An EPR-based self-learning approach to material modelling

Faramarzi, Asaad; Javadi, Akbar A.; Alani, Amir M.

DOI:
10.1016/j.compstruc.2013.06.012

An EPR-based Self-learning Approach to Material Modelling

Asaad Faramarzi (Corresponding Author), PhD, MSc, BSc, AHEA
Department of Civil Engineering
School of Engineering
University of Greenwich
Central Avenue
Chatham Maritime
Kent
ME4 4TB

Tel: +44 1634 883126
Email: A.Faramarzi@gre.ac.uk

Amir M. Alani, PhD, MSc, BSc (Hons), CEng, FIMechE, FHEA, MCIHT
Department of Civil Engineering
School of Engineering
University of Greenwich
Central Avenue
Chatham Maritime
Kent ME4 4TB

Tel: +44 1634 883293
Email: m.alani@gre.ac.uk

Akbar A. Javadi, PhD, MSc, BSc
Computational Geomechanics Group
College of Engineering Mathematics and Physical Sciences
University of Exeter
Harrison Building
North Park Road
Exeter
EX4 4QF

Tel: +44 1392 723909
Email: A.A.Javadi@ex.ac.uk

An EPR-based Self-learning Approach to Material Modelling

Abstract

In this paper a new EPR-based self-learning method is presented for modelling the constitutive behaviour of materials using evolutionary polynomial regression (EPR). The proposed approach takes advantage of the rich stress-strain data buried in non-homogenous structural tests. The load-deformation data collected from experiment are used to iteratively train EPR-based material model using finite element simulations of the structural test. Two numerical examples are presented to illustrate the application of the proposed approach. It is shown that the EPR model gradually improves during the self-learning training and provides accurate prediction for the constitutive behaviour of the material.

Keywords: Self-learning; Finite Element; Evolutionary Computation; Material Modelling; EPR

1-Introduction

Evolutionary polynomial regression (EPR) is a new hybrid technique for creating true or pseudo-polynomial models from observed data by integrating the power of least square regression with the efficiency of genetic algorithm (GA) [1]. EPR is proven to be capable of learning complex non-linear relationships from a large set of data, and it has many desirable features for engineering applications. The EPR technique has been successfully applied to modelling a wide range of complex engineering problems including stability of slopes [2]; liquefaction of soils [3]; mechanical behaviour of rubber concrete [4], torsional strength of reinforced concrete beams [5] and many other applications in Civil and Mechanical engineering. The use of EPR to develop material constitutive models (as an alternative to conventional material modelling) has also been proposed by the authors and their co-workers [6-10].

When using EPR for material modelling, the raw experimental or in-situ data are directly used for training the EPR model. Since the EPR learns the constitutive relationships directly from raw data, it is the shortest route from experimental research to numerical modelling. In this approach there are no material parameters to be identified and as more data become available, the model can be improved by re-training of the EPR using the additional data. Furthermore, the incorporation of an EPR model in finite element procedure avoids the need for complex yield/failure functions, flow rules, etc. An EPR model can be incorporated in a finite element code/procedure in the same way as a conventional constitutive model.

The training of EPR material models that has been described in previous works [6-10] is a straightforward approach in which a set of experimentally measured stress-strain data has been used to develop the EPR-based material model. However one of the drawbacks of this approach is that a large number of experiments and data are always required which is costly

and may not even be available at all cases. A single test on a sample of a material provides a set of stress-strain data for a single stress path and is most likely not sufficient to develop EPR models with acceptable accuracy. In addition, obtaining a homogenous state of stress/strain in experiments is very difficult and in some cases impractical, especially for complex cases of loading.

In this paper a completely different approach is presented for training of EPR-based material models. The proposed approach does not require several experiments or homogeneous conditions in the test sample. In this method, the global response information, for example load-deformation data, from an experiment is used to train the EPR. A similar method to this approach was initially proposed by Ghaboussi et al. [11] to train Neural Network (NN) material model and it was named auto-progressive training approach. Many other researchers have continued that work and used auto-progressive or later SelfSim approach to develop NN models for different materials [12-18]. However it is generally known that NN models are presented in the form of large and complex weight matrices and biases which is not readily accessible to user. Also, the ANN models do not give an insight to the user on how the input parameters affect the outputs. In order to overcome the problems associated with NN-based SelfSim method, this paper utilises EPR and the self-learning algorithm to develop transparent mathematical expressions that describe the constitutive behaviour of materials. In what follows a description of EPR technique is provided and then the EPR-based self-learning approach is presented followed by numerical examples to demonstrate the proposed approach.

2-Introduction to Evolutionary Polynomial Regression (EPR)

While there are various data-driven methods based on artificial intelligence, artificial neural network (ANN) and genetic programming (GP) are among the most well known techniques that have been used to model civil and mechanical engineering problems.

ANN use models composed of many processing elements (neurons) connected by links of variable weights (parameters) to form black box representations of systems. ANNs are capable of dealing with large amount of data and learn complex model functions from examples, by training sets of input and output data. ANNs have the ability to model complex, nonlinear processes without having to assume the form of the relationship between input and output variables [11]. However, ANN suffers from some drawbacks; for instance the structure of a neural network (e.g. model inputs, transfer functions, number of hidden layers, etc) must be identified a priori. Another disadvantage of ANNs is the large complexity of the network structure, as it represents the knowledge in terms of a weight matrix and biases which are not readily accessible to user. In addition, parameter estimation and over-fitting are other disadvantages of models constructed by ANN [1, 19].

Genetic programming (GP) is another modelling approach that has been used to model engineering phenomena. GP is an evolutionary computing method that generates transparent and structured mathematical expressions to represent the system being studied. The most common type of GP method is symbolic regression, which was proposed by Koza [20]. This technique creates mathematical expressions to fit a set of data points using the evolutionary process of genetic programming. The genetic programming procedure mimics natural selection as the ‘fitness’ of the solutions in the population improves through successive generations. However, the GP also has some limitations. It is proven that GP is not very

powerful in finding constants and, more importantly, that it tends to produce functions that grow in length over time [1].

To avoid the problems associated with ANN and GP, a new data mining technique called evolutionary polynomial regression (EPR) is used in this study. EPR is a combination of Genetic Algorithm (GA) and Least Square (LS) regression which uses an evolutionary search for exponents of polynomial expressions using a GA engine. The models developed by EPR are concise mathematical equations that allow user to have an insight on how the input parameters are correlated to the output. EPR avoids the over-fitting of models and pushes the models towards simpler structures. In addition EPR avoids unnecessary terms representative of noise in the data and can get models that accurately represent the data. A typical formulation of the EPR expression can be presented as [1]:

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

where y is the estimated output of the system; a_j is a constant value; F is a function constructed by process; $\mathbf{X}$ is the matrix of input variables; f is a function defined by user; and m is the number of terms of expression excluding the bias term a_0. The general functional structure represented by $F(\mathbf{X}, f(\mathbf{X}), a_j)$ is constructed from elementary functions by EPR using GA strategy. The GA is employed to select the useful input vectors from $\mathbf{X}$ to be combined together. The building blocks (elements) of the structure of F, are defined by the user based on understanding of the physical process. While the selection of feasible structures to be combined is done through an evolutionary process, the parameters a_j are estimated by the least square method.

As the EPR starts, the modelling procedure commences by evolving equations. By increasing the number of evolutions it gradually picks up the different participating parameters in order

to form equations representing the constitutive relationship. The level of accuracy at each stage is measured using the coefficient of determination (CoD), i.e., fitness function as:

$$\mathrm{CoD} = 1 - \frac{\sum_N (Y_a - Y_p)^2}{\sum_N (Y_a - \frac{1}{N}\sum_N Y_a)^2} \quad (2)$$

where Y_a is the actual input 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 generation and maximum number of terms) are not satisfied, the current model should go through another evolution in order to obtain a new model [1].

In order to get the best symbolic model(s) of the system being studied, EPR is provided with different objective functions to optimise. The original EPR methodology used only one objective (i.e., the accuracy of data fitting) to explore the space of solutions while penalising complex model structures using some penalisation strategies [1]. However the single-objective EPR methodology showed some shortcomings, such as [22]:

(i) Its performance was exponentially decreasing with increasing the number of polynomial terms.

(ii) The returned models by single-objective EPR were ranked either by their fitness or structural complexity. However, ranking models by structural complexity requires some subjective judgements, and therefore the selection process was often biased by the analyst's experience rather than being purely based on mathematical/statistical criteria.

(iii) While searching for models with *m* terms, it often happens that models with "*m-1*" terms are found with a better accuracy but are discarded because there could be a model that fits better the training data although with more complexity.

To overcome these drawbacks, multi-objective genetic algorithm (MOGA) strategy has been added to EPR. In the case of multi-objective strategy two or three objective functions are introduced in which one of them will control the fitness of the models, while at least one objective function controls the complexity of the models. The multi-objective strategy returns a trade-off surface (or line) of complexity versus fitness which allows the user to select the most appropriate model for the phenomenon studied [22]. In this study the multi-objective EPR is used to develop the EPR-based models. Further details of the EPR technique can be found in [1, 21-22].

3- EPR-based self-learning algorithm

The EPR-based self-learning approach is an algorithm in which the training of an EPR-based material model will take place in an incremental iterative procedure using load-displacement response of a structural test of the material. In fact the self-learning algorithm is an inverse finite element problem in which the type and form of the material model is not known a priori unlike conventional inverse FE analyses. The EPR-based self-learning algorithm can be described in the following steps:

Step 1: The applied load(s) and measured deformation(s) of a structural test at all stages of loading are recorded. Two finite element (FE) models of the structural test are created to simulate the loading stages and measured displacements. Initially the material behaviour is unknown and therefore an elastic behaviour is assumed only for the first stage of the FE analyses.

Step 2a: At the i^{th} load increment, the measured loads are applied on the first FE model (FE model "A") and a FE analysis using the current EPR model is carried out. The FE

analysis computes the stresses and strains in all integration points of the FE model. It is likely that the deformations obtained from FE do not match the actual measured deformations. The resultant stresses of FE analysis are recorded since it is assumed to be a good estimate of the actual stress. However, the computed strains are discarded since they are likely to be a poor estimate of the actual strains.

Step2b: In parallel to "step 2a" at the i^{th} load increment, the measured deformations are imposed on the second FE model (FE model "B") and a FE analysis using the same EPR model is performed. The FE analysis computes the stresses and strains in all integration points of the FE model. The strains obtained from FE analysis are assumed to be a good estimate of the actual stress and are recorded but the computed stresses are considered to be a poor estimate of the actual stresses.

Step3: The stresses data from step2a and the strains data from step2b are collected to form stress-strain pairs of data and are used to train EPR to obtain EPR models. After the training, EPR returns a Pareto-front curve which enables the user to select the most suitable model based on the degree of accuracy and complexity. Once the desired EPR model is chosen, the Jacobian matrix is calculated as described in [7-10]. The computed Jacobian matrix will then be implemented in the FE models in step2a and 2b to repeat the same loading stage until the solution is converged, i.e., results of FE models "A" and "B" are similar. The self-learning algorithm is performed for all the loading increments until the entire load is applied. Each cycle of self learning algorithm that completes the applied load is called a pass. If necessary the process is repeated for one, two or several passes to obtain an EPR material model with a better accuracy. If this is required, then for the next passes the FE models in step 2a and 2b

will start with the last EPR model that is developed at the end of the previous pass instead of an assumed elastic behaviour.

The EPR Self-learning algorithm is illustrated in Figure 1.

The developed EPR model with this approach can be used in the analysis of other types of structural tests with the same material. However it should be noted that the developed EPR model is only valid for the range of the stress and strain values that is encountered during its training in the self-learning process. Any attempt to use the developed EPR models for loading conditions that may lead to stresses or strains beyond these values can end up with unacceptable errors.

The self learning approach that is used to develop EPR models is clearly different from the straightforward approach where the results of some homogenous experiments are used to train and validate the EPR models (e.g., [6]). The structural test that is used to train EPR models using self-learning approach contains a range of stress-strain paths (i.e., stress and strain values at every integration point of all elements) and therefore this makes it a very rich source of data to train the EPR. This can be considered as the most important difference and advantage of the self-learning approach over the more straightforward approach of training EPR.

4-Numerical Examples

In order to examine the proposed EPR-based self-learning approach, two illustrative examples are presented in this section. The structural responses of both examples are generated synthetically using FE simulations. In the first example the applied nodal load and corresponding deformations of a truss structure is used to develop an EPR model via the self-

learning algorithm. In the second example the structural response of a 2D plate subject to non-uniform loading is recorded and used to evaluate the capability of the EPR-bsed self-learning approach.

4-1 Example 1:

A 2D truss structure with 15 axial force elements is considered for this example. The structure, its geometry and boundary conditions are shown in Figure 2. The truss is subjected to a 100 kN load, applied to the middle of the structure at the base (point C).

The load-deformation data was generated using FE simulation of the truss structure under the applied load. It is assumed that during the hypothetical experiment, deformation of node "C" and its corresponding applied load were measured and recorded as shown in Figure 3. From these data only, the EPR material model is trained to predict the stress-strain relationship of the elements of the truss using the self-learning algorithm described in the previous section. To perform the FE simulation and generate the response data, a Ramberg-Osgood model was chosen as material model of the truss elements. The general form of the Ramberg-Osgood model is shown in Equation 3 [23]:

$$\varepsilon = \frac{\sigma}{E} + \frac{2\beta\sigma_0}{3E}\left(\frac{\sigma}{\sigma_0}\right)^n \tag{3}$$

The following values were selected for the parameters in equation 3; $E = 20.0 \times 10^9 Pa;\ \sigma_0 = 1.0 \times 10^7\ Pa;\ \beta = 2.34;\ n = 3.$

Inputs and outputs of the EPR model

In general, there are two different strategies to train EPR-based material models in terms of their input/output parameters. In the first strategy the total values of stresses and strains are used to train and develop models (total stress-strain strategy) while in the second one the incremental values of variables are employed (incremental stress-strain strategy). The source of data, the material behaviour and the way the trained model is to be used in numerical analyses, have significant effects on the choice of training strategy and input and output parameters. An EPR model, formulated in the form of total stress-strain relationships, may be used for modelling the behaviour of materials that are not strongly path-dependant [9, 10]. The second strategy is composed of an input set providing the EPR with the information relating to the current state units (current stresses and strains) and an output that predicts the next state of stress and strain relevant to an input stress or strain increment [7, 8]. In the examples presented in this study, the first strategy (total stress-strain strategy) is used to train and develop EPR models.

After generating the experimental data, two finite element models of the truss structure are created. One of these models is used for a load controlled analysis (FE model "A") and the other one for displacement controlled analysis (FE model "B"). In order to allow the FE models "A" and "B" to start the analysis for the first increment of the loading, an elastic behaviour is considered for truss elements (i.e., $E = 20.0 \times 10^9$). Once the above elastic modulus is embedded in both FE models "A" and "B", the first increments of the FE analyses are carried out. For the first increment, in the FE model "A", 5kN is applied to node "C" and in the FE model "B", its corresponding measured deformation (0.5 mm) is imposed on node "C". These values are obtained from the results of the simulated experiments (presented in Figure 3). After the FE analyses are complete, the stress and strain data at integration points of all truss elements are collected from the FE models "A" and "B" respectively. In order to allow an efficient training process, before the training of EPR, the data set is pruned (e.g.

duplicated data are removed to reduce the required time for training). In this example the values of axial stress are considered as input and axial strain as the output. To adjust the EPR settings, the maximum number of terms was set to 5, and the exponents were limited to [0, 1, 2, 3, 4]. These values were found to be suitable after a number of trial and error runs. Before training the EPR, the data were randomly shuffled in order to make sure that the obtained models are not biased on a particular part of the data.

Once the training of EPR is completed, the best model (representing the material behaviour considering the degree of complexity and accuracy) is chosen from the Pareto-front curve given by EPR. The selected EPR model is implemented in the same FE models (i.e. FE models "*A*" and "*B*") to assess its prediction capability. If the results of the FE models "*A*" and "*B*" are close, the algorithm is continued to the next load increment, otherwise the procedure is repeated until this condition is satisfied. This process is carried out for the entire applied loading (1st pass). After the whole load is applied using this method, the algorithm is repeated one more time to improve the accuracy of the EPR model representing the constitutive behaviour. The selected EPR models at the end of the 5th, 10th, 15th, 20th loading increments for pass 1 are presented in Equations 4 to 7. Equations 8 to 11 present the EPR models obtained for the 2nd pass at the end of the same increments. The CoD values of these equations are shown in Table 1. The prediction capability of the EPR models is compared with the Ramberg-Osgood model (Equation 3, which was used to create the results of the simulated experiment) in Figure 4 and 5 for passes 1 and 2 respectively. Figure 4 and 5 show that the EPR model evolves as the increments increase during the self-learning process. In addition, from Table 1 and Figure 5 it is evident that the EPR models in pass 2 provide a better prediction compare with those in pass 1 at the same load increments.

$$\varepsilon_{11} = 4.32 \times 10^{-6} + 5.48 \times 10^{-11}\sigma_{11} + 2.44 \times 10^{-25}\sigma_{11}^{3} \quad (4)$$

$$\varepsilon_{11} = 2.72 \times 10^{-6} + 4.79 \times 10^{-11}\sigma_{11} + 5.06 \times 10^{-25}\sigma_{11}^{3} \quad (5)$$

$$\varepsilon_{11} = 1.93 \times 10^{-5} + 4.71 \times 10^{-11}\sigma_{11} + 5.34 \times 10^{-25}\sigma_{11}^{3} \quad (6)$$

$$\varepsilon_{11} = -2.95 \times 10^{-5} + 5.07 \times 10^{-11}\sigma_{11} + 2.91 \times 10^{-18}\sigma_{11}^{2} + 6.14 \times 10^{-25}\sigma_{11}^{3} \quad (7)$$

$$\varepsilon_{11} = -8.39 \times 10^{-7} + 5.44 \times 10^{-11}\sigma_{11} + 4.06 \times 10^{-18}\sigma_{11}^{2} + 6.37 \times 10^{-25}\sigma_{11}^{3} \quad (8)$$

$$\varepsilon_{11} = -7.12 \times 10^{-7} + 5.65 \times 10^{-11}\sigma_{11} + 4.10 \times 10^{-18}\sigma_{11}^{2} + 6.24 \times 10^{-25}\sigma_{11}^{3} \quad (9)$$

$$\varepsilon_{11} = -8.65 \times 10^{-6} + 5.88 \times 10^{-11}\sigma_{11} + 4.34 \times 10^{-18}\sigma_{11}^{2} + 6.25 \times 10^{-25}\sigma_{11}^{3} \quad (10)$$

$$\varepsilon_{11} = -2.75 \times 10^{-5} + 6.23 \times 10^{-11}\sigma_{11} + 5.07 \times 10^{-18}\sigma_{11}^{2} + 6.39 \times 10^{-25}\sigma_{11}^{3} \quad (11)$$

The EPR-based material models are implemented in the widely used general-purpose finite element code, ABAQUS, through its user defined material subroutine, UMAT. UMAT updates the stresses and provide the material Jacobian matrix (or stiffness matrix) for every increment in every integration point [24]. To implement the EPR models in UMAT, they are rearranged to the form of $\sigma_{11} = f(\varepsilon_{11})$ relationship and then the derivatives of the rearranged equations are determined with respect to ε_{11} to compute the Jacobian matrix. Further details on numerical implementation of EPR-based material models in FE analysis are presented in Faramarzi et al. [10]. It is worth mentioning that the implementation of EPR models in FE is more straightforward than NN as it only involves performing the direct derivation of the mathematical equations. Another advantage of the EPR-based models over the NN material models is that EPR models are transparent compendious mathematical expressions which clearly give an insight to the user of how the input parameters can affect the output.

Since in this hypothetical example an analytical model is used to create the structural response, the material constitutive behaviour is known a priori. Therefore it is possible to compare the developed EPR models with the actual constitutive model. However, when actual experimental results are used to train EPR models (where material behaviour is not known a priori), the performance and prediction capability of the developed EPR models is assessed by comparing the results of the FE model "*A*" with actual experimental results. This is shown in Figure 6 for this example where the displacement of point C, predicted by FE model "*A*" using the final EPR models at the end of the first and second passes, is compared with actual experimental results. Figure 6 shows that the deformation predicted by EPR model at the end of the second pass is improved compared to the first pass.

4-2 Example 2:

In this example a 2D aluminium plate subject to an in-plane non-uniform biaxial tension is considered. The geometry of the plate and its loading and boundary conditions are presented in Figure 7. Due to symmetry only a quarter of the plate is modelled and presented in this Figure. Similar to Example 1, the structural response of the plate was synthetically generated from FE model of the plate whereby the behaviour of the aluminium plate was represented by an elasto-plastic model within ABAQUS. It is assumed that during the hypothetical experiment the deformations of nodes 1 to 5 in Figure 7 were recorded. These deformations together with the applied loads are used to implement the EPR-based self-learning algorithm and construct the EPR material models. Two FE models of the plate are created and for the first increment of the FE analyses, an elastic modulus is assumed. During the self-learning algorithm the EPR models are constructed using the stress and strain data collected from the

two FE models. The following parameters are chosen as input and output variables of the EPR models:

Input: $\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}$

Output: $\sigma_{11}, \sigma_{22}, \sigma_{12}$

At every step of the self-learning algorithm three EPR models are developed, one for each stress variables, in the form of $\sigma_{ij} = f(\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12})$. The Jacobian matrix is constructed by determining the derivatives of the EPR models with respect to the strains. Three passes of the algorithm were performed in order to improve the accuracy of the EPR models. Figure 8 shows the predictions provided by the EPR model at the end of the 1st, 2nd, and 3rd passes at node 3. In this Figure the actual deformation of node 3 is also shown for comparison. It can be seen that the EPR models provide a good prediction for deformation of this node and the behaviour of the structure in general, and it certainly improves as 2nd and 3rd passes of self-learning algorithm are performed. It should be mentioned that in general the prediction capability and convergence of the EPR models vary with the number of measured structural response and their locations. The minimum required number of points at which the structural response is recorded and their positions depend on the complexity of the material behaviour and the richness of the experimental data collected from those points.

5- Summary and Conclusions

A new EPR-based self-learning is presented to train and develop EPR-based material models. In the previous applications of EPR in material modelling, a single or a set of directly measured stress-strain data from experiment(s) on specimen(s) of the material were used to

train EPR. Usually one single test on a specimen of a material provides one stress path. Therefore various tests on the material are required to obtain different stress paths and cover the range of the stress and strain values that the developed EPR model may encounter in its future applications. Self-learning training of EPR models presents a different approach to the straightforward applications of EPR as there is not a priori set of data available to train EPR. Instead, the stress-strain data buried in a structural test is extracted in an incremental iterative strategy to train and develop EPR models using the global response of the structural test (e.g. load-deformation data). This gives the benefit of reduction in number of the material tests that need to be carried out. The EPR self-learning approach is an inverse FE problem in which the material model is not known in advance and is created and evolved during the self-learning procedure. The self-learning approach in this paper is assessed using two numerical examples on a truss structure and an aluminium 2D plate. Although two rather simple examples are used to assess the EPR self-learning method, this approach is generic and can be applied to various materials.

The EPR-based material model developed in this approach can be used for solving boundary value problems in the same way as the conventional FE method. The incorporation of an EPR-based material model in FE procedure avoids the need for complex yielding/plastic potential/failure functions, flow rule, etc.; there is no need to check yielding, to compute the gradients of the plastic potential curve or to update the yield surface. In addition the implementation of EPR models in FE is more straightforward than NN models as it only involves performing the direct derivation of mathematical equations. The EPR models are transparent concise mathematical expressions which allow the user to have an understanding on how the input parameters can affect the output.

References

[1] O. Giustolisi, D.A. Savic, symbolic data-driven technique based on evolutionary polynomial regression, Journal of Hydroinformatics 8(2006) 207-22.

[2] A. Ahangar-Asr, A. Faramarzi A, A.A. Javadi, A New Approach for Prediction of the Stability of Soil and Rock Slopes, Engineering Computations 27(2010) 878-93.

[3] M. Rezania, A. Faramarzi, A.A. Javadi, An evolutionary based approach for assessment of earthquake-induced soil liquefaction and lateral displacement, Engineering Applications of Artificial Intelligence 24(2011) 142-53.

[4] A. Ahangar-Asr, A. Faramarzi, A.A. Javadi, O. Giustolisi, Modelling mechanical behaviour of rubber concrete using evolutionary polynomial regression, Engineering Computations 28(2011) 492-507.

[5] A. Fiore, L. Berardi, G.C. Marano, Predicting torsional strength of RC beams by using Evolutionary Polynomial Regression, Advances in Engineering Software 47(2012) 178-87.

[6] A. Javadi, M. Rezania, Intelligent finite element method: An evolutionary approach to constitutive modelling, Advanced Engineering Informatics 23(2009) 442-51.

[7] A.A. Javadi, A. Faramarzi, A. Ahangar-Asr, Analysis of behaviour of soils under cyclic loading using EPR-based finite element method, Finite Element in Analysis and Design 58(2012) 53-65.

[8] A. Faramarzi, A.A. Javadi, A.M. Alani, EPR-based material modelling of soils considering volume changes, Computers & Geosciences 48 (2012) 73-85.

[9] A. Faramarzi, Intelligent computational solutions for constitutive modelling of materials in finite element analysis. PhD Thesis 2011, University of Exeter, UK.

[10] A. Faramarzi, A.A. Javadi, A. Ahangar-Asr, Numerical Implementation of EPR-based Material Models in Finite Element Analysis, Computers & Structures 118(2013) 100-108.

[11] J. Ghaboussi, D.A. Pecknold, M. Zhang, R.M. Haj-Ali, Autoprogressive training of neural network constitutive models, International Journal for Numerical Methods in Engineering 42(1998) 105-26.

[12] H.S. Shin, G.N. Pande, On self-learning finite element code based on monitored response of structures, Computers and Geotechnics 27(2000) 161-78.

[13] W. Aquino, J.C. Brigham, Self-learning finite elements for inverse estimation of thermal constitutive models, International Journal of Heat and Mass Transfer 49(2006) 2466–78.

[14] Y.M.A. Hashash, C. Marulanda, J. Ghaboussi, S. Jung, Novel Approach to Integration of Numerical Modelling and Field Observations for Deep Excavations, ASCE Journal of Geotechnical and Geoenvironmental Engineering 132(2006) 1019-31.

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

[16] G.J. Yun, J. Ghaboussi, A.S. Elnashai, Self-learning simulation method for inverse nonlinear modelling of cyclic behaviour of connections, Computer Methods in Applied Mechanics and Engineering 197(2008) 2836–57.

[17] Y.M.A. Hashash, Q. Fu, J. Ghaboussi, P.V. Lade, C. Saucier. Inverse analysis–based interpretation of sand behaviour from triaxial compression tests subjected to full end restraint, Canadian Geotechnical Journal 46(2009) 768–91.

[18] Y.M.A. Hashash, H. Song, A. Osouli. Three-dimensional inverse analyses of a deep excavation in Chicago clays, International Journal for Numercial and Analytical Methods in Geomechanics 35(2011) 1059-75.

[19] O. Giustolisi, D. Laucelli, Increasing generalisation of input-output artificial neural networks in rainfall-runoff modelling, Hydrological sciences 2005; 3(50):439-457.

[20] J.R. Koza, Genetic programming: On the programming of computers by natural selection. 1992, MIT Press, Cambridge. MA.

[21] A. Doglioni, O. Giustolisi, D.A. Savic , B.W Webb. An investigation on stream temperature analysis based on evolutionary computing. Hydrological processes, 22(2008) 315-326.

[22] O. Giustolisi, D.A. Savic, Advances in data-driven analyses and modelling using EPR-MOGA, Journal of Hydroinformatics 11(2009) 225-36.

[23] W. Ramberg, W.R. Osgood, Description of stress-strain curves by three parameters. National advisory committee for aeronautics, 1943; Technical Note No. 902.

[24] ABAQUS User Subroutine Manual (2007): (version 6.8-3) Dassault Systems.

List of Tables:

List of Figures:

Table 1: The CoD values of the EPR equations at 5^{th}, 10^{th}, 15^{th}, and 20^{th} loading increments

Pass 1	Equation No.	4	5	6	7
	CoD (%)	99.986	99.997	99.994	99.990
Pass 2	Equation No.	8	9	10	11
	CoD (%)	99.663	99.971	99.948	99.990

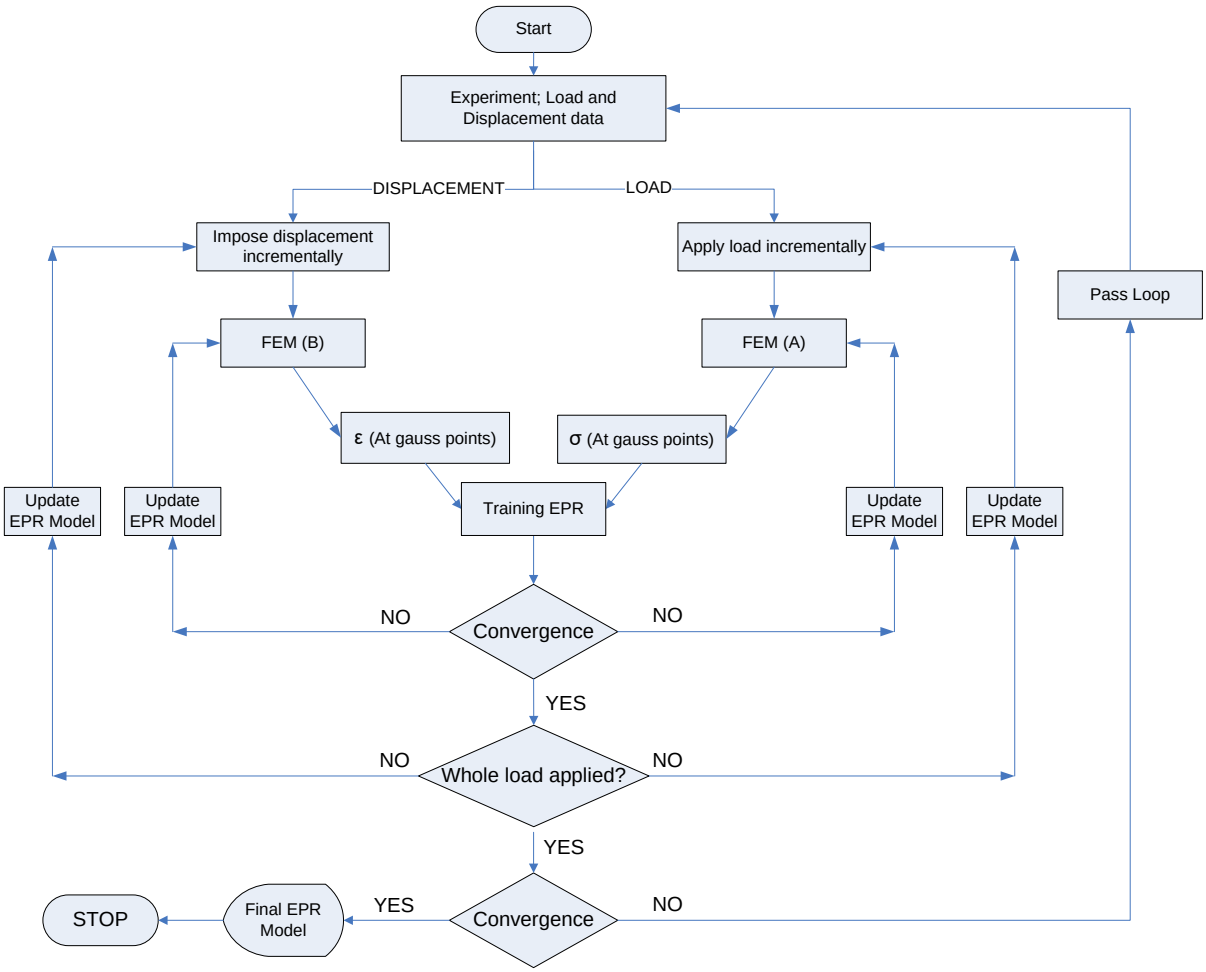

Figure 1: EPR self-learning algorithm

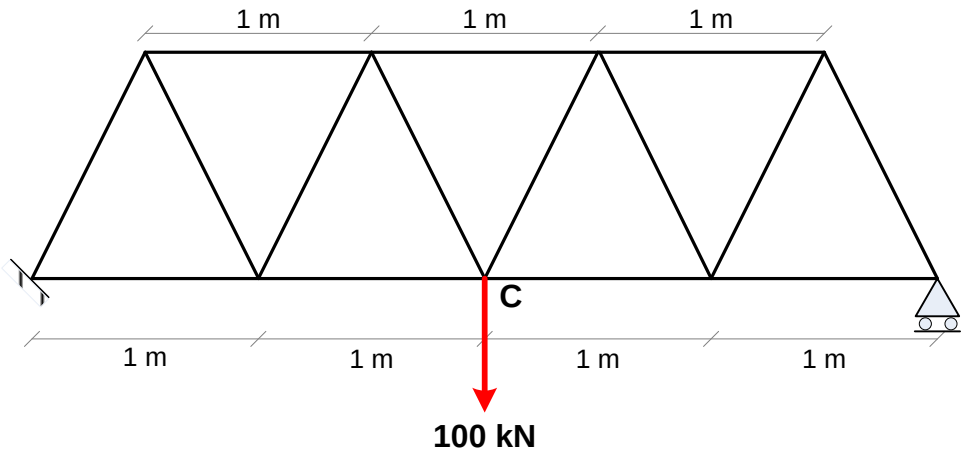

Figure 2: Truss structure in illustrative Example

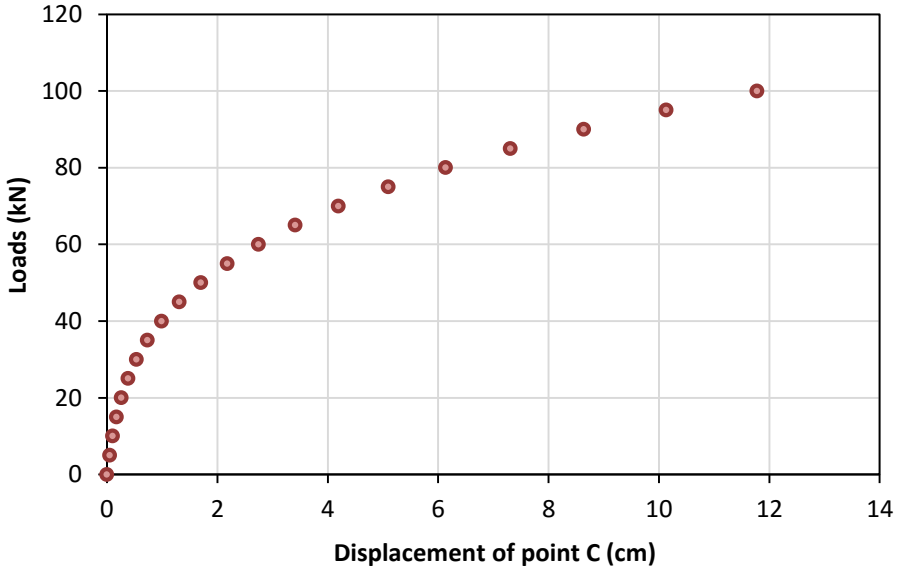

Figure 3: deformation of node "C" in Figure 2 due to the applied load

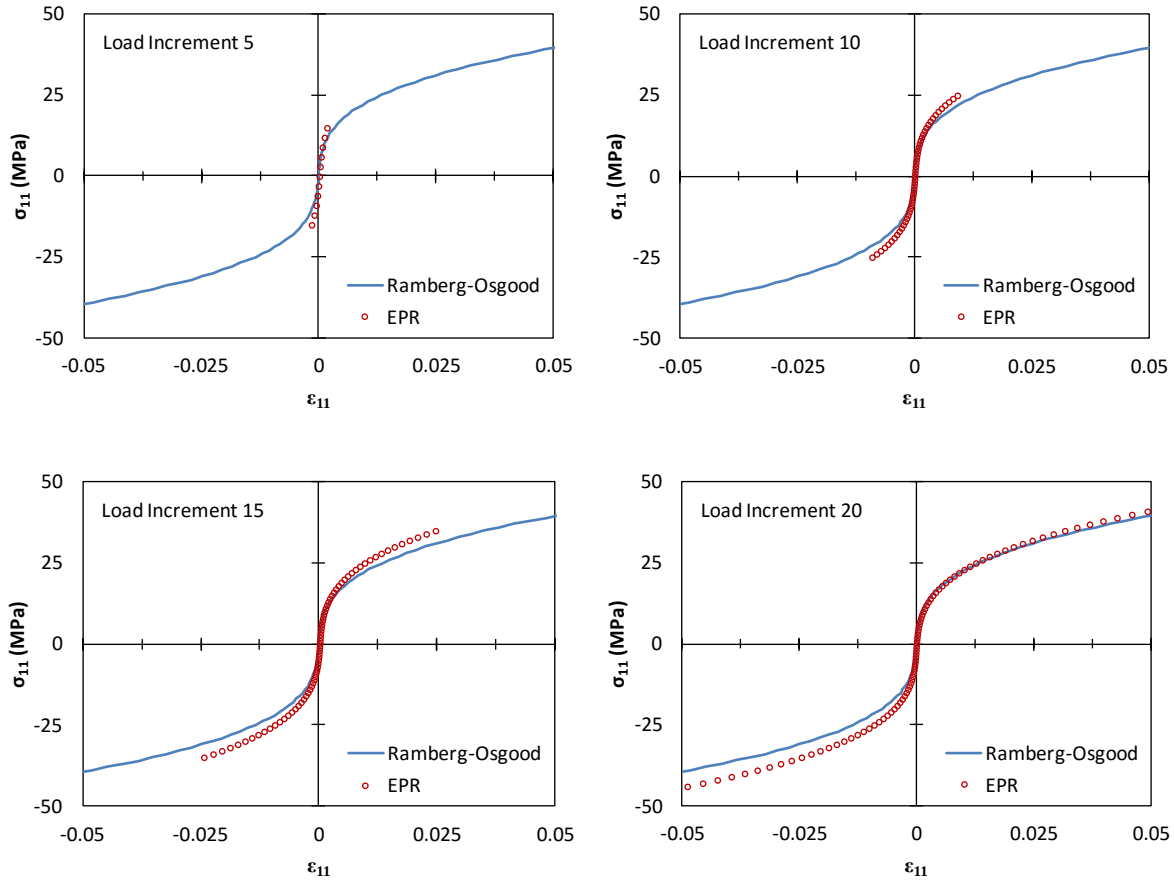

Figure 4: Comparison of EPR models with Ramberg-Osgood model during the first pass of self-learning process

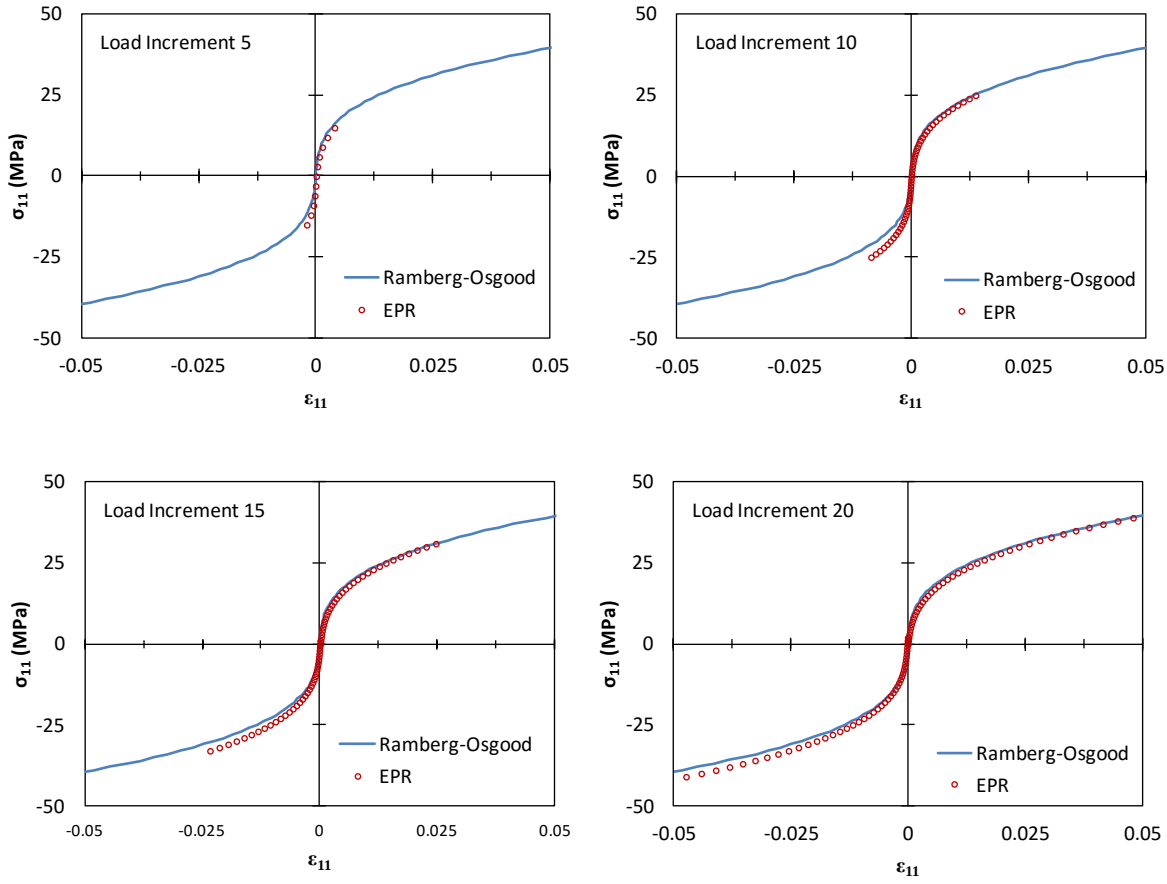

Figure 5: Comparison of EPR models with Ramberg-Osgood model during the second pass of self-learning process

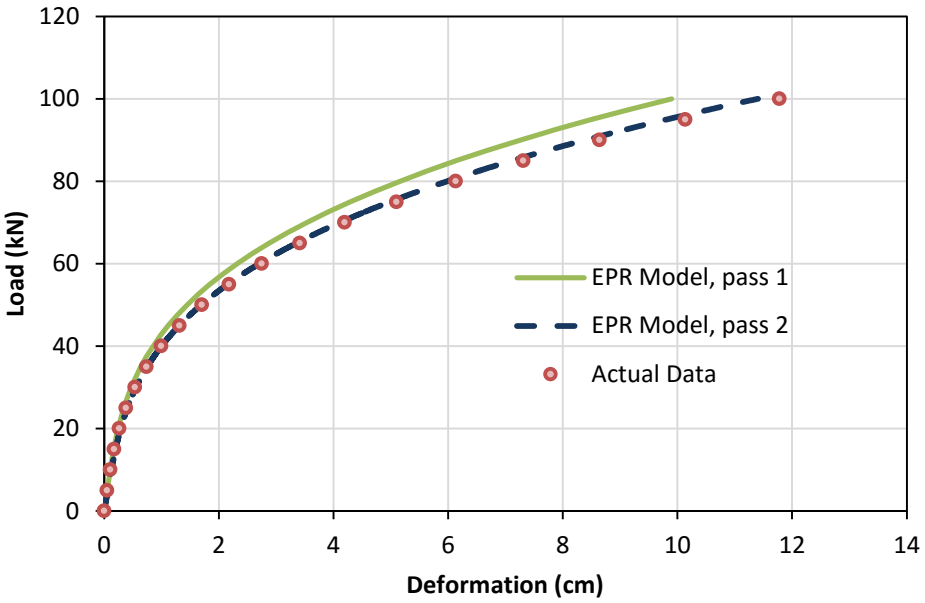

Figure 6: Displacement of node "C" predicted by FE model "*A*" using EPR material models

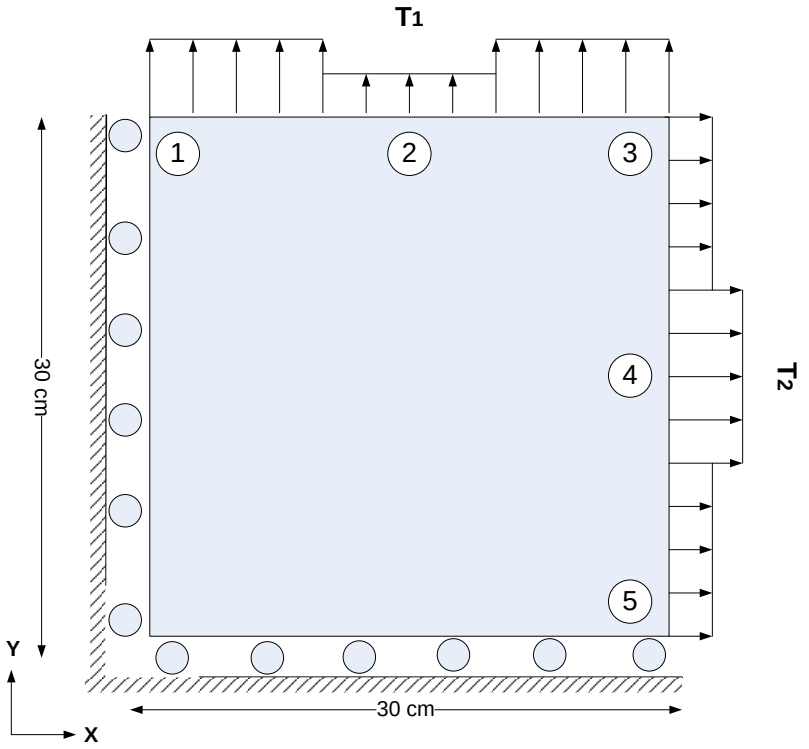

Figure 7: Geometry, loading and boundary conditions of example 2

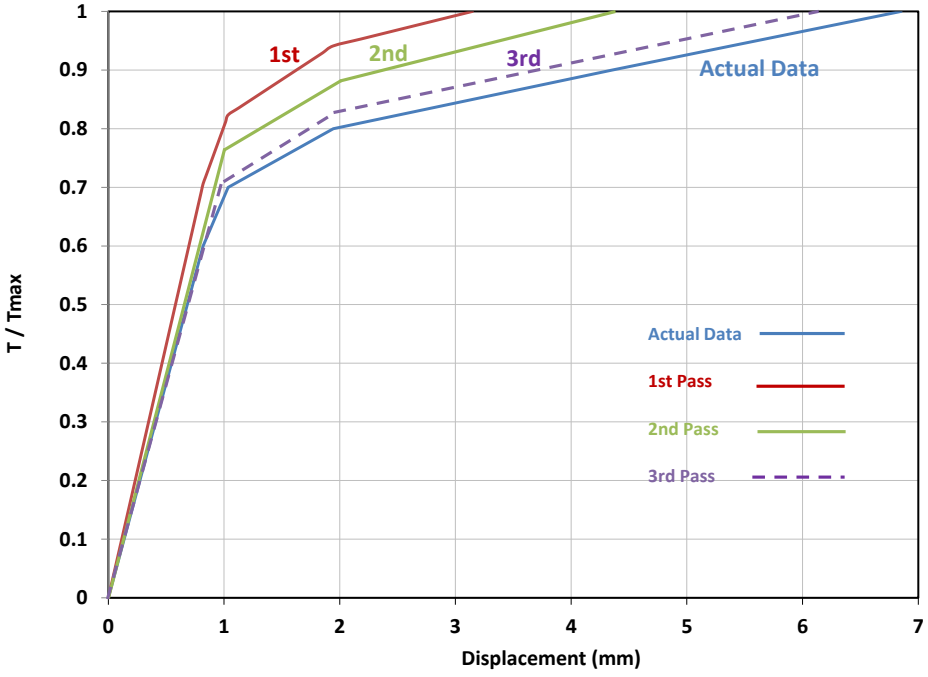

Figure 8: Predictions provided by 1st, 2nd and 3rd passes of EPR self-learning for Displacement of node 3 in Example 2