Computer Assisted Mechanics and Engineering Sciences, **12**: 379–391, 2005.

Neural procedures for the hybrid FEM/NN analysis of elastoplatic plates

Łukasz Kaczmarczyk, Zenon Waszczyszyn
Institute of Computer Methods in Civil Engineering, Cracow University of Technology
Warszawska 24, 31-155 Kraków, Poland, e-mail: lukasz, zenwa@twins.pk.edu.pl

(Received July 28, 2004)

A neural procedure was formulated in [4] as BPNN (Back-Propagation Neural Networks) for the simulation of generalized RMA (Return Mapping Algorithm). This procedure was evaluated to be too large to make a corresponding hybrid FEM/BPNN numerically efficient. That is why two new procedures NP1 and NP2 were formulated. A description of their efficiency is presented in the paper, related to the computation number of computer operations and CPU time, carried out by FEM program FEAP and two hybrid programs FEAP/NP1 and FEAP/NP2.

1. Introduction

Trained NNs (Neural Networks) are usually numerically very efficient in the operational phase. This is a basis to formulate procedures as NNs trained off-line and then insert them into FEM programs. Of course, it is reasonable to design such hybrid FEM/NN programs if they are more efficient than corresponding pure FEM programs. This concerns especially material nonlinear problems which need a complex analysis of constitutive equations. Neural procedures can be also used for the so-called implicit modelling of material relationships [1] which can be then explored in hybrid FEM/NN programs.

The first steps towards the approach discussed above were made in [2], where a neural procedure was used in the analysis of a relatively simple problem of plane elastoplastic stress state. A BPNN (Back-Propagation NN) was formulated for simulation of RMA (Return-Mapping Algorithm) [3], which is applied in the analysis of constitutive equations at each plastically active Gauss point of plane FEs. A simple model of Huber–Mises material with linear strain hardening was assumed. A corresponding numerical procedure was applied to generate a great number of patterns for training and testing of a BPNN. Inserting a trained off-line BPNN into a FE program (in [4] system ANKA [5] was used) gives a hybrid system ANKA/BPNN. It was more numerically efficient than the pure FE program ANKA in which the numerical RMA procedure was used. The numerical efficiency was measured by the decrease of the number of iterations in the Newton-Raphson algorithm on the FE system level.

The approach was developed in [4], where RMA was generalized on cross-sectional level and numerically simulated at Gauss points of reduced integration in bending plate FEs. A great problem for neural simulation was related to numerical generation of patterns for corresponding BPNNs training and testing. The patterns can not be computed only on the base of material constitutive relations but also internal plate constitutive constrains, associated with the Kirchhoff hypotheses, had to be taken into account. The Lobatto quadrature formula was used, cf. [6], for numerical integration of plane stresses along the plate thickness. Despite restriction of the number of Lobatto points to $J = 5$ (because of pure bending only $J^+ = (J-1)/2 = 2$ points with plane stress were considered) a formulated BPNN: 9-40-30-8 was very large since it had 1878 network parameters.

It affected the decrease the numerical efficiency of both the neural procedure and ANKA/BPNN program. In what follows the discussed procedure is called NP [4].

'A priori' approximation of the elastoplastic stresses along the plate thickness and low numerical efficiency of the hybrid system ANKA/BPNN induced us to reformulate the approach from [4]. In the present paper two new neural procedures NP1 and NP2 are discussed. The procedure NP1 follows the approach developed in [4]. The selection of other input and output variables made it possible to diminish significantly the size of corresponding BPNN and increase the number of Lobatto points to $J = 9$, i.e. $J^+ = 4$. This could be achieved generating the training and testing patterns in the so-called subjective approach, i.e. taking patterns from a set of special cases of computed plates. The second procedure NP2 is formulated at the level of Lobatto point. The main goal of this procedure is better prediction of initial values of plastic equivalent strain in internal iterations within a Newton-Raphson procedure inserted in program FEAP. A corresponding hybrid program is called FEAP/NP2.

The main goal of the paper is the analysis of numerical efficiency of two new hybrid systems. This will be done by means of the FEAP program [6] and corresponding hybrid programs FEAP/NP1 and FEAP/NP2. The computer simulations were performed on two plates of geometrical data, boundary conditions, and loads different from those in the examples of plates used for the neural procedures training. The efficiency is evaluated by comparison of CPU time and the number of operations needed for execution of procedures in programs FEAP, FEAP/NP1 and FEAP/NP2.

In order to make the paper easier to read all the assumptions are collected in Sec. 2 and the algorithms associated with neural procedures are included in Appendix.

2. Assumption used in the paper

2.1. Kirchhoff hypotheses, generalized strains and stresses

Only rectangular plates of constant thickness $h =$ const are considered, so Cartesian coordinates x, y, z are applied and $z \in [-h/2, h/2]$.

1. The first Kirchohoff hypothesis implies the following displacement of plate layer at distance z from the midsurface with deflection $w(x, y)$:

$$u = -z\frac{\partial w}{\partial x}, \qquad v = -z\frac{\partial w}{\partial y}; \tag{1}$$

2. Assumption of small rotations of the normal to the plate midsurface leads to a generalized strain vector:

$$\boldsymbol{\kappa} = \{\kappa_x, \kappa_y, \kappa_{xy}\}, \tag{2}$$

$$\kappa_x = -\frac{\partial^2 w}{\partial x^2}, \qquad \kappa_y = -\frac{\partial^2 w}{\partial y^2}, \qquad \kappa_{xy} = -\frac{\partial^2 w}{\partial xy}; \tag{3}$$

3. The second Kirchhoff hypothesis introduces plane stress state in any plate layer and induces a generalized stress vector:

$$\mathbf{m} = \{m_x, m_y, m_{xy}\}, \tag{4}$$

$$m_x = \int_{-h/2}^{h/2} z\sigma_x \mathrm{d}z, \qquad m_y = \int_{-h/2}^{h/2} z\sigma_y \mathrm{d}z, \qquad m_{xy} = \int_{-h/2}^{h/2} z\tau_{xy} \mathrm{d}z. \tag{5}$$

2.2. Elastoplastic material relations

The classical equations of the plane stress state are related to the plastic flow theory and small (infinitesimal) strains are based on the following assumptions (matrix notation and calculus are used in many places of the paper):

1. Total strains (and their rates or increments) can be split into elastic and plastic strains:

$$\boldsymbol{\varepsilon} := \{\varepsilon_x, \varepsilon_y, \gamma_{xy}\} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p; \tag{6}$$

2. Hooke relation for the elastic strains:

$$\boldsymbol{\sigma} := \{\sigma_x, \sigma_y, \tau_{xy}\} = \mathbf{E}\boldsymbol{\varepsilon}^e, \quad \text{where } \mathbf{E} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix}. \tag{7}$$

3. Associated flow rule:

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \dot{\lambda}\mathbf{P}\boldsymbol{\sigma}, \quad \text{where } \mathbf{P} = \frac{1}{3}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix}; \tag{8}$$

4. Huber–Mises yield surfaces (J_2 yield criterion) with linear isotropic strain hardening:

$$F := \sqrt{\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{P}\boldsymbol{\sigma}} - \sqrt{\frac{2}{3}}(\sigma_Y + He^{\mathrm{p}}) = 0, \tag{9}$$

$$\dot{e}^{\mathrm{p}} = \dot{\lambda}\sqrt{\frac{2}{3}\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{P}\boldsymbol{\sigma}}; \tag{10}$$

where: σ_{Y} – yield point, H – strain hardening parameter, e^{p} – effective plastic strain. The parameters σ_{Y}, H are taken from the uniaxial tension test which gives the characteristics of material with the strain hardening moduli E^{p}, shown in Fig. 1a. Parameters H and E^{p} are related to each other by a simple formula:

$$H = \frac{EE^{\mathrm{p}}}{E - E^{\mathrm{p}}}. \tag{11}$$

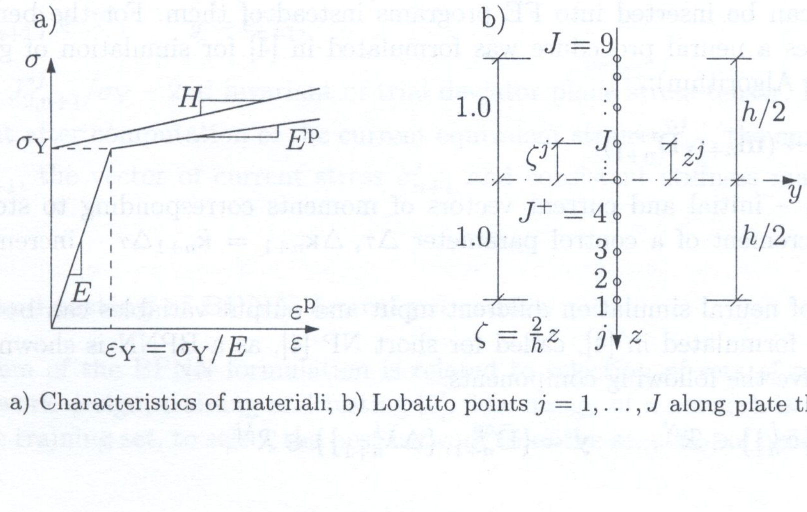


Fig. 1. a) Characteristics of materiali, b) Lobatto points $j = 1, \ldots, J$ along plate thickness

2.3. Numerical integration and discretization along the plate thickness

Because of nonlinear distribution of stress along the plate thickness $\sigma_{\alpha\beta}(z)$ integrals (5) can usually be integrated only by means of quadrature formulas. Following [4] Lobatto formulas [6] are used both for function $f(z)$ and its finite increments $\Delta f = \dot{f}\Delta\tau$, where: $\dot{f}$ – rate of f, τ – control parameter of the considered deformation process [5]. In Fig. 1b Lobatto points $j = 1, \ldots, J$ are shown for $J = 9$ and a corresponding number $J^{+} = (J-1)/2 = 4$ which is used in the analysis of plate pure bending (for antisymmetric functions $f(z)$).

Let us follow the application of Lobatto formula in the calculation of vector of moments $\bar{\mathbf{m}}$ and the cross-sectional tangent stiffness matrix

$$\bar{\mathbf{m}} = \int_{-1}^{1} \bar{\boldsymbol{\sigma}}\zeta \mathrm{d}\zeta \approx 2\sum_{j=1}^{J+} W_j \zeta_j \bar{\boldsymbol{\sigma}}^j, \tag{12}$$

$$\Delta\bar{\mathbf{m}} = \bar{\mathbf{D}}^{\mathrm{ep}}\Delta\bar{\boldsymbol{\kappa}}, \quad \text{where} \quad \bar{\mathbf{D}}^{\mathrm{ep}} \approx 2\sum_{j=1}^{J+} W_j \zeta_j^2 \bar{\mathbf{E}}^{\mathrm{ep}j} \tag{13}$$

and dimensionless variables are used from among listed below:

$$\zeta = \frac{2}{h}z, \quad \bar{\boldsymbol{\sigma}} = \frac{1}{\sigma^{\mathrm{Y}}}\boldsymbol{\sigma}, \quad \bar{\boldsymbol{\varepsilon}} = \frac{E}{\sigma_{\mathrm{Y}}}\boldsymbol{\varepsilon}, \quad \bar{e}^{\mathrm{p}} = \frac{1}{\varepsilon_{\mathrm{Y}}}e^{\mathrm{p}}, \quad \Delta\bar{\boldsymbol{\kappa}} = \frac{h}{2\varepsilon_{\mathrm{Y}}}\Delta\boldsymbol{\kappa},$$
$$\Delta\bar{\lambda} = E\Delta\lambda, \quad \bar{\mathbf{m}} = \frac{4}{\sigma_{\mathrm{Y}}h^2}\mathbf{m}, \quad \bar{\mathbf{D}}^{\mathrm{ep}} = \frac{8}{Eh^2}\mathbf{D}^{\mathrm{ep}}. \tag{14}$$

In Lobatto formulas (12)–(13) W_j are weight parameters [6] and for the sake of simplicity superscript j, denoting Lobatto point, were omitted in (14). Strains and stresses at Lobatto points are computed by formulas:

$$\Delta\bar{\boldsymbol{\varepsilon}}^j = \zeta^j\Delta\bar{\boldsymbol{\kappa}}, \qquad \bar{\boldsymbol{\sigma}}^j = (\bar{\mathbf{E}}^{\mathrm{ep}}\bar{\boldsymbol{\varepsilon}})^j, \tag{15}$$

where: $\bar{\mathbf{E}}^{\mathrm{ep}j}$ – tangent stiffness matrix of plane stress state at Lobatto point.

3. Neural procedures and their training

3.1. Formulation of BPNNs

Neural procedures correspond to trained NNs, formulated as computer simulations of numerical functions which can be inserted into FE programs instead of them. For the bending analysis of elastoplastic plates a neural procedure was formulated in [4] for simulation of generalized RMA (Return Mapping Algorithm):

$$(\mathbf{m}_n, \Delta\boldsymbol{\kappa}_{n+1}) \to (\mathbf{m}_{n+1}, \mathbf{D}^{\mathrm{ep}}_{n+1}), \tag{16}$$

where: $\mathbf{m}_n, \mathbf{m}_{n+1}$ – initial and current vectors of moments corresponding to steps n and $n+1$ related to the increment of a control parameter $\Delta\tau$, $\Delta\boldsymbol{\kappa}_{n+1} = \dot{\boldsymbol{\kappa}}_{n+1}\Delta\tau$ – increment of vector of curvatures (2).

For purposes of neural simulation different input and output variables can be selected. In Fig. 2a the procedure formulated in [4], called for short NP [4], as a BPNN is shown. The input and output vectors have the following components:

$$\mathbf{x} = \{\Delta\bar{\boldsymbol{\kappa}}_{n+1}, \{\bar{\boldsymbol{\sigma}}^j_n\}\} \in \mathcal{R}^N, \qquad \mathbf{y} = \{\bar{\mathbf{D}}^{\mathrm{ep}}_{n+1}, \{\Delta\bar{\lambda}^j_{n+1}\}\} \in \mathcal{R}^M, \tag{17}$$

where: $N = 3 + 3J^{+}$, $M = 6 + J^{+}$ – dimension of input and output spaces. The increment of curvatures $\Delta\bar{\boldsymbol{\kappa}}_{n+1}$ and initial stresses at Lobatto points $\{\bar{\boldsymbol{\sigma}}_n^j\}$ are used as BPNN inputs. The consistent cross-sectional stiffness matrix $\bar{\mathbf{D}}_{n+1}^{\text{ep}}$ and increments of yielding multiplier $\{\Delta\bar{\lambda}_{n+1}^j\}$ at Lobatto points are computed as BPNN outputs. Then using formulas shown in Appendix the current stress $\{\boldsymbol{\sigma}_{n+1}^j\}$ at Lobatto points can be computed and then current moments $\mathbf{m}_{n+1}$.

The neural procedure NP1 has the following input and output vectors, Fig. 2b:

$$\mathbf{x} = \{\bar{\mathbf{m}}_{n+1}^{*}, \{\bar{e}_n^{\text{p}j}\}\} \in \mathcal{R}^N, \qquad \mathbf{y} = \{\bar{\mathbf{m}}_{n+1}, \{\Delta\bar{e}_{n+1}^{\text{p}j}\}\} \in \mathcal{R}^M, \tag{18}$$

where: $\bar{\mathbf{m}}_{n+1}^{*}$ – vector of trail elastic moments in generalized RMA (16), $\bar{e}_n^{\text{p}j}$, $\Delta\bar{e}_{n+1}^{\text{p}j}$ – initial equivalent strain and its increment at Lobatto point, $N = M = 3 + J^{+}$ – number of inputs and outputs. Having the values of outputs the current effective stress at Lobatto points $\{\boldsymbol{\sigma}_{n+1}^j\}$ can be computed using the algorithm discussed in Appendix, as well as the consistent stiffness matrices $\mathbf{E}_{n+1}^{\text{ep}j}$ and $\mathbf{D}_{n+1}^{\text{ep}}$.

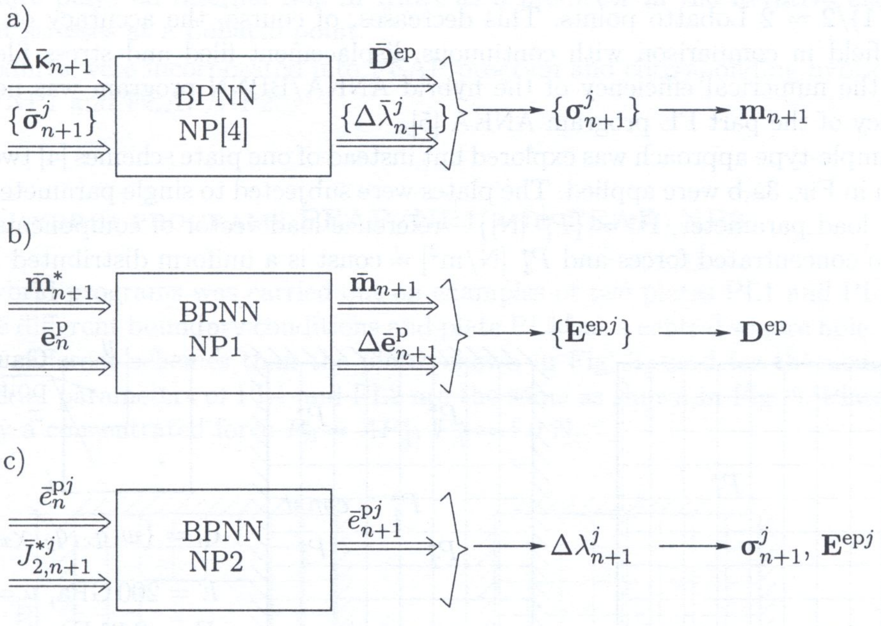

a)
$\Delta\bar{\boldsymbol{\kappa}}_{n+1}$
$\{\bar{\boldsymbol{\sigma}}_{n+1}^j\}$
BPNN NP[4]
$\bar{\mathbf{D}}^{\text{ep}}$
$\{\Delta\bar{\lambda}_{n+1}^j\}$
$\{\boldsymbol{\sigma}_{n+1}^j\}$
$\mathbf{m}_{n+1}$
b)
$\bar{\mathbf{m}}_{n+1}^{*}$
$\bar{\mathbf{e}}_n^{\text{p}}$
BPNN NP1
$\bar{\mathbf{m}}_{n+1}$
$\Delta\bar{\mathbf{e}}_{n+1}^{\text{p}}$
$\{\mathbf{E}^{\text{ep}j}\}$
$\mathbf{D}^{\text{ep}}$
c)
$\bar{e}_n^{\text{p}j}$
$\bar{J}_{2,n+1}^{*j}$
BPNN NP2
$\bar{e}_{n+1}^{\text{p}j}$
$\Delta\lambda_{n+1}^j$
$\boldsymbol{\sigma}_{n+1}^j$, $\mathbf{E}^{\text{ep}j}$

Fig. 2. BPNNs used in procedure NP[4] and in new procedures NP1 and NP2

The neural procedure NP2 was formulated to be used at each Lobatto point. NP2 is based on a simple BPNN whose input and output vectors are shown in Fig. 2c, i.e:

$$\mathbf{x} = \{\bar{e}_n^{\text{p}j}, \bar{J}_{2,n+1}^{*j}\} \in \mathcal{R}^2, \qquad y = \bar{e}_{n+1}^{\text{p}j}, \tag{19}$$

where: $\bar{J}_{2,n+1}^{*j} = J_{2,n+1}^{*j}/\sigma_Y$ – 2nd invariant of trial deviator plane stress tensor. It has been shown in Appendix that after computation of the current equivalent strain $e_{n+1}^{\text{p}j}$, the current value plastic increment $\Delta\lambda_{n+1}^j$, the vector of current stress σ_{n+1}^j and consistent stiffness matrix $\mathbf{E}_{n+1}^{\text{ep}j}$ can be calculated.

3.2. Training and testing of BPNN neural procedure

The main problem of the BPNN formulation is related to selection of sets of patterns which are used for the network design, training and testing [7]. The design of a network is usually preformed on subsets of the training set, to select the best network from the sequence of candidate BPNNs [8].

The quality of such a BPNN is measured by different network errors [8]. Then the validated BPNNs are trained on the full training set of patterns and tested on an independent testing set.

In [2, 4] two approaches were used for generating patterns by numerical procedure or a FEM program. The firs approach was called a "subjective" or "example-type" approach since the patterns were taken from FE computations at several schemes of plates. The second approach was called an "objective" or "constitutive-type" since patterns could be generated only on the base of constitutive equations. The constitutive-type patterns are more general since they are independent of geometrical data, boundary conditions and loading patterns of the plate schemes used in the first approach.

The constitutive-type approach was successfully explored in the analysis of a plane stress problem [2] but it was difficult to generate it over cross-section of the bending plate. A serious difficulty is related to the Kirchhoff constraints which couple constitutive equation of all the Lobatto points. An attempt was made to overcome in [4] by considering special loading paths but they had to be related, in fact, to selected plate schemes in order to generate FEM patterns.

As a result of research discussed in [4] a large network BPNN: 9-40-38-8 was formulated using $J = 5$ Lobatto points, i.e. considering coupling constitutive equations at two plate layers corresponding to $J^+(5-1)/2 = 2$ Lobatto points. This decreases, of course, the accuracy of approximation of the stress field in comparison with continuous displacement filed and stress filed plate bending. Moreover, the numerical efficiency of the hybrid ANKA/BPNN program was nearly the same as the efficiency of the part FE program ANKA [5].

The example-type approach was explored but instead of one plate schemes [4] two plate schemes of data shown in Fig. 3a,b were applied. The plates were subjected to single parameter load $\mathbf{P} = \Lambda\,\mathbf{P}^*$, where: Λ – load parameter, $\mathbf{P}^* = \{P_i^*\ [\mathrm{N}]\}$ – reference load vector of components $i = 1, 2, 3$ corresponding to concentrated forces and $P_4^*\ [\mathrm{N/m^2}] = \mathrm{const}$ is a uniform distributed load.

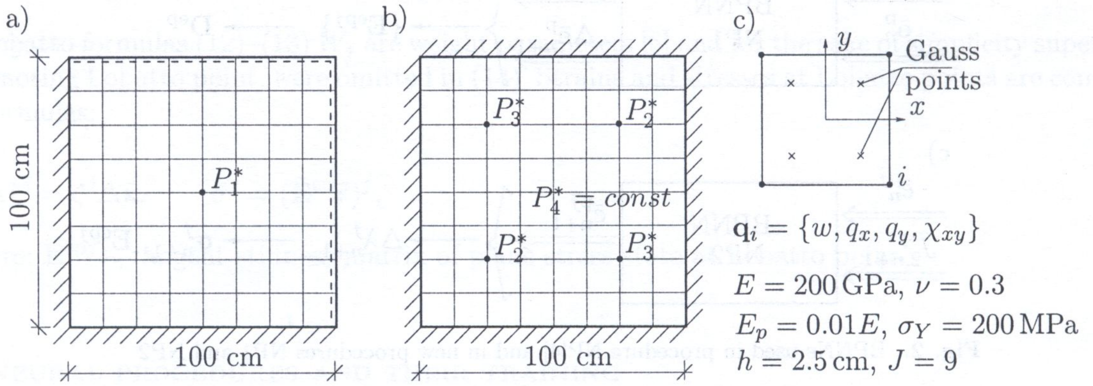

a)
b)
c)
100 cm
100 cm
100 cm
P_1^*
P_3^*
P_2^*
$P_4^* = const$
P_2^*
P_3^*
y
x
Gauss points
i
$\mathbf{q}_i = \{w, q_x, q_y, \chi_{xy}\}$
$E = 200\,\mathrm{GPa},\ \nu = 0.3$
$E_p = 0.01E,\ \sigma_Y = 200\,\mathrm{MPa}$
$h = 2.5\,\mathrm{cm},\ J = 9$

Fig. 3. a, b) Schemes of plates, c) 4-node element of 16 DOF with four Gauss points of reduced integration

Square, 4-node compatible FE with 4×4 Gauss points of reduced integration and $J = 9$ Lobatto points were assumed (i.e. $J^+ = 4$ is twice higher than $J^+ = 2$ in [4]), cf. Fig. 3c. The patterns were generated in 384 Gauss points using six combinations of reference loads $|P_i^*| = 1.0$ N and variables load parameter step $\Delta\Lambda$ to reach $\Lambda \leq 260$. In case of NP1 procedure $P = 1.5e5$ patterns were generated by means of FEAP program [6]. A set of $L = 2.0e4$ patterns was randomly selected for the network training and the remaining $T = 1.3e5$ patterns were used for the network testing.

The candidate network for the procedure NP1 were assumed to be BPNNs of structure 7-H-7, corresponding to $N = M = 7$ inputs and outputs, cf. (18). One hidden layer of $H = 5, \ldots, 20$ bipolar sigmoidal and identity linear output activation functions [8] were assumed. The NN simulator [9] was used and Rprop learning method [8] was explored. The initial parameters of considered BPNNs were evaluated by the simulated annealing method using procedures from [9].

A standard cross-validation method was used for the design of an optimal number of hidden neurons. The design process started from $H = 5$ and then the computation process $H + 5 \to H$

started. For the trained network BPNN: 7-15-7 the optimal value $H = 15$ gave the mean square error [8] $MSEL \approx MSET \approx 0.002$.

The design of neural procedure NP2 was performed on the base of patterns of generated by above mentioned plate schemes. A set of $6.7e5$ patterns was generated at four Lobatto points corresponding to $J^+ = 4$. From among them $L = 1.0e4.0$ patterns were randomly selected for the training of candidate networks BPNN: 2-H-1. Following the cross-validation procedures applied to design of NP1 the network BPNN: 2-15-1 was the optimal one (network errors were $MSEL \approx MSET \approx 0.001$).

The designed procedures NP1 and NP2 correspond to small networks BPNN: 7-15-7 and BPNN: 2-15-1, respectively. Similarly as in [4], procedure NP1 simulates action of the generalized RMA. The neural procedure from [4], NP [4], corresponds to a large BPNN: 9-40-30-8 with 1878 network parameters and BPNN: 7-15-7 in NP1 has only 232 parameters. It is worth emphasizing that NP1 was formulated for $J = 9$ Lobatto points and NP [4] for $J = 5$.

The procedure NP2 is applied not at the plate cross-sectional level but at the Lobatto point level. The procedure plays an internal role in RMA as a predictor in the iterative algorithm for computing current stresses at a Lobatto point.

The new procedures were incorporated into FEAP program and corresponding hybrid programs are called FEAP/NP1 and FEAP/NP2.

4. Testing of hybrid programs FEAP/NP1 and FEAP/NP2

Testing of new hybrid programs was carried out on examples of two plates PL1 and PL2 shown in Fig. 4. Plates have different boundary conditions and plate PL1 has a central square hole. The plates PL1 and PL2 are different schemes than the plates shown in Fig. 3, used for the neural network training. All material parameters of PL1 and PL2 are the same as shown in Fig. 3. Plates PL1 and PL2 are loaded by a concentrated force $P_A = \Lambda P_A^*$, $P_A^* = 1.0$ N.

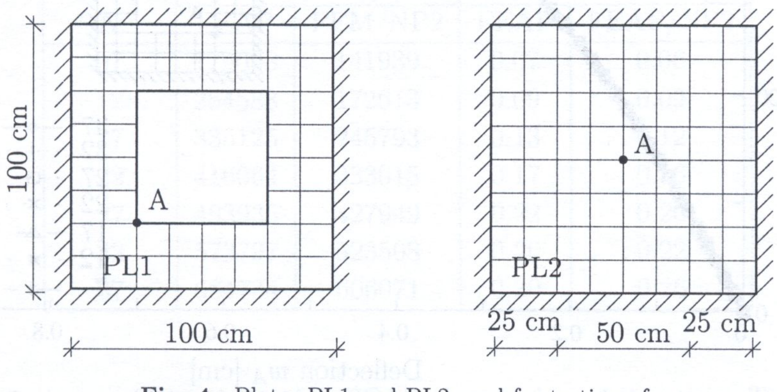


Fig. 4. Plates PL1 and PL2 used for testing of hybrid programs FEAP/NP1 and FEAP/NP2

In Fig. 5 there are shown equilibrium paths $P_\text{A} - w_\text{A}$ for plates PL1 and PL2 computed by the program FEAP and hybrid program FEAP/NP1. There no are differences between the results by FEAP and FEAP/NP2 since the procedure NP2 can not change the results obtained by FEAP – the procedure can only reduce the number of iterations steps in at the Lobatto point level.

In Fig. 6 there is shown the influence of the number $N = 7, 12, \ldots, 37$ of load parameter increments $\Delta\Lambda$ (or ΔP for $P_\text{A}^* = 1.0\,\text{N}$) on the equilibrium points $P_\text{A} - w_\text{A}$ for plate PL1. It is visible that all the points corresponding to different length of $\Delta\Lambda$ are placed at practically the same curve $P_\text{A}(w_\text{A})$.

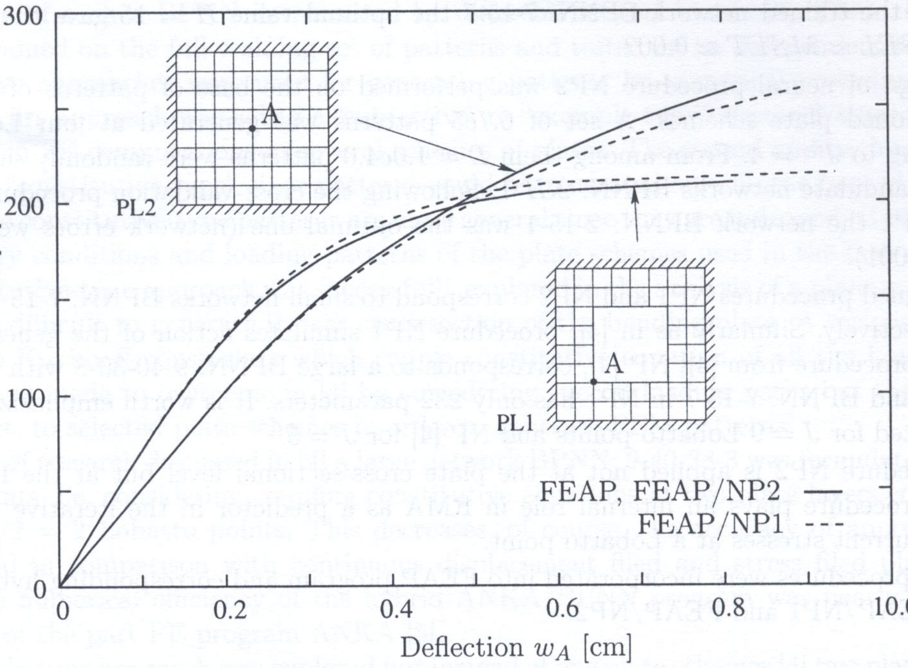


Fig. 5. Equilibrium paths $P_A - w_A$ for plates PL1 and PL2 computed by different computer programs

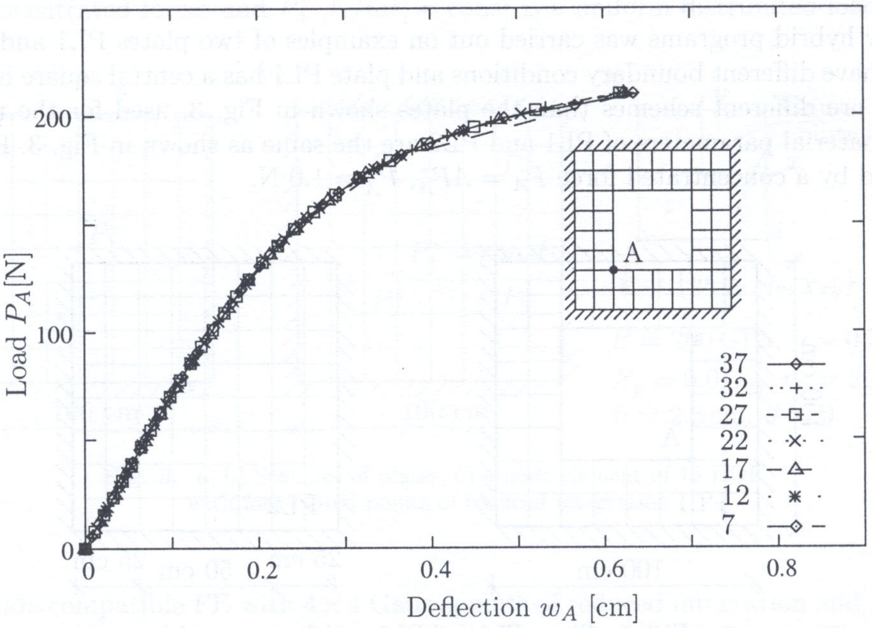


Fig. 6. Equilibrium paths points of plate PL1 corresponding to computed by FEAP and FEAP/NP2 the number of load increments $N = 7, 12, \ldots, 37$

5. Comparison of results of computations by programs FEAP, FEAP/NP1 and FEAP/NP2

Due to the use of consistent stiffness matrices the iteration is quadratically convergent for each iteration step in the Newton–Raphson method [3]. The neural simulations are related either to the cross-section level (procedure NP1) or to the Lobatto point level (procedure NP2). That means that the numerical efficiency of neural procedures can be measured with respect to the number of internal iterations performed within each Newton–Raphson step. Because of a very short execution time per one iteration of procedures NP1 and NP2 the total number of iterations or CPU times were counted, associated with one equilibrium path using different number N of increment steps.

The computations were performed on Intel Pentium 4 CPU 2.00 GHz using the program FEAP, FEAP/NP1 and FEAP/NP2. In what follows the total number of internal TIT or the time of their execution TIME [sec] are given as associated with computation of N incremental steps. The exact number of iteration TIT was counted by the GPROF program [11] but TIME could be only roughly estimated. Of course, the corresponding figures for FEAP are associated with the same iterations which were simulated by FEAP/NP1 and FEAP/NP2.

In Table 1 there are shown results for $N = 37$ steps for both tested plates PL1 and PL2. On the base of figures listed in Table 1 the computational efficiency of procedures NP1 and NP2 measured by decrease of number of iterations or CPU time in comparison with numerical procedures explored in FEAP, can be evaluated as 1.8%–20%.

Table 1. Total number of internal iterations TIT and their execution total time TIME [sec] for $N = 37$ loading steps

Plate	TIT for:		TIME [sec] for:		
	FEAP	FEAP/NP2	FEAP	FEAP/NP1	FEAP/NP2
PL1	664734	606071	0.30	0.24	0.26
PL2	447397	438931	0.48	0.43	0.45

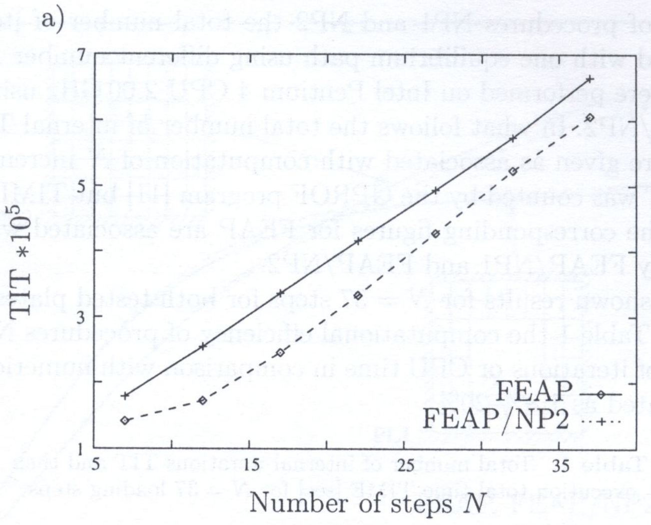
A more detailed analysis was made for the efficiency of procedure NP2. In Table 2 and Fig. 7 there are shown results for TIT and TIME depending on the number of loading steps N.

Table 2. Total number of internal iterations TIT and their execution time TIME [sec] depending on the number of loading steps N for plate PL1

Number of steps	TIT for:		TIME [sec] for:	
N	FEAP	FEM/NP2	FEAP	FEAP/NP2
07	179008	141939	0.07	0.06
12	254588	172613	0.09	0.09
17	336125	245793	0.13	0.12
22	416064	333015	0.17	0.16
27	493935	427949	0.22	0.20
32	573797	523508	0.26	0.22
37	664734	606071	0.30	0.26

From the Fig. 7a it is evident that using FEAP the number of iterations TIT depends linearly on the number of loading steps N. A deviation of TIME from linear dependence on N, cf. Fig. 7b, is caused by difficulty with measuring the execution time. The graphics TIT(N) and TIME(N) made for FEAP/NP2 are affected by the neural simulation errors. Taking into account the discussed effects the numerical efficiency of neural procedure NP2 inserted into FEAP/NP2 can be roughly evaluated as decreasing by about 17% the number of internal iterations made by the corresponding numerical procedure inserted into FEAP.

The main question that should be answered concerns the numerical efficiency of the hybrid procedures from the point view of execution time whole programs. It was stated that in case of FEAP the number of total internal iterations TIT makes up about 4% of the total execution time. From this point of view the total efficiency of hybrid programs FEAP/NP1 and FEAP/NP2 can be evaluated by about 0.7% decrease of the time executed by FEAP.



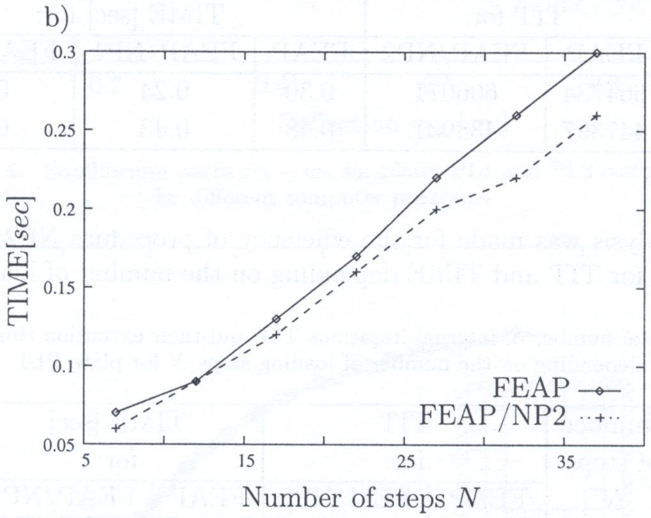


Fig. 7. a) Total number of internal iterations TIT and b) Total time TIME [sec] for execution of TIT for plate PL1 and different loading steps N

5.1. Final remarks and some conclusions

In the paper two new neural procedures NP1 and NP2 have been formulated. The first procedure NP1 follows the approach in [4], where a procedure called for short NP [4] also simulates a generalized RMA (Return Mapping Algorithm) on the cross-sectional level. The main goal of the formulated NP1 has been to eliminate defects of NP [4], i.e. a large size of corresponding network BPNN formulated in [4] and its dependence on the approximation of stress field distribution along the plate thickness by $J = 5$ Lobatto point quadrature formula.

The second procedure NP2 has been formulated on the Lobatto point level, cf. Appendix, in order to predict better the initial values in the iterative computation of stresses at the considered Lobatto point.

Both procedures NP1 and NP2 were trained on patterns generated by the FE program FEAP using the "example-type" approach related to the computation of six schemes of elastoplastic plate bending. After training the neural procedures were inserted into the hybrid programs FEAP/NP1 and FEAP/NP2.

The numerical efficiency of procedures NP1 and NP2 has been examined by computation of two plates PL1 and PL2 by means of programs FEAP, FEAP/NP1 and FEAP/NP2. The numerical efficiency has been defined a as decrease of, the number of total internal iterations TIT (or the corresponding execution time) carried out by means of neural procedures inserted into FEAP/NP1 or FEAP/NP2 in comparison with TIT execute by FEAP.

It has been stated that the numerical efficiency of hybrid programs, i.e. also numerical procedures, is on average about 17%. From the viewpoint of the execution of total program the efficiency of hybrid programs is significantly lower since it can by evaluated as a decrease by about 0.7% of the number of numerical operations (or corresponding computational time) performed.

The results mentioned above were obtained for a very simple elastoplastic material (initially isotropic with the Huber–Misis–Hencky yield function surface, isotropic strain hardening and associated flow rule). In cases of more complicated materials, the neural procedures can also be formulated [10] and their numerical efficiency and efficiency of corresponding hybrid programs may be much more prominent.

Appendix

A.1. Computation following procedure NP [4]

The algorithm is related to the input and output vector (17) which can be written in following form:

$$\mathbf{x} = \{\Delta\boldsymbol{\kappa}_{n+1}, \boldsymbol{\sigma}_n\}, \quad \mathbf{y}_{(8x1)} = \{\Delta\boldsymbol{\lambda}_{n+1}, \mathbf{D}^{\mathrm{ep}}_{n+1}\}, \tag{A1}$$

where: n, $n+1$ – starting and current states, $\boldsymbol{\sigma}_n = \{\sigma^j\}_n$ – vector of stress at $j = 1, \ldots, J^+$ Lobatto points, $\Delta\boldsymbol{\lambda}_{n+1} = \{\Delta\lambda^j\}_{n+1}$ – vector of yielding parameter increments at J^+ Lobatto points. The algorithm of computation $\boldsymbol{\sigma}_{n+1}$ and $\mathbf{m}_{n+1}$ starts from known values of NP [4] procedure, Fig. 2a and formulas related to RMA [3]:

1. At each Lobatto point $j = 1, \ldots, J^+$ compute algorithmic stiffness matrix

$$\boldsymbol{\Xi}^j := [\mathbf{E}^{-1} + \Delta\lambda^j\mathbf{P}]^{-1}; \tag{A2}$$

2. Compute vectors of consistent stress and current stress:

$$\begin{aligned} c\Delta\boldsymbol{\sigma}^j_{n+1} &= \boldsymbol{\Xi}(\Delta\lambda^j), \\ \boldsymbol{\sigma}^j_{n+1} &= \boldsymbol{\sigma}^j_n + \Delta\boldsymbol{\sigma}^j_{n+1}; \end{aligned} \tag{A3}$$

3. Using Lobatto formula 12 compute vector of cross-sectional generalized stress (vector of moments)

$$\mathbf{m}_{n+1} = \sum_{j=1}^{J+} W_j \boldsymbol{\sigma}^j_{n+1} z^j. \tag{A4}$$

A.2. Neural algorithm related to neural procedure NP1

The following input and output vectors are accepted in neural procedure NP1, cf.(18):

$$\mathbf{x} = \{\mathbf{m}^*_{n+1}, \mathbf{e}^{\mathrm{p}}_n\}, \qquad \mathbf{y} = \{\mathbf{m}_{n+1}, \Delta\mathbf{e}^{\mathrm{p}}_{n+1}\}, \tag{A5}$$

where: $\mathbf{e}^{\mathrm{p}}_n = \{e^{\mathrm{p}j}_n\}$, $\mathbf{e}^{\mathrm{p}}_{n+1} = \{e^{\mathrm{p}j}_{n+1}\}$ – vectors of equivalent strains at Lobatto points $j = 1, \ldots, J^+$.

The algorithm for inputs and outputs vectors (A5) enables us to make computation at Gauss points. The algorithm for FEAP/NP2 has sequence following steps:

1. Table vector of current curvature increments $\Delta\boldsymbol{\kappa}_{n+1}$ and compute trail stress

$$\mathbf{m}^*_{n+1} = \mathbf{m}_n + \mathbf{D}\Delta\boldsymbol{\kappa}_{n+1}; \tag{A6}$$

2. At each Lobatto point j increment of equivalent strain is simulated by NP2 output $\Delta\mathbf{e}^{\mathrm{p}}_{n+1} = \{\Delta e^{\mathrm{p}j}_{n+1}\}$ so compute

$$\mathbf{e}^{\mathrm{p}}_{n+1} = \mathbf{e}^{\mathrm{p}}_n + \Delta\mathbf{e}^{\mathrm{p}}_{n+1}, \qquad \Delta\lambda^j = \frac{e^{\mathrm{p}j}_{n+1}}{\sigma_{\mathrm{Y}} + He^{\mathrm{p},j}_n}, \tag{A7}$$

where: σ_Y, H – yield point and strain hardening modulus of material characteristics shown in Fig 1a;

3. Using (A2) compute Ξ^j;

4. Using (A3) compute $\boldsymbol{\sigma}^J_{n+1}$;

5. Compute consistent elastoplastic matrix $\mathbf{E}^{\mathrm{ep}j}$ at Lobatto points j:

$$\mathbf{a}^j_{n+1} = \mathbf{P}\boldsymbol{\sigma}^j_{n+1}, \qquad \mathbf{b}^j_{n+1} = \Xi^j\mathbf{a}^j_{n+1}, \tag{A8}$$

$$\beta^j_{n+1} = \frac{\frac{2}{3}H\mathbf{a}^{\mathrm{T}j}_{n+1}\boldsymbol{\sigma}^j_{n+1}}{1 - \frac{2}{3}H\Delta\lambda^j}, \tag{A9}$$

$$\mathbf{E}^{\mathrm{ep}j} = \Xi^j - \mathbf{b}^j_{n+1}\mathbf{b}^{\mathrm{T}j}_{n+1}/(\beta^j_{n+1} + \mathbf{a}^{\mathrm{T}j}_{n+1}\mathbf{b}^j_{n+1}); \tag{A10}$$

6. Compute consistent cross-sectional, elastoplastic stiffness matrix using (13):

$$\mathbf{D}^{\mathrm{ep}} = 2\sum_{j=1}^{J+} W_j z_j^2 \mathbf{E}^{\mathrm{ep}j}. \tag{A11}$$

A.3. NP2 as predictor of initial value of $\Delta\lambda^j$

Neural network is used at each Lobatto point j. According to (19) it has the following input vector and scalar output:

$$\mathbf{x} = \{e^{\mathrm{p}j}_n, J^{*j}_{2,n+1}\}, \qquad y = e^{\mathrm{p}j}_{n+1}. \tag{A12}$$

Using NP2 the following algorithm is preformed:

1. Compute current stress and strain vector

$$\boldsymbol{\varepsilon}^j_{n+1} = \boldsymbol{\varepsilon}^j_n + \Delta\boldsymbol{\kappa}^j_{n+1}, \qquad \boldsymbol{\sigma}^*_{n+1} = \mathbf{E}(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_n); \tag{A13}$$

2. If $F^{*j}_{n+1} = J^{*j}_{2,n+1} - \sqrt{\frac{2}{3}}(\sigma_Y + He^{\mathrm{p}}) < 0$ then plastic process is passive, otherwise plastic process is active so is necessary to find consistent plastic multiplayer $\Delta\lambda^j$. Equation $F(\Delta\lambda^j) = 0$ is solved by employing Newton's method, whose initial value for an iterative solution is computed by NP1. From NP1 $e^{\mathrm{p}j}_{n+1}$ is computed and next from (A8) initial (predictor) value of plastic multiplier.

3. Using (A2) compute Ξ^j;

4. Compute consistent elastoplastic matrix $\mathbf{E}^{\mathrm{ep}j}$ at Lobatto points j from (A10).

ACKNOWLEDGMENTS

Financial support by the Polish Committee for Scientific Research, Grant No. 8T07A 031 20 "Hybrid analysis of elastoplastic structures basing on interaction FEM and artificial neural networks" is gratefully acknowledged.

REFERENCES

[1] T. Furukawa, G. Yagawa. Implicit constitutive modelling for viscoplasticity using neural networks, *Int. J. Num. Meth. Eng.*, **43**: 195–219, 1998.

[2] Z. Waszczyszyn, E. Pabisek. Hybrid NN/FEM analysis of elatoplastic plane stress problem, *Comp. Assisti. Mech. Eng. Sci.*, **6**: 177–188, 1999.

[3] J.C. Simo, T.J.R. Hughes. *Computational Inelasticity,* Springer-Verlag, New York, 1998.

[4] Z. Waszczyszyn, E. Pabisek. Neural network supported FEM analysis of elastoplastic plate binding, *Research News*, Budapest UTE, Special Issue 2000/4: 12–19, 2000.

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

[6] R. Taylor. *FEAP – A Finite Element Analysis Program.* Version 7.4 Theory Manual, Univ. of California at Berkley, 2002.

[7] S. Haykin. *Neural Networks – A Systematic Introduction*, 2nd Ed., John Wiley, 1999.

[8] Z. Waszczyszyn (Ed.). *Neural Networks in the Analysis and Design of structures* CISM Courses and Notes No. 404, Springer, Wien – New York, 1999.

[9] A. Zell (Ed.) *SNNS – Stuttgart Neural Simulator*, User's Manual, Version 4.2, Univ. of Stuttgart and Univ. Tübingen, 1998

[10] R. Haj-Ali, D.A. Pecknold, J. Ghaboussi, G.Z. Voyiadjis. Simulated micromechanical models using artificial neural networks, *J. Eng. Mech.*, **127**: 730–738, 2001.

[11] http://www.gnu.org/manual/gprof-2.9.1/gprof.html