Developing constitutive models from EPR-based self-learning finite element analysis

Faramarzi, Asaad; Javadi, Akbar A.

DOI:
10.1002/nag.2747

Developing constitutive models from EPR-based self-learning finite element analysis

Ali Nassr [a], Akbar Javadi [a, *], Asaad Faramarzi [b]

[a] Department of Engineering, University of Exeter, North Park Road, Exeter, EX4 4QF
[b] Department of Civil Engineering, School of Engineering, University of Birmingham, Edgbaston, Birmingham, B15 2TT

Abstract

A constitutive model that captures the material behaviour under a wide range of loading conditions is essential for simulating complex boundary value problems. In recent years, some attempts have been made to develop constitutive models for finite element analysis using self-learning simulation (SelfSim). Self-learning simulation is an inverse analysis technique that extracts material behaviour from some boundary measurements (e.g., load and displacement). In the heart of the self-learning framework is a neural network which is used to train and develop a constitutive model that represents the material behaviour. It is generally known that neural networks suffer from a number of drawbacks. This paper utilizes evolutionary polynomial regression (EPR) in the framework of self-learning simulation within an automation process which is coded in Matlab environment. EPR is a hybrid data mining technique that uses a combination of a genetic algorithm and the least square method to search for mathematical equations to represent the behaviour of a system. Two strategies of material modelling have been considered in the self-learning simulation-based finite element analysis. These include a total stress-strain strategy applied to analysis of a truss structure using synthetic measurement data and an incremental stress-strain strategy applied to simulation of triaxial tests using experimental data. The results show that effective and accurate constitutive models can be developed from the proposed EPR-based self-learning finite element method. The EPR-based self-learning FEM can provide accurate predictions to engineering problems. The main advantages of using EPR over neural network are highlighted.

Keywords: finite element, self-learning simulation, data mining, evolutionary techniques

---

*Corresponding author. Tel: +44 1392 723640

E-mail address: *A.A.Javadi@exeter.ac.uk (Akbar Javadi)*

1 - Introduction

Numerical methods such as finite element method (FEM) play an important role in solving many engineering problems. One of the main components of the finite element analysis is the constitutive model which is used to represent the behaviour of materials at the point or element level. In their basic form, constitutive models describe the stress-strain relationship. The successful application of finite element simulations in engineering problems is largely dependent on the choice of an appropriate constitutive model that represents the material behaviour. For large scale applications, significant attention should be paid to efficient incorporation of constitutive models within the finite element analysis to improve convergence behaviour and reduce time of analysis [1].

In the last two decades, with developments in computational software and hardware, the field of constitutive modelling has been extended beyond the classical elastoplastic theories, to computer aided pattern recognition approaches which have been introduced for modelling of a wide range of engineering problems. A number of data driven techniques such as artificial neural network (ANN), genetic programing (GP) and fuzzy logic have been used to model material behaviour. In particular, ANNs have been widely used to model the constitutive behaviour of various materials. In recent years, some research has been done on the development of constitutive models using another data mining technique, the evolutionary polynomial regression (EPR) and their implementation in the finite element method [2, 3].

The inverse analysis technique has also been successfully applied to extract material behaviour for various geotechnical problems using neural network [4]. Self-learning simulation (SelfSim) has been developed to link experimental testing and numerical modelling of soils [4]. Although there has been valuable research on the self-learning FEM using ANN and the demonstration of the advantages that ANN offers in constitutive modelling, it is generally known that ANNs also suffer from a number of drawbacks. For example, when using ANNs the number of neurons, number of hidden layers, transfer function, etc. must be determined a priori, requiring a time-consuming trial and error procedure. Moreover, the black box nature, the large complexity of the network structure, and the lack of interpretability have prevented the ANNs from achieving their full potential [2, 3, 5, 6].

In this paper evolutionary polynomial regression (EPR) is used in the framework of self-learning FEM for modelling of material behaviour. The proposed method eliminates most of the drawbacks of neural network.

Two strategies are used to train the EPR based constitutive models within the self-learning algorithm: (i) total stress-strain strategy and (ii) incremental stress-strain strategy, using synthetic and experimental data respectively.

2 – Material constitutive modelling using artificial neural network

The concept of using artificial neural network (ANN) for constitutive modelling of different materials has been well developed over the last few decades. For the first time, Ghaboussi and Wu [7] proposed to use ANN for modelling the behaviour of concrete and composite materials. Since then, other researchers continued to apply this approach to modelling the behaviour of different materials including concrete, soils, rocks and etc. Sankarasubramanian and Rajasekaran [8] used experimental data to train an ANN model to represent a nonlinear hypoelastic behaviour of reinforced concrete structures. Ghaboussi et al., [9] presented a strategy to capture the highly nonlinear behaviour of sands using a backpropagation neural network. Ghaboussi and Sidarta [10] proposed nested adaptive neural network (NANN) to represent constitutive model for geomaterials and used it to construct models for drained and undrained behaviour of sands in triaxial tests. NANN takes advantage of the nested structure of the material test data, and represents it in the architecture of the neural network. Millar [11] demonstrated that using an ANN can provide new capabilities over a broad range of problem areas in rock mechanics and rock engineering. Keshavaraj et al. [12] attempted to describe the stress-strain behaviour of fabrics such as polyester, under biaxial strain conditions using ANN trained with experimental permeability data. These works have shown that ANN based constitutive models have the capability to capture and represent the nonlinear behaviour of materials. A neural network model is fundamentally different from the conventional constitutive models because ANN is trained directly using laboratory data to learn the material behaviour rather than on assumptions made to develop a constitutive mode [13]. Training an ANN with sufficient relevant information can generalize material behaviour to new loading conditions. The main advantage of ANN models over conventional material models is their ability to extract nonlinear and complex interaction between variables without the need to assume the basic form of the relationship between input and output variables. Among various types of ANN, multi-layer feed forward network is known to be the most appropriate to describe the nonlinear functions, and so far, has been the only type of neural network used to represent the material constitutive behaviour (e.g. [1]).

2-1 Neural network based finite element method

The implementation of neural network in finite element analysis has been presented in different ways by a number of researchers. Javadi et al. (e.g. [14]) used a neural network for constitutive modelling of complex materials. They developed an intelligent finite element method based on the incorporation of a back-propagation neural network in finite element analysis. The method was applied to a number of engineering problems and it was shown that ANNs can be efficient in capturing and representing the constitutive behaviour of complex materials. Furukawa and Hoffman [15] presented an algorithm to implement ANN into finite element analysis to describe monotonic and cyclic plastic deformation. They used two ANNs to learn the kinematic hardening and isotropic hardening behaviour of materials. After training the developed constitutive model was incorporated in a commercial FE code, MARC, using its user subroutine utility for material models. Haj-Ali and Kim [16] developed nonlinear ANN based constitutive models for fibre reinforced polymer (FRP). They used experimental data obtained from off-axis tension and compression tests performed with coupons cut from a monolithic composite plate. The results of the developed ANNs models showed good agreement with the experimental results, however, in case of compression the proposed model could generate much closer results than in case of tension. The developed ANN models were implemented as user defined material models for the composite plate in ABAQUS as the FEM engine. The comparison between the ANN-FE model with the experimental results showed that within a small range of strain, very good agreement was achieved, however, some diversion accrued as strain increased. Kessler [17] implemented an ANN based constitutive model in finite element analysis (using Abaqus) for prediction of rheological behaviour of Aluminium. The results demonstrated that the ANN model has a superior capability over conventional constitutive models to mirror experimental data.

In general, the implementation of any constitutive model in finite element analysis must provide the material stiffness matrix, also called Jacobian matrix J, as:

$$J = \frac{\partial(d\underline{\sigma_i})}{\partial(d\underline{\varepsilon_j})} \qquad (1)$$

where σ_i and ε_i are the vectors of stresses and strains respectively. Hashash et al., [1] addressed some of the issues related to the use of ANN based constitutive models in finite element analysis with a number of numerical examples. They defined the material stiffness matrix, required in incremental finite element analysis, as:

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\varepsilon_j} = \frac{\partial({}^{n+1}\sigma_i - {}^{n}\sigma_i)}{\partial^{n+1}\Delta\varepsilon_j} = \frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\Delta\varepsilon_j} \qquad (2)$$

In the above equation $^{n+1}$ refers to the next state of stresses and strains. The differentiation of equation (2) can lead to calculation the material stiffness matrix (Jacobian) which can provide efficient convergence of the global solution [1]. Other researchers used direct derivatives of ANN equation (equation 3) and suggested a procedure to determine the first order partial derivation of the ANN model (e.g. [18, 19]).

$$D_{NN} = \frac{\partial\sigma}{\partial\varepsilon} \qquad (3)$$

3 – Self-learning simulation methodology

The SelfSim methodology is an extension of the autoprogressive algorithm originally introduced by Ghaboussi et al. [9]. The auto-progressive approach is a technique used for training ANN-based constitutive model in which the extracted information from a global load- deformation response of a structural test is used as training data for neural network model. It is generally known that ANNs require large amount of data in order to capture and learn the material behaviour. Normally, having large amount of data from a single test on one sample is not possible. The Self-Sim approach is used to overcome this issue by using rich stress-strain data embedded in non-homogenous structural tests, to train the ANN models. The developed material model from this approach is extracted from an iterative non-linear finite element analysis of the test sample and gradually improves the stress-strain data for training the ANN [9]. Sidarta and Ghaboussi [20] applied the autoprogressive algorithm using a series of non-uniform experimental tests (traixial compression tests with end friction) on sandy soil with different densities. Nested adaptive neural networks were utilized in this work. The trained ANNs models were used in forward analysis of the traixial tests with end friction and implemented in a hypothetical test without end friction. The results showed that the trained ANNs models could learn the behaviour of sand very well in case of end frication and reasonably well without end friction. Shin and Pande [21] proposed a strategy to develop a self-learning finite element model using a neural network based constitutive model (NNCM). This methodology was similar to the one was proposed by Gahboussi et al. [9]. It was shown that the choice of the position of monitoring points could affect the training program and hence the convergence of the NNCM towards the standard solution. The position of the load was also changed in order to demonstrate that the neural network model had been sufficiently trained to be able to perform analysis of any boundary value problem in which the material law corresponded to the trained ANN model.

Hashash et al. [4, 22] and Jung et al. [23] described the SelfSim approach as an analysis framework for implementation and extension of the auto-progressive algorithm. The framework was built based on satisfying the conditions of equilibrium and compatibility. The SelfSim approach was applied to a number of geotechnical problems. Qingwei et al. [24] presented application of SelfSim to simulation of laboratory tests including a triaxial compression test and a triaxial torsional shear test. They showed that SelfSim is able to establish a direct link between laboratory testing and soil constitutive modelling and capture the soil behaviour under complex loading conditions. The developed model was used to predict the load-settlement behaviour of a simulated strip footing.

Jung et al., [23] addressed the limitations of neural network-based (SelfSim) models with predictions beyond the data that are used for training and suggested to add data from other sources such as field measurements and laboratory tests in order to improve the prediction capabilities of developed models. Hashash and Song [25] used the self-learning simulation (SelfSim) approach to extract soil constitutive behaviour. They demonstrated three different problems; a triaxial test with frictional loading plates, deformations due to a deep excavation and seismic site response from a downhole array. Although the developed models illustrated the ability to predict the soil behaviour in complex conditions with good accuracy, however, the authors noted that selecting the SelfSim and ANN parameters is an empirical task and requires personal experience. This can be reflected in the lack of interpretability of neural network models. Hashash et al. [26] presented a comparison of two different inverse techniques for learning the behaviour of deep excavations in urban environment. The first technique was an optimization method in which the material parameters of the hardening soil constitutive model of PLAXIS were optimized using a genetic algorithm (GA). The second method was Self-learning simulation (Self-Sim) using ANN based constitutive model which was used to extract the soil behaviour. They presented a comparison of the computed lateral deformations and surface settlements from both inverse analyses. Although the GA could find the optimal solution of the problem even with noisy error function, the main disadvantage of this method is the high computational cost. Sung-woo and Hashash [27] applied the self-learning algorithm on a direct shear test to generate a soil constitutive model to solve a deep excavation case study problem. The non-uniform stress-strain behaviour from very few consolidated undrained direct shear tests was extracted by using the self-learning framework. The results showed that the developed models can predict the global responses, such as vertical ground surface settlements and lateral wall deflections around deep excavation. The above works have shown that the self-learning approach based on ANN is a robust tool for developing constitutive material models and a direct link between experimental testing and numerical simulation [27]. In this paper, EPR is presented as an effective alternative to ANN in the self-learning algorithm that addresses the shortcomings of the ANNs. The efficiency of the developed method is illustrated by application to two boundary value problems.

4 – Evolutionary polynomial regression (EPR)

As mentioned above, the Self-learning simulation (SelfSim) has been shown to be very efficient in training of neural network-based constitutive models for finite element analysis with limited data and has been successfully applied to a number of geotechnical problems. Although this method is a major contribution in the development of constitutive models, the main disadvantages of ANNs (such as, the black box nature and the complexity of model structure) remain unresolved. In this paper, a new data mining technique (evolutionary polynomial regression, EPR) is introduced for constitutive modelling in the self-learning finite element method that addresses some of the shortcomings of ANNs. EPR is a new hybrid technique based on evolutionary computing, aimed to search for polynomial structures representing the behaviour of a system [28]. EPR implements numerical and symbolic regression to perform evolutionary polynomial structure. The algorithm utilizes polynomial structure to take advantage of their appropriate mathematical properties. The main idea of the EPR is to use evolutionary search for exponents of polynomial expressions by means of a genetic algorithm (GA). This allows an efficient search for explicit equations that represent the behaviour of a system and offers more control on the complexity of the structures generated [29]. A typical formulation of EPR expression can be stated as [28, 29]:

$$Y = \sum_{j=1}^{m} F(\mathbf{X}, f(\mathbf{X}), a_j) + a_0 \qquad (4)$$

where Y is the estimated vector of output of the system; a_j is a constant; F is a function constructed by the process; X is the matrix of input variables; f is a function defined by the user and m is the number of terms of expression excluding the bias term ($a_{\mathbf{0}}$) [28]. Genetic algorithm is utilized to select the useful input vectors from X to be integrated together. The building blocks of the structure of F are defined by the user based on understanding of the physical process. While the selection of feasible structures is done during an evolutionary process, the parameters a_j are determined by the least square method. The first step in identification of the model structure is to convert equation (4) into the following vector form [29].

$$Y_{Nx1}(\Theta, Z) = [\ I_{Nx1} \quad Z^j_{Nxm}] \ x \ [\ a_0 \quad a_1 \quad .... \quad a_m]^T = Z_{Nxd} \ x \ \Theta^T_{dx1} \quad (5)$$

where $Y_{Nx1}(\Theta, Z)$ is the least squares estimate vector of the N target values; Θ_{dx1} is the vector of d= m+1 parameters $a_{\mathbf{j}}$ and $a_{\mathbf{0}}$ (Θ^T is the transposed vector); Z_{Nxd} is a matrix generated by I (unitary vector) for bias $a_{\mathbf{0}}$, and m vectors of variables Z^j. For a fixed j variables Z_j are a product of the independent predictor vectors of inputs, X = <X_1X_2…. X_k>.

In general, EPR is a two-steps technique for constructing a mathematical model. In the first step, it searches for the best form of the function structure and in the second step, it uses the least square method to find the adjustable parameters of the symbolic structures. In this way, EPR algorithm searches for the best set of input combinations and related exponents simultaneously. The matrix of input parameters X is given as [28]:

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{1k} \\ X_{21} & X_{22} & X_{2k} \\ \dots & \dots & \dots \\ X_{N1} & X_{N2} & X_{NK} \end{bmatrix} = [\ X_1 \quad X_2 \quad X_3 \ldots\ldots\ldots X_k\ ] \quad (6)$$

where the k^{th} column of X represents the candidate variables for the j^{th} term of Equation (5). Therefore, the j^{th} term of Equation (5) can be written as

$$Z^j_{Nx1} = [\ (X_1)^{ES(j,1)} . (X_1)^{ES(j,2)} \ldots\ldots (X_k)^{ES(j,k)}\ ] \quad (7)$$

where, Zj is the j^{th} column vector whose elements are products of candidate-independent inputs and ES is a matrix of exponents. Therefore, the problem is to find the matrix ES_{kxm} of exponents the values of which can be within user-defined bounds. For example, if a vector of candidate exponents for variables (inputs) in X is selected to be EX [0, 2, 3] and m (the number of terms without bias is 4, and k is 3 (the number of candidate-independent variables/inputs), then polynomial regression problem is to find a matrix of exponents ES_{4x3} [28]. An example of such a matrix is given here:

$$ES = \begin{bmatrix} 0 & 2 & 3 \\ 0 & 2 & 2 \\ 2 & 3 & 0 \\ 2 & 2 & 0 \end{bmatrix} \quad (8)$$

If this matrix is substituted into Equation (7) the following set of expressions is obtained:

$$Z_1 = (X_1)^0 \,.\, (X_2)^2 .\, (X_3)^3 = X_2^2 \,.\, X_3^3$$

$$Z_2 = (X_1)^0 \,.\, (X_2)^2 .\, (X_3)^2 = X_2^2 \,.\, X_3^2$$

$$Z_3 = (X_1)^2 \,.\, (X_2)^3 .\, (X_3)^0 = X_1^2 \,.\, X_2^3$$

$$Z_4 = (X_1)^2 \,.\, (X_2)^2 .\, (X_3)^0 = X_1^2 \,.\, X_2^2$$

The equation (5) it can be written as:

$$Y = a_\circ + a_1 .\, Z_1 + a_2 .\, Z_2 + a_3 .\, Z_3 + a_4 .\, Z_4 = a_\circ + a_1 \,.\, X_2^2 \,.\, X_3^3 + a_2 \,.\, X_2^2 \,.\, X_3^2 + a_3 \,.\, X_1^2 \,.\, X_2^3 + a_4 \,.\, X_1^2 \,.\, X_2^2 \quad (9)$$

The presence of zero in the exponent matrix ensures the ability to exclude some of the inputs from the regression model. The modelling procedure of EPR starts from a constant mean of output values. By increasing the number of evolutions it gradually picks up the different parameters in order to construct equations representing the constitutive relationship.

In general, there are different objective functions used in EPR. The original EPR algorithm used a single objective function (fitness control) to explore the solution space while penalising complex model structures using some penalization techniques [28]. Although the single objective strategy has been applied in different applications, it has been shown to have some drawbacks. For example, its performance can considerably deteriorate with increasing the number of terms. Also the selection of the best model relies on the complexity of the model which requires user’s experience and sometimes more accurate models with less complexity could be missed.

To address these problems, a multi-objective genetic algorithm (MOGA) has been included in the EPR where at least two objectives are included so that one can control the fitness of the models and the other can control the complexity of structure of the models [29]. In this work a multi-objective strategy has been used to develop the EPR based constitutive models. More details of the EPR strategy can be found in e.g.[28–30].

The accuracy of the developed EPR models is calculated at each stage based on the coefficient of determination (CoD) i.e., the fitness function as:

$$CoD = 1 - \frac{\sum_n (Y_a - Y_b)^2}{\sum_n (Y_a - \frac{1}{N}\sum_n Y_a)^2} \quad (10)$$

where Y_a is the actual output value; Y_p is the EPR predicted value and N is the number of data points on which the CoD is computed. If the model fitness is not acceptable or other termination criteria (e.g., maximum number of generations or maximum number of terms) are not satisfied, the current model should go through another evolution in order to obtain a new model [28]. The flow diagram of EPR procedure is shown in Figure 1.

4 – 1 EPR based constitutive modelling

EPR has been proposed as an effective alternative to other types of data mining techniques such as neural network. An EPR based constitutive model provides a unified approach to constitutive modelling. It has many advantages in representing the constitutive behaviour of complex materials. For example, the incorporation a neural network based constitutive model (NNCM) in equation (2) could result in a set of equations with complex mathematical structure that would not provide the user with a meaningful relationship between the input and output parameters of the material model.

EPRCM is able to learn and extract the material behaviour directly from experimental data. Consequently, it is the shortest route from experiments to numerical modelling [2, 3]. Models developed by EPR are concise mathematical equations that give the user a good understanding of the effect of input variables on the predicted output. EPR was first used for environmental modelling by Dolglioni et al. [31]. Its application was then extended to a wide range of civil engineering problems including constitutive modelling (e.g. [2, 3, 5]). EPR has been used to model the complex behaviour of saturated and unsaturated soils and comparison with experimental data has shown a very close agreement. Results from a number of comparative studies have shown that the EPR models outperform the ANNs [32].

4-2 Implementation of EPR in finite element analysis

The methodology of incorporating EPR in finite element analysis was first presented by Javadi and Rezania [6]. They showed that a properly trained EPR based constitutive model (trained on experimental data) can be readily implemented in a FE model. Like NNCM, an EPR based constitutive model does not require complex yield function, plastic potential, failure function, flow rule, etc. There is no need to check yielding, calculate the gradients of the plastic potential curve and update the yield surface, etc. The EPR-based FEM methodology was applied to a number of boundary value problems and the results were compared to those obtained from FE analyses using conventional constitutive models [2, 6]. In addition, they highlighted that although the work was primarily focused on soils, the methodology can be applied to other materials that have complex constitutive behaviours. Faramarzi [3] presented the implementation of trained EPR models in FE analysis using ABAQUS (as the finite element engine) through its user defined material module (UMAT and VUMAT) which are used to update the stresses and provides the Jacobian for every increment in every integration point. They showed that it is possible to construct the material stiffness (Jacobian) matrix using partial derivatives of the trained EPR models. The EPR based Jacobian matrix was integrated in FE code and the EPR-based FEM was applied to a number of boundary value problems including 2D, 3D and cyclic loading analyses. Two strategies were successfully applied to train the EPR model: incremental strategy and total stress -strain strategy. The results from these analyses were compared with those obtained from conventional finite element method using Cam-Clay and Mohr-Coulomb models among others. The results have shown that an EPR-based constitutive model (EPRCM) can be implemented in a finite element model in the same manner as a conventional constitutive model, with several advantages [3].

4 -3 EPR based self-learning FEM

Faramarzi et al. [33] proposed a new approach for training of EPR model. This approach is similar to the auto-progressive training proposed by Ghaboussi et al. [9] and Hashash et al. (e.g. [22, 25]). However, the EPR based Self-Sim model has only been applied to relatively simple examples using hypothetical simulated data. Also, the development of this methodology was done more or less manually. This means that a significant time was consumed to develop a trained model. In this paper the self-learning FEM has been developed using ABAQUS as the finite element engine through its user material subroutine (UMAT) to implement the EPRCM in the FE code [34] . The multiobjective function in EPR was used and linking of ABAQUS with EPR was done in Matlab environment in a fully automated iterative loop. The full procedure of the EPR-based self-learning FEM is shown in Figure 2. The process starts by running two finite element analyses initialized with elastic model in parallel (FEA and FEB). The finite element A (FEA) simulates the behaviour of the structure under applied forces and determines stresses and strains at each integration point. The methodology assumes that, since the applied boundary forces are accurate and the equilibrium condition is satisfied, the computed stresses will be acceptable as approximation of actual stresses that are experienced throughout the test. However, the computed strains form this analysis could be considered as poor approximation of actual strains, due to the difference between the computed and measured displacements. In parallel, the finite element B (FEB) analyses the structure using the same initial elastic model in which the measured boundary displacements are imposed. The strains obtained from this analysis are assumed to be accurate approximation of the actual strains, whereas the stresses may be a poor approximation of the actual stresses due to the difference between the computed and measured boundary forces. The stresses obtained from FEA and the strains obtained from FEB are collected to form stress-strain pairs of data and used to retrain the ANN model. The analyses of the finite element models A and B and subsequent training of the EPR model construct the SelfSim learning cycle. The analyses of finite elements A and B are repeated and an EPR model is developed from the results which is updated at each iteration. Convergence is considered to be achieved when the results of both analyses (FEA and FEB) are matched. Each cycle of SelfSim that accomplishes the applied load is called a pass [24, 25]. More than one pass could be required to extract the accurate material behaviour by retraining of the EPR model.

4 – 4 Training strategy

There are two main strategies (total stress-strain strategy and incremental stress-strain strategy) that can be used to train ANN or EPR to generate a constitutive model representing the material behaviour. In the first strategy, strains are used as input and stresses as output while in the second strategy, the incremental values are employed to build up the constitutive models. There are several factors that should be taken into account in choosing the best strategy and specifying the input and output parameters to train the EPRCM. These include the source of data, the way the trained EPR is to be used, and the training strategy used [2, 3].

The total stress-strain strategy can be utilized for modelling of materials that are not path dependent. This algorithm has been applied to different boundary value problems [3, 9, 21].

The incremental strategy is more suitable for materials that are path-dependent and has been used as a technique for training ANN based constitutive models [9]. In this paper both strategies are adopted and each one is applied on an engineering application.

5 – Numerical applications

To verify the EPR-based self-learning algorithm, two examples are used by employing both training strategies. The first example is a truss structure subjected to a concentrated load. In this example, a set of synthetic data is used to develop the EPR model employing the total stress-strain strategy. In the second example, experimental data from a set of conventional traixial tests are used and the developed EPR-based self-learning FEM is utilized to simulate the traixial experiments. In this example, the incremental stress-strain scheme is used. In general, when using the EPR-based self-learning algorithm, a number of points should be specified first, including the number and location of the monitoring points that are used to measure the response of the structure at each load increment. Choosing the appropriate number and locations for the monitoring points plays an important role in guaranteeing accurate results for the boundary value problem [21]. In addition, all the data should be normalized within the range [0 1] for training of the EPR and denormalized when the best EPR model is chosen for next load increment. This process is implemented within the Matlab code.

5-1 Example one: Truss structure

A 2D truss structure with 13 axial force elements is considered in the first example. The geometry, boundary conditions and loading are shown in Figure 3. The truss is subjected to a concentrated load (100 KN) at node 3. The load–displacement data were generated using FE simulation using an elastic-plastic model with hardening (using tabulated data option) in ABAQUS. In this example, one monitoring point was enough to represent the response of the structure to the loading condition. The load and the corresponding displacement at node 3 (the monitoring point) were considered as the experimental measurements and used in the self-learning process. Two finite element models FEA and FEB were created and the self-learning process was initialized first with an elastic modulus of 3000 kPa. The total stress-strain strategy was employed in this example in which the values of axial stresses were considered as input and axial strains as output, $\sigma_{11} = \mathcal{F}(\varepsilon_{11})$. In the EPR settings, the maximum number of terms was set to 6 and the exponents were set to be in range of [0 1 2 3

4 5]. These settings were specified following a trial and error process of EPR runs. Before each run the training data were randomly shuffled (in the Matlab code) to ensure that the obtained EPR models were not biased towards a particular part of the training data. Furthermore, to reduce the required time for EPR training, duplicated data were removed. The applied load was divided into 10 increments and at each increment an EPR model was chosen based on the highest CoD and used for the next increment to derive the Jacobian matrix.

The final EPR model was:

$$\sigma_{11} = 73.38\,x10^5 \varepsilon_{11}^5 - 8.82\,x10^3\ \varepsilon_{11}^4 + 71.28\,x\,10^5\,\varepsilon_{11}^3 + 43.59x10^3\,\varepsilon_{11}^2 + 3x10^3\,\varepsilon_{11} + 0.053 \tag{11}$$

with CoD of 99.86%. It is shown in Figure 4 that during the self-learning procedure, the prediction capability of EPR was gradually improved towards the expected behaviour.

Convergence of the FEA and FEB was achieved after several cycles of self-learning (within a single pass) and the above EPR-based model was used for the analysis. To verify the developed EPR, the results of load and displacement of point (n_3) in the EPR-based SelfSim model and the original model were compared. It can be seen from Figure 5 that the developed EPR model is able to predict the deformation of the truss with one pass of self-learning with very good accuracy within both elastic and plastic regions.

5- 2 Example two: Traixial experiment

The main target of the self-learning algorithm is to develop a constitutive model that is trained directly from experimental or field data and is used to predict the behaviour of other structures with the same material under different loading conditions. In this example the behaviour of a clay (kaolin) in a traixial experiment is analyzed under consolidated drained (CD) conditions. The experimental data reported in Cekerevac and Laloui [35] were used as the measurement data for the self-learning algorithm.

The incremental stress-strain strategy was employed in this example in which invariants of stresses and strains were used for training. Generally, the constitutive relationship is given in the form of δσ = Dδε [36], where (D) is material stiffness (or Jacobian) matrix. This matrix can be expressed in terms of modulus of elasticity (E) and Poisson's ratio (μ). For the traixial tests, the parameters mean effective stress p'^{i}, deviator stress q^{i}, volumetric strain ε_v^i, axial strain ε_y^i and increment of axial strain $\Delta\varepsilon_y^i$ were chosen as input parameters corresponding to the current state of stresses and strains in a load increment i, while deviator stress q^{i+1} corresponding to the input increment of the axial strain $\Delta\varepsilon_y^i$ was used as the output parameter. The traixial test results on the clay (Cekerevac and Laloui, 2004) [35] presented the shear and volumetric behaviour of the soil sample. For traixial test conditions, due to the axisymmetric nature of the problem, these stresses and strains can be written as:

$$p' = (\sigma_1' + 2\sigma_3')/3 \tag{12}$$

$$q = \sigma_1' - \sigma_3' \tag{13}$$

$$\varepsilon_v = \varepsilon_y + 2\varepsilon_r \tag{14}$$

$$\varepsilon_y = (\varepsilon_q + \varepsilon_v)/2 \tag{15}$$

$$\varepsilon_q = 2(\varepsilon_y + \varepsilon_r)/3 \tag{16}$$

where σ_1' and σ_3' are the major and minor effective principle stresses, and ε_q and ε_r are the deviator and radial strains respectively. In order to build the Jacobian matrix, at each run an EPR-based model with highest CoD was chosen and the value of E was calculated as:

$$E = \frac{q^{i+1} - q^{i}}{\Delta\varepsilon_y^i}$$

(17)

while the value of μ was assumed to be 0.3 for simplicity. Six monitoring points were specified on the top of the sample, monitoring the vertical deformations. Figure 6 shows the FEA and FEB simulations with their boundary conditions. Experimental data from 6 triaxial tests conducted at different confining pressures from 100 to 600 kPa were used for training of EPR within the self-learning algorithm. Each confining pressure was applied individually and one EPR based model was developed to represent the soil behaviour for each confining pressure. The procedure started by assuming an initial value for Young's modulus E for the first run only, which is in the linear portion of the global stress-strain curve. The initial value of E was set for all confining pressures to 20 x10^3 kPa. Once the Jacobian matrix was constructed, it was implemented in Abaqus via its UMAT. The same procedure as described in Example 1 was followed for preparing the data for training of EPR models. The EPR settings for each confining pressure were specified by trial and error. For all confining pressures, the exponents were limited to the range [-1 0 1 2 3] and the maximum number of terms was set to 8. The input and output parameters were set as follows:

$$q^{i+1} = \mathcal{F}(\varepsilon_v^i, \varepsilon_y^i, \Delta\varepsilon_y^i, q^i, p'^i) \quad (18)$$

Figure 7 shows the actual data that were used to extract the pressure-displacement data as the measurement data (applied pressure and corresponding displacement) in FEA and FEB. The actual measurements were smoothed in order to have uniform data, to avoid the discrepancy of data points and to generate more data for better training of EPR. In the dataset, for the soils that exhibited softening behaviour, the data after the failure points were removed. Modelling of the softening behaviour introduces additional challenges in training of the EPR (or ANN) models which is outside the scope of the present work.

Six EPR models were developed with CoD values 99% to 100%. For example, the best EPR model after one self-learning pass for confining pressure of 600 kPa is as follows:

$$q^{i+1} = 180\,\Delta\varepsilon_y\,(0.13\varepsilon_v + 1)^3 - 153\,(0.13\varepsilon_v + 1)^3 - 1190\,\Delta\varepsilon_y\,(0.13\varepsilon_v + 1)^3\,(0.0048\,p - 2.88)^2 - 6.4\,X10^{-9}\varepsilon_y^2\,q^3 + \left(\frac{0.142\,p - 8.22}{\Delta\varepsilon_y}\right) + 0.233\,q^2 + 1.002\,q - 86.21\,\varepsilon_v + 3862.44 \quad (19)$$

Figures 8 and 9 show comparison between the stress-strain relationships predicted using the EPR-based self-learning FEM and the original data for different confining pressures. From Figure 8 it can be seen that for confining pressures 100, 200 and 300 kPa, convergance was achived (analyeses of FEA and FEB approximately mached) only after one cycle of one pass of the self-learning algorithm and there is a good match between the model predictions and the actual data. For the confining pressures 400, 500 and 600 kPa, convergence was achived after two, three and two cycles of one pass respectively (Figure 9).

The difference between the different confining pressures could be related to the training data, especially within the plastic region. It can be noted that during the self-learning cycles, the performance of the EPR based models improved signifecantly. This is because during cycles much more data were generated which improved the accuracy of training and predictions of

EPR.The results show that EPR has been able to learn and predict the material behaviour under different conditions with very good accuracy. Figure 10 shows the results of stress paths (relationship between mean effextive stress and deviator stress) of the developed EPR based models and the actual data, showing excellent agreeement.

6 – Summary and Conclusion

The conventional approach to constitutive modelling using data mining techniques requires a significant amount of data which could be costly and not available in all cases. Furthermore, obtaining a homogenous stress-strain state in experiments could be very challenging, especially for complex loading conditions. The self-learning algorithm has been proposed to tackle this issue and construct a constitutive model that can capture a complex material behaviour by employing a neural networks based model.

However, ANN has a number of drawbacks. The main shortcoming of ANN is related to its black box nature and the fact that the relationship between input and output variables is described in terms of a weight matrix and biases that are not easily accessible to users. On the other hand, the ANN solutions can become large and complex to interpret, consequently it is difficult to be implemented in FEM. Taking the advantages of an alternative data mining technique called EPR, a new framework of EPR-based self-learning simulation has been established in this paper. An EPR-based self-learning FE model was developed as an efficient approach to eliminate most of the shortcomings of NNCM. The main advantage of using EPR in the self-learning FEM over a neural network is that it gives transparent and structured equations representing the constitutive behaviour of material which can be readily implemented in FE code. The implementation of EPR in the FE procedure is straightforward. In the EPR-based self-learning FEM, there is no need to check yielding, to compute the gradients of plastic potential curve, to update the yield surface, etc. The whole process of the EPR based SelfSim was coded in Matlab, starting from running Abaqus, preparing data for training, selecting the best EPR model, differentiating it and updating the UMAT file within an iterative loop. This significantly simplifies the way of EPR training, reducing the time required for analysis, minimizing the possibility of errors occurring during running programs and establishing the possibility to apply it to more complex material behaviour.

Two examples were used to verify the capabilities of EPR using different training strategies. In the first example synthetic data were used as measurements and the developed EPR-based self-learning model showed very good prediction for elastic-plastic behaviour. In the second example, a series of traixial drained test data were used for training the EPR and the developed EPR models gave accurate predictions compared with the actual data within one or several cycles of one pass of the self-learning algorithm. Note that applying EPR-based SelfSim on more complex behaviour may require several passes to capture the material behaviour. The results show that EPR can be a robust tool for linking laboratory (or field) testing and constitutive modelling. It should be noted that the trained EPR model, like any other data mining technique, is good at interpolation but could be not so good at extrapolation. Therefore, any attempt to use EPR models developed using the self-learning finite element method outside the range of the training data may not provide reliable results.

References

1. Hashash Y, Jung S, Ghaboussi J. Numerical implementation of a neural network based material model in finite element analysis. *International Journal for Numerical Methods in Engineering 2004; 59(7), 989-1005.*

2. Rezania M. Evolutionary polynomial regression based constitutive modelling and incorporation in finite element analysis. *PhD thesis 2008; University of Exeter.*

3. Faramarzi A. Intelligent computational solutions for constitutive modelling of materials in finite element analysis. *PhD thesis 2011; University of Exeter.*

4. Hashash Y, Maruland C, Ghaboussi J, Jung S. Novel Approach to integration of numerical modelling and field observations for deep excavations. *ASCE Journal of Geotechnical and Geoenviromental Engineering 2006; 123(8) pp. 1019–31.*

5. Javadi A, Rezania M. Applications of artificial intelligence and data mining techniques in soil modelling. *Geomechanics and Engineering, An International Journal 1 2009a; (1), 53–74.*

6. Javadi A, Rezania M. Intelligent finite element method: an evolutionary approach to constitutive modelling. *Advanced Engineering Informatics 2009b; 23 (4), 442–451.*

7. Ghaboussi J, Wu X. Knowledge-based modelling of material behaviour with neural networks. *Journal of Engineering Mechanics Division 1991; 117(1), pp.153-64.*

8. Sankarasubramaian G, Rajasekaran S. Constitutive modelling of concrete using a new failure criterion. *computers and structures 1996; 58(5), 1003-1014.*

9. Ghaboussi J, Pecknold, D.A, Zhang M, Haj-Ali R.M. Autoprogressive Training of neural network constitutive models. *International Journal for Numerical Methods in engineering 1998; 42(1), pp.105-26.*

10. Ghaboussi J, Sidarta D.E. "New nested adaptive neural networks (NANN) for constitutive modelling." *Computers and Geotechnics 1998; 22(1), 29-52.*

11. Millar M, 2008. Parallel distributed processing in Rock engineering system. PhD thesis 2008; *Imperial college of science, technology and medicine.*

12. Keshavaraj R, Tock R.W, Haycook D. Airbag fabric material modelling of nylon and polyester fabrics using very simple neural network architecture, *Journal of applied polymer science 1996; 60, 2329-2338*

13. Zhu J-H, Zaman, M.M, Anderson S.A. "Modelling of soil behaviour with a recurrent neural network." *Canadian Geotechnical Journal 1998a; 35(5), 858-872.*

14. Javadi A, Tan T.P, Zhang M. "Neural network for constitutive modelling in finite element analysis.", *Computer Assisted Mechanics and Engineering Sciences 2003; 10, 375-381.*

15. Furukawa T, Hoffman M. Accurate cyclic plastic analysis using a neural network material model. *Engineering Analysis with Boundary Elements 2004; 28 (3), 195–204.*

16. Haj-Ali R, Kim H. Nonlinear constitutive models for FRP composites using artificial neural networks. *Mechanics of Materials 2007; 39(12), 1035–1042.*

17. Kessler B.S, El-Gizawy A.S, Smith D.E. Incorporating neural network material models within finite element analysis for rheological behaviour *prediction. Journal of Pressure Vessel Technology 2007; 192 (2), 58–65.*

18. Shin H. *Neural network based constitutive models for finite element analysis. PhD dissertation 2001; University of Wales Swansea, UK.*

19. Shin H, Pande G. Identification of elastic constants for orthotropic materials from a structural test. *Computers and Geotechnics 2003; 30(7), 571–577.*

20. Sidarta D, Ghaboussi J. “Modeling constitutive behaviour of materials from nonuniform material tests.” *Computers and Geotechnics 1998; 221, 53.*

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

22. Hashash Y, Fu Q, Ghaboussi J, Lade P, Saucier C. Inverse analysis– based interpretation of sand behaviour from triaxial compression tests subjected to full end restraint. *Canadian Geotechnical Journal 2009; 46*(7), 768–791.

23. Jung, S, Ghaboussi J, Marulanda C. Field calibration of time-dependent behaviour in segmental bridges using self-learning simulation. *Engineering Structures 2007; 29(10), 2692–2700.*

24. Qingwei Fu, Hashash Y, Jung S, Ghaboussi J. “Integration of laboratory testing and constitutive modelling of soils.” *Computer and Geotechnics 2007; 34(5), pp. 330-345.*

25. Hashash Y, Song H. The Integration of Numerical Modelling and Physical Measurements through Inverse Analysis in Geotechnical Engineering. *KSCE Journal of Civil Engineering 2008 12(3), pp.165-76.*

26. Hashash Y, Levasseur S, Osouli A, Finno R, Malecot Y. Comparison of two inverse analysis techniques for learning deep excavation response. *Computers and Geotechnics 2010; 37(3), 323–333.*

27. Sung -Woo M, Hashash Y. From direct simple shear test to soil model development and supported excavation simulation: Integrated computational- experimental soil behaviour characterization framework. *Journal of Geoenvironment Engineering 2015; 141(11):04015050.*

28. Giustolisi O, Savic D. A symbolic data-driven technique based on evolutionary polynomial regression. *Journal of Hydroinformatics 2006; 8(3), pp.207-22*.

29. Giustolisin O, Savic D. A. Advances in data-driven analyses and modelling using EPR-MOGA. *Journal of Hydroinformatics 2009; 11*(3-4), 225-236.

30. Doglioni A. *A Novel Hybrid Evolutionary Technique for Environmental Hydraulic Modelling. PhD Thesis 2004; Technical University of Bari, Italy.*

31. Doglioni A, Giustolisi O, Savic D.A., Webb B.W. "An investigation on stream temperature analysis based on evolutionary computing. *Hydrological Processes 2008; 22(3) 315-326.*

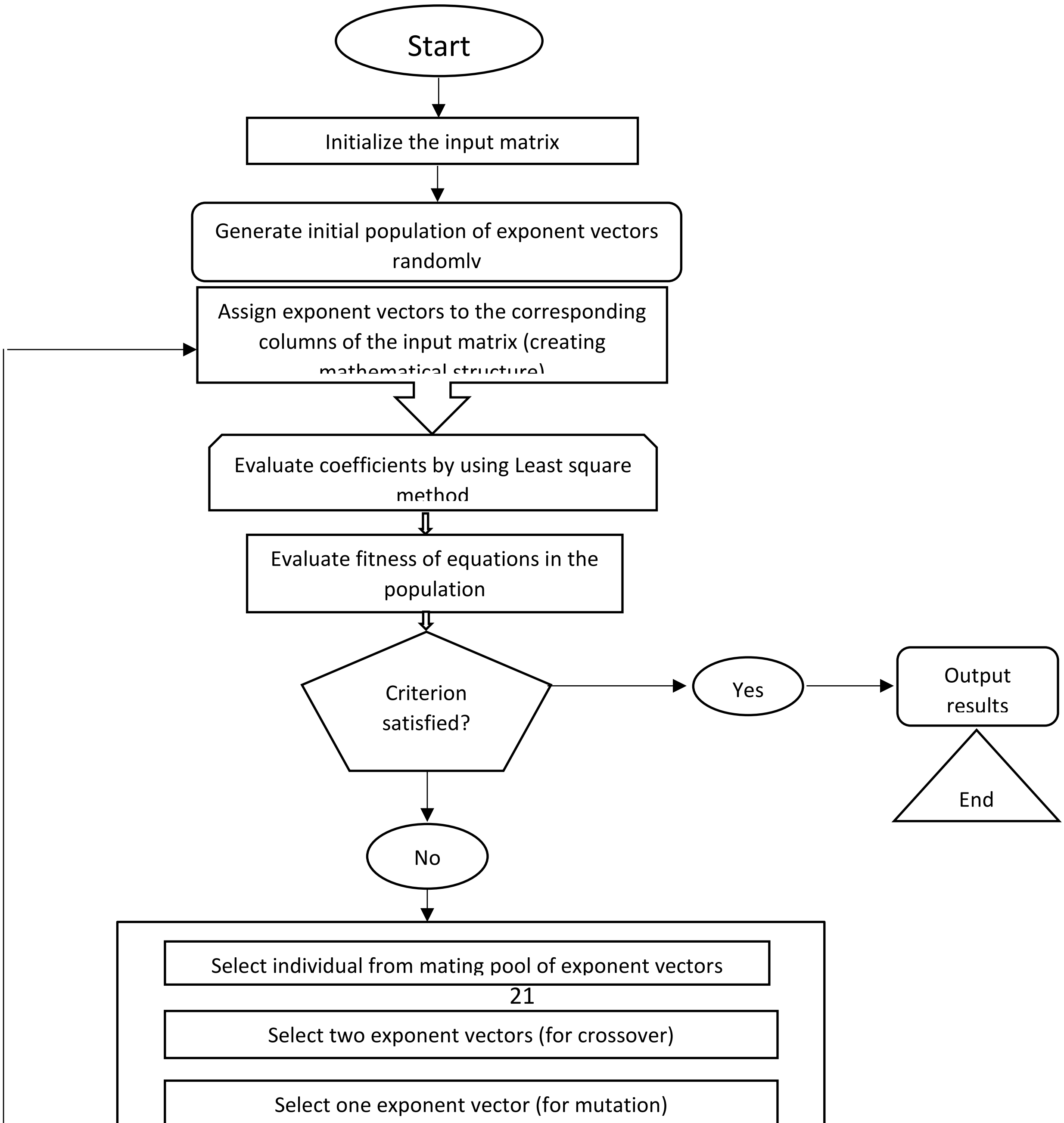
32. Ahangar A. Application of an evolutionary data mining technique for constitutive modelling of geometerials. *PhD thesis 2012; University of Exeter.*

33. Faramarzi A, Javadi A, Amir M. An EPR-based self-learning approach to material modelling. *Computers and structures 2014;137 pp 63-71.*

34. ABAQUS User Subroutine Manual (2016): (version 6.16) Dassault System.

35. Cekerevac C, Laloui L. Experimental study of thermal effects on the mechanical behaviour of a clay. *Numerical and Analytical Methods in Geomechanics 2004; 28 (3), 209–228.*

36. Owen D.R.J, Hinton E. Finite Element in Plasticity*: Theory and Practice 1980; Pineridge Press Limited, Swansea.*


Start
Initialize the input matrix
Generate initial population of exponent vectors randomly
Assign exponent vectors to the corresponding columns of the input matrix (creating mathematical structure)
Evaluate coefficients by using Least square method
Evaluate fitness of equations in the population
Criterion satisfied?
Yes
Output results
End
No
Select individual from mating pool of exponent vectors
Select two exponent vectors (for crossover)
Select one exponent vector (for mutation)

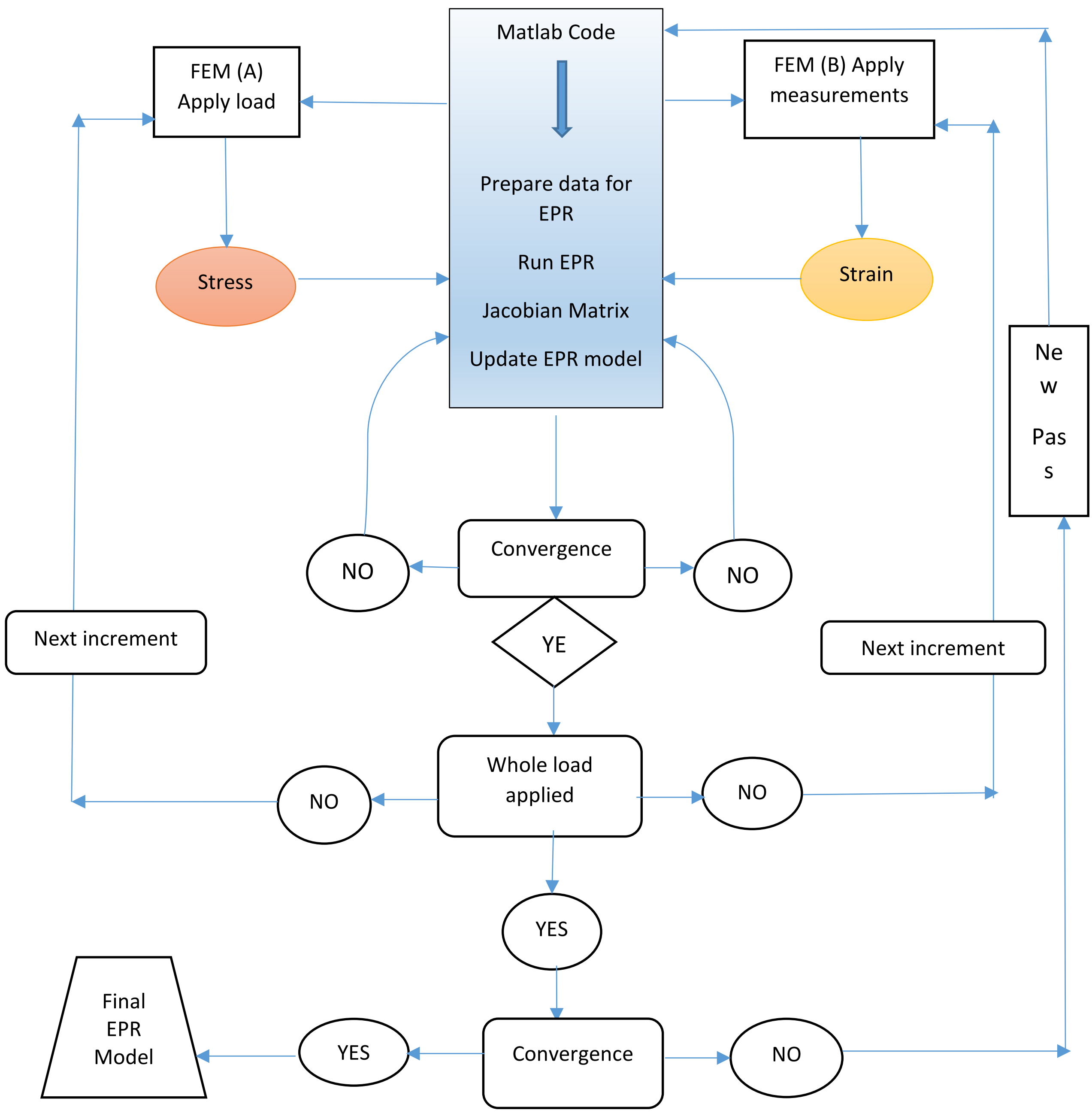


Figure 2. The flow chart of the proposed automation process of EPR-based self-learning

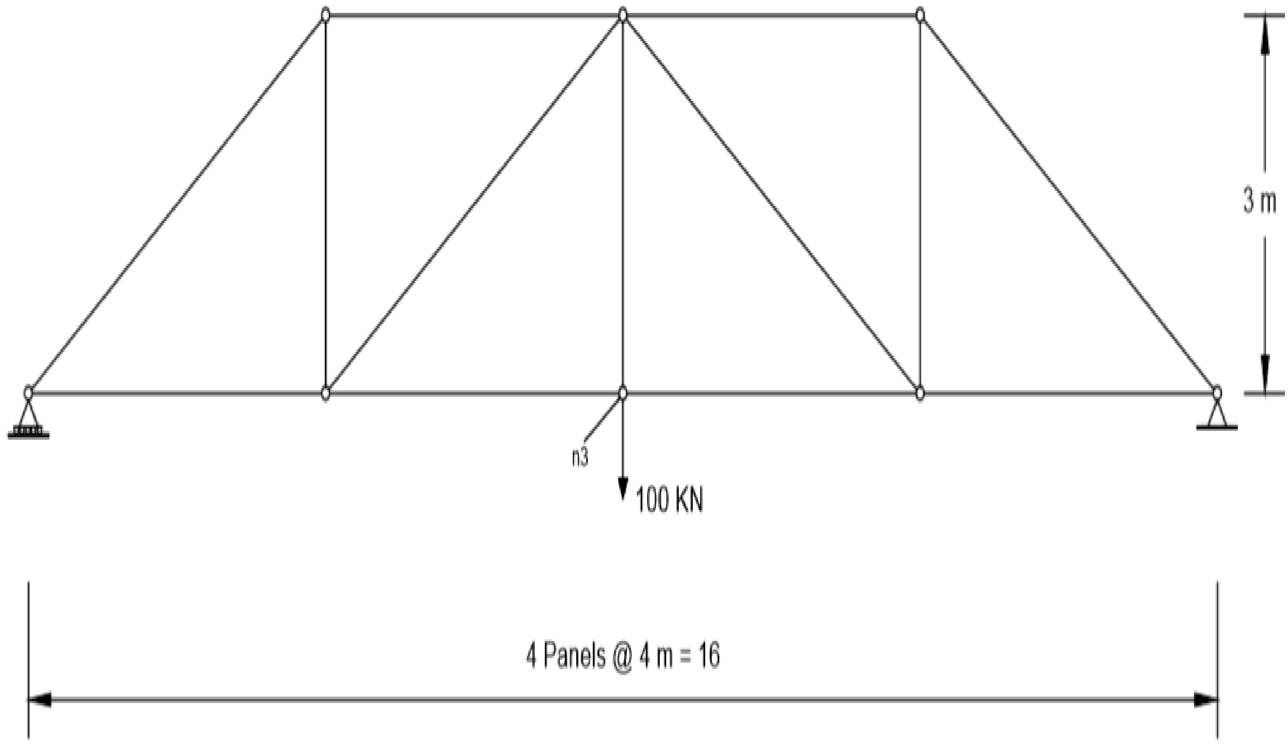


Figure. 3 Truss structure and the applied load.

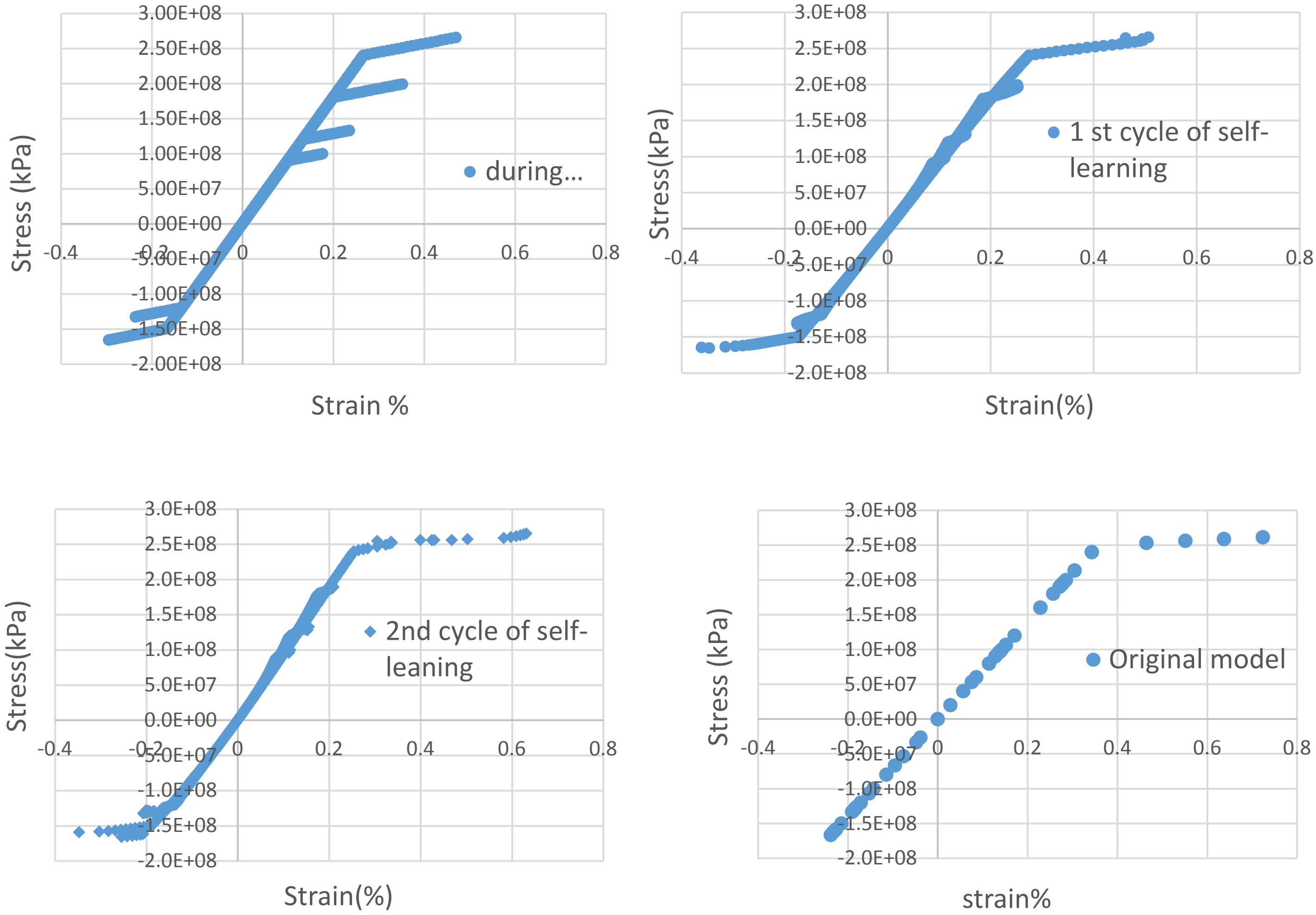


Figure 4. Stress-strain results of EPR based self-learning model and the original model

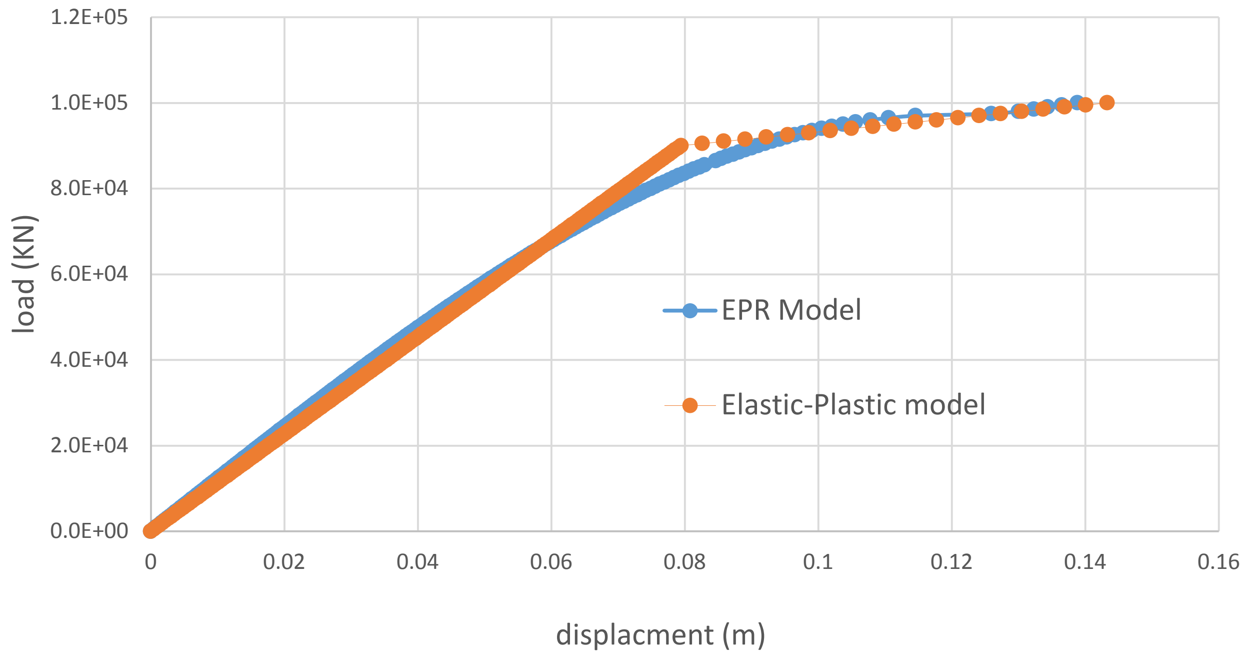


Figure 5. Deformation of node 3 (n_3) predicted by EPR-based self-learning model and the original model.

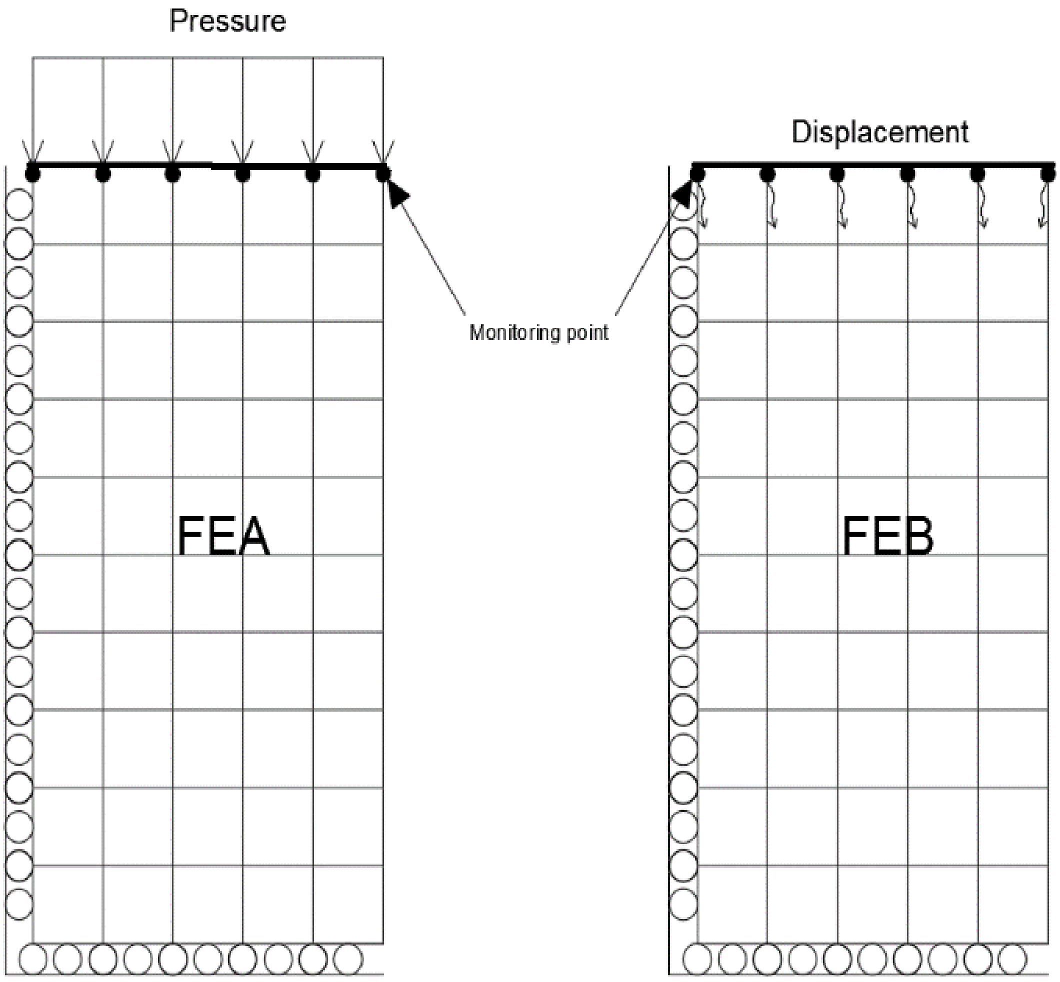


Fig. 6 FEA and FEB models of the traixial tests

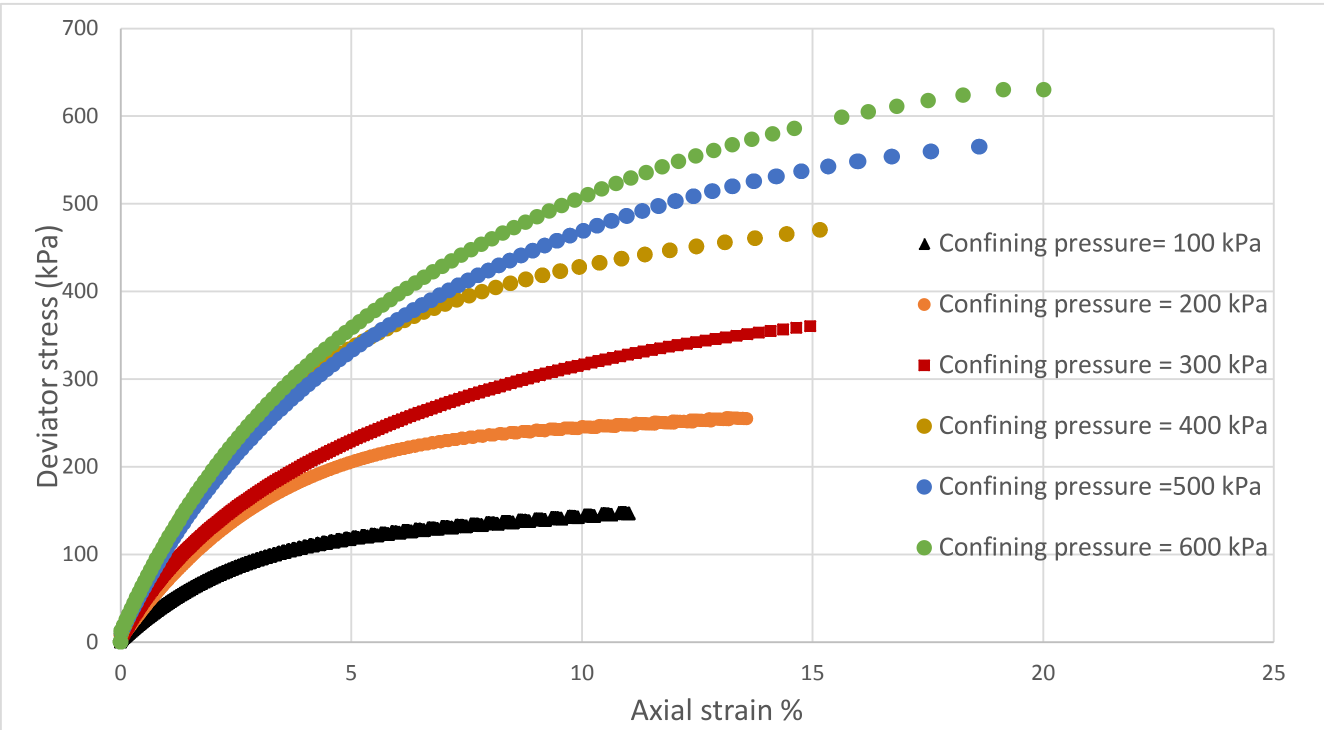
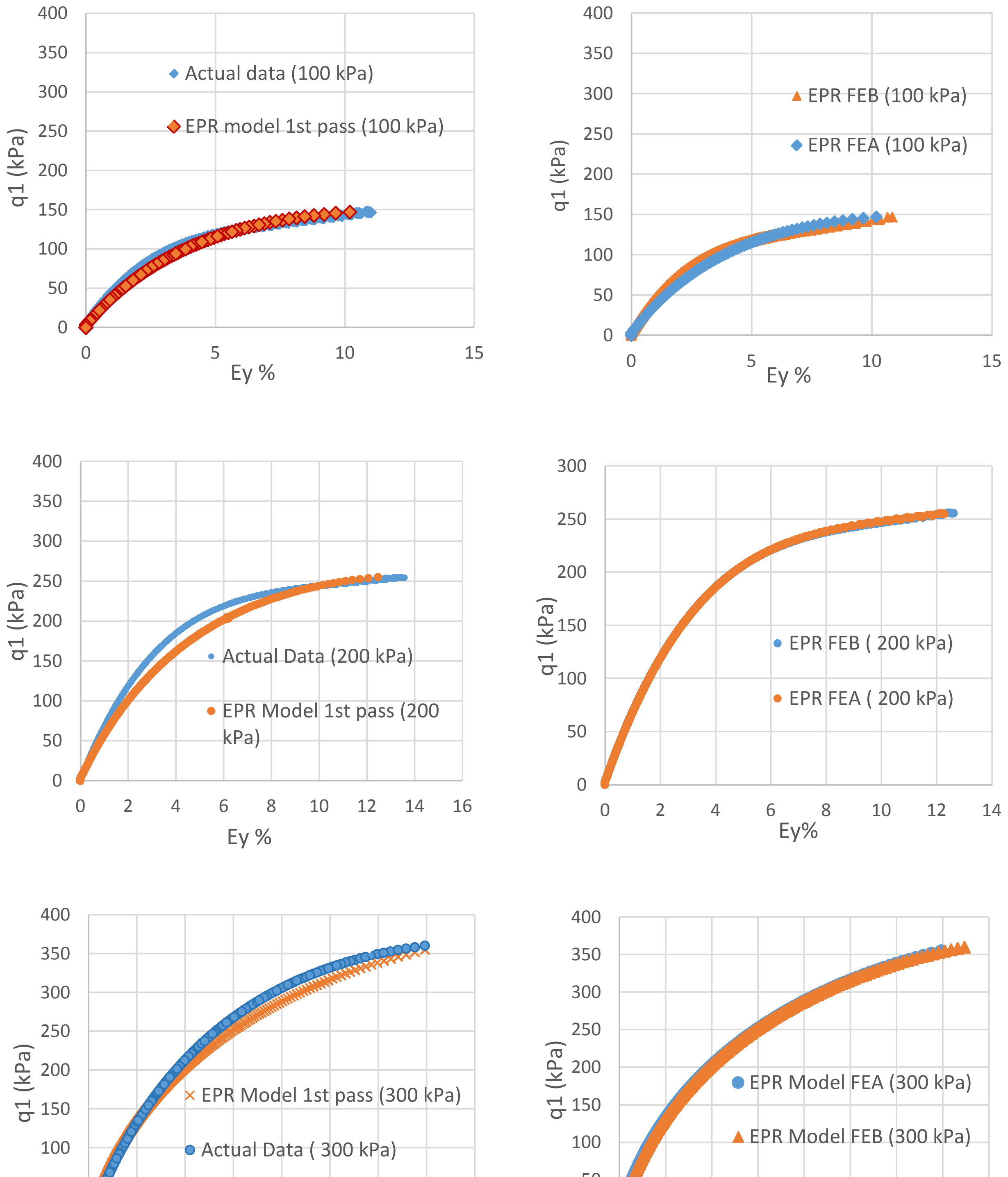
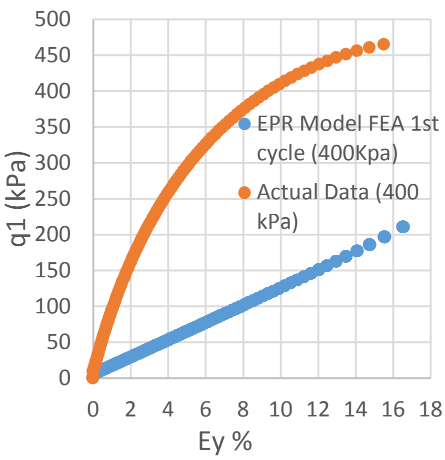
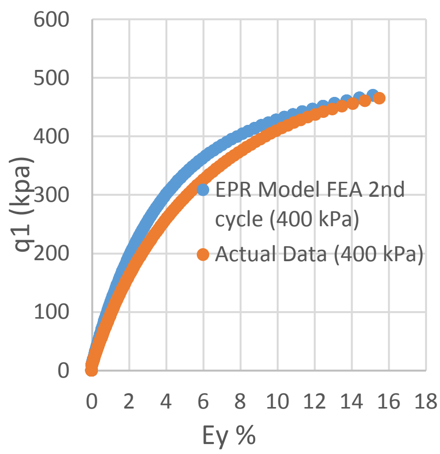


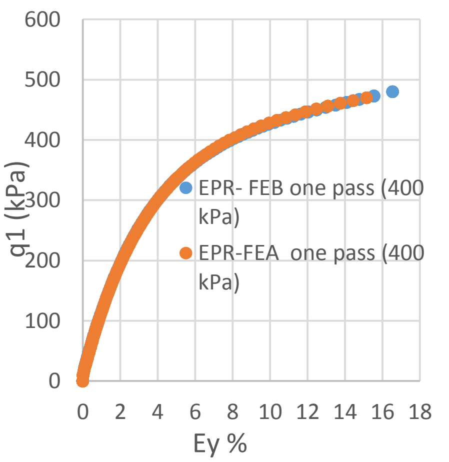
Figure 7. Actual experimental data of traixial test on kaolin clay (after [35]).

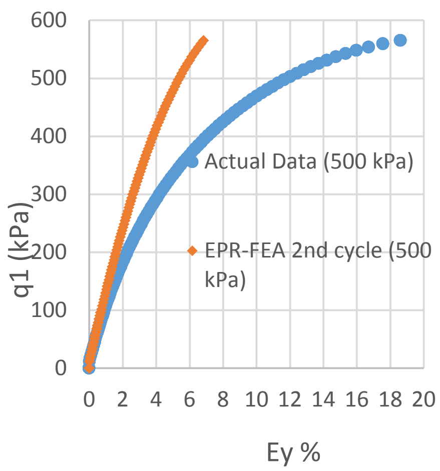


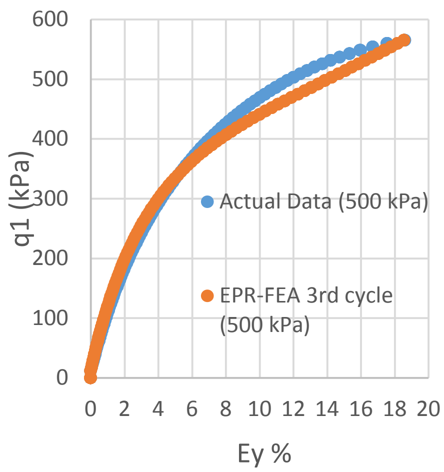
Actual data (100 kPa)
EPR model 1st pass (100 kPa)
q1 (kPa)
Ey %
EPR FEB (100 kPa)
EPR FEA (100 kPa)
q1 (kPa)
Ey %
Actual Data (200 kPa)
EPR Model 1st pass (200 kPa)
q1 (kPa)
Ey %
EPR FEB (200 kPa)
EPR FEA (200 kPa)
q1 (kPa)
Ey%
EPR Model 1st pass (300 kPa)
Actual Data (300 kPa)
q1 (kPa)
EPR Model FEA (300 kPa)
EPR Model FEB (300 kPa)
q1 (kPa)

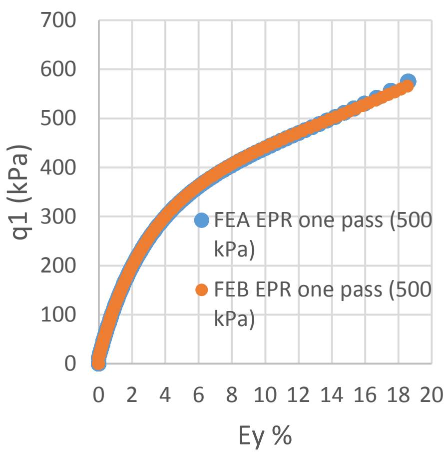

q1 (kPa)
Ey %
EPR Model FEA 1st cycle (400Kpa)
Actual Data (400 kPa)

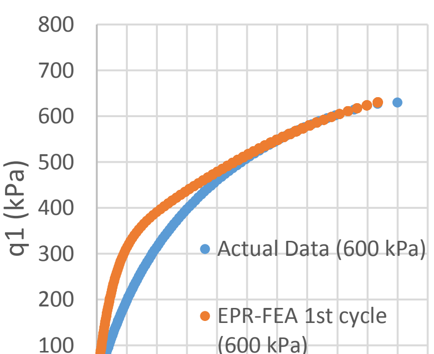

q1 (kpa)
Ey %
EPR Model FEA 2nd cycle (400 kPa)
Actual Data (400 kPa)

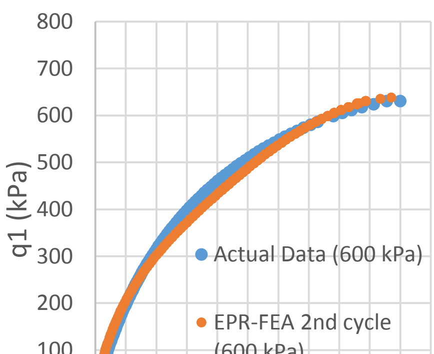

q1 (kPa)
Ey %
EPR- FEB one pass (400 kPa)
EPR-FEA one pass (400 kPa)

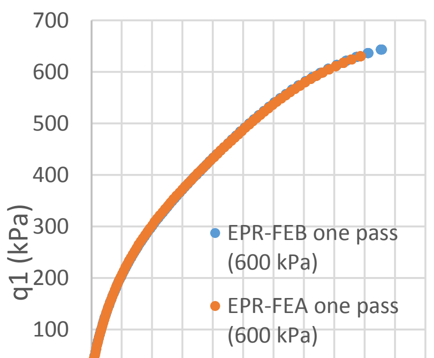

q1 (kPa)
Ey %
Actual Data (500 kPa)
EPR-FEA 2nd cycle (500 kPa)


q1 (kPa)
Ey %
Actual Data (500 kPa)
EPR-FEA 3rd cycle (500 kPa)


q1 (kPa)
Ey %
FEA EPR one pass (500 kPa)
FEB EPR one pass (500 kPa)


q1 (kPa)
Actual Data (600 kPa)
EPR-FEA 1st cycle (600 kPa)


q1 (kPa)
Actual Data (600 kPa)
EPR-FEA 2nd cycle (600 kPa)


q1 (kPa)
EPR-FEB one pass (600 kPa)
EPR-FEA one pass (600 kPa)

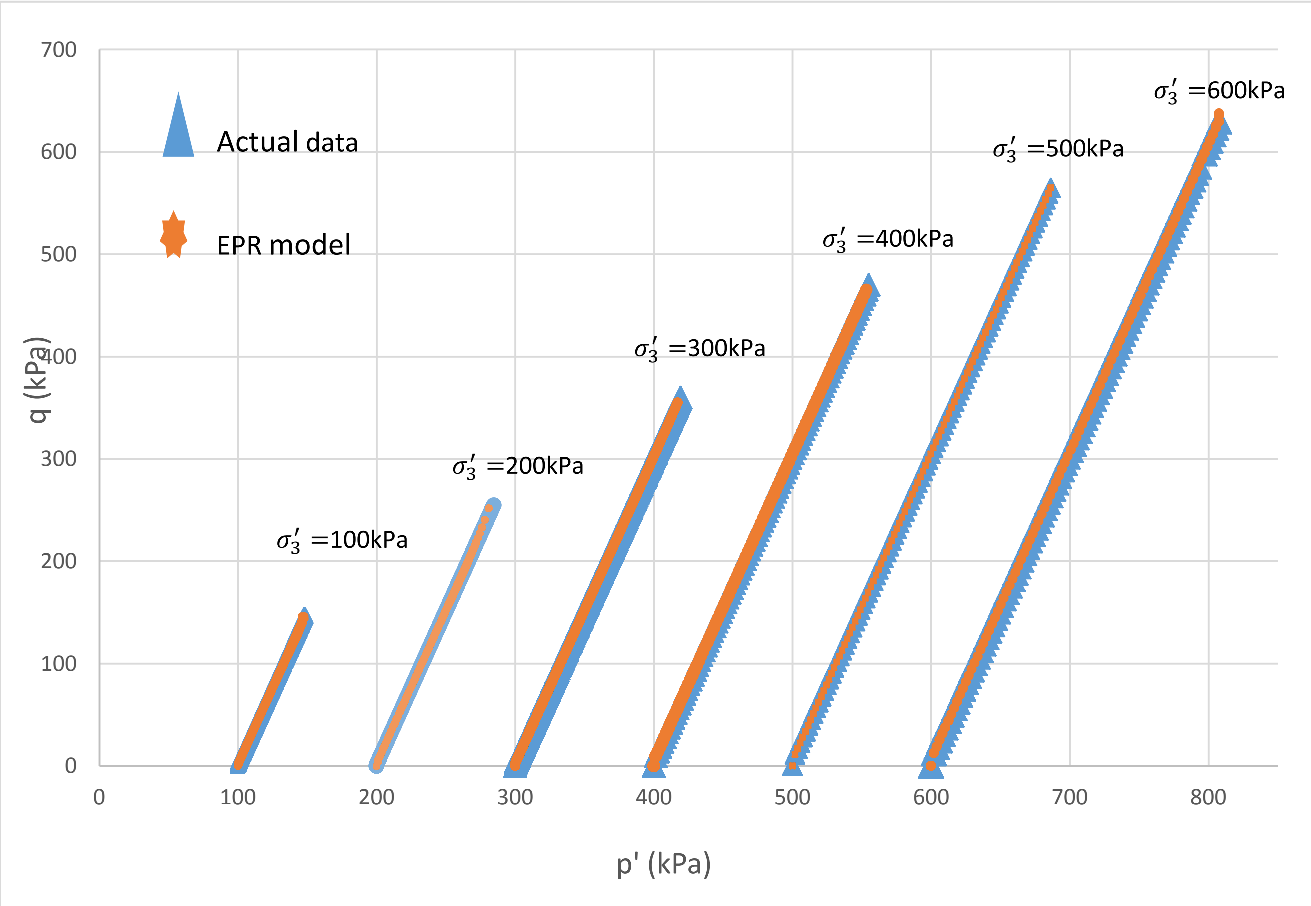


Figure. 10 Comparison of (p’-q) curves of the developed EPR models and actual data.