ANN REPRESENTATION OF FINITE ELEMENT METHOD FOR GROUNDWATER FLOW SIMULATION

Lars Weber[1], K.-P. Holz[2]

ABSTRACT

Artificial neural networks (ANN) are increasingly in use in hydraulic engineering. They represent a data-driven technique which serves for determining nonlinear transfer functions between input to (action) and output from (reaction) physical systems. The finite element (FE) method on the contrary is a numerical technique to solve deterministically described physical processes.

Within current research it is investigated how and whether numerical FE-modeling can be supported by the use of neural networks. The approach is taken on the on the level of an individual finite element. The FE coefficient matrices are generated numerically for different geometrical configurations and order of interpolation functions. Then ANNs are used to train memories and forecast the coefficients on the element matrix level. Thus a hybrid representation of deterministically described physical processes within numerical models is obtained.

The investigations are performed for groundwater flow simulation. Aspects of element geometry, element configurations, order of trial function and material properties are considered for triangular elements.

KEY WORDS

neural networks, finite element method, groundwater flow, Poisson and Laplace equation

INTRODUCTION

Artificial neural networks are frequently used for pattern and image recognition, for signal processing and data compression, for black-box representation of physical systems and mapping of general functions etc. (Patterson 1996). They are applied in different fields such as mechanics of structures and materials (Waszczyszyn / Ziemianski, 2001) and hydraulic engineering. Typical applications are predictions of rainfall-runoff in river basins (Mason/Price/Tem´me 1996, Minns 1996), on-line forecast of high-water levels for flood protection (Bazartseren/Holz 2003) and the prediction of flow in two-dimensional shallow-

[1] Research Engineer: Institute fuer Bauinformatik, Brandenburg University of Technology at Cottbus, Konrad-Wachsmann-Allee 1, 03044 Cottbus, Germany, Phone (+49) 355/69-2008, FAX (+49) 355/69-2188, weber@bauinf.tu-cottbus.de

[2] Professor Institute fuer Bauinformatik, Brandenburg University of Technology. at Cottbus, Konrad-Wachsmann-Allee 1, 03044 Cottbus, Germany, Phone (+49) 355/69-2262, FAX (+49) 355/69-2188, holz@bauinf.tu-cottbus.de

water systems (Dibike/Abbott 1999), (Becker/Delgado, 2001). The nonlinear transfer functions between input and output from the physical systems were presented by the ANN.

Here another application of ANN will be considered.

In FE-modeling huge numbers of element matrices have to be calculated numerically within each simulation run. Elements differ by geometry but hardly by physics and approximation functions within a given application determined by sets of partial differential equations. So it seems worth considering whether computational savings could be obtained by "forecasting" the integrated element coefficient matrices by ANN. This leads to a hybrid approach on the level of applied mathematical methods. Finite element coefficient matrices may be generated either by standard numerical integration techniques or by "forecast" using ANN in a mixed environment. Solving global equation systems of typical FE-modeling systems is not affected.

The terminology "hybrid" is applied in analogy to approaches in the past mixing digital and analog computers for simulation of system (groundwater) or physical and numerical discrete models for river flow (Holz, 1976), (Funke / Croshank, 1978). Essential element was the dynamic interaction of both techniques / sub-models within one simulation

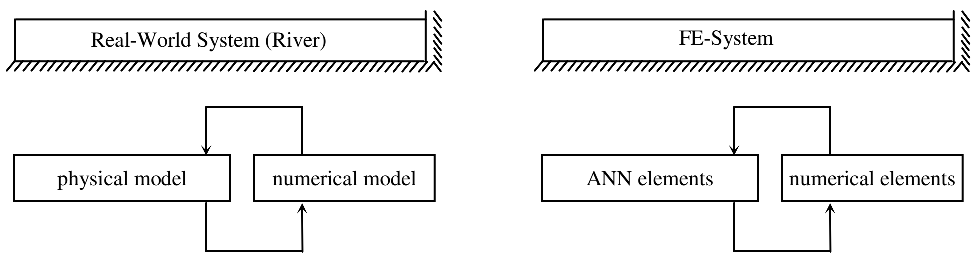


Figure 1: Principle of hybrid model

The investigations are performed for solving the Poisson equation, here representing groundwater flow systems. This application has been chosen deliberately rather for checking the accuracy of the approach than because of simulating physics. Errors can be analyzed because of existence of a variation principle for this class of problems. Finite element formulations tested are based on triangular elements. Linear and quadratic approximation functions are investigated.

NUMERICAL MODEL

As example for testing the approach, a potential flow problem is considered. In this case groundwater flow has been selected. Starting from Darcy's Filter Law $q = -C \cdot \boldsymbol{d}H$, where q is the volumetric flow, C representing hydraulic conductivity and $\mathbf{d}H$ the gradient of the hydraulic pressure and introducing this into the mass conservation equation, within which w represents a source or sink term, the differential equations are obtained. In matrix form holds

$$\boldsymbol{d}^T \cdot (-\mathbf{C} \cdot \boldsymbol{d}H) = w \ ; \qquad \boldsymbol{d}^T = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}\right].$$

This equation may alternatively be generated starting from a variation formulation in terms of the potential function $\prod$.

$$\delta\prod = \delta\int (\boldsymbol{d}^T H \cdot \mathbf{C} \cdot \boldsymbol{d}\, H)\, dA = 0$$

The variation returns, omitting the boundary integral

$$\int \delta\,(\boldsymbol{d}H)^T \cdot \mathbf{C} \cdot (\boldsymbol{d}H)\; dA - \int \delta\, H \cdot w\; dA = 0\,.$$

The potential may be used as a measure for error analysis after numerical discrete and neural hybrid solution.

Potential problems are conveniently solved by the finite element (FE) method. The state variable H, which represents physics, is interpolated by a trial function on the base of an element of given geometry. The variation process leads to an equation system in terms of the parameters of the trial functions. For the investigations performed, polynomial trial functions of Lagrange type and triangular finite elements are used. On the global level the state variable is represented in global coordinates $H(x,z)$ which are substituted by natural coordinates $H(\lambda_1,\lambda_2,\lambda_3)$ on the element level e. For any arbitrary trial function of Lagrange type $\Omega_e(\lambda_i)$ with $i=1,2,3$ holds $\Omega_e(\lambda) = \boldsymbol{h}_e^T \cdot \boldsymbol{s}(\lambda_i)$, where $\boldsymbol{h}_e$ is the support vector at the individual element multiplied by the trial function written in natural coordinates $\boldsymbol{s}(\lambda_i)$. The size of the support vector is determined by the order of the trial function. The gradient of the potential field $(\mathbf{d}H)$ can be described by:

$$\boldsymbol{d}H(x) = \boldsymbol{\lambda}_x{}^T \cdot \boldsymbol{S}_\lambda{}^T \cdot \boldsymbol{h}_e\,,$$

where $\boldsymbol{S}_\lambda$ is gradient of the trial functions and $\boldsymbol{\lambda}_X$ the derivative of natural coordinates with respect to global coordinates. All elements have to be assembled into the global full system which is done by summation over all elements. The integration is substituted by summation giving

$$\sum_e \int \delta\,\boldsymbol{h}_e{}^T \cdot \boldsymbol{S}_\lambda \cdot \boldsymbol{\lambda}_x \cdot \mathbf{C} \cdot \boldsymbol{\lambda}_x{}^T \cdot \boldsymbol{S}_\lambda{}^T \cdot \boldsymbol{h}_e\; dA = \sum_e \int (\delta\,\boldsymbol{h}_e^T \cdot \boldsymbol{s} \cdot \boldsymbol{w})\; dA$$

This formulation represents an equation system which can be solved after application of proper boundary conditions. Of special interest within the context of this paper is the contribution of the individual element represented by the element matrix

$$\boldsymbol{K}_e = \int (\boldsymbol{S}_\lambda \cdot \boldsymbol{\lambda}_x) \cdot \mathbf{C} \cdot (\boldsymbol{\lambda}_x{}^T \cdot \boldsymbol{S}_\lambda{}^T)\; dA\,,$$

which is to be substituted by a neural network representation.

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are simplified models of the human brain. ANN can be classified by their architecture (single layer, multi layer and recursive), learning method (supervised, unsupervised etc.), type of learning (Hebbian learning, error correction learning etc.) and type of usage (optimization, associative memory, prediction etc.) (Patterson 1996).

In this research project a multi layer feed forward back propagation network with supervised learning is used which is rather common for engineering problems. Neural networks are generally mapping an input vector $\boldsymbol{x}^{(p)}$ into an output vector $\boldsymbol{y}^{(p)}$: $\boldsymbol{x}^{(p)} \rightarrow \boldsymbol{y}^{(p)}$, for $p = 1,\ldots,P$; p is the number of patterns.

Relating to the finite element matrix on the element level, input parameters are the hydraulic conductivity coefficients represented by a matrix $\mathbf{C}$, the derivatives of local natural coordinates λ_i with respect to the global coordinates x_j as contained in the matrix $\boldsymbol{\lambda}_x$ as well as the contributions from the trial functions $\boldsymbol{S}_\lambda$ for which an appropriate description has to be found. The problem simplifies as the matrices $\mathbf{C}$ can be reduced to scalar.

Within the ANN mapping from input to output is performed by setting up a network composed of processing units (neurons) and connections between them. Signals traveling along the connections are weighted, summed up in a propagation function and activated by a function F to become the output (input for next layer) $y = F(G)$, $G = \sum \mathrm{w}_{ij} * \mathrm{x}_j + b$. Here G is the cumulative input attraction, w_{ij} are the weights and b is the threshold parameter.

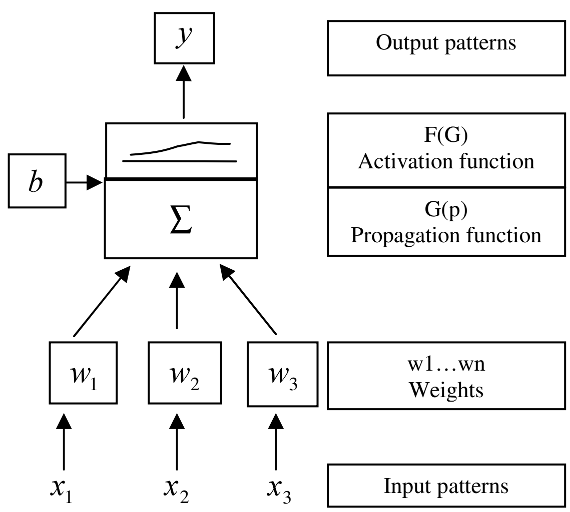


Figure 2: Composition of a single neural net unit

The activation function $F(G)$ can be of different type. In this approach the popular sigmoid activation function $\frac{1}{1+e^{-G}}$ is used.

For the problem given one input, one hidden and one output layer have been used. The neurons in one layer were connected to all neurons in the next layer. The weights w_{ij} and biases b_i parameters are calculated iteratively for training the network.

The training process needs sets of (know) input $x^{(p)}$ and (known) target $t^{(p)}$ patterns. The error between both has been computed by $E = \frac{1}{2} \cdot \sum_{i=1}^{m} (t_i^p - y_i^p)$ for each iterative training step. Unless this error is not beyond a given bound the network weights are updated in backward direction (back propagation) from output layer trough the hidden layer to the input layer according to $w_{ij}(s+1) = w_{ij}(s) + r_{ij}(s)$ where s is the number of the iteration step and r_{ij} is the reinforcement of the network parameters. The reinforcement is computed by the learning rule $r_{ij}(s) = \Delta w_{ij}(s) = -\eta \cdot \frac{\partial E}{\partial w_{ij}}$ which is a gradient approach with the learning rate η. For faster learning results the Levenberg-Marquardt learning algorithm (MATLAB13 reference) was used also. The described network type ends up in good results for the given problem.

TRAINING PARAMETERS FOR THE HYBRID APPROACH

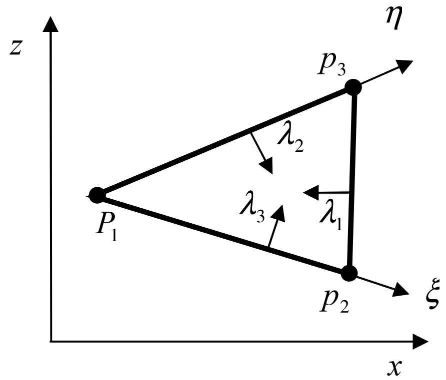
The finite element coefficient matrix has been found to be

$$\boldsymbol{K}_e = \int_{\Omega_e} (\boldsymbol{S}_\lambda \cdot \boldsymbol{\lambda}_x) \cdot \mathbf{C} \cdot (\boldsymbol{\lambda}_x{}^T \cdot \boldsymbol{S}_\lambda{}^T)\, dA$$

The three matrices $\boldsymbol{C}$, $\boldsymbol{\lambda}_x$ and $\boldsymbol{S}_\lambda$ are to be discussed.

The matrix values $\boldsymbol{C}$ represent the hydraulic conductivity, so to say material properties. These are given and constant on the element level. Constant do not represent any difficulty for the ANN learning process. They can be handled as constant factors by multiplication. In this study isotropic soil has been assumed so that $\boldsymbol{C}$ reduces to a scalar factor C and thus can be taken out from the integral.

The second matrix $\boldsymbol{\lambda}_x$ represents the derivatives of the natural coordinates λ_i with respect to global coordinates x_j and thus contains the information about an element size and shape and rotation. The transformation between both coordinate systems is described in the figure 3.



natural coordinates

$\lambda_1 = 1 - \xi - \eta$

$\lambda_2 = \xi$

$\lambda_3 = \eta$

local coordinates

$\xi(x) = m_{11} \cdot x + m_{12} \cdot z + m_{13}$

$\eta(x) = m_{21} \cdot x + m_{22} \cdot z + m_{23}$

Figure 3: Coordinate transformations in a triangle

According to these relations, the matrix $\boldsymbol{\lambda}_x$ for the derivative of the natural coordinates with respect to global coordinates is

$$\boldsymbol{\lambda}_X = \begin{bmatrix} \frac{\partial \lambda_1}{\partial x} & \frac{\partial \lambda_1}{\partial z} \\ \frac{\partial \lambda_2}{\partial x} & \frac{\partial \lambda_2}{\partial z} \\ \frac{\partial \lambda_3}{\partial x} & \frac{\partial \lambda_3}{\partial z} \end{bmatrix} = \begin{bmatrix} -m_{11} - m_{21} & -m_{12} - m_{22} \\ m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

It describes the geometrical properties of an element such as scaling (size), rotation and shearing (shape) by just four independent parameters m_{11} m_{22} m_{12} and m_{21}. These values are elements of the input parameter vector $\mathbf{x}^{(p)}$ of the ANN. The matrix $\boldsymbol{S}_\lambda$ is representing the trial function and thus physics in terms of the state variable. This will be inspected on the different levels of approximation order.

LINEAR TRIAL FUNCTIONS

For linear trial functions the derivative of the trial functions by the natural coordinates is the unit matrix. The element matrix K_e is a symmetric 3x3 matrix.

$$\begin{array}{lll} \frac{1}{2}C \cdot (\lambda_x values) & -\frac{1}{2}C \cdot (\lambda_x values) & -\frac{1}{2}C \cdot (\lambda_x values) \\ \boldsymbol{K}_e = -\frac{1}{2}C \cdot (\lambda_x values) & \frac{1}{2}C \cdot (\lambda_x values) & \frac{1}{2}C \cdot (\lambda_x values) \\ -\frac{1}{2}C \cdot (\lambda_x values) & \frac{1}{2}C \cdot (\lambda_x values) & \frac{1}{2}C \cdot (\lambda_x values) \end{array}$$

For linear trial functions the matrix $\boldsymbol{S}_\lambda$ is the unit matrix and the proportionality factor C is a scalar value. The analytical integration for the triangle for linear trial functions leads to scalar values which are equal for each element of the element matrix. For this approach the values to be trained are the relevant λ_x values within the element matrix. The values contain the geometrical behavior of scaling of the triangle.

QUADRATIC TRIAL FUNCTIONS

For quadratic trial functions the matrix containing the derivatives of the quadratic trial functions with respective to the natural coordinates $\boldsymbol{S}_\lambda$ has a more complex structure.

$$S_\lambda = \begin{bmatrix} 4\cdot\lambda_1 - 1 & 0 & 0 \\ 0 & 4\cdot\lambda_2 - 1 & 0 \\ 0 & 0 & 4\cdot\lambda_3 - 1 \\ 4\cdot\lambda_2 & 4\cdot\lambda_1 & 0 \\ 0 & 4\cdot z_3 & 4\cdot\lambda_2 \\ 4\cdot\lambda_3 & 0 & 4\cdot\lambda_1 \end{bmatrix}$$

Figure 6: $\boldsymbol{S}_\lambda$ matrix for quadratic trial functions

The values of this matrix could be trained by a neural network also. Within this approach the results of this matrix in combination with the integration of the triangle were calculated as scalar values and added to the neural trained matrix with the geometrical behavior. The training of this matrix is equal to the training for linear approximation functions.

CUBIC TRIAL FUNCTIONS

Although the differential equation of second order for groundwater flow can be represented completely by quadratic trial functions the authors tried to simulate the flow filed with cubic trial functions also. The $\mathbf{S}_\lambda$ matrix contains the derivatives of the ten cubic trial functions with respective to the natural coordinates in a 3x10 matrix. In a first approach the values of this matrix associated with the integral formulation of the triangle were calculated by numerical Gaussian integration and added again to the summed values from the affine coordinate transformation within the $\boldsymbol{\lambda}_x$ matrix. The results were put together with the soil parameters in the 10x10 element matrix and traced back in the FE-method calculation.

By reason of the complexity of the cubic $\mathbf{S}_\lambda$ matrix a new approach to train the behavior of the matrix in connection to the analytical integration in the triangle is in test phase. The cubic hybrid problem formulation leads to a high reduction in the computational costs by comparison to the common FE-method.

REALIZATION BY ANN

The generated training data for the training of the matrix comes from a dataset with generalized triangles presenting different inner angles of the triangle.

The training and testing of the hybrid model is exemplarily implemented for groundwater flow in an excavation pit. Isotropic and homogeneous soil is assumed. The figure describes the computational domain and boundary conditions. Physics of the computational domain is described by the Laplace equation for groundwater flow: $\boldsymbol{d}^T \cdot (-\mathbf{C} \cdot \boldsymbol{d}H) = 0$.

The testing example was first run with FE method for linear, quadratic and cubic trial functions. The results of the FE-calculations differ in calculation time from fast for linear trial to slower for cubic trial functions. The results of the FE calculations were presented in figure 4 and 5.

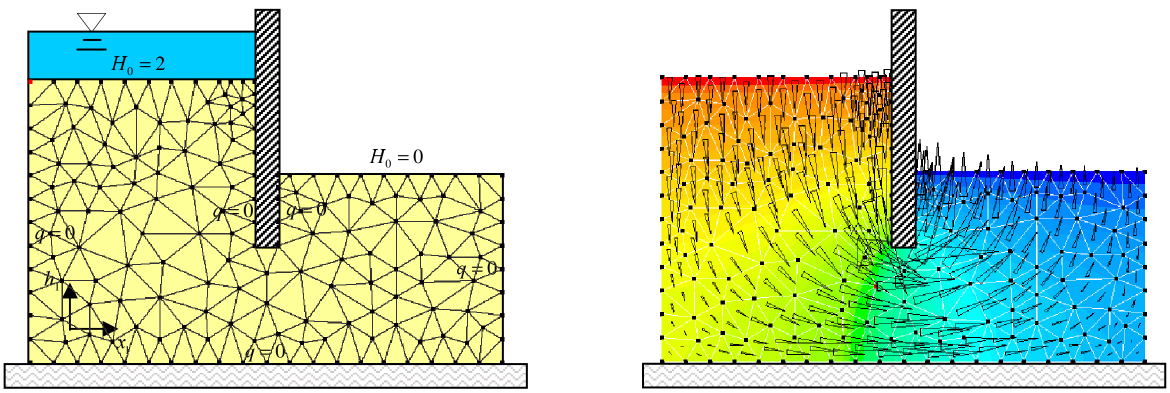


Figure 4: Excavation pit with flow distribution and flow velocity calculated by FE-method

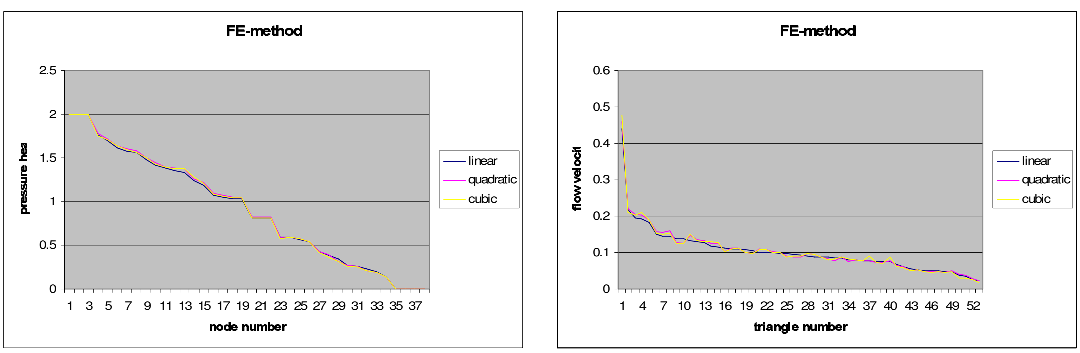


Figure5: FE-calculation with 52 triangles for linear, quadratic and cubic trial functions

LINEAR TRIAL FUNCTIONS

As described above the values of the geometrical behavior within the matrix $\boldsymbol{\lambda}$ has to be trained by an ANN. Within the element matrix the $\boldsymbol{\lambda}$ values are different but contain similar combinations of factors for scaling of the triangle. These parts were trained and combined for all $\boldsymbol{\lambda}$ values within the element matrix. A three-layer feed forward backpropagation network with four input neurons, describing the m-values of the coordinates, three output neurons, describing the similar parts of the geometric behavior and a miscellaneous number of hidden neurons was used. The training process stops after a predefined minimum network error was reached. The trained values were combined with the scalar values for physical behavior of the soil and values for the trial functions in combination with the integration in a triangle to build the element matrix. After the hybrid step the element matrices were traced back into the FE-calculation.

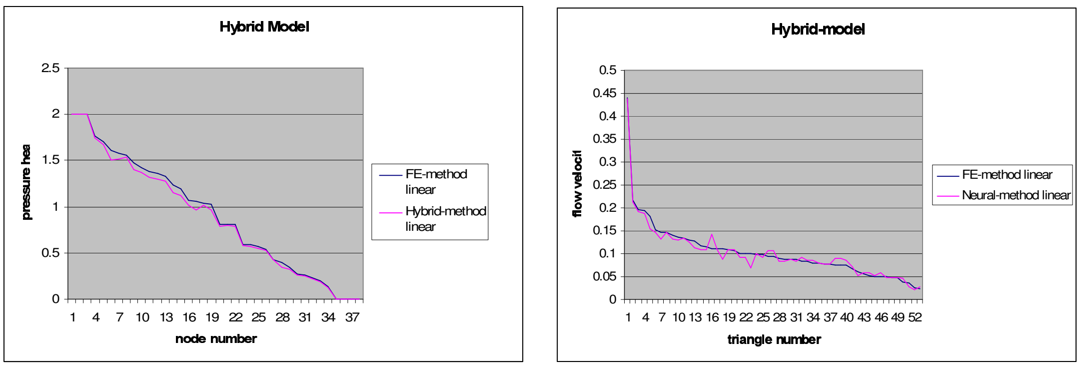


Figure 6: pressure head and flow velocity for FE-method and Hybrid-model with linear trial functions (ANN: 4-30-3, MSE: 2,12875e-006, 500 Epochs)

QUADRATIC TRIAL FUNCTION

For quadratic trial functions the combinations of factors for the geometric behavior of the $\boldsymbol{\lambda}_x$ values were trained... A three layer feed forward backpropagation network with four input neurons, three output neurons and a miscellaneous number of hidden neurons was selected for training also. In this paper the values of the derivations of the trial functions in combination with the integration in the triangle were calculated as scalar values. After training these values were combined with the soil parameters and the neural determined values at the appropriate position within the 6x6 element matrix. A three layer feed forward backpropagation network with four input neurons, three output neurons and a miscellaneous number of hidden neurons was selected for training also. Due to the fact that complex matrix calculations within FE-method were represented by the ANN the quadratic hybrid problem formulation reduces the computational costs of time by comparison to the common FE-method. The results of the hybrid model were presented in figure 7.

OUTLOOK

Investigations for cubic trial functions have been started already and show good results also. The training of the $\mathbf{S}_{\lambda}$ matrix for higher order trial functions by ANN in combination with the analytical integration in a triangle are follow ups. Especially for cubic trial functions the computational costs with the hybrid-model can be highly reduced.

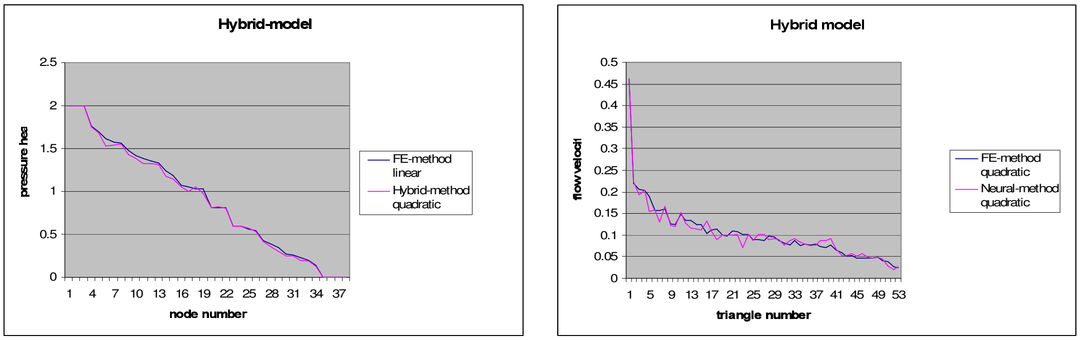


Figure 7: pressure head and flow velocity for FE-method and hybrid-model with quadratic trial functions (ANN: 4-30-3, MSE: 2,12875e-006, 500 Epochs)

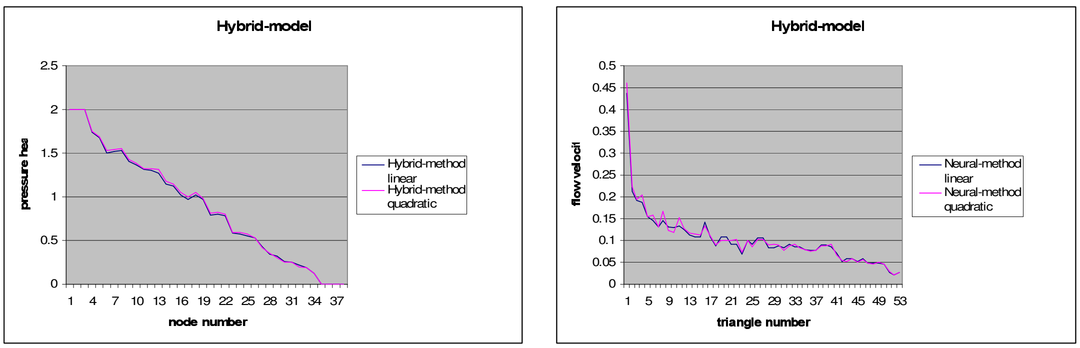


Figure 8: Comparison between hybrid-linear and hybrid quadratic approach

CONCLUSIONS

- The training of element matrices for FE-method partially by ANN was successful.
- The generation of universal neural elements for linear, quadratic and cubic triangular elements was implemented.
- A combined hybrid model using ANN and FE-method generates fast and good results for groundwater flow test cases.
- The hybrid model reduces computational costs for higher order trial functions.
- The neural elements can be universally applied to various potential flow problems (e.g. temperature flow) and computational domains.

REFERENCES

Y. Dibike and M. Abbott (1999) „Application of artificial neural networks to the simulation of a two dimensional flow.“ *Journal of hydraulic research*, Vol. 37, p.435-446.

Anthony W. Minns (1996) „Extended rainfall-runoff modelling using artificial neural networks; *Hydroinformatics 96”*, p. 207-213.

E.R.Funke and N.L. Croshank (1978) „A hybrid Model of the St. Lawrence River Estuary.“ *A.S.C.E. Proceedings 17th Coastal Engineering Conference*, Hamburg, p. 2855-2869, 1978

K.-P.Holz, L.H.C.Chua (2004) “Hybrid River Modeling Using Neural Network and FE-Method.” *Advances in Hydro-Science and -Engineering*, Volume VI, The University of Mississippi, NCCHE, ISBN 0-937099-12-0, pp. 418-419, Brisbane, Australia.

Zenon Waszczyszyn and Leonard Ziemanski (2001) “Neural networks in mechanics of structures and materials – new results and prospects of applications.” *Computers and structures 79*, p 2261-2276.

Klaus Jürgen Bathe (1990) “Finite Element Procedures in Engineering Analysis” Springer Berlin, Heidelberg, New York 1990, ISBN 0-387-15602-X.

Simon Haykin (1999) “Neural Networks: a comprehensive foundation 2nd ed.” Prentice Hall, ISBN 0-13-273350-1

B.Bazartseren, G.Hildebrandt, K.-P.Holz (2003) “Short-term water level prediction using neural networks and neuro-fuzzy approach” *Neurcomputing Journal Special issue: Evolving Solution with Neural Networks*, (Ed. A. Fanni and A. Uncini) Volume/Issue 55/3-4, 2003. pp. 439-450, ISSN 0925-2312

J.C. Mason, R.K. Price, A. Tem´me (1996) „A neural network model of rainfall-runoff using radial basis functions“ J*ournal of hydraulic research,* Vol. 34, 1996, No. 4, p. 537-547

R.M. Benning, T.M. Becker, A. Delgado (2001) “*Initial Studies of predicting flow fields with an ANN hybrid*” Advances in Engineering Software32 (2001), p.895-901

L. H. C. Chua, K.-P. Holz (2005) “Hybrid Neural Network – Finite Element River Flow Model”, *ASCE Journal of Hydraulic Engineering*, 313, No. 1, pp. 52-59, ISSN 0733-9429/2005/1-52-59

Weber, L., Holz K.P. (2005) Hybrid Modelling – Mixing Finite and Neural Elements, *International Symposium on Stochastic Hydraulics* 2005, 23.-24.05 2005, Nijmegen, The Netherlands, ISBN: 90-805649-9-0

Patterson Dan (1996) “Artificial Neural Networks, Theorie and Applications” Prentice Hall, New York, ISBN 3-8272-9531-9

The MathWorks Company (2006) “The MATLAB reference and documentation” *http://www.mathworks.com*