variables obey a set of predefined relations.

The NNCM learns from experimental data and forms neural connections stimuli from the learning process. Because of its unique learning, training and predicting characteristics, the neural network model has great potential in soil engineering applications, particularly for the situations where good experimental data are available and where conventional constitutive modelling may be difficult and time consuming.

There are quite a few references regarding the application of neural network (NN) in modelling of soil behaviour. Initially the application of NN for constitutive modelling was proposed by Ghaboussi et al (1990; 1991) for concrete. Later on Ellis et el. (1992) and Ghaboussi et al. (1994) applied the concept of NN constitutive modelling to model the behaviour of geomaterials. These works indicated that neural network based constitutive models can effectively capture nonlinear material behaviour. These models are versatile and have the capacity to continuously learn as additional material response data become available.

The very common procedure of using NNs for conventional constitutive modelling involves training a neural network using laboratory (or in-situ) data to learn the material behaviour. The advantages of using NN when it is trained directly from some experimental (or in-situ) data is obvious. The trained network is then used to predict the behaviour of the material under new loading conditions. If the training data contains enough and relevant information, the trained network should be able to generalise the material behaviour to new loading conditions.

Among various types of neural networks, multi-layer feed-forward back-propagation network is known to be the most suitable architecture to describe the nonlinear relationships, and so far, has been the only type of neural network used to describe material constitutive behaviour (Hashash et al. 2004). The role of the NN is to attribute a given set of output vectors to a given set of input vectors. When applied to the constitutive description, the physical nature of these input-output data is determined by the measured quantities like stresses, strains, pore pressures, temperatures, etc. A NN based constitutive model consists of several layers of computational nodes, or neurons. A typical NN based constitutive model is shown schematically in Figure 2.1.



**Figure 2.1** A simple neural network based constitutive model and the adaptive determination of its hidden layers during training process, showing (a) initial neural network based constitutive model at the beginning of training and (b) final neural network based constitutive model at the end of training.

In the simple example shown in Figure 2.1 one input layer, two hidden layers and one output layer are considered for the network. Three principal strain components ( $\varepsilon_1$ ,  $\varepsilon_2$  and  $\varepsilon_3$ ) for an assumed 2-D medium are input and a forward pass through the

network including very simple computations, results in the prediction of three corresponding principal stresses ( $\sigma_1, \sigma_2$  and  $\sigma_3$ ) at the output layer. Every neuron in each layer is connected to every neuron in the next layer and each such connection has associated with it a “connection weight”. The knowledge stored in the developed network is represented by the set of connection weights. The neural network is trained by appropriately modifying its connection weights, through the set of “training cases”, until the predicted stress variables agree satisfactorily with the correct stress variables. The “back-propagation” term (Rumelhart et al. 1986) refers to the algorithm by which the observed error in the predicted stress variables is used to modify the connection weights.

Encouraged by the attractive features of neural networks, after exploration of the potential of NN for constitutive modelling during early 90’s; a large number of NNCMs for different materials including soils and rocks have been developed. In the following a detailed review of major published studies on constitutive modelling of soil and/or rock using artificial neural networks is presented.

Millar and Clarici (1994) showed the capability of NNs for modelling purposes in the field of rock mechanics by modelling the deformation of a particular rock sample with NN, based on some laboratory test results of axial stress-axial strain measurements. Four different NN models, in terms of number of hidden neurons, were developed and it was shown that they are able to predict the axial stress-axial strain relationship with good accuracy. In this work, the structure of developed NN was based on back-propagation multilayer perceptron architecture, and the input-output set for training the model was

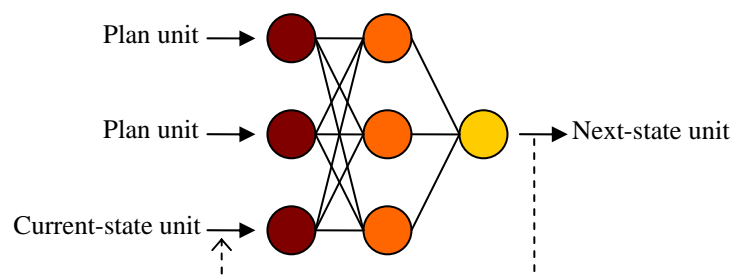
$$\text{Inputs: } \sigma_3, \varepsilon_1, \varepsilon_3 \text{ and } \left( \text{sign} \left[ \frac{d\sigma_1}{d\varepsilon_1} \right] \right)$$

$$\text{Output: } \sigma_1$$

where  $\sigma_1$  is the major principal stress,  $\sigma_3$  is the confining pressure,  $\varepsilon_1$  is the major principal strain,  $\varepsilon_3$  is the minor principal strain and  $\text{sign} \left[ \frac{d\sigma_1}{d\varepsilon_1} \right]$  is the sign of the gradient of the stress-strain curve. Although the proposed models could predict the

major principal stress,  $\sigma_1$ , correctly; however the fact that the latter input parameter was required for the NN training implied that some indication of the history of stress state was necessary as input in the training process. Also it was not clear how such trained NN model would be used in engineering practice. A nonlinear constitutive model of such form would seem sensible in that it indicates that the major principal stress is related to the state of strain of the rock and whether the rock has past its peak strength. However it is difficult to imagine a situation where all such information would be available within an analysis without the value of the major principal stress being known already.

Ellis et al. (1995) modelled the stress-strain relation of sands using artificial neural network and observed good agreement between laboratory data and modelling results. A series of undrained triaxial tests on mortar sand was used to develop the models. Two different types of architecture were used to evaluate the ability of NN for modelling sand behaviour. They were the conventional neural network without feedback and the sequential NN with feedback (Figure 2.2).



**Figure 2.2** Architecture of a typical sequential NN.

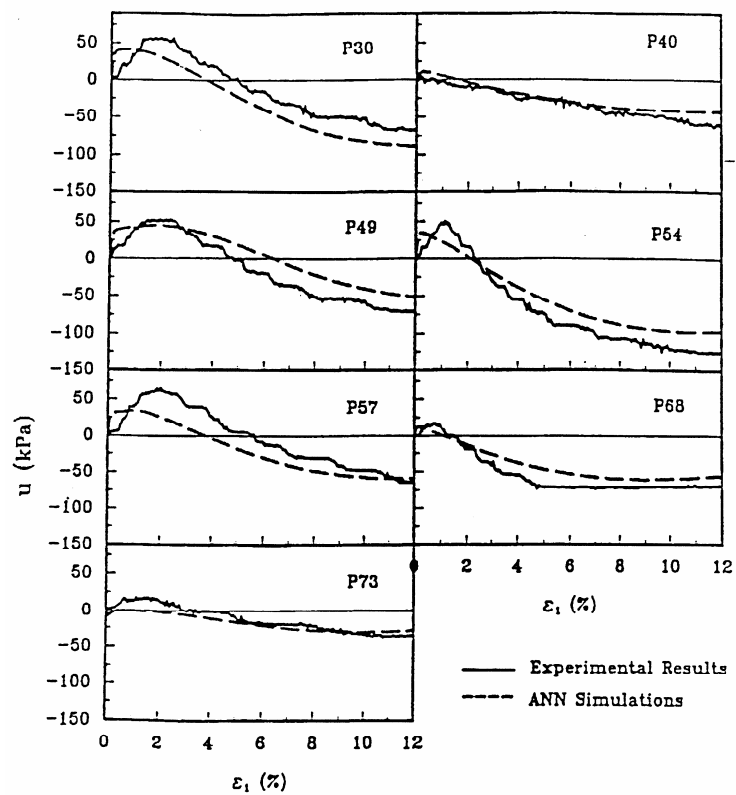
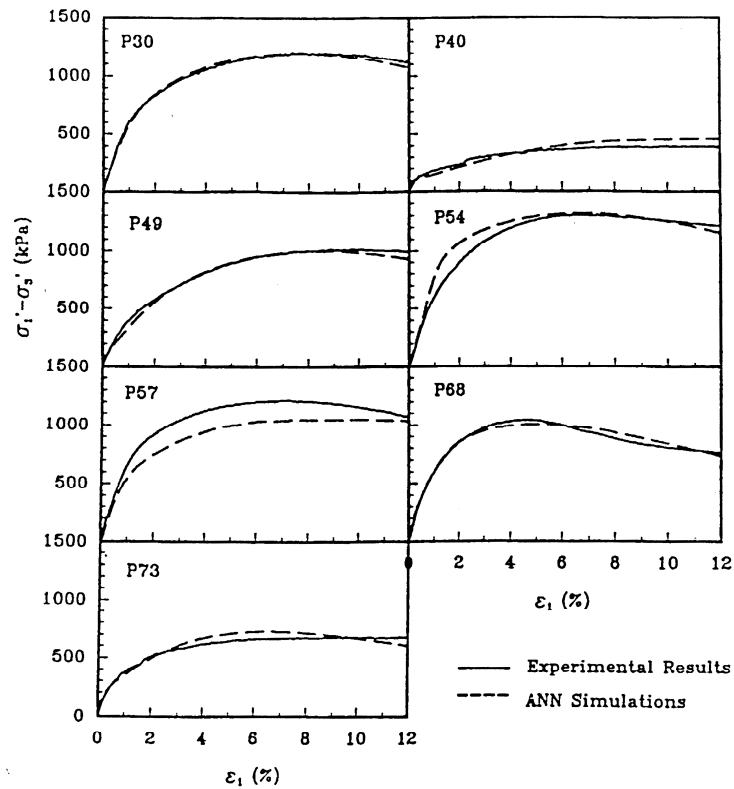
Using this type of NN (i.e., sequential NN) at the initial phase of the training a pattern is input to the plan units. Feed forward process occurs as in standard back propagation, producing the first output pattern. This output is then copied back to the current state units for the next feed forward process. The recurrent NN has the potential in incorporating the path dependency of mechanical behaviour into the model. In order to accommodate this aspect, the input-output parameters for the model should be variables of time. Based on the results it was found that the sequential NN worked better than the regular NN. Thus the authors proposed a sequential network with three layers which had 10 neurons in the intermediate layer and its input-output parameters were

Inputs:  ${}^i\sigma_1$ ,  $\sigma_3$ ,  ${}^i\varepsilon_1$ ,  ${}^i u$ ,  $OCR$ ,  $D_r$  and  $C_u$

Output:  ${}^{i+1}\sigma_1$  and  ${}^{i+1}u$

where  $u$  is the pore water pressure,  $OCR$  is the over consolidation ratio which reflects the previous stress history,  $D_r$  is the initial relative density and  $C_u$  is the coefficient of uniformity which characterises the grain size distribution of sand. Also a constant value of 0.0405% was used for the axial strain increment,  $\Delta\varepsilon_1$ . Figures 2.3(a) and 2.3(b) show the NN predictions during training phase.

Although, based on the reported graphs, the results of NN predictions were not significantly attractive, even at training stages, it was stated that the developed NN model was able to “effectively” capture stress-strain relationship of mortar sand. However, it was argued later that a prescribed strain rate (0.0405% per minute) has to be defined in order to make predictions with this model (Najjar and Basheer 1996). This issue limits the developed network as applicable only to the specific situation with a strain rate of 0.0405% per minute.



**Figure 2.3** Neural network predictions on training data for different sands showing (a) deviator stress predictions and (b) pore water pressure predictions (Ellis et al. 1995).

Millar and Calderbank (1995) showed that a single multilayer feedforward artificial neural network is able to predict the deformability behaviour of rock. Data used to train the neural network model in this work, was derived from the results of a series of simulations of triaxial tests using commercial explicit finite difference software, FLAC. The authors made some modifications in their NN training approach in order to resolve the deficiencies associated with the earlier work (Millar and Clarici 1994) and make their model worthy for immediate use as a stand alone constitutive relationship in a numerical modelling code. For this purpose the authors used the same NN architecture as their earlier work, but they revised the way that the input-output parameters were introduced to NN for training procedure. The input-output parameter sets used for the training of their revised NN based constitutive model were

Inputs:  ${}^i\sigma_1$ ,  ${}^i\sigma_3$ ,  ${}^i(\Delta\varepsilon_1)$  and  ${}^i(\Delta\varepsilon_3)$

Output:  ${}^{i+1}\sigma_1$  and  ${}^{i+1}\sigma_3$

where  $(\Delta\varepsilon_1)$  and  $(\Delta\varepsilon_3)$  are the increments of major and minor principal strains, respectively. The data, which were produced by triaxial test simulation in FLAC using strain softening model available within this software, had to be scaled within the interval between -0.5 and 0.5 for NN training process. Also the value of minor principal stress,  $\sigma_3$ , was considered not to be identical for a single test. This was done through the superposition of a component of noise to the input values on each presentation of the data to the NN. The optimum NN structure obtained for constitutive relationship was then used to develop a user defined constitutive model, called NN UDM, back in FLAC. Although the accuracy of the NN model over the training data was good however its prediction ability was so poor and the actual behaviour of the NN UDM was far from desired behaviour when it was used in place of the standard strain softening constitutive model within FLAC.

Amorosi et al. (1996) also tried to increase the possible options for selection of constitutive models for geotechnical applications through adoption of a neural network based representation of the geomaterial constitutive behaviour. The data obtained from undrained triaxial tests on a particular clay type (Vallericca clay) was used to develop the NN model. The input-output parameter sets used in this work were

Inputs:  ${}^i\sigma_1$ ,  $\sigma_3$ ,  ${}^i u$ ,  ${}^i(\Delta\varepsilon_1)$  and  $OCR$

Output:  ${}^{i+1}\sigma_1$  and  ${}^{i+1}u$

The constitutive behaviour of Vallericca clay was shown to be adequately represented with the trained NN model. The developed model had a back propagation multilayered perceptron architecture with three layers. Obviously, the input and output layers used 5 and 2 nodes respectively. Also one intermediate layer was used which contained four nodes.

Logar and Turk (1997) presented the modelling of a constitutive law for soft soils by artificial neural network. The results from oedometer loading tests on a silty soil were used to train a feed forward neural network. Base on the source of the available data the following input-output set was used to develop the model

Inputs:  $\sigma'$ ,  $Z$ ,  $w_0$ ,  $w_l$  and  $w_p$

Output:  $e$

where  $\sigma'$  is effective stress,  $Z$  is the depth from which the sample was taken,  $w_0$  is natural water content,  $w_l$  is liquid limit,  $w_p$  is plastic limit and  $e$  is void ratio. The optimum NN structure was obtained by a single hidden layer consisting of 35 hidden neurons. The results for approximation of oedometer curves by NN were relatively accurate compared to the experimental measurements with average error of around 10% for the training phase. The trained NN was used to determine the tangential oedometer modulus by the equation

$$E_{oed} = -\left(1 + {}^i e\right) \frac{\Delta\sigma'}{\Delta e} = -\left(1 + {}^i e\right) \frac{{}^{i+1}\sigma' - {}^i\sigma'}{{}^{i+1}e - {}^i e} \quad (2.1)$$

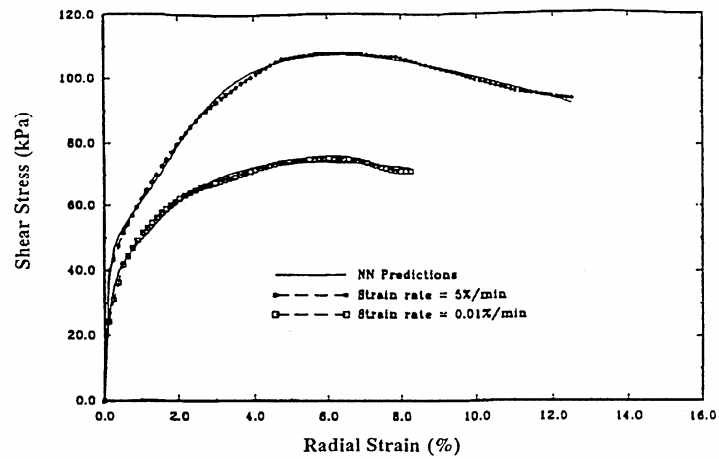
The above equation was then used, instead of the elastic parameter, in a finite element code to model the amount of settlement on top of an embankment. The results were reported to be comparable with the results obtained using a cap model for deformation; however, based on the information provided in the paper, it is not clear how the Equation (2.1) was incorporated into the FE procedure.

Penumadu and Chameau (1997) presented the soil behaviour modelling within a unified environment based on NN. The same triaxial data as used in Ellis et al. (1995) was used for training and testing of the NN sand model. Also stress-strain data obtained from a series of strain controlled undrained triaxial tests on clay was used for training and testing the NN clay model. The same type of NN as the one used in Ellis et al. (1995) (feed back sequential NN) was again used in this work. The NN architecture and results for Mortar sand were identical to those presented in Ellis et al. (1995); however for clay a different NN architecture including one hidden layer with 10 nodes was selected. The input-output set for clay model was

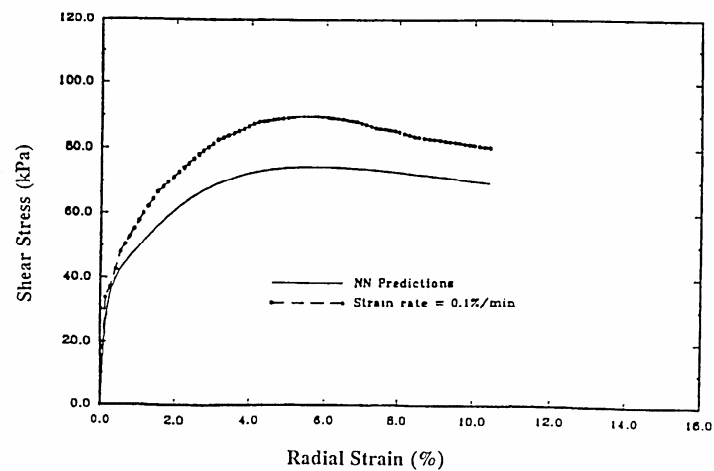
Input:  ${}^i\tau, \dot{\epsilon}, {}^i\epsilon, {}^i(\Delta\epsilon)$

Output:  ${}^{i+1}\tau$

where  ${}^i\tau$  is shear stress and  $\dot{\epsilon}$  is the rate of strain increment. The results of the NN prediction of stress-strain relationship of clay during training and testing phases are shown in Figure 2.4.



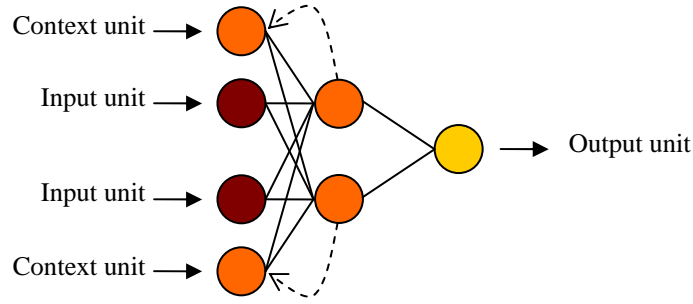
(a)



(b)

**Figure 2.4** Neural network prediction of stress-strain relationship of clay for (a) training and (b) testing (Penumadu and Chameau 1997).

Zhu et al. (1998a) presented a recurrent neural network (RNN) model for simulating and predicting shear behaviour of two different kinds of soil. A recurrent neural network is a neural network where the connections between the units form a directed cycle. Recurrent neural networks must be approached differently from feedforward neural networks, both when analysing their behaviour and training them. Hidden nodes in an RNN can transmit their outcomes to both input layer and output layer simultaneously (Elman 1990). A typical architecture of an RNN with one hidden layer is shown in Figure 2.5.



**Figure 2.5** A typical recurrent neural network.

Laboratory based experimental data were used for modelling including a set of strain controlled undrained tests and a set of stress controlled drained tests performed on a residual Hawaiian volcanic soil. The choice of input-output variables was different due to different sources of data. For the strain controlled test the goal was to measure stress response of the specimen to a given strain value, therefore the selected input-output variables for RNN training were

Input:  ${}^i q$ ,  ${}^i \sigma'_3$ ,  ${}^i u$ ,  ${}^i \varepsilon_1$ ,  ${}^i (\Delta \varepsilon_1)$  and  ${}^i e$

Output:  ${}^{i+1} q$  and  ${}^{i+1} u$

where  $q = \sigma_1 - \sigma_3$  is deviatoric stress.

In contrast, as for a stress controlled test, the shear stress and stress increment were known in advance, the goal was to measure the strain response of the specimen. Thus for such test results, the selected input-output parameters for RNN training were

Input:  ${}^i \sigma'_1$ ,  ${}^i \sigma'_3$ ,  ${}^i (\Delta \sigma'_1)$ ,  ${}^i (\Delta \sigma'_3)$ ,  ${}^i u$ ,  ${}^i \varepsilon_1$ ,  ${}^i \varepsilon_v$  and  ${}^i e$

Output:  ${}^{i+1} \varepsilon_1$  and  ${}^{i+1} \varepsilon_v$

In both models, RNN structure with one hidden layer containing 20 nodes was found to generate the minimal sum squared error. Good agreement between modelling results and laboratory experimental data showed the efficiency of the RNN approach in modelling complex soil behaviour. The authors assumed that such an RNN model is applicable to other soils if proper input and output parameters are chosen, although this can not be a very general conclusion based on other researchers findings.

Zhu et al. (1998b) published a similar work in which the same NNCM (in terms of network type, architecture and input-output set) was used to model soil behaviour, using generally the same data as in Zhu et al.(1998a). However in this work the authors proposed that in the network structure one hidden layer with 20 and 35 nodes is suitable for the modelling of the strain controlled undrained tests and stress controlled drained tests respectively.

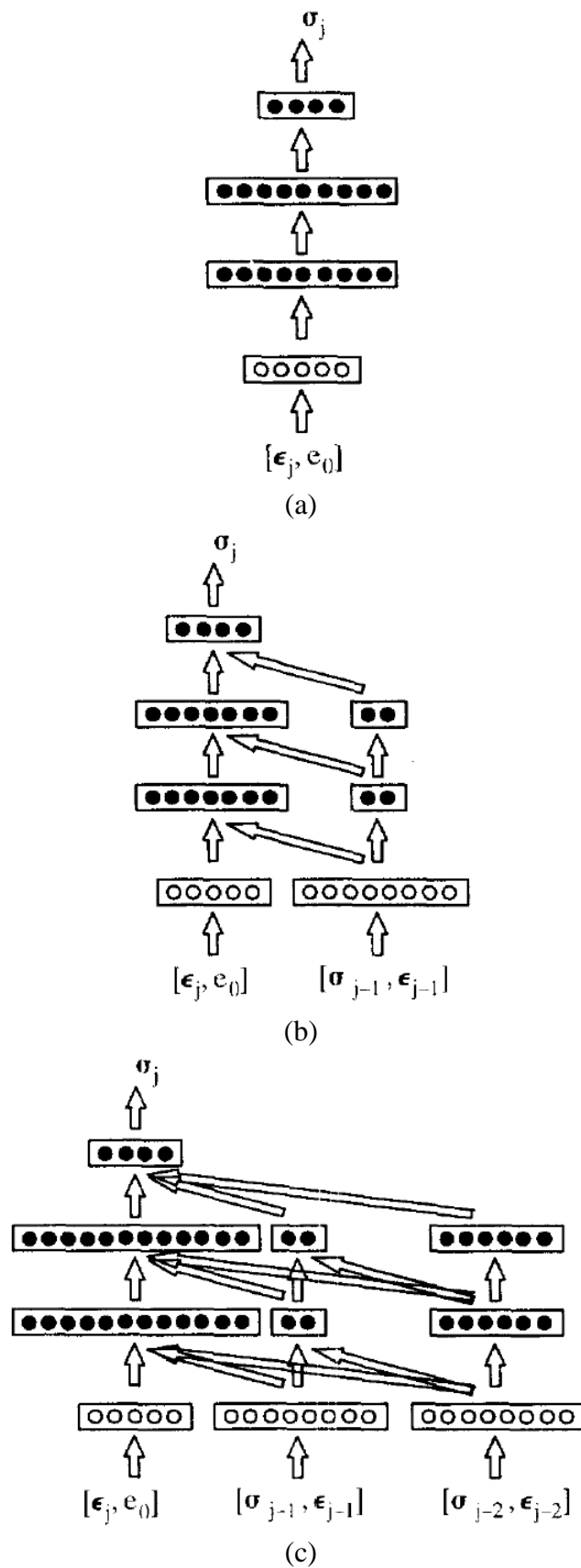
Ghaboussi et al. (1998) described a new indirect method, called autoprogessive training, for training neural network material models from structural tests to learn complex stress-strain behaviour of materials. The global data measured from a structural load-deflection test was used to train the network. The main premise of the work was that the structural tests usually generate a large number of spatial patterns of stresses and strains that can be used for training. The term “autoprogessive training” referred to a process in which the neural network is itself an integral part of the iterative algorithm that is used to create the stress-strain training cases from the global response data. This method differs from common applications of NN models in the sense that there is not a known set of data to train the network, but the material model is extracted iteratively from global measurements using nonlinear finite elements (Haj-Ali et al. 2001). The applications discussed in this paper show a procedure that can be used to create the stress-strain training data for the neural network material model, having knowledge of the global load vs. deflection response of the structure. In contrast to previous applications of neural networks to constitutive modelling, in this method there was not “a priori” a set of directly measured information that accurately represents the material behaviour but this information must be extracted from the recorded structural response. Based on the results of two simple examples explained in this paper, the trained neural network in this way looked consistent, but a true evaluation of the trained neural network material model would require comparisons with measured response data for other structures, constructed of the same material but of different geometrical forms, which had not been used in the training. This was not shown in the paper. Also the minimum number of measured structural responses, and their type and locations on the structure, that are required in order to uniquely determine a neural network material model is an important theoretical issue that remained to be addressed in this paper.

In addition, it should be noted that geotechnical materials (e.g. soils and rocks) exhibit varied and uncertain behaviour due to the complex and imprecise physical processes

associated with the formation of these materials in particular and mediums that can be formed from these materials (e.g., embankments and trenches) in general; this is in contrast to most other civil engineering materials (e.g., steel) and structures (e.g., truss) which exhibit far greater homogeneity and isotropy. Thus such methodology of training a neural network material model is less likely to be effectively applicable to geotechnical problems.

Sidarta and Ghaboussi (1998) modified the earlier Ghaboussi et al. (1998) work in order to develop a neural network based constitutive model for geomaterials using autopgressive training. They used a non-uniform material test which had a non-uniform distribution of stresses and strains within the specimen. Then the measured boundary forces and displacements were applied in a finite element model of the test to generate the input and output data for training the neural network material model. Using the data generated in that way, the autopgressive method was used to train the neural network material model.

Three drained triaxial tests on Sacramento River sand were considered for their work. The tests were performed with end friction condition, and the relative densities of the samples ranged from loose to medium dense to dense. The measured axial forces and confining pressures were directly from the test data. The radial displacements of the outer surface of each sample were determined by assuming a parabolic distribution. These measured force and displacement boundary conditions were used in the autopgressive method. The components of stress and strain, which were required to train the neural network material model, were constructed artificially in the finite element model of the test. The developed neural network material models with their input-output parameters are schematically shown in Figure 2.6.



**Figure 2.6** The autoprogressively trained neural network material models showing (a) neural network material model for loose sand (b) neural network material model for medium dense sand (c) neural network material model for dense sand (Sidarta and Ghaboussi 1998).

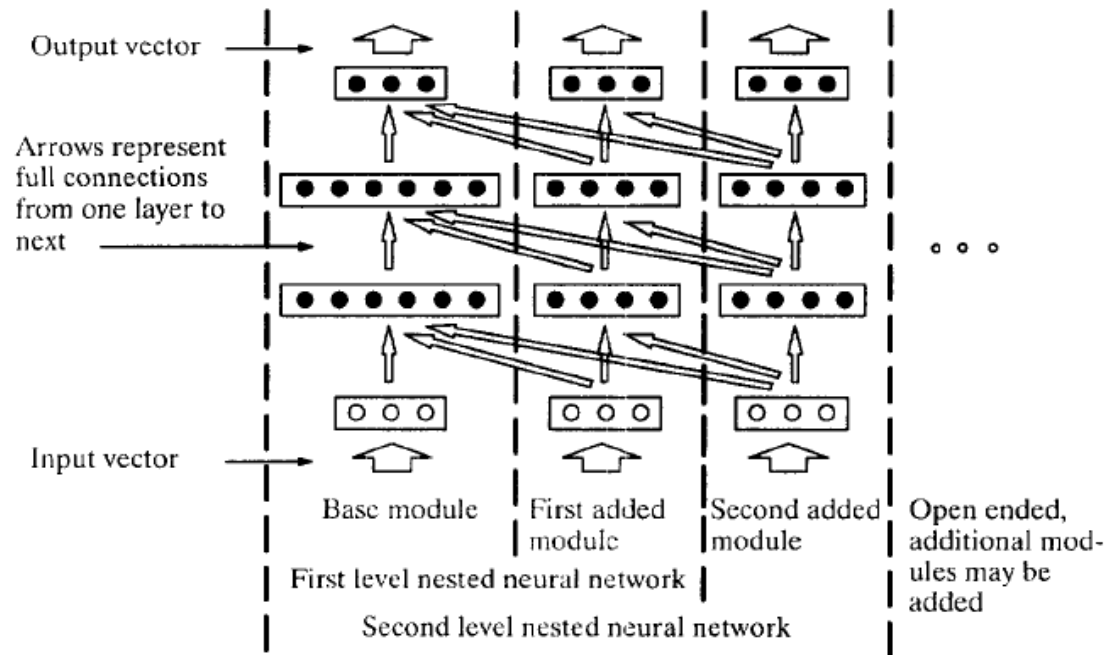
In Figure 2.6,  $\varepsilon$  stands for all components of strain (i.e.  $\varepsilon_{rr}$ ,  $\varepsilon_{zz}$ ,  $\varepsilon_{\theta\theta}$ ,  $\varepsilon_{rz}$ ),  $\sigma$  stands for all components of stress (i.e.  $\sigma_{rr}$ ,  $\sigma_{zz}$ ,  $\sigma_{\theta\theta}$ ,  $\sigma_{rz}$ ),  $e_0$  is the initial void ratio. Also the indices  $j-1$  and  $j-2$  denote the first history point module and the second history point module, respectively. Figure 2.6 clearly indicates that the material behaviour becomes increasingly more complex with increasing soil density.

The trained neural networks were first used in finite element analysis of actual triaxial test with end friction and next they were used in finite element analyses of hypothetical tests with no end friction. The results of the analysis with end friction matched well with those of the actual experiment. However the results of the forward analysis of the hypothetical tests with no end friction showed significant differences with the actual experimental results.

The work presented in this paper, introduced a significant improvement over conventionally trained neural network based constitutive models for geomaterials. The attraction of the non-uniform test, used in this study, is that a range of stress levels and a variety of stress paths may be represented in a single test, therefore the test results contain information on material behaviour for different stress levels and stress paths. If that information could be extracted, then the results of a single non-uniform material test may be sufficient for training a neural network constitutive model and there is no need for a large number of conventional triaxial tests with different stress paths to produce the training data.

Ghaboussi and Sidarta (1998) explained a nested adaptive neural network (NANN) for constitutive modelling. The issue presented in this work was in companion with what was presented in Sidarta and Ghaboussi (1998). The material behaviour data has an inherent structure and one type of such inherent internal structure in data is the nested structure. Basically nested adaptive neural networks take advantage of the nested structure of the material test data, and reflect it in the architecture of the neural network. A nested neural network consists of several modules. The starting point of building a NANN is to develop a base module to represent the material behaviour in the lowest function space in the data structure. This base module is a standard multi-layer feed-forward neural network. The base module is then augmented by attaching added modules to form higher level NANN. The process is theoretically open ended and more and more modules can be added. The added modules themselves are also standard

multi-layer feed-forward neural networks. Figure 2.7 shows a general NANN in symbolic form.



**Figure 2.7** Symbolic representation of a typical nested adaptive neural network (Ghaboussi and Sidarta 1998).

The developed NANNs were applied in modelling of the drained and undrained behaviour of Sacramento River sand, using triaxial compression test results. The objective was to model a material behaviour in both drained and undrained conditions for a range of initial void ratios and initial confining pressures. First a base module was developed and then the history point modules were added. The trained base module together with the two successive levels of NANN which were obtained by attaching the first and second history point modules are

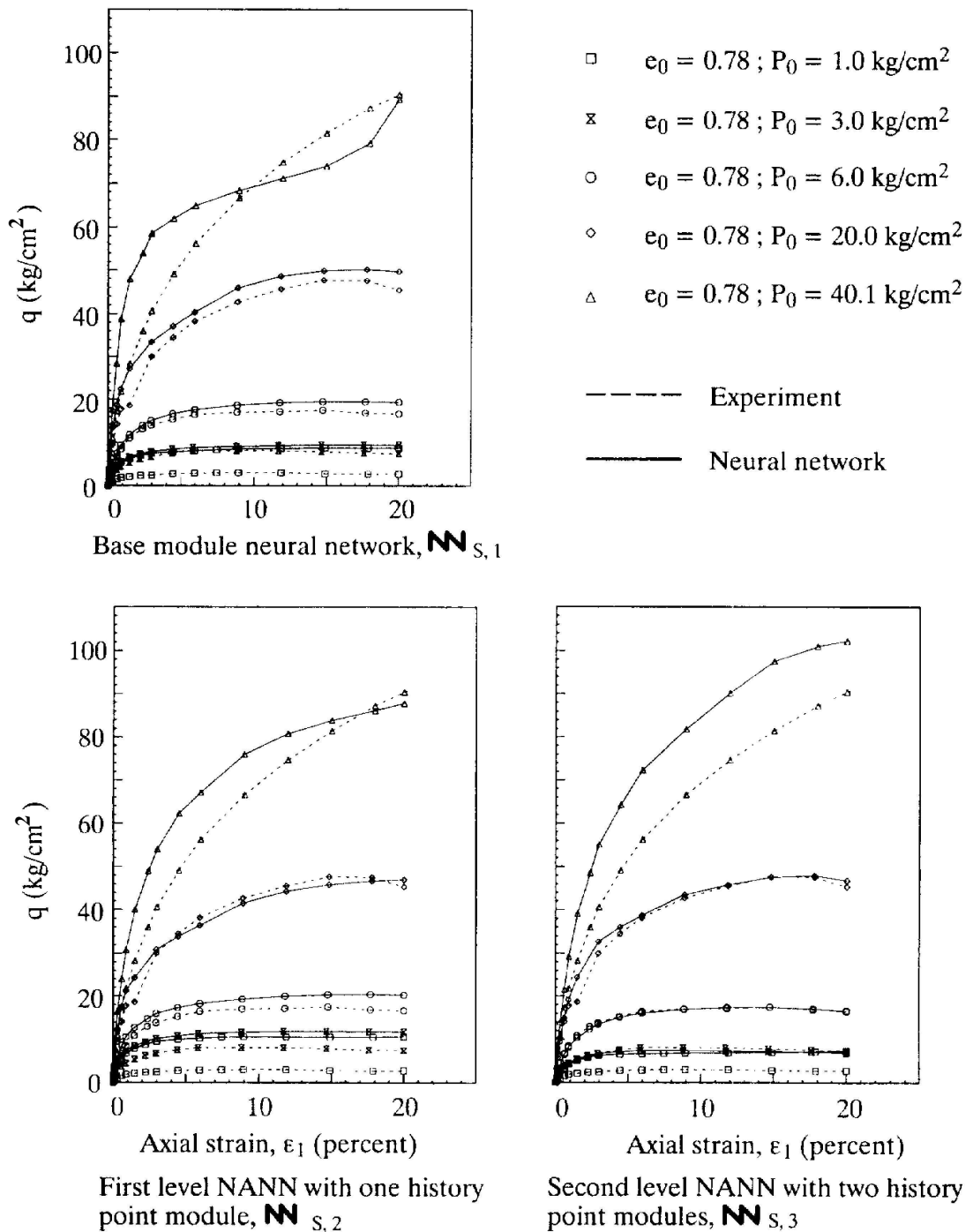
$$\Delta s_j = \Delta s_j \mathbf{NN}_{s,1} (\Delta \varepsilon_j, s_j, \varepsilon_j, e_0 : 8 | 2 - 12 | 2 - 12 | 3) \quad (2.2)$$

$$\Delta s_j = \Delta s_j \mathbf{NN}_{s,2} ([\Delta \varepsilon_j, s_j, \varepsilon_j, e_0], [s_{j-1}, \varepsilon_{j-1}, e_0] : 8, 6 | 2 - 12, 2 - 8 | 2 - 12, 2 - 8 | 3) \quad (2.3)$$

$$\Delta s_j = \Delta s_j \mathbf{NN}_{s,3} ([\Delta \varepsilon_j, s_j, \varepsilon_j, e_0], [s_{j-1}, \varepsilon_{j-1}, e_0], [s_{j-2}, \varepsilon_{j-2}, e_0] : 8, 6, 6 | 2 - 12, 2 - 8, 2 - 30 | 2 - 12, 2 - 8, 2 - 30 | 3) \quad (2.4)$$

In the above equations, the symbol **NN** was introduced to denote the output of a multilayer feed forward neural network. In these equations  $s = \{\sigma, u\}$ ,  $\sigma = \{p', q\}$ ,  $u$  = pore pressure,  $\varepsilon = \{\varepsilon_v, \varepsilon_d\}$ ,  $e_0$  = initial void ratio,  $p' = (\sigma'_1 + 2\sigma'_3)/3$  = mean effective stress,  $q = \sigma_1 - \sigma_3$  = deviatoric stress,  $\varepsilon_v = \varepsilon_1 + 2\varepsilon_3$  = volumetric strain,  $\varepsilon_d = \varepsilon_1 - \varepsilon_3$  = deviatoric strain. In Equations (2.3) and (2.4), indices  $j-1$  and  $j-2$  denote the first history point module and the second history point module, respectively.

In Equation (2.2) the input layer has eight nodes and the output layer has three nodes, also the two hidden layers start with two nodes and during the training their number of nodes adaptively increases to 12. The number of input neurons increases to 14 and 20 in Equation (2.2) and Equation (2.3), respectively. The hidden layers of the second history point added module grew from two to 30 nodes each during the training. This is an indication that the effect of history points on the material behaviour becomes increasingly more complex and difficult for the neural network to learn. This is clear that with increasing the number of history points, the number of inputs can significantly increase, which after even a few steps this can make the network massively complex that can result in much higher time and computational cost. Also there is no indication that such network is able to capture the constitutive relationship effectively and generalise it to new loading conditions, avoiding the overfitting problem. Figure 2.8 shows the performance of the trained nested adaptive neural network on drained triaxial compression tests. Considerable errors, particularly for higher confining pressures, are visible.



**Figure 2.8** Results of trained NANN on drained triaxial compression tests (Ghaboussi and Sidarta 1998).

Another neural network based constitutive relationship to model stress-strain and volume change behaviour of sand and gravel under drained triaxial compression test conditions was presented by Penumadu and Zhao (1999). The NN models presented in this paper were developed based on a large database comprised of nearly 250 triaxial test data collected from literature. Two neural network sand models (Sand-Low and Sand-High) were developed to model the sand test results in the low confining pressure

(less than 700 kPa) and high confining pressure (higher than 700 kPa) range. The division at 700 kPa was chosen arbitrarily by the authors. Also a single model was developed for gravel test results.

A sequential neural network structure (see Figure 2.2) was used and like other NN based models, back-propagation algorithm was employed for training the neural networks. The final optimum neural network architecture had three layers with eleven neurons in input layer, fifteen neurons in the hidden layer and two neurons in output layer. The fifteen hidden units were determined using a trial and error procedure. The selected input-output variables for NN training were

Inputs:  ${}^i\sigma_d$ ,  ${}^i\varepsilon_v$ ,  $\sigma'_3$ ,  ${}^i\varepsilon_1$ ,  ${}^i(\Delta\varepsilon_1)$ ,  $e$ ,  $n_s$ ,  $h$ ,  $D_{50}$ ,  $C_u$ ,  $C_c$

Output:  ${}^{i+1}\sigma_d$ ,  ${}^{i+1}\varepsilon_v$

Seven of the eleven inputs were used to describe the hardness of the mineral ( $h$ ), shape factor ( $n_s$ ), equivalent particle size and their distribution ( $D_{50}$ ,  $C_u$ ,  $C_c$ ), void ratio ( $e$ ) and effective confining pressure ( $\sigma'_3$ ). The current state units of stress and strain were represented with three inputs using deviator stress ( ${}^i\sigma_d$ ), axial strain ( ${}^i\varepsilon_1$ ) and volumetric strain ( ${}^i\varepsilon_v$ ). For a given specimen conditions and current state units the objective of neural network was prediction of two outputs, deviator stress ( ${}^{i+1}\sigma_d$ ) and volumetric strain ( ${}^{i+1}\varepsilon_v$ ) of the next state of an input axial strain increment ( ${}^i(\Delta\varepsilon_1)$ ).

An interesting feature for training the network in this research was that a fixed set of axial strain increments were chosen consistently for all the test data. This means that the value of strain increment was chosen to increase at a constant magnitude (e.g. 0.1%). The original experimental data (deviator stress-axial strain and volumetric strain-axial strain) were not recorded at a specific strain increment. The authors obtained the training pattern corresponding to the considered strain increment by digitalising the data and using cubic spline interpolation (Press et al. 1992). It was observed that the neural network material models obtained in this research were able to represent the constitutive behaviour of cohesionless soil with reasonable accuracy. This NNCM was later used in Penumadu et al. (2000) to simulate triaxial tests.

Habibagahi and Bamdad (2003) used neural network to describe the mechanical behaviour of unsaturated soils. A multilayer perceptron, sequential architecture with feed back capability was chosen for their network. Triaxial test results on Lateritic gravel, reported by Toll (1988), were used as database. The final network which was obtained through a trial and error procedure had three layers with 9 neurons in the input layer and three neurons in the output layer. The optimal number of nodes in the hidden layer was worked out to be five. The input-output parameters set for this first neural network constitutive model for unsaturated soils were

Inputs:  ${}^i q$ ,  ${}^i \varepsilon_v$ ,  ${}^i E(U_a - U_w)$ ,  ${}^i \varepsilon_1$ ,  ${}^i (P - U_a)$ ,  ${}^i (P - U_w)$ ,  $(U_a - U_w)$ ,  $S_r$ ,  $\rho_d$  and  $\theta$

Outputs:  ${}^{i+1} q$ ,  ${}^{i+1} \varepsilon_v$  and  ${}^{i+1} E(U_a - U_w)$

In the input parameters set, four neurons, namely, soil water content  $\theta$ , dry density  $\rho_d$ , degree of saturation  $S_r$ , and soil suction  $(U_a - U_w)$ , represent the initial condition of the specimen before shearing. The other six neurons, namely, axial strain  ${}^i \varepsilon_1$ , change in suction  ${}^i E(U_a - U_w)$ , mean effective stress with respect to pore air and water pressures ( ${}^i (P - U_a)$  and  ${}^i (P - U_w)$ ), volumetric strain  ${}^i \varepsilon_v$ , and deviatoric stress  ${}^i q$  are the input variables that must be updated incrementally during training based on the outputs received from the previous increment of training.

It was shown that the trained network was able to model the mechanical behaviour (stress-strain, volume change and change in suction) of unsaturated soils with reasonable accuracy. The authors also proposed that the model may be used to simulate triaxial tests (artificial tests) under similar conditions as those of the source of training database; this could be time and cost saving compared to sophisticated laboratory experiments.

In addition to the works that have already been mentioned, some other researchers have also applied NN for constitutive modelling of geomaterials using different datasets (Banimahd et al. 2005; Najjar et al. 1999; Wu et al. 2001). The results of these works also show the capability of NN in stress-strain prediction of different soils.

## 2.4 NNCM implementation into finite element method

As has been described in the previous section, to date, many researchers attempted to model the various modes of the constitutive behaviour of geomaterials with neural networks. Although these works are different in terms of their details and terminology; however most of their results have indicated that NNs can reasonably represent these materials response under different load paths. From this, in theory it is seen that, in a numerical analysis tool such as FE, it is possible to replace a conventional (analytical) constitutive model with a suitably trained NN. However the concentration of most of the investigations has been on the description of the constitutive behaviour itself, as a result little is known about the performance of NNCMs in engineering analysis. The main reason for this appears to be the fact that there are considerable difficulties in incorporating a general NNCM in finite element codes (Shin and Pande 2002).

Shin and Pande (2000) presented a self learning FE code in which a NNCM was used in the code instead of conventional constitutive models and showed that the application of a constitutive law in the form of a neural operator leads to some qualitative improvement in the application of FE in engineering practice. They presented a procedure where data for training neural network based constitutive model (NNCM) were acquired from planned monitoring of structural tests. Indeed, unlike conventional procedures where generally material testing is performed to extract the stress-strain relationship and identify material parameters, in their procedure inverse analyses were carried out to identify material parameters from monitored global structural response. In this way the self learning capability of the software was expected; however for this purpose the results of structure behaviour needed to be available in advance. It is obvious that depending on the mesh size of the problem under consideration, huge amount of data may be accumulated with increasing the number of self learning cycles which can result in severe computer storage and CPU time problems during training. For this problem a limited number of monitoring points in the structure are selected and the data corresponding to these points are used to train the NNCM. Selection of the number and location of monitoring points is therefore of considerable importance in identifying a reliable NNCM. It was stated that such trained NNCM will need to be treated with caution for modelling the behaviour of other structures; as it is apparent that a NNCM may predict the correct response at a few points, yet may be completely inadequate to predict the response at others.

Shin and Pande (2001) showed that in their self learning intelligent finite element code the tangential constitutive matrix of the material can be computed as it is possible to obtain partial derivatives of the neural network, which has been trained through total stress and strain data. The capabilities of the developed intelligent FE code were illustrated by analysing a rock specimen under uniaxial cylindrical compression (with fixed ends). Shin and Pande (2002) proposed a strategy to generate additional data from general homogeneous material tests in order to train NNCM. This was done by taking advantage of isotropy when it is applicable to the material under consideration. A boundary value problem of a circular cavity in a plane stress plate was modelled with the intelligent FE code using NNCM trained with the enhanced dataset. The intelligent FE analyses showed comparable results with FE analyses using conventional constitutive models.

Drakos and Pande (2006) presented a NNCM and stated that the model is equivalent of the hardening soil model. Synthetic data for training the NNCM was generated using the Hardening Soil Model (HSM) available in the commercial software PLAXIS and choosing a set of arbitrary parameters, typical of sands, for the HSM. The performance of the trained NNCM was then validated by using this model instead of HSM model for numerical analysis of two simple foundation and excavation problems.

Also Lefik and Schrefler (2003) used a neural network for constitutive modelling of nonlinear material behaviour and highlighted some of the difficulties in the constitutive description in incremental form.

Hashash et al. (2004) described some of the issues related to the numerical implementation of a NNCM in finite element analysis and derived a closed-form solution for material stiffness matrix for the neural network constitutive model.

Javadi and his co-workers carried out extensive research on application of neural networks in constitutive modelling of complex materials in general and soils in particular. They have developed an intelligent finite element method (NeuroFE code) based on the incorporation of a back-propagation neural network (BPNN) in finite element analysis. The intelligent finite element model was then applied to a wide range of boundary value problems including several geotechnical engineering applications (2004a; 2005; 2004b; 2003; 2002) and has shown that NNs can be very efficient in

learning and generalising the constitutive behaviour of complex materials such as soils, rocks and others.

Although it has been shown by various researchers that NNs offer great advantages in constitutive modelling of materials in finite element analysis; however, despite their good performance on the available data, these networks have some shortcomings. One of the disadvantages of the NNCM is that the optimum structure of the NN (such as number of inputs, hidden layers, transfer functions, etc.) must be identified a priori which is usually obtained using a time consuming trial and error procedure. Also, the main drawback of the NN approach is the large complexity of the network structure, as it represents the knowledge in terms of a weight matrix together with biases which are not accessible to user. In other words NN models give no clue on the way inputs affect the output and are therefore considered as a black box class of model. The lack of interpretability of NN models has inhibited them from achieving their full potential in real world problems (Lu et al. 2001).

# CHAPTER 3

## FINITE ELEMENT METHOD

### 3.1 Introduction

This section gives insight into the finite element (FE) method. It covers the history and background of finite element analysis.

The finite element method is a numerical analysis technique for obtaining approximate solutions to a wide variety of engineering problems (Huebner et al. 2001). This method was designed to study stresses in airframe structures and then adapted to a wider field of mechanics. Most of the problems encountered in engineering design and analysis can be expressed in terms of differential equations that describe the response of a physical system subjected to external influences. The solutions to most of such equations are too complex for analytical methods and are commonly obtained using approximate numerical techniques. Finite elements are used to solve a complex problem by dividing the problem into smaller problems and solving them separately. Thus this method looks at a model as made up of small inter-connected sub-regions or elements. The idea of the finite element method is that a “solution region” i.e., a model can be analysed or approximated by replacing the region with a finite number of distinct elements. These elements can then be placed in different ways to make up complex problems.

## 3.2 Finite element procedure

In general the solution procedure for a continuum problem using the finite element method involves the following basic steps (Cheung et al. 1996):

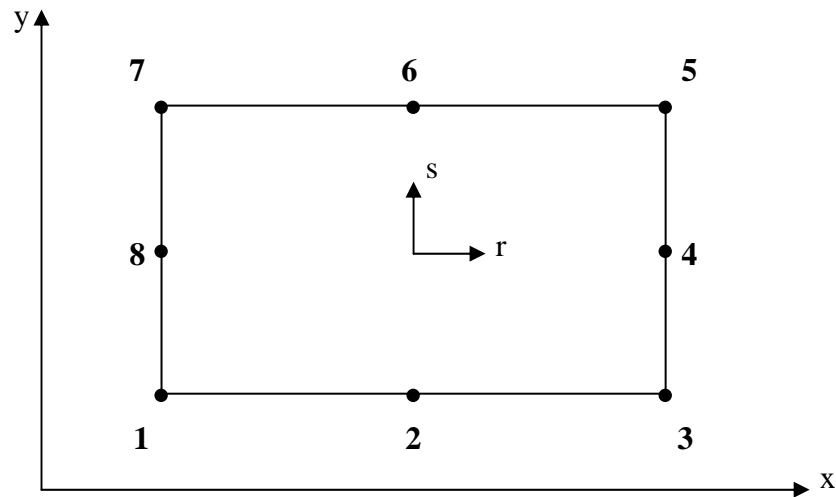
- i) Discretising the problem domain into a number of sub-regions known as finite elements. The field variables are assigned at the nodes of each element with the nodal values of these field variables being the unknown parameters of the problem.
- ii) Selection of element interpolation functions to represent variation of the field variables over the element.
- iii) Evaluation of individual element properties. This involves approximating the governing differential equations using a simpler system of algebraic equations over the element domain. The approximation is commonly achieved using either variational techniques or weighted residual approaches. Galerkin's weighted residual approach is adopted in this research for its simplicity and accuracy.
- iv) Formation of elements stiffness matrix.
- v) Assembling the element properties to obtain the system equations that will represent the overall system.
- vi) Imposing boundary conditions to modify the global system of equations using the known values of the nodal variables at the continuum boundary.
- vii) Solving the system of equations for the unknown nodal variables using conventional numerical analysis techniques.
- viii) Finally, further computations to evaluate additional important parameters such as strains, stresses and other physically meaningful quantities from the computed nodal variables and element properties.

## 3.3 General formulation

### 3.3.1 Basics of FE Formulation

The basic idea behind the finite element method is to divide the structure, body or region being analysed into a suitable number of elements with associated nodes and to choose the most appropriate element type to model most closely the actual physical behaviour. The number of elements used and their variation in size and type within a given region are primarily matters of engineering judgment. These elements may be one, two or three dimensional. Discretisation results in the specification of the finite element mesh and involves two distinct but related tasks: nodes definitions and elements definitions. The nodes are always numbered consecutively from one to the total number of nodes present. The nodal numbering pattern has a strong influence on execution time in a computer program (for large problems). Usually the nodes are numbered in such a way so as to minimise the bandwidth of the assemblage matrix. Node definition completes when the coordinates of each of the nodes are also specified. The element numbering scheme is completely arbitrary. To define the elements, one needs to number them consecutively from one to the maximum number of the elements present.

The nodes associated with each element must be specified. In addition, the material property data to be used for each element should be specified (Stasa 1985). The choice of appropriate element for a particular problem is one of the major tasks that must be carried out by analyst. However, the elements must be small enough to give usable results and yet large enough to reduce computational effort. Small elements are generally desirable where the results are changing rapidly, such as where changes in geometry occur; large elements can be used where results are relatively constant (Daryl 2002). Figure 3.1 shows a quadratic, eight-node, rectangular element, which has been used in the developed FE programs for this work, associated with eight nodes per elements, three at each side of the element. Such an element has two unknowns (i.e., degrees of freedom) for uncoupled problems and three unknowns for coupled problems of flow and deformation in a region.



**Figure 3.1** The rectangular element (quadratic order).

For the deformation based finite element procedure a set of so called shape functions has to be considered so that they exclusively define the state of deformation within each element in terms of its nodal displacement values. The shape functions for a typical eight-node element are given below in terms of the serendipity coordinates  $r$  and  $s$  (Stasa 1985)

$$N_1 = \frac{1}{4}(1-r)(1-s)(-r-s-1)$$

$$N_2 = \frac{1}{4}(1-r^2)(1-s)$$

$$N_3 = \frac{1}{4}(1+r)(1-s)(r-s-1)$$

$$N_4 = \frac{1}{2}(1+r)(1-s^2)$$

$$N_5 = \frac{1}{4}(1+r)(1+s)(r+s-1)$$

$$N_6 = \frac{1}{2}(1-r^2)(1+s)$$

$$N_7 = \frac{1}{4}(1-r)(1+s)(-r+s-1)$$

$$N_8 = \frac{1}{2}(1-r)(1-s^2)$$

(3.1)

So a typical shape function can be written as

$$N_i = N_i(r, s) \quad (3.2)$$

where,

$$r = r(x, y)$$

$$s = s(x, y)$$

The components of displacement,  $\bar{u}$  and  $\bar{v}$ , at any point within an element can be defined in terms of their nodal values.

$$\begin{Bmatrix} \bar{u} \\ \bar{v} \end{Bmatrix} = [N] \begin{Bmatrix} u \\ v \end{Bmatrix}_{nodal} \quad (3.3)$$

The approximation of the displacement by Equation (3.3) makes it possible to formulate the equilibrium equation for each element which can then be used to describe the element's deformational behaviour. Assuming plane strain conditions apply, the strains within the element are related to the nodal displacements by Equation (3.4).

$$\begin{aligned} \Delta \varepsilon_x &= \frac{\partial(\Delta u)}{\partial x} \\ \Delta \varepsilon_y &= \frac{\partial(\Delta v)}{\partial y} \\ \Delta \gamma_{xy} &= \frac{\partial(\Delta u)}{\partial y} + \frac{\partial(\Delta v)}{\partial x} \\ \Delta \varepsilon_z &= \Delta \gamma_{xz} = \Delta \gamma_{zy} = 0 \\ \{\Delta \varepsilon\}^T &= \{\Delta \varepsilon_x \ \Delta \varepsilon_y \ \Delta \gamma_{xy}\} \end{aligned} \quad (3.4)$$

Replacing the displacements obtained from the approximations given by Equation (3.3), the following relationship is acquired.

$$\{\Delta \varepsilon\} = [B] \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix}_{nodal} = [B] \{\Delta d\}_{nodal} \quad (3.5)$$

The matrix [B] in Equation (3.5) contains the derivatives of the shape functions  $N$ . The derivatives of a typical shape function,  $N_i$ , can be written as

$$\frac{\partial N_i}{\partial r} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial r} \quad (3.6a)$$

and,

$$\frac{\partial N_i}{\partial s} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial s} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial s} \quad (3.6b)$$

or in matrix form as

$$\begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (3.7)$$

The  $2 \times 2$  matrix on the right hand side of Equation (3.7) is known as the “Jacobian matrix” and is denoted by  $J$ , or

$$J = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \quad (3.8)$$

The infinitesimal area element  $dx dy$  is related to an infinitesimal area element in the  $(r, s)$  coordinates by

$$dxdy = |\det J| drds \quad (3.9)$$

Using the appropriate constitutive relationship, the stresses can now be determined as

$$\{\Delta\sigma\} = [D]\{\Delta\varepsilon\} \quad (3.10)$$

where  $\{\Delta\sigma\}$  is the vector of stress components and  $[D]$  is the constitutive matrix.

For a single element the primary variable, displacement, has now been related to the two secondary variables, stress and strain. These relationships can then be used to calculate the incremental potential energy of a single element, which is defined as

$$\text{Incremental potential energy } (\Delta E) = \text{Incremental Strain energy } (\Delta W) - \text{Incremental work done by applied loads } (\Delta L) \quad (3.11)$$

The incremental strain energy in an element and the incremental work done by external loads are given by Equations (3.12) and (3.13) respectively.

$$\Delta W = \frac{1}{2} \int \{\Delta\varepsilon\}^T \{\Delta\sigma\} dVol \quad (3.12)$$

$$\Delta L = \int_{Vol} \{\Delta d\}^T \{\Delta F\} dVol + \int_{Surface} \{\Delta d\}^T \{\Delta T\} dSurface \quad (3.13)$$

where for problems with two degrees of freedom at each node

$$\begin{aligned} \{\Delta d\}^T &= \{\Delta u, \Delta v\} = \text{displacements} \\ \{\Delta F\}^T &= \{\Delta F_x, \Delta F_y\} = \text{body forces} \\ \{\Delta T\}^T &= \{\Delta T_x, \Delta T_y\} = \text{surface tractions} \end{aligned}$$

Equilibrium requires that a body's potential energy is minimum. The condition when this occurs can be found by differentiating the expression for incremental potential energy and setting it to zero. The stress and strain terms in Equation (3.12) can now be replaced by the approximations found previously and the resulting expression along with Equation (3.13) substituted into Equation (3.11) to give the potential energy in terms of the unknown nodal displacements. The minimum potential energy is then found by summing up the potential energy of all the individual elements and then differentiating this expression with respect to the nodal displacements. The result is Equation (3.14), which becomes zero when the expression in square brackets is equal to zero. This expression, when written in the form of Equation (3.15), represents the equilibrium equation for the finite element assemblage.

$$\delta \Delta E = \sum_{i=1}^N \left( \{\delta \Delta d\}_n^T \right)_i \left[ \int_{Vol} [B]^T [D][B] dVol \{\Delta d\}_n - \int_{Vol} [N]^T \{\Delta F\} dVol - \int_{Surface} [N]^T \{\Delta T\} dSurface \right]_i = 0 \quad (3.14)$$

$$\sum_{i=1}^N [K_E]_i (\{\Delta d\}_n)_i = \sum_{i=1}^N \{\Delta R_E\}_i \quad (3.15)$$

where

$$[K_E] = \int_{Vol} [B]^T [D][B] dVol = \text{Element stiffness matrix.}$$

$$\{\Delta R_E\} = \int_{Vol} [N]^T \{\Delta F\} dVol + \int_{Surface} [N]^T \{\Delta T\} dSurface = \text{Right hand side load vector.}$$

$$\{\Delta d\}_n = \text{Vector of unknown displacements.}$$

In Equation (3.9) the determinant of the Jacobian matrix is referred to as  $|\det J|$  which can be replaced by

$$|J| = \frac{\partial x}{\partial r} \frac{\partial y}{\partial s} - \frac{\partial y}{\partial r} \frac{\partial x}{\partial s} \quad (3.16)$$

Also the volume integral in Equation (3.14) can be transformed into the global coordinate system by using the determinant of the Jacobian matrix as

$$dVol = t \, dx \, dy = t \, |J| \, dr \, ds \quad (3.17)$$

where  $t$  is the thickness of the element. Therefore the stiffness matrix of a single element can be calculated in terms of the serendipity coordinates  $r$  and  $s$  as

$$[K_E] = \int_{-1}^1 \int_{-1}^1 t [B]^T [D] [B] |J| \, dr \, ds \quad (3.18)$$

The overall stiffness matrix of the finite element mesh can then be found by summing the contribution of each individual element. Before integrating Equation (3.18), the  $[B]$  matrix which contains the derivatives of the shape functions must be transformed into the natural coordinate system. This requires the use of the chain rule.

If Equation (3.7) is pre-multiplied by  $J^{-1}$ , the desired result will be

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \end{bmatrix} \quad (3.19)$$

Since quadratic order rectangular elements are considered as isoparametric element, it can be written as

$$x = \sum N_i x_i \quad (3.20a)$$

and,

$$y = \sum N_i y_i \quad (3.20b)$$

where, the assumptions are made over the total number of nodes present in the element and  $x_i, y_i$  are the coordinates of the nodes. Therefore, the Jacobian matrix becomes

$$J = \begin{bmatrix} \sum \frac{\partial N_i}{\partial r} x_i & \sum \frac{\partial N_i}{\partial r} y_i \\ \sum \frac{\partial N_i}{\partial s} x_j & \sum \frac{\partial N_i}{\partial s} y_j \end{bmatrix} \quad (3.21)$$

### 3.3.2 Determination of the local element characteristics

Element characteristics mean the element stiffness matrices and nodal unknown vectors. The word “local” refers to the fact that the element characteristics are derived in a local reference system, which usually change from element to element and determined numerically for each element. The element characteristics, the local stiffness matrices and nodal unknown vectors may be determined numerically for each element (Stasa 1985).

### 3.3.3 Transformation of the element characteristics

The element characteristics are transformed from the local coordinate system to the global system. The transformation of the local element characteristics needs to be performed only when the unknown parameter function is a vector such as the (nodal) displacements, and then only when the local coordinate system is used (Stasa 1985).

### 3.3.4 Assemblage of the global element characteristics

The global element stiffness matrices and global element nodal force vectors must be assembled to form the assemblage element stiffness matrix and nodal unknown vector to find the properties of the overall system modelled by the network of elements. The matrix equations for the system have the same form as the equations for an individual element except that they contain many more terms because they include all nodes. The unknown parameters functions have the same value at any given node regardless of the element containing (Stasa 1985).

### 3.3.5 Imposition of the boundary conditions

The boundary conditions of the problem must be considered to modify the system of equations and prepare them for the solution phase. At this stage, known nodal values of the dependent variables are imposed (Huebner et al. 2001).

### 3.3.6 Solution

The assembly process gives a set of simultaneous equations that must be solved to obtain the unknown nodal values of the problem. For engineering applications of the finite element formulation presented in section 3.3.1, the material behaviour can be assumed to be linear or nonlinear depending on the material parameters used in the assembly of constitutive matrix,  $[D]$ . For each case the solution may be obtained by any of the methods suitable to a system of algebraic equations.

In linear finite element analysis, one of the most popular methods to solve the system of algebraic equations is direct Gauss elimination method. However for nonlinear problems a direct solution of the system of equations is generally impossible and an iterative scheme must be adopted (Owen and Hinton 1980).

#### 3.3.6.1 Gaussian elimination and back substitution

Gaussian elimination is the name given to a well known method of solving simultaneous equations by successively eliminating unknowns. In this work, Gaussian elimination and back substitution method of solution has been used to find the final values of unknown nodal vectors for linear problems; it has also been used to solve the system of equations at every iteration, for nonlinear problems. The general concept of Gaussian elimination and back substitution can be described by considering a linear system of simultaneous equations in a matrix form as

$$Ax = b$$

where,  $A$  is  $(n \times n)$ ,  $b$  and  $x$  are  $(n \times 1)$ . If  $\det A \neq 0$ , then both sides of the above equation are multiplied by  $A^{-1}$  to write the unique solution for  $x$  as  $x = A^{-1}b$ . However, the explicit construction of  $A^{-1}$ , say by the cofactor approach is computationally expensive and prone to round off errors. Instead, an elimination scheme is preferred. The Gaussian elimination approach for solving  $Ax = b$  is discussed below. The Gaussian elimination process can be considered as an algorithm suitable for computer implementation as following

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{1j} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{2j} & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{3j} & a_{3n} \\ & & & & \\ & & & & \\ & & & & \\ a_{i1} & a_{i2} & a_{i3} & a_{ij} & a_{in} \\ & & & & \\ a_{n1} & a_{n2} & a_{n3} & a_{nj} & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \\ \\ \\ x_i \\ \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \\ \\ \\ b_i \\ \\ b_n \end{bmatrix} \quad (3.22)$$

Gaussian elimination is a systematic approach to successively eliminate variables  $x_1, x_2, x_3, \dots, x_{n-1}$  until only one variable,  $x_n$ , is left. This results in an upper triangular matrix with reduced coefficients and reduced right side. This process is called forward elimination. It is then easy to determine  $x_n, x_{n-1}, \dots, x_3, x_2, x_1$  successively by the process of back-substitution. The first step, with  $A$  and  $b$  written as follows

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{1j} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{2j} & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{3j} & a_{3n} \\ & & & & \\ & & & & \\ & & & & \\ a_{i1} & a_{i2} & a_{i3} & a_{ij} & a_{in} \\ & & & & \\ a_{n1} & a_{n2} & a_{n3} & a_{nj} & a_{nn} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \\ \\ \\ b_i \\ \\ b_n \end{bmatrix} \quad \text{Start of step } k = 1 \quad (3.23)$$

The reduction process at this step is

$$a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}} \cdot a_{1j} \quad (3.24)$$

$$b_i^{(2)} = b_i - \frac{a_{i1}}{a_{11}} \cdot b_1 \quad (3.25)$$

The reduction is carried out for all the elements in the shaded area in Equation (3.23) for which  $i$  and  $j$  range from 2 to  $n$ . In this way the elements in rows 2 to  $n$  of the first column are changed to zero since  $x_1$  is eliminated at the start of the second step. In the computer implementation there is no need to set them to zero. The second step,  $A$  and  $b$  can be written as follows

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{1j} & a_{1n} \\ 0 & a_{22}^{(1)} & a_{22}^{(1)} & a_{2j}^{(1)} & a_{2n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{33}^{(2)} & a_{3j}^{(2)} & a_{3n}^{(2)} \\ & & & & \\ 0 & a_{i2}^{(1)} & a_{i3}^{(1)} & a_{ij}^{(1)} & a_{in}^{(1)} \\ & & & & \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & a_{nj}^{(1)} & a_{nn}^{(1)} \end{bmatrix} \begin{matrix} \\ \\ \text{Start of step } k = 2 \\ \\ \\ \end{matrix} \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(1)} \\ \\ b_i^{(1)} \\ \\ b_n^{(1)} \end{bmatrix} \quad (3.26)$$

The elements in the shaded area above are reduced at this step. At step  $k$ , elements in the shaded area below are reduced

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{1j} & a_{1n} \\ 0 & 0 & a_{22}^{(1)} & a_{2j}^{(1)} & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{3j}^{(2)} & a_{3n}^{(2)} \\ & & & & \\ 0 & 0 & 0 & a_{k+1,k+1}^{(k-1)} & a_{k+1,j}^{(k-1)} & a_{k+1,n}^{(k-1)} \\ 0 & 0 & 0 & a_{i,k+1}^{(k-1)} & a_{ij}^{(k-1)} & a_{in}^{(k-1)} \\ & & & & & \\ 0 & 0 & 0 & a_{n,k+1}^{(k-1)} & a_{nj}^{(k-1)} & a_{nn}^{(k-1)} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \text{Start of step } k \\ \\ \\ \end{matrix} \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \\ b_{k+1}^{(k-1)} \\ b_i^{(k-1)} \\ \\ b_n^{(k-1)} \end{bmatrix} \quad (3.27)$$

The general reduction scheme with limits on indices may be written as follows

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \cdot a_{kj}^{(k-1)} \quad i, j = k + 1, \dots, n \quad (3.28)$$

After  $(n-1)$  steps



$$[H]\{\Delta d\} + \{F\} = 0 \quad (3.32)$$

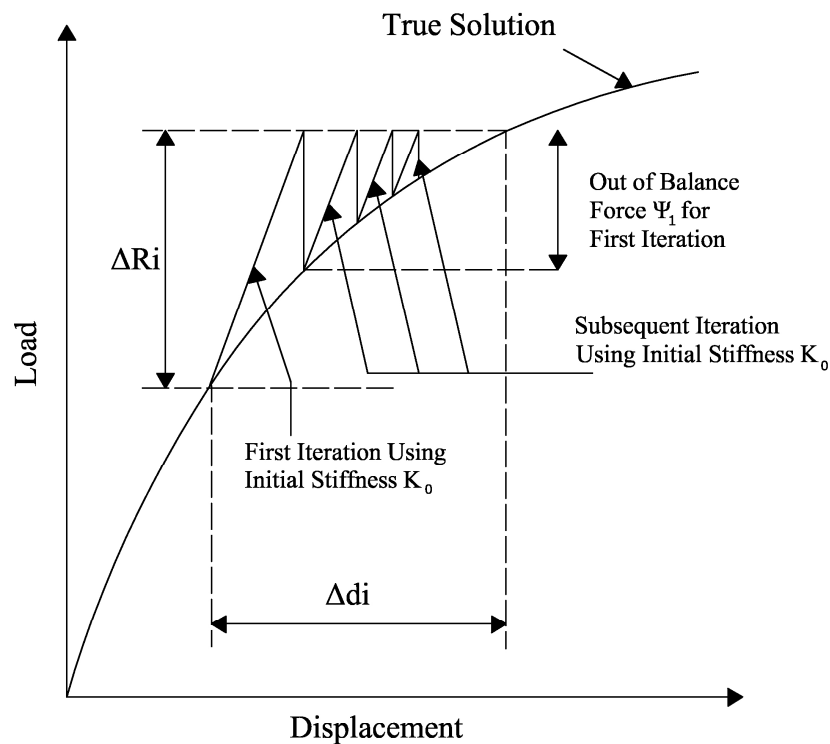
in which  $\{\Delta d\}$  is the vector of unknown displacements,  $\{F\}$  is the vector of applied loads and  $[H]$  is the assembled stiffness matrix. Many techniques have been proposed for the solution of non-linear finite element analyses. One thing they all have in common is that to overcome the constantly changing behaviour of the nonlinear material, the analysis is discretised into a series of increments. The most popular iterative techniques are the direct iteration (Booth 1966), the tangent stiffness (Britto and Gunn 1987) and the Newton-Raphson methods. The disadvantage of the direct iteration method is that convergence of the solution scheme is not guaranteed and can not be predicted at the initial solution stage (Owen and Hinton 1980). The second method is generally thought to be inferior, as the results obtained when using it depend on the size of the increment assumed in the analysis.

The Newton-Raphson method is relatively insensitive to the increment size and is the default non-linear solution technique used in this thesis. In this method a system of residual forces is assumed to exist, so that

$$\psi = [H]\{\Delta d\} + \{F\} \neq 0 \quad (3.33)$$

The residual forces  $\psi$  can be interpreted as a measure of the departure of Equation (3.32) from equilibrium. Since  $[H]$  is a function of  $\{\Delta d\}$  and possibly its derivative, then at any stage of the process  $\psi$  is a function of  $\{\Delta d\}$  too. For the first increment the technique begins by solving the global system of equations assuming the initial global stiffness matrix is determined in the same manner as described in Section 3.3.1. If the material behaviour was linear elastic the solution found would be correct as the stiffness matrix would be constant during the increment. However, for the general case the material behaviour is not linear and therefore the solution will not be correct because the material properties will vary during the increment. The nodal displacements, and therefore the element strains calculated from the stiffness assumed for the first iteration will not satisfy the constitutive model. The constitutive model is then integrated along the incremental strain paths to obtain an estimate of the stress changes. These stress changes are added to the stresses at the beginning of the increment and then integrated to give equivalent nodal forces which are then compared to the externally applied loads.

Because the first iteration did not give the correct solution, equilibrium will not be satisfied and there will be a finite difference. This difference, known as the out of balance forces, is then added to the right hand side load vector. The stiffness matrix is then recalculated and the system of equations solved again including the out of balance forces found from the previous iteration. This technique is repeated until the correct solution is found. It is possible to make the technique more efficient by using the same stiffness matrix for each iteration. This saves a considerable amount of computation effort as for each increment the stiffness matrix has to be evaluated and inverted only once, at the expense of possibly having to use more iterations to converge on the correct solution. The technique, known as the modified Newton-Raphson method, is illustrated graphically in Figure 3.2.



**Figure 3.2** Graphical representation of the modified Newton-Raphson scheme.

The accuracy of the Newton-Raphson method is dependent upon the precision with which the incremental stresses are determined. Performing this calculation is not straightforward as the way in which the material properties, and hence the stresses and strains, vary during an increment are not known. The mathematical procedure that performs this task is called a stress point algorithm and there are many available in the literature. The technique used in this work is a substepping stress point algorithm in which the strains are assumed to vary linearly during the increment and are then divided

into smaller sub-steps. For each sub-step the stresses are found by integrating the constitutive equations using a numerical integration scheme. The default option in this work and chosen for the analyses presented in this thesis is the modified Euler method. The overall stress change during an increment is then found by summing the stress changes found from each sub-step. Due to the iterative nature of the solution technique required to solve non-linear finite element problems, the results will never correspond exactly to the correct solution. The Newton-Raphson method described here could iterate indefinitely but may never reach the true solution. For this reason it is necessary to define a convergence criteria, that when satisfied tells the algorithm that the solution found is accurate to an acceptable degree and the analysis can now move on to the next increment. Of course the true solution is not known; the convergence criterion employed in this work is based on the values of the residual forces and applied forces as

$$\frac{\sqrt{\left[ \sum_{i=1}^N (\psi_i^r)^2 \right]}}{\sqrt{\left[ \sum_{i=1}^N (F_i)^2 \right]}} \times 100 \leq Tolerance \quad (3.44)$$

where  $N$  is the total number of degrees of freedom in the problem,  $r$  denotes the iteration number. The convergence occurs if the norm of the residual forces becomes less than “Tolerance” times the norm of the total applied force. Also the tendency of the norm of the residual force gives an indication of the convergence of the solution. If the value of the norm of the residual force is seen to increase, the solution has begun to diverge from the true solution and the analysis should be aborted. If the value of the norm of the residual force continues to reduce, this is an indication that the solution is converging.

### 3.4 Constitutive relationships

Over the years many different soil models have been developed to capture important aspects of soil behaviour. Although all these models vary in terms of their approach but the common point among them is that they all invoke a nonlinear soil response. The finite element codes developed for this work contain nonlinear elastic constitutive model (Duncan and Chang 1970), elastic-plastic constitutive models with linear strain hardening (Mohr-Coulomb and Drucker-Prager) and nonlinear elastic-plastic strain hardening constitutive model (Modified Cam clay). The details and formulations of

these practiced constitutive relationships can be found in many finite element reference books (e.g., Owen and Hinton 1980).

### **3.5 Coupled stress-strain-flow analysis in saturated porous media**

The finite element theory presented in the previous section is generally applicable to any static problem of structural mechanics. However, there are some features unique to geotechnical engineering that must be added to the theory before any real problems can be analysed. The most significant difference between saturated soil and other engineering materials is that it is composed of two separate phases, the soil skeleton and the fluid which fills the pores between the individual soil particles. In soil mechanics the term consolidation describes the volumetric deformations of soil that occur due to changes in effective stress. The rate at which consolidation occurs in saturated soil is determined by the rate at which the pore fluid can flow out of the soil. Hence the processes of deformation and pore fluid flow in soil are coupled. Consolidation is important in many geotechnical problems such as, foundations, excavations and embankments. It is therefore often important to include coupled consolidation finite element analyses of geotechnical problems.

#### **3.5.1 Principle of effective stress**

All porous geomaterials are composed of solids and pore fluids. Effective stress mechanics involves the “explicit consideration of the portions of the material stress carried by soil or rock skeleton and by the fluid phase constituents”. Effective stress is the portion of the material stress that is transmitted through the soil phase of the porous skeleton. The pore fluid pressure or pore pressure is the portion of the material stress transmitted through the pore fluids. In this work it is assumed that the only pore fluid constituent is water and it does not transmit shear stresses.

$$\{\Delta\sigma\} = \{\Delta\sigma'\} + \{\Delta p\} \quad (3.35)$$

where

$\{\Delta\sigma\}$  = the change in total stress

$\{\Delta\sigma'\}$  = the change in effective stress

$\{\Delta p\}$  = the change in pore water pressure

Two extreme cases can be considered. First, under drained conditions, in the absence of pore water pressure effective stresses and total stresses are the same. For this case the theory presented in section 3.3 will suffice. Second, in an undrained case in which there is no overall volume change allowed, new constitutive matrix should be considered by coupling the stiffness of pore water and the stiffness of soil skeleton.

### 3.5.2 Background

In 1941 Biot published his derivation of consolidation theory with an analytical solution. He coupled the solution of equations for pore fluid diffusion with the equations of deformation for the porous solids. In this way, the calculation of time-dependant displacements, strains, pore fluid pressure and effective stresses, became feasible. Biot made the following assumptions in his formulation

- i) The material is isotropic and made up of solid soil particles and a fluid.
- ii) The pore water is incompressible.
- iii) The soil is fully saturated.
- iv) The material is linear elastic
- v) Small strain theory applies.
- vi) Darcy's law governs the flow of pore water through the soil.

The first assumption is not realistic; however, it is a good approximation for engineering purposes. The second assumption is reasonable since the pore fluid in soils is typically water which is essentially incompressible. By recent developments in numerical techniques and computational tools, such as the finite element method, many researchers expanded Biot's theory in order to obtain more realistic answers for complex geotechnical problems (Ghaboussi and Wilson 1973; Sandhu and Wilson

1969; Simon et al. 1986a; Simon et al. 1986b; Zienkiewicz 1985; Zienkiewicz et al. 1980). Lewis and Schrefler (1987) developed a general formulation of multiphase flow in a porous medium based on Biot's formulation, which included the cases of single phase (consolidation), multiphase, and saturated-unsaturated flow. They discussed finite element procedures for both the space and time discretisation aspects of consolidation problems. They also presented linear elastic and nonlinear constitutive relationships.

### 3.5.3 Finite element formulation

The general equations developed from discretisation in space of the equilibrium and continuity equations have been documented by many researchers (Lewis and Schrefler 1987; Zienkiewicz 1985). Lewis and Schrefler developed the following equations

$$[K] \frac{d\bar{u}}{dt} + [L] \frac{d\bar{p}}{dt} = \frac{dF}{dt} \quad (3.36)$$

and

$$[L]^T \frac{d\bar{u}}{dt} + [S] \frac{d\bar{p}}{dt} + [H] \bar{p} = Q \quad (3.37)$$

where

[K] is the tangent stiffness matrix of the solid phase,

$$[K] = \int_{\Omega} [B]^T [D] [B] d\Omega \quad (3.38)$$

[L] is the coupling matrix between the solid and fluid phases,

$$[L] = \int_{\Omega} [B]^T m [N] d\Omega - \int_{\Omega} [B]^T [D] \frac{m}{3K_s} [N] d\Omega \quad (3.39)$$

[H] is the permeability matrix of the porous skeleton,

$$[H] = \int_{\Omega} [\nabla N]^T \frac{k}{\mu} [\nabla N] d\Omega \quad (3.40)$$

[S] is the compressibility matrix,

$$[S] = \int_{\Omega} [N]^T s [N] d\Omega \quad (3.41)$$

in which the scalar coefficient  $s$  is considered as

$$s = \frac{1-\phi}{K_s} + \frac{\phi}{K_w} - \frac{1}{(3K_s)^2} m^T [D] m \quad (3.42)$$

$F$  is the external force vector,

$$[F] = \int_{\Omega} [N]^T db d\Omega + \int_{\Gamma} [N]^T d\hat{t} d\Gamma \quad (3.43)$$

$Q$  is the vector of boundary flows,

$$Q = \int_{\Gamma} [N]^T q d\Gamma + \int_{\Omega} [\nabla N]^T \frac{k}{\mu} \nabla \rho g h d\Omega \quad (3.44)$$

In the above equations  $\bar{u}$  and  $\bar{p}$  are the approximate solutions,  $[N]$  is the shape function,  $[\nabla N]$  is the shape function derivative, and  $\Omega$  is the element domain and  $\Gamma$  is the element boundary. Also in the equations above  $[D]$  is the elastic-plastic constitutive matrix,  $m$  is the vector equivalent of the Kronecker delta (i.e., the identity vector which is equal to unity for the normal stress components and zero for the shear stress components),  $b$  is the vector of body forces,  $\hat{t}$  is the vector of surface tractions,  $k$  is the absolute permeability matrix of the material,  $\mu$  is the dynamic viscosity of the pore water pressure,  $K_s$  and  $K_w$  are the bulk moduli of the soil grains and pore water, respectively,  $\phi$  is the porosity of the soil,  $q$  is the vector of applied fluid flux, and  $\rho$ ,  $g$  and  $h$  are water density, gravity, and elevation head respectively.

As it is mentioned in Equation (3.3) the approximate solutions are related to the nodal values of the field variables through,

$$\begin{aligned} \{\bar{u}\} &= [N] \{u\} \\ \{\bar{p}\} &= [N] \{p\} \end{aligned} \quad (3.45)$$

where  $\{u\}$  and  $\{p\}$  are vectors of the nodal values of soil matrix displacement and pore water pressure, respectively, and their time derivatives are the vector of displacement increments and pore water pressure increments, respectively.

Akers (2001) proposed modifications to Equations (3.39) and (3.42), following the issues he argued about the accuracy of these equations. The modified equations are

$$[L] = \int_{\Omega} [B]^T m [N] d\Omega \quad (3.46)$$

and

$$s = \frac{1-\phi}{K_s} + \frac{\phi}{K_w} \quad (3.47)$$

### 3.5.4 Discretisation in time domain

In order to complete the numerical solution, the rate forms of the equations presented for the coupled consolidation situation, must be discretised in the time domain by approximating  $\bar{u}$  and  $\bar{p}$  with a linear variation within each time step  $\Delta t$ . Using the finite difference approach, integration of an arbitrary function  $y$  over a time interval  $\Delta t$  is

$$\int_t^{t+\Delta t} y(t) dt = [(1-\alpha) y_t + \alpha y_{t+\Delta t}] \Delta t = (\alpha \Delta y + y_t) \Delta t \quad (3.48)$$

$$\Delta y = y_{t+\Delta t} - y_t$$

where  $y_t$  is the value of  $y$  at time  $t$ , and  $\alpha$  is the time interpolation parameter ( $0 \leq \alpha \leq 1$ ).  $\alpha = 0$  is for forward (Euler) interpolation,  $\alpha = 0.5$  is for central (Crank-Nicolson) interpolation and  $\alpha = 1$  is for backward interpolation. Lewis and Schrefler (1998) showed that the solution is conditionally stable for  $0 \leq \alpha < 0.5$  and  $\alpha = 1$  and unconditionally stable for  $0.5 \leq \alpha < 1$ .

Applying Equation (3.48) to the primary variables in Equations (3.36) and (3.37), the following form of the coupled governing equations is obtained

$$[K] \{\bar{u}\}_{t+\Delta t} + [L] \{\bar{p}\}_{t+\Delta t} = [K] \{\bar{u}\}_t + [L] \{\bar{p}\}_t + F \quad (3.49)$$

and

$$\begin{aligned} [L]^T \{\bar{u}\}_{t+\Delta t} + [S] \{\bar{p}\}_{t+\Delta t} + \alpha \Delta t [H] [\bar{p}]_{t+\Delta t} &= [L]^T \{\bar{u}\}_t + [S] \{\bar{p}\}_t + \\ \alpha \Delta t [H] [\bar{p}]_t - \Delta t [H] [\bar{p}]_t + Q \Delta t & \end{aligned} \quad (3.50)$$

In compact matrix-vector form these equations can be rewritten as

$$\begin{bmatrix} [K] & [L] \\ [L]^T & [S] + \alpha \Delta t [H] \end{bmatrix} \begin{Bmatrix} \Delta \bar{u} \\ \Delta \bar{p} \end{Bmatrix} = \begin{Bmatrix} F \\ (Q - [H] \{ \bar{p}_o \}) \Delta t \end{Bmatrix} \quad (3.51)$$

This is the system of equations that must be solved to calculate displacements and pore water pressure in a saturated deforming porous media.

# CHAPTER 4

## DATA MINING TECHNIQUES

### 4.1 Introduction

In recent years, by pervasive developments in computational software and hardware, several computer aided pattern recognition and data-driven approaches have been emerged and developed. The main idea is that a pattern recognition system (e.g., neural network or genetic programming) learns adaptively from experience and extracts various discriminants, each appropriate for its purpose. Although there are other general purpose data-driven techniques, artificial neural networks (ANNs) and genetic programming (GP) are the most widely used pattern recognition methods that have been utilised to model complex engineering problems and capture nonlinear interactions between various parameters in a system. In this chapter a general portrait of the commonly used data-driven techniques is presented in order to provide a better understanding of the features of each technique.

An artificial intelligence based model construction is usually divided into three stages, these are: (i) identification, (ii) parameter estimation and (iii) validation.

A physical system with an output  $y$ , dependent on a set of inputs  $\mathbf{X}$  and parameters  $\mathbf{\theta}$ , can be mathematically formulated as

$$y = F(\mathbf{X}, \boldsymbol{\theta}) \quad (4.1)$$

where  $F$  is a function in an  $m$ -dimensional space where  $m$  is the number of inputs. Data-driven techniques, like GP and ANN, tend to reconstruct  $F$  from input-output data. GP generates a population of expressions for  $F$ , coded in tree structures of variable size, and performs a global search of the best fit expression for  $F$ . ANN goal, on the other hand, is to map  $F$  rather than to find a feasible structure for it.

## 4.2 Artificial neural network

### 4.2.1 General theory

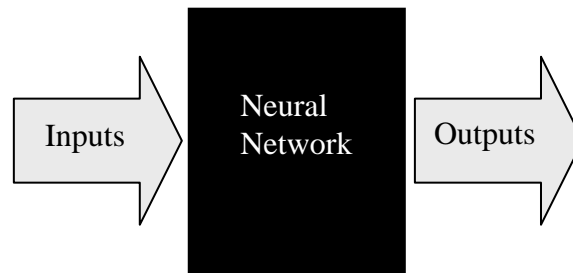
Artificial neural networks (ANNs) are computational models broadly inspired by the organisation of the human brain. The most important features of a neural network are its abilities to learn and to be error tolerant. In other words an artificial neural network is able to acquire, represent, and compute a mapping from a multivariate space of information to another given a set of data representing that mapping (Garrett 1994). Artificial neural network models are adaptive and capable of generalisation. They can handle imperfect or incomplete data, and can capture nonlinear and complex interactions among variables of a system. Because of these strengths, the artificial neural network is emerging as a powerful tool for modelling.

### 4.2.2 The architecture of artificial neural network

#### 4.2.2.1 ANN function

At the most abstract description, a neural network can be considered as a black box, where data is fed in on one side, and processed by the neural network which then produces an output according to the supplied input (Caudill 1991). Although a neural network can usually process any kind of data, (e.g., qualitative or quantitative information) the data fed into the neural network should be pre-processed (e.g., filtered, transformed) to enable faster training and better performance. In fact, the selection, pre-

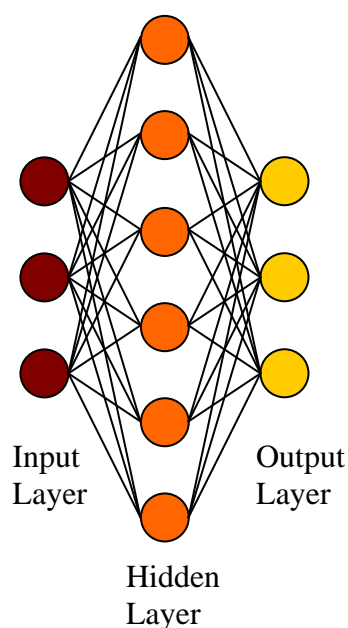
processing, and coding of information is one of the main issues to deal with when working with neural networks. Figure 4.1 shows the function of the neural network.



**Figure 4.1** Neural network function.

#### 4.2.2.2 Layers

A closer look at the black box reveals that its interface to the outside domain consists of an input layer and an output layer of neurons. The neurons are the processing units within the neural network and are usually arranged in layers. The information is propagated through the neural network layer by layer. The layers of a multilayer network play different roles. The layer that produces the network output is called an output layer. The layer that gets the inputs is called input layer. All other layers are called hidden layers. The number of hidden layers can be varied based on the application. Figure 4.2 illustrates the simplified architecture of a neural network with one hidden layer.



**Figure 4.2** General architecture of a neural network with one hidden layer.

### 4.2.2.3 Neurons, connections, weights and biases

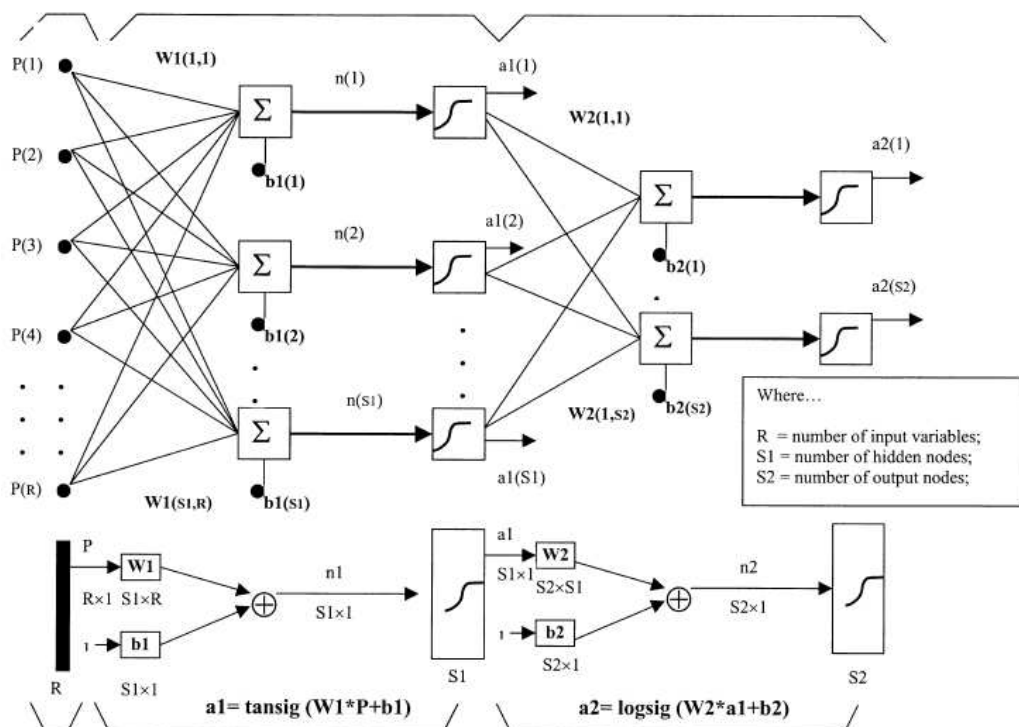
As mentioned each layer is composed of several processing units, i.e., neurons, and these processing units are fully connected to processing units of the succeeding layer. Connections are the paths between neurons where all the information flows within a neural network. A neuron collects information from all preceding neurons relative to the flow of the information and propagates its output to the neurons in the following layer. The output of each preceding neuron,  $n(i)$ , is modulated by a correspondent weight,  $W_i$  ( $S1,R$ ), and bias,  $b_i(S1)$ . This process is realised by the following formula

$$n(i) = W_i(S1,R) \times a_{i-1}(S1) + b_i(S1) \quad (4.2)$$

where  $n(i)$  represents the activity of the neuron. This activity is then modified by transfer function,  $f$ , and becomes the final output of the neuron (Dayhoff 1990).

$$a_i(S1) = f(n(i)) = f(W_i(S1,R) \times a_{i-1}(S1) + b_i(S1)) \quad (4.3)$$

This signal is then propagated to the neurons of the next layer. Figure 4.3 depicts this process. It is common for the number of inputs to a layer to be different from the number of neurons. The network shown in Figure 4.3 has  $R$  inputs ( $R$  neurons in the input layer),  $S1$  neurons in the hidden layer, and  $S2$  neurons in the output layer.

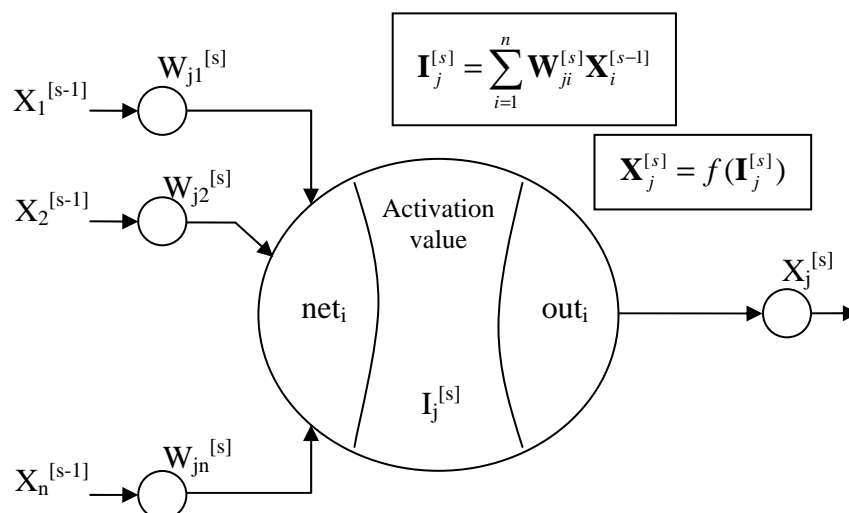


**Figure 4.3** A detailed schematic view of different layers and processing units in a neural network (Rahman et al. 2001).

The most frequently used and efficient learning procedure for multi-layer neural networks is the back-propagation learning algorithm based on the generalised delta rule by Rumelhart et al. (1994). In the next section a brief description of back-propagation learning algorithm is provided.

### 4.2.3 Learning rule

The learning rule refers to the mechanism that is used to adjust the weights and biases of the networks to achieve some desired network behaviour. Several learning rules have been developed by different authors. The back-propagation learning rule is the most popular one that has been used to train nonlinear, multilayered networks to perform function approximation and pattern classification (Rahman et al. 2001). The back-propagation learning rule can be used to adjust the weights and biases of networks in order to minimise the sum-squared error of the network. This is done by continually changing the values of the network weights and biases in the direction of steepest descent with respect to error. Derivatives of the error vector are calculated for the network's output layer and then back-propagated through the network until derivatives of error are available for each hidden layer. The basic architecture of a back-propagation processing unit is shown in Figure 4.4.



**Figure 4.4** The back-propagation processing unit.

A back-propagation processing unit transfers its inputs as

$$\mathbf{X}_j^{[s]} = f(\mathbf{I}_j^{[s]}) = f\left(\sum_{i=1}^n \mathbf{W}_{ji}^{[s]} \mathbf{X}_i^{[s-1]}\right) \quad (4.4)$$

where the transfer function  $f$  can be any differentiable function. Usually,  $f$  is taken as a sigmoid or a hyperbolic tangent function

$$f(Z) = (1.0 + e^{-\lambda Z})^{-1} \quad (\text{sigmoid}) \quad (4.5)$$

$$f(Z) = \frac{(1.0 - e^{-\lambda Z})}{(1.0 + e^{-\lambda Z})} \quad (\text{hyperbolic tangent}) \quad (4.6)$$

where  $\lambda$  is called a ‘gain coefficient’ used to control the slope of the transfer function.

#### 4.2.4 Training

Training refers to the process that repeatedly applies input vectors to the network and calculates errors with respect to the target vectors and then finds new weights and biases with the learning rule. It repeats this cycle until the sum-squared error falls beneath an error goal, or a maximum number of epochs is reached. Training a feed-forward network with the back-propagation learning rule is most frequently used in function approximation and pattern recognition. The training parameters specify the maximum number of epochs to train, the sum-squared error goal, and the learning rate. The sizes of changes that are made in the weights and biases at each epoch are specified with the learning rate. Small learning rates result in long training times but guarantee that the network's values do not jump over valleys in the error surface that lead to lower errors. Normally the simple back-propagation is very slow because it requires small learning rates for stable learning. Some techniques such as momentum and adaptive learning rate; and an alternative method to gradient descent, Levenberg-Marquardt optimisation (Demuth and Beale 1996) are used to improve the speed and general performance of back-propagation. Momentum decreases back-propagation sensitivity to small details in the error surface. This helps the network avoid getting stuck in shallow minima, which would prevent the network from finding a lower error solution. An adaptive learning rate can decrease the training time by trying to keep the learning step size as large as

possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface. Levenberg-Marquardt optimisation is a more sophisticated method than the gradient descent method. It is much faster than gradient descent, but requires more memory. It will calculate the derivatives for each error on a network's output with respect to each weight of a layer and returns the Jacobian matrix of derivatives of each network error with respect to each weight in the layer. The rule for updating can be expressed as

$$\Delta \mathbf{W} = (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} \mathbf{J}^T \mathbf{e} \quad (4.7)$$

where  $\mathbf{J}$  is the Jacobian matrix of derivatives of each error to each weight,  $\mu$  is a scalar,  $\mathbf{I}$  is the unity matrix, and  $\mathbf{e}$  is an error vector. If the scalar  $\mu$  is very large, the above expression approximates gradient descent, while if it is small the above expression becomes the Gauss-Newton method. The Gauss-Newton method is faster and more accurate near an error minimum, so the aim is to shift towards the Gauss-Newton method as quickly as possible. Thus,  $\mu$  is decreased after each successful step and increased only when a step increases the error.

Usually eight training parameters need to be specified before the actual training. They will dictate the maximum number of epochs, the error goal, the minimum error gradient, the initial value for the scalar  $\mu$ , the multipliers for increasing and decreasing  $\mu$ , and the maximum value for  $\mu$ . Training continues until the error goal is met, the minimum error gradient occurs, the maximum value of  $\mu$  occurs, or the maximum number of epochs is reached.

Explanation of more details relating to artificial neural networks is out of the scope of this thesis. Texts describing aspects and features of ANN models and architectures in greater detail can be found in the references (e.g., Lippmann 1987; Flood and Kartam 1994).

#### **4.2.5 The application of ANN in geomechanics**

So far ANNs have been applied to a wide range of geotechnical engineering problems such as pile bearing capacity (e.g., Abu-Kiefa 1998; Goh 1996), site characterisation (e.g., Juang et al. 2001), soil behaviour (e.g., Zhu et al. 1998), liquefaction potential

(e.g., Juang and Chen 1999), slope stability (e.g., Lu and Rosenbaum 2003), underground openings (e.g., Benardos and Kaliampakos 2004; Javadi 2006) and many others. Although it has been shown by various researchers that ANNs offer great advantages in the analysis of many geotechnical engineering problems, but in general they suffer from a number of drawbacks. One of the main disadvantages of the ANN is that the optimum structure of the network (such as number of inputs, hidden layers, transfer functions, etc.) must be identified a priori, which is usually done through a time consuming trial and error procedure. Another major disadvantage of neural network based models is the large complexity of the network structure, as it represents the knowledge in terms of a weight matrix and biases which are not accessible to user understanding. In addition, as ANNs perform function approximation through large parameterisation and the use of simple functional structures (transfer functions); parameter estimation and overfitting problems represent other major disadvantages of a model constructed by ANN (Giustolisi 2002).

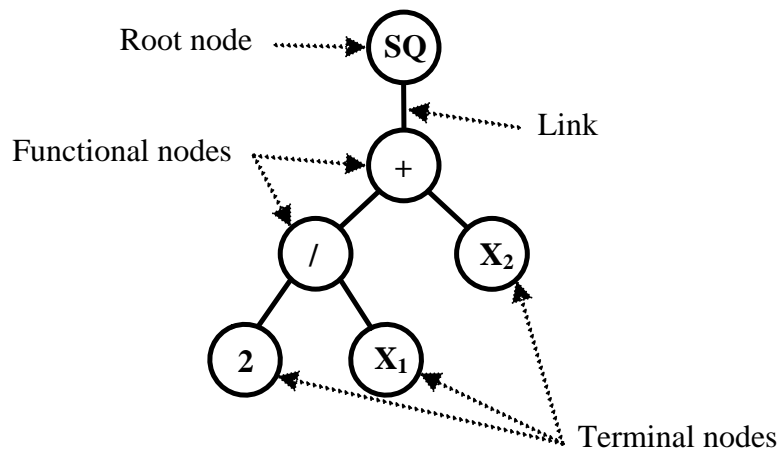
## **4.3 Genetic programming**

### **4.3.1 General theory and background**

Genetic programming which was introduced in the early 90s by Koza (1992), is an evolutionary computing method that generates a transparent and structured representation of the data provided. Evolutionary algorithms (EAs) are search techniques based on computer implementations of some of the evolutionary mechanisms found in nature (such as selection, crossover and mutation) in order to solve a function identification problem. The function identification problem is to search for a function in a symbolic form that fits a set of experimental data.

Genetic algorithm (GA) and genetic programming (GP) are the major types of evolutionary algorithms. GP is a generalisation and an extension of GA. GAs are generally used in parameter optimisation to evolve the best values for a given set of model parameters, whereas GPs give the basic structure of the approximation model together with the values of its parameters. While a GA uses a string of numbers to represent the solution, the GP combines a high level symbolic representation with the search efficiency of the GA to form the best possible model for the system.

Representation schemes in genetic programming are composed of nodes which are elements from a terminal set (constants e.g., 2 and/or variables e.g.,  $x_1$ ,  $x_2$ , etc.) and a functional set (mathematical operators that generate the regression model e.g.,  $\pm$  and  $x^y$ , etc.). A typical genetic programming tree, representing the simple algebraic expression  $(2/x_1+x_2)^2$  is shown in Figure 4.5.

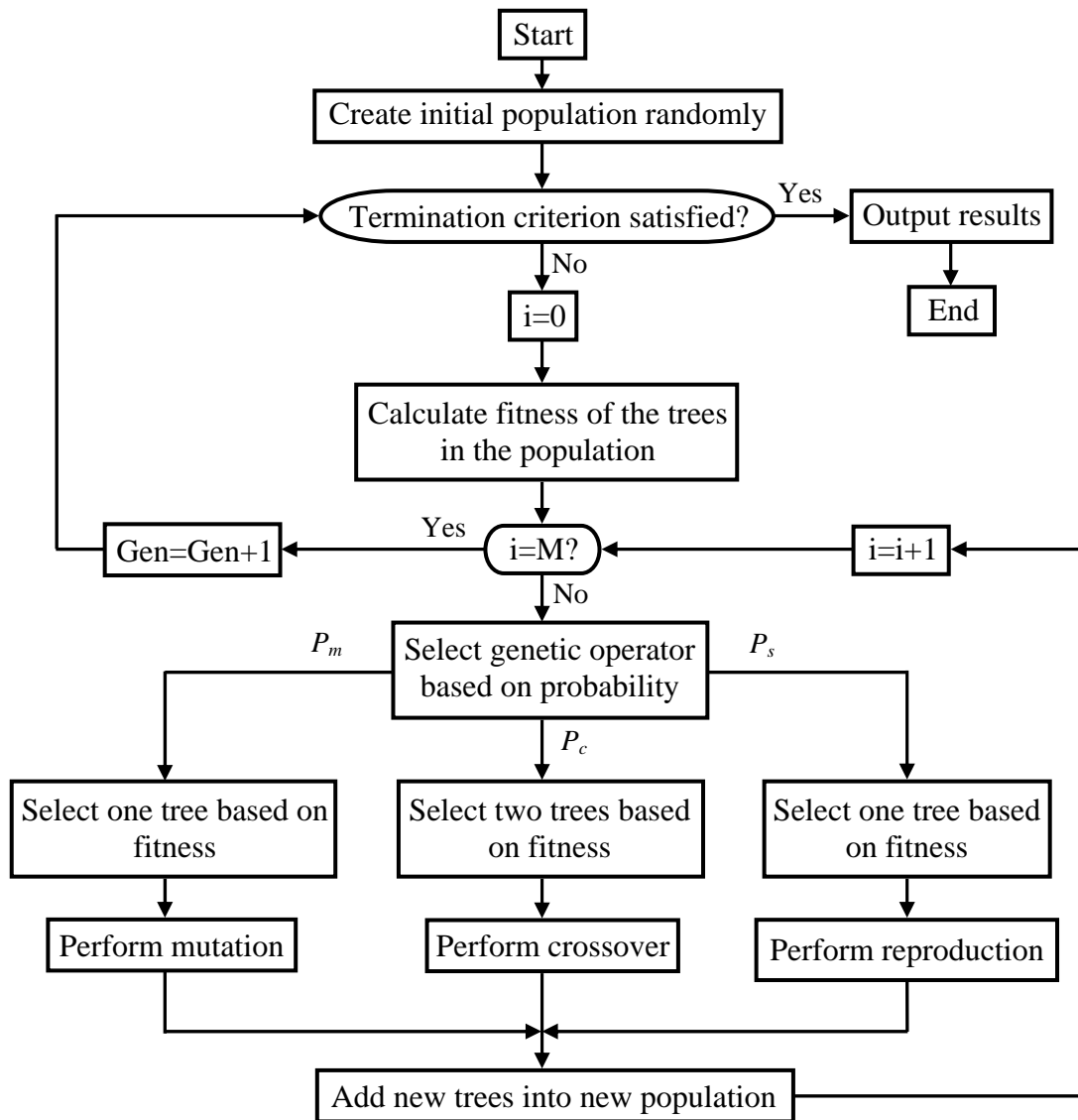


**Figure 4.5** Typical GP tree representing function  $\left(\frac{2}{x_1} + x_2\right)^2$ .

The functional set can be subdivided into binary nodes which take any two arguments (like addition) and unary nodes which take one argument (like a square root). The solution domain is created by the recursive composition of elements from the functional set for any internal node and from the terminal set for any external nodes (leaf nodes). Whenever a node in a tree is created from the functional set, a number of links equal to the number of arguments the function takes is created to radiate out from that node. The result of this process is a set of random trees of different sizes and shapes, each exhibiting a different fitness with respect to the objective function. If the set of applied functions is sufficiently rich, tree structures are capable of representing hierarchical programs of any complexity. These functions may include arithmetic operators (like  $+$ ,  $\times$ ,  $-$ ), mathematical functions (like  $\sin$ ,  $\cos$ ,  $\ln$ ), Boolean operators (like AND, OR), logical expressions (like IF, THEN), iterative functions (like DO, CONTINUE, UNTIL) or other user defined functions (Sette and Boullart 2001).

### 4.3.2 Overview of genetic programming process

The nature of genetic programming (GP) allows the user to gain additional information on how the system performs, i.e. gives an insight into the relationship between input and output data. Once a population of computer programs has been randomly created, the process of evolving the population proceeds using the simple principles as for GAs, with the minor difference that, strings of functions and terminals are reproduced, crossed over and mutated rather than strings of binary codes. Evolutionary algorithms maintain a population of structures that evolve according to the rules of natural selection and some operators inspired from natural genetics such as reproduction or crossover. Each individual in the population receives a measure of its fitness in the current environment. The fitness criteria are calculated by the objective function i.e., how good the individual is at competing with the rest of the population. At each generation a new population is created by the process of selecting individuals according to their fitness and breeding them together using the genetic operators (crossover and mutation). The existing population will then be replaced with the new population. The procedure stops when the termination criterion, which can be either the maximum number of generations or a particular allowable error, is satisfied. After the termination criterion is met, the single best program in the final population is designated as the result of the GP process. A typical flow diagram for a genetic programming procedure is illustrated in Figure 4.6. The basic genetic programming operators (reproduction, crossover and mutation) are described in the following sections.



**Figure 4.6** Typical flow diagram for a genetic programming procedure where  $M$  is the number of generations,  $P_m$  is the probability of mutation;  $P_c$  is the probability of crossover and  $P_s$  is the probability of reproduction (Rezania and Javadi 2007).

### 4.3.3 Initial population

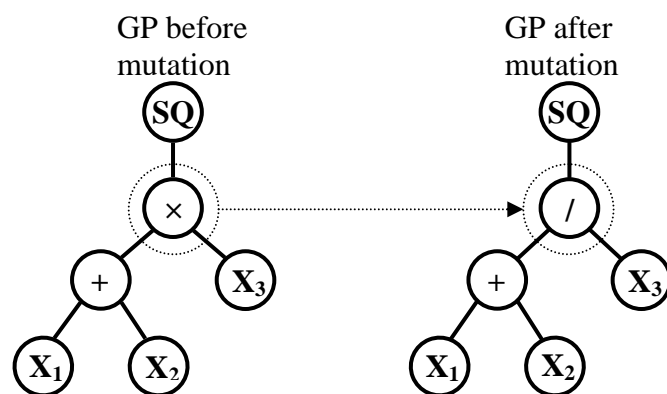
The first step in genetic programming is to create an initial population of  $n$  tree structures (computer programs) by randomly selecting functions and terminals from the user defined sets. The fitness of each structure is then evaluated according to how well it performs the desired objective, and a corresponding fitness value is assigned.

### 4.3.4 Reproduction

The second stage in the process involves selecting a proportion of the initial population to be copied to the next generation. This is done probabilistically according to the fitness of each tree structure. The method of selection used is known as the Roulette Wheel selection which is essentially a random selection with some bias based upon the fitness score. The overall result is that while the choices are made at random, those trees with high fitness scores have a higher probability of being selected. Those with lower scores can also be selected for further use; however, they are less probable. The programs which are successfully selected for reproduction then enter to the new population.

### 4.3.5 Mutation

In genetic programming the mutation procedure involves selecting a casual node within a randomly selected tree and replacing it with another randomly selected node from the same function or terminal set, except by itself. The selected nodes must have the same number of arguments. Also, a functional node replaces a functional node and a terminal node (variable or constant) replaces a terminal one. This procedure, which is shown in Figure 4.7, is called the allele mutation.



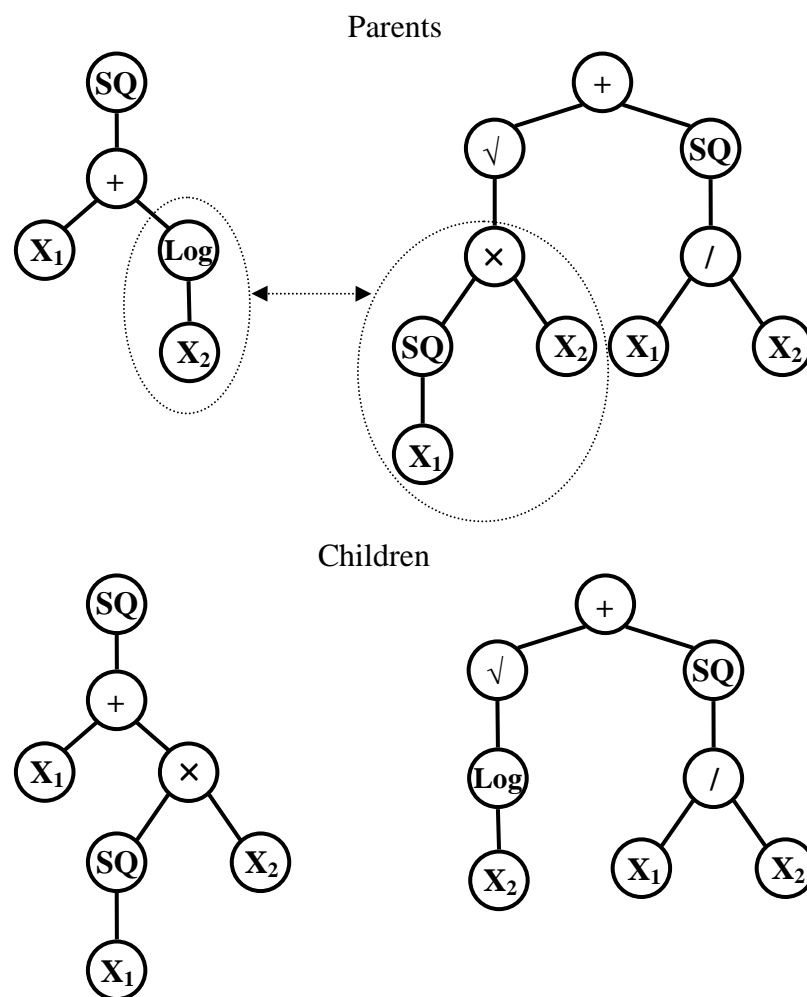
**Figure 4.7** Typical mutation operation in GP.

### 4.3.6 Crossover

To develop new genetic information, and hopefully improve the fitness of the population, the tree structures in the mating pool undergo crossover. Crossover is one of the basic genetic operators that helps to evolve the model structures. In genetic programming, the crossover operator creates new offspring that consist of genetic material taken from the parents and is implemented in the following way

- i) Two trees are selected from the population.
- ii) One node is selected randomly within each of these trees.
- iii) The sub-trees under the selected nodes are swapped, thus two offspring belonging to the new population are obtained.

A typical cross over operation is shown in Figure 4.8.



**Figure 4.8** Typical cross over operation in GP.

### **4.3.7 The application of GP in geomechanics**

Application of genetic programming strategy in the field of civil engineering is quite new and original, and it has just started to be used in the field of geotechnical engineering. Indeed the very pioneering works relating to investigation of the capability of genetic programming in the field of geotechnics have been published recently by the author of this thesis and his coworkers (Javadi and Rezanian 2006; Javadi et al. 2006; Rezanian and Javadi 2007).

Javadi et al. (2006) used genetic programming to find relationships to evaluate lateral displacements due to liquefaction during an earthquake. This is a very complex geotechnical problem because a large number of parameters are involved (i.e., parameters describing the earthquake strength, geology of the site and the soil characteristics). Based on the details provided in this paper, it was shown that the GP based models for lateral spreading determination, offer an improvement over the most commonly used model, multi linear regression model, (Youd et al. 2002) for this problem.

Also Rezanian and Javadi (2007) utilised the genetic programming for prediction of settlement of shallow foundations on cohesionless soils. It was shown that the application of the traditional methods for prediction of settlement of shallow foundations could lead to very large errors. A new GP based model was developed by the authors and the developed model was presented in this paper. Comparison of the results shows that the predictions by the proposed GP model provide significant improvements over the traditional methods and also outperforms the ANN based models.

## **4.4 Evolutionary polynomial regression**

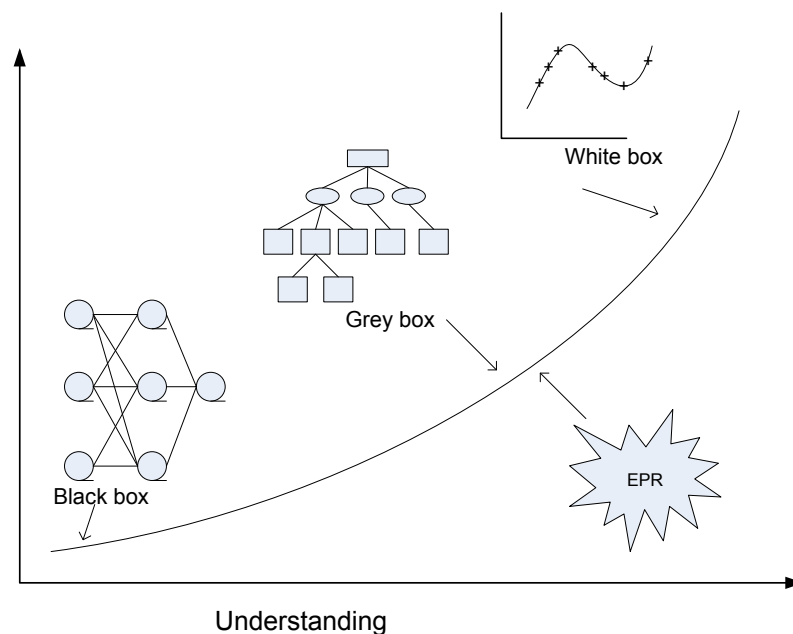
### **4.4.1 Overview**

Evolutionary polynomial regression (EPR) is a data-driven method based on evolutionary computing, aimed to search for polynomial structures representing a

system. Genetic programming and neural network are both very powerful non-linear modelling techniques, but they have their own drawbacks. GP tends to search for mathematical expressions for  $F$  in Equation (4.1) using an evolutionary approach, but the parameter values (vector  $\theta$ ) are generated as non-adjustable constants, referred to as ephemeral random constants. Therefore the constants do not necessarily represent optimal values as in numerical regression methods and good structures of  $F$  can be missed in the process. Furthermore the number of terms in GP based expressions can greatly exceed and the evolutionary search within GP can be quite slow. Some of the disadvantages of ANN approach have been highlighted in the previous section.

#### 4.4.2 EPR model construction

In the context of artificial intelligence based modelling, EPR is classified as a symbolic grey box technique which can construct clearly structured model expressions for a given set of data. A schematic representation of EPR classification in comparison with other modelling methods is shown in Figure 4.9.



**Figure 4.9** EPR classification among modelling techniques.

To avoid the problem of mathematical expressions growing rapidly in length with time associated with GP, in EPR the evolutionary procedure is conducted in a way that it searches for the exponents of a polynomial function with a fixed maximum number of terms, rather than performing a general evolutionary search as used in normal GP.

Furthermore, during one execution it returns a number of expressions with increasing numbers of terms up to a limit set by the user, to allow the optimum number of terms to be selected. The general form of expression used in EPR can be presented as (Giustolisi and Savic 2006)

$$y = \sum_{j=1}^m F(\mathbf{X}, f(\mathbf{X}), a_j) + a_0 \quad (4.8)$$

where  $y$  is the estimated vector of output of the process;  $a_j$  is a constant;  $F$  is a function constructed by the process;  $\mathbf{X}$  is the matrix of input variables;  $f$  is a function defined by the user; and  $m$  is the maximum number of terms of the target expression. The first step in identification of the model structure is to transfer Equation (4.8) into the following vector form

$$\mathbf{Y}_{N \times 1}(\boldsymbol{\theta}, \mathbf{Z}) = \begin{bmatrix} \mathbf{I}_{N \times 1} & \mathbf{Z}_{N \times m}^j \end{bmatrix} \times [a_0 \ a_1 \ \dots \ a_m]^T = \mathbf{Z}_{N \times d} \times \boldsymbol{\theta}_{d \times 1}^T \quad (4.9)$$

where  $\mathbf{Y}_{N \times 1}(\boldsymbol{\theta}, \mathbf{Z})$  is the least squares estimate vector of the  $N$  target values;  $\boldsymbol{\theta}_{d \times 1}$  is the vector of  $d=m+1$  parameters  $a_j$  and  $a_0$  ( $\boldsymbol{\theta}^T$  is the transposed vector); and  $\mathbf{Z}_{N \times d}$  is a matrix formed by  $\mathbf{I}$  (unitary vector) for bias  $a_0$ , and  $m$  vectors of variables  $\mathbf{Z}^j$  that for fixed  $j$  are a product of the independent predictor vectors of inputs,  $\mathbf{X} = \langle \mathbf{X}_1 \ \mathbf{X}_2 \ \dots \ \mathbf{X}_k \rangle$ .

In general, EPR is a two-stage technique for constructing symbolic models; (i) initially, using standard genetic algorithm (GA), it searches for the best form of the function structure, i.e., a combination of vectors of independent inputs,  $\mathbf{X}_{s=1:k}$ , and (ii) secondly it performs a least squares regression to find the adjustable parameters,  $\boldsymbol{\theta}$ , for each combination of inputs. In this way a global search algorithm is implemented for both the best set of input combinations and related exponents simultaneously, according to the user-defined cost function. The matrix of inputs,  $\mathbf{X}$ , is considered as:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1k} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2k} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3k} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{Nk} \end{bmatrix} = [\mathbf{X}_1 \ \mathbf{X}_2 \ \mathbf{X}_3 \ \dots \ \mathbf{X}_K] \quad (4.10)$$

where  $k^{\text{th}}$  column of  $\mathbf{X}$  represents the candidate variable for the  $j^{\text{th}}$  term of Equation (4.9). Therefore the  $j^{\text{th}}$  term of  $\mathbf{Z}$  in Equation (4.9) can be written as

$$\mathbf{Z}_{N \times 1}^j = \left[ (\mathbf{X}_1)^{\mathbf{ES}(j,1)} \cdot (\mathbf{X}_2)^{\mathbf{ES}(j,2)} \cdot (\mathbf{X}_3)^{\mathbf{ES}(j,3)} \cdot \dots \cdot (\mathbf{X}_k)^{\mathbf{ES}(j,k)} \right] \quad \forall j = 1 \dots m \quad (4.11)$$

where,  $\mathbf{Z}^j$  is the  $j^{\text{th}}$  column vector whose elements are products of candidate independent inputs and ES is a matrix of exponents. The aim is to find the matrix  $\mathbf{ES}_{m \times k}$  of exponents whose elements can assume values within user defined bounds. For example, if a vector of candidate exponents for input parameters,  $\mathbf{X}$ , is chosen to be  $\mathbf{EX} = [-2, -1, 0, 1, 2]$ , the number of terms ( $m$ ) is assumed to be 4, and the number of independent inputs ( $k$ ) is 3, the polynomial regression problem is to search for a matrix of exponent  $\mathbf{ES}_{4 \times 3}$ . An example of such a matrix can be like the following

$$\mathbf{ES}_{4 \times 3} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 2 & -1 \\ 1 & 0 & 0 \\ -2 & 1 & 0 \end{bmatrix} \quad (4.12)$$

Each exponent in ES corresponds to a value from the user defined vector EX. Also each row of ES determines the exponents of the candidate variables of  $j^{\text{th}}$  term in Equations (4.8) and (4.9). By implementing the above values in Equation (4.11), the following set of expressions is obtained

$$\begin{aligned} \mathbf{Z}_1 &= (\mathbf{X}_1)^{-1} \cdot (\mathbf{X}_2)^0 \cdot (\mathbf{X}_3)^1 = \mathbf{X}_1^{-1} \cdot \mathbf{X}_3 \\ \mathbf{Z}_2 &= (\mathbf{X}_1)^0 \cdot (\mathbf{X}_2)^2 \cdot (\mathbf{X}_3)^{-1} = \mathbf{X}_2^2 \cdot \mathbf{X}_3^{-1} \\ \mathbf{Z}_3 &= (\mathbf{X}_1)^1 \cdot (\mathbf{X}_2)^0 \cdot (\mathbf{X}_3)^0 = \mathbf{X}_1 \\ \mathbf{Z}_4 &= (\mathbf{X}_1)^{-2} \cdot (\mathbf{X}_2)^1 \cdot (\mathbf{X}_3)^0 = \mathbf{X}_1^{-2} \cdot \mathbf{X}_2 \end{aligned} \quad (4.13)$$

Therefore based on the matrix given in Equation (4.12), the expression of Equation (4.9) is given as

$$\mathbf{Y} = a_0 + a_1 \mathbf{Z}_1 + a_2 \mathbf{Z}_2 + a_3 \mathbf{Z}_3 + a_4 \mathbf{Z}_4 = a_0 + a_1 \frac{\mathbf{X}_3}{\mathbf{X}_1} + a_2 \frac{\mathbf{X}_2^2}{\mathbf{X}_3} + a_3 \mathbf{X}_1 + a_4 \frac{\mathbf{X}_2}{\mathbf{X}_1^2} \quad (4.14)$$

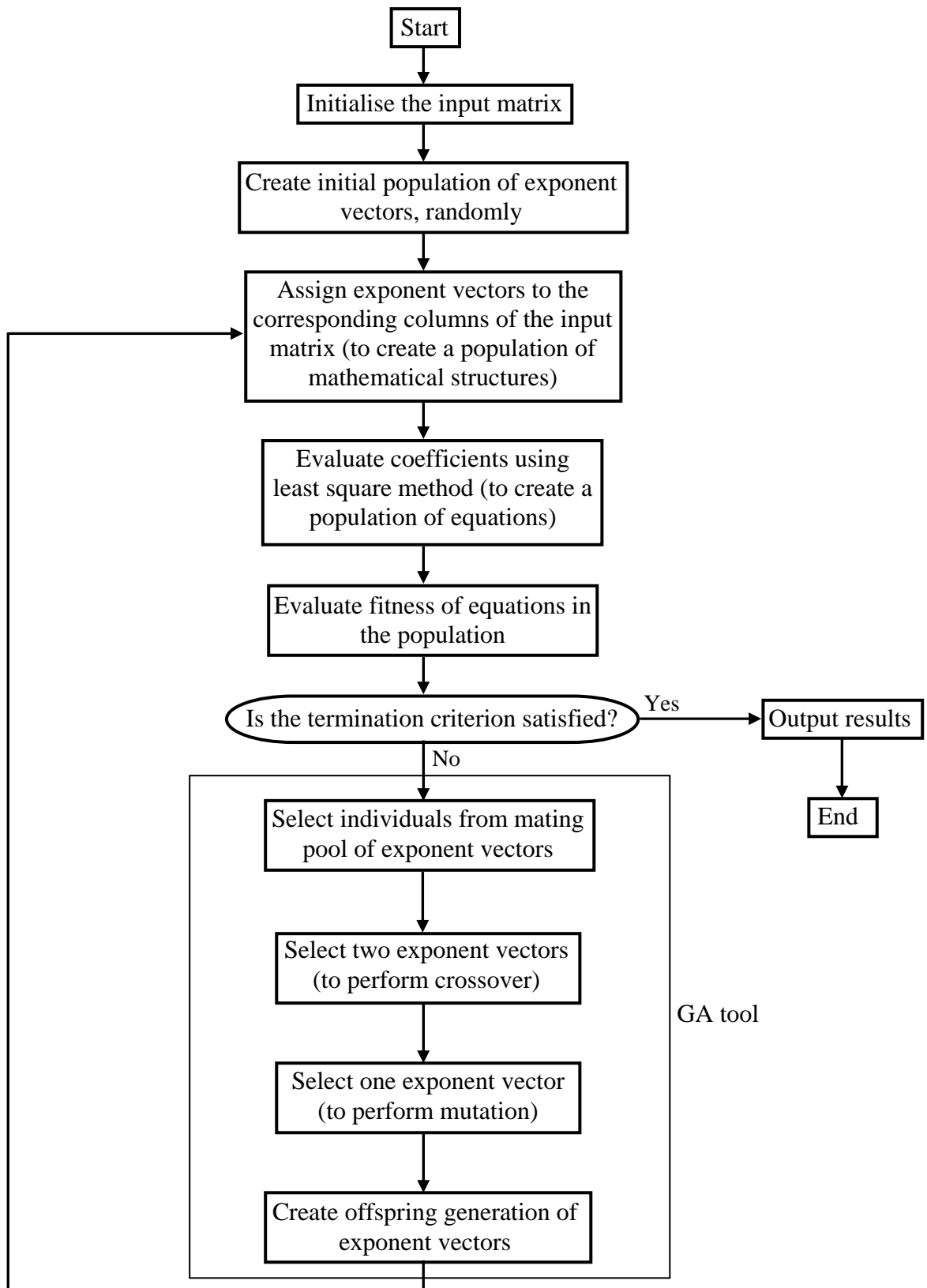
The adjustable parameters,  $a_j$ , can now be evaluated, by means of the linear least squares (LS) method based on minimisation of the sum of squared errors (SSE) as the cost function. Note that each row of ES determines the exponents of the candidate variables of  $j^{\text{th}}$  term in Equations (4.8) and (4.9). Each of the exponents in ES corresponds to a value from the user-defined vector EX. This allows the transformation

of the symbolic regression problem into one of finding the best ES, i.e., the best structure of the EPR equation, e.g., in Equation (4.14).

The global search for the best form of Equation (4.14) is performed by means of a standard GA over the values in the user defined vector of exponents (i.e., EX). The GA operates based on Darwinian evolution which begins with random creation of an initial population of solutions. Each parameter set in the population represents the individual's chromosomes. Each individual is assigned a fitness based on how well it performs in its environment. Through crossover and mutation operations, with the probabilities  $P_c$  and  $P_m$  respectively, the next generation is created. Fit individuals are selected for mating, whereas weak individuals die off. The mated parents create a child (offspring) with a chromosome set which is a mix of parents' chromosomes. For example if parent one has chromosome ABCDE and parent two has chromosome FGHIJ one possible chromosome for the child is ABHIJ where the position between B and H is the crossover point. It is also possible that one parent chromosome undergoes mutation operation to form the offspring (e.g., ABCDO from parent one). Standard GA uses binary strings of 0's and 1's to form the chromosomes. Instead, in EPR integer GA coding is used to determine the location of the candidate exponents of EX in the matrix ES. For example the positions in EX= [-2, -1, 0, 1, 2] correspond to the following string for the matrix of Equation (4.12) and the expression of Equation (4.14)

$$[2\ 3\ 4, 3\ 5\ 2, 4\ 3\ 3, 1\ 4\ 3] \quad (4.15)$$

Additionally, it is clear that the presence of one zero in EX assures the ability to exclude some of inputs and/or input combinations from the regression equation. The EPR process stops when the termination criterion, which can be either the maximum number of generations, the maximum number of terms in the target mathematical expression or a particular allowable error, is satisfied. A typical flow diagram for the EPR procedure is illustrated in Figure 4.10.



**Figure 4.10** Typical flow diagram for the EPR procedure (Rezania et al. 2008).

### 4.4.3 Least squares solution by singular value decomposition

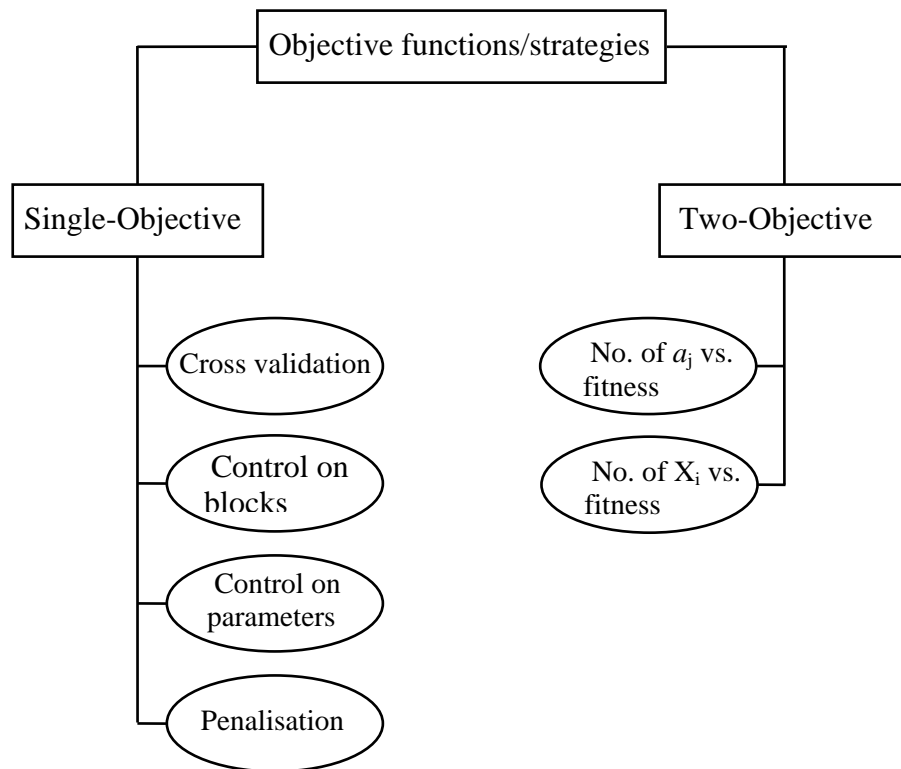
Computing  $a_j$  in Equation (4.14) is an inverse problem that corresponds to solving an over-determined linear system as a LS problem. This problem is traditionally solved by Gaussian elimination. However, an evolutionary search procedure may generate candidate solutions (e.g., a combination of exponents of X) that correspond to an ill-conditioned inverse problem. This often means that the rectangular matrix  $Z_{N \times d}$

$$\mathbf{Z} = \left[ \mathbf{I}_{N \times 1} \quad \mathbf{Z}_{N \times 1}^1 \quad \mathbf{Z}_{N \times 1}^2 \quad \mathbf{Z}_{N \times 1}^3 \quad \dots \quad \mathbf{Z}_{N \times 1}^m \right]_{N \times (m+1) = N \times d} \quad (4.16)$$

may not be of full rank (if a solution contains a column of zeros) or the columns  $Z^j$  are linearly dependent. This could pose serious problems to Gaussian elimination and a more robust solver is therefore needed. Parameter estimation of  $a_j$  (or  $\theta$ ) in EPR is performed by means of singular value decomposition (SVD) of the matrix Z (Doglioni 2004). This approach makes the process of finding the solution to the LS problem more robust, although in general the SVD is slower than Gaussian elimination (Golub and Van Loan 1993).

### 4.4.4 Model selection

EPR is able to work with different objective functions to optimise in order to get the best symbolic model or set of symbolic models. EPR can work both in single-objective configuration and in multi-objective configuration. Figure 4.11 presents an overview of main available objective functions/strategies in EPR. EPR introduces a kind of multidimensional strategy for model selection, based on a comprehensive analysis of complexity and fitness of models.



**Figure 4.11** Overview on main objective functions/strategies in EPR.

It is widely accepted that the best modelling approach is also the simplest that fits the purpose of the application. The so-called principle of parsimony states that for a set of otherwise equivalent models of a given phenomenon one should choose the simplest one to explain a set of data.

Therefore, the fitness in regression-based models should also include a measure of trade-off between the model complexity (i.e. addition of new parameters) and the quality of fit. This kind of trade-off can be basically achieved in two ways

- i) Single-objective: an objective function is used so that it could control the fitness of the models without allowing that unnecessary complexities are introduced in the models.
- ii) Multi-objectives: at least two objective functions are introduced; in this case one objective function will control the fitness of the models, while at least one objective function controls the complexity of the models. This approach implies the advantage of returning a set of non-dominated models each one presenting fitting and complexity features which vary along the Pareto front representative of the model solutions. Therefore, the user is not required to

assume the number of building blocks *a-priori*, but he/she just sets the maximum number of building blocks, while the control on the complexity will let them vary according to the fitness. Then, the Pareto front represents the trade-off surface, or line, of complexity vs. fitness, which is required. The trade-off surface allows the user to achieve a lot of purposes of the modelling approach to the phenomenon studied (Doglioni 2004).

#### 4.4.4.1 Single objective approach

For a given set of observations or data, a regression-based technique needs to search among a large if not an infinite number of possible models to explain those data. By varying the exponents for the columns of matrix X, and by searching for the best-fit set of parameters  $\theta$ ; the EPR methodology searches among all those models. It does, however, require an objective function that will ensure the best fit, without the introduction of unnecessary complexity. Unnecessary complexity is here defined as the addition of new terms or combinations of inputs that fit some noise in the raw data rather than the underlying phenomenon. Therefore, the key objective here is to find a systematic means to avoid the problem of over-fitting. There are three possible approaches to this problem

- i) To penalise the complexity of the expression by minimising the number of terms;
- ii) To control the variance of  $a_j$  constants (the variance of estimates) with respect to the their values; and
- iii) To control the variance of  $a_j \cdot Z_j$  terms with respect to the variance of residuals.

A detailed description of the three possible approaches in single-objective configuration is now provided

*Complexity Penalisation*

In order to choose a model of optimal complexity corresponding to smallest prediction (generalisation) error for future data, one needs to be able to compare two models with different levels of complexity and model fit. The sum of squared errors (SSE) is normally used to guide the search toward the best-fit model

$$\mathbf{SSE} = \frac{\sum_{i=1}^N (y_a - y_p)^2}{N} \quad (4.17)$$

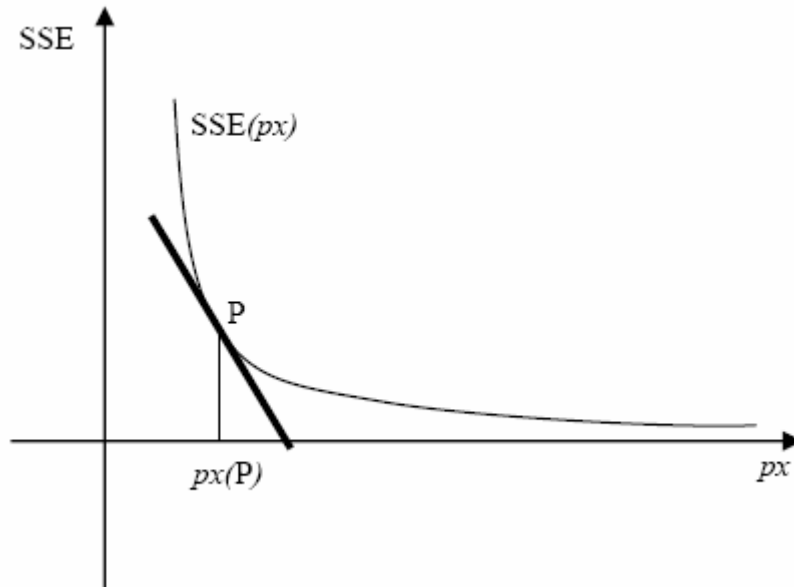
where  $y_a$  are the target values in the training dataset and  $y_p$  are the model predictions computed by using the polynomial expression obtained by EPR. In order to allow the trade-off between the quality of fit (SSE) and the model complexity (number of input combinations), the following penalisation of complexity (PCS) fitness function was proposed by Doglioni (2004)

$$PCS = \frac{\mathbf{SSE}}{(Nd - px + 1)^\alpha} \quad (4.18)$$

where  $Nd = k \times m$  is the maximum number of inputs that can be considered,  $px$  is the actual number of inputs selected by the GA and  $\alpha$  is an adjustable exponent, which controls the degree of pressure to control complexity. This form of the fitness function will be better understood if the derivative of the fitness function with respect to  $px$  is derived

$$\frac{\partial}{\partial px} \left( \frac{\mathbf{SSE}}{(Nd - px + 1)^\alpha} \right) = \left( \frac{1}{(Nd - px + 1)^\alpha} \right) \left( \frac{\partial \mathbf{SSE}}{\partial px} + \frac{\alpha \cdot \mathbf{SSE}}{Nd - px + 1} \right) \quad (4.19)$$

The fitness decreases with respect to  $px$  if the derivative in Equation (4.19) is negative (see Figure 4.12).



**Figure 4.12** SSE variations vs.  $px$ .

Therefore the following inequality should hold

$$\frac{\alpha \cdot \text{SSE}}{Nd - px + 1} \leq -\frac{\partial \text{SSE}}{\partial px} = -\text{VAR}_{px}(\text{SSE}) \quad (4.20)$$

In other words, the addition of another combination of inputs  $X$ , needs not only to be justified on the basis of decreasing SSE, but also needs to take into account the terms  $(Nd-px)$  and  $a \cdot \text{SSE}$ . The concept is shown in Figure 4.12. The bold line is the derivative of SSE with respect to  $px(P)$ , while the curve is the natural SSE variation due to the increase in the number of input parameters. Equation (4.20) requires a value of the SSE derivative at  $P$  greater than or equal to the term on the left side of the inequality. Equation (4.20) illustrates that when the actual number of inputs  $px$  approaches to maximum  $Nd$  the left term of the inequality increases and, consequently, a high absolute-variation of SSE is required ( $\text{VAR}(\text{SSE})$  is always negative). This results in penalisation of complex structures by way of controlling the total number of inputs in the formula.

#### *Variance of $a_j$*

EPR may control the polynomial term contribution to variance of  $Y$  expressed through their parameters during GA search. It might be argued that low constant value with respect to variance of estimates corresponds to terms that begin to describe noise in

preference to the underlying function of phenomena. Therefore the variance of estimated constant values is used to eliminate those parameters whose value is not sufficiently larger than the variance of the estimated value. Hence the variance in estimation of parameters obtained by EPR can be computed as

$$\text{var}(\theta) = [\text{var}(a_0) \quad \text{var}(a_1) \quad \text{var}(a_2) \quad \dots \quad \text{var}(a_m)] \quad (4.21)$$

Assuming that constants are from a Gaussian probability density function, the following expression is used

$$\gamma \cdot \sqrt{|\text{var}(a_j)|} = \gamma \cdot |\text{StD}(a_j)| > |a_j| \Rightarrow a_j = 0 \quad (4.22)$$

where StD is the standard deviation of estimated constants,  $a_j$  (from the diagonal elements of the covariance matrix) and  $\gamma=2.578$  is the value from the table of the standard normal distribution related to the confidence interval of 99%. Equation (4.22) states that if, for example, the modulus of the estimated constant value is lower than  $2.578|\text{StD}(a_j)|$ , which corresponds to a confidence level of 99%, the constant value is assumed to be equal to zero.

#### *Variance of $a_j \cdot Z_j$*

EPR may control the polynomial term contribution to variance of Y through evaluating  $a_j \cdot Z_j$  with respect to variance of the noise in the raw data during GA search. Indeed, a level of noise may exist under which the variance of the terms  $a_j \cdot Z_j$  will describe noise, causing over-fitting related problems. This level of noise is not known *a priori* and, therefore, the residual vector E ( $E = (Y - Y(\theta, Z))$ ) could be used to estimate noise. In this manner, the standard deviation of E can be compared with the standard deviation of terms  $a_j \cdot Z_j$ , as

$$|\text{StD}(a_j \cdot Z_j)| < \beta \cdot |\text{StD}(E)| \Rightarrow a_j = 0 \quad (4.23)$$

where  $\beta$  is a user selected tuning parameter. It is not easy to choose  $\beta$ , but it is possible to consider  $\beta=1$  as giving a pressure to EPR for formulae having variance of each term greater than the variance of the residuals.

#### 4.4.4.2 Multi objective approach

The multi-objective approach in EPR (MO-EPR) is aimed at searching for those model structures, which on one hand comply with the fitness and on the other hand limit the structural complexity. In this approach the control of fitness and complexity is demanded to different singly acting objective functions. The objectives represented by the functions are mutually conflicting, then their optimisation returns a trade-off surface of models. In this scenario, multi-objective modelling in hybrid evolutionary computing enables the user to

- i) Find a set of feasible symbolic models.
- ii) Make a robust choice.
- iii) Gain in computational efficiency developing simultaneously a set of models with variable parsimony levels.

MO-EPR tackles a multi-model strategy by varying the structural parsimony (i.e., the number of constant values in the equation) and working on the objective function used in single-objective EPR. Then, MO-EPR finds the set of symbolic expressions that perform well according to two (or more) conflicting criteria considered simultaneously, the level of agreement between simulated and observed measurements, and structural parsimony of the expressions obtained. The objective functions used are

- i) Maximisation of the fitness.
- ii) Minimisation of the total number of inputs selected by the modelling strategy.
- iii) Minimisation of the length of the model expression.

The solutions searched are ranked according to the Pareto dominance criterion. MO-EPR reduces the computational time required by the multiple executions of EPR, which would otherwise be required for each one of the objective functions introduced in the previous section. The user is presented with the expressions (models) that dominate others in the population of solutions. The Pareto set of solutions is likely to be the best set of expressions required for the analysis of the problem.

The objective functions commonly used to measure the fitness of the symbolic structures are based on the sum of squared errors (SSE) or on the penalisation of complex structures (PCS). The result of the single-objective EPR optimisation consists of a set of equally good models. They might be easily ranked according to their SSE, rather than according to their structural complexity. In fact, sorting models according to their structural complexity is usually a complex task. The multi-objective strategy is implemented to improve both the post-processing phase and the general modelling framework of EPR. Such strategy allows ranking models according to both the coefficient of determination (CoD) and structural complexity. The three objective functions implemented in MO-EPR are

- i) (1-CoD), which has the same meaning as the SSE.

$$CoD = 1 - \frac{N-1}{N} \frac{\sum_N (Y - Y_{mes})^2}{\sum_N (Y_{mes} - avg(Y_{mes}))^2} = 1 - k \cdot SSE$$

$$k = \frac{2(N-1)}{\sum_N (Y_{mes} - avg(Y_{mes}))^2} \quad (4.24)$$

where N is the number of data on which the COD is computed.

- ii) The number of constant values  $a_j$ .  
 iii) The total number of inputs involved in the symbolic expression (% of  $\mathbf{X}_i$ ).

Note that the total number of inputs corresponds to the number of times that each input is involved in the symbolic expression. The user must set the maximum number of constant values, which puts an upper limit on the maximum number of the symbolic expression inputs. Therefore, MO-EPR looks for the best non-dominated models with respect to both structural complexity and fitness performance, i.e., placed on the best Pareto front. Therefore, a direct multi-model approach is provided where the post-processing phase is improved by MO-EPR, which returns models ranked according to both their fitness and their structural complexity.

A further advantage of MO-EPR is the increased pressure to achieve structural parsimony because a large number of  $a_j$  values or a large total number of inputs must be justified by the fitness of the model (note that the Pareto dominance criterion and the

function are to be minimised). The introduced objective functions can be used all together or in a two-objective configuration as

- i)  $(1 - \text{CoD})$  vs.  $(\% \text{ of } \mathbf{X}_i)$ .
- ii)  $(\text{CoD})$  vs.  $(\# \text{ of } a_j)$ .

The choice of the Pareto dominance criterion for the multi-objective optimisation results in the following main advantages (Doglioni 2004):

- i) It is reasonably fast for few objective functions in comparison with the total amount of time required by multiple single-objective sessions.
- ii) It deals *simultaneously* with multiple solutions.
- iii) It is able to provide a uniformly distributed range of Pareto solutions.

#### 4.4.5 Extension of evolutionary polynomial regression

EPR allows pseudo-polynomial expressions as in Equations (4.8) and (4.9), allowing structures such as

$$\begin{aligned}
 Y &= a_0 + \sum_{j=1}^m a_j \cdot (X_1)^{\text{ES}(j,1)} \cdot \dots \cdot (X_k)^{\text{ES}(j,k)} \cdot f\left((X_1)^{\text{ES}(j,k+1)}\right) \cdot \dots \cdot f\left((X_k)^{\text{ES}(j,2k)}\right) && \text{case 1} \\
 Y &= a_0 + \sum_{j=1}^m a_j \cdot f\left((X_1)^{\text{ES}(j,1)} \cdot \dots \cdot (X_k)^{\text{ES}(j,k)}\right) && \text{case 2} \\
 Y &= a_0 + \sum_{j=1}^m a_j \cdot (X_1)^{\text{ES}(j,1)} \cdot \dots \cdot (X_k)^{\text{ES}(j,k)} \cdot f\left((X_1)^{\text{ES}(j,k+1)} \cdot \dots \cdot (X_k)^{\text{ES}(j,2k)}\right) && \text{case 3} \quad (4.25) \\
 Y &= g\left(a_0 + \sum_{j=1}^m a_j \cdot (X_1)^{\text{ES}(j,1)} \cdot \dots \cdot (X_k)^{\text{ES}(j,k)}\right) && \text{case 4}
 \end{aligned}$$

Thus, EPR's model space may be extended by the structures in Equations (4.25), which remain based on polynomial regression as in Equation (4.9). User-specified functions  $f$  reported in Equations (4.25) may be natural logarithmic, exponential, tangent hyperbolic etc. Note that the last structure in Equations (4.25) requires the assumption of an invertible function  $g$ , because of the subsequent stage of parameter estimation. The term pseudo-polynomial expression is used here because the parameters of any of the expressions in (4.25) can be computed as for a linear problem and/or for true polynomial expressions. Moreover, the Equations (4.25) are transformed into the form

of Equation (4.9) during evolutionary search. Finally, the inclusion of exponential and logarithmic functions in the general expression of Equations (4.25) allows EPR to explore a large space of formulae where the analyst's understanding of the physical process warrants their inclusion. However, if such functions are not naturally describing the phenomenon being modeled EPR search would find exponent values for such inputs to be equal to zero.

#### **4.4.6 The application of EPR in geomechanics**

In the beginning EPR methodology was mainly used for environmental modelling by its developers (Giustolisi and Savic 2006, Giustolisi et al. 2007, Doglioni et al. 2008). However the capability and outstanding performance of EPR approach on dealing with problems related to other disciplines of civil engineering including geotechnical, structural and earthquake engineering have been investigated by the author of this thesis and his coworkers. The results of these studies have been published in several conference and journal papers (Javadi and Rezanian 2008, Javadi et al. 2007, Rezanian and Javadi 2006, Rezanian and Javadi 2008a, Rezanian and Javadi 2008b, Rezanian et al. 2008a, Rezanian et al. 2008b).

# CHAPTER 5

## CONSTITUTIVE MODELLING OF SOILS USING EVOLUTIONARY POLYNOMIAL REGRESSION

### 5.1 Introduction

A constitutive law or a material model is the mathematical description of how a material responds to various loadings. This is one of the most intensely researched fields within soil mechanics because of its complexity and the importance of accurate constitutive models for practical engineering problems. The attraction of material modelling has been enhanced significantly with the great increase in development and application of modern and sophisticated computational methods such as the finite element and finite difference methods. In geomechanics, these constitutive laws play a crucial role in providing reliable results from the numerical simulations.

Recent advances in artificial intelligence methods, especially neural network based algorithms, have provided a fundamentally new basis for a different approach to material modelling. A comprehensive review of published works in the field of neural network based material modelling has been provided in chapter 2.

The main difference between analytical and NN based approaches for material modelling is that, while in the analytical approach the material modelling consists of mathematical formulation of constitutive equations and determination of material parameters (Desai and Siriwardane 1984), the neural network based constitutive models directly acquire the information on the material behaviour from the experimental data sets (Jung and Ghaboussi 2006).

Although it has been shown by many researchers that NNs offer great advantages in constitutive modelling of materials; however despite their good performance, these networks have some shortcomings. The main drawback of the NN approach is the large complexity of the network structure, as it represents the knowledge in terms of a weight matrix together with biases which are not accessible to user; in other words NN models give no clue on the way inputs affect the output and are therefore considered as a black box class of model. The other major disadvantage of the NNCM is that the optimum structure of the NN (such as hidden layers, number of neurons, transfer functions, etc.) must be identified a priori. The lack of interpretability of NN based models has restrained them from obtaining their full potential in real world problems (Lu et al. 2001) as the credibility of the artificial intelligence method depends on its ability to explain its conclusion (Widham and Loparo 1989).

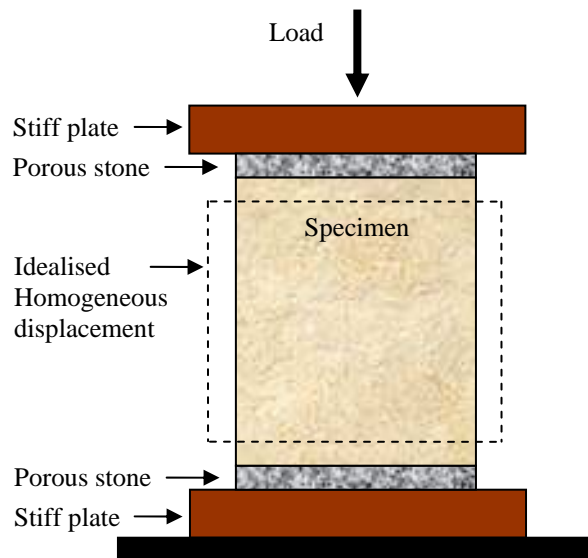
In this chapter the application of the new evolutionary polynomial regression approach for constitutive modelling of nonlinear materials, like soils, is introduced. The strategy uses polynomial structures to take advantage of their favorable mathematical properties. As described in chapter 4, the key idea behind the EPR modelling is to use evolutionary search for exponents of polynomial expressions by means of a GA engine. This allows

- i) Easy computational implementation of the algorithm.
- ii) Efficient search for an expression (formula).
- iii) Improved control of the complexity of the expression generated.
- iv) A small number of search parameters to be pre-specified.

The performance of the new technique in capturing soil nonlinear constitutive relationships will be discussed with several examples.

## 5.2 Constitutive modelling of soils using EPR

Modelling of soil behaviour plays an important role in dealing with issues related to soil mechanics and foundation engineering. Modelling the stress-strain behaviour of geomaterials, enables understanding of different geotechnical phenomena, and makes the required analyses possible. To evaluate the potential of using EPR to derive functions describing the constitutive behaviour of soils, a feasibility study was performed using real experimental data from conventional triaxial tests. In a conventional triaxial test, the boundary conditions are imposed in a way that the material within the sample is subjected to a uniform state of stress; this is done by putting no restriction for lateral displacements at the top and bottom of the sample; thus the specimen represents a material point and undergo just one stress-strain path at each confining pressure. A schematic view of such test, which is the most common type of triaxial experiment, is shown in Figure 5.1.



**Figure 5.1** A schematic view of conventional triaxial experiment.

### 5.2.1 Drained condition

A series of conventional drained triaxial tests was conducted to obtain the required data for model development. The data consisted of the results from five separate triaxial tests, each at a different confining cell pressure,  $\sigma_3$ . The range of  $\sigma_3$  for the tests was

between 50 kPa and 500 kPa. These tests were strain-controlled with a constant rate of increment of axial strain.

### 5.2.1.1 Input and output parameters and data preparation

The choice of input and output quantities is determined by both the source of the data and the way the trained EPR model is to be used. It was shown in chapter (2) that a very typical scheme to train most of the neural network based material models includes an input set providing the network with the information relating to the current state units (e.g. current stresses and current strains) and then a forward pass through the neural network yields the prediction of the next expected state of stress and/or strain relevant to an input strain or stress increment (Ghaboussi et al 1998, Habibagahi and Bamdad 2003). The same idea has been utilised in this research. Thus the inputs to the EPR model were chosen as; the current state of deviator stress,  ${}^i q$ , the cell pressure,  $\sigma_3$ , the current axial strain,  ${}^i \varepsilon_1$ , and the axial strain increment,  ${}^i(\Delta\varepsilon_1)$ . First three of these four input parameters were used to describe the current state of the stress and strain to the model. For the current stress-strain state, the objective of the EPR constitutive model was to capture the output parameter: deviator stress for the next state ( ${}^{i+1} q$ ), corresponding to the input incremental axial strain,  ${}^i(\Delta\varepsilon_1)$ . The resultant input and output vectors were therefore

Input:  ${}^i q, \sigma_3, {}^i \varepsilon_1, {}^i(\Delta\varepsilon_1)$

Output:  ${}^{i+1} q$

The target, as stated before, was to find the constitutive relationship in the general form of Equation (4.8), where the matrix of inputs (i.e.,  $\mathbf{X}$ ), for each particular confining pressure, is

$$X = \begin{bmatrix} {}^1q & \sigma_3 & {}^1\varepsilon_1 & {}^1(\Delta\varepsilon_1) \\ {}^2q & \sigma_3 & {}^2\varepsilon_1 & {}^2(\Delta\varepsilon_1) \\ {}^3q & \sigma_3 & {}^3\varepsilon_1 & {}^3(\Delta\varepsilon_1) \\ \dots & \dots & \dots & \dots \\ {}^iq & \sigma_3 & {}^i\varepsilon_1 & {}^i(\Delta\varepsilon_1) \end{bmatrix}$$

The input parameter for deviatoric stress,  ${}^iq$ , is the input variable that must be updated incrementally during the training process, based on the outputs received from the previous increment of training. Also the input parameter for axial strain,  ${}^i\varepsilon_1$ , is the input variable that must be updated based on the previous increment of strain  ${}^{i-1}(\Delta\varepsilon_1)$ , in the input set. Confining pressure,  $\sigma_3$ , is a constant variable for each test.

As the original database was related to a series of conventional strain-controlled triaxial tests (where  $\Delta\varepsilon_1$  was predetermined and constant); before training the model with the input data, the stress states relating to each confining pressure were randomly shuffled in order to find different values of  $\Delta\varepsilon_1$  at each state. In this way, the training database can represent a much wider pattern for soil behaviour which makes the trained model capable of giving different output values for different axial strain increments at each stress state. The algorithm for data shuffling was simply based on randomly selecting two data rows and swapping them together. This approach was also used to obtain more data points in the region of small axial strain values to better define the large changes that are often observed in this region for the values of deviator stress. The major drawback of a fixed small constant value for axial strain increment, which is commonly used in training the neural network material models, is that it yields a very large training set for each triaxial test. But with using varying  $\Delta\varepsilon_1$  values, it was found that good modelling capabilities were obtained without the need for a large size of training pattern data. The original data corresponding to the five stress paths included 1126 stress states; after the data preparation process the resultant dataset consisted of 1544 stress states. Some part of the data became duplicated depending on the random ranking of the stress states in the shuffling process. In order to obtain an efficient training any duplicated stress-strain relationships in the expanded dataset was removed. This reduced the 1544 cases of stress states in the dataset to just 537 cases. It was observed that, in this way not only the training data could represent more features of soil behaviour but the size of data could also be reduced by half which is desirable as it minimises the computational

time. It should be noted that unlike ANN based constitutive models (Zhu et al., 1998; Hashash et al., 2004) in EPR the values of all participating parameters can be fed into the model as they are, without any need for normalisation or calibration.

### **5.2.1.2 Training and validation subsets**

The database was divided into two separate sets. One set was used as training part to obtain the constitutive model and the other one was used as validation part to appraise the applicability of the trained constitutive model. The training set included the triaxial test results related to three confining cell pressures of 50 kPa, 200 kPa and 500 kPa; this consisted of 318 stress states of the prepared dataset (around 59% of the data). The results from triaxial tests at 100 kPa and 400 kPa confining pressures were kept to examine the generalisation capability of the developed EPR based constitutive model; this consisted of the remaining 219 stress states in the prepared dataset (around 41% of the data).

### **5.2.1.3 EPR procedure and results**

After feeding the data and before starting the evolutionary procedure, a number of constraints can be put in place to control the evolved models in terms of EPR structure, type of functions used, length of equations, number of terms, range of exponents and number of generations. It can be seen that there is great potential in achieving different models for a particular problem which enables the user to gain additional information for different scenarios.

#### *Finding the Optimal Structure*

A series of single objective EPR analyses was considered in order to find the optimum EPR algorithm for constitutive modelling in terms of EPR structure and type of the function utilised. For simplicity, in all the analyses the maximum number of terms was set to 5 and the range of potential exponents was selected as integer values between -2 and 2. Among the candidate exponents it is advisable to include the value zero, which

helps in discarding those variables or inputs which are not useful for models (Doglioni et al. 2008). For each analysis a specific combination of EPR structure and function was selected. At each particular setting during the evolutionary process, different participating parameters are gradually picked up in order to form the equations representing the input-output relationship. The level of accuracy for the output EPR models at each generation is evaluated based on the fitness function as described in the previous chapter. If the model fitness is not acceptable or the other termination criteria (in terms of maximum number of generations and maximum number of terms) are not satisfied, the current model should go through another evolution in order to obtain a new model. The EPR analyses with these settings were performed on a Pentium 4 personal computer with 3.00 GHz of processor speed and 512 MB of memory. The results of these analyses are summarised in Tables 5.1 to 5.4.

**Table 5.1** The summary of results obtained for EPR based constitutive models using EPR structure type 1.

Function	Maximum generation	Duration	Error in training	COD value for validation (%)	Profile
None	303	03' 27" 23	8.00E-04	99.8999	Converging
Logarithm	1853	10' 25" 54	8.00E-04	99.8521	Converging
Exponential	-	-	-	-	Stuck
Tangent hyperbolic	1232	10' 26" 23	6.00E-04	99.8682	Converging
Secant hyperbolic	1447	11' 40" 46	7.00E-04	99.9276	Converging

**Table 5.2** The summary of results obtained for EPR based constitutive models using EPR structure type 2.

Function	Maximum generation	Duration	Error in training	COD value for validation (%)	Profile
None	303	03' 27" 23	8.00E-04	99.8996	Converging
Logarithm	754	04' 27" 55	9.00E-04	99.9118	Converging
Exponential	236	04' 02" 06	1.00E-03	99.9037	Converging
Tangent hyperbolic	289	04' 20" 37	9.00E-04	99.8727	Converging
Secant hyperbolic	899	04' 38" 12	8.00E-04	99.8476	Converging

**Table 5.3** The summary of results obtained for EPR based constitutive models using EPR structure type 3

Function	Maximum generation	Duration	Error in training	COD value for validation (%)	Profile
None	448	09' 26" 08	9.00E-04	99.8996	Converging
Logarithm	1129	10' 50" 73	9.00E-04	99.9118	Converging
Exponential	-	-	-	-	Stuck
Tangent hyperbolic	1798	10' 01" 05	8.00E-04	99.8727	Converging
Secant hyperbolic	1158	12' 24" 52	7.00E-04	99.8476	Converging

**Table 5.4** The summary of results obtained for EPR based constitutive models using EPR structure type 4

Function	Maximum generation	Duration	Error in training	COD value for validation (%)	Profile
None	416	04' 37" 78	8.00E-04	99.8538	Converging
Logarithm	-	-	-	-	Stuck
Exponential	-	-	-	-	Stuck
Tangent hyperbolic	-	-	-	-	Stuck
Secant hyperbolic	-	-	-	-	Stuck

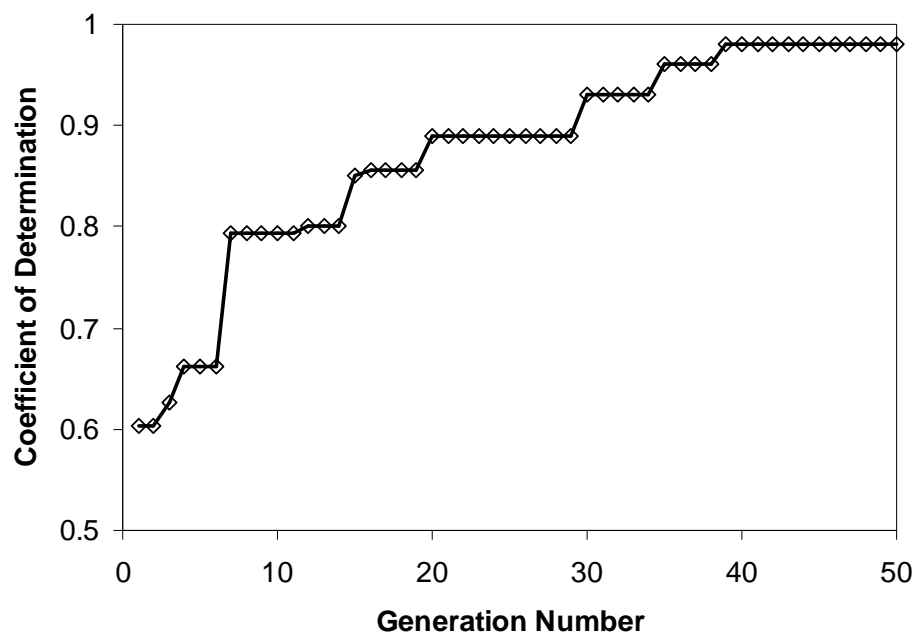
Note: For EPR structure type 4 the data must be normalised between 0 and 1.

From the results shown in Tables 5.1 to 5.4, it can be seen that generally making of available functions in EPR, increases the computational time while it does not necessarily increase the accuracy of the predictions. It should be noted that the application of these functions also lead to more complex relationships for constitutive modelling. For EPR structure type 4, the input and output data need to be normalised between 0 and 1 which is not desirable as normalisation makes the final model more complicated. The best results, in terms of simplicity, lower number of generations and shorter computational time, were obtained with EPR structures 1 and 2 without use of available functions. The results of EPR analyses for these two structures are identical as without using a function these two structures are basically the same. Consequently EPR structure 2 without use of any function can be selected as optimum combination of EPR structure and function for constitutive modelling. This is also in agreement with the findings reported in Rezanian et al. (2008).

*EPR Based Material Model*

To show the efficiency of the EPR approach in finding the constitutive relationships for the soil represented with the provided data, the results for different stages of one optimal analysis are presented here. For this specific analysis the optimal EPR structure for material modelling (i.e., EPR structure type 2 with the use of no function) was used. The multi-objective strategy was also employed in which the fitness of models was evaluated based on the trade-off between the model complexity (i.e., addition of new terms) and the quality of fit (i.e., COD value). The range of exponents was set as integer values between -5 and 5, including -0.5 and 0.5. The maximum number of terms in an equation was set to 15, which was found to be adequate.

The EPR procedure with the setting outlined above was found to be very efficient in constitutive modelling. Figure 5.2 shows the variation of the coefficient of determination (COD) with the number of generations. It is shown that the COD value increases rapidly with increasing the number of generations, even at early generations.



**Figure 5.2** Increase in COD value at early generations up to 25 seconds after starting the EPR procedure on drained triaxial data.

As mentioned above, during the evolutionary search, EPR model selection was lead by a multi-objective strategy. The two objectives assumed for this search were: (i) the number of constants  $a_i$  and (ii) the model fitness to data, in term of coefficient of determination estimated on the training set of data. This approach ranks the returned

formulae according to the fitness to data and the number of constant values. This represents a great advantage in the analysis of results, since sorting models according to their structural complexity vs. fitting to training data is useful during the model selection phase (Doglioni et al. 2008). After completion of EPR analysis the following equations were obtained.

$${}^{i+1}q = 275.65 \quad (5.1)$$

$${}^{i+1}q = 1.01 {}^i q + 15.84 \quad (5.2)$$

$${}^{i+1}q = 0.09 \sigma_3 \sqrt{{}^i(\Delta \varepsilon_1)} + 0.97 {}^i q + 11.94 \quad (5.3)$$

$${}^{i+1}q = -7.80 \sqrt{{}^i \varepsilon_1} \sqrt{{}^i(\Delta \varepsilon_1)} + 2.74 \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} + 0.98 {}^i q + 7.84 \quad (5.4)$$

$${}^{i+1}q = -2.13E - 9 \frac{\sigma_3^4 {}^i \varepsilon_1 {}^i(\Delta \varepsilon_1)}{{}^i q} - 8.98 \sqrt{{}^i \varepsilon_1} \sqrt{{}^i(\Delta \varepsilon_1)} + 3.56 \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} + 0.98 {}^i q + 4.69 \quad (5.5)$$

$${}^{i+1}q = -1.19 \frac{\sigma_3^4 {}^i \varepsilon_1^2 {}^i(\Delta \varepsilon_1)^2}{{}^i q^2} + 2.83E - 10 \sigma_3^4 \sqrt{{}^i \varepsilon_1} \sqrt{{}^i(\Delta \varepsilon_1)} + 62.30 \sqrt{{}^i(\Delta \varepsilon_1)} + 0.96 {}^i q - 19.01 \frac{{}^i q \sqrt{{}^i(\Delta \varepsilon_1)}}{\sigma_3} + 4.24 \quad (5.6)$$

$${}^{i+1}q = -1.12 \sqrt{\sigma_3} + 9.40 \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} - 0.44 \sqrt{{}^i q} \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} + 6.22E - 10 \sigma_3^4 \sqrt{{}^i \varepsilon_1} \sqrt{{}^i(\Delta \varepsilon_1)} + 1.01 {}^i q - 2.56 \frac{{}^i q^2 \sqrt{{}^i(\Delta \varepsilon_1)}}{\sigma_3^2} + 7.65 \quad (5.7)$$

$${}^{i+1}q = -9.61E - 9 \frac{\sigma_3^4 {}^i \varepsilon_1^3 {}^i(\Delta \varepsilon_1)^3}{{}^i q^2} + 9.46 \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} - 1.30 \sqrt{\sigma_3} + 5.34E - 10 \sigma_3 \sqrt{{}^i \varepsilon_1} \sqrt{{}^i(\Delta \varepsilon_1)} - 0.38 \sqrt{{}^i q} \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} + 1.01 {}^i q - 9.25 \frac{{}^i q \sqrt{{}^i(\Delta \varepsilon_1)}}{\sigma_3} + 9.60 \quad (5.8)$$

$${}^{i+1}q = -2.77E - 6 \frac{\sigma_3^3 {}^i \varepsilon_1^3 {}^i(\Delta \varepsilon_1)^3}{{}^i q^2} + 26.75 {}^i(\Delta \varepsilon_1) \sqrt{\frac{{}^i \varepsilon_1}{{}^i q}} - 1.46 \sqrt{\sigma_3} - 0.40 \sqrt{{}^i q} \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} + 9.56 \sqrt{\sigma_3} \sqrt{{}^i(\Delta \varepsilon_1)} + 5.55E - 10 \sigma_3^4 \sqrt{{}^i \varepsilon_1} \sqrt{{}^i(\Delta \varepsilon_1)} + 1.01 {}^i q - 12.52 \frac{{}^i q \sqrt{{}^i(\Delta \varepsilon_1)}}{\sigma_3} + 12.30 \quad (5.9)$$

$$\begin{aligned}
 {}^{i+1}q = & -0.0068 \frac{\sigma_3^2 \varepsilon_1^3 (\Delta \varepsilon_1)^2}{q^2} + 71.43 \sqrt{(\Delta \varepsilon_1)} + 4.16 \sqrt{\sigma_3^i (\Delta \varepsilon_1)} + 2.71E - 7 \sigma_3 \sqrt{\varepsilon_1^i (\Delta \varepsilon_1)} + \\
 & 80.01 \frac{\sqrt{q}}{\sigma_3} + 7.90 \frac{\varepsilon_1^i (\Delta \varepsilon_1) \sqrt{q}}{\sigma_3} - 30.06 \frac{q \sqrt{(\Delta \varepsilon_1)}}{\sigma_3} + 0.99^i q - 0.27 \sqrt{q \sigma_3^i (\Delta \varepsilon_1)} - \\
 & 12.40
 \end{aligned} \tag{5.10}$$

$$\begin{aligned}
 {}^{i+1}q = & -3.52E - 5 \frac{\sigma_3^3 \varepsilon_1^3 (\Delta \varepsilon_1)^2}{q^2} + 62.56 \sqrt{(\Delta \varepsilon_1)} + 4.30 \sqrt{\sigma_3^i (\Delta \varepsilon_1)} + 0.46 \varepsilon_1^i (\Delta \varepsilon_1) \sqrt{\frac{q}{\sigma_3}} + \\
 & 3.22E - 7 \sigma_3 \sqrt{\varepsilon_1^i (\Delta \varepsilon_1)} + 12.50 \sqrt{\frac{q}{\sigma_3}} - 0.26 \sqrt{q \sigma_3^i (\Delta \varepsilon_1)} - 24.73 \frac{q \sqrt{(\Delta \varepsilon_1)}}{\sigma_3} + \\
 & 0.95^i q + 1.95E - 5 \frac{q^2}{(\Delta \varepsilon_1)} \sqrt{\frac{\varepsilon_1}{\sigma_3}} - 11.56
 \end{aligned} \tag{5.11}$$

$$\begin{aligned}
 {}^{i+1}q = & -1.54E - 6 \frac{\sigma_3^2 \varepsilon_1^3 (\Delta \varepsilon_1)^3}{q} + 129.28 \frac{\varepsilon_1^i (\Delta \varepsilon_1)}{\sqrt{q \sigma_3}} + 61.70 \sqrt{(\Delta \varepsilon_1)} + 15.43 \sqrt{\frac{q}{\sigma_3}} + \\
 & 5.51E - 10 \sigma_3^4 \sqrt{\varepsilon_1^i (\Delta \varepsilon_1)} - 0.30 \sqrt{q \sigma_3^i (\Delta \varepsilon_1)} + 0.94^i q - 27.64 \frac{q \sqrt{(\Delta \varepsilon_1)}}{\sigma_3} + \\
 & 4.24E - 6 \frac{q^3}{\sqrt{\sigma_3}} - 1.51E - 10^i q^4 - 14.08
 \end{aligned} \tag{5.12}$$

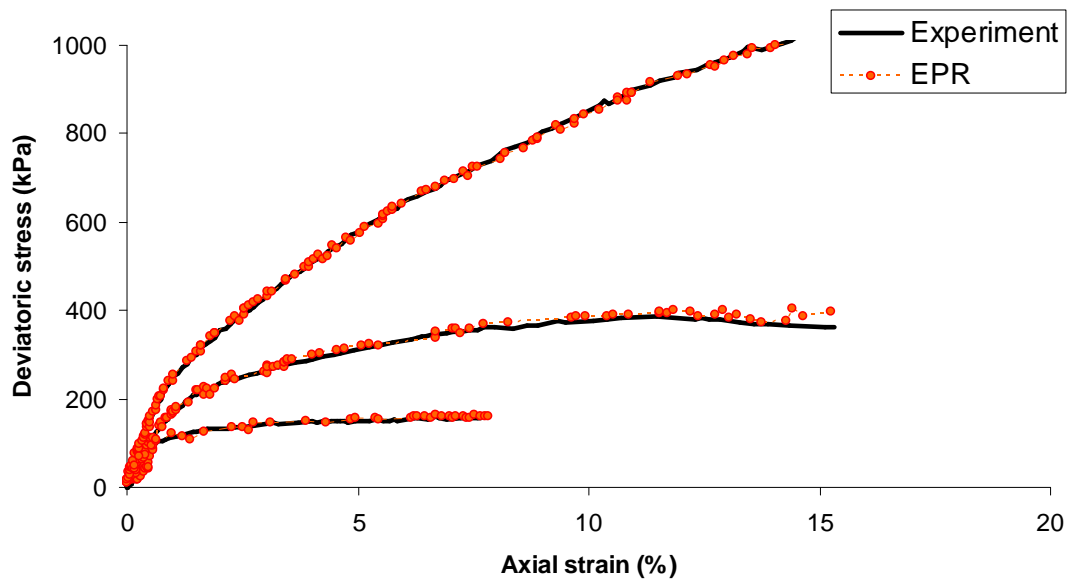
$$\begin{aligned}
 {}^{i+1}q = & -0.00057 \frac{\sigma_3^2 \varepsilon_1^3 (\Delta \varepsilon_1)^3}{q^2} + 161.44 \frac{\varepsilon_1^i (\Delta \varepsilon_1)}{\sqrt{q \sigma_3}} + 69.75 \sqrt{(\Delta \varepsilon_1)} + 5.90 \sqrt{\sigma_3^i (\Delta \varepsilon_1)} + \\
 & \frac{0.82}{\sqrt{(\Delta \varepsilon_1)}} + 20.03 \sqrt{\frac{q}{\sigma_3}} + 6.03E - 10 \sigma_3^4 \sqrt{\varepsilon_1^i (\Delta \varepsilon_1)} - 0.34 \sqrt{q^i \varepsilon_1^i (\Delta \varepsilon_1)} + 0.94^i q - \\
 & 31.39 \frac{q \sqrt{(\Delta \varepsilon_1)}}{\sigma_3} + 4.70E - 6 \frac{q^3}{\sqrt{\sigma_3}} - 1.68E - 10^i q^4 - 23.42
 \end{aligned} \tag{5.13}$$

$$\begin{aligned}
 {}^{i+1}q = & -2.80E - 6 \frac{\sigma_3^3 \varepsilon_1^3 (\Delta \varepsilon_1)^3}{q^2} + 292.84 \sqrt{\frac{(\Delta \varepsilon_1)}{\sigma_3}} + 0.11 \sqrt{\frac{\sigma_3}{(\Delta \varepsilon_1)}} + 10.96 \sqrt{\sigma_3^i (\Delta \varepsilon_1)} - \\
 & 1.95 \sqrt{\sigma_3} + 2.74 \sqrt{q} + 6.71E - 10 \sigma_3^4 \sqrt{\varepsilon_1^i (\Delta \varepsilon_1)} - 0.45 \sqrt{q \sigma_3^i (\Delta \varepsilon_1)} + 0.88^i q - \\
 & 0.001 \varepsilon_1^2 \sqrt{q \sigma_3} - 28.89 \frac{q \sqrt{(\Delta \varepsilon_1)}}{\sigma_3} + 0.93 \frac{q^i (\Delta \varepsilon_1) \sqrt{\varepsilon_1}}{\sigma_3} + 0.02^i q^i \varepsilon_1 - 8.47
 \end{aligned} \tag{5.14}$$

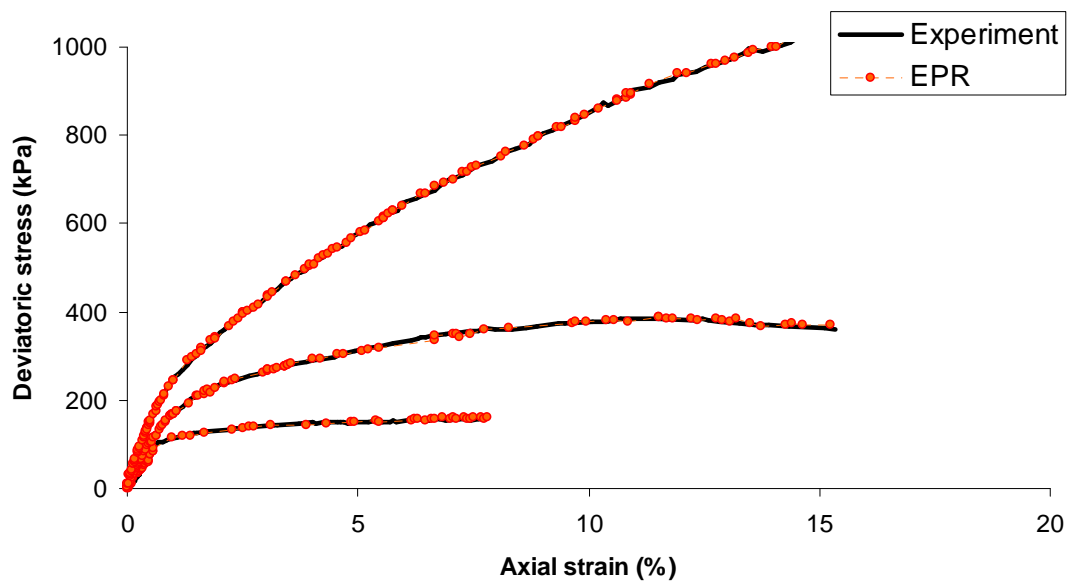
$$\begin{aligned}
 {}^{i+1}q = & -3.22E - 5 \frac{\sigma_3^3 \varepsilon_1^3 (\Delta \varepsilon_1)^2}{q^2} + 273.48 \sqrt{\frac{(\Delta \varepsilon_1)}{\sigma_3}} + 10.43 \sqrt{\sigma_3^i (\Delta \varepsilon_1)} + 0.005 \frac{\sigma_3}{\sqrt{(\Delta \varepsilon_1)}} - \\
 & 0.0009 \varepsilon_1^2 \sqrt{q \sigma_3} + 4.24E - 7 \sigma_3^4 \sqrt{\varepsilon_1^i (\Delta \varepsilon_1)} + 0.84^i q + 0.49 \varepsilon_1^i (\Delta \varepsilon_1) \sqrt{\frac{q}{\sigma_3}} + \\
 & 3.05 \sqrt{q} - 0.45 \sqrt{q \sigma_3^i (\Delta \varepsilon_1)} - 0.07 \sigma_3 - 25.49 \frac{q \sqrt{(\Delta \varepsilon_1)}}{\sigma_3} + 0.013^i q^i \varepsilon_1 + \\
 & 5.53E - 10 \frac{q^3 \sqrt{\sigma_3}}{(\Delta \varepsilon_1) \sqrt{\varepsilon_1}} - 16.78
 \end{aligned} \tag{5.15}$$

It should be mentioned that model selection from the results of EPR analysis requires some subjective judgment as well. This process is often biased by the analyst's experience rather than being purely based on mathematical criteria (Rezania et al. 2008). As it can be seen in Equations (5.1), (5.2) and (5.3) the first three evolutionary steps do not include the effect of all input parameters. Consequently these equations are not appropriate for modelling purposes. Just after the fourth evolution EPR starts to include all participating parameters to form the models. The results of EPR training on the three stress-strain curves are shown in Figure 5.3. Figure 5.3a shows the stress-strain relationships predicted by Equation (5.4) (as marker points) against those expected and used as training data (as solid lines). Figure 5.3b shows the same results using Equation (5.15). It can be seen that excellent agreement was obtained between the experimentally determined values and the EPR predicted values. One particular attraction of the results is that even the model produced at early stages of EPR evolution is capable of learning the material behaviour with very good accuracy (see Figure 5.3a). However the model precision improves noticeably with increasing the number of evolutions as shown in Figure 5.3b.

Once the training is complete, the performance of the trained EPR model is validated using the validation data which have not been used as a part of the model building process. The purpose of validation is to examine the capabilities of the trained model to generalise the training to conditions that have not been seen during the training phase. In order to demonstrate the capability of trained EPR models over the validation data, Equations (5.4) and (5.15) were used to predict the stress-strain curves for the two remaining confining pressures for which the data were not used in the training process and the results are shown in Figure 5.4.

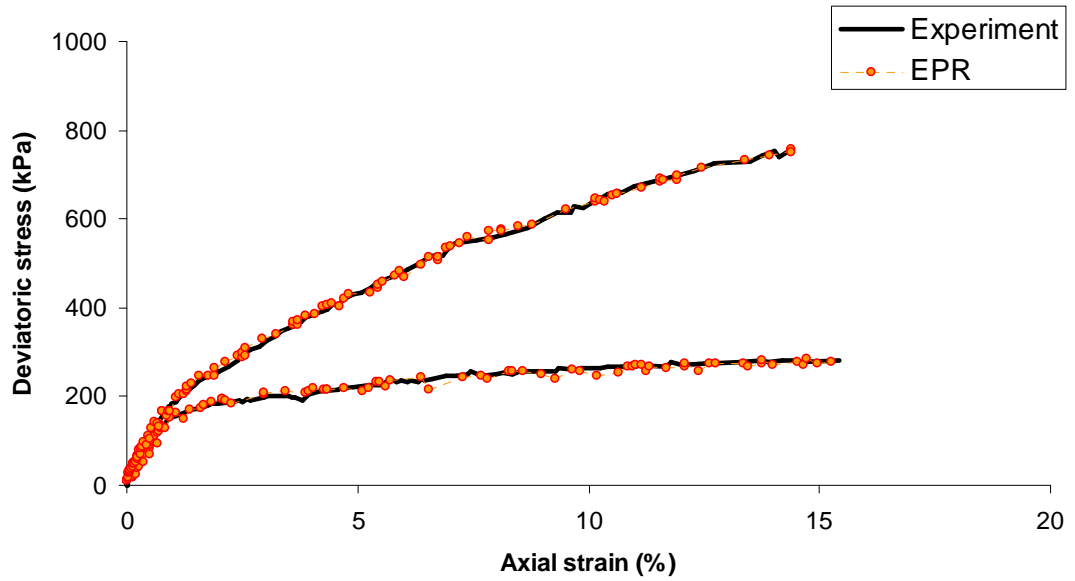


(a)

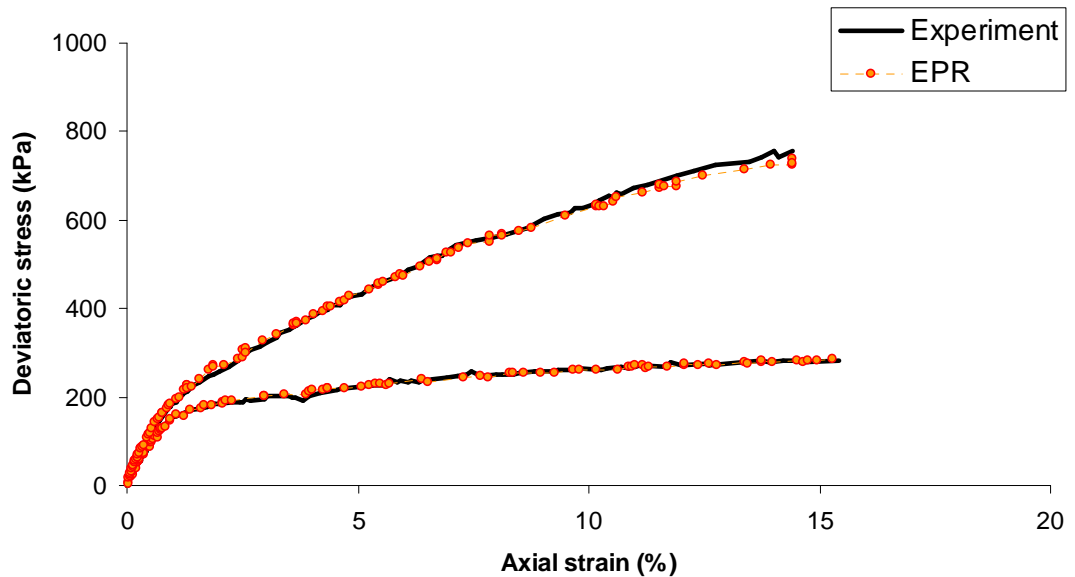


(b)

**Figure 5.3** Results of training the EPR models using (a) Equation (5.4) and (b) Equation (5.15).



(a)



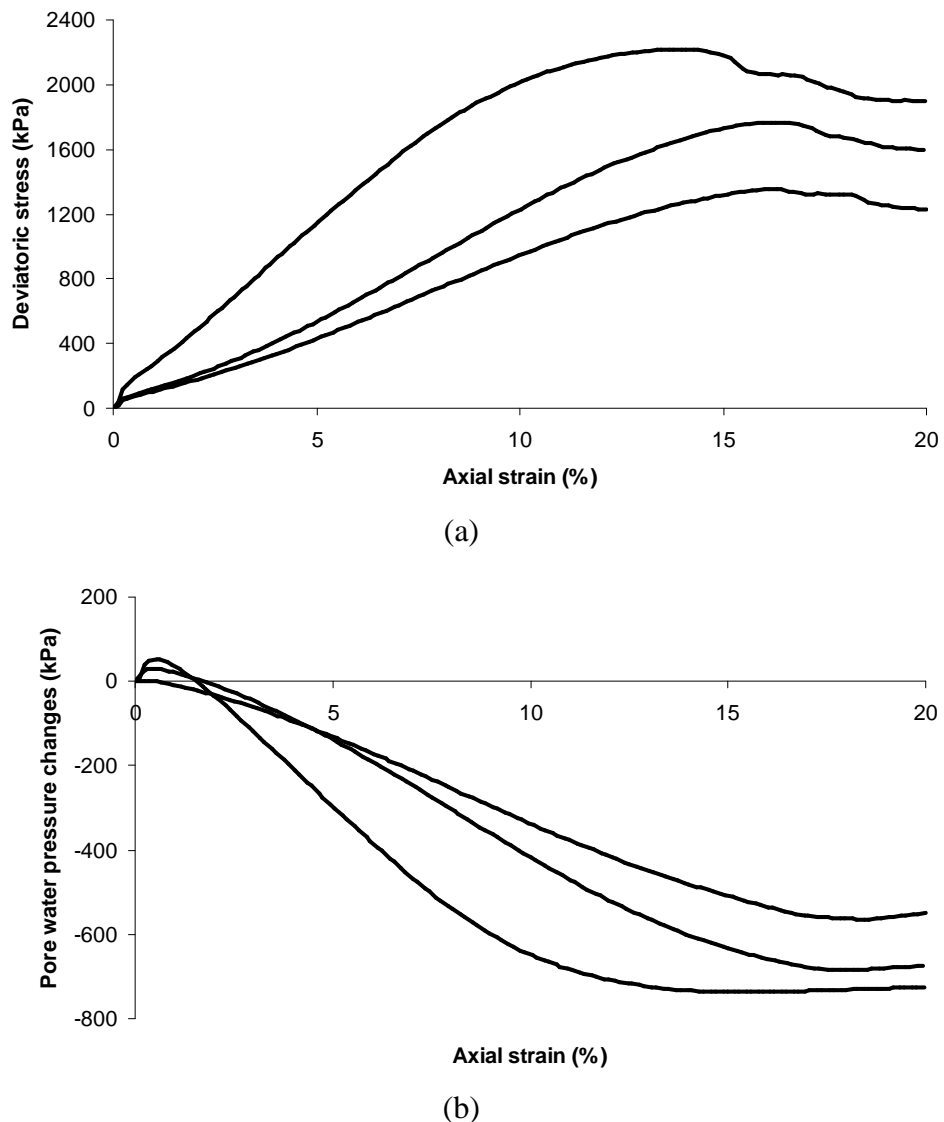
(b)

**Figure 5.4** Results of testing the EPR models using (a) Equation (5.4) and (b) Equation (5.15).

Excellent agreement between modelling results and laboratory experimental data are observed in Figure 5.4, which demonstrates the high capability of the EPR based material model in generalising the constitutive relationship for unseen cases.

### 5.2.2 Undrained condition

For the undrained condition the results of triaxial tests performed on Nevada sand and reported by Arulmoli et al. (1992) were used. The tests were performed under isotropically consolidated undrained condition. Three effective consolidation pressures (40 kPa, 80 kPa and 160 kPa) were used during the testing program. The results of all individual tests in terms of deviatoric stress,  $q$ , versus axial strain,  $\epsilon_1$  and pore pressure change,  $u - u_0$ , versus axial strain,  $\epsilon_1$ , are plotted in Figure 5.5. As it can be seen in the figures the negative water pressure indicates that the samples had high pressures of over consolidation. In all three cases, the tests were continued until axial strain of about 20 percent; however, the samples yielded at axial strain of around 10 to 15 percent, after this strain the samples generally showed strain softening behaviour.



**Figure 5.5** The results of undrained triaxial test on Nevada sand in terms of (a) deviatoric stress versus axial strain and (b) pore pressure change versus axial strain.

### 5.2.2.1 Input and output parameters

With regard to the available data for EPR training, the inputs to the EPR model for undrained triaxial tests were chosen as; the current state of deviator stress  ${}^i q$ , the consolidation pressure,  $\sigma_3$ , the effective confining pressure,  ${}^i \sigma'_3$ , the current axial strain  ${}^i \varepsilon_1$ , and the axial strain increment,  ${}^i(\Delta\varepsilon_1)$ . The first four of these input parameters were used to describe the current state of the stress and strain to the model. For the current stress-strain state, the objective of the EPR constitutive model was to capture the output parameter: deviator stress for the next state,  ${}^{i+1} q$ , corresponding to the input incremental axial strain,  ${}^i(\Delta\varepsilon_1)$ . It should be noted that the effect of pore water pressure at every stage of each test is implicitly considered in the input data as it is embedded in the difference of the effective confining pressure with consolidation pressure. The resultant input and output vectors were therefore

Input:  ${}^i q, {}^i \sigma'_3, \sigma_3, {}^i \varepsilon_1, {}^i(\Delta\varepsilon_1)$

Output:  ${}^{i+1} q$

Like the drained tests described in the previous section, the undrained data were also related to a series of strain-controlled tests (i.e.,  $\Delta\varepsilon_1$  was predetermined and constant); thus before training the model with the input data, the same procedure as described in section 5.2.1.1 was utilised in order to enhance the data by randomly shuffling the data to find different values of  $\Delta\varepsilon_1$  at each stress state.

### 5.2.2.2 Training and validation subsets

For undrained condition, the database was divided into two separate subsets. One set was used for training and the other one was used for validation of the trained EPR model. The training set included the triaxial test results related to two consolidation pressures of 40 kPa and 160 kPa and the results from triaxial test at 80 kPa consolidation pressure was kept to examine the generalisation capability of each developed EPR based material model.

### 5.2.2.3 EPR procedure and results

The initial setting of EPR procedure for constitutive modelling of soil under undrained condition was kept identical to those described for the drained condition in section 5.2.1.3. Because the stress-strain response of undrained tests on Nevada sand shows more complex behaviour (including strain softening feature), two different conditions were considered: (i) first the part of data just related to strain hardening behaviour of the triaxial samples (strains up to 15 percent) was considered, (ii) secondly the entire data (related to strains up to 20 percent) was examined to train the EPR models. The following equations ((5.16) to (5.30)) were obtained after completion of EPR analyses for the strain hardening part of the data.

$${}^{i+1}q = 965.36 \quad (5.16)$$

$${}^{i+1}q = 0.394 {}^i q^2 {}^i \sigma_3^{-1} + 154.89 \quad (5.17)$$

$${}^{i+1}q = 14.03 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \sigma_3} + 0.97 {}^i q + 21.48 \quad (5.18)$$

$${}^{i+1}q = 15.92 {}^i (\Delta \varepsilon_1) - 8.27E-11 {}^i q^3 {}^i \sigma_3 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \varepsilon_1 \sigma_3^{-1}} + 1.02 {}^i q - 18.48 \quad (5.19)$$

$${}^{i+1}q = -729.15 \sqrt{\frac{{}^i (\Delta \varepsilon_1)}{{}^i \sigma_3}} + 18.58 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \sigma_3} + 0.99 {}^i q - 5.67E-9 {}^i q^3 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \varepsilon_1} + 32.84 \quad (5.20)$$

$${}^{i+1}q = 1.78E-5 \sigma_3^4 \sqrt{\frac{{}^i \varepsilon_1 {}^i (\Delta \varepsilon_1)}{{}^i q {}^i \sigma_3}} - 108.08 \sqrt{\frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{{}^i \sigma_3}} + 0.99 {}^i q + 18.61 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \sigma_3} - 1.43E-8 {}^i q^2 {}^i \sigma_3 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \varepsilon_1} + 34.55 \quad (5.21)$$

$${}^{i+1}q = -55.73 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{{}^i \sigma_3} + 5.76E-8 \frac{\sigma_3^5 {}^i (\Delta \varepsilon_1)}{{}^i \sigma_3} + 18.82 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \sigma_3} - 27.90 \sqrt{\frac{{}^i q}{{}^i \sigma'}} + 1.01 {}^i q - 1.78E-7 {}^i q {}^i (\Delta \varepsilon_1) \sqrt{\frac{{}^i \varepsilon_1}{{}^i \sigma'}} + 33.17 \quad (5.22)$$

$${}^{i+1}q = -0.17 \frac{{}^i \varepsilon_1^5 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \sigma_3}}{\sigma_3^2 \sqrt{{}^i q}} - 10.59 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma_3}} + 1.50E-6 \frac{\sigma_3^4 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma_3}} + 21.73 {}^i (\Delta \varepsilon_1) \sqrt{{}^i \sigma_3} - 25.51 \sqrt{\frac{{}^i q}{{}^i \sigma_3}} + 1.01 {}^i q - 2.24E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 30.87 \quad (5.23)$$

$$\begin{aligned}
 {}^{i+1}q = & -7.92 \frac{{}^i \varepsilon_1^4 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i q} {}^i \sigma'_3 \sigma_3} - 8.39 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 9.21E-9 \frac{\sigma_3^5 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 19.38 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} + \\
 & 0.80 \sqrt{{}^i \sigma'_3} {}^i \varepsilon_1 {}^i (\Delta \varepsilon_1) - 28.36 \sqrt{\frac{{}^i q}{{}^i \sigma'_3}} + 0.99 {}^i q - 2.38E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 32.82
 \end{aligned} \quad (5.24)$$

$$\begin{aligned}
 {}^{i+1}q = & -404.05 \frac{{}^i \varepsilon_1^5 {}^i (\Delta \varepsilon_1)}{{}^i q {}^i \sigma'_3 \sqrt{\sigma_3}} + 7.52 {}^i (\Delta \varepsilon_1)^2 \sqrt{\frac{\sigma_3}{{}^i \sigma'_3}} - 11.46 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 3.19E-4 \frac{\sigma_3^3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + \\
 & 17.29 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} + 3.39 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - 25.75 \sqrt{\frac{{}^i q}{{}^i \sigma'_3}} + 0.99 {}^i q - 2.28E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + \\
 & 26.82
 \end{aligned} \quad (5.25)$$

$$\begin{aligned}
 {}^{i+1}q = & -1.75 \frac{{}^i \varepsilon_1^5 {}^i (\Delta \varepsilon_1)}{{}^i \sigma'_3 \sigma_3} + 18.06 {}^i (\Delta \varepsilon_1)^2 \sqrt{\frac{\sigma_3}{{}^i \sigma'_3}} - 15.03 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 3.57E-4 \frac{\sigma_3^3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + \\
 & 20.59 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} - 21.51 \sqrt{\frac{{}^i q}{{}^i \sigma'_3}} + 0.10 {}^i \sigma'_3 \sqrt{{}^i (\Delta \varepsilon_1)} - 0.21 {}^i (\Delta \varepsilon_1)^3 \sqrt{\frac{{}^i q}{{}^i \varepsilon_1}} + 0.99 {}^i q - \\
 & 2.47E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 31.81
 \end{aligned} \quad (5.26)$$

$$\begin{aligned}
 {}^{i+1}q = & -2.74 \frac{{}^i \varepsilon_1^5 {}^i (\Delta \varepsilon_1)}{\sigma_3 \sqrt{{}^i q} {}^i \sigma'_3} + 19.03 {}^i (\Delta \varepsilon_1)^2 \sqrt{\frac{\sigma_3}{{}^i \sigma'_3}} - 14.57 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 3.47 \frac{\sigma_3^3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} - \\
 & 20.30 \sqrt{{}^i (\Delta \varepsilon_1)} + 20.24 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} + 2.88 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - 23.35 \sqrt{\frac{{}^i q}{{}^i \sigma'_3}} - \\
 & 0.22 {}^i (\Delta \varepsilon_1)^3 \sqrt{\frac{{}^i q}{{}^i \varepsilon_1}} + 0.99 {}^i q - 2.30E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 34.46
 \end{aligned} \quad (5.27)$$

$$\begin{aligned}
 {}^{i+1}q = & -147.83 \frac{{}^i \varepsilon_1^4 {}^i (\Delta \varepsilon_1)}{{}^i \sigma'_3 \sqrt{{}^i q} \sigma_3} + \frac{2781.12}{{}^i \sigma'_3} - 186.51 \sqrt{\frac{{}^i \varepsilon_1 {}^i (\Delta \varepsilon_1)}{{}^i \sigma'_3}} + 1.19E-8 \sigma_3^5 \sqrt{\frac{{}^i (\Delta \varepsilon_1)}{{}^i \sigma'_3}} - \\
 & \frac{733.42}{\sqrt{\sigma_3}} + \frac{1.04}{\sqrt{{}^i \varepsilon_1}} + 105.46 \sqrt{{}^i (\Delta \varepsilon_1)} + 16.64 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} - 5.30 \sqrt{\frac{{}^i q \sigma_3}{{}^i \sigma'_3}} + 1.01 {}^i q - \\
 & 1.04 \sqrt{\frac{{}^i q}{{}^i \varepsilon_1}} - 2.13E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 142.80
 \end{aligned} \quad (5.28)$$

$$\begin{aligned}
 {}^{i+1}q = & 1.71E-10 \sqrt{\frac{{}^i \sigma'_3 \sigma_3 {}^i (\Delta \varepsilon_1)}{{}^i q}} + \frac{2846.61}{\sqrt{{}^i \sigma'_3}} - 181.76 \sqrt{\frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{{}^i \sigma'_3}} + 1.20E-5 \sigma_3^5 \sqrt{\frac{{}^i (\Delta \varepsilon_1)}{{}^i \sigma'_3}} - \\
 & \frac{715.73}{\sqrt{\sigma_3}} + \frac{1.0545}{\sqrt{{}^i \varepsilon_1}} + 95.18 \sqrt{{}^i (\Delta \varepsilon_1)} - 5.19 \sqrt{\frac{{}^i q \sigma_3}{{}^i \sigma'_3}} + 16.84 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} - 1.06 \sqrt{\frac{{}^i q}{{}^i \varepsilon_1}} - \\
 & 0.73 \frac{{}^i \varepsilon_1^4 {}^i (\Delta \varepsilon_1) \sqrt{{}^i q}}{{}^i \sigma'_3 \sigma_3} + 1.01 {}^i q - 2.12E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 141.39
 \end{aligned} \quad (5.29)$$

$$\begin{aligned}
 {}^{i+1}q = & 1.67E-10 {}^i\sigma'_3 \sqrt{\frac{{}^i(\Delta\varepsilon_1)}{{}^iq}} - 26.10 \frac{{}^i\varepsilon_1^4 {}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3 \sigma_3} + \frac{2975.37}{{}^i\sigma'_3} - 176.04 \sqrt{\frac{\sigma_3^i {}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3}} - \frac{728.09}{\sqrt{\sigma_3}} + \\
 & 1.22E-8 \sigma_3^5 \sqrt{\frac{{}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3}} + 95.31 \sqrt{{}^i(\Delta\varepsilon_1)} + \frac{1.07}{{}^i\varepsilon_1} + 16.51 {}^i(\Delta\varepsilon_1) \sqrt{\sigma_3} - 3.57E-9 {}^i\sigma_3^3 {}^i\varepsilon_1 - \\
 & 5.55 \sqrt{\frac{{}^iq \sigma_3}{{}^i\sigma'_3}} - 1.07 \sqrt{\frac{{}^iq}{{}^i\varepsilon_1}} + 1.02 {}^iq - 2.07E-8 {}^iq^3 {}^i(\Delta\varepsilon_1) + 140.66
 \end{aligned} \quad (5.30)$$

Also the equations listed in below ((5.31) to (5.45)) were obtained after completion of EPR analyses using the entire undrained triaxial test data.

$${}^{i+1}q = 965.36 \quad (5.31)$$

$${}^{i+1}q = 0.394 {}^iq^2 {}^i\sigma_3^{-1} + 154.89 \quad (5.32)$$

$${}^{i+1}q = 14.03 {}^i(\Delta\varepsilon_1) \sqrt{\sigma_3} + 0.97 {}^iq + 21.48 \quad (5.33)$$

$${}^{i+1}q = 15.92 {}^i(\Delta\varepsilon_1) - 8.27E-11 {}^iq^3 {}^i\sigma_3^i {}^i(\Delta\varepsilon_1) \sqrt{{}^i\varepsilon_1 \sigma_3^{-1}} + 1.02 {}^iq - 18.48 \quad (5.34)$$

$${}^{i+1}q = -729.15 \sqrt{\frac{{}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3}} + 18.58 {}^i(\Delta\varepsilon_1) \sqrt{\sigma_3} + 0.99 {}^iq - 5.67E-9 {}^iq^3 {}^i(\Delta\varepsilon_1) \sqrt{{}^i\varepsilon_1} + 32.84 \quad (5.35)$$

$$\begin{aligned}
 {}^{i+1}q = & 1.78E-5 \sigma_3^4 \sqrt{\frac{{}^i\varepsilon_1 {}^i(\Delta\varepsilon_1)}{{}^iq {}^i\sigma'_3}} - 108.08 \sqrt{\frac{\sigma_3^i {}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3}} + 0.99 {}^iq + 18.61 {}^i(\Delta\varepsilon_1) \sqrt{\sigma_3} - \\
 & 1.43E-8 {}^iq^2 {}^i\sigma_3^i {}^i(\Delta\varepsilon_1) \sqrt{{}^i\varepsilon_1} + 34.55
 \end{aligned} \quad (5.36)$$

$$\begin{aligned}
 {}^{i+1}q = & -55.73 \frac{\sigma_3^i {}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3} + 5.76E-8 \frac{\sigma_3^5 {}^i(\Delta\varepsilon_1)}{{}^i\sigma'_3} + 18.82 {}^i(\Delta\varepsilon_1) \sqrt{\sigma_3} - 27.90 \sqrt{\frac{{}^iq}{{}^i\sigma'}} + 1.01 {}^iq - \\
 & 1.78E-7 {}^iq {}^i(\Delta\varepsilon_1) \sqrt{\frac{{}^i\varepsilon_1}{{}^i\sigma'}} + 33.17
 \end{aligned} \quad (5.37)$$

$$\begin{aligned}
 {}^{i+1}q = & -0.17 \frac{{}^i\varepsilon_1^5 {}^i(\Delta\varepsilon_1) \sqrt{{}^i\sigma'_3}}{\sigma_3^2 \sqrt{{}^iq}} - 10.59 \frac{\sigma_3^i {}^i(\Delta\varepsilon_1)}{\sqrt{{}^i\sigma'_3}} + 1.50E-6 \frac{\sigma_3^4 {}^i(\Delta\varepsilon_1)}{\sqrt{{}^i\sigma'_3}} + 21.73 {}^i(\Delta\varepsilon_1) \sqrt{\sigma_3} - \\
 & 25.51 \sqrt{\frac{{}^iq}{{}^i\sigma'_3}} + 1.01 {}^iq - 2.24E-8 {}^iq^3 {}^i(\Delta\varepsilon_1) + 30.87
 \end{aligned} \quad (5.38)$$

$$\begin{aligned}
 {}^{i+1}q = & -7.92 \frac{{}^i \varepsilon_1^4 ({}^\Delta \varepsilon_1)}{\sqrt{{}^i q} {}^i \sigma'_3 \sigma_3} - 8.39 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 9.21E-9 \frac{\sigma_3^5 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 19.38 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} - \\
 & 28.36 \sqrt{\frac{{}^i q}{{}^i \sigma'_3}} + 0.80 \sqrt{{}^i \sigma'_3} {}^i \varepsilon_1 {}^i (\Delta \varepsilon_1) + 0.99 {}^i q - 2.38E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + 32.82
 \end{aligned} \tag{5.39}$$

$$\begin{aligned}
 {}^{i+1}q = & -404.05 \frac{{}^i \varepsilon_1^5 {}^i (\Delta \varepsilon_1)}{{}^i q {}^i \sigma'_3 \sqrt{\sigma_3}} + 7.52 {}^i (\Delta \varepsilon_1)^2 \sqrt{\frac{\sigma_3}{{}^i \sigma'_3}} - 11.46 \frac{\sigma_3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + 3.19E-4 \frac{\sigma_3^3 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} + \\
 & 17.29 {}^i (\Delta \varepsilon_1) \sqrt{\sigma_3} + 3.39 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - 25.75 \sqrt{\frac{{}^i q}{{}^i \sigma'_3}} + 0.99 {}^i q - 2.28E-8 {}^i q^3 {}^i (\Delta \varepsilon_1) + \\
 & 26.82
 \end{aligned} \tag{5.40}$$

$$\begin{aligned}
 {}^{i+1}q = & 44.77 \frac{{}^i (\Delta \varepsilon_1)^4}{\sigma_3^2} - 191.03 \sqrt{{}^i (\Delta \varepsilon_1)} + 17.54 \sqrt{\sigma_3} {}^i (\Delta \varepsilon_1) + 1.49 \sqrt{\frac{{}^i \sigma'_3}{{}^i (\Delta \varepsilon_1)}} - 9.01 \sqrt{{}^i \sigma'_3} + \\
 & 20.72 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - 3.18E-3 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1)^4 + 0.98 {}^i q - 1.98E-3 \frac{{}^i q^2 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} - \\
 & 2.16E-7 \frac{{}^i q {}^i \sigma'_3 {}^i \varepsilon_1^3 {}^i (\Delta \varepsilon_1)}{\sqrt{\sigma_3}} + 51.25
 \end{aligned} \tag{5.41}$$

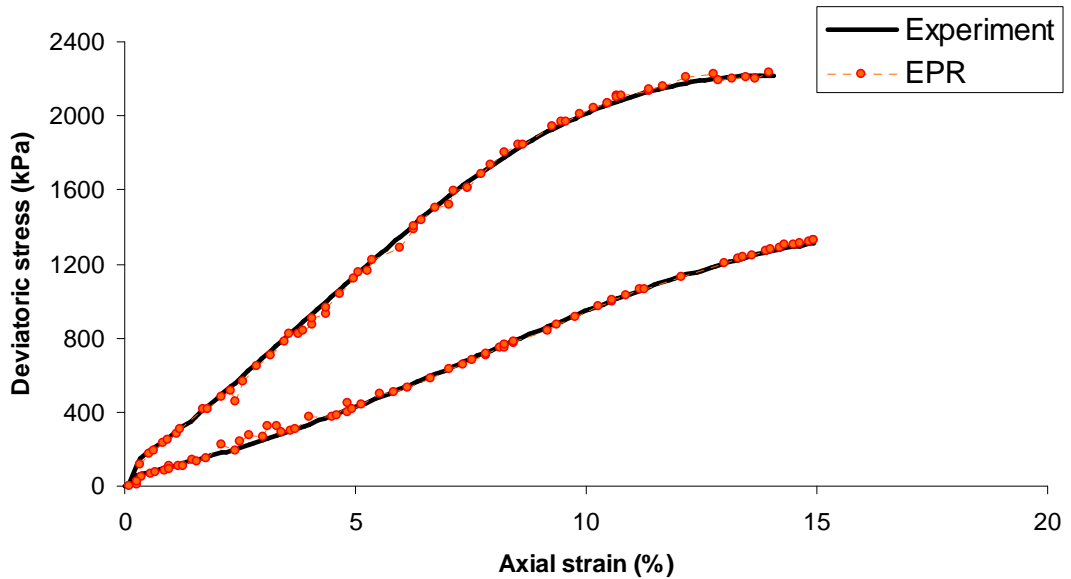
$$\begin{aligned}
 {}^{i+1}q = & 1.38 \frac{{}^i (\Delta \varepsilon_1)^4}{\sigma_3} - 204.95 \sqrt{{}^i (\Delta \varepsilon_1)} + 17.62 \sqrt{\sigma_3} {}^i (\Delta \varepsilon_1) - 10.97 \sqrt{{}^i \sigma'_3} + 20.78 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - \\
 & 3.55E-3 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1)^4 + 0.05 \frac{{}^i \sigma'_3}{\sqrt{{}^i (\Delta \varepsilon_1)}} + 1.01 {}^i q - 2.18E-7 \frac{{}^i q {}^i \sigma'_3 {}^i \varepsilon_1^3 {}^i (\Delta \varepsilon_1)}{\sqrt{\sigma_3}} - \\
 & 1.95E-3 \frac{{}^i q^2 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} - 3.64E-10 {}^i q^3 \sqrt{\sigma_3} + 88.89
 \end{aligned} \tag{5.42}$$

$$\begin{aligned}
 {}^{i+1}q = & 1.33 \frac{{}^i (\Delta \varepsilon_1)^4}{\sigma_3} - \frac{0.01}{{}^i (\Delta \varepsilon_1)^4 \sqrt{\sigma_3}} - 197.26 \sqrt{{}^i (\Delta \varepsilon_1)} + 17.52 \sqrt{\sigma_3} {}^i (\Delta \varepsilon_1) + 1.93 \sqrt{\frac{{}^i \sigma'_3}{{}^i (\Delta \varepsilon_1)}} - \\
 & 12.60 \sqrt{{}^i \sigma'_3} + 21.29 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - 1.98E-3 \frac{{}^i q^2 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} - 3.53E-3 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1)^4 - \\
 & 2.15E-7 \frac{{}^i q {}^i \sigma'_3 {}^i \varepsilon_1^3 {}^i (\Delta \varepsilon_1)}{\sqrt{\sigma_3}} - 1.24E-10 {}^i q^3 \sqrt{{}^i \sigma'_3} + 1.02 {}^i q + 73.92
 \end{aligned} \tag{5.43}$$

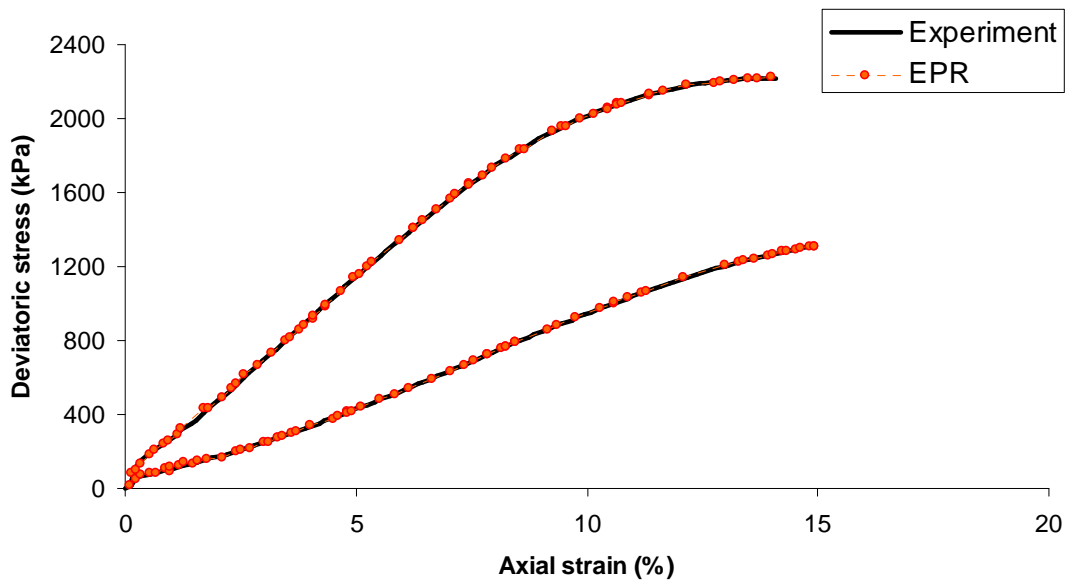
$$\begin{aligned}
 {}^{i+1}q = & \frac{2.12E-9}{\sqrt{{}^i q}} + \frac{1.34 {}^i (\Delta \varepsilon_1)^4}{\sigma_3} - \frac{0.01}{\sqrt{\sigma_3} {}^i (\Delta \varepsilon_1)^4} - 197.33 \sqrt{{}^i (\Delta \varepsilon_1)} + 17.52 \sqrt{\sigma_3} {}^i (\Delta \varepsilon_1) + \\
 & \frac{1.92 \sqrt{{}^i \sigma'_3}}{\sqrt{{}^i (\Delta \varepsilon_1)}} - 12.47 \sqrt{{}^i \sigma'_3} + 21.30 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1) - 3.54E-3 \sqrt{{}^i \sigma'_3} {}^i (\Delta \varepsilon_1)^4 + 1.02 {}^i q - \\
 & 2.15E-7 \frac{{}^i q {}^i \sigma'_3 {}^i \varepsilon_1^3 {}^i (\Delta \varepsilon_1)}{\sqrt{\sigma_3}} - 1.98E-3 \frac{{}^i q^2 {}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma'_3}} - 1.26E-10 {}^i q^3 \sqrt{{}^i \sigma'_3} + 72.02
 \end{aligned} \tag{5.44}$$

$$\begin{aligned}
 {}^{i+1}q = & -0.86 \frac{{}^i q {}^i (\Delta \varepsilon_1)^2}{{}^i \sigma_3' \sqrt{{}^i \varepsilon_1}} - 2.16E-8 {}^i q^3 \sqrt{{}^i (\Delta \varepsilon_1)} + 8.29E-8 \frac{{}^i q^3 \sqrt{{}^i \sigma_3'} {}^i (\Delta \varepsilon_1)^3}{{}^i \sigma_3' {}^i \varepsilon_1^2} + 1.04 {}^i q - \\
 & 2.94E-3 {}^i \varepsilon_1^3 {}^i (\Delta \varepsilon_1)^2 - 2.84E-5 \frac{{}^i q^2 \sqrt{{}^i \sigma_3'} {}^i (\Delta \varepsilon_1)^2}{{}^i \sigma_3' {}^i \varepsilon_1} - 2.02E-4 \sqrt{{}^i \sigma_3'} {}^i \varepsilon_1^4 {}^i (\Delta \varepsilon_1) - \\
 & 3.75E-10 \frac{\sqrt{{}^i \sigma_3'} {}^i (\Delta \varepsilon_1)^5}{{}^i (\sigma_3')^2 \sqrt{{}^i \varepsilon_1}} + 1.46 {}^i \sigma_3' {}^i (\Delta \varepsilon_1) - 1147.34 \frac{{}^i q \sqrt{{}^i \varepsilon_1}}{{}^i \sigma_3'^4 {}^i (\Delta \varepsilon_1)} + 9076.56 \frac{{}^i (\Delta \varepsilon_1)^2}{{}^i \sigma_3'^2} - \\
 & 2.75E-3 \sqrt{{}^i \sigma_3'} {}^i (\Delta \varepsilon_1)^4 + 193.64 \frac{{}^i q^2 \sqrt{{}^i \sigma_3'} {}^i (\Delta \varepsilon_1) {}^i \varepsilon_1^3}{{}^i (\sigma_3')^4} + 85.44 \frac{{}^i (\Delta \varepsilon_1)}{\sqrt{{}^i \sigma_3'}} - 9.78
 \end{aligned} \tag{5.45}$$

For the first case, where just the part of data related to strain hardening behaviour was considered, the EPR simulations associated with the experimental results used in the training phase are shown in Figure 5.6. Figure 5.6a shows the stress-strain curves predicted by Equation (5.19) (as marker points) against those expected and used as training data (as solid lines). Figure 5.6b shows the same results using Equation (5.30) (at the final evolution). The accuracy of results even at early evolutions reflects the power of EPR in capturing the material behaviour.



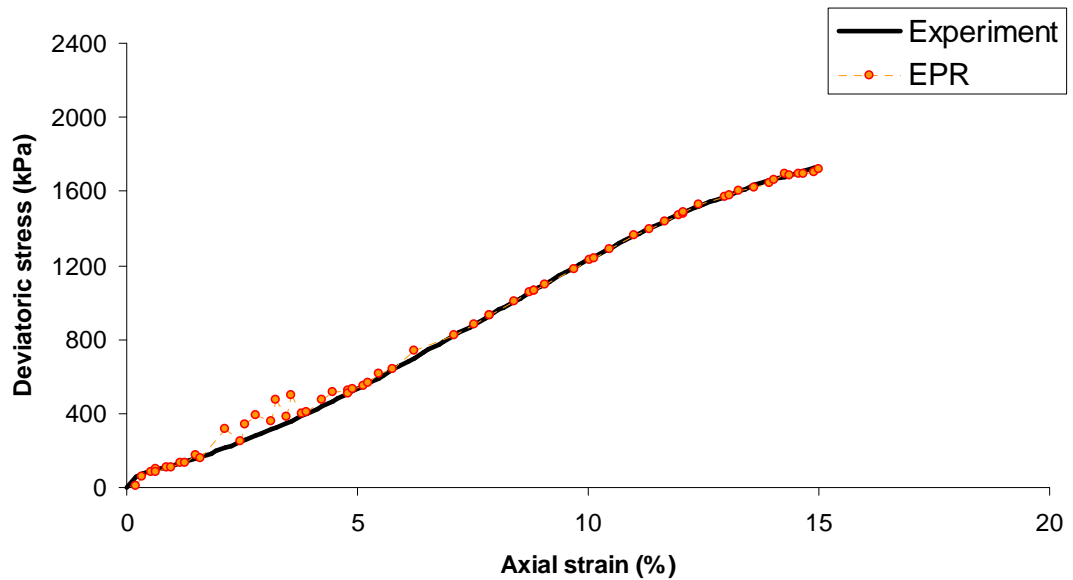
(a)



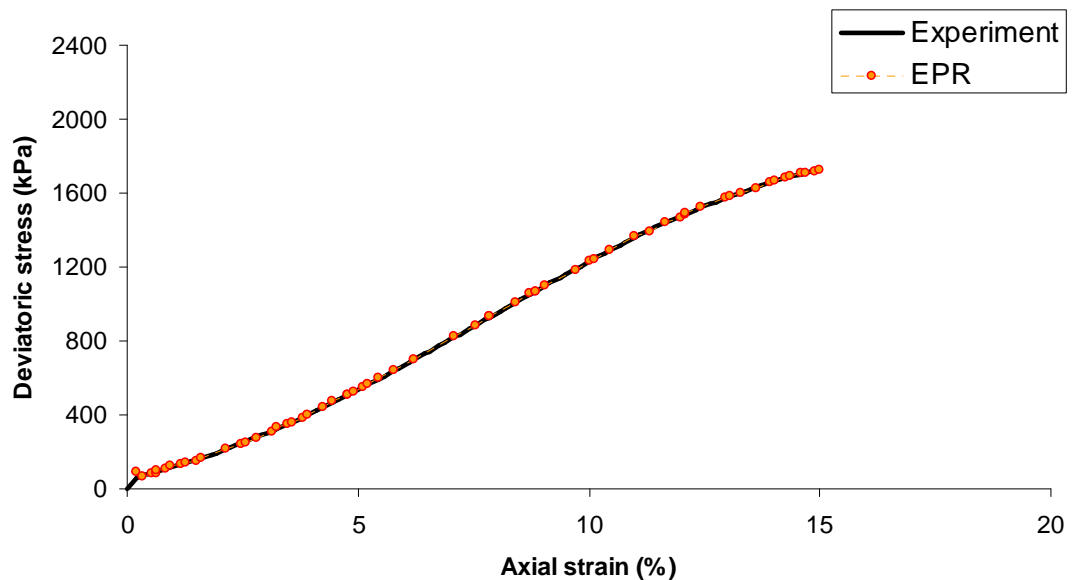
(b)

**Figure 5.6** Results of training the EPR models using (a) Equation (5.19) and (b) Equation (5.30).

To demonstrate the predictive ability of the trained EPR models, the same equations were used to calculate stress-strain curves for the remaining test which did not form a part of the training set. It can be seen from Figure 5.7 that the trained models can predict the stress-strain behaviour of the soil at a different consolidation pressure with a very high precision.



(a)

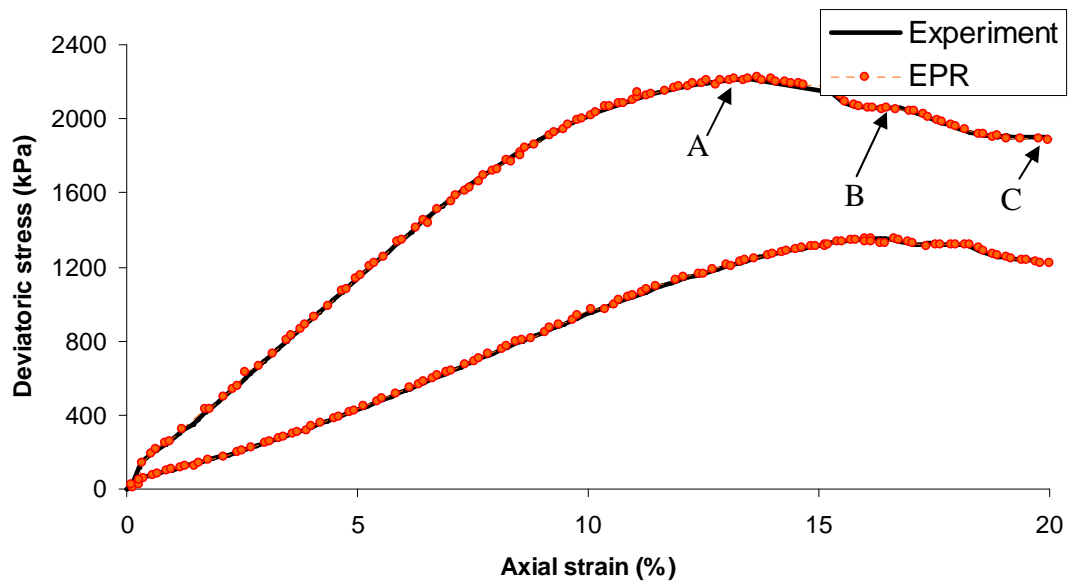


(b)

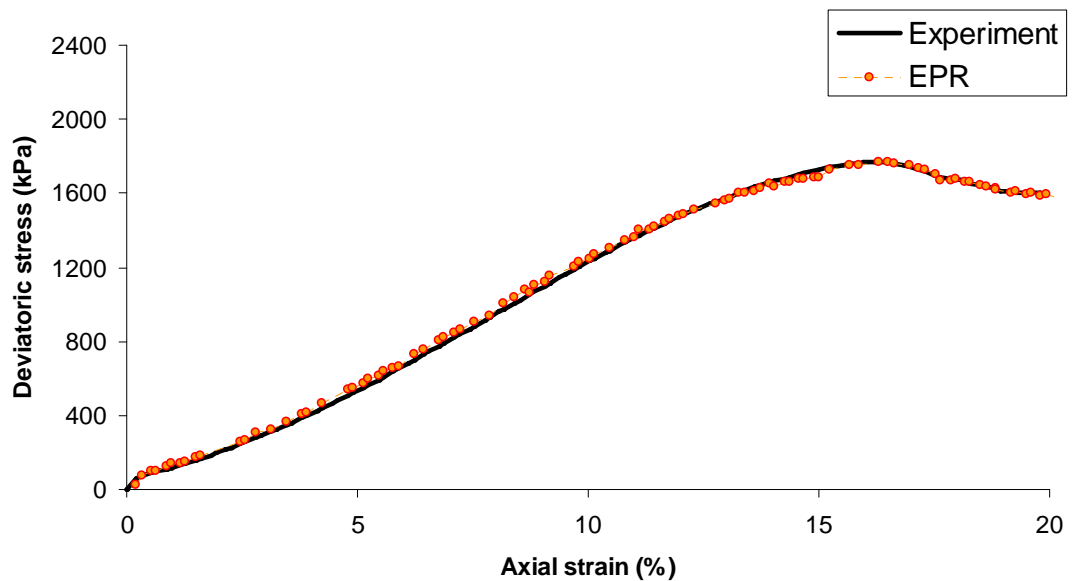
**Figure 5.7** Results of testing the EPR models using (a) Equation (5.19) and (b) Equation (5.30).

For the second case, where the data related to entire range of strain was used to train the EPR based constitutive models, Equation (5.44) was selected to show the results of

EPR predictions over training and testing data sets (see Figures 5.8 and 5.9).



**Figure 5.8** Results of training the EPR models using Equation (5.44).



**Figure 5.9** Results of testing the EPR models using Equation (5.44).

The purpose of the second trial was to assess the quality of EPR predictive capabilities over the following regions

- i) The peak strength region (e.g., point A on stress-strain curve relating to confining pressure of 160 kPa in Figure 5.8).
- ii) The post peak region (e.g., AB zone on stress-strain curve relating to

confining pressure of 160 kPa in Figure 5.8).

- iii) The residual strength region (e.g., BC zone on stress-strain curve relating to confining pressure of 160 kPa in Figure 5.8).

From the results shown in Figures 5.8 and 5.9 it can be seen that the trained EPR constitutive model is capable of modelling the transition from elastic to plastic and strain softening behaviour with excellent precision.

### **5.3 Application of EPR based material models for triaxial test simulation**

Triaxial tests, which are the most commonly used experiments to study the behaviour of geomaterials, are generally known to be relatively expensive and time-consuming to operate. Also, triaxial tests on one particular soil are usually repeated at different initial conditions (e.g., different confining pressures or relative densities) in order to obtain comprehensive information on the material behaviour. Here the idea is to develop an alternative approach (i.e., using a triaxial test simulator) to complement real experiments and obtain additional information on the soil behaviour without the need to perform numerous real tests. This can be very useful as it saves a lot of effort needed to perform the actual experiments.

In the previous sections, it was shown that evolutionary polynomial regression method is effectively capable of finding the constitutive behaviour of soils if it is properly trained with relevant data. In this section it will be shown that an EPR based model which has been trained to capture soil behaviour based on a limited number of triaxial test results can be used as a stand alone tool to generate additional virtual triaxial data.

#### **5.3.1 Drained triaxial test simulator**

For drained condition, the final mathematical model (equation) which was developed in section 5.2.1.3, is used to show the EPR capability in generating virtual triaxial data. Basically the input parameters to the model are those which were used in model

development. In section 5.2.1.3 it was shown that EPR accurately modelled the influence of confining pressure on the mechanical behaviour of soil. Therefore for a given confining pressure the purpose of the EPR model is to predict the corresponding stress-strain curve. From plasticity theory of soils, it is well known that the current state of stress and strain have an important influence on the next state of stress and strain increments. It is thus, important to use the so called “feed-forward” concept for simulating stress-strain curves with an EPR based model. The feed-forward algorithm also ensures that the simulation phase of the soil model will be similar to the expected behaviour of soil in a real triaxial test. The EPR based simulator has to build the entire stress-strain curves incrementally, this can be done based on the predictions corresponding to the loading conditions of previous step. At the initial phase of the simulation, the preliminary values of the current state units are fed into the EPR model. Through the feed-forward process, using the EPR model, the output value (next stress state) is produced; this makes the first pattern for the virtually generated stress-strain path. The output value is then copied back to the current state units for the next feed-forward sweep. This concept is described in steps as follows

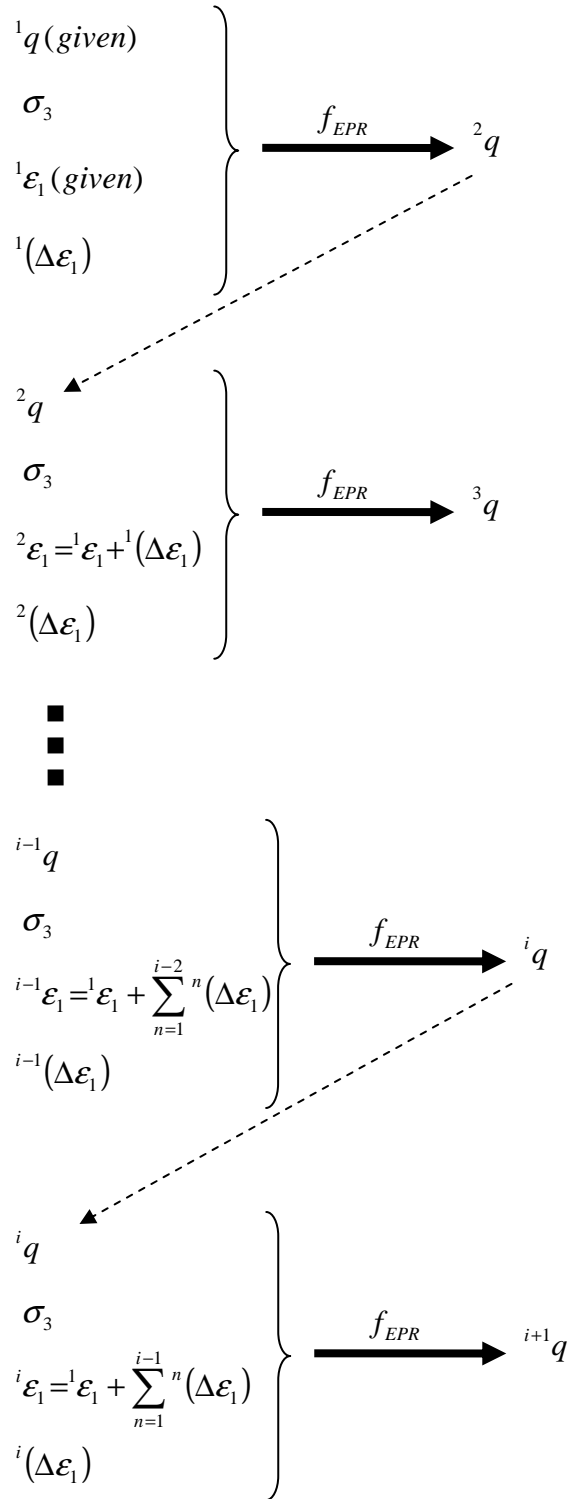
- i) A particular value of confining pressure ( $\sigma_3$ ) is considered.
- ii) A set of strain increment values ( $\Delta\varepsilon_1$ ) is considered. These values are not constant i.e., the value of strain increment is chosen to increase at a constant magnitude (e.g., 0.5%). This ensures that the model is accurately responsive at different strain increments.
- iii) The value of axial strain is given for the initial increment ( ${}^1\varepsilon_1 = E - 07 \approx 0$ ), after that at each stage the value of axial strain is equal to the cumulative value of all preceding values of strain increment plus the initial value of axial strain

$${}^i\varepsilon_1 = {}^1\varepsilon_1 + \sum_{n=1}^{i-1} \Delta\varepsilon_1$$

- iv) The value of deviator stress is given for the initial increment (e.g.,  ${}^1q = E - 07 \approx 0$ ), for this given value, using the EPR based material model and other parameters' values, the value of deviator stress for the next step ( ${}^2q$ ) is calculated. This new value of deviator stress will itself be used for the following increment's calculation.

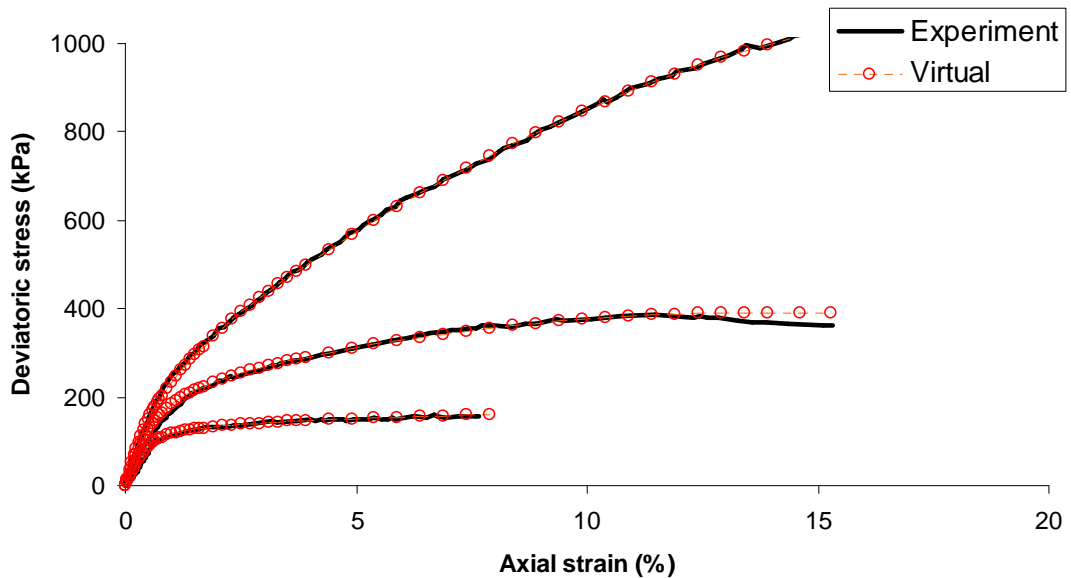
- v) Steps 3 and 4 will be repeated until the stress-strain path for the entire domain of axial strain is obtained.

The procedure explained above is schematically shown in Figure 5.10.

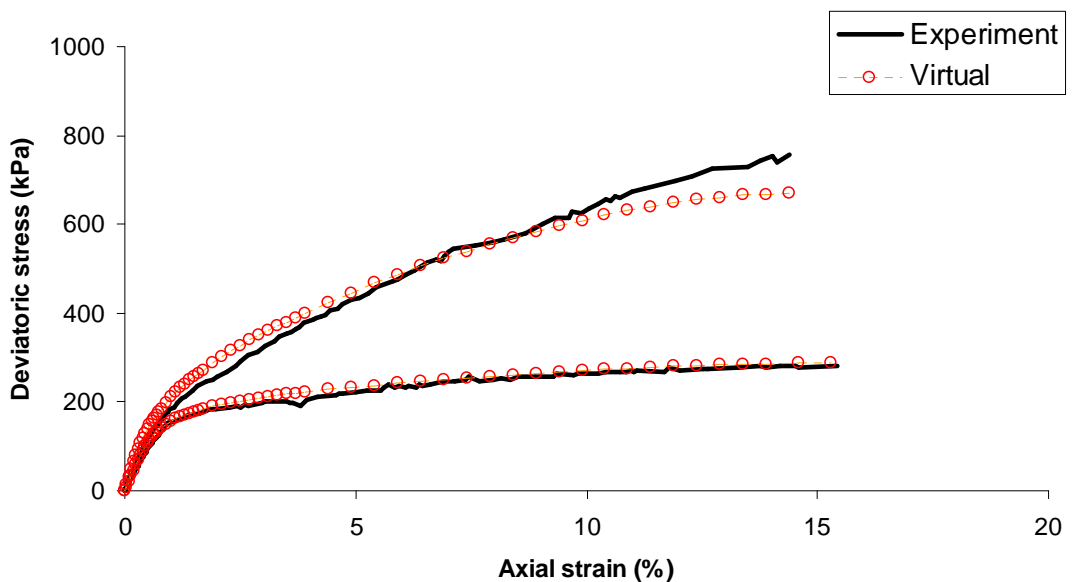


**Figure 5.10** The schematic sketch of feed-forward algorithm for drained triaxial test simulator.

The virtual drained triaxial test results obtained in this way are shown in Figures 5.11 and 5.12. As it can be seen in these figures, excellent agreement between the real data and the virtually generated data, using EPR based material model and feed forward concept, emphasises the great capability of the strategy in simulating triaxial tests.



**Figure 5.11** The comparison between the results of drained triaxial test simulator and the tests used for training the EPR model.



**Figure 5.12** The comparison between the results of drained triaxial test simulator and two real tests which have not been part of EPR model construction.

### 5.3.2 Undrained triaxial test simulator

Another attempt is made to develop a strategy for simulating undrained triaxial tests using EPR based material models. As it was discussed in the previous chapter, in all works performed for neural network based material modelling in undrained condition (e.g. Ellis et al. 1995, Amorosi et al. 1996, Zhu et al. 1998, Penumadu and Zhao 1999), two output parameters were considered for the networks (i.e., with the given values of current state units, the next state of deviator stress and next state of pore water pressure were sought). Indeed one of the advantages of neural networks is that they can have more than one output parameter. However the mathematical models constructed by EPR can just have one output parameter. Because the stand alone simulator needs to be able to model both the changes of deviator stress and pore water pressure, and it is in contrast with having just one output parameter, the decision was made to develop two different models in order to obviate this problem. The model which, with a given set of stress and strain units (including the effect of pore water pressure), is able to predict the next state of deviator stress has already been developed and explained in section 5.2.2. It is therefore necessary to develop a separate model to predict the value of pore water pressure, at each stage of the test, relevant to the current values of state units. Here this later model is named “UT-EPR-A” and the former model, described in section 5.2.2, is named “UT-EPR-B”.

The input-output parameters for model “UT-EPR-A” are therefore considered as follows

Input:  $q, \sigma_3, \varepsilon_1$

Output:  $\sigma'_3$

The data division and initialisation criteria for EPR procedure was kept identical as those described in sections 5.2.1 and 5.2.2. It should also be noted that just the part of undrained data related to strain hardening behaviour of triaxial samples (strains up to 15 percent) was considered for this evaluation. The equations listed below have been obtained after completion of EPR analysis for model “UT-EPR-A”.

$${}^i\sigma'_3 = 345.99 \quad (5.46)$$

$${}^i\sigma'_3 = 0.39 {}^i q + 18.77 \quad (5.47)$$

$${}^i\sigma'_3 = -4.04\sqrt{{}^i q} + 0.47 {}^i q + 60.28 \quad (5.48)$$

$${}^i\sigma'_3 = 0.87 \sigma_3 - 0.59\sqrt{{}^i q \sigma_3} + 0.50 {}^i q - 4.03 \quad (5.49)$$

$${}^i\sigma'_3 = 0.97 \sigma_3 - 0.85\sqrt{{}^i q \sigma_3} - 0.10 \frac{{}^i q \sqrt{{}^i \varepsilon_1}}{\sqrt{\sigma_3}} + 0.59 {}^i q + 5.09 \quad (5.50)$$

$${}^i\sigma'_3 = 3.02 \frac{{}^i \varepsilon_1^3}{\sqrt{{}^i q}} + 1.11 \sigma_3 - 1.15\sqrt{{}^i q \sigma_3} - 0.15 \frac{{}^i q {}^i \varepsilon_1}{\sqrt{\sigma_3}} + 0.72 {}^i q + 3.83 \quad (5.51)$$

$${}^i\sigma'_3 = 77.65 \frac{{}^i \varepsilon_1^3}{{}^i q} + 1.14 \sigma_3 - 1.26\sqrt{{}^i q \sigma_3} - 0.51 \frac{{}^i q \sqrt{{}^i \varepsilon_1}}{\sqrt{\sigma_3}} + 0.82 {}^i q - 2.75E - 4 \frac{{}^i q^2}{\sqrt{\sigma_3}} + 5.69 \quad (5.52)$$

$${}^i\sigma'_3 = 2.06 \frac{{}^i \varepsilon_1^3}{\sqrt{{}^i q}} + 2.77 {}^i \varepsilon_1 \sqrt{\sigma_3} + 5.87E - 3\sigma_3^2 - 6.11\sqrt{{}^i q} - 1.12 {}^i \varepsilon_1 \sqrt{{}^i q} - 0.07 \sigma_3 \sqrt{{}^i q} + 0.74 {}^i q + 47.63 \quad (5.53)$$

$${}^i\sigma'_3 = 2.30 \frac{{}^i \varepsilon_1^3}{\sqrt{{}^i q}} + 3.09 {}^i \varepsilon_1 \sqrt{\sigma_3} + 5.95E - 3\sigma_3^2 - 6.15\sqrt{{}^i q} - 1.25 {}^i \varepsilon_1 \sqrt{{}^i q} - 0.07 \sigma_3 \sqrt{{}^i q} + 0.76 {}^i q + 9.46E - 6 \frac{{}^i q^4}{\sigma_3^2 {}^i \varepsilon_1^4} + 44.29 \quad (5.54)$$

$${}^i\sigma'_3 = 2.58 \frac{{}^i \varepsilon_1^3}{\sqrt{{}^i q}} + 2.03 {}^i \varepsilon_1 \sqrt{\sigma_3} + 6.0E - 3\sigma_3^2 + 1.43E - 7\sigma_3^4 \sqrt{{}^i \varepsilon_1} - 7.46\sqrt{{}^i q} - 1.29 {}^i \varepsilon_1 \sqrt{{}^i q} - 6.35E - 4\sigma_3^2 \sqrt{{}^i q} + 0.82 {}^i q + 1.39E - 7 \frac{{}^i q^4}{\sigma_3 {}^i \varepsilon_1^4} + 39.58 \quad (5.55)$$

$${}^i\sigma'_3 = 77.12 \frac{{}^i \varepsilon_1^2}{\sqrt{{}^i q \sigma_3}} + 0.06 {}^i \varepsilon_1^3 + 1.33 {}^i \varepsilon_1 \sqrt{\sigma_3} + 6.11E - 3\sigma_3^2 + 3.28E - 5\sigma_3^3 \sqrt{{}^i \varepsilon_1} - 7.24\sqrt{{}^i q} - 1.35 {}^i \varepsilon_1 \sqrt{{}^i q} - 7.53E - 4\sigma_3^2 \sqrt{{}^i q} + 1.68E - 7 \frac{{}^i q^4}{\sigma_3 {}^i \varepsilon_1^4} + 0.87 {}^i q + 35.64 \quad (5.56)$$

$$\begin{aligned}
 {}^i\sigma'_3 = & -\frac{1.52E-7}{\sigma_3 \sqrt{{}^i\varepsilon_1}} - 118.95 \frac{{}^i\varepsilon_1}{\sqrt{\sigma_3}} + 44.64 {}^i\varepsilon_1 + 1.98 {}^i\varepsilon_1^2 + 5.0E-3 \sigma_3^2 + 1.33E-7 \sigma_3^4 \sqrt{{}^i\varepsilon_1} - \\
 & 11.72 \sqrt{{}^iq} - 1.95 {}^i\varepsilon_1 \sqrt{{}^iq} - 5.07E-4 \sigma_3^2 \sqrt{{}^iq} + 0.88 {}^iq + 1.16E-8 \frac{{}^iq^4}{{}^i\varepsilon_1^4 \sqrt{\sigma_3}} + 69.39
 \end{aligned} \quad (5.57)$$

$$\begin{aligned}
 {}^i\sigma'_3 = & -4890.48 \frac{{}^i\varepsilon_1}{\sigma_3^2} - \frac{1.31E-7}{\sigma_3 \sqrt{{}^i\varepsilon_1}} - 17.10 \sqrt{{}^i\varepsilon_1} + 30.31 {}^i\varepsilon_1 + 0.05 {}^i\varepsilon_1^3 + 5.11E-3 \sigma_3^2 + 2.91E-5 \sigma_3^3 \sqrt{{}^i\varepsilon_1} - \\
 & 9.80 \sqrt{{}^iq} - 1.40 {}^i\varepsilon_1 \sqrt{{}^iq} - 5.92 \sigma_3^2 \sqrt{{}^iq} + 0.84 {}^iq + 1.23E-8 \frac{{}^iq^4}{{}^i\varepsilon_1^4 \sqrt{\sigma_3}} + 64.47
 \end{aligned} \quad (5.58)$$

$$\begin{aligned}
 {}^i\sigma'_3 = & -0.05 \frac{\sigma_3 {}^i\varepsilon_1^4}{{}^iq} - 233.23 \frac{{}^i\varepsilon_1^2}{\sqrt{{}^iq} \sigma_3} - \frac{1.37E-7}{\sigma_3 \sqrt{{}^i\varepsilon_1}} - 43.31 \sqrt{{}^i\varepsilon_1} + 45.35 {}^i\varepsilon_1 - 1.40 {}^i\varepsilon_1 \sqrt{{}^iq} + \\
 & 0.10 {}^i\varepsilon_1^3 - 8.18 \sqrt{{}^iq} + 5.06E-3 \sigma_3^2 + 3.31E-5 \sigma_3^2 \sqrt{{}^i\varepsilon_1} - 6.29E-4 \sigma_3^2 \sqrt{{}^iq} + \\
 & 0.78 {}^iq + 1.25E-8 \frac{{}^iq^4}{{}^i\varepsilon_1^4 \sqrt{\sigma_3}} + 65.95
 \end{aligned} \quad (5.59)$$

$$\begin{aligned}
 {}^i\sigma'_3 = & -0.05 \frac{\sigma_3 {}^i\varepsilon_1^4}{{}^iq} + \frac{33.72}{\sqrt{{}^iq} \sigma_3} - 247.86 \frac{{}^i\varepsilon_1^2}{\sqrt{{}^iq} \sigma_3} - \frac{33.72}{\sqrt{\sigma_3} {}^i\varepsilon_1} - 60.85 \sqrt{{}^i\varepsilon_1} + 54.47 {}^i\varepsilon_1 + \\
 & 0.10 {}^i\varepsilon_1^3 + 4.90E-3 \sigma_3^2 + 3.24E-5 \sigma_3^3 \sqrt{{}^i\varepsilon_1} - 8.89 \sqrt{{}^iq} - 1.52 {}^i\varepsilon_1 \sqrt{{}^iq} - \\
 & 6.10E-4 \sigma_3^2 \sqrt{{}^iq} + 0.79 {}^iq + 1.19E-8 \frac{{}^iq^4}{{}^i\varepsilon_1^4 \sqrt{\sigma_3}} + 86.31
 \end{aligned} \quad (5.60)$$

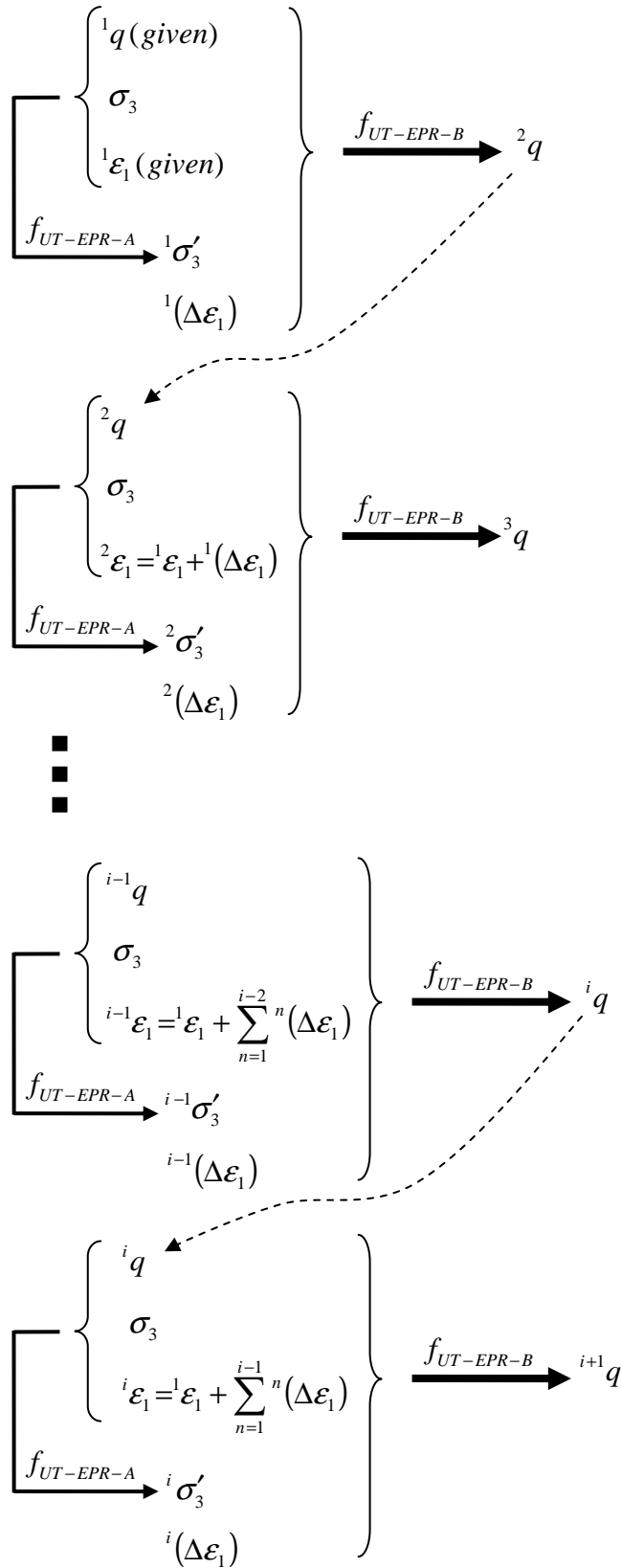
The “feed-forward” concept as described in section 5.3.1 was employed and extended to be used for simulating undrained condition. The procedure is explained in details as follows

- i) A particular value of consolidation pressure ( $\sigma_3$ ) is considered.
- ii) A set of strain increment values ( $\Delta\varepsilon_1$ ) is considered. These values are not constant but they increase at a constant magnitude (e.g., 0.5%). This ensures that the model is accurately responsive at different strain increments.
- iii) The value of axial strain is given for the initial increment ( ${}^1\varepsilon_1 = E-07 \approx 0$ ), after that at each stage the value of axial strain is equal to the cumulative value of all preceding values of strain increment plus the initial value of axial strain

$${}^i\varepsilon_1 = {}^1\varepsilon_1 + \sum_{n=1}^{i-1} n (\Delta\varepsilon_1)$$

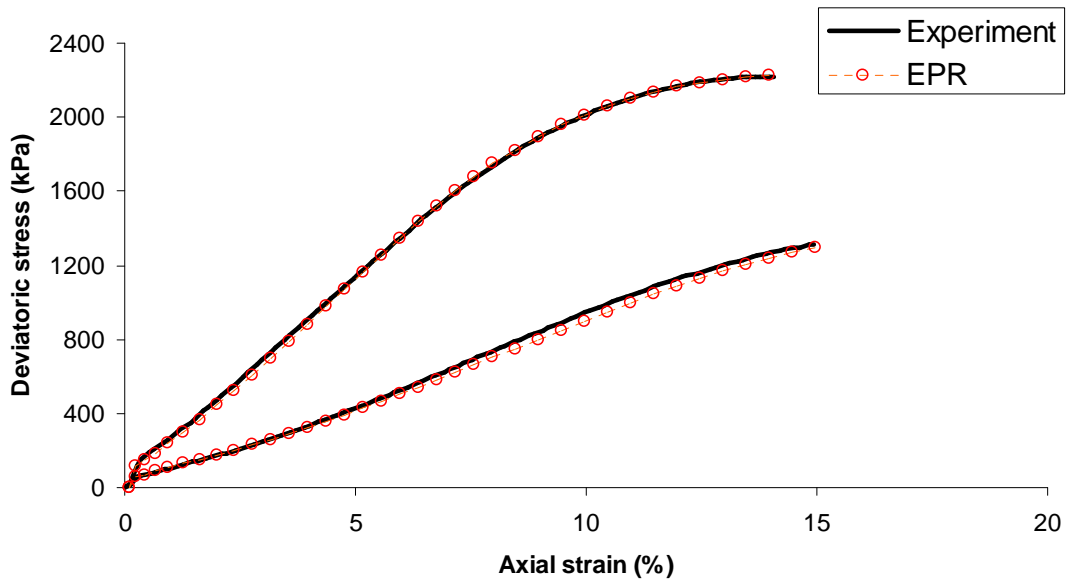
- iv) The value of deviator stress is given for the initial increment (e.g.,  ${}^1q = E - 07 \approx 0$ ), however after the first step, the calculated output of each step will be used as the input value for next step. Using the model UT-EPR-A and other parameters' values (i.e.,  $\sigma_3$  and  ${}^i\varepsilon_1$ ), the value of effective confining pressure ( ${}^i\sigma'_3$ ) for the current step is calculated.
- v) Using all the existing values of stresses and strains, the model UT-EPR-B is utilised to calculate the next value of deviator stress ( ${}^{i+1}q$ ) corresponding to the current imposed strain disturbance,  ${}^i(\Delta\varepsilon_1)$ . This new value of deviator stress will itself be used for next increment's calculations.
- vi) The steps 'iii' to 'v' will be repeated until the stress-strain path for the entire domain of axial strain is obtained.

Figure 5.13 gives an illustrative insight of the feed-forward concept in generating undrained triaxial data.

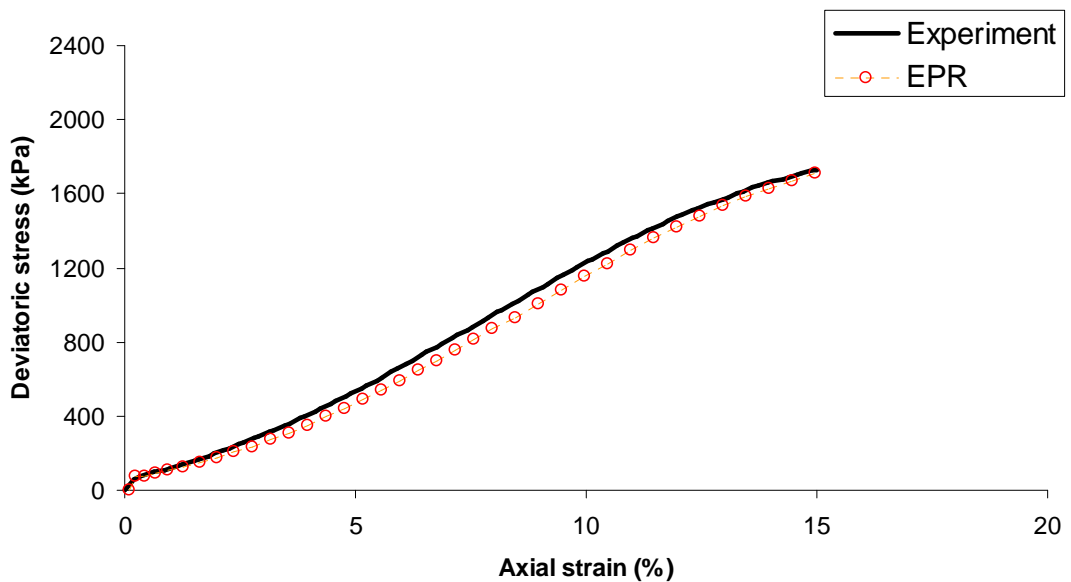


**Figure 5.13** The schematic sketch of feed-forward algorithm for undrained triaxial test simulator.

The virtual undrained triaxial test results obtained in this way are shown in Figures 5.14 and 5.15. Again, it can be seen that the feed-forward concept is of great advantage in simulating triaxial test results in undrained condition.



**Figure 5.14** The comparison between the results of undrained triaxial test simulator and the tests used for training the EPR model.



**Figure 5.15** The comparison between the results of undrained triaxial test simulator and the real test which has not been part of EPR model construction.

## 5.4 Discussion

Compared with mathematical constitutive modelling, artificial neural network based constitutive models are known to have a significant disadvantage in that they are unable to provide a clear understanding of the mechanisms governing the stress-strain characteristics. This limitation has been tackled with EPR based material modelling approach as this method is capable of providing a transparent correlation for the general stress-strain relationship.

EPR based constitutive modelling is a very useful alternative when mathematical constitutive modelling becomes very complex or the models contain parameters that are difficult or expensive to measure in practice. Like NN based models, EPR based constitutive models do not rely on an a priori assumption of a physical model, but instead, they rely directly upon observed or measured behaviour. As such when the behaviour of representative elemental volumes of much larger physical extent is required to be modelled, data obtained from field observations or in situ tests can be used to train an appropriate EPR model.

Another major merit of EPR based models is the precision of predictions achieved from this method. When the constitutive behaviour of a particular geomaterial is known to be simple but better accuracy is required than what is available with a corresponding mathematical approach, the EPR offers a fundamental alternative.

Similar to NN based models, an additional advantage of EPR based material modelling is in the possibility of updating and improving the obtained material model by retraining the model when additional experimental data becomes available.

Like other artificial intelligence methods, one of the most important aspects of EPR models is the training phase. The dataset used for training must be as large as needed for the EPR to capture (i.e., learn and model) the characteristics of the soil behaviour. However it was observed that the optimum amount of data used for EPR training is not necessarily the largest sum of data. If the training data is properly organised in the way that it covers and represents different features of the soil behaviour, it is desirable to limit its size to control the complexity of trained EPR models and the computational effort needed for their training.

It was shown that by using an appropriate strategy, it is possible to employ the developed EPRCM as a stand alone partial triaxial data generator. This is particularly useful when the results from material tests are only available for a limited number of stress-strain curves.

It should be mentioned that the models described in this chapter have been trained and validated using limited number of data sets developed for two different conditions (drained and undrained) and for specific types of soil, with an aim to demonstrate the potential of EPR technique. The generic application of EPR, however, will require further testing.

# CHAPTER 6

## INTELLIGENT FINITE ELEMENT METHOD AND NUMERICAL EXAMPLES

### 6.1 Introduction

Simulation techniques, and in particular the finite element method, have been used successfully to predict the response of systems across a whole range of industries including aerospace and automotive, biomedical, chemical processes, geotechnical engineering and many others. In these numerical analyses, the behaviour of the actual material is approximated with that of an idealised material that deforms in accordance with some constitutive relationships. Therefore, the choice of an appropriate constitutive model that adequately describes the behaviour of the material plays an important role in the accuracy and reliability of the numerical predictions. During the past few decades several constitutive models have been developed for various materials. Most of these models involve determination of material parameters, many of which have no physical meaning. In spite of considerable complexities of constitutive theories, due to the erratic and complex nature of some materials such as soil, rock, composites, etc, none of the existing constitutive models can completely describe the real behaviour of these materials under various stress paths and loading conditions.

The modelling of constitutive behaviour of materials has progressed by refining the plethora of empirically based constitutive laws. Equations describing constitutive behaviour have been developed from a need for an appropriate mathematical form to fit an experimental observation rather than by improving the knowledge of physical mechanisms involved. This situation is characteristic of the innate complexity introduced when utilising natural materials (such as soil, rock, etc) in engineering applications. Important first-order effects such as the anisotropy or heterogeneity are routinely ignored in analyses conducted by engineers due to their complexity.

Due to significant advances in computational power and the development of efficient solvers, the degree of sophistication and realism of models has increased dramatically. Available packages offer new and improved interfaces, better visualisation of results, more options for automation of the search for design solutions, etc, however the constitutive laws used in the analyses remain largely unchanged from those procedures used decades ago. Application of powerful numerical techniques in new high growth fields is being held back by the lack of appropriate material models.

In this chapter the developed constitutive material models based on evolutionary polynomial regression, as described in chapter 5, is integrated in finite element analysis. It should be emphasised that although the work proposed here will mostly focus on soils, the methodology proposed is generic and could well be applied to other materials that have constitutive models that are difficult to accurately define in numerical codes used in other engineering disciplines.

## **6.2 Background**

In conventional constitutive modelling of materials, an appropriate mathematical model is initially selected and the parameters of this model (material parameters) are then identified from appropriate physical tests on representative samples to capture the material behaviour. When these constitutive models are used in finite element analysis, the accuracy with which the selected material model represents the various aspects of the actual material behaviour affects the accuracy of the finite element prediction. As discussed in chapter 2, in the past few years, the use of artificial neural networks has been introduced as an alternative approach to constitutive modelling of materials

(Ghaboussi et al 1991, Ellis et al. 1992, Millar and Clarici 1994). The role of NNs in constitutive modelling was also studied by a number of other researchers (Ellis et al. 1995, Penumadu and Chameau 1997, Zhu et al. 1998, Ghaboussi et al. 1998, Sidarta and Ghaboussi 1998, Ghaboussi and Sidarta 1998, Penumadu and Zhao 1999 etc.). These studies indicated that neural network based constitutive models can capture nonlinear material behaviour with high accuracy. The potential advantage of using NN based constitutive model in a finite element framework when they are trained directly from experimental (or in-situ) data is that in this case, an unknown conventional analytical constitutive description can be directly replaced with a suitably trained neural network. The source of knowledge for NN is not a symbolic formula but a set of experimental data in this case. However, as discussed in chapter 2, NNCMs have shown some drawbacks as well. For example Millar and Calderbank (1995) used the data produced by triaxial test simulation in FLAC using strain softening model available within this software for training their NNCM. However when the optimum NN structure obtained for constitutive relationship was used as a user defined constitutive model back in FLAC, although the accuracy of the NNCM over the training data was good but its prediction ability was so poor and the actual behaviour of the NNCM was far from the desired behaviour when it was used in place of the standard strain softening constitutive model within FLAC. The problem relating to the rather inferior generalising capability of NNCMs was also reported by other researchers (e.g., Ellis et al. 1995, Penumadu and Chameau 1997). Generally one of the main disadvantages of the NNCM is that the optimum structure of the NN (such as number of inputs, hidden layers, transfer functions, etc.) must be identified a priori which is usually obtained using a time consuming trial and error procedure. Another main drawback of the NNCM approach is the large complexity of the network structure, as it represents the knowledge in terms of a weight matrix together with biases that are not accessible to user. As a black box class of models, NN models give no information on the way the inputs affect the output. The lack of interpretability of NN models has inhibited them from achieving their full potential in real world problems (Lu et al. 2001).

Although one of the main purposes of material modelling is to carry out structural analysis, to date not many researchers considered the integration of the neural network based constitutive models (NNCMs) in structural analysis techniques such as finite element method (Shin 2001). The main reason for this appears to be the fact that there are considerable difficulties in incorporating a general NNCM in finite element codes

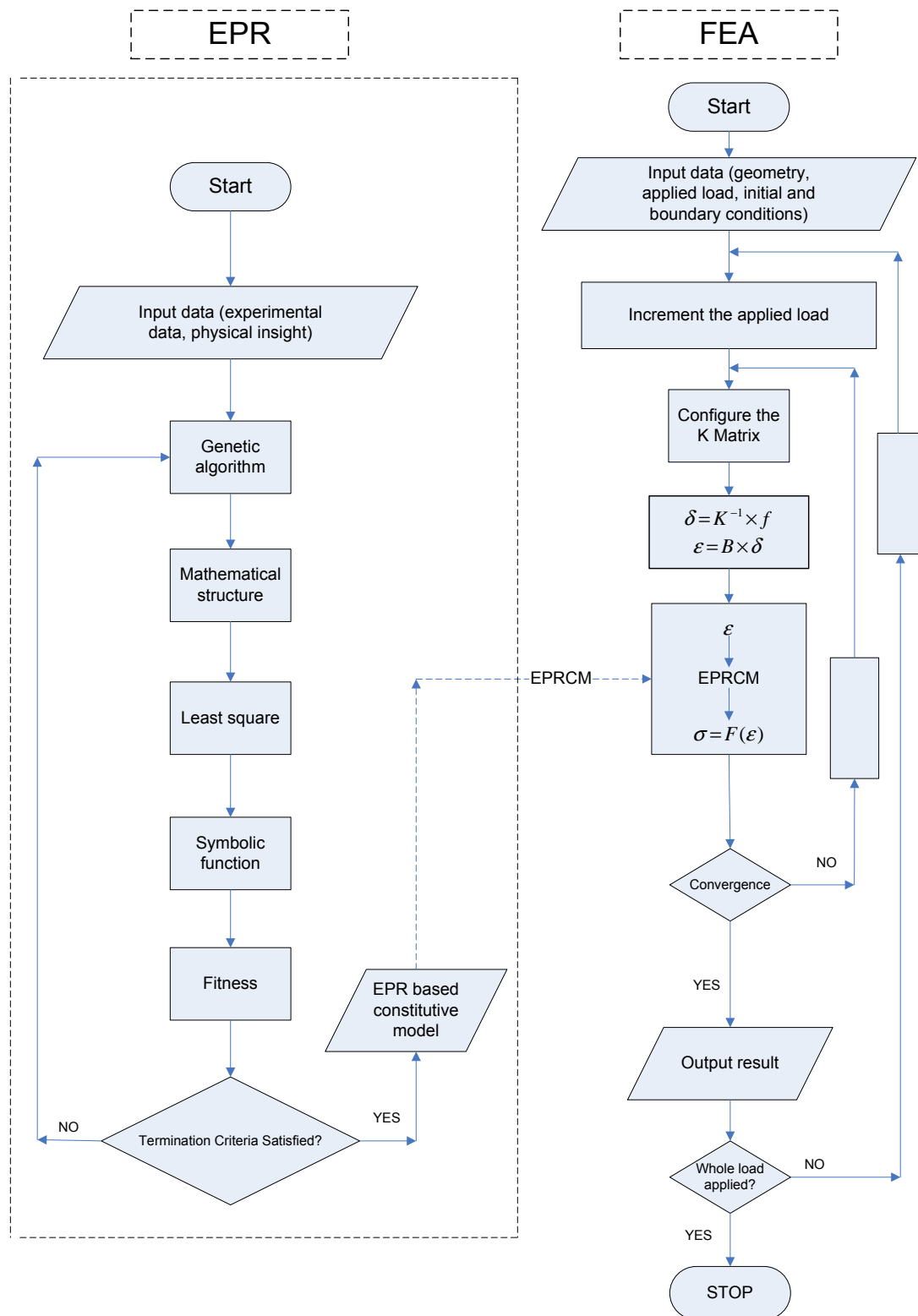
(Shin and Pande 2002). However, more recently it has been shown that NNCM can be practically incorporated in a finite element (or finite difference) code as a material model (e.g., Shin and Pande 2000, Javadi et al. 2003a). As it is already discussed in chapter 2, Shin and Pande (2000) presented a hybrid FE-ANN model and showed that the application of a constitutive law in the form of a neural operator leads to some qualitative improvement in the application of FE in engineering practice. They presented a procedure where data for training neural network based constitutive model were acquired from planned monitoring of structural tests. Indeed, unlike conventional procedures where generally material testing is performed to extract the stress-strain relationship and identify material parameters, in their procedure inverse analyses were carried out to identify material parameters from monitored global structural response. In this way the self learning capability of the software was expected; however for this purpose the results of structure behaviour needed to be available in advance. It is obvious that depending on the mesh size of the problem under consideration, huge amount of data may be accumulated with increasing the number of self learning cycles which can result in severe computer storage and CPU time problems during training. For this problem a limited number of monitoring points in the structure are selected and the data corresponding to these points are used to train the NNCM. Selection of the number and location of monitoring points is therefore of considerable importance in identifying a reliable NNCM. It was stated that such trained NNCM will need to be treated with caution for modelling the behaviour of other structures; as it is apparent that an NNCM may predict the correct response at a few points, yet may be completely inadequate to predict the response at others.

Javadi and his coworkers have carried out extensive research into application of neural networks in constitutive modelling of complex materials. They have developed an intelligent finite element method (NeuroFE code) based on the incorporation of a back propagation neural network (BPNN) in finite element analysis. (e.g., Javadi et al. 2002, 2003, 2004a, 2004b, 2005). In their work, they used actual material test results to extract stress-strain relationship and to train the NNCM. It has been shown that NNCMs trained in this way can be very efficient in learning and generalising the constitutive behaviour of complex materials and give better results when they are employed in a finite element code to analyse structures or domains made of the material under consideration.

### 6.3 Intelligent finite element method

Details of evolutionary polynomial regression have been described in chapter 4. EPR has several features which make it suitable for material modelling. First, it has learning capability. It can acquire, store and retrieve knowledge. This feature gives the EPR a self-organisation capability when new material information is presented. Second, it has an ability to generalise what it has learned, i.e., after an EPR has been trained on sufficient amount of test data it is able to generalise the behaviour to unseen cases. Third, it has fault tolerance capability. EPR is able to generalise correct material behaviour even if the training data are incomplete or partially incorrect (noisy data). This feature makes EPR especially suitable to extract material behaviour directly from noisy material test data. Fourth, EPR learns material behaviour directly from the measured stress-strain relationship without any assumption or postulations. Finally and most importantly EPR gives transparent and structured equations representing the constitutive behaviour of the material which is fairly straight forward to be incorporated in the finite element procedure.

An intelligent finite element code has been developed based on the integration of an EPR based constitutive model (EPRCM) in a finite element framework. In the developed methodology (EPR-FEM), the EPR is used as a unified framework for constitutive modelling of materials in finite element analysis. In this model, the EPRCM replaces the role of a conventional constitutive model. The source of knowledge for EPR is a set of raw experimental (or in-situ) data representing the mechanical response of the material to applied load. When EPR is used for constitutive description, the physical nature of the input-output data for the EPR is clearly determined by the measured quantities, e.g., stresses, strains, etc. The manner, in which EPRCM is incorporated in a FE code, is depicted in Figure 6.1.



**Figure 6.1** The incorporation of EPR based material model in a finite element code.

The function of a suitably trained EPR based constitutive model in a FE model is described in the following.

At every element's integration point

- i) For a given initial point:  $\{ {}^0(q_1), {}^0(\sigma_3), {}^0(\varepsilon_1), {}^0(\Delta\varepsilon_1) \}$  the role of the trained EPRCM, is to generate the value  ${}^1(q_1)$  on the output.
- ii) For the  $i+1^{\text{th}}$  load increment, the input pattern for the EPRCM contains: (1) the values of  $\{ {}^i(q_1), {}^i(\sigma_3), {}^i(\varepsilon_1) \}$  which have already been calculated in the previous loading step, and (2) an arbitrary value of  ${}^i(\Delta\varepsilon_1)$ . The new value of  ${}^{i+1}(q_1)$  is then calculated for the next step.
- iii) During each load increment, at each iteration the material model modulus for each element integration point can be calculated as

$$E_{EPR} = \frac{{}^i(\Delta q_1)}{{}^i(\Delta \varepsilon_1)} \quad (6.1)$$

$$\text{where } {}^i(\Delta q_1) = {}^{i+1}(q_1) - {}^i(q_1) \quad \text{and} \quad {}^i(\Delta \varepsilon_1) = {}^{i+1}(\varepsilon_1) - {}^i(\varepsilon_1)$$

The constitutive relationships are generally given in the form (Owen and Hinton, 1980)

$$\sigma = D\varepsilon \quad (6.2)$$

For an elastic material,  $D$  is the elastic modulus tensor for the matrix. Assuming that the matrix is elastic and isotropic, matrix  $D$  is given in terms of Young's modulus,  $E$ , and Poisson's ratio,  $\nu$ . For a plane strain case, for example

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix} \quad (6.3)$$

for a plane stress case

$$D = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \quad (6.4)$$

and for an axisymmetric case

$$D = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \nu/(1-\nu) & 0 & \nu/(1-\nu) \\ \nu/(1-\nu) & 1 & 0 & \nu/(1-\nu) \\ 0 & 0 & (1-2\nu)/2(1-\nu) & 0 \\ \nu/(1-\nu) & \nu/(1-\nu) & 0 & 1 \end{bmatrix} \quad (6.5)$$

In the EPR based finite element procedure, during each load increment the calculated  $E_{EPR}$  from Equation (6.1) is used as the young's modulus to form the material stiffness matrix ( $D$ ) for every particular element which is being considered at the time. Therefore with  $E_{EPR}$  being recalculated through the FE procedure, the element stiffness matrix for every single element is updated at each load increment as

$$\int_{\Omega} B^T D B d\Omega \quad (6.6)$$

where  $B$  is the strain matrix and  $\Omega$  is the elemental area. Consequently the global stiffness matrix for a particular problem is refined for each loading step. The whole procedure ensures that the constitutive model follows the actual behaviour of the material, both at the element level and at the global level. This is schematically shown in Figure 6.2.

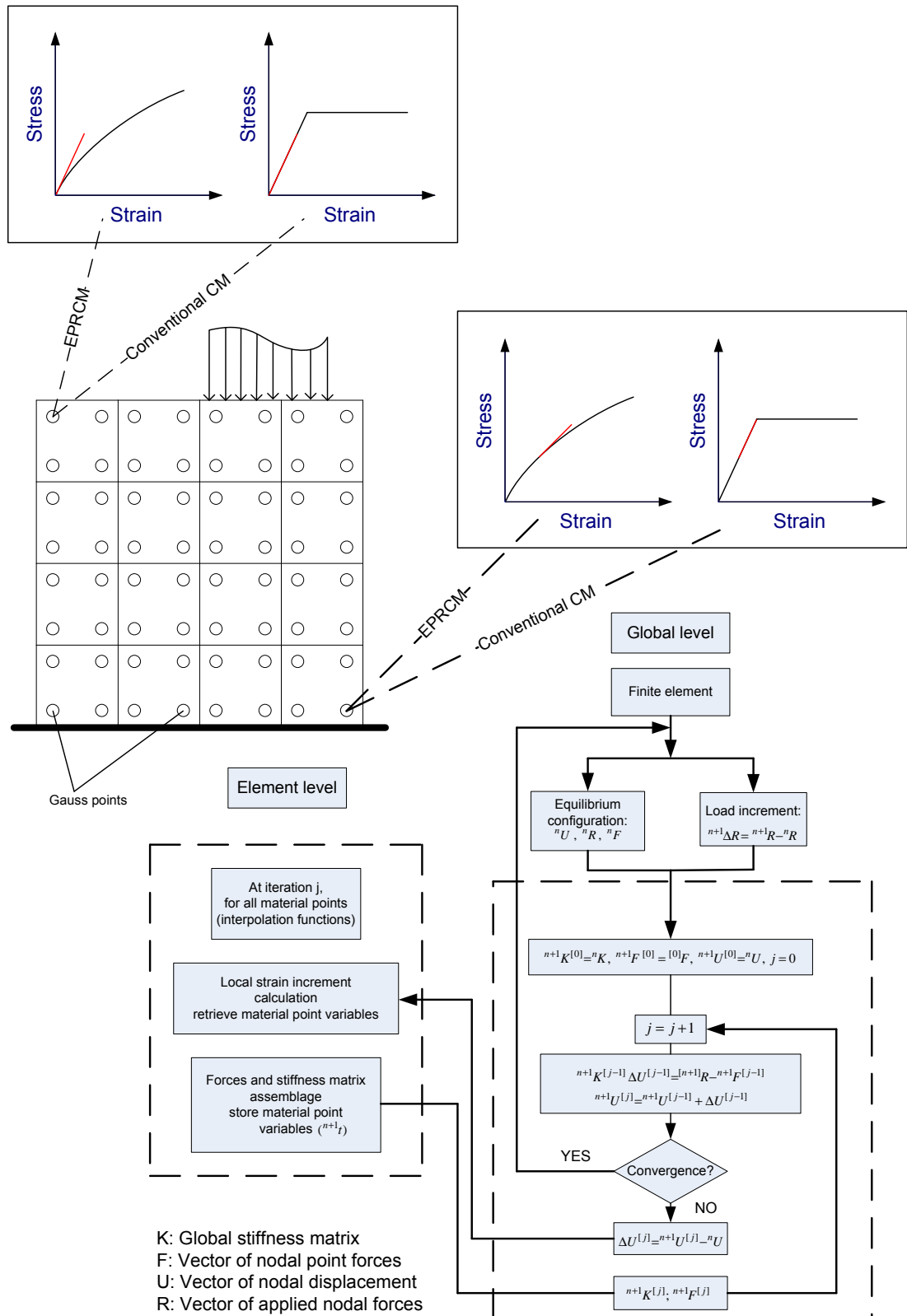
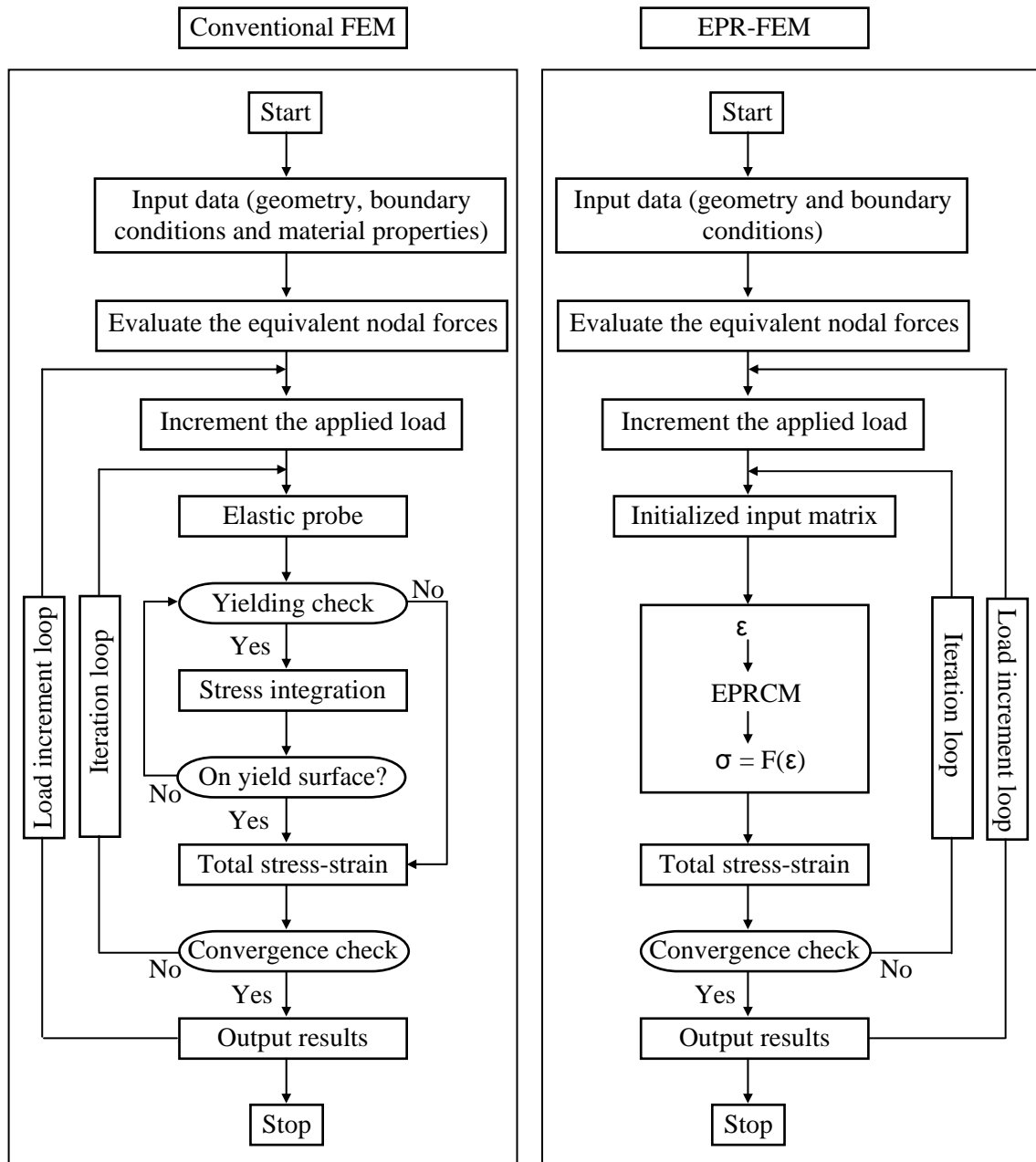


Figure 6.2 Finite element solution strategy in nonlinear analysis using EPRCM.

In contrast to the procedure illustrated in Figure 6.2, in a conventional elasto-plastic FE analysis, the element stiffness must be calculated for the first iteration of the first load increment assuming elastic behaviour. Every other time the element stiffness matrix is recalculated, it is controlled with a check that whether an element yielded in previous step or not, if the answer is yes for a particular element the corresponding element stiffness matrix has to be refined to account for any plastic deformation of the material and subsequently the elasto-plastic matrix ( $D_{ep}$ ) must be employed for that element. Apart from this change the element stiffness formulation process is identical to that for elastic materials.

It should also be mentioned that the incorporation of EPRCM in a FE code is more straightforward than a conventional constitutive model as such intelligent FE model does not require complex yielding/plastic potential/failure functions, flow rules, etc. There is no need to check yielding, compute the gradients of the plastic potential curve, update the yield surface or redistribute stresses. The flow charts in Figure 6.3 compare nonlinear FE procedures using a conventional constitutive model and the developed EPRCM.



**Figure 6.3** Comparison of conventional FEM and EPR-FEM.

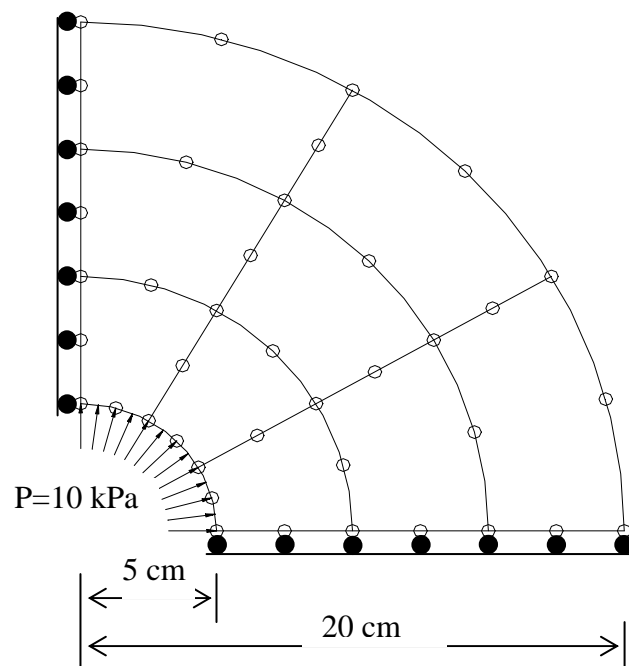
### 6.3.1 Numerical examples

When a material model has been developed, it should operate in a stable form in numerical analyses. As EPRCM has been introduced for the first time in this work, consecutively no research on this aspect of EPRCM has been performed to date. Therefore the focus of this section is to validate the EPRCM, which was generated in chapter 5, in FE analyses and to illustrate the capabilities of the developed computational methodology, which was introduced in previous section, in practice. Initially the feasibility of using EPRCM in FE analysis is examined through simple linear and nonlinear structural examples. Then more sophisticated geotechnical examples are set up and using EPRCM which has been trained with experimental data described in chapter 5 and incorporating it in a conventional finite element framework, general FE analyses are carried out for the problems. The results of EPR based FE analyses are compared with the results from conventional material models with parameters which have been obtained from the same data sets as those used for developing the EPRCM. Finally the potential application of EPR based constitutive modelling for coupled situations is presented through an example. It is expected that this will lead to a deeper understanding of the performance of an EPRCM in finite element framework and provide the way for the use of EPRCMs in practical problems.

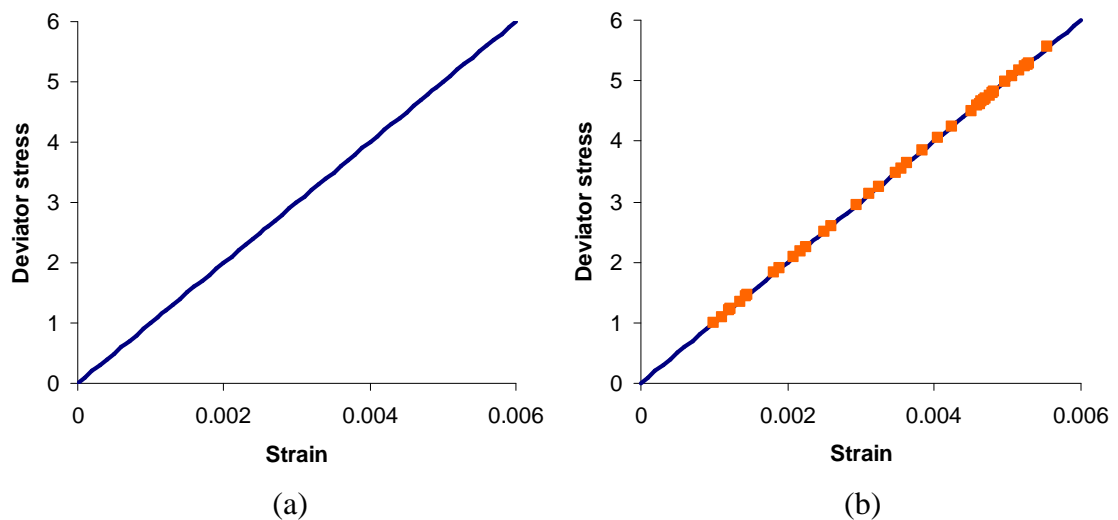
#### 6.3.1.1 Example 1

This example involves a thick circular cylinder conforming to plane strain conditions. Figure 6.4 shows the geometric dimensions and the element discretisation employed in the solution. Due to the symmetry, only a quarter of the domain has been analysed. 9 eight-node isoparametric elements have been used in the discretisation. The cylinder is made of linear elastic material with a Young's modulus of  $E=1000$  kPa and a Poisson's ratio of 0.3 (Hinton and Owen, 1980). This example was deliberately kept simple in order to verify the computational methodology by comparing the results to available analytical solutions as well as those of a linear elastic finite element model. The loading case considered involves an internal pressure of 10 kPa with boundary conditions as shown in Figure 6.4. Figure 6.5(a) shows a linear elastic stress-strain relationship with a slope of 1000 that represents elastic modulus of 1000 kPa for the material. The data

from this figure were used to train the EPR in order to capture the linear stress-strain relationship for the material.



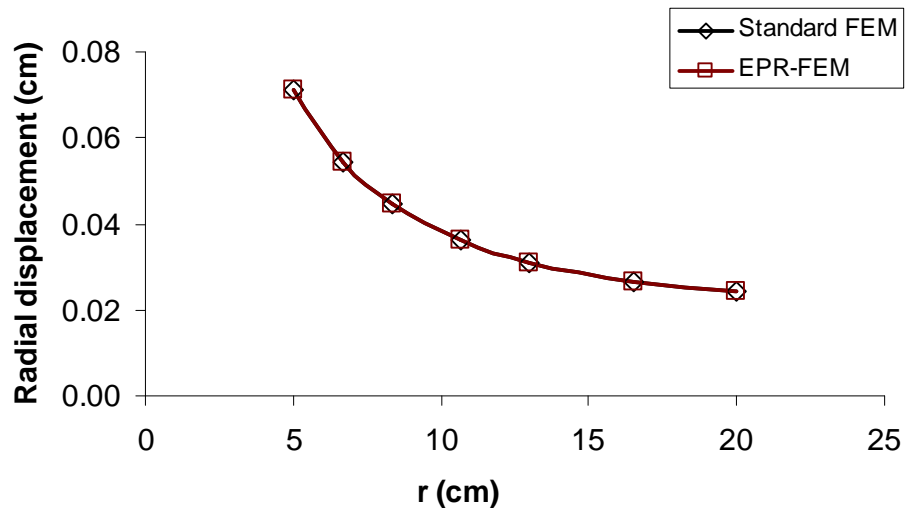
**Figure 6.4** Finite element mesh in symmetric quadrant of a thick cylinder with the boundary conditions.



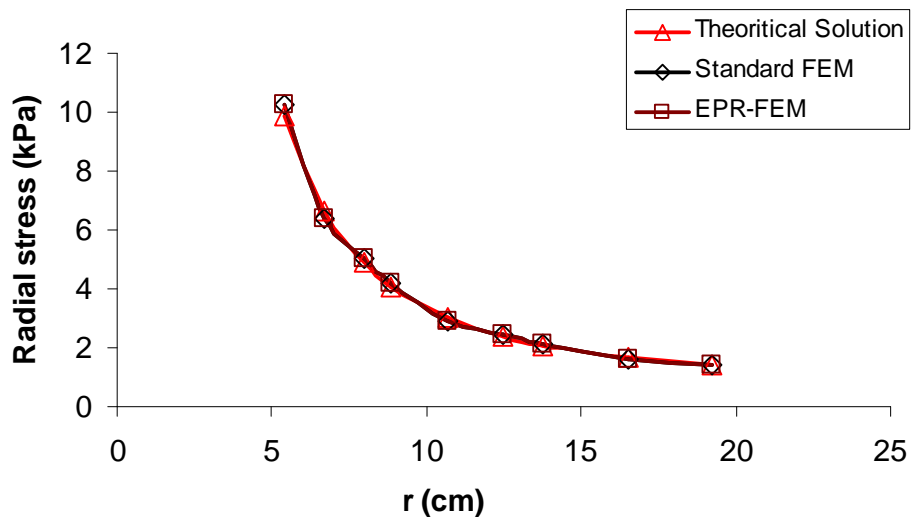
**Figure 6.5** (a) Linear stress-strain relationship used for training, (b) the results of EPR predictions for stress-strain values.

The EPR-based finite element model incorporating the trained EPR was used to analyse the behaviour of the cylinder under the applied internal pressure. The results are compared with analytical solutions as well as those obtained using a standard linear

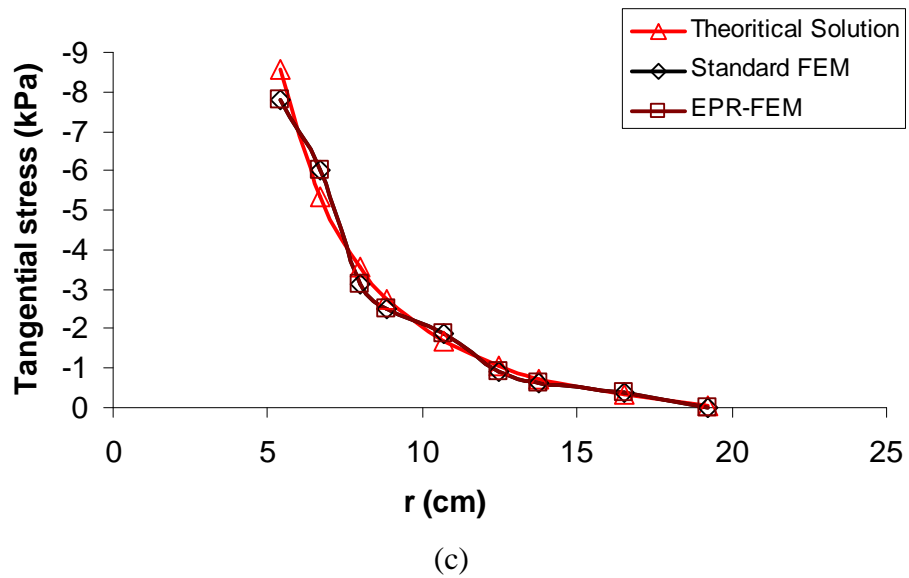
elastic finite element method. Figure 6.6 shows the radial displacements, radial stresses and tangential stresses along a radius of the cylinder, predicted by the three different methods. Comparison of the results shows that the results obtained using the EPR based FEM are in excellent agreement with those attained from the analytical solution as well as the standard finite element analysis. This shows the potential of the developed EPR based finite element method in deriving constitutive relationships from raw data using EPR and using these relationships to solve boundary value problems.



(a)



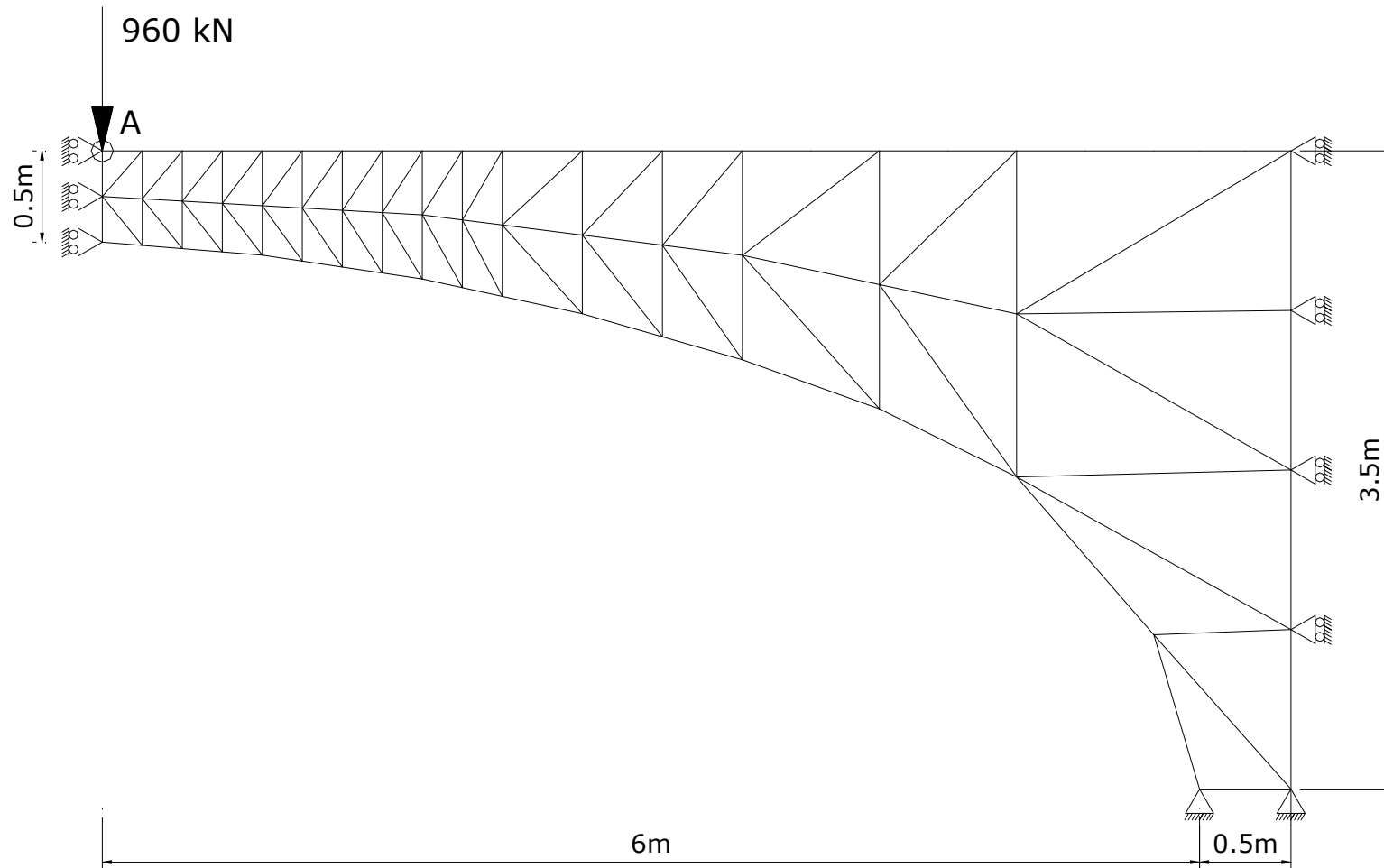
(b)



**Figure 6.6** Comparison of the results of the EPR-FEM, standard FEM and theoretical solution in terms of (a) radial displacement, (b) radial stress and (c) tangential stress.

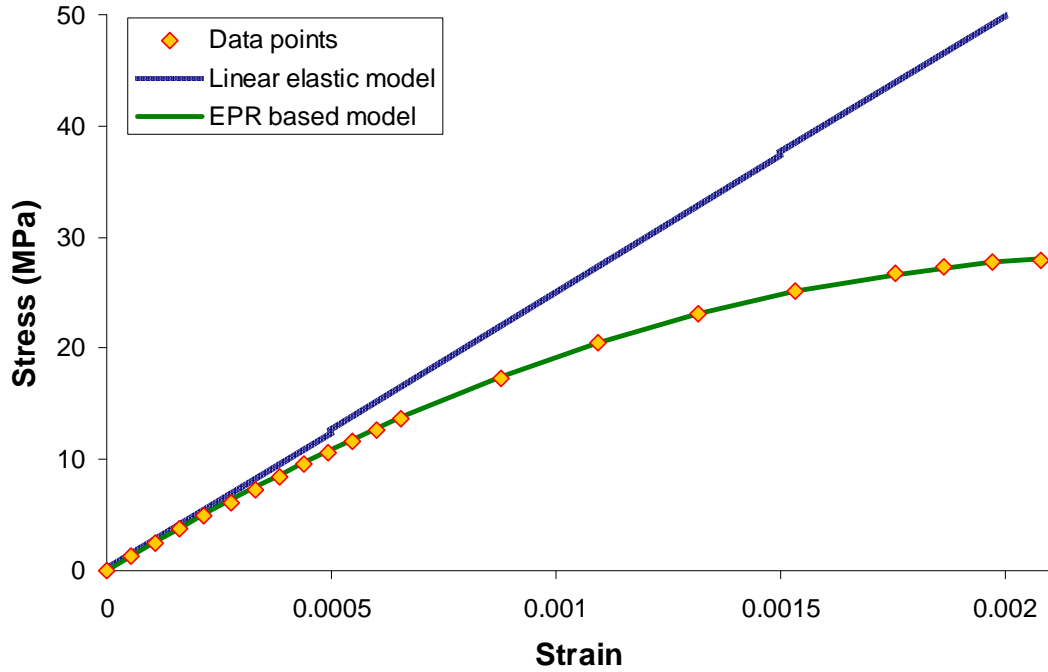
### 6.3.1.2 Example 2

The second example is an arch bridge with dimensions of 12 m length and 3.5 m height. Due to symmetry just half of the bridge was analysed. The geometry of the bridge together with the FE mesh and boundary conditions are shown in Figure 6.7. As seen in the figure, 68 triangle elements have been used in the discretisation of the problem.



**Figure 6.7** Geometry of the bridge and the FE mesh showing boundary conditions.

This example has been selected to verify the proposed computational methodology by comparing the results of the linear elastic finite element solution with the results obtained from an intelligent EPR based nonlinear FE model. For the EPR based nonlinear FE analysis the stress-strain curve for bridge material shown in Figure 6.8 was adopted into the analysis.



**Figure 6.8** EPR based stress-strain curve passing through experimental stress-strain data points (adopted from Marsono 2000).

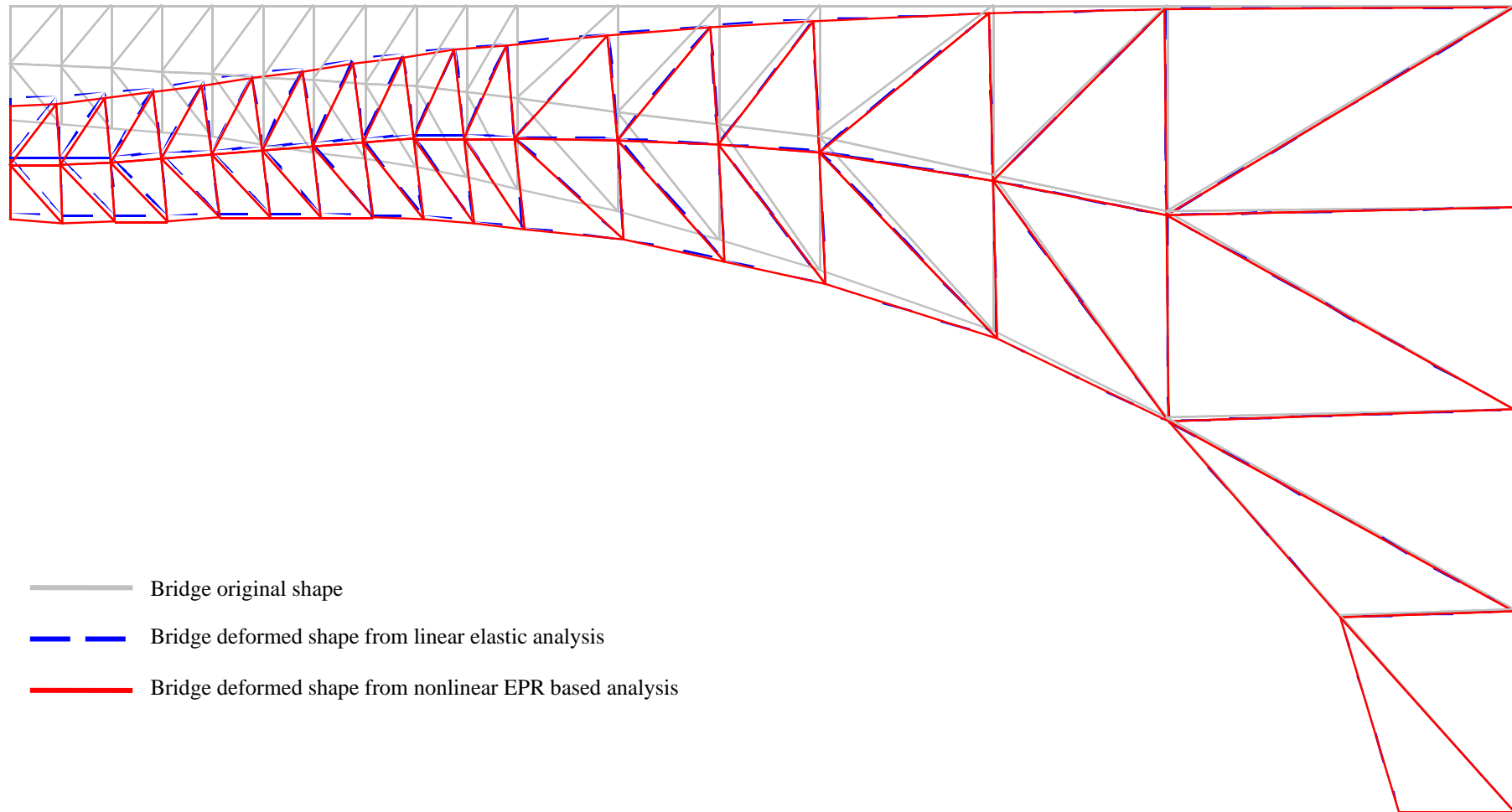
The stress-strain data points shown in Figure 6.8 were obtained from the results of a compression test over a concrete material. The selected EPR model for the curve passing through experimental data points is

$$\sigma = (-5.3E+12) \varepsilon_1^2 + (2.45E+10) \varepsilon_1 \quad (6.7)$$

By definition, in the nonlinear FE procedure the tangential elastic modulus of the material at each strain can be obtained from the derivative of the Equation (6.7), therefore the EPR based elastic modulus of the concrete material can be taken as

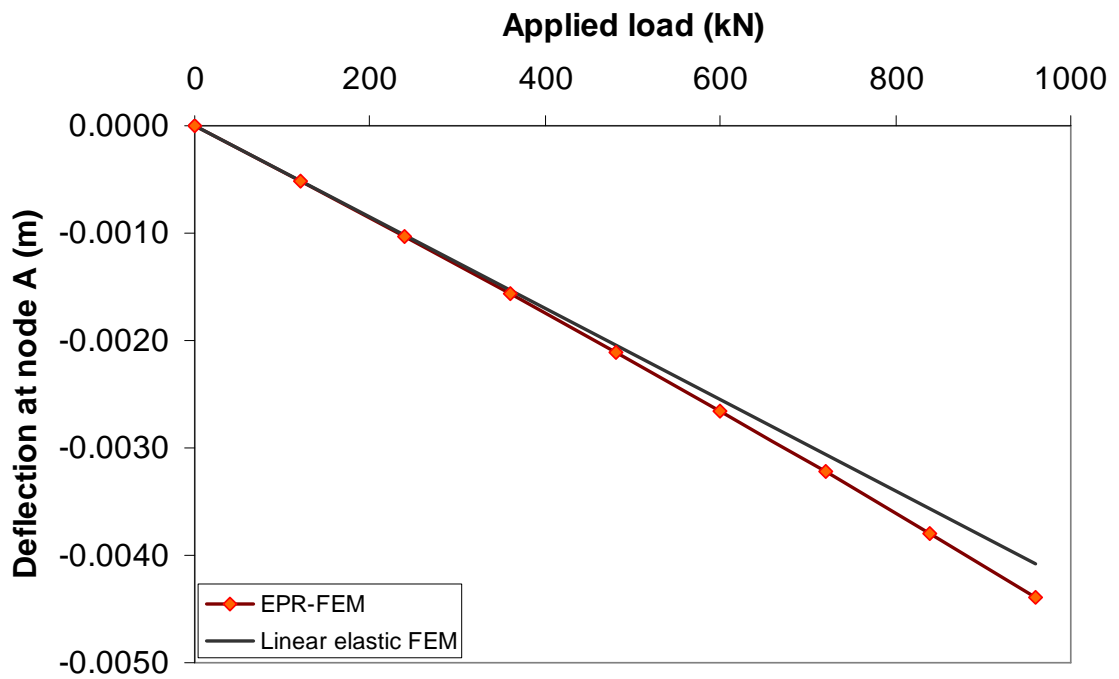
$$E_{EPR} = (-1.06E+13) \varepsilon_1 + (2.45E+10) \quad (6.8)$$

For conventional linear analysis the initial value of  $E$  equal to 24.5 GPa was taken as the elastic modulus of the bridge material. The bridge was analysed under a vertical point load of 960 kN at point “A” as shown in Figure 6.7 as well as its own weight. For both analyses the Poisson’s ratio was kept constant (0.2) and the unit weight of the bridge material was taken as 25 kN/m<sup>3</sup>. Figure 6.9 shows the comparison between the results of linear finite element analysis and those of the proposed intelligent finite element analysis. In order to clearly distinguish the results obtained by the two separate analyses, the results were magnified by a factor of 100. From the figure it can be seen that EPR based model was able to capture the nonlinear stress-strain relationship and reflect this relationship inside the FE procedure with high precision.



**Figure 6.9** Comparison of the results obtained from conventional linear elastic analysis and intelligent nonlinear EPR based analysis.

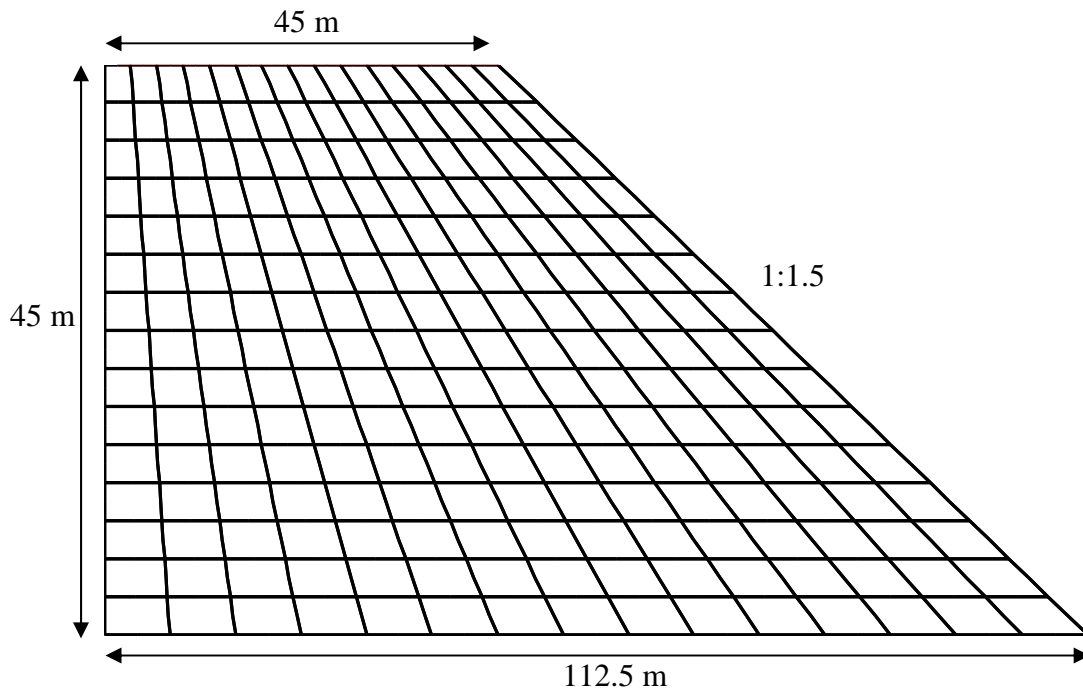
The difference between the values of deflection at point “A” obtained from the two separate analyses is highlighted in Figure 6.10.



**Figure 6.10** Comparison between the values of deflection at the centre of bridge calculated from conventional linear and EPR based nonlinear FE models.

### 6.3.1.3 Example 3

This example involves the analysis of stresses, strains and displacements in an embankment of Mohr-Coulomb material subjected to gravity loading. The finite element mesh used included 225 eight-node isoparametric elements and 736 nodes. This is illustrated in Figure 6.11 together with the geometry of the embankment.



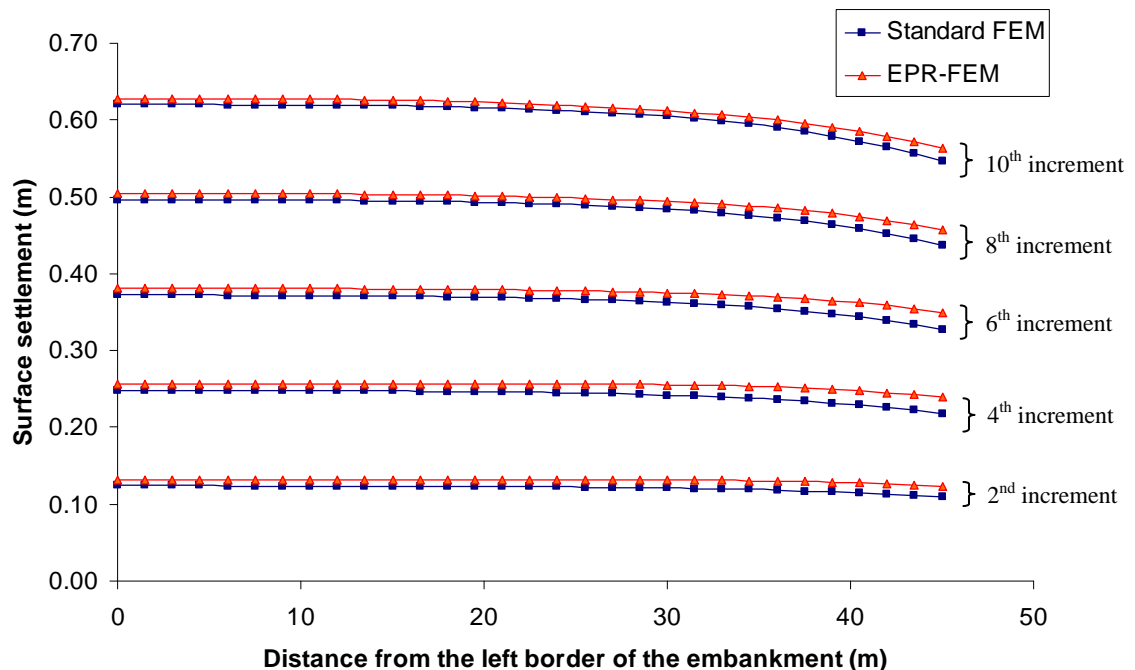
**Figure 6.11** Geometry of the embankment and the FE mesh.

The results from a series of triaxial tests, described in section 5.2.1.1, were used in this example for the training of the EPR model with an incremental stress-strain (tangential stiffness) strategy. It was assumed that the soil tested was representative of the material of the embankment. The test data were arranged as explained in section 5.2.1.1 of chapter 5 and used to model the stress-strain relationship for the soil. The obtained EPRCM in the form of equation 5.15 was incorporated in the EPR based finite element model. The developed FE model incorporating the EPRCM, as described in the previous section, was then used to simulate the behaviour of the embankment under the gravity loading. For the conventional finite element analyses, the Mohr-Coulomb model was selected in which the material parameters used for this model were obtained from the triaxial test results over samples of silty soil with 5 percent clay, as summarised in Table 6.1.

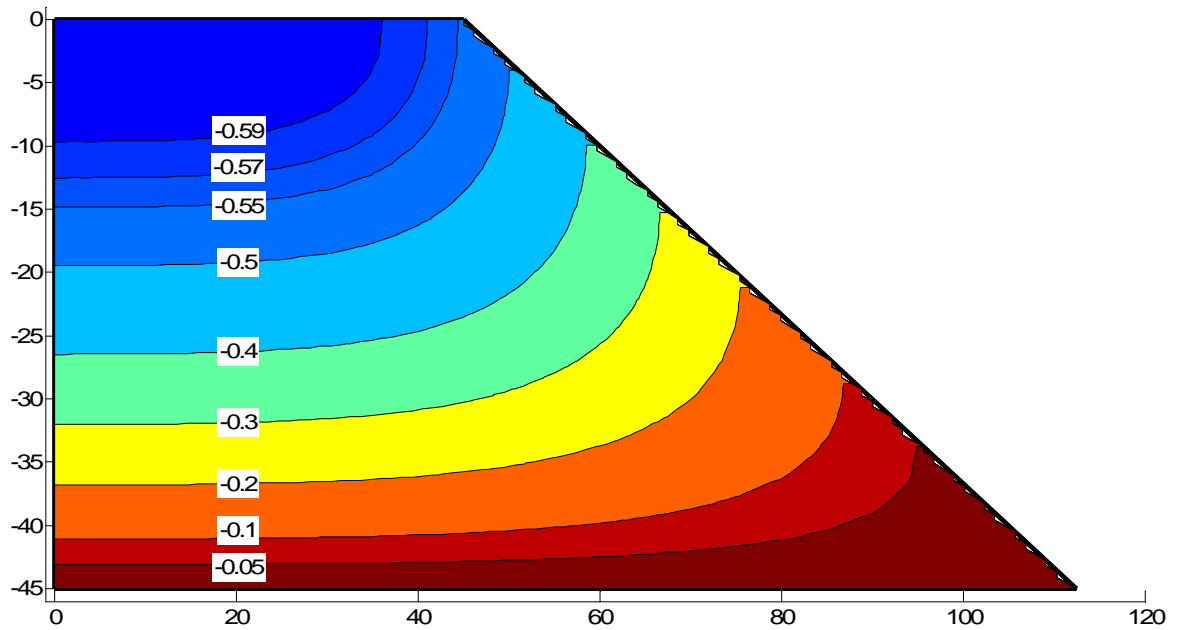
**Table 6.1** Material parameters for Mohr-Coulomb model.

$c'$ (kPa)	$\phi'$ (deg.)	$\nu$	$E$ (kPa)	$\gamma$ ( $kN/m^3$ )
20	30	0.33	20000	16

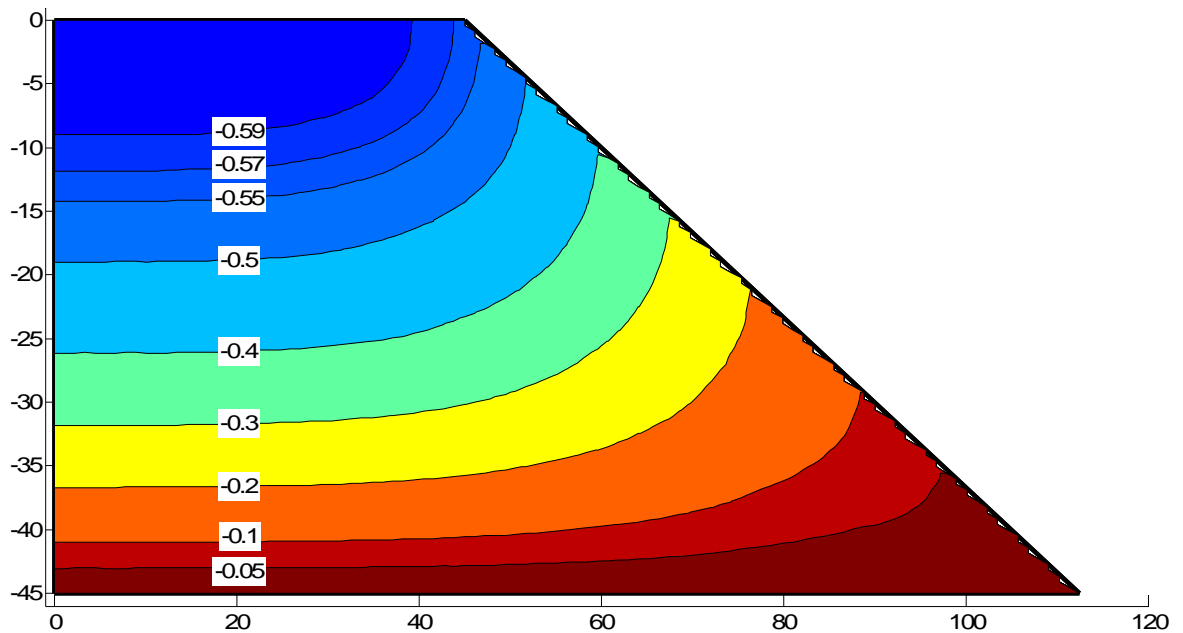
The comparison of results in terms of surface settlement, contours of displacement in X and Y directions, and major stresses and strains are shown in Figures 6.12 to 6.16. The standard finite element results were obtained from analysing the embankment using Mohr-Coulomb elastic-plastic model and the intelligent based finite element results were obtained where the raw data from the triaxial tests were directly used in deriving the EPR based constitutive relationship and consequently this model was incorporated into the intelligent finite element code. For both analyses, the total weight of embankment was applied incrementally. From the results obtained, it can be seen that intelligent finite element method is capable of capturing the complex constitutive relationships of the soil and offer very good prediction of the behaviour of the embankment under gravity loading.



**Figure 6.12** Comparison of the computed results for the surface settlement of embankment at different increments using standard FEM and EPR-FEM.

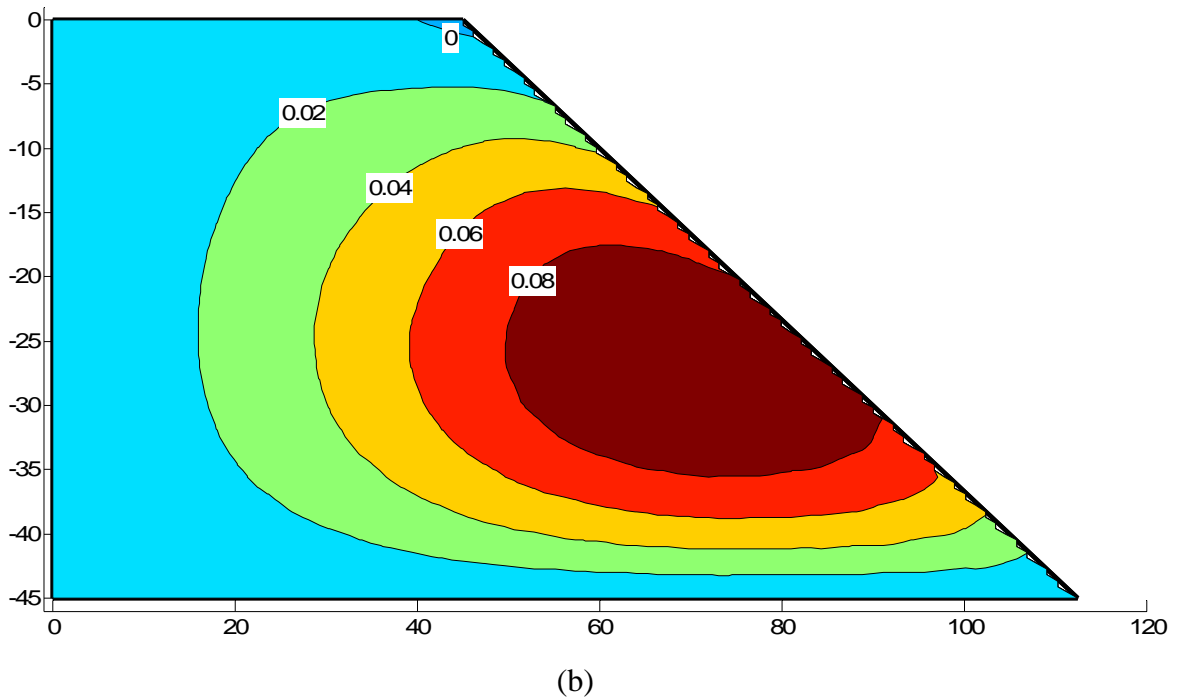
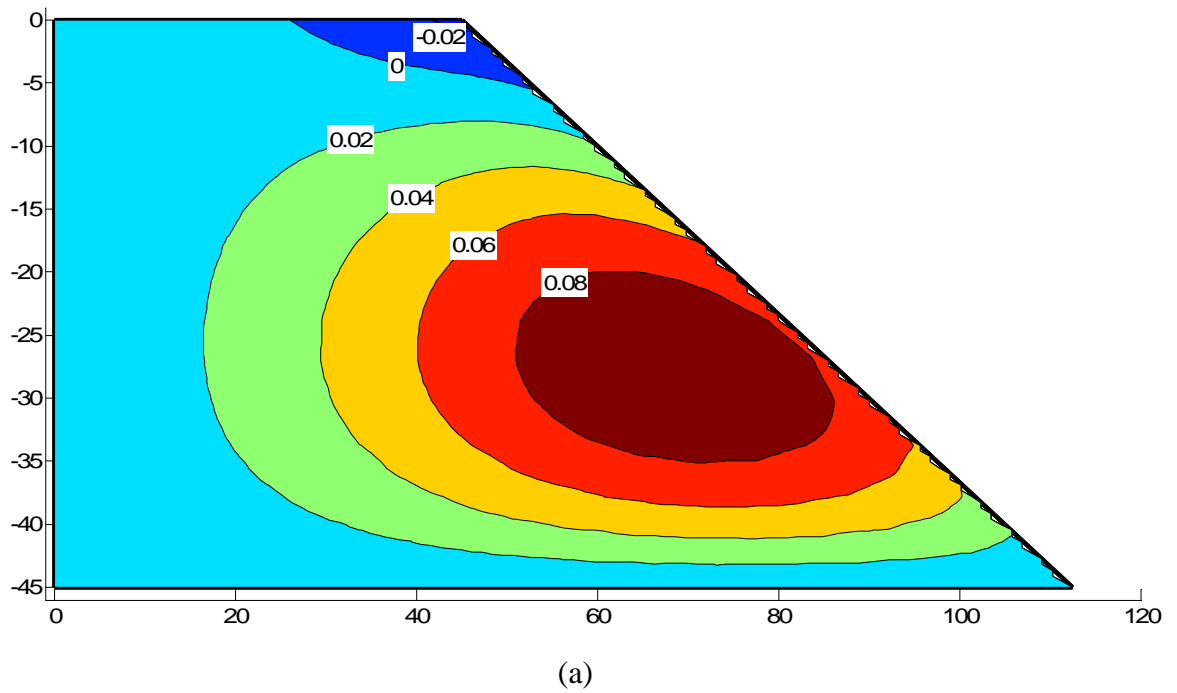


(a)

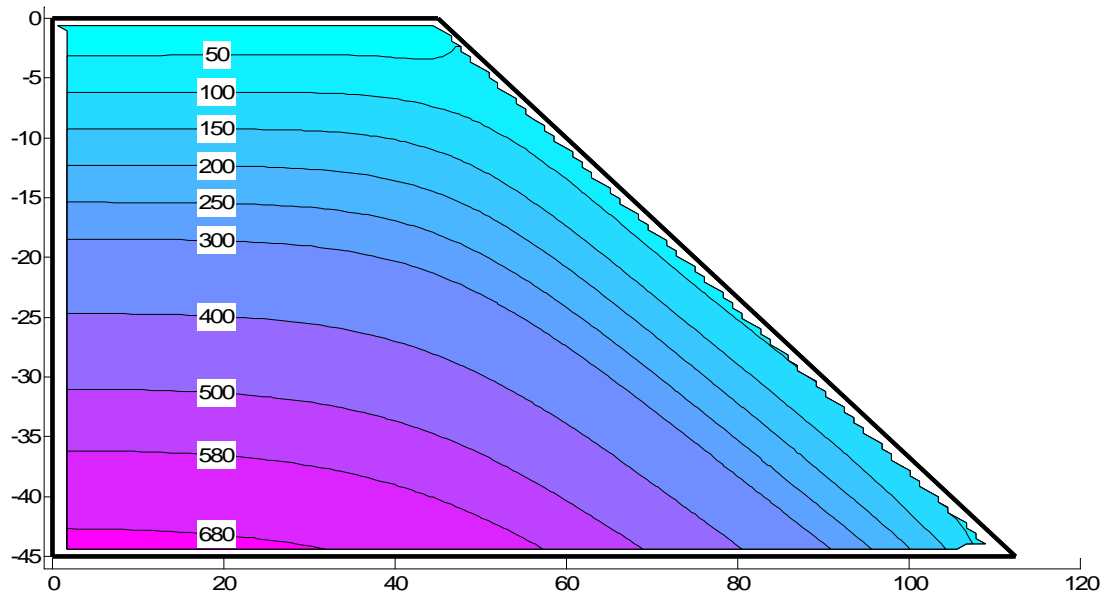


(b)

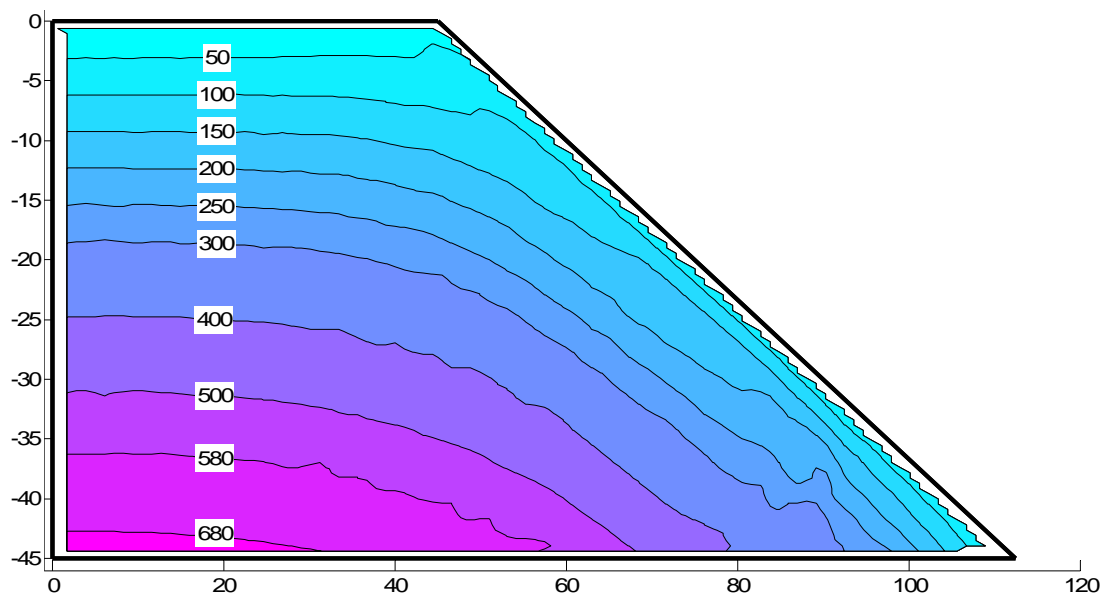
**Figure 6.13** Comparison of the results for displacements in Ydirection showing (a) displacement contours from standard FE analysis and (b) displacement contours from EPR based FE analysis.



**Figure 6.14** Comparison of the results for displacements in Xdirection showing (a) displacement contours from standard FE analysis and (b) displacement contours from EPR based FE analysis.

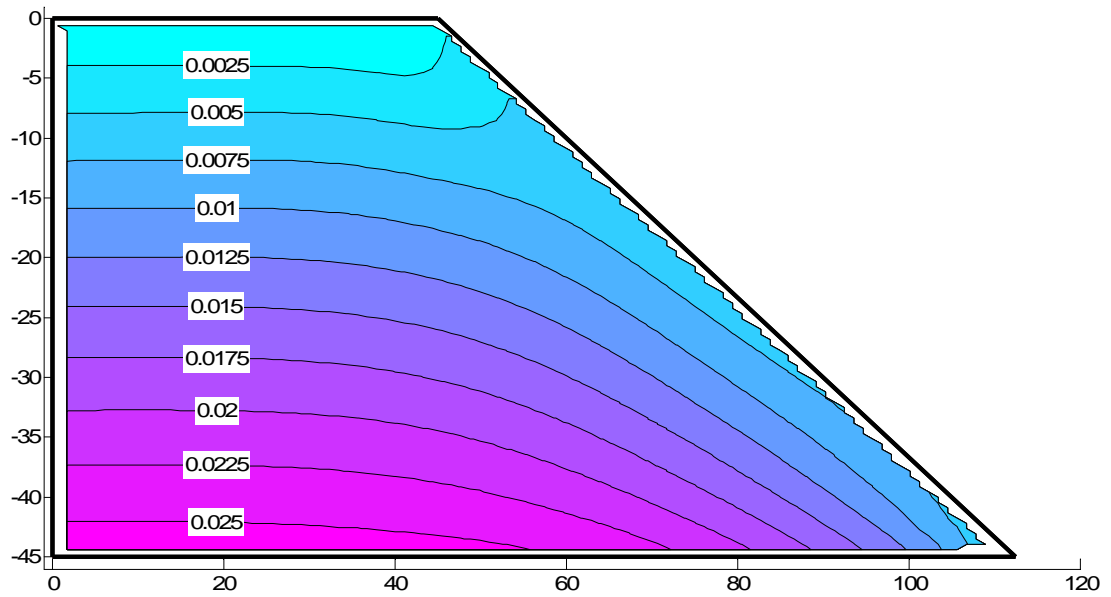


(a)

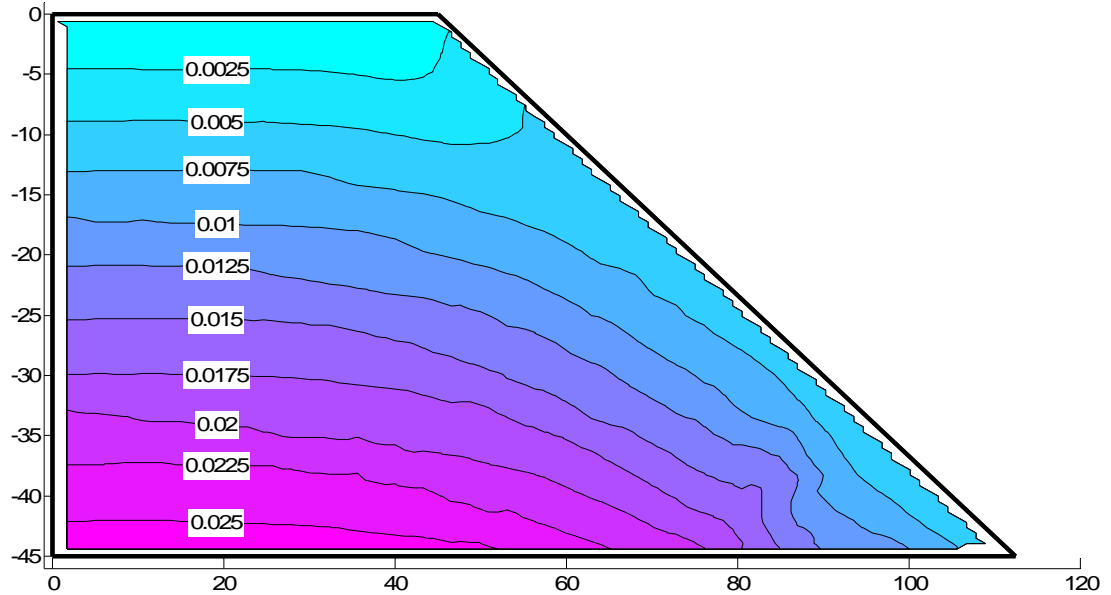


(b)

**Figure 6.15** Comparison of the results for major principal stress showing (a) stress contours from standard FE analysis and (b) stress contours from EPR based FE analysis.



(a)

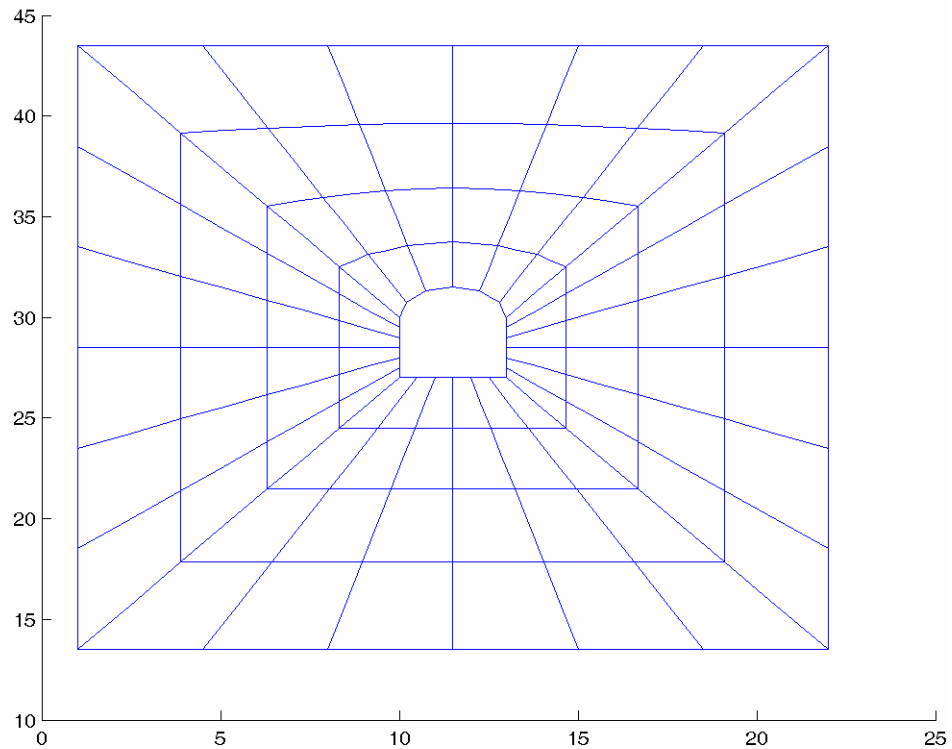


(b)

**Figure 6.16** Comparison of the results for major principal strain showing (a) strain contours from standard FE analysis and (b) strain contours from EPR based FE analysis.

### 6.3.1.4 Example 4

This example involves the analysis of deformation around a tunnel subjected to gravity and excavation loadings. The geometry of the tunnel and the finite element mesh are shown in Figure 6.17. The finite element mesh includes 96 eight-node isoparametric elements and 336 nodes. The depth of the tunnel crown from the ground surface is 12 m.



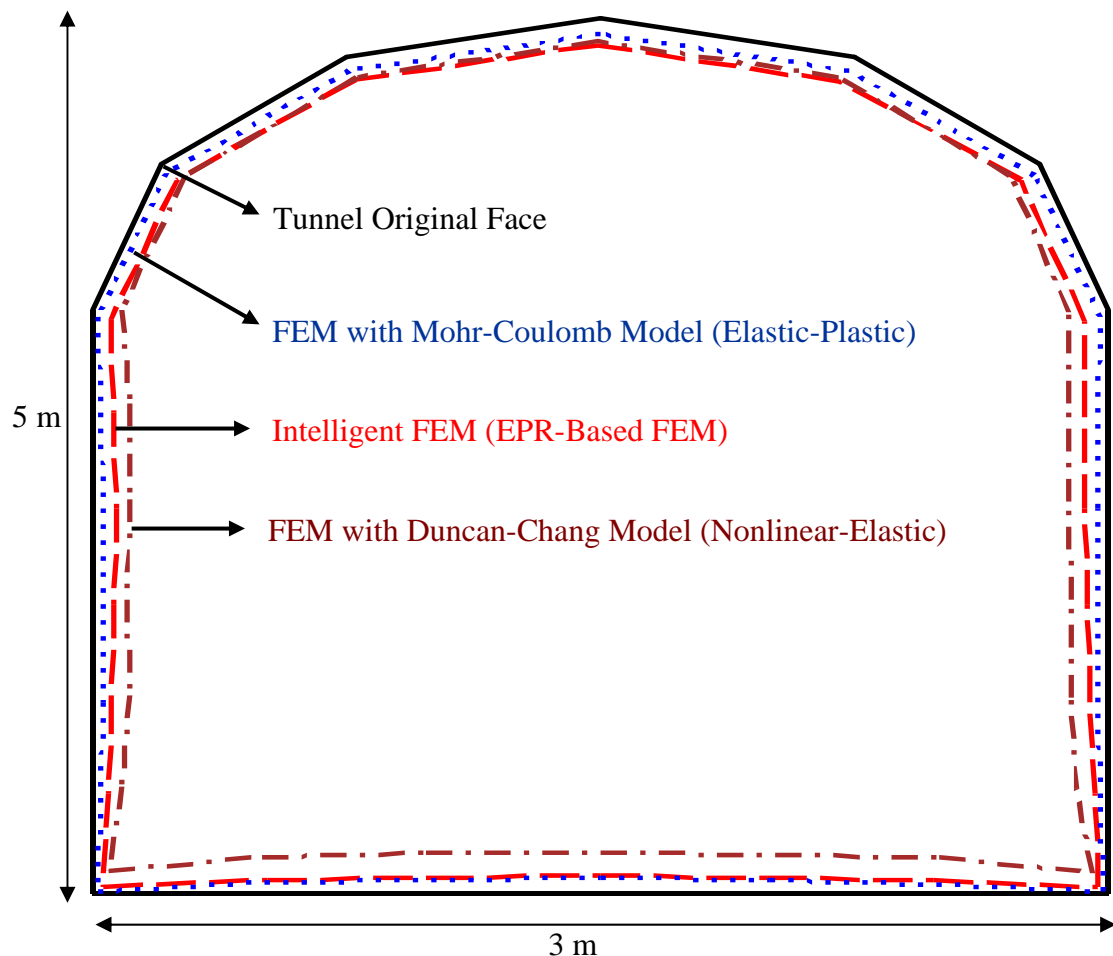
**Figure 6.17** Geometry of the tunnel and the FE mesh.

The results from the same series of triaxial tests as used for the embankment example were utilised in this analysis for the training of the EPR based constitutive models. It was assumed that the soil tested was representative of the material around the tunnel. The obtained EPRCM in the form of Equation (5.15) was incorporated in the EPR based finite element model. The developed FE model incorporating the EPRCM was then used to simulate the behaviour of the tunnel under gravity and excavation loadings. For the conventional finite element analyses, the results of the triaxial tests were used to derive the material parameters for the Mohr-Coulomb (elastic-plastic) and Duncan-Chang (nonlinear elastic) models for the soil. These parameters are summarised in Table 6.2.

**Table 6.2** Material parameters for Duncan-Chang (1970) and Mohr-Coulomb models.

$c'$ (kPa)	$\phi'$ (deg.)	$R_f$	$K$	$n$	$\nu$	$E$ (kPa)	$\gamma$ ( $kN/m^3$ )
20	30	0.8121	159.44	0.6233	0.33	20000	16

Figure 6.18 shows a comparison between the deformation in the tunnel face predicted using standard finite element analyses with the Duncan-Chang nonlinear elastic and Mohr-Coulomb elastic-plastic models as well as the proposed EPR based finite element method (EPR-FEM) where the raw data from the triaxial tests were directly used in deriving the EPR-based constitutive model.

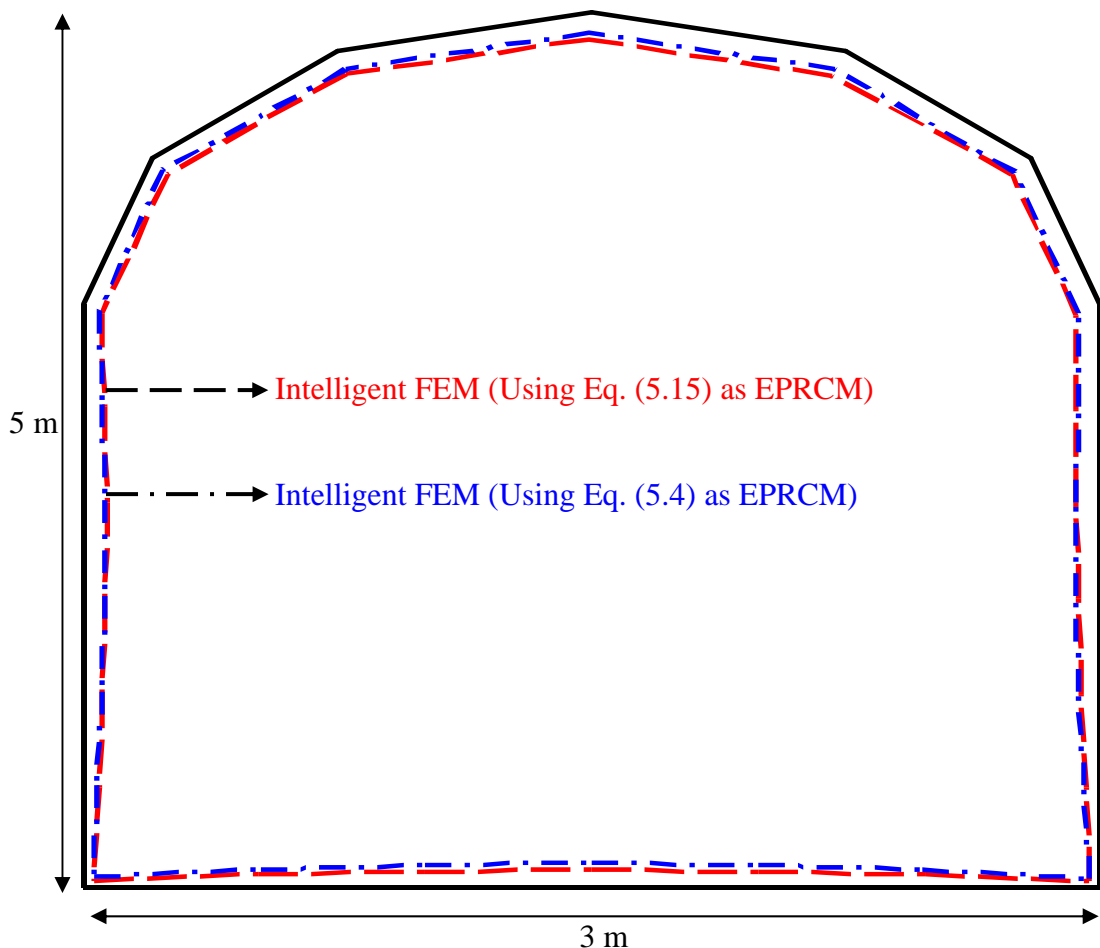


**Figure 6.18** Comparison of the results of the intelligent FEM and conventional FE analyses.

Figure 6.18 shows that the patterns of deformation are similar in all 3 analyses. The FE analysis using the Duncan-Chang model seems to have over-predicted the displacements in the tunnel whereas the results of the intelligent FE analysis and the FE analysis using Mohr-Coulomb model are very close. Despite the relatively small difference between the results from the different analyses, it can be argued that the

EPR-FEM results are more reliable, as this method used the original raw experimental data to learn the constitutive relationships for the material and it did not assume any particular constitutive relationships, yield conditions, etc. From the results obtained, it is shown that the developed EPR based finite element method is also capable of capturing more complex constitutive relationships of materials and can offer very realistic prediction of the behaviour of structures.

Furthermore, in order to examine the effect of using different EPRCM relationship on the results, the simplest equation obtained to represent the constitutive behaviour of the soil under consideration (i.e., Equation (5.4)) was implemented into the FE procedure and the analysis was repeated. Figure 6.19 shows the results of this analysis compared to the results obtained using Equation (5.15).

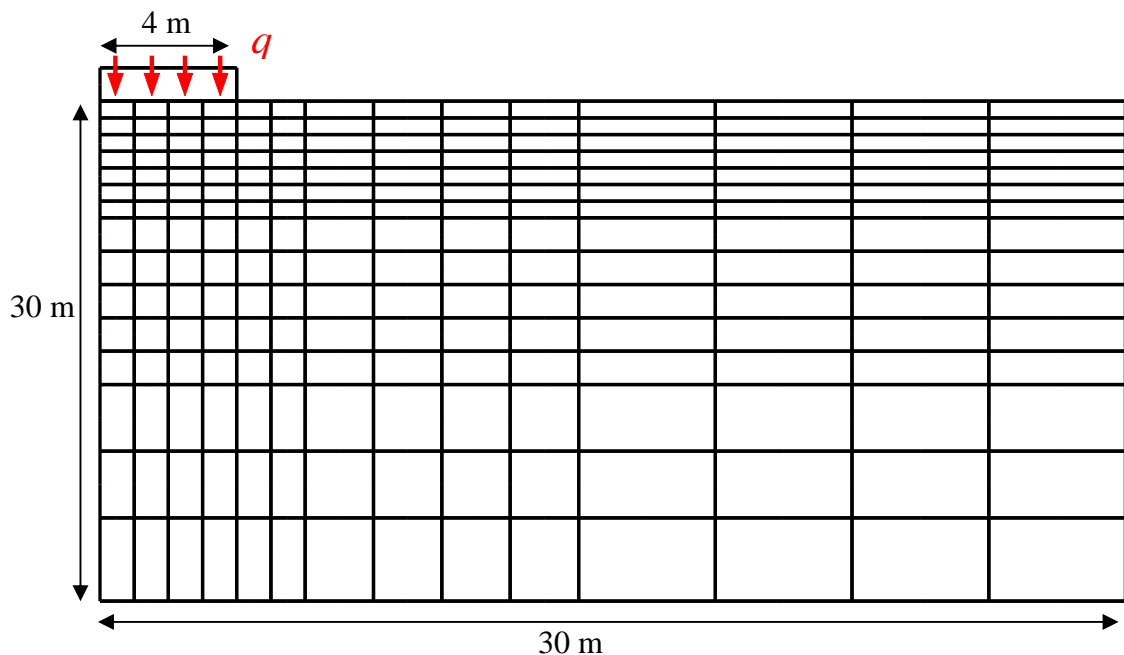


**Figure 6.19** Comparison between deformation results of the tunnel wall using two different EPRCMs.

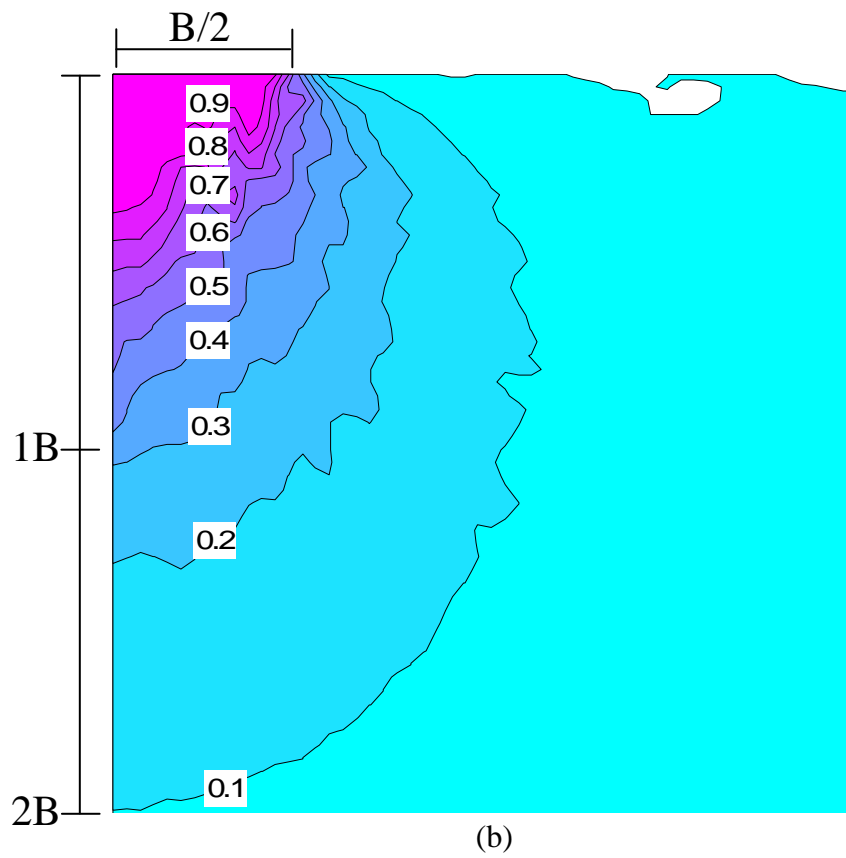
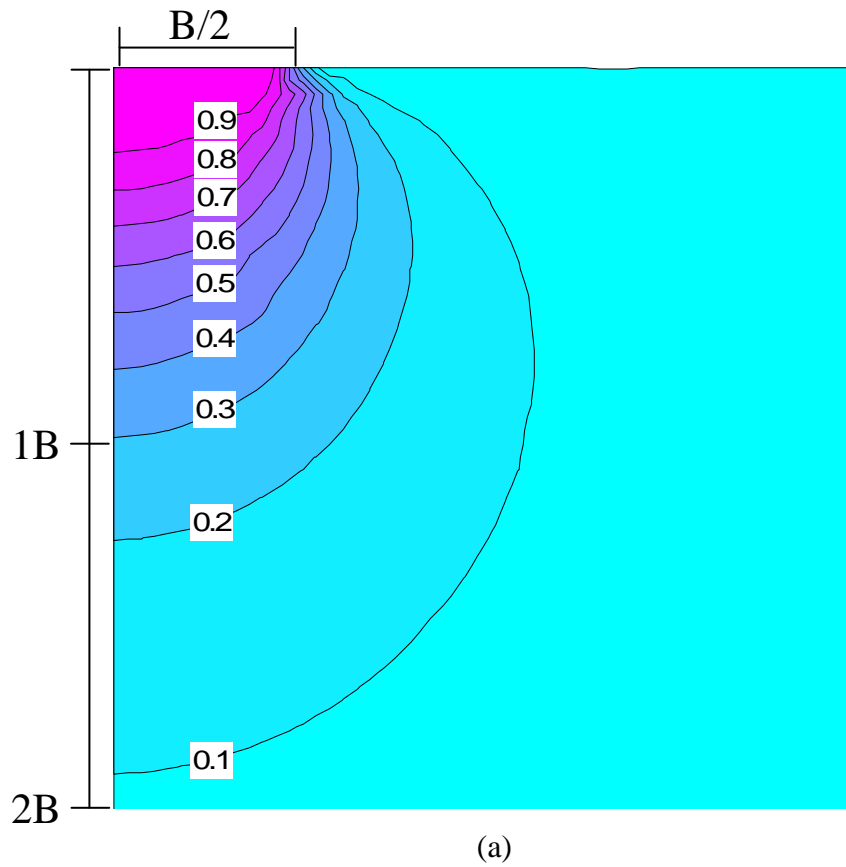
As it can be seen in the Figure 6.19 the FE results using the simplest EPRCM and the most complex EPRCM are very close to each other. This also implies the robustness of the intelligent FE procedure and indicates that even the very simple EPR based constitutive relationship which is developed to represent the soil behaviour is representative enough to give realistic results.

### 6.3.1.5 Example 5

In this example the surface settlement and the vertical stress distribution under a circular footing is modelled using the EPR based FEM. The soil properties were considered the same as those used in examples 3 and 4. Reflecting the circular geometry of the foundation, this problem was modelled using axisymmetric FE formulation. Figure 6.20 shows the finite element mesh of the problem which includes 210 isoparametric elements (eight-node) and 689 nodes. Smaller elements were used close to the region of higher stresses. The problem was analysed under the footing pressure of 200 kPa. Figure 6.21 presents the resulting vertical stress contours due to the applied pressure, obtained from (a) the pressure bulb charts presented in Lambe and Whitman (1979) and (b) the EPR based finite element model presented in this study.



**Figure 6.20** Geometry of the footing and the FE mesh.



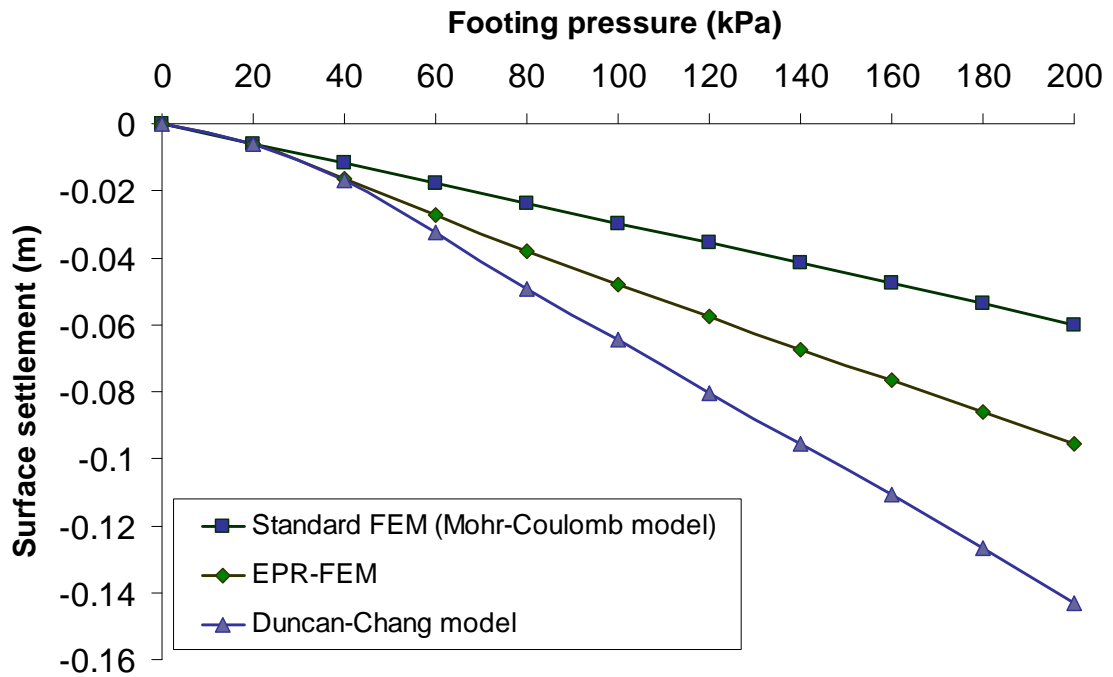
**Figure 6.21** Contours of vertical stress obtained from (a) charts presented in Lambe and Whitman (1979) and (b) results of EPR-FEM.

As it can be seen in the Figure 6.21 contours of vertical stress obtained by the intelligent FE model have the correct general shape, however these contours are not at exact locations along the centre line beneath the footing as those obtained from the charts. This is due to the fact that in EPR based FE model the elasticity modulus is not constant for the entire analysis but it is updated at each load increment and separately for every single point in the mesh based on the results of previous increment.

A comparison between the EPR-FEM results and the results of the design charts in terms of the locations of the contours beneath the centre line of the footing is also presented in Table 6.3. In addition, Figure 6.22 shows the amount of footing settlement obtained from the numerical analyses using three different constitutive models including Mohr-Coulomb, Duncan-Channg and EPRCM. From the results shown in Figure 6.22, it can be seen that the results are close. Perhaps it could be argued that the EPR-FEM results are more accurate as it does not require a priory any idealised constitutive relationship for the soil, as it is the case for the other two models, and it obtained the constitutive relationship directly from the actual experimental data.

**Table 6.3** Comparison of EPR-FEM and closed form results for circular footing.

Contour	EPR-FEM	Chart
0.9	0.38B	0.26B
0.8	0.46B	0.36B
0.7	0.53B	0.45B
0.6	0.63B	0.54B
0.5	0.79B	0.64B
0.4	0.95B	0.79B
0.3	1.04B	0.97B
0.2	1.29B	1.26B
0.1	1.94B	1.86



**Figure 6.22** Comparison of the results for surface settlement of the footing obtained from the intelligent FEM and conventional FE analyses.

### 6.3.2 Coupled stress-strain-flow analysis

The finite element formulation for coupled stress-strain-flow analysis, described in chapter 3, was implemented into the developed FE program in order to analyse the behaviour of saturated soils. Terzaghi's one dimensional consolidation problem was analysed with the developed program and the results of the analysis were compared to the analytical solution in order to verify the correct implementation of the coupled formulation in the intelligent FE program.

#### 6.3.2.1 Verification Example

Terzaghi one dimensional consolidation problem consists of a column of soil restrained from lateral displacement. In this problem, the deformation and flow of pore fluid occur in only one dimension. One dimensional consolidation conditions apply when a layer of saturated clay of uniform thickness is subjected to an instantaneous uniform surcharge,  $q$ , of large lateral extent. Terzaghi derived the following partial differential equation

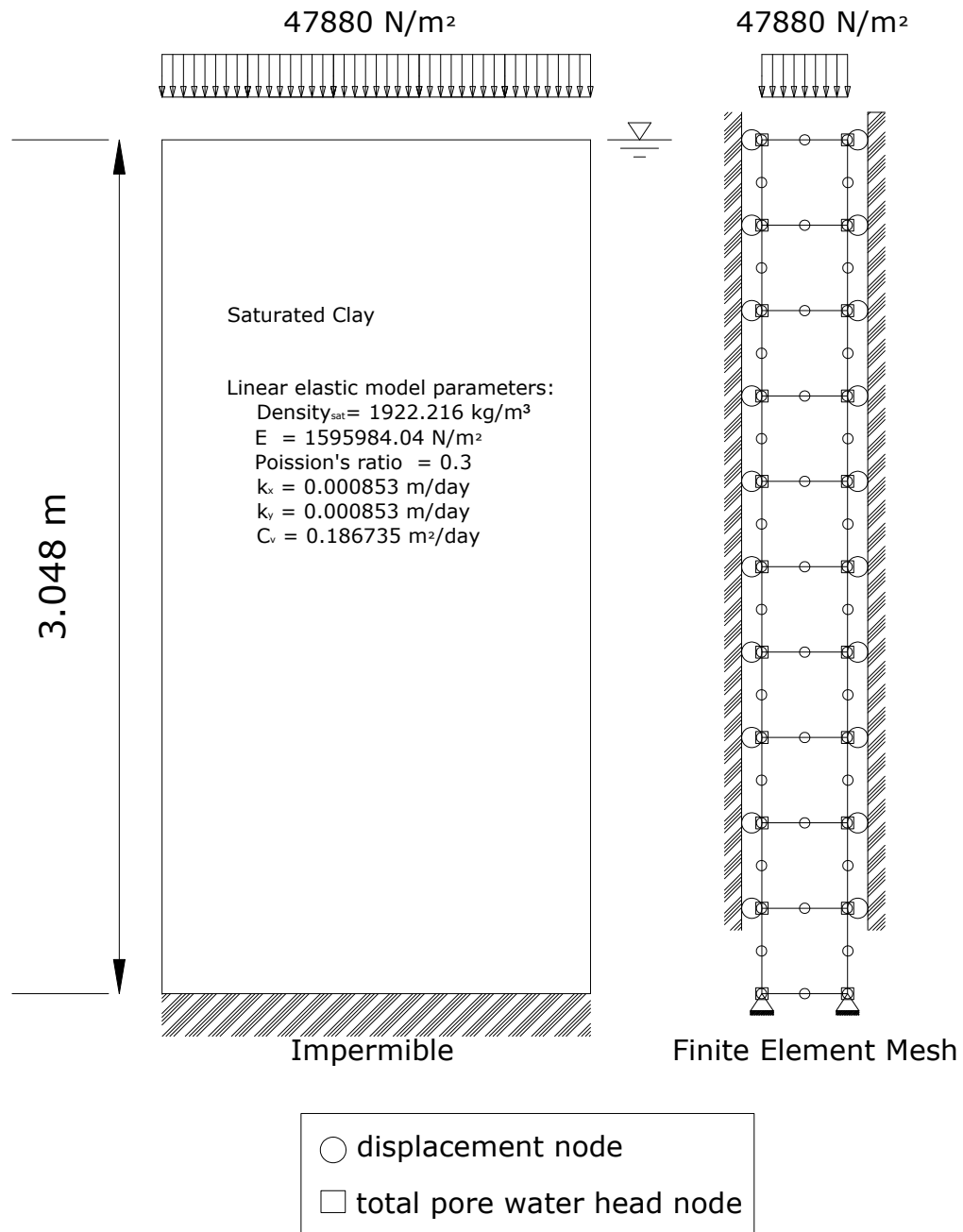
describing the dissipation of excess pore pressures and associated deformation of the soil in one dimensional consolidation

$$c_v \frac{\partial^2 u}{\partial z^2} = \frac{\partial u}{\partial t} \quad (6.9)$$

where  $c_v$  ( $L^2/T$ ) is the coefficient of consolidation which is determined by the compressibility and permeability of the soil,  $u$  ( $F/L^2$ ) is the excess pore pressure in the soil,  $z$  is depth of the point under consideration in the soil and  $t$  is time. It should be noted that the following assumptions were considered for the derivation of the above analytical solution for the one dimensional consolidation problem

- i) The soil medium is fully saturated.
- ii) The pore water in soil medium is incompressible.
- iii) The soil particles are incompressible.
- iv) The pore water flow through the soil medium is governed by Darcy's law.
- v) The soil material has linear elastic behaviour.

Figure 6.23 illustrates the geometry, boundary and initial conditions, material properties and finite element mesh for the validation problem. The finite element mesh for this problem includes 10 elements in which every element has eight nodes for displacement and four nodes for pore water pressure.

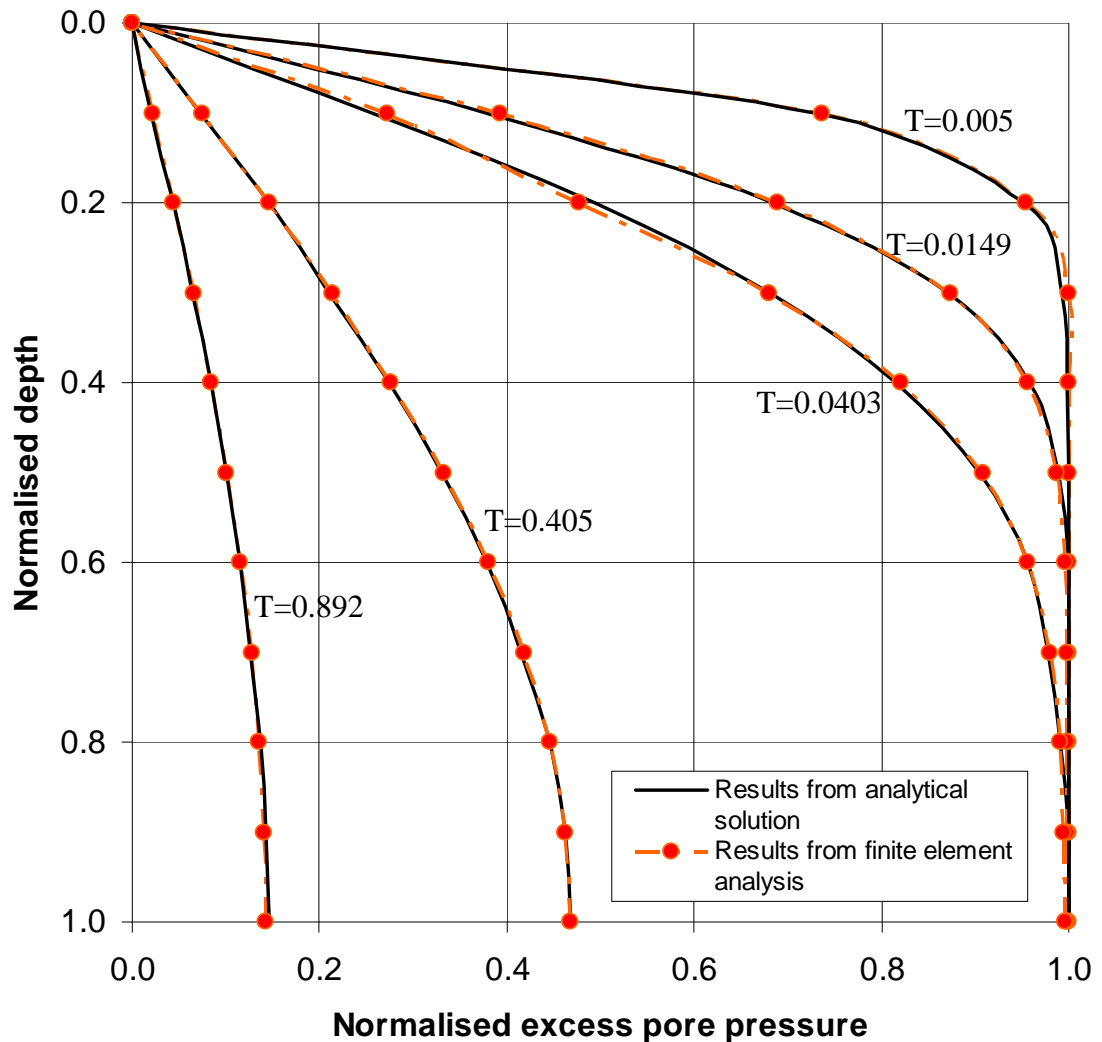


**Figure 6.23** Geometry of the verification problem and the FE mesh showing boundary conditions.

Figure 6.24 shows the plot of excess pore pressure changes with depth and time obtained from the results of FE analysis and compared with the results of Terzaghi's analytical solution. The results are shown at five different times which are represented by the dimensionless time factor,  $T$ , defined as

$$T = \frac{c_v t}{H^2} \quad (6.10)$$

The excess pore pressure is normalised by the magnitude of the traction,  $q$ , and the depth,  $z$ , is normalised by the length of the longest drainage path,  $H$ . As it is seen in Figure 6.24 the level of agreement between the results from FE analysis and analytical solution is excellent.



**Figure 6.24** Pore water pressure changes for the verification problem computed using analytical solution and finite element analysis.

### 6.3.2.2 Intelligent coupled analysis

In this example, the behaviour of the soil is studied in a coupled condition. An isotropically consolidated undrained triaxial compression test is simulated using the developed finite element code and the data for training and validation of the EPRCM are generated by repeating the test at different constant cell pressures and using the

Modified Cam Clay soil model (Roscoe and Burland 1968). The material parameters assumed for the soil are

$\lambda = 0.14$  (slope of the virgin consolidation line),

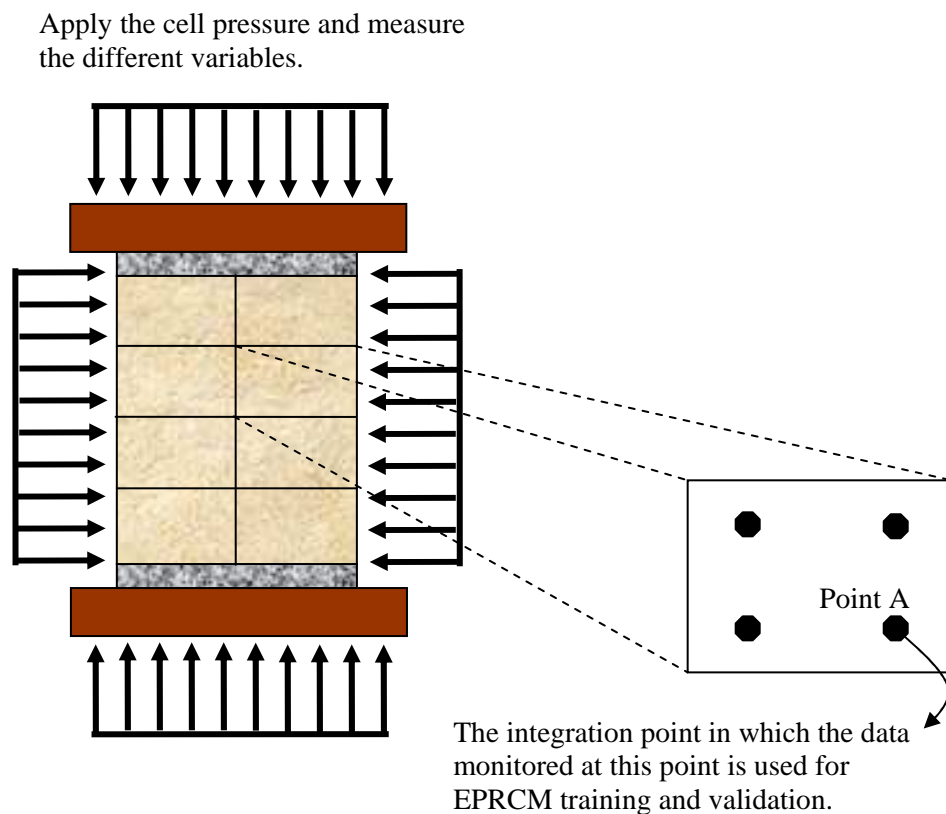
$\kappa = 0.015$  (slope of the unloading/reloading lines in the  $v:Ln p'$  space),

$M = 0.8$  (slope of the critical state line in the  $q:p'$  space),

$P_o = 100.0$  kPa (isotropic preconsolidation pressure indicating the size of the yield surface),

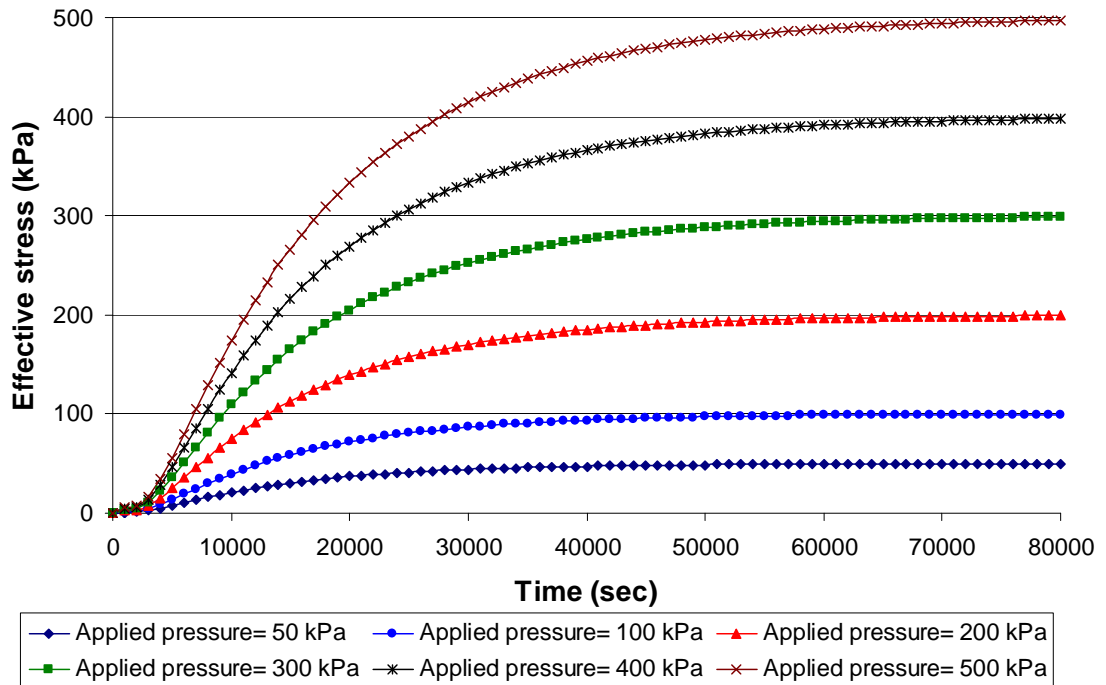
$N = 2.68$  (intercept of the virgin consolidation line with vertical axis).

The saturated specimen simulated for this test is 38 mm in diameter and 76 mm in height, similar to that of a typical triaxial sample. The finite element model is schematically shown in Figure 6.25.

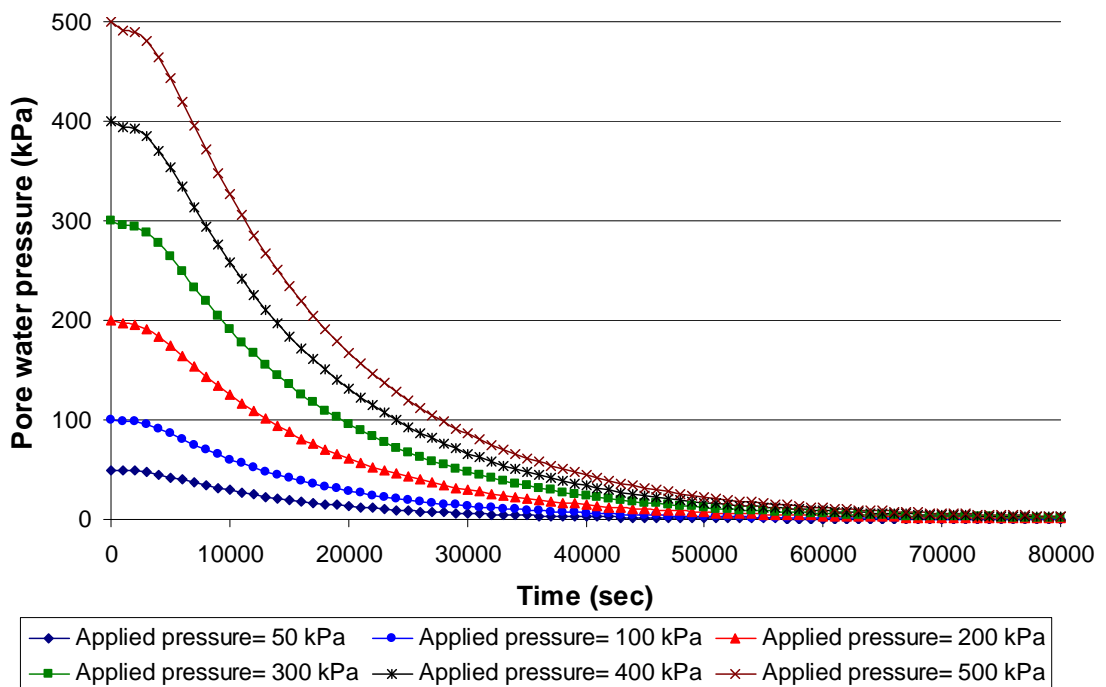


**Figure 6.25** A schematic view of the finite element model used to generate the required data for development of EPR based constitutive relationship. For each particular cell pressure the variations of different parameters including stresses, strains, pore water pressure, velocity, hydraulic gradient and displacement at the specific integration point A are monitored and used for EPRCM development.

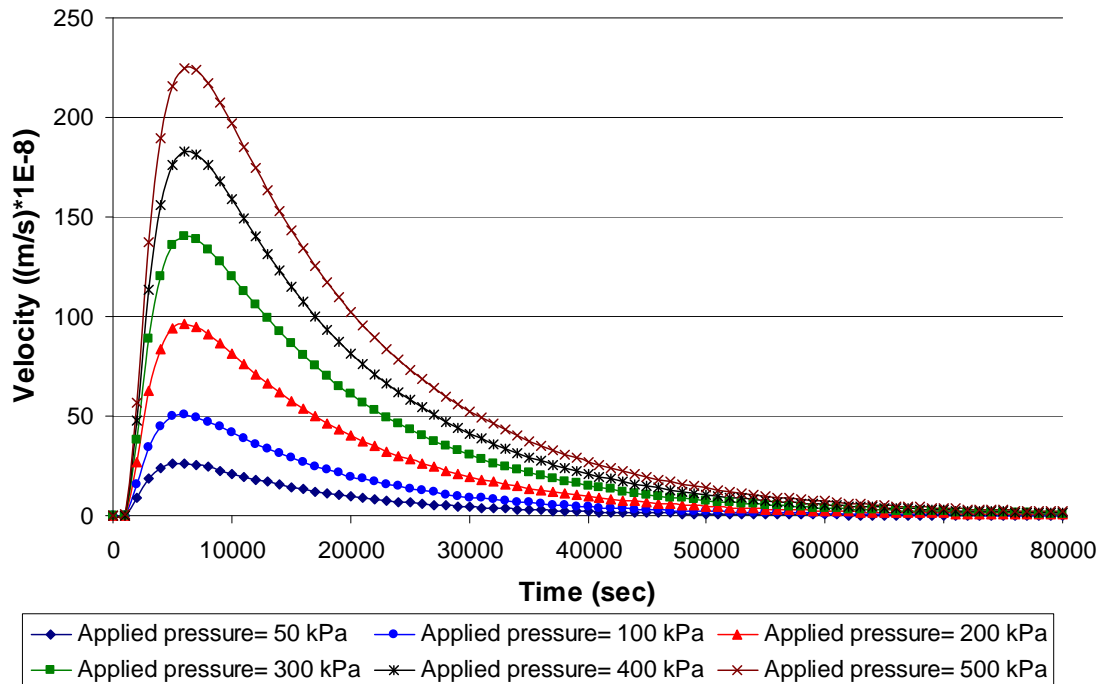
The data generated by numerical simulation of isotropically consolidated samples at different confining pressures of 50, 100, 200, 300, 400 and 500 kPa are shown in Figures 6.26 and 6.27.



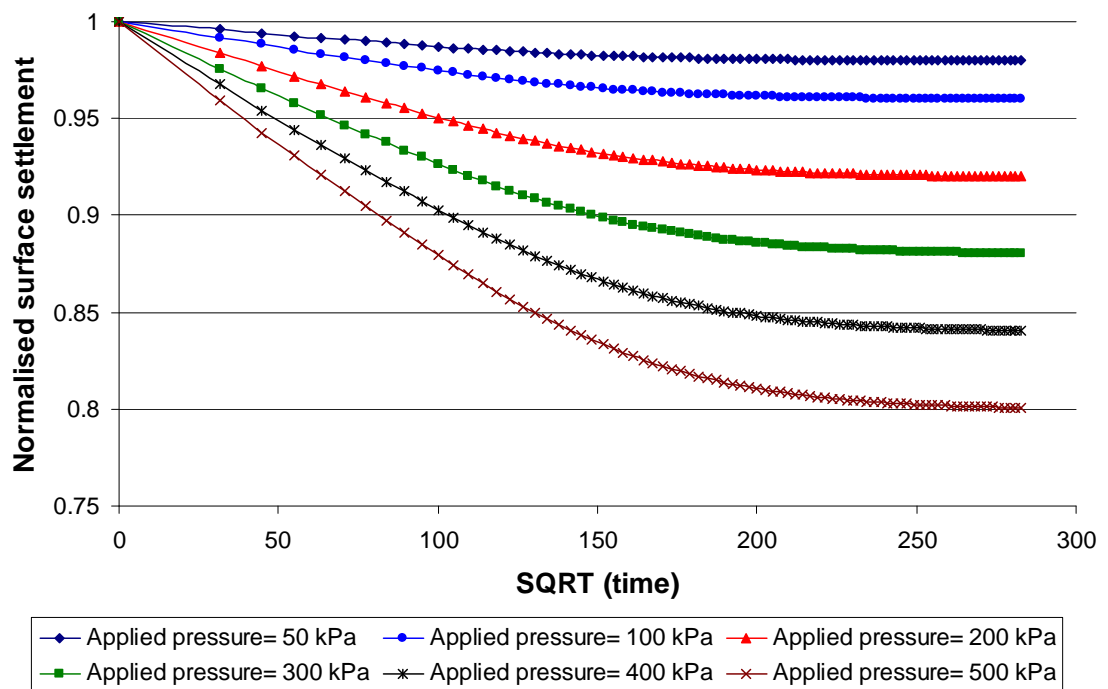
(a)



(b)



**Figure 6.26** The results of the numerical simulation of isotropic consolidation test showing (a) the variation of effective stress with time at different applied pressures, (b) the variation of pore water pressure with time at different applied pressures, and (c) the variation of velocity of flow with time at different applied pressures.



**Figure 6.27** The variation of surface settlement of the specimen (normalised) with square root of time at different applied pressures.

To examine the capability of EPR for analysing coupled conditions, the data from the simulated isotropic consolidation tests at all confining pressures, except the one under 300 kPa pressure, were used to generate the required evolutionary polynomial regression based relationships. Two different EPR models were considered with separate sets of input-output variables. In the first set, three variables including time ( $t$ ), confining pressure ( $\sigma$ ) and axial strain ( $\varepsilon_1$ ) were considered to predict the effective stress ( $\sigma'$ ) at each time. In the second model the first three input variables were the same as the first model, also it included two additional input variables as velocity of flow ( $v$ ) and pore water pressure ( $u$ ) at each time to predict the permeability ( $k$ ) of the soil under experiment. For each set, separate EPR models were developed among which the simplest and most ideal models were selected to be used in the finite element procedure. These models were obtained using EPR type 2, with no function and the range of exponents as  $[-1, -0.5, 0, 0.5, 1]$ . Also the single objective criterion was utilised for EPR training. The selected models were

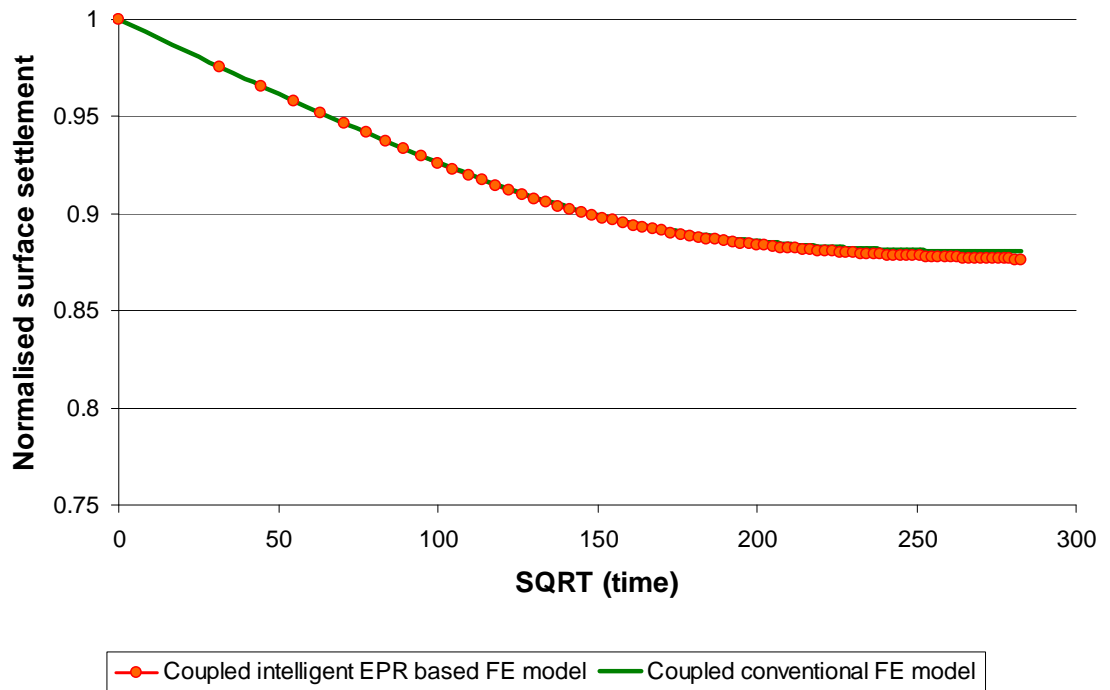
$$\sigma' = -3.8317E - 6t \sqrt{\sigma \varepsilon_1} - 0.05 \sigma + 4.12E - 4 \sigma \sqrt{t} + \frac{4.9642E - 12 \sigma \sqrt{\varepsilon_1}}{t} \quad (6.11)$$

$$k = 0.04038 \varepsilon_1 + 3.2945E - 5 u \sqrt{\sigma} - 1.4143E - 4 \sqrt{tv} - 0.084521 \sqrt{\sigma} + 0.026655 \sqrt{u} + 4.0981 \quad (6.12)$$

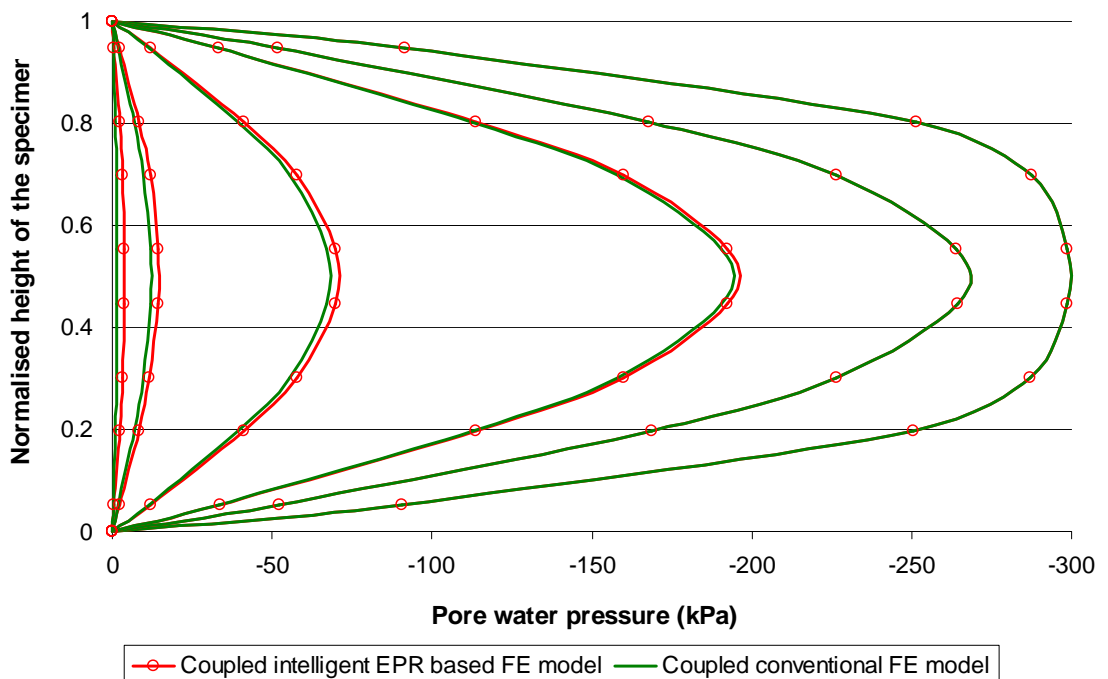
The selected EPR models for predicting effective stress and permeability were then incorporated into the intelligent finite element model. From Equation (6.11) the elastic modulus of the soil can be obtained by taking the derivative of this equation as

$$E_{EPR} = -\frac{7.6634E - 7t \sqrt{\sigma}}{\sqrt{\varepsilon_1}} + \frac{9.9284E - 13 \sigma}{t \sqrt{\varepsilon_1}} + 9.94 \quad (6.13)$$

The intelligent EPR based FE model was then used to simulate the behaviour of the soil at the confining pressure of 300 kPa which was not introduced to the EPR during training. Figures 6.28 and 6.29 show the comparison of the results obtained from the intelligent FE as well as the conventional FE using the modified Cam-Clay model.



**Figure 6.28** Comparison of the numerical results for the variations of the surface settlement with square root of time using intelligent EPR based and conventional FE models.



**Figure 6.29** Pore water pressure changes with time along the specimen computed using coupled conventional FE model and coupled intelligent EPR based FE model.

As it can be seen in Figures 6.28 and 6.29, excellent agreement between the finite element generated data and the results predicted from coupled intelligent FE code is achieved which highlights the potential of the developed technique in more complex situations.

## 6.4 Conclusions

An intelligent finite element method (EPR-FEM) has been developed based on the integration of an EPRCM in a finite element framework. In the developed methodology, the EPRCM is used as a substitute for the conventional constitutive models for the material. The efficiency and adaptability of the proposed method have been demonstrated by successful application to several boundary value problems. The results of the analysis have been compared to those obtained from conventional FE analyses using the Duncan-Chang nonlinear elastic model as well as Mohr-Coulomb (elastic-plastic) model. The result shows that EPRCM can be successfully implemented in a finite element model as an effective substitute for complicated conventional material models. As compared with traditional constitutive models, the EPR based constitutive model has several advantages including

- (i) It provides a unified approach to constitutive modelling of all materials.
- (ii) It provides a transparent and structured relationship of the constitutive behaviour of the material.
- (iii) It does not require any arbitrary choice of the constitutive (mathematical) model. The incorporation of an EPRCM in FE procedure avoids the need for complex yielding/plastic potential/failure functions, flow rules, etc. There is no need to check yielding, to compute the gradients of the plastic potential curve or to update the yield surface.
- (iv) There are no material parameters to be identified.
- (v) As EPR learns the material behaviour directly from raw experimental data the EPR based constitutive model is the shortest route from experimental research (data) to numerical modelling.
- (vi) The numerical parameters of the EPR based constitutive models are easily and automatically defined and EPRCM can be incorporated in a FE code in a

very natural manner. A trained EPRCM can be incorporated in a FE code/procedure in the same way as a conventional constitutive model. It can be incorporated either as incremental or total stress-strain strategies. An intelligent FE method can be used for solving boundary value problems in the same way as a conventional FEM.

- (vii) An additional advantage of EPRCM is that as more data becomes available, the material model can be improved by re-training the EPR.

# CHAPTER 7

## CONCLUSIONS AND RECOMMENDATIONS

### 7.1 Concluding remarks

An EPR based material modelling methodology, particularly for geomaterials, is developed in this study. The methodology is generic and can be extended for other types of materials, as well. With this material modelling methodology, the stress-strain behaviour of a material is captured within evolutionary based structured expressions trained directly on the stress-strain data obtained from experiments. The feasibility of this approach is validated through constitutive modelling of the behaviour of soils under drained and undrained conditions. Also the optimal EPR structure for training the soil behaviour is investigated. From the analysis and the performance of EPR based material models, it can be concluded that the use of EPR for the modelling of material behaviour is viable and promising. Such a method does not make a priori assumptions about the material behaviour, but rather bases its prediction of stress-strain behaviour on the experimental data with which it has been trained.

In addition, separate algorithms are proposed based on a feed-forward concept, for drained and undrained conditions for utilising the developed EPR based constitutive models as stand alone triaxial data generators. The simulators are successfully used in

predicting material behaviour over stress paths that have not been experienced by the model during training.

This thesis presents a fundamentally different approach to constitutive modelling of materials in finite element analysis. An intelligent finite element method is presented for modelling engineering problems. The method is based on the integration of an EPR based material model in the finite element procedure.

The methodology was first tested on two structural examples with linear and nonlinear behaviours. As for the engineering applications, it is essential that a FE analysis incorporating an EPRCM converges at each load increment (otherwise the EPRCM has no meaning). The application of the methodology particularly to the nonlinear structural example showed a concrete evidence that this would be viable and it was useful in forming the foundation of the methodology for application to geotechnical problems which have more complicated nonlinear behaviour. The feasibility of the developed intelligent FE model was then illustrated by successful applications to a number of geotechnical boundary value problems. For each case, the results of the analysis were compared with those obtained from conventional FE analyses using some of the most commonly used constitutive models. From the comparisons of the results, the following points can be highlighted

- i) The EPRCM based FE model is a robust methodology as for all cases it gives realistic results,
- ii) EPRCM can be of efficient use in FE analysis as the numerical solution using EPR based material model converges at all load increments (does not lead to impractical results),
- iii) As EPRCM represents the real nonlinear behaviour of the material, literally the results of the FE analyses incorporating an EPRCM should give more accurate predictions.

The Intelligent FE methodology was then extended to study stress-strain-flow behaviour in a coupled consolidation example. The data for this study was generated by finite element simulation of an isotropically consolidated triaxial compression test. Unlike neural networks, EPR can have just one output; therefore, two separate equations were trained for predicting elastic modulus and permeability parameters. Incorporating these

equations in the FE model, the results from intelligent numerical analysis of the sample under consolidation (at a confining pressure which was not introduced to the model during training) showed excellent agreement with the results of conventional FE simulation. This verification example provided a window of opportunity to expanding this methodology to more complicated conditions.

The main advantages of using an EPR based model are that it provides a unified approach to constitutive modelling of materials (i.e., every aspect of material behaviour can be implemented within a unified environment of an EPR equation); the incorporation of an EPR based constitutive model in a finite element procedure avoids the need for complex yielding/plastic potential/failure functions, flow rules, etc.; there is no need to check yielding, to compute the gradients of the plastic potential curve or to update the yield surface; there are no material parameters to be identified and the network is trained directly from experimental data.

The EPR is capable of learning the material behaviour directly from raw experimental data; therefore, EPRCM is the shortest route from experimental research (data) to numerical modelling. It was shown that even if the simplest EPRCM among several available EPR based material models (trained from a certain set of data) is selected for incorporating in FE, it can lead to reliable numerical results. This emphasises the effectiveness of the methodology whenever appropriate experimental data are available. Another advantage of evolutionary polynomial regression based constitutive model is that as more experimental data become available, the quality of the model prediction can be improved by learning from the additional data covering a wider range of stress paths; therefore, the material model can become more general and robust. It should be noted that, for practical problems, the data used for training the EPRCM should cover the range of stresses and strains that are likely to be encountered in practice. This is due to the fact that EPR, like any other artificial intelligence technique, is good at interpolation but not so good at extrapolation. Therefore, any attempt to use EPR based finite element model for loading conditions that may lead to stresses or strains outside the range of the stresses and strains used in training of the EPRCM may lead to unacceptable errors.

A properly trained EPRCM can be incorporated in a FE code in the same way as a conventional constitutive model. It can be incorporated either as incremental or total

stress-strain strategies. An intelligent FE method can be used for solving boundary value problems in the same way as a conventional FEM. It should be highlighted that although the work proposed here was mainly focused on soils, the methodology proposed is generic and could well be applied to other materials that have constitutive models that are difficult to accurately define in numerical codes used in other engineering disciplines.

## 7.2 Recommendations for further work

The EPR based approach to model the material behaviour in computational mechanics, provides an alternative way for modelling the complex behaviour of engineering materials including soils. As has been discussed in this study, there are many benefits to use an EPR based material model; however this work is a first step in developing intelligent EPR based constitutive models and more work can be done in the future. A list of further research needs arising out of this study are listed in the following

- The approach developed in this study can be extended to model the stress-strain behaviour of geomaterials under cyclic loading, depending on the existence of relevant experimental results, and an appropriate representation scheme can be designed.
- Considering the complex nature of some materials such as soil, rock, composites, etc., in the modelling of the behaviour of these materials the coupling between macro and micro properties may provide better understanding of the mechanisms that dictate the constitutive behaviour of these materials. This can be studied with the approach developed in this work; however experimental data on these aspects of such materials need to be compiled and processed. In addition, the approach can be extended to study the behaviour of other granular materials or any other engineering material e.g., biomaterials.
- The illustrative examples presented using the intelligent EPR based FE model, have not considered the possibility of predicting Poisson's ratio. The application of the proposed method in calculating this parameter should be investigated when the variation of this parameter is reflected in the experimental data.

## References

- Abu-Kiefa, M.A. (1998). "General regression neural networks for driven piles in cohesionless soils.", *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(12), 1177-1185.
- Akers, S.A. (2001). "Two-Dimensional Finite Element Analysis of Porous Geomaterials at Multikilobar Stress Levels.", Virginia Tech University, Blacksburg, Virginia.
- Amorosi, A., Millar, D.L., and Rampello, S. (1996). "On the Use of Artificial Neural Networks as Generic Descriptors of Geomaterial Mechanical Behaviour.", *Proceeding of the Internatopnal Symposiom on Prediction and Performance in Rock Mechanics and Rock Engineering, EUROCK '96, Torino, Italy*, 161-168.
- Arulmoli, K., Muraleetharan, K.K., Hosain, M.M. and Fruth, L.S. (1992). "VELACS Laboratory Testing Program, Soil Data Report.", The Earth Technology Corporation, Irvine, California, Report to the National Science Foundation, Washington D.C., March.
- Banimahd, M., Yasrobi, S.S., and Woodward, P.K. (2005). "Artificial neural network for stress-strain behavior of sandy soils: Knowledge based verification.", *Computers and Geotechnics*, 32(5), 377-386.
- Benardos, A.G. and Kaliampakos, D.C. (2004). "Modelling TBM performance with artificial neural networks.", *Journal of Tunnelling and Underground Space Technology*, 19(6), 597-605.
- Biot, M.A. (1941). "General theory of three-dimensional consolidation.", *Journal of Applied Physics*, 12.

- Caudill, M. (1991). "Neural network training tips and techniques.", *AI Expert*, 6(1), 56-61.
- Chandrupatla, T.R. and Belegundu, A.D. (1991). *Introduction to Finite Elements in Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Cheung, Y.K., Lo, S.H. and Leung, A.Y.T. (1996). *Finite Element Implementation*, Blackwell Science, Oxford.
- Daryl, L.L. (2002). *Finite Element Method*, Wadsworth Group, Brooks Cole.
- Dayhoff, J.E. (1990). *Neural Network Architectures: An Introduction*, Van Nostrand Reinhold, New York.
- Demuth, H. and Beale, M. (1996). *Neural network toolbox - for use with MATLAB*, The MATH WORK Inc.
- Desai, C.S. and Siriwardane, H.J. (1984). *Constitutive Laws for Engineering Materials with Emphasis on Geological Materials*, Prentice-Hall, Englewood Cliffs, NJ.
- Desai, C.S., Somasundaram, S. and Frantziskonis, G. (1986). "A Hierarchical Approach for Constitutive Modeling of Geologic Materials.", *International Journal for Numerical and Analytical Methods in Geomechanics*, 10(3), 225-257.
- Doglioni, A. (2004) "A novel hybrid evolutionary technique for environmental hydraulic modelling.", Ph.D. dissertation, Technical University of Bari, Italy.
- Doglioni, A., Giustolisi, O., Savic, D.A. and Webb, B.W. (2008) "An investigation on stream temperature analysis based on evolutionary computing.", *Hydrological Processes*, 22(3) 315-326.
- Drakos, S., Shin, H.S. and Pande, G.N. (2006). "Finite elements with artificial intelligence.", *Proceeding of the 2nd International Congress on Computational Mechanics and Simulation*, Guwahati, India, Paper No. 297.
- Duncan, J.M. and Chang, C.Y. (1970). "Non-linear analysis of stress and strain in soils.", *ASCE Journal of the Soil Mechanics and Foundation Division*, 96(SM5), 1629-1653.
- Einstein, H.H. and Hirschfeld, R.C. (1973). "Model studies on mechanics of jointed rock.", *ASCE Journal of the Soil Mechanics and Foundation Division*, 99(SM3), 229-248.
- Ellis, G., Yao, C. and Zhao, R. (1992). "Neural network modelling of mechanical behaviour of sand." *Proceedings of the 9th ASCE Conference on Engineering Mechanics*, Texas, 421-424.
- Ellis, G.W., Yao, C., Zhao, R. and Penumadu, D. (1995). "Stress-Strain Modeling of Sands using Artificial Neural Networks.", *ASCE Journal of Geotechnical Engineering Division*, 121(5), 429-435.

- 
- Elman, J.L. (1990). "Finding structure in time.", *Cognitive Science*, 14, 179–211.
- Flood, I. and Kartam, N. (1994). "Neural networks in civil engineering II principles and understanding.", *ASCE Journal of Computing in Civil Engineering*, 8(2), 149-162.
- Garrett, J.H. (1994). "Where and why artificial neural networks are applicable in civil engineering.", *Journal of Computing in Civil Engineering*, 8(2), 129-130.
- Ghaboussi, J. and Wilson, E.L. (1973). "Flow of compressible fluid in porous elastic media.", *International Journal for Numerical Methods in Engineering*, 5, 419–442.
- Ghaboussi, J., Carret, J. and Wu, X. (1990). "Material modelling with neural networks.", *Proceedings of the international conference on numerical methods in engineering: theory and applications*, Swansea, UK, 701-717.
- Ghaboussi, J., Carret, J. and Wu, X. (1991). "Knowledge-based modelling of material behaviour with neural networks.", *Journal of Engineering Mechanics Division*, 117(1), 132-153.
- Ghaboussi, J., Lade, P.V. and Sidarta, D.E. (1994). "Neural network based modelling in geomechanics.", *Proceedings of the 8th International Conference on Computer Methods and Advances in Geomechanics*, Morgantown, WV, 153-164.
- Ghaboussi, J., Pecknold, D.A., Zhang, M. and Haj-Ali, R.M. (1998). "Autoprogressive training of neural network constitutive models.", *International Journal for Numerical Methods in Engineering*, 42(1), 105-126.
- Ghaboussi, J. and Sidarta, D.E. (1998). "New nested adaptive neural networks (NANN) for constitutive modeling.", *Computers and Geotechnics*, 22(1), 29-52.
- Giustolisi, O. (2002). "Some techniques to avoid overfitting of artificial neural networks.", *Proceeding of the 5th International Conference of Hydroinformatics*, Cardiff, UK, 1465-1477.
- Giustolisi, O. and Savic, D.A. (2006). "A symbolic data-driven technique based on evolutionary polynomial regression.", *Journal of Hydroinformatics*, 8(3), 207-222.
- Giustolisi, O., Doglioni, A., Savic, D.A. and Webb, B.W. (2007) "A multi-model approach to analysis of environmental phenomena.", *Environmental Modelling and Software*, 22(5), 674-682.
- Goh, A.T.C. (1996). "Pile driving records reanalyzed using neural networks.", *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 122(6), 492-495.
- Golub, G.H. and Van Loan, C.F. (1993). *Matrix Computations*, The Johns Hopkins University Press Ltd., London, UK.
- Habibagahi, G. and Bamdad, A. (2003). "A neural network framework for mechanical behavior of unsaturated soils.", *Canadian Geotechnical Journal*, 40(3), 684-693.

- Haj-Ali, R.M., Pecknold, D.A., Ghaboussi, J. and Voyiadjis, G.Z. (2001). "Simulated micromechanical models using artificial neural networks.", *Journal of Engineering Mechanics*, 127(7), 730–738.
- Hashash, Y.M.A., Jung, S. and Ghaboussi, J. (2004). "Numerical implementation of a neural network based material model in finite element analysis.", *International Journal for Numerical Methods in Engineering*, 59, 989-1005.
- Huebner, K.H., Dewhurst, D.L., Smith, D.E. and Byrom, T.G. (2001). *The Finite Element Method for Engineers*, John Wiley and Sons, New York.
- Javadi, A.A. (2006). "Prediction of air losses in compressed air tunnelling using neural network.", *Journal of Tunnelling and Underground Space Technology*, 21, 9-20.
- Javadi, A.A. and Rezaia, M. (2006). "A new genetic programming-based evolutionary approach for constitutive modeling of soils.". *Proceeding of the 7th World Congress on Computational Mechanics*, Los Angeles, California, USA.
- Javadi, A.A. and Rezaia, M. (2008). "A new approach to data-driven modeling in geotechnical engineering.", *International Journal of Geomechanics and Geoengineering*, (Under Review).
- Javadi, A.A., Rezaia, M. and Nezhad, M.M. (2007). "A new approach to data-driven modeling in civil engineering.", *Proceeding of the 1st International Conference on Digital Communications and Computer Applications*, Amman, Jordan, 19-22 March, 2007, 17-24.
- Javadi, A. A., Rezaia, M. and Nezhad, M.M. (2006). "Evaluation of liquefaction induced lateral displacements using genetic programming.", *Computers and Geotechnics*, 33(4-5), 222-233.
- Javadi, A.A., Tan, T.P. and Elkassas, A. S. I. (2004a). "An intelligent finite element method.", *Proceeding of the 11th International EG-ICE Workshop*, Weimar, Germany, 16-25.
- Javadi, A.A., Tan, T.P. and Elkassas, A.S.I. (2005). "Intelligent finite element method.", *Proceeding of the 3rd MIT Conference on Computational Fluid and Solid Mechanics*, Cambridge, Massachusetts, USA.
- Javadi, A.A., Tan, T.P., Elkassas, A.S.I. and Zhang, M. (2004b). "Intelligent finite element method: Development of the algorithm.", *Proceeding of the 6th World Congress on Computational Mechanics-WCCM VI*, Beijing, China.
- Javadi, A.A., Tan, T.P. and Zhang, M. (2003). "Neural network for constitutive modelling in finite element analysis.", *Computer Assisted Mechanics and Engineering Sciences*, 10, 375-381.
- Javadi, A.A., Zhang, M. and Tan, T.P. (2002). "Neural network for constitutive modelling of material in finite element analysis.", *Proceedings of the 3rd International Workshop/Euroconference on Trefftz Method*, Exeter, UK, 61-62.

- Juang, C.H. and Chen, C.J. (1999). "CPT-based liquefaction evaluation using artificial neural networks.", *Computer-Aided Civil and Infrastructure Engineering*, 14(3), 221-229.
- Juang, C.H., Jiang, T. and Christopher, R.A. (2001). "Three-dimensional site characterisation: neural network approach.", *Geotechnique*, 51(9), 799-809.
- Jung, S. and Ghaboussi, J. (2006). "Neural network constitutive model for rate-dependent materials,", *Computers and Structures*, 84(15-16), 955-963.
- Kawamoto, T., Ichkawa, Y. and Kyoya, T. (1988). "Deformation and fracturing behavior of discontinuous rock mass and damage mechanics theory.", *International Journal for Numerical and Analytical Methods in Geomechanics*, 12(1), 1-30.
- Klaus, J.B. (1996). *Finite Element Procedures*, Prentice-Hall, Upper Saddle River, New Jersey.
- Koza, J.R. (1992). *Genetic Programming: on the Programming of Computers by Natural Selection*, Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA.
- Lade, P.V. and Duncan, J.M. (1975). "Elastoplastic stress-strain theory for cohesionless soil.", *ASCE Journal of Geotechnical Engineering Division*, 101(GT10), 1037-1053.
- Lamb, T.W. and Whitman, R.V. (1979). *Soil Mechanics*. John Wiley and Sons, New York.
- Lefik, M. and Schrefler, B.A. (2003). "Artificial neural network as an incremental non-linear constitutive model for finite element code.", *Computer Methods in Applied Mechanics and Engineering*, 192, 3265-3283.
- Lewis, R.W. and Schrefler, B.A. (1987). *The finite element method in the deformation and consolidation of porous media*, John Wiley and Sons, Chichester, UK.
- Lewis, R.W. and Schrefler, B.A. (1998). *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*, John Wiley and Sons, Chichester, UK.
- Lippmann, R.P. (1987). "An introduction to computing with neural nets.", *IEEE Acoustics, Speech and Signal Processing Magazine*, 4(2), 4-22.
- Logar, J. and Turk, G. (1997). "Neural network as a constitutive model of soil.", *Zeitschrift für Angewandte Mathematik und Mechanik*, 77(S1), 195-196.
- Lu, M., AbouRizk, S.M. and Hermann, U.H. (2001). "Sensitivity analysis of neural networks in spool fabrication productivity studies.", *ASCE Journal of Computing in Civil Engineering*, 15(4), 299-308.
- Lu, P. and Rosenbaum, M.S. (2003). "Artificial neural networks and grey systems for the prediction of slope stability.", *Natural Hazards*, 30(3), 383-395.

- Marsono A.K. (2000). "Reinforced Concrete Shear Wall With Regular Staggered Openings. ", Ph.D. dissertation, Volume 1, University of Dundee, UK.
- Millar, D.L. and Calderbank, P.A. (1995). "On the investigation of multilayer feedforward neural network model of rock deformability behaviour.", Proceeding of The 8th International Congress on Rock Mechanics, Tokyo, Japan, 933-938.
- Millar, D.L. and Clarici, E. (1994). "Investigation of back-propagation artificial neural networks in modelling the stress-strain behaviour of sandstone rock.", Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 3326-3331.
- Najjar, Y.M. and Basheer, I.A. (1996). "Discussion of stress-strain modeling of sands using artificial neural networks.", ASCE Journal of Geotechnical Engineering Division, 122(11), 949-950.
- Najjar, Y.M., Basheer, I.A. and Ali, H. E. (1999). "On the use of neuronets for simulating the stress-strain behavior of soils.", Proceeding of the 7th International Symposium on Numerical Models in Geomechanics-NUMOG VII, Graz, Austria, 657-662.
- Owen, D.R.J. and Hinton, E. (1980). Finite Elements in Plasticity: Theory and Practice, Pineridge Press, Swansea.
- Penumadu, D. and Chameau, J.L. (1997). "Geomaterial modelling using neural networks.", Artificial Neural Networks for Civil Engineering: Fundamentals and Applications, N. Kartman, I. Flood, and Garrett, eds., ASCE, 160-184.
- Penumadu, D. and Zhao, R. (1999). "Triaxial compression behavior of sand and gravel using artificial neural networks.", Computers and Geotechnics, 24(3), 207-230.
- Penumadu, D., Zhao, R. and Frost, D. (2000). "Virtual geotechnical laboratory experiments using a simulator.", International Journal for Numerical and Analytical Methods in Geomechanics, 24(5), 439-451.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). Numerical recipes, Cambridge University Press, UK.
- Rahman, M.S., Wang, J., Deng, W. and Carter, J.P. (2001). "A neural network model for the uplift capacity of suction caissons.", Computers and Geotechnics, 28(4), 269-287.
- Rezania, M. and Javadi, A.A. (2006) "Application of evolutionary programming techniques in geotechnical engineering.", Proceeding of the 6th European Conference on Numerical Methods in Geotechnical Engineering, Schweiger H.F. (ed.), Graz, Austria, 677-682.
- Rezania, M. and Javadi, A.A. (2007). "A new genetic programming model for predicting settlement of shallow foundations.", Canadian Geotechnical Journal, 44(12), 1462-1473.

- Rezania, M. and Javadi, A.A. (2008a) "Predicting settlement of shallow foundations using evolutionary polynomial regression.", *Computer-Aided Civil and Infrastructure Engineering*, (Under Review).
- Rezania, M. and Javadi, A.A. (2008b); "Settlement prediction of shallow foundations; a new approach.", *Proceeding of the 2nd British Geotechnical Association International Conference on Foundations, Dundee, Scotland, Paper No. 539*.
- Rezania, M., Javadi, A.A. and Giustolisi, O. (2008a) "An evolutionary-based data mining technique for assessment of civil engineering systems.", *Journal of Engineering Computations*, (Accepted, in print).
- Rezania, M., Javadi, A.A. and Giustolisi, O. (2008b) "Evaluation of liquefaction potential based on CPT results using evolutionary polynomial regression.", *Computers and Geotechnics*, (Under Review).
- Roscoe, K.H. and Burland, J.B. (1968). "On the generalised stress-strain behaviour of "wet" clay". *Engineering Plasticity*, (Eds.: Heyman J. and Leckie F.A.), Cambridge University Press.
- Roscoe, K.H. and Schofield, A.N. (1963). "Mechanical behaviour of an idealized 'wet' clay.", *Proceeding of the 2nd European Conference on Soil Mechanics and Foundation Engineering, Wiesbaden, Germany*, 47-54.
- Rumelhart, D.E., Hinton, G.E. and Williams, R. J. (1986). "Learning representations by back-propagating errors.", *Nature*, 323, 533–536.
- Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1994). *Learning Internal Representation by Error Propagation in Parallel Distributed Processing*, Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA.
- Sandhu, R.S. and Wilson, E.L. (1969). "Finite-element analysis of seepage in elastic media.", *ASCE Journal of the Engineering Mechanics Division* 95(3), 641-652.
- Sette, S. and Boullart, L. (2001). "Genetic Programming: Principles and Applications.", *Engineering Applications of Artificial Intelligence*, 14(6), 727-736.
- Shin, H.S. (2001). "Neural network based constitutive models for finite element analysis.", Ph.D. dissertation, University of Wales Swansea, UK.
- 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. (2001). "Intelligent finite elements." *Proceeding of Asian-Pacific Conference for Computational Mechanics-APCOM 01, Sydney, Australia*, 1301-1310.
- Shin, H.S. and Pande, G.N. (2002). "Enhancement of data for training neural network based constitutive models for geomaterials.", *Proceeding of The 8th International Symposium on Numerical Models in Geomechanics-NUMOG VIII, Rome, Italy*, 141-146.

- Sidarta, D.E. and Ghaboussi, J. (1998). "Constitutive modeling of geomaterials from non-uniform material tests.", *Computers and Geotechnics*, 22(1), 53-71.
- Simon, B.R., Wu, J.S.-S. and Zienkiewicz, O.C. (1986a). "Evaluation of u - w and u - finite element methods for the dynamic response of saturated porous media using one-dimensional models.", *International Journal for Numerical and Analytical Methods in Geomechanics*, 10(5), 461-482.
- Simon, B.R., Wu, J.S.-S., and Zienkiewicz, O.C. (1986b). "Evaluation of higher order mixed and hermitean finite element procedures for dynamic analysis of saturated porous media.", *International Journal for Numerical and Analytical Methods in Geomechanics*, 10(5), 483-499.
- Stasa, F.L. (1985). *Applied Finite Element Analysis for Engineers*, The Dryden Press, New York.
- Tikhonov, A.N. (1963). "Solution of incorrectly formulated problems and the regularization method.", *Doklady Akademii Nauk SSSR*, 151, 501-504.
- Toll, D.G. (1988). "The behavior of unsaturated compacted naturally occurring gravel," University of London, London, UK.
- Widham, L.E. and Loparo, K.A. (1989) "Artificial intelligent simulation and modeling: critical survey.", In: *Artificial intelligence, simulation and modeling*, (Eds: Widham, L.E., Loparo, K.A. and Nielsen, N.R.), John Wiley and Sons, New York, 1-45.
- Wu, H.C., Zhang, X.B., Bao, T. and Al-Jibouri, S.H. (2001). "Modeling the stress-strain relation for granite using finite element-neural network hybrid algorithms.", *Proceeding of the 10th International Conference on Computer Methods and Advances in Geomechanics*, Tucson, Arizona, 241-245.
- Youd, T.L., Hansen, C.M. and Bartlett, S.F. (2002). "Revised multilinear regression equations for prediction of lateral spread displacement.", *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 128(12), 1007-1017.
- Zhu, J.-H., Zaman, M.M. and Anderson, S.A. (1998a). "Modeling of soil behavior with a recurrent neural network.", *Canadian Geotechnical Journal*, 35(5), 858-872.
- Zhu, J.-H., Zaman, M.M. and Anderson, S.A. (1998b). "Modeling of shearing behavior of a residual soil with recurrent neural network.", *International Journal for Numerical and Analytical Methods in Geomechanics*, 22(8), 671-687.
- Zienkiewicz, O.C. (1985). "Transient analysis - some recent developments.", *Proceeding of the International Conference on Numerical Methods in Engineering: Theory and Applications- NUMETA 85*, Swansea, UK, 3-8.
- Zienkiewicz, O.C., Chang, C.T. and Bettess, P. (1980). "Drained, undrained, consolidating and dynamic behavior assumptions in soils.", *Geotechnique*, 30(4), 385-395.