VOL. 11, NO. 20, OCTOBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

CELL BY CELL ARTIFICIAL NEURAL NETWORK MODEL FOR PREDICTING LAMINAR, INCOMPRESSIBLE, VISCOUS FLOW

Osama Sabir and T. M. Y. S. Tuan Ya
Department of Mechanical Engineering, Universiti Teknologi Petronas, Malaysia
E-Mail: ofasabir@gmail.com

ABSTRACT

In this research, a cell-by-cell artificial neural network approach is used to predict the velocity vectors of steady-state, viscous, incompressible, laminar flows in a two-dimensional computational domain. The flow behaviour is characterized by the initial flow velocity, and the geometry of the wall boundaries. A feedforward neural network architecture is applied in this research. The model is trained using Levenberg-Marquardt and Bayesian regularization backpropagation algorithms. The training data for the model are obtained by solving the Navier-Stokes equations for two-dimensional, steady-state, viscous, incompressible, laminar flow using commercial ANSYS Fluent software. The results show that the predicted values produced by the model is in good agreement with the simulation data. Even though the introduction of artificial neural networks at the cell level increases the complexity of the training process, this drawback is compensated by the increase in flexibility (generality) of the model. More importantly, the results show that the cell-by-cell artificial neural network approach is capable of providing an accurate prediction of the fluid velocity field for the flow investigated in this research. The outcomes designate that the new ANN approach is capable of getting an accurate velocity vector prediction as several statistical parameters confirmed. Since all the computation cost took place in the training phase, the new approach calculated the result faster than the traditional numerical methods. Such simulation provides a reliable perception about the fluid behaviour with respect to momentum and equations. In addition to the preceding recorded data, the proposed method considers the geometrical boundaries profile as a major contribution for ANN training phase.

Keywords: artificial neural networks, computational fluid dynamics, cell-by-cell, flow velocity, geometrical boundaries profile.

INTRODUCTION

The ANN is generally used for classification, prediction, patterns recognitions, data filtering and optimization. Artificial neural networks provide a significant research tool for engineering problems that are complex, nonlinear, and with state of uncertainty. Both experimental and numerical techniques have been widely used by scientists in order to gain insight into the fluid flows behaviour. Even though experimental techniques are usually reliable, these techniques are very time-consuming and costly since there is a need to acquire apparatus, instruments and/or equipment, fabricate model prototypes and build a dedicated experimental rig. With advancements in computing hardware and software over the years, numerical techniques using computational fluid dynamics (CFD) have made it possible for scientists to probe into the flow behaviour and underlying physics of fluid flows within a shorter time frame compared to experimental techniques. However, even though CFD techniques are capable of attaining faster results compared to experimental techniques, numerical techniques are still somewhat time-consuming.

The artificial neural network (ANN) can easily overcome the complexity and get faster results due to its flexibility and automatic perception. Unlike other numerical techniques, ANN is capable of dealing with problems where there is lack of a proper physical model and problems in which uncertainties are present. More importantly, ANN is a promising method that can be used to predict the behaviour of fluid flows. The basic idea of ANN is to attach a group of arrays representing the inputs to an equivalent output array. When new inputs are entered, the ANN predicts the outputs instantaneously by applying what it has been trained. Pulido-Calvo and Portela, applied computational neural networks (CNN) to predict the daily stream flows in Portuguese rivers. The feed forward CNN is used to forecast one-day ahead daily flows considering that only flows in previous days are available for the calibration of the models [1]. The artificial or computational neural networks (ANN or CNN) is a non-linear mathematical structure capable of demonstrating the complex non-linear processes that relate the inputs to the outputs of a system. The transient flow can be considered as a time series-forecasting network that will allow to estimate the variables in the passage of time [2].

Benning *et al*. [3, 4] reported initial results on the possibility of using back propagation neural networks to predict the flow field for the simple case of isothermal-steady flow around a solid cylinder. The hybrid artificial intelligence established to predict flow fields for different Reynolds's number ranging from 1 to 60. The algorithm is improved by combining ANN and conventional numerical methods [3, 5]. Valyuhov *et al*., promoted the method of weighted residuals (MWR) to solve Navier-Stokes equations. MWR algorithm is based on the use of the neuron functions approximation to solve Navier- Stokes equations on a uniform computational grid. In a Cartesian computational grid, the neural net function is used to calculate velocity components and the pressure [6-8].

Ben-Nakhi *et. al.,* [9] use ANN as a tool for inter-model comparison with CFD results for cases in which

CFD produces different solutions for different discretization schemes; such cases are referred to as ill posed problems. These ANN models are accurately predicting free convection in partitioned enclosures which can be used for lending support to the authenticity of CFD solutions. Ranković and Savić, [10], investigate the ability of FNN model to predict the nondimensional velocity of the gas that flows along a porous wall. The two-layer FNN with Levenberg–Marquardt learning was constructed. A database was generated using finite difference method by writing a FORTRAN code. The ANN structure was designed and trained using the MATLAB Neural Network Toolbox. Panigrahi *et al.,* [11], used ANN and fuzzy-logic models to predict statistical quantities such as mean velocity profiles and Reynolds stresses behind a square cylinder mounted in the free stream of a wind tunnel based on hot-wire anemometry readings.

In this paper, a novel fluid flow prediction based on artificial neural networks approach is developed. The computational domain is divided into an equal number of cells with equal size and uniform spacing (similar to a Cartesian grid) and the prediction took place on cell-by-cell basis. The design and development of the cell-by-cell ANN approach is described in detail in this paper, beginning with a description on the governing equations for viscous, incompressible, laminar flows in a two-dimensional computational domain and the discretization of the independent variables on a staggered grid. A feed forward neural networks is tested and Bayesian regulation backpropagation training algorithms is implemented for the new models. The ANN model is evaluated using a number of statistical indices in order to determine the capability and accuracy of the ANN model.

METHODOLOGY

A clear understanding of the problem and well defined model with clear independent variables leads to decent neural network design. The flexibility of Neural networks models for particular application leads to more difficulty in the designing phase due to the specific features of the application. There are various choices have to be decided, but very few guidelines to help the programmer through them. Some assistances can be found in the literature [12-15], but these guidelines are applicable for certain applications and they are never endorsed for alternative applications. In this paper, a viscous, incompressible flow is simulated in a rectangular domain Ω. The laminar flow is governed by whole system of equations (Navier-Stokes) throughout time and the external forces are negligible. The two dimensional Navier-Stokes equations are represented by:

Momentum equations:

$$\frac{\partial u}{\partial t}+\frac{1}{\rho}\frac{\partial p}{\partial x} \quad =\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)-\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+\right)$$

$$\frac{\partial v}{\partial t}+\frac{1}{\rho}\frac{\partial p}{\partial x} \quad =\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)-\left(u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+\right) \tag{1}$$

Continuity equation:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \;\; in\,\Omega \tag{2}$$

The solution of the differential equations depends on the boundary conditions and initial conditions of each case. The inertial forces and viscous forces were ignored since there no change in the geometry and the fluid behavior. From the momentum equations, the velocity vector (u,v) is considered the dependent variables (ANN outputs). Position vector (x,y) and initial velocity condition (u_p) are represent the independent variables (ANN inputs). The velocity boundary condition is ignored since a non-slip wall condition is chosen for all cases.

Neural network design

An ANN designing can be assessed by the architecture of the network and the neurodynamics of the neural network. Neural networks architecture concerns about calculating the suitable number of input neurons, output neurons, hidden layers and the number of the hidden neurons inside each hidden layer. Furthermore, the nature of the connections between each layers and between the neurons themselves is required to be define. Neurodynamics describes the transfer functions behaviour for both neurons and layers, error calculation functions and the data scaling functions [16].

The number of hidden layers and the number of neurons within each hidden layer are changing according to the complexity of the tasks which the (ANN) model should perform [16, 17]. All the calculations process in the neural networks is done in the hidden layers, one hidden layer with a sufficient number of hidden neuron is capable of predicting any continuous function. In addition of increasing the computational cost, increasing the number of hidden layers also compromise the network ability to generalize. Overfitting happens when the neural network is not able to learn the general patterns and commit to individual points (local minima). The poor predicting performance is caused by many computations nodes calculating small observations which decrease the degrees of freedom. In other words, the number of weights, and the size of the training set determine the generalization for the model. Therefore, just one and two hidden layers is tested in this paper. An adjustment to the input variables will be made, if the two hidden layers turn to be insufficient. Lapeds and Forber [18], and Hecht-Nielsen [19] suggested that one or two hidden layers with an suitable number of neurons is sufficient to model any non-linear systems and adding more layers will not improve the results.

An experiment is conduct to choose the neural network that performs best with the least number of hidden neurons. Several neural networks are created with

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

different numbers of hidden neurons. After the training the performance of each neural network is plotted versus the number of hidden neurons (Figure-1). From the error graph, the network with the least error is selected.

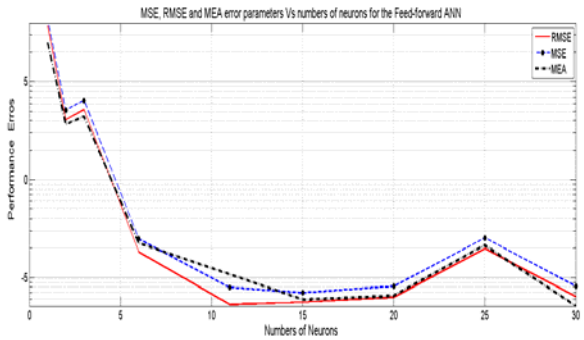

Figure-1. Errors parameters vs numbers of neurons.

Figure-2 illustrates MAPE, RMSE and MEA error parameters values between the simulation data and (ANN) model predictions with different neurons numbers for overall dataset. The objective of calculating these parameters is to measure the differences between the predicted data and the simulation data. Although some statistical parameters are already used in the training phase, but these extra parameters will indicate clearly if there is overfitting in the (ANN) model.

The purpose of neural network training is to determine the weights vector that satisfy the global minima of the error function. The training process has to differentiate between all the local minima and the global minimum or it will compromise the algorithm generalization. Generalization is the idea that a model based on a sample of the data is suitable for predicting the general population. The learning methods includes iteratively reduce the error function between identified inputs neurons and the known output neurons. Two parameters controlled the training process the momentum term and the number of training iterations. The momentum term deals with the random starting weights vectors in order to increase the chances of finding the global minima. The number of training iterations determines the point which the training should be stopped without getting trapped in some local minima.

CBC model implementation

Discretization is required in order to solve the system of equations (1), (2) and (3) using any numerical techniques such as finite difference methods. The solution is applicable only after transforming the continuous equations in the entire domain into their discrete equivalents for each individual cell. All the calculations will be done at cell level, then all the discrete outcomes would be combined to provide the solution for entire domain. On a staggered grid, the cell borders contain the normal velocity components and temperature and pressure, are stored at cell centres [20]. The initial conditions and the boundaries condition is placed on ghost cells stripe outside the domain. The purpose of these ghost cells is to compute the desire Navier-Stokes solutions throughout the entire computational domain from adjacent cells.

The cell by cell approach is initiated based on the specifics nature of thermal fluid flow. The same previous discretization is applied in order to change the function space domain from continues to discrete. In (ANN) cell by cell approach, the calculation is done at the cell level as well, but the model would be already trained before running the simulations. The independent variables are discretized using the same staggered grid in finite difference methods. The entire computational domain can be covered by using this three adjacent cells technique. Similarly, to other numerical approaches, an interpolation procedure might be introduced to calculate missing values of the dependent variable in the spaces outside the computational domain.

Collecting the fundamental data for the (ANN) variables is difficult task to obtain. Many factors control this process like the design of (ANN), the dependent variables we use, the applications and the outcomes of the (ANN). In the case of two dimensional steady state laminar flow the data is extracted from Fluent ANSYS® Academic Research, Release 15.0. ANSYS® is well-known software with a reputation of providing high quality CFD simulations. The training data is exported from ANSYS® to ASCII file and Excel file. For each different case the data were randomly divided into training set, testing set and validating set. The training data set is used to adjust the (ANN) weights and biases according to their error functions. The validation data are used to measure network generalization, and to halt training when the generalization is stop improving. The testing data set provides an independent measure of network performance during and after training.

The (ANN) model results are obtained by applying some errors minimizing functions such as the sum of squared errors and least absolute deviations. These error functions are not considered as the final evaluation criteria since other common forecasting evolution methods such as the mean absolute percentage error (MAPE) are typically not decreased in neural networks. Different error assessing factors are tested in the validation phase for the training data and for the (ANN) model result data.

RESULTS

The (ANN) training data is generated using computational fluid dynamics methods (fluent Ansys and from numerical code in MATLAB). A total of 221 nodes is collected from predefined rectangular domain with dimensions ($1\times\sqrt{2}$) for scaling and interpolating purposes. A belt of the boundaries conditions nodes is added to the predefined location. For each different case the data were randomly divided into training set, testing set and validating set. The training data set is used to adjust the (ANN) weights and biases according to their error functions. The validation data are used to measure network generalization, and to halt training when the generalization

VOL. 11, NO. 20, OCTOBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

is stop improving. The testing data set provides an independent measure of network performance during and after training.

The regression factor is defined as a statistical process for estimating the relationships between a dependent variable and one or more independent variables. The regressions between the simulation data and the predictive data is showed in the Figure-2. The predictive data accomplish great conform to the existing data with regression factor equal R=0.99789 for the feed forward (ANN).

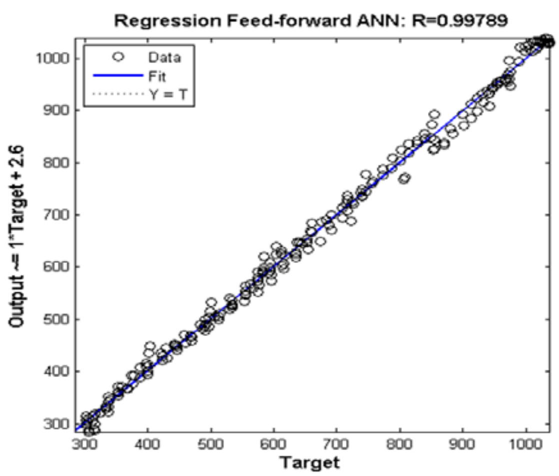

Figure-2. The regressions between the simulation data and the feed-forward (ANN) data.

A set of error parameters is calculated in order to measure the performance of a model fitting to the existing data. The (ANN) model with the highest accuracy (the smallest error parameter values) is chosen as the best network configuration. These parameters are calculated by measure the difference between the predictive quantities and the current values. These parameters (statistical indexes) can be mathematically expressed by equations (3) -(11).

$$MAE\ (mean\ absolute\ error) = \frac{1}{N}\sum_{i=1}^{N}\left|T_i^{predicted} - T_i^{true}\right| \tag{3}$$

$$MSE\ (mean\ squared\ error) = \frac{1}{N}\sum_{i=1}^{N}\left(T_i^{predicted} - T_i^{true}\right)^2 \tag{4}$$

$$RMSE\ (root\ mean\ squared\ error) = \sqrt[2]{\frac{1}{N}\sum_{i=1}^{N}\left(T_i^{predicted} - T_i^{true}\right)^2} \tag{5}$$

$$MARE\ (mean\ absolute\ relative\ error) = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{T_i^{predicted} - T_i^{true}}{T_i^{true}}\right| \tag{6}$$

$$MSRE\ (mean\ squared\ relative\ error) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{T_i^{predicted} - T_i^{true}}{T_i^{true}}\right)^2 \tag{7}$$

$$RMSRE (root\ mean\ squared\ relative\ error) = \sqrt[2]{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{T_i^{predicted} - T_i^{true}}{T_i^{true}}\right)^2} \tag{8}$$

$$MAPE\ (mean\ absolute\ percentage\ error) = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{T_i^{predicte\ d} - T_i^{true}}{T_i^{true}}\right|\% \tag{9}$$

$$MSPE\ (mean\ squared\ percentage\ error) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{T_i^{predicted} - T_i^{true}}{T_i^{true}}\right)^2\% \tag{10}$$

$$MSPE\ (mean\ squared\ percentage\ error) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{T_i^{predicted} - T_i^{true}}{T_i^{true}}\right)^2\% \tag{11}$$

The mean square error smallest value achieved when 15 hidden neurons are used, but the ratio of the error value change between 11 neurons and 20 neurons is small. The root mean square error gradient shows the increasing value of the error for one neuron is insignificant in the range from minimum value at 11 hidden neurons to 20 hidden neurons. In the interval between 15-20 neurons the upward sloping line for the mean absolute error increasing value from the lowest value can be neglected. Therefore, the error minimum values are reached when choosing an (ANN) model with 15 hidden neurons. Table-1, shows the error does not tend towards zero but to a limit as the number of neurons reach the optimal number. The changes in error values versus number of neurons subject to many parameters other than the number of neurons.

The (ANN) models is tested to measure it capability for predicting the flow velocity vectors. Figure-3 illustrates the general architecture of Feedforward (ANN).

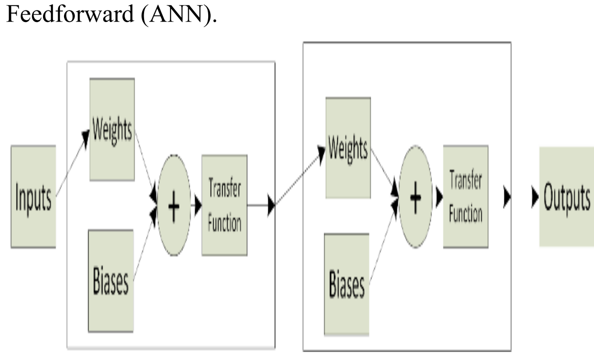

Figure-3. The architecture of Feed-forward (ANN).

ARPN Journal of Engineering and Applied Sciences

Table-1. The error parameters for Feed-forward (ANN).

Number of neurons	1	10	15	20	25	30
MAE	3752.3724	5.2913662	5.6145354	6.2784210	19.788357	6.3695846
MSE	20306639	150.21244	209.11806	411.51367	3574.5816	343.30493
RMSE	4506.2881	7.9004653	7.0094557	8.1285122	25.458341	8.1191367
MARE	25.052760	0.1088538	0.0594190	0.0648006	0.2131305	0.0512575
MSRE	267535.01	0.000505	0.000311	14.415290	110.7504	5.9149186
RMSRE	517.15221	0.021506	0.017467	1.5207092	4.8332301	1.2875699
MAPE	2505.2760	1.46353	1.176208	6.4800638	21.313053	5.1257525
MSPE	2675350171	5.046027	3.105251	144152.90	1107504.3	59149.186
RMSPE	2.081657	2.150637	1.746695	152.07092	483.32301	128.75699

For the flow velocity vectors, the differences between the simulation data and the predictive data are showed in the following Figures (4-6).

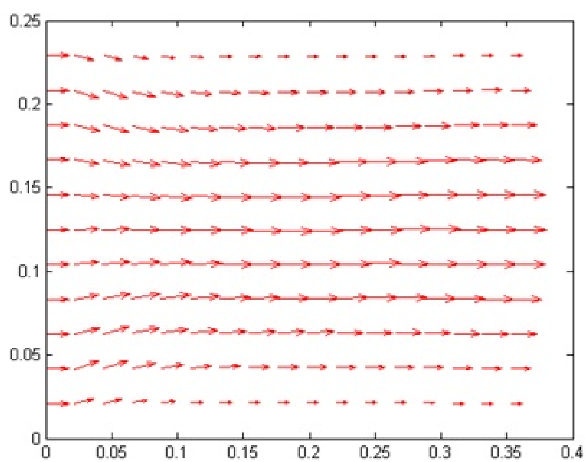

Figure-4. The feed-forward (ANN) predictive velocity vectors.

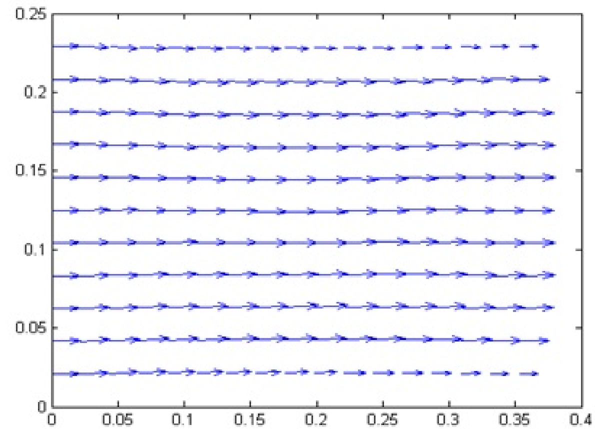

Figure-5. The CFD simulation velocity vectors.

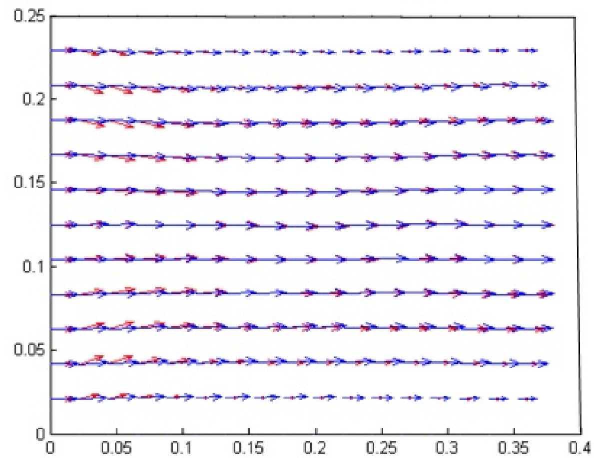

Figure-6. The differences between CFD simulation and the (ANN) velocity vectors.

CONCLUSIONS

In the present study, a new (ANN) approach is suggested for calculating the velocity vectors of the fluid flow. The new (ANN) model is validated by comparing the velocity vectors prediction outcomes with the simulation data. The (ANN) model manage to predict the velocity vectors in the defined boundary domain. The excellent outcomes prove the concept of (ANN) capabilities to predict the flow velocity vectors with acceptable margin of error and in less time. The new approach is a promising to be implanted on transient flow and predict the velocity vector through time. The optimal (ANN) structure for the velocity vectors should be investigated for different types of flow other than Laminar.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support provided by the Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme (FRGS) (Grant no.: FRGS/2/2014/TK01/UTP/02/3). The authors wish to extend their greatest appreciation to the staff of the High Performance computing Centre, Universiti Teknologi PETRONAS, for their technical support.

REFERENCES

[1] Pulido-Calvo, I. and M.M. Portela, Application of neural approaches to one-step daily flow forecasting in Portuguese watersheds. Journal of Hydrology, 2007. 332(1–2): p. 1-15.

[2] Zhang, G.P. and M. Qi, Neural network forecasting for seasonal and trend time series. European Journal of Operational Research, 2005. 160(2): p. 501-514.

[3] Benning, R.M., T.M. Becker, and A. Delgado, Initial studies of predicting flow fields with an ANN hybrid. Advances in Engineering Software, 2001. 32(12): p. 895-901.

[4] Benning, R.M., T.M. Becker, and A. Delgado, Principal Feasability Studies Using Neuro \– Numerics for Prediction of Flow Fields. Neural Process. Lett., 2002. 16(1): p. 1-15.

[5] Morshed, J. and J.J. Kaluarachchi, Application of artificial neural network and genetic algorithm in flow and transport simulations. Advances in Water Resources, 1998. 22(2): p. 145-158.

[6] Kretinin, A.V., *et al*. Neuronet's method of weighted residuals for computer simulation of hydrodynamics problems. in Neural Networks (IJCNN), The 2010 International Joint Conference on. 2010. IEEE.

[7] Kretinin, A., Y.A. Bulygin, and M. Kirpichev. Method of Weighted Residuals on the Base of Neuronet's Approximations for Computer Simulation of Hydrodynamics Problems. in Proceedings of IEEE 6th International Conference on Computational Cybernetics ICCC 2008. 2008.

[8] Valyuhov, S., A. Kretinin, and A. Burakov, Neural Network Modeling of Hydrodynamics Processes.

[9] 10. Ben-Nakhi, A., M.A. Mahmoud, and A.M. Mahmoud, Inter-model comparison of CFD and neural network analysis of natural convection heat transfer in a partitioned enclosure. Applied Mathematical Modelling, 2008. 32(9): p. 1834-1847.

[10] Ranković, V. and S. Savić, Application of feedforward neural network in the study of dissociated gas flow along the porous wall. Expert Systems with Applications, 2011. 38(10): p. 12531-12536.

[11] Panigrahi, P., *et al*., Prediction of turbulence statistics behind a square cylinder using neural networks and fuzzy logic. Journal of fluids engineering, 2003. 125(2): p. 385-387.

[12] Palmer, A., J. José Montaño, and A. Sesé, Designing an artificial neural network for forecasting tourism time series. Tourism Management, 2006. 27(5): p. 781-790.

[13] Azoff, E.M., Neural Network Time Series Forecasting of Financial Markets. 1994: John Wiley \& Sons, Inc. 196.

[14] Kaastra, I. and M. Boyd, Designing a neural network for forecasting financial and economic time series. Neurocomputing, 1996. 10(3): p. 215-236.

[15] Zhang, G., B. Eddy Patuwo, and M. Y. Hu, Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 1998. 14(1): p. 35-62.

[16] Nelson, M.M. and W.T. Illingworth, A practical guide to neural nets. Vol. 1. 1991: Addison-Wesley Reading, MA.

[17] Haykin, S. and N. Network, A comprehensive foundation. Neural Networks, 2004. 2(2004).

[18] Lapades, A. and R. Ferber, How neural network works. Neural information processing systems, American Institute of Physics, New York, 1988: p. 442-456.

[19] Hecht-Nielsen, R. Theory of the backpropagation neural network. in Neural Networks, 1989. IJCNN., International Joint Conference on. 1989. IEEE.

[20] Kampanis, N.A. and J.A. Ekaterinaris, A staggered grid, high-order accurate method for the incompressible Navier–Stokes equations. Journal of Computational Physics, 2006. 215(2): p. 589-613.