CIVIL AND ENVIRONMENTAL ENGINEERING REPORTS

No. 7 2011

ON SOLVING SELECTED PROBLEMS OF LINEAR ALGEBRA BY MEANS OF NEURAL NETWORKS

Maria MRÓWCZYŃSKA*
University of Zielona Góra, Faculty of Civil and Environmental Engineering,
Institute of Building Engineering
ul. prof. Z. Szafrana 1, 65-516 Zielona Góra, Poland

The paper presents selected practical applications and results of computer simulations from the field of numerical linear algebra realized by means of neural networks. Bearing in mind aspects of applications, it has been decided that priority should be given to the description of the problem of soling over-determined linear systems in the norm l_2 and the norm l_1.

Keywords: neural networks, linear algebra, linear systems

1. INTRODUCTION

The problem of solving linear systems is one of basic tasks present in a wide class of fields of science. A preferred method of estimating parameters of linear models (Gauss-Markov models) is the least squares method, which enables the reduction of a random influence of measurement errors being in the Gauss distribution to a value determined by the norm l_2. The norm l_2 leads closer towards the solution in the other norms [7], but when the observation vector is in disagreement with the Gauss distribution an optimum criterion for optimisation can be the norm l_1, and in special cases the norm l_∞. The interdisciplinary character of the subject of artificial neural networks provides a favourable strategy for the optimisation of models describing phenomena and processes existing in nature. Neural networks circumferential in structure have been applied for solving over-determined systems of linear equations on the basis of the minimisation an objective function (energy function) in a particular norm. Algorithms carrying out standard matrix operations (inversion and pseudo

* E-mail: m.mrowczynska@ib.uz.zgora.pl

inversion of matrices, specifying values and eigenvectors, SVD decomposition) work as a result of the application of neural networks in which signals flow in one direction (one direction networks). When permanent integrators are properly chosen the process of progression towards the solution is faster.

2. MATERIALS AND METHODS

The most common basic tasks and application tasks realised in the field of geodesy concern estimating components of the vector of parameters of over-determined systems of linear equations

$$\mathbf{Ax} \cong \mathbf{l}\,, \tag{1}$$

where: $\mathbf{A} = [a_{ij}] \in R^{m \times n}$ ($m>n$) - the matrix of a model with real entries, $\mathbf{l} \in R^m$ - the observation vector, $\mathbf{x} = [x_1, x_2, ..., x_n]^T \in R^n$ - the estimated vector of parameters. The minimisation of the criterion

$$\| \mathbf{Ax} - \mathbf{l} \| \rightarrow \text{minimum} \tag{2}$$

requires the formulation of a form of an energy function (Lapunov function) whose bottom energy state corresponds to the solution expected $\mathbf{x}^*$. In general we will define the energy function (objective function) as

$$E(\mathbf{x}) = \sum_{i=1}^{m} \omega[v_i(\mathbf{x})]\,, \tag{3}$$

where $\omega[v_i(\mathbf{x})]$ represents a convex function in relation to the vector of parameters $\mathbf{x}$ in the whole space R^n, which will next be called the *weight function,* and its derivative in relation to the correction $v_i(\mathbf{x})$ - the *activation function* [2]. For this reason for $\omega_i(v_i) = p_i v_i^2 / 2$ ($p_i > 0$) the standard activation model of a square energy function (the weighed criterion of the least squares) has the form

$$E(\mathbf{x}, \mathbf{P}) = \frac{1}{2} \sum_{i=1}^{m} p_i v_i^2(\mathbf{x}) = \frac{1}{2} (\mathbf{Ax} - \mathbf{l})^T \mathbf{P} (\mathbf{Ax} - \mathbf{l}) \tag{4}$$

(the ratio ½ simplifies the transformations) with the weight matrix $\mathbf{P} = \text{diag}(p_1, p_2, ..., p_m)$. A convex function used in linear algebra is the logistic function represented by

$$\omega_i(v_i) = \frac{\alpha}{\beta}\ln[\cosh(\alpha v_i(\mathbf{x})], \tag{5}$$

whose parameters $\alpha > 0$ and $\beta > 0$ impose an optimisation strategy. The energy function corresponding to this function is defined as follows:

$$E(\mathbf{x}, \alpha, \beta) = \frac{\alpha}{\beta}\ln\{\cosh[\beta v_i(\mathbf{x})]\}. \tag{6}$$

Another option in the class of convex functions is the function with the form

$$\omega_i[v_i(\mathbf{x})] = |v_i(\mathbf{x})|, \tag{7}$$

in this case the energy function

$$E(\mathbf{x}) = \sum_{i=1}^{m} |v_i(\mathbf{x})|, \tag{8}$$

undergoes minimisation, whose form is expressed by the formula of the rule of the minimum of absolute deviation.

3. NEURAL NETWORKS SOLVING SYSTEMS OF LINEAR EQUATIONS

Solving systems of linear equations is one of basic tasks of optimising neural networks with a circular structure presented in fig. 1.

It results from the dependence that the solution of the system of linear equations (1) is equivalent to the minimisation of a square function without limits. Gradient methods are included into effective optimisation methods, but for a large value of the index cond(**A**) convergence of these methods is slow. The process of estimation of the value of parameters of the function (4) can be described by means of the system of differential equations

$$\frac{d\boldsymbol{x}}{dt} = -\mu\nabla E_1(\mathbf{x}) \tag{9}$$

where $\mu_j > 0$ - the learning ratio, and the gradient of the energy function on the assumption that **P**=**I**, is described by the dependence $\nabla E_1[\mathbf{x}] = \mathbf{A}^T(\mathbf{Ax} - \mathbf{l})$. Then, the system of equations (1) written in a scalar form looks as follows

$$\frac{dx_j}{dt} = -\sum_{p=1}^{n} \mu[\sum_{i=1}^{m} a_{ip} (\sum_{k=1}^{n} a_{ik} x_k - l_i)] . \tag{10}$$

Bearing in mind that the function $E_1(\mathbf{x})$ is a Lapunov function (a random real function whose changes during the algorithm are not positive) and the Hessian $\mathbf{B} = \mathbf{A}^T \mathbf{A}$, the solution of the system of equations is asymptotically stable $(t \to \infty)$ [7]. A scheme of the architecture of a neural network intended for solving systems of linear equations **Ax=l** is presented in fig.1.

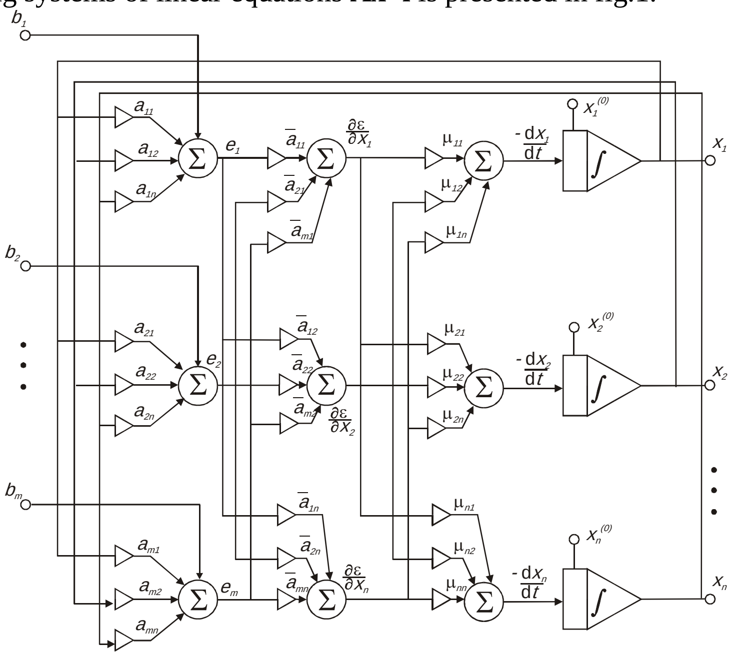

Fig. 1. Structure of a neural network intended for solving systems of linear equations

Disagreement between the distribution of observation errors and the normal distribution excludes the use of the classic method of the least squares. Then, the criterion of the mean square error is not resistant to disturbances and data deviate from the model intended (outliers). A solution to this problem is to replace the square function (4) with the logistic function (5) whose corresponding function is the energy function

$$E(\mathbf{x}, \alpha, \beta) = \frac{\alpha}{\beta} \sum_{i=1}^{m} \ln\{\cosh[\beta v_i(\mathbf{x})]\} . \tag{11}$$

The minimisation of the above criterion function consists in solving the system of differential equations

$$\frac{dx_j}{dt} = -\sum_{p=1}^{n} \mu_{jp} \{ \sum_{i=1}^{m} [a_{ip} g_i (\sum_{k=1}^{n} a_{ik} x_k - l_i)] \} , \tag{12}$$

where

$$g_i(v_i) = \frac{\partial \omega(v_i)}{\partial v_i} = \frac{\partial \{ \frac{\alpha}{\beta} \ln[\cosh(\beta v_i(\mathbf{x}))] \}}{\partial v_i(\mathbf{x})} = \alpha \tanh[\beta v_\mathrm{i}(\mathbf{x})]. \tag{13}$$

For large values of the ratio α and small values of the ratio β the results of minimisation correspond to the results obtained by means of the procedure (10), and a change in the value of these ratios leads closer to the results of equalisation according to the rule of the least modules, because $\tanh[\beta v_i(\mathbf{x})] \approx 1 - 2\mathrm{e}^{-\beta v_i(\mathbf{x})}$, and the value of the activation function $g_i(v_i)$ approaches the value of the signum function [4].

For the distribution of observation errors undergoing the Cauchy distribution, which has higher values for arguments more distant from the average in comparison to the values of the Gauss distribution, an optimum minimisation criterion is the norm l_1. By modifying the objective function (3) to the form (7) as a convex weight function, we obtain the irregular objective function (8) (energy function), whose minimisation requires special procedures of mathematical programming [1] or the application of an algorithm with the use of neural networks, which is simple to achieve. The problem of the minimisation of the energy function (8) in the norm l_1 consists in solving the system of differential equations

$$\frac{d\boldsymbol{x}}{dt} = \mu \sum_{i=1}^{m} a_{ij} \mathrm{sgn}[v_i(\mathbf{x})], \tag{14}$$

and the modified activation function (modified signum function)

$$\mathrm{sgn}[v_i(\mathbf{x})] = \begin{cases} 1 & \text{gdy} \quad v_i(\mathbf{x}) > 0 \\ -1 & \text{gdy} \quad v_i(\mathbf{x}) < 0 \end{cases} \tag{15}$$

determines the sign of the left-sided or the right sided derivative in the neighbourhood of the point **x** (function (8) is continuous, but it is not differentiable in relation to **x**). Values of parameters obtained by means of equalisation in the norm l_1 correspond to the values of observation medians on the assumption that the matrix **A** is a full rank matrix.

At this point, it is necessary to add that apart from the objective function (8), whose form is the formulation of the rule of absolute deviations as a “natural” robust estimation, a number of weight functions were arbitrarily formulated in order to identify outstanding observations and to eliminate their unfavourable influence on estimation results. In order to define a weight function it is necessary to consider the condition of continuity and to limit the activation function of the resistant estimator, its characteristic feature is the breakdown point α^* as a specified limit of random errors. The average value is not a resistant estimator, because for $\alpha^* = 1/\mathrm{m}$ even a single observation changes the value of the estimator. For the median and $\alpha^* = 0{,}5$, the estimator breaks down when the lumber of outstanding observations is at least half of all the observations carried out [9]. The most popular weight function is the Huber function [5], because the estimator which results from the application of this function with a specific limit of random errors is an estimator with the smallest variance in the class of functions satisfying this limitation.

4. RESULTS AND DISCUSSION

A numerical solution of the equalisation of a levelling network (fig. 2) with minimum limitations of degrees of freedom according to the rules described by the models (4), (6) and (8), is presented on the example below.

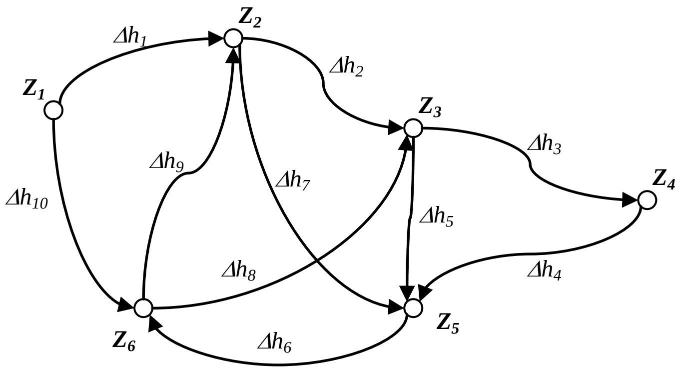

Fig 2. Structure of a levelling network undergoing equalisation

Table 1. Data (simulation)

No.	Observation codes	Free expressions $\Delta h[mm]$
1	1-2	+0,7
2	2-3	+1,6
3	3-4	-1,5
4	4-5	+1,2

5	3-5	-0,9
6	5-6	-0,5
7	2-5	+1,6
8	6-3	+0,6
9	6-2	-1,8
10	1-6	+1,4

Table 2. Value of parameters $\Delta h\ [mm]$

A. Model (4)	B. Model (6) α=1; β=20	C. Model (6) α=20; β=1	D. Model (8)
0	0	0	0
-0,72	-0,86	-1,04	-0,81
+0,94	+0,86	+0,65	0,70
-0,63	-0,71	-0,80	-0,70
+0,50	+0,43	+0,49	+0,50
m_0=0,97 mm	$\hat{m}$ =0,97 mm	$\hat{m}$ =1,01 mm	$\hat{m}$ =0,98 mm

Hence, we see that the tasks of solving over-determined systems of linear equations carried out numerically are approximately in agreement with the assumptions adopted, and the amount of calculations necessary to determine arithmetical operations is small in this case.

One direction neural networks can be used to carry out standard matrix operations, which include the determination of the converse of a positively definite square matrix **B** rank r ($\mathbf{B} = \mathbf{A}^T \mathbf{A}$). In order to carry out this operation it is necessary to design an adequate structure of a neural network which will minimize an energy function. It results from the formula of the inversion **C** of the matrix **B** that **CB=I**. By multiplying this equation by the non-zero vector $\mathbf{x} = [x_1, x_2, ..., x_n]$ (the vector undergoes normalisation) we obtain **CBx=x**. On this basis the definition of the criterion function (energy function) assumes the form [8]

$$E = \| \mathbf{CBx} - \mathbf{x} \|^2 . \tag{16}$$

At this point let us pay attention to the fact that the vector **x** represents a teaching vector and at the same time an assigned vector. The operation of networks belonging to the type of auto-associating networks merely consists in adapting the weights of V_{ij} ($i = 1,2,..., n$; $j = 1,2,..., n$) of the matrix $\mathbf{C} = \mathbf{B}^{-1}$ on the basis of the algorithm of error back propagation, according to the formula

$$\frac{dC_{ij}}{dt} = -\mu V_j (y_i - x_i) , \tag{17}$$

where y_i - actual value of the neural network output signal , x_i - known value of the output signal.

Example 1. The inversion $\mathbf{B}^{-1}$ of the positively definite symmetrical matrix **B** $(\det(\mathbf{B}) \neq 0)$

$$\mathbf{B}=\begin{bmatrix} 3 & -2 & 3 & -1 & 0 \\ -2 & 6 & 4 & -2 & 8 \\ 3 & 4 & 4 & 0 & -2 \\ -1 & -2 & 0 & 5 & 3 \\ 0 & 8 & -2 & 3 & 8 \end{bmatrix} \quad \mathbf{C}=\mathbf{B}^{-1}=\begin{bmatrix} 0{,}2502 & -0{,}1083 & -0{,}0133 & -0{,}0723 & 0{,}1322 \\ -0{,}1083 & 0{,}0032 & 0{,}0963 & -0{,}0424 & 0{,}0367 \\ -0{,}0133 & 0{,}0963 & 0{,}1105 & 0{,}0995 & -0{,}1060 \\ -0{,}0723 & -0{,}0424 & 0{,}0995 & 0{,}1654 & 0{,}0052 \\ 0{,}1322 & 0{,}0367 & -0{,}1060 & 0{,}0052 & 0{,}0598 \end{bmatrix}$$

Example 2. The inversion $\mathbf{B}^{-1}$ of the positively definite asymmetrical matrix **B** $(\det(\mathbf{B}) \neq 0)$

$$\mathbf{B}=\begin{bmatrix} 3 & 7 & -5 & 0 & 1 \\ -2 & 3 & 8 & -2 & 0 \\ 4 & 6 & 5 & 0 & -2 \\ -8 & 4 & 6 & -2 & 3 \\ -2 & -1 & 0 & 4 & -2 \end{bmatrix} \quad \mathbf{C}=\mathbf{B}^{-1}=\begin{bmatrix} -0{,}5191 & -1{,}2022 & 0{,}8247 & 0{,}4831 & -0{,}3596 \\ 0{,}2472 & 0{,}3820 & -0{,}2022 & -0{,}1348 & 0{,}1236 \\ -0{,}4494 & -0{,}8764 & 0{,}6404 & 0{,}4270 & -0{,}2247 \\ -0{,}9078 & -2{,}2303 & 1{,}4337 & 1{,}0225 & -0{,}3539 \\ -1{,}4202 & -3{,}4494 & 2{,}1437 & 1{,}6292 & -0{,}9101 \end{bmatrix}$$

By analogy to the calculation of the inversion of the matrix by means of the Gauss method, we have a completely feasible method of realising this task by means of neural networks, which consists in solving a system of differential equations

$$\frac{dx_j}{dt} = -\mu v_j(\mathbf{x}), \tag{18}$$

where correction $v_j(\mathbf{x}) = \sum_{i=1}^{n} a_{ji} x_i - l_j$ for j=1,2,…,n.. For each consecutively calculated column of the inversion matrix it is necessary to successively adopt: $\mathbf{l}_1 = [1,0,0,0,0]^{\mathrm{T}}, \mathbf{l}_2 = [0,1,0,0,0]^T, \mathbf{l}_3 = [0,0,1,0,0]^T, \mathbf{l}_4 = [0,0,0,1,0]^T, \mathbf{l}_5 = [0,0,0,0,1]^T$.

The result obtained represents the matrix $\mathbf{L} = \mathbf{B}^{-1}$.

It is commonly known that if the matrix **B** is a square non-singular matrix then the minimisation of the criterion function (4) (on the assumption that $\mathbf{P} = \mathbf{I}$) leads to the estimator of the least squares $\mathbf{x}^* = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{l}$. There also exists a converse of the non-singular matrix, which can be determined on the basis of known eigenvalues $\{\lambda_i\}$ and eigenvectors $\{w_i\}$ on the basis of the equation [3]

$$\mathbf{B}^{-1} = \sum_{i=1}^{r} \frac{1}{\lambda_i} w_i w_t^T \ , \tag{19}$$

where r denotes the rank of the matrix **B**.

We will begin searching for eigenvalues and eigenvectors of the symmetrical matrix by means of neural networks with presenting the symmetric and non-singular matrix

$$\mathbf{B} = \begin{bmatrix} 4 & -1 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 2 & 4 & -1 \\ -1 & -1 & 4 & 3 & 0 \\ 0 & 0 & -1 & 0 & 2 \end{bmatrix}$$

in the form $\mathbf{B} = \mathbf{w}\Lambda\mathbf{w}^T$ $(\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, ..., \lambda_n])$, which is bilaterally multiplied by a random non-zero vector **x**, and we obtain

$$\mathbf{w}\Lambda\mathbf{w}^T\mathbf{x} - \mathbf{B}\mathbf{x} = \mathbf{0} . \tag{20}$$

Then, considering the dependence $\mathbf{w}^T\mathbf{w} = \mathbf{1}$, we will write

$$\mathbf{w}^T\mathbf{w}\mathbf{x} - \mathbf{x} = \mathbf{0} . \tag{21}$$

On the basis of the two final equations the energy function defined will assume the form (explanations concerning the relationship between teaching networks can be found in paper [3])

$$E = \frac{1}{2}\{\| \mathbf{w}^T \Lambda\mathbf{w}\mathbf{x} - \mathbf{B}\mathbf{x} \|^2 + \| \mathbf{w}^T\mathbf{w}\mathbf{x} - \mathbf{x} \|^2 . \tag{22}$$

We obtain the minimum of this function by solving the system of differential equations

$$\frac{d\lambda_i}{dt} = -\mu u_i \hat{z}_i \tag{23}$$

$$\frac{d\mathbf{w}_i}{dt} = -\mu[\mathbf{x}\hat{u}_i + (\mathbf{y}^{(1)} - \mathbf{d}^{(1)}) z_i + (\mathbf{y}^{(2)} - \mathbf{d}^{(2)}) u_i] \tag{24}$$

The results of the numerical realisation of the calculation of eigenvalues and eigenvectors (the following values of signals have been adopted: $x_i(t) = i\sin\omega t$ where $i = (1, \ldots, 5)$ for $\omega = 1e^7$ and $\mu = 0{,}01$) and the verification of the solution to the task are presented below:

a) eigenvalues

$$\Lambda = \begin{bmatrix} 4{,}2058 & 0 & 0 & 0 & 0 \\ 0 & 6{,}5373 & 0 & 0 & 0 \\ 0 & 0 & 2{,}1177 & 0 & 0 \\ 0 & 0 & 0 & 1{,}3413 & 0 \\ 0 & 0 & 0 & 0 & -1{,}9176 \end{bmatrix}$$

b) matrix of eigenvectors

$$\mathbf{w} = \begin{bmatrix} 0{,}8338 & -0{,}3548 & -0{,}0759 & 0{,}4162 & 0{,}0011 \\ -0{,}4790 & -0{,}0824 & -0{,}2483 & 0{,}8207 & -0{,}1687 \\ 0{,}1048 & 0{,}6527 & -0{,}1202 & 0{,}2372 & 0{,}7016 \\ 0{,}1973 & 0{,}7261 & 0{,}2081 & 0{,}0938 & -0{,}6179 \\ -0{,}0728 & -0{,}1282 & 0{,}9434 & 0{,}2271 & 0{,}1913 \end{bmatrix}$$

c) verification of the solution $\mathbf{B} = \mathbf{w}\Lambda\mathbf{w}^T$

$$\mathbf{B} \cong \mathbf{B}^{/} = \begin{bmatrix} 3{,}9911 & -0{,}9904 & -0{,}9961 & -0{,}9720 & 0{,}0167 \\ -0{,}9904 & 1{,}9890 & -0{,}0114 & -0{,}9948 & 0{,}0315 \\ -0{,}9961 & -0{,}0114 & 1{,}9930 & 3{,}9932 & -1{,}0043 \\ -0{,}9720 & -0{,}9948 & 3{,}9932 & 2{,}9814 & 0{,}0023 \\ 0{,}0167 & 0{,}0315 & -1{,}0043 & 0{,}0023 & 2{,}0136 \end{bmatrix}$$

The linear task of the least squares $\mathbf{Ax} \cong \mathbf{l}$ can be solved by means of the distribution of the matrix **A** in relation to particular values (the *SVD* distribution). Then [6]

$$\mathbf{A}^{+} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^{T} \in R^{n\times m} \tag{25}$$

where $\mathbf{V} \in R^{n\times n}$ and $\mathbf{U} \in R^{m\times m}$ are orthogonal, and $\mathbf{S}^{-1}$ is the matrix of the inverses of singular values $\mathbf{S}^{-1} = \mathrm{diag}(1/\boldsymbol{\sigma}_1,\ldots,1/\boldsymbol{\sigma}_r,0,\ldots,0) \in R^{n\times m}$ $\boldsymbol{\sigma}_1 \geq \ldots \geq \boldsymbol{\sigma}_r > 0$. In order to verify whether the pseudoinverse $\mathbf{A}^{+}$ of the matrix **A** has been determined correctly, it is necessary to check whether the dependence $\mathbf{A} = \mathbf{A}\,\mathbf{A}^{+}\mathbf{A}$ is satisfied. The pseudoinverses of the matrix **A** in the form

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 1 & 4 \\ 3 & 4 & 1 & -3 \\ 3 & 5 & 1 & 21 \\ 3 & 6 & 1 & 4 \\ 3 & 7 & 1 & 0 \end{bmatrix},$$

are determined on the basis of singular values

$$\mathbf{S}^{-1} = \begin{bmatrix} 0{,}0430 & 0 & 0 & 0 & 0 \\ 0 & 0{,}0940 & 0 & 0 & 0 \\ 0 & 0 & 0{,}4075 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

of the components of the matrix

$$\mathbf{V} = \begin{bmatrix} -0{,}1914 & -0{,}4325 & 0{,}8224 & -0{,}3162 \\ -0{,}3080 & -0{,}8106 & -0{,}4980 & 0 \\ -0{,}0638 & -0{,}1442 & 0{,}2741 & 0{,}9487 \\ -0{,}9298 & 0{,}3675 & -0{,}0231 & 0 \end{bmatrix}$$

and the components of the matrix

$$\mathbf{U} = \begin{bmatrix} -0{,}2007 & -0{,}0736 & 0{,}8765 & -0{,}4285 & -0{,}0489 \\ 0{,}0396 & -0{,}5442 & 0{,}3336 & 0{,}7296 & 0{,}2416 \\ -0{,}9340 & 0{,}2089 & -0{,}0957 & 0{,}1828 & 0{,}2035 \\ -0{,}2670 & -0{,}4548 & -0{,}1383 & 0{,}0160 & -0{,}8381 \\ -0{,}1202 & -0{,}6693 & -0{,}3036 & -0{,}5001 & 0{,}4420 \end{bmatrix}.$$

The pseudoinverse of the matrix **A**, calculated according to the relationship (25), is:

$$\mathbf{A}^{+} = \begin{bmatrix} 0{,}2984 & 0{,}1336 & -0{,}0329 & -0{,}0257 & -0{,}0735 \\ -0{,}1696 & -0{,}0267 & 0{,}0159 & 0{,}0663 & 0{,}1142 \\ 0{,}0995 & 0{,}0445 & -0{,}0110 & -0{,}0086 & -0{,}0245 \\ -0{,}0028 & -0{,}0235 & 0{,}0455 & -0{,}0037 & -0{,}0155 \end{bmatrix},$$

and the verification of the solution $\mathbf{A} = \mathbf{A}\mathbf{A}^{+}\mathbf{A}$ equals

$$\mathbf{A} \cong \mathbf{A}' = \begin{bmatrix} 2{,}9991 & 0{,}9985 & 0{,}9997 & 3{,}9972 \\ 3{,}0000 & 3{,}9994 & 1{,}0000 & -3{,}0004 \\ 3{,}0030 & 4{,}9997 & 1{,}0001 & 21{,}0004 \\ 3{,}0060 & 6{,}0000 & 1{,}0002 & 4{,}0012 \\ 3{,}0090 & 7{,}0003 & 1{,}0003 & 0{,}0020 \end{bmatrix}.$$

It is also necessary to add that the pseudoinverse $\mathbf{A}^{+}$ of the matrix $\mathbf{A} \in R^{m\times n}$ can be determined by means of the factorization **QR** of the matrix **A**, where $\mathbf{Q} \in R^{m\times m}$ is a matrix with orthonormal columns, $\mathbf{R} \in R^{m\times n}$ is a triangular or trapezoidal matrix. Then, the pseudoinverse $\mathbf{A}^{+}$ is calculated from the dependence [6]:

$$\mathbf{A}^{+} = \mathbf{R}^{+}\mathbf{Q}^{+} = \mathbf{R}^{-1}\mathbf{Q}^{T} . \tag{26}$$

The factors of the distribution **QR** of the matrix **A** are matrixes with the forms:

$$\mathbf{Q} = \begin{bmatrix} -0{,}4472 & 0{,}7819 & 0{,}4249 & -0{,}0860 & -0{,}0271 \\ -0{,}4472 & 0{,}1303 & -0{,}5662 & 0{,}6486 & 0{,}2041 \\ -0{,}4472 & -0{,}0869 & -0{,}4490 & 0{,}7438 & 0{,}1939 \\ -0{,}4472 & -0{,}3041 & 0{,}0471 & 0{,}0575 & -0{,}8379 \\ -0{,}4472 & -0{,}5212 & 0{,}5432 & 0{,}1236 & 0{,}4669 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} -6{,}7082 & -10{,}2859 & -2{,}2361 & -11{,}6260 \\ 0 & -4{,}6043 & 0 & -0{,}3041 \\ 0 & 0 & 0 & -5{,}8416 \\ 0 & 0 & 0 & -17{,}6800 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and the pseudoinverse

$$\mathbf{A}^{+} = \begin{bmatrix} 0{,}2984 & 0{,}1336 & -0{,}0329 & -0{,}0257 & -0{,}0735 \\ -0{,}1696 & -0{,}0267 & 0{,}0159 & 0{,}0663 & 0{,}1142 \\ 0{,}0995 & 0{,}0445 & -0{,}0110 & -0{,}0086 & -0{,}0245 \\ -0{,}0028 & -0{,}0235 & 0{,}0455 & -0{,}037 & -0{,}0155 \end{bmatrix}.$$

5. CONCLUSIONS

The problems of solving selected tasks of matrix algebra by means of the technique of neural networks presented in the paper are becoming more and more important because they can be used in a number of fields of technology. With little complexity and refinement of the mathematical apparatus and the use of software implementation it is possible to obtain results almost in real time. The choice of an adequate form of the convex function discussed in the paper in the aspect of the specificity of a particular problem (e.g. resistance to disturbances), makes it possible to choose the most favourable approach to the solution of an over-determined system of linear equations without the necessity to determine the converse of a Hessian matrix. The problem of determining the inversion of eigenvalues and eigenvectors of a square matrix, included into standard matrix operations, can be successfully solved by means of neural networks on condition that an energy function is properly defined for a particular operation, and constant integrators are properly chosen. The parallel operation of the stable algorithms presented numerically shortens the time used to solve the tasks.

BIBLIOGRAPHY

1. Andrews D. F.: *A robust metod multiple linear regression*, Technometrics 16, 1974.
2. Cichocki A., Unbehauen R.: *Neural Networks for Solving Systems if Linear Equations and Related Problems*, IEEE Transactions on Circuits and Systems-I: Fundamenral Theory and Applications, Vol. 39, No 2, 1992.
3. Fukunaga K.: *Introduction to statistical pattern recognition*. Wyd. 2, Akademic Press, San Diego, 1990.
4. Gil J.: *Przykłady zastosowań sieci neuronowych w geodezji*, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra 2006.
5. Huber P. J.: *Robust statistics*, New York: Wiley, 1981.
6. Kiełabasiński A., Schwelick H., *Numeryczna algebra liniowa*, Wydawnictwo Naukowo – Techniczne, Warszawa 1993.
7. Korn G. A., Korn T. M.: *Matematyka dla pracowników naukowych i inżynierów*, cz.1, PWN, Warszawa 1983.
8. Osowski S.: *Sieci neuronowe*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1996.
9. Ostasiewicz W.: *Statystyczne metody analizy danych*, Wydawnictwo Akademii Ekonomicznej im. O. Langego we Wrocławiu, Wrocław 1999.

ZAGADNIENIE ROZWIĄZYWANIA WYBRANYCH ZADAŃ ALGEBRY LINIOWEJ ZA POMOCĄ SIECI NEURONOWYCH

S t r e s z c z e n i e

W pracy przedstawiono wybrane zastosowania praktyczne i wyniki symulacji komputerowych z zakresu numerycznej algebry liniowej, realizowanej za pomocą sieci neuronowych. Mając na względzie aspekty zastosowań, uznano za celowe nadać priorytet opisowi zagadnienia wyrównania nadokreślonych układów liniowych w normie l_2 oraz w normie l_1. Do standardowych operacji numerycznych zaliczono również algorytmy obliczania inwersji macierzy kwadratowych oraz wyznaczania ich wartości własnych i wektorów własnych.