Computer Assisted Mechanics and Engineering Sciences, 6: 177–188, 1999.

Hybrid NN/FEM analysis of the elastoplastic plane stress problem

Zenon Waszczyszyn and Ewa Pabisek
Institute of Computer Methods in Civil Engineering, Cracow University of Technology, ul. Warszawska 24, 31-155 Kraków, Poland

(Received January 27, 1999)

The back-propagation neural network was trained off line in order to simulate operation of the return mapping algorithm. Selection of patterns and the neural network training as well as testing processes are discussed in detail. The network was incorporated into the FE computer code ANKA as a neural procedure. The hybrid neural-network/finite-element-method program ANKA-H was used for the analysis of two elastoplastic plane stress examples: i) perforated tension strip, ii) notched beam. The results of computations point out quite good accuracy of the hybrid analysis. Some prospects of development of hybrid NN/FEM programs are given at the end of paper.

1. INTRODUCTION

From among many nonstandard methods of data processing which have been developed in recent years the artificial neural networks (ANNs) are worth emphasizing. ANNs have been applied to the analysis of a great amount of problems in science and technology. This concerns also structural engineering and especially mechanics of structures and materials [13].

Computer simulation of ANNs and their application for computation is called for short neurocomputing. Neurocomputing has many special features, which distinguishes it from standard computer data processing. One of them is worth mentioning, i.e. ANNs can be used to map input into output data without known relations between them. This corresponds especially to the Back-Propagation Neural Network (BPNN), cf. e.g. [14, 15]. BPNN is composed of layers and is trained and tested by means of specially selected patterns which are completed as sets of known input/output data [2, 9].

BPNNs can be efficiently used not only as particular simulators but they can be incorporated into standard computational programs as neural procedures. This idea leads to hybrid neural-network/computational computer programs. Of course, it is expedient to apply neural procedures if they are more efficient than computational procedures are, cf. e.g. [3, 12].

Special attention should be paid to material nonlinear problems where neural procedures can be applied to the analysis of constitutive equations. The BPNNs were earlier used to formulate the stress-strain relations in concrete [1]. The moment-curvature relation was established on the base of experimental data [4] or analytical formulae [6]. The inversion of uniaxial Ramberg-Osgood relation was performed in [19].

In [6, 17] the idea of implementation of hybrid procedures in the finite difference or finite element programs was sketched. The aim of this paper is to formulate an "objective" neural procedure for the analysis of elastoplastic plane stress constitutive equations and use the procedure in a hybrid BPNN/FEM computer code.

2. Basic relations and mapping algorithm

2.1. Elastoplasticity equations

The classical equations related to the plastic flow theory are used applying the following assumptions:

a) Huber-Mises-Hencky yield surface (J_2 - yield criterion):

$$F \equiv \frac{1}{2}\boldsymbol{\sigma}^T \mathbf{P} \boldsymbol{\sigma} - \frac{1}{3}\sigma_e^2 = 0, \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress vector and $\mathbf{P}$ is the digital matrix, cf. [8]:

$$\boldsymbol{\sigma}_{(3\times1)} = \{\sigma_x, \sigma_y, \tau_{xy}\}, \qquad \mathbf{P}_{(3\times3)} = \begin{bmatrix} 2/3 & -1/3 & 0 \\ -1/3 & 2/3 & 0 \\ 0 & 0 & 2 \end{bmatrix}; \tag{2}$$

b) isotropic strain hardening with respect to the effective stress σ_e

$$\sigma_e = \sigma_0 + H\varepsilon_p, \tag{3}$$

where σ_0 is the uniaxial tension yield point, H is the linear strain-hardening parameter and ε_p is the plastic strain intensity accumulated of the increments [8]:

$$\Delta\varepsilon_p = \Delta\lambda\sqrt{\frac{2}{3}\boldsymbol{\sigma}^T\mathbf{P}\boldsymbol{\sigma}}, \tag{4}$$

where $\Delta\lambda$ is the increment of plastic multiplier λ;

c) small strains imply the addition of elastic and plastic strains:

$$\Delta\boldsymbol{\varepsilon} = \Delta\boldsymbol{\varepsilon}^e + \Delta\boldsymbol{\varepsilon}^p \quad \text{for} \quad \Delta\boldsymbol{\varepsilon} = \{\Delta\varepsilon_x, \Delta\varepsilon_y, \Delta\gamma_{xy}\}; \tag{5}$$

d) associated flow rule:

$$\Delta\boldsymbol{\varepsilon}^p = \Delta\lambda\frac{\partial F}{\partial\boldsymbol{\sigma}} \equiv \Delta\lambda\mathbf{P}\boldsymbol{\sigma}; \tag{6}$$

e) initially isotropic material which obeys Hooke's relation:

$$\Delta\boldsymbol{\sigma} = \mathbf{E}\,\Delta\boldsymbol{\varepsilon}^e, \qquad \mathbf{E} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix}. \tag{7}$$

2.2. Return mapping algorithm

In the FE analysis of plane stress elastoplasticity the following problem has to be solved:

starting from the known stress vector $\boldsymbol{\sigma}^A$ (or $\boldsymbol{\sigma}^B$ at the yield surface) and the increment of strain vector $\Delta\boldsymbol{\varepsilon}$, find the actual stress vector $\boldsymbol{\sigma}^D$ and corresponding consistent stiffness matrix $\mathbf{E}_D^{ep}$.

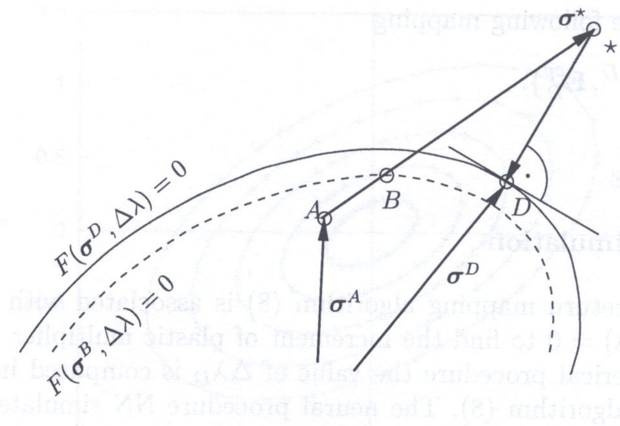


Fig. 1. Scheme of return mapping algorithm

The above problem is analyzed by a numerical procedure which can be briefly summarized as elastic prediction and orthogonal return to the actual yield surface or as return mapping algorithm [10].

In Fig. 1 a scheme of the algorithm is shown in the stress space. A special case is considered if the starting point A is inside the yield surface and elastic increment $\Delta\boldsymbol{\sigma}^e = \mathbf{E}\,\Delta\boldsymbol{\varepsilon}$ crosses the yield surface $F^B \equiv F(\boldsymbol{\sigma}^B) = 0$ at the point B. The main problem is how to find the point D at the actual yield surface $F^D \equiv F(\boldsymbol{\sigma}^D \Delta\lambda) = 0$.

The return mapping algorithm can be written in the following steps:

1. $\boldsymbol{\sigma}^B = \boldsymbol{\sigma}^A + \alpha\mathbf{E}\Delta\boldsymbol{\varepsilon},$
2. $F(\boldsymbol{\sigma}^B) = 0 \longrightarrow \alpha,$
   if $\alpha < 1$ *then* elastic range *else* go to 3 ,
3. $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}^B + (1-\alpha)\mathbf{E}\Delta\boldsymbol{\varepsilon},$
4. Initial value $\Delta\lambda,$

NN

5. $\mathbf{I}^* = (\mathbf{I} + \Delta\lambda\mathbf{E}\mathbf{P})^{-1},$
6. $\boldsymbol{\sigma} = \mathbf{I}^*\boldsymbol{\sigma}^*,$
7. $\sigma_Y^2 = \boldsymbol{\sigma}^T\mathbf{P}\boldsymbol{\sigma},$
8. $F(\Delta\lambda) \equiv \frac{1}{2}\sigma_Y^2 - \frac{1}{3}\left(\sigma_e^B + \Delta\lambda\sqrt{\tfrac{2}{3}\sigma_Y^2}\right)^2 = 0 \longrightarrow \Delta\lambda_D,$ (8)
9. $\mathbf{I}^D = \mathbf{I}(\Delta\lambda_D),$
10. $\boldsymbol{\sigma}^D = \mathbf{I}^D\boldsymbol{\sigma}^*,$
11. $\Delta\varepsilon_p^D = \Delta\lambda_D\sqrt{\tfrac{2}{3}(\boldsymbol{\sigma}^D)^T\mathbf{P}\boldsymbol{\sigma}^D},$
12. $\sigma_e^D = \sigma_e^B + H\Delta\varepsilon_p^D,$
13. $A = \frac{4}{9}\left(\sigma_e^D\right)^2 H\Big/\left(1 - \tfrac{2}{3}H\Delta\lambda\right),$
14. $\mathbf{E}^D = \mathbf{I}^D\mathbf{E}, \quad \mathbf{s} = \mathbf{P}\boldsymbol{\sigma}^D, \quad \mathbf{a} = \mathbf{E}^D\mathbf{s},$
15. $\mathbf{E}_D^{ep} = \mathbf{E}^D - \dfrac{\mathbf{a}\mathbf{a}^T}{A + \mathbf{s}^T\mathbf{a}}$

Algorithm (8) realizes the following mapping

$$\{\boldsymbol{\sigma}^B, \Delta\boldsymbol{\varepsilon}\} \longrightarrow \{\boldsymbol{\sigma}^D, \mathbf{E}_D^{ep}\}. \tag{9}$$

3. BPNN PROCEDURE

3.1. Neural network simulation

The crucial part of the return mapping algorithm (8) is associated with the solution of strongly nonlinear equation $F(\Delta\lambda) = 0$ to find the increment of plastic multiplier $\Delta\lambda_D$ at the actual yield surface $F^D = 0$. In numerical procedure the value of $\Delta\lambda_D$ is computed in an iterative way which follows the steps 5-8 of algorithm (8). The neural procedure NN simulates the part of the return mapping algorithm discussed above.

The Back-Propagation Neural Network (BPNN) was used to formulate the neural procedure. The input and output vectors

$$\mathbf{x}_{(7\times1)} = \{\bar{\boldsymbol{\sigma}}^B, \Delta\bar{\boldsymbol{\varepsilon}}; \chi\}, \qquad \mathbf{y}_{(4\times1)} = \{\Delta\bar{\lambda}, \bar{\boldsymbol{\sigma}}^D\} \tag{10}$$

are composed of the following dimensionless variables:

$$\bar{\boldsymbol{\sigma}} = \boldsymbol{\sigma}/\sigma_e^B, \qquad \Delta\bar{\boldsymbol{\varepsilon}} = E\Delta\boldsymbol{\varepsilon}/\sigma_e^B, \qquad \Delta\bar{\lambda} = E\Delta\lambda, \qquad \chi = H/E. \tag{11}$$

In Fig. 2 the scheme of BPNN procedure is shown. The strain hardening parameter χ was fixed. Thus, in fact, only six components of the input vector are used. Looking at algorithm (8) it is clear that the output stress vector $\bar{\boldsymbol{\sigma}}^D$ is a function of the output variable $\Delta\bar{\lambda}$. Despite the correlation of output variables the output vector $\mathbf{y}$ was composed of $\Delta\bar{\lambda}$, $\bar{\boldsymbol{\sigma}}^D$ in the mapping $\mathbf{x} \to \mathbf{y}$.

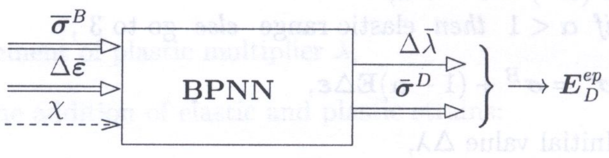


Fig. 2. Scheme of BPNN procedure

After computation of $\Delta\bar{\lambda}$ and $\bar{\boldsymbol{\sigma}}^D$ by means of BPNN procedure the consistent stiffness matrix $\mathbf{E}_D^{ep} = E \cdot \bar{\mathbf{E}}_D^{ep}$ is computed using formulae mentioned in algorithm (8) as steps 11–15.

3.2. Formulation of patterns for the BPNN training and testing

Patterns for the training of BPNNs are usually taken from examples corresponding to special cases and fixed values of parameters. The values of parameters other than those applied to training are used for neural network testing. Of course, generalization properties of such a formulation of BPNN are limited only to considered special cases. This approach was called in [17] the *subjective patterns approach*. In this paper the patterns for training and testing were taken from the Gauss points of finite elements used for the analysis of a perforated plate.

To be independent of special cases of the considered plane stress state, i.e. for special shape of domain, boundary conditions and load applications, *objective patterns* are formulated. The patterns relate only to constitutive relations of the material considered.

Due to dimensionless variables (11) the computed patterns are independent of current instant. The points corresponding to the stress vector $\bar{\boldsymbol{\sigma}}^B$ are uniformly distributed at the scaled yield surface $\bar{F}^B = 0$. In Fig. 3 the contour lines of the yield surface are shown for the shear stresses $\pm\bar{\tau}_{xy}^B = \text{const}$. For the whole yield surface (both for the positive and negative stresses $\bar{\tau}_{xy}^B$) 360 points

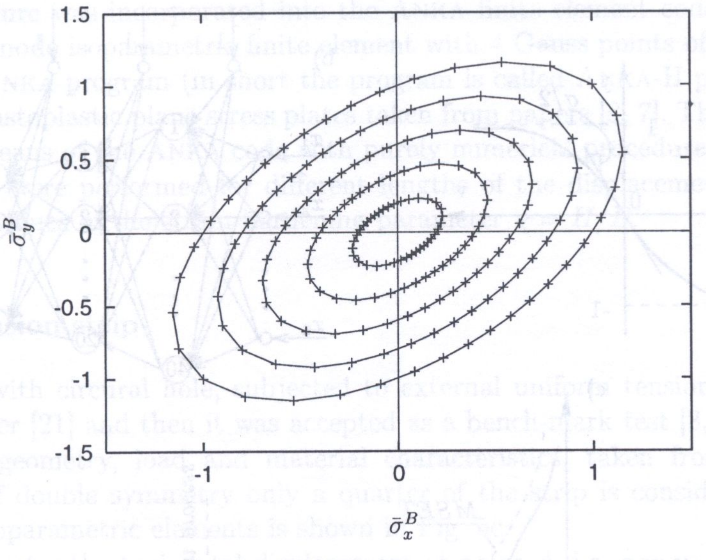


Fig. 3. Contour lines on the yield surface $\bar{F}^B = 0$ at $\bar{\tau}_{xy}^B = const.$ and fixed points $\bar{\sigma}^B$

$\bar{\sigma}^B$ were selected. To each point an increment of strain vector $\Delta\bar{\varepsilon}$ was randomly selected from the range $\Delta\bar{\varepsilon}_x, \Delta\bar{\varepsilon}_y, \Delta\bar{\gamma}_{xy} \in [-10.9, 10.0]$. In such a way about $1 \star 10^6$ patterns were formulated and values of $\Delta\bar{\lambda}$ and $\bar{\sigma}^D$ were computed following algorithm (8).

Obviously, a part of formulated patterns $\{\bar{\sigma}^B, \Delta\bar{\varepsilon}\}_D$ gave elastic responses, i.e. $\Delta\bar{\lambda} = 0$ and $|\sigma_e^D| < |\sigma_e^B|$. Those patterns were eliminated and only from among plastically active patterns the number of patterns $L = 3349$ and $T = 9044$ was randomly selected for learning (training) and testing processes, respectively.

3.3. Selection of BPNN, its training and testing

The neuron features, network architecture and learning method should match well the problem considered.

Because of positive and negative values of the output stress vector components $\{\bar{\sigma}_x, \bar{\sigma}_y, \bar{\tau}_{xy}\}$ the activation function was assumed as the bipolar sigmoid, cf. Fig. 4a:

$$y(v) = \frac{1 - \exp(-\sigma v)}{1 + \exp(-\sigma v)} \in (-1, 1) \quad \text{for} \quad \sigma > 0 \tag{12}$$

The computer simulation was performed by means of the SNNS program [11] (Stuttgart Neural Network Simulator). In this program the value $\sigma = 1$ was fixed.

Three-layer BPNN (two hidden layers and one output layer) was formulated, cf. Fig. 4b. On the basis of numerical experiments 40 and 20 neurons were fixed for the first and second hidden layers, respectively.

From among many learning methods which are accessible in the SNNS program the *Rprop* (Resilient-propagation) *method* was used. The method can be classified as a local adaptative learning method for which the learning rate η_{ij}^l is associated with each synaptic weight w_{ij}^l (more details on the Rprop method can be found in [9, 11]). According to the authors' experience Rprop seems to be one of the most efficient learning methods.

Before the network training all the output variables were scaled to the range $[-0.9, 0.9]$ in order to work in good saturation interval of the activation function.

The learning process was being continued with respect to the Mean-Square-Error

$$MSE(s) = \frac{1}{P} \sum_{p=1}^{P} \sum_{i=1}^{4} \left(t_i^{(p)} - y_i^{(p)}\right)^2, \tag{13}$$

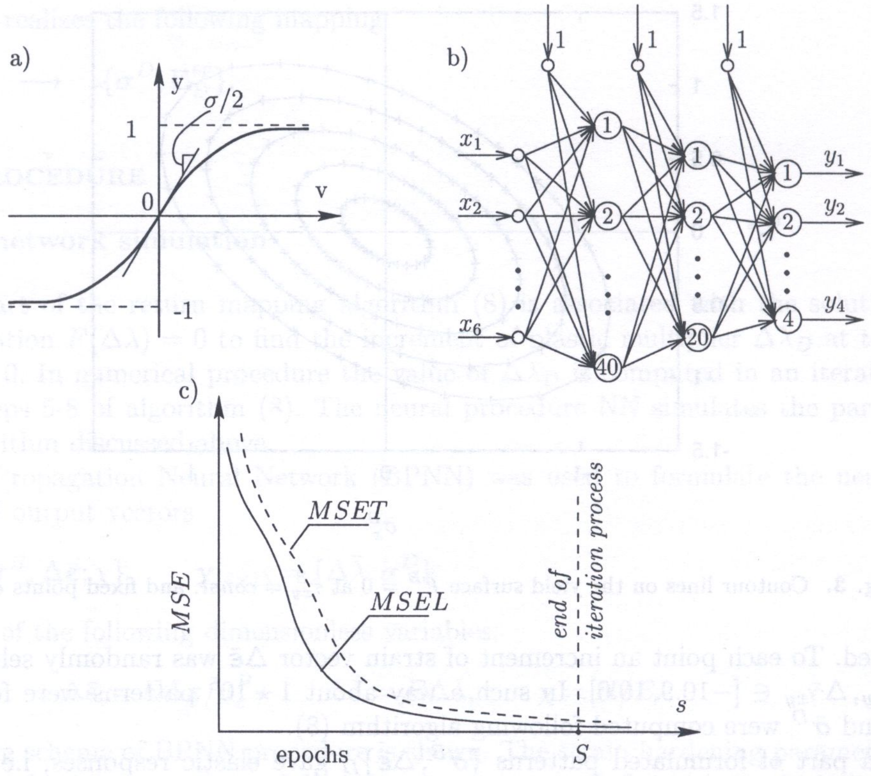


Fig. 4. a) Bipolar sigmoid, b) Three-layer BPNN, c) Relation of *MSE* error and number of epochs s

where: P – number of patterns; $t_i^{(p)}, y_i^{(p)}$ – i-th target and computed output variables, for the p-th pattern respectively, s – number of epochs (one epoch corresponds to one presentation and error backpropagation of all patterns P).

The SNNS program enables us to have graphics for both the learning error *MSEL* and training error *MSET* at the same number of epochs s. For the established BP neural network of the structure 6-40-20-4 the iteration process was continued up to $S = 34000$ epochs cf. Fig. 4c. The corresponding errors are

$$MSEL(S) \approx 8 \star 10^{-5}, \qquad MSET(S) \approx 10 \star 10^{-5}$$

for the number of randomly selected training and testing patterns $L = 3349$ and $T = 9044$, respectively.

In order to compare the computational time used by the neural procedure the same number of 10000 the same patterns were analyzed by the numerical procedure and neural procedure. The performed computations pointed out that the neural procedure needed about 40–50% of the time consumed by the numerical procedure. The larger the value of the components of strain increment vector $\Delta\varepsilon_i$ the more iterations the numerical procedure needs, whereas the computational time of neural procedure is practically constant (independent of values of $\Delta\varepsilon_i$).

4. Hybrid program and its applications

4.1. Hybrid NN/FEM program

BPNN discussed above was trained and tested off line and then the neural network was incorporated into the FEM program as a neural procedure. The main goal of this procedure is to simulate the return mapping algorithm in each Gauss point of reduced integration inside the plane finite elements.

The neural procedure was incorporated into the ANKA finite element code [16]. The attention was focused on the 8-node isoparametric finite element with 4 Gauss points of reduced integration. The hybrid BPNN/ANKA program (in short the program is called ANKA-H program) was used to the analysis of two elastoplastic plane stress plates taken from papers [8, 7]. The computations were carried out also by means of the ANKA code with purely numerical procedures.

The computations were performed for different lengths of the displacement control parameter and for the different values of the strain hardening parameter $\chi = H/E$.

4.2. Perforated tension strip

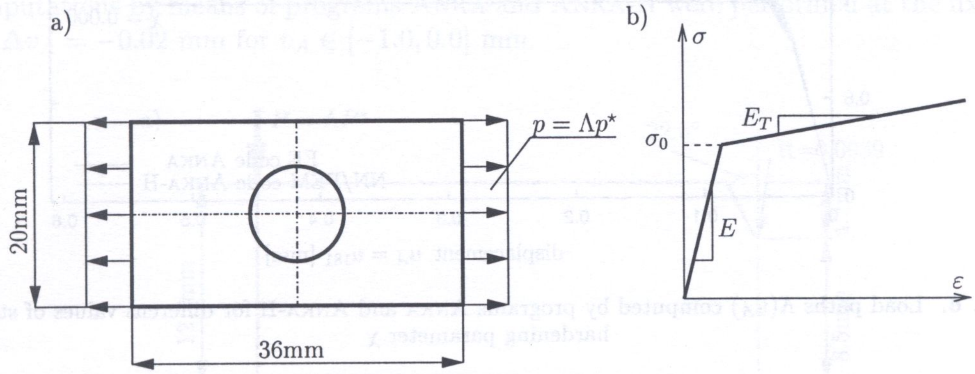
A rectangular strip with circural hole, subjected to external uniform tension was analyzed in an earlier published paper [21] and then it was accepted as a bench-mark test [8, 10].

Data concerning geometry, load and material characteristics, taken from [8], are shown in Fig. 5a,b. Because of double symmetry only a quarter of the strip is considered. The mesh of 8 node quadrilateral isoparametric elements is shown in Fig. 5c.

As a control parameter the horizontal displacement at point A, i.e. $\tau = u_A = u_{181}$ was used. By means of the computational program ANKA and the hybrid program ANKA-H the load paths $\Lambda(u_A)$ were computed, where: Λ – load parameter depicted in Fig. 5a.

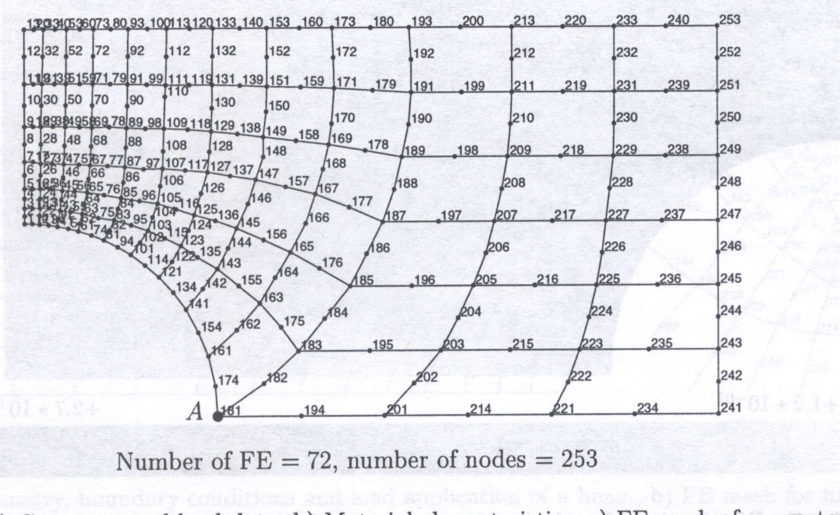
The length of the control parameter $\Delta\tau = \Delta u_A$ was assumed to be constant during the whole deformation process. The computations were carried out by ANKA at various values of the control

a) b)



$$E = 7 \star 10^4 \text{ N/mm}^2, \quad E_T = 2.24 * 10^3 \text{N/mm}^2, \quad \chi = E_T/E = 0.032,$$
$$\sigma_0 = 243 \text{N/mm}^2, \quad \nu = 0.2, \quad p^\star = 121.5 \text{N/mm}^2$$

c)



Number of FE = 72, number of nodes = 253

Fig. 5. a) Geometry and load data, b) Material characteristics, c) FE mesh of a quarter of strip

parameter Δu_A= 0.01, 0.02, 0.08 mm. Similarly as in [17] it was stated that the length of control parameter Δu_A does not affect the load path $\Lambda(u_A)$.

In [17] it was observed that in case of greater errors $MSE(S)$ of the neural procedure trained off line the load path $\Lambda(u_A)$, computed by the hybrid program ANKA-H is sensitive to the length of control parameter Δu_A. In the case of BPNN discussed in Section 3.3 the load path is practically non-sensitive to changes of $\Delta u_A \in [0.01, 0.8]$. That is why computations by ANKA-H were performed at $\Delta u_A = 0.02$ mm (a shorter length was used to have more smooth graphics).

In Fig. 6 load paths $\Lambda(u_A)$ are shown as continuous lines for ANKA and as broken lines for ANKA-H. The neural procedure was trained at the value of strain hardening parameter $\chi = 0.032$. This procedure was used to compute load paths for other values of the strain hardening parameter χ. From the results shown in Fig. 6 it can be concluded that for $\chi \in [0.0, 0.07]$ the accuracy of

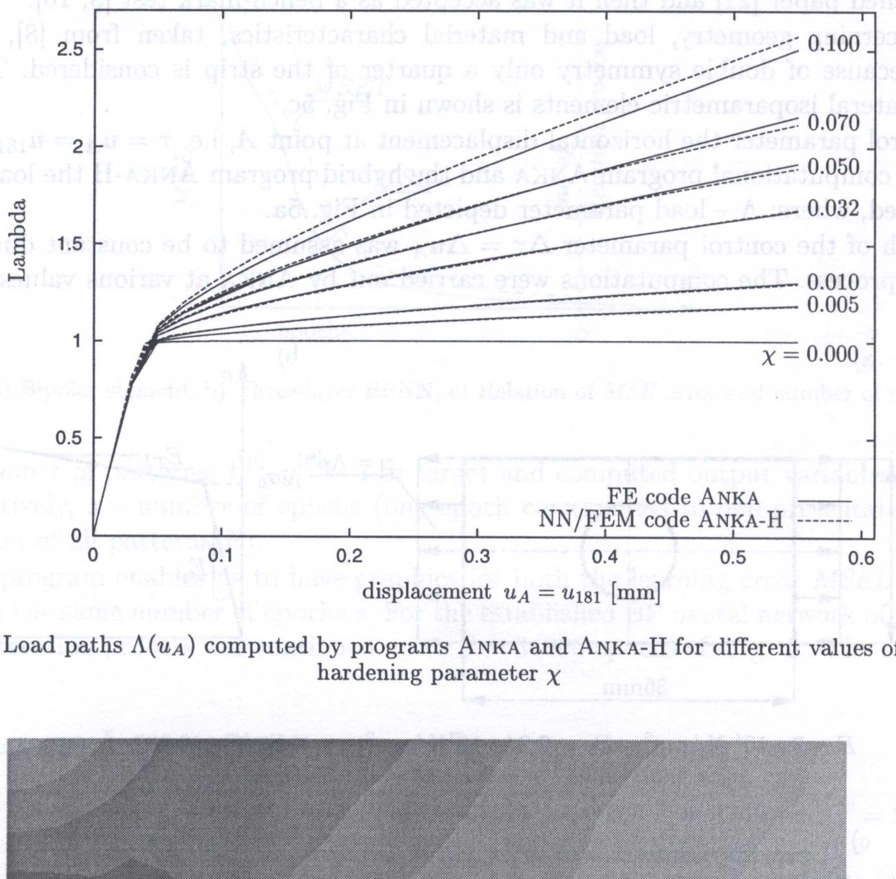


Fig. 6. Load paths $\Lambda(u_A)$ computed by programs ANKA and ANKA-H for different values of strain hardening parameter χ

Fig. 7. Distribution of equivalent stress σ_e zones for $\chi = 0.032$ and $u_A = 0.55$ mm

neural simulation is quite satisfactory from the viewpoint of analysis of all the FE system. For the strain hardening parameter $\chi = 0.10$ there are discepancies for the load paths computed by FE code ANKA and the hybrid program ANKA-H.

In Fig. 7 the postprocessed results are shown with respect to distribution of equivalent stresses σ_e zones, corresponding to $\chi = 0.032$ and $u_A = 0.55$ mm. It was evaluated that in this case about 78% of effective stresses σ_e in Gauss points were beyond the yield points σ_0.

4.3. Notched beam

The other example was taken from [7], where a notched bar was analyzed by means of various models of elastoplastic material and various FE programs. Contrary to [7], where the plane strain state was assumed, the plane stresses are analyzed in the present paper.

The beam of geometry, boundary conditions and applied load is considered as a plate of unit thickness, cf. Fig. 8a. Material of isotropic, linear strain hardening is assumed, cf. Fig. 5b, and data are shown in Fig. 8a.

Similarly as in the previous example the 8-node isoparametric elements were used with 4 Gauss points of reduced integration. The mesh of FEs is shown in Fig. 8b and node distribution in the vicinity of the notch is shown in Fig. 8c.

Vertical displacement of the load application point A was used as the displacement control (parameter $\tau = v_1 = v_A$), cf. Fig. 8a.

The computations by means of programs ANKA and ANKA-H were performed at the fixed length of the step $\Delta v_A = -0.02$ mm for $v_A \in [-1.0, 0.0]$ mm.

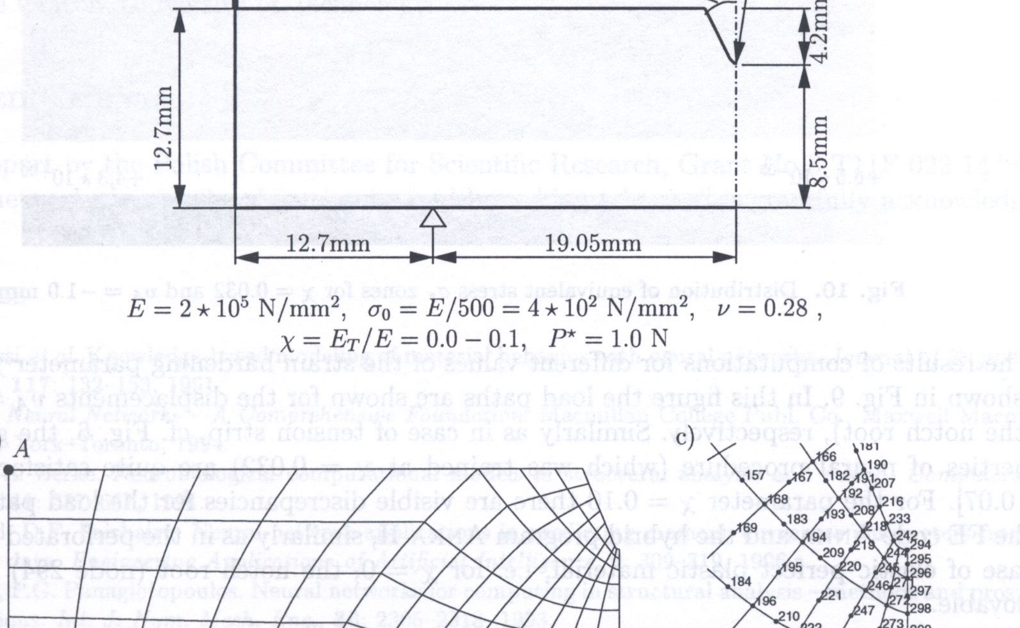


Fig. 8. a) Geometry, boundary conditions and load application of a beam, b) FE mesh for half of beam, c) Nodes in vicinity of notch

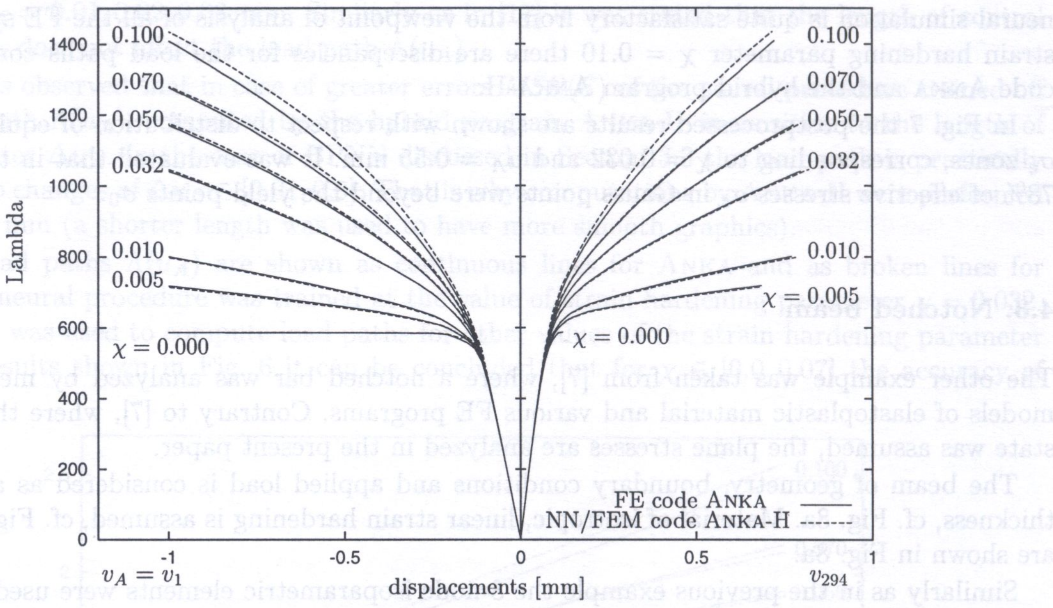


Fig. 9. Load paths $\Lambda(v_A)$ computed by programs ANKA and ANKA-H for different values of strain hardening parameter χ

Fig. 10. Distribution of equivalent stress σ_e zones for $\chi = 0.032$ and $v_A = -1.0$ mm

The results of computations for different values of the strain hardening parameter $\chi \in [0.0, 0.10]$ are shown in Fig. 9. In this figure the load paths are shown for the displacements $v_A = v_1$ and v_{294} (at the notch root), respectively. Similarly as in case of tension strip, cf. Fig. 6, the generalization properties of neural procedure (which was trained at $\chi = 0.032$) are quite satisfactory for $\chi \in [0.0, 0.07]$. For the parameter $\chi = 0.10$ there are visible discrepancies for the load paths computed by the FE code ANKA and the hybrid program ANKA-H, similarly as in the perforated tension strip. In case of elastic perfect plastic material, i.e. for $\chi = 0$, the notch root (node 294) is practically immovable.

In Fig. 10 distribution of zones is shown with respect to the effective stress σ_e at the value of control parameter $v_A = -1.0$ mm. A concentration of stresses is visible in the vicinity of load and support reaction applications as well as in the vicinity of the notch root. It was evaluated that in this case about 46% of effective stresses σ_e in Gauss points were beyond the yield points σ_0.

In order to compare processor time used by the program the computation was carried out for the strain hardening parameter $\chi = 0.005$ and 65 steps $\Delta v_A = -0.02$ mm. In case of ANKA the processor time was 74 sec at 145 global iterations. The processor time used by the ANKA-H progam was 38 sec at 83 iterations. Then the computation was repeated for $\chi = 0.07$. The processor time was 78 sec at 146 iterations for ANKA and the corresponding figures were 64 sec at 117 iterations if ANKA-H program.

5. FINAL REMARKS

Neural procedures open doors to formulation of new types of computer programs. In the paper the back-propagation neural network is discussed as a neural procedure for the simulation of return mapping algorithm operation. The network trained and tested off line on numerical patterns was incorporated into a FE program. The hybrid NN/FEM program ANKA-H was verified on two examples of plane stress state. It was stated that the accuracy of results by the hybrid and purely computational programs, ANKA-H and ANKA respectively, are very close to each other.

It is rather difficult to conclude an opinion on the efficiency of the hybrid program. The computational time of neural procedure is in average about 40–50% of the time consumed by the numerical procedure. The execution time related to the analysis of structures by ANKA-H can be even about 50% shorter than that achieved by ANKA. A deeper explanation of the mentioned computational profits which gave the the hybrid program needs further investigations.

In the paper a very simple model of material is considered, cf. Section 2.1. A promising prospect is that neural procedures can be easily formulated for much more complex materials. It is worth emphasizing that not only theoretical models of material but also experimental data can be used for the training and testing of corresponding neural networks.

Other prospects are related to the application of neural procedures on the cross-sectional level, i.e. with respect to generalized variables, cf. [18], then to the analysis of the finite element characteristics [20] and as solvers for the whole FE system with bilateral or unilateral constraints [5].

The research associated with the above mentioned implementation of hybrid neural-network/finite-element programs is developed at the Institute of Computer Methods in Civil Engineering of the Cracow University of Technology.

ACKNOWLEDGEMENTS

Financial support by the Polish Committee for Scientific Research, Grant No 8 T11F 022 14 "Artificial neural networks in structural mechanics and bone biomechanics" is gratefully acknowledged.

REFERENCES

[1] J. Ghaboussi *et al.* Knowledge-based modeling of material behavior with neural networks. *Journal of Engineering Mechanics*, **117**: 132–153, 1991.

[2] S. Haykin. *Neural Networks – A Comprehensive Foundation.* Macmillan College Publ. Co., Maxwell Macmillan Inter., New York–Toronto, 1994.

[3] P. Hajela, L. Berke. Neurobiological computational models in structural analysis and design. *Computers and Structures*, **41**: 657–667, 1991.

[4] M.N. Jadid, D.F. Fairbairn. Neural-network applications in predicting moment–curvature parameters from experimental data. *Engineering Applications of Artificial Intelligence*, **9**: 309–319, 1996.

[5] S. Kortesis, P.G. Panagiotopoulos. Neural networks for computing in structural analysis – methods and prospects of applications. *Int. J. Num. Mech. Eng.*, **36**: 2305–2318, 1993.

[6] G. Mucha, Z. Waszczyszyn. Hybrid neural-network/computational program for bending analysis of elastoplastic beams, in: A. Garstecki, J. Rakowski, eds., *Computer Methods in Mechanics*, Proc. XIIIth Polish Conf. Computer Meth. Mech., Vol.3, 949–956, Poznań Univ. of Technology, 1997.

[7] D.R.J. Owen *et al.*, Stresses in a partly yielded notched bar – an assessment of three alternative programs. *Int. J. Num. Meth. Eng.*, **6**: 63–73, 1973.

[8] E. Ramm, A. Matzenmiller. Computational aspects of elasto-plasticity in shell analysis. In: D.R.J. Owen *et al.*, eds., *Computational Plasticity – Models, Software and Applications*, P. I, 711–734, Pineridge Press, Swansea, 1987.

[9] R. Rojas. *Neural Networks – A Systematic Introduction.* Springer-Verlag, Berlin–Heildelberg–New York, 1996.

[10] J.C. Simo, R.L. Taylor. A return mapping algorithm for plane stress elastoplasticity. *Int. J. Num. Meth. Eng.*, **22**: 649–670, 1986.

[11] *SNNS – Stuttgart Neural Network Simulator.* User Manual, Version 4.1. Institute for Parallel and Distributed High Performance Systems, Rep. No 6/95, Univ. of Stuttgart, 1995.

[12] Z.P. Szewczyk, A.K. Noor. A hybrid neurocomputing/numerical strategy for nonlinear analysis, *Computers and Structures*, **58**: 661–677, 1996.

[13] Z. Waszczyszyn, ed. *Neural Networks in the Analysis and Design of Structures.* Lecture notes of the CISM Advanced School on Neural Networks in Mechanics of Structures and Materials, Udine, October 19–23, 1998, in preparation for publishing by Springer-Verlag, Wien–New York.

[14] Z. Waszczyszyn. Some new results in applications of backpropagation neural networks in structural and civil engineering. In: B.H.V. Topping, ed., *Advances in Engineering Computational Technology*, 173–187, Civil-Comp Press, Edinburgh, 1998.

[15] Z. Waszczyszyn. Some recent and current problems of neurocomputing in civil and structural engineering. In: B.H.V. Topping, ed., *Advances in Engineering Computational Technology*, 43–58, Civil-Comp Press, Edinburgh, 1996.

[16] Z. Waszczyszyn, C. Cichoń, M. Radwańska. *Stability of Structures by Finite Element Methods*, Elsevier, Amsterdam, 1994.

[17] Z. Waszczyszyn, E. Pabisek, G. Mucha. Hybrid neural-network/computational programs to the analysis of elastic-plastic structures. In: H. Mang, F.G. Rammerstorfer, eds., *Discretization Methods in Structural Mechanics II*, Kluwer Academic Publ., p.10, (in press).

[18] Z. Waszczyszyn, E. Pabisek. Bending analysis of elastic-plastic plates by a hybrid NN/FE program. In: J. Chróścielewski, W. Pietraszkiweicz, eds., *Shell Structures – Theory and Applications*, Proc. 6th Conf. SSTA-98, 277–278, TU Gdańsk, Gdańsk, 1998.

[19] K. Yamamoto. Modeling and hysteric behavior with neural network and its application to nonlinear dynamic response analysis. In: *Application of Artificial Intelligence in Engineering*, Proc. 7th Intern. Conf., 475–486, 1992.

[20] P. Zeng. Artificial neural network computing in structural engineering. In: B.H.V. Topping, ed., *Developments in Neural Networks and Evolutionary Computing for Civil and Structural Engineering*, 37–50, Civil-Comp Press, Edinburgh, 1995.

[21] O.C. Zienkiewicz, S. Valliappan, I.P. King. Elasto-plastic solutions of engineering problems – 'Initial stress', finite element approach, *Inst. J. Num. Meth. Eng.*, **1**: 75–100, 1969.