ANN-based Model of Convection in a Square Cavity

Detailed Study

Tadej Kodelja
Igor Grešovnik

Centre of Excellence for Biosensors, Instrumentation and Process Control

Revision 1, September 2013.
(revision 0: Avg. 2013)

Warning: This is a long document; **please don't send it to a printer**!

Contents:

1 Introduction ... ***1***

2 ANN Modeling Software ... ***2***

3 Natural Convection Basics ... ***4***

4 Generating Data with Physical Model ... ***6***

5 ANNs Trainings and Error Estimation ... ***8***

5.1 Maximum Errors on Different Training Sets ... **8**

6 Error Estimation ... ***11***

6.1 Error Estimation on ANN Trained with 1000 Training Sets ... **11**

6.1.1 Nusselt Number (*Nu*) ... 11

6.1.2 Maximum Velocity in x Direction on Center Line (V_x) ... 12

6.1.3 Maximum Velocity in x Direction on Center Line (V_y) ... 13

7 One Dimensional Parametric Studies ... ***15***

7.1.1 Center Point ... 15

7.1.2 Points on Line ... 19

8 Two Dimensional Parametric Studies ... ***23***

1 INTRODUCTION

This report provides detailed description of results of the artificial neural network-based modeling of the natural convection parameters in a square cavity.
The model was constructed on basis of the Robert Vertnik's physical based numerical simulator [1]-[4], which was used to produce training data. Construction and analysis of the model was performed by the *NeuralShell*, a flexible multilayered interpreter-centered ANN modeling software *Investigative Generic Library* (*IGLib*). Concepts used in this software have evolved in several years of research work on optimization, inverse analysis and approximation models, which finally culminated in the *IGLib* [6]-[14] library that is used as code base for demanding technical applications. The code depicts on several third party libraries used for the solution of different tasks, which are mentioned in the IGLib web page. Among these, the Aforge.Net [15] library is used as the library for neural networks-related tasks.
The process modeled is briefly described in Section 3. Section 2 contains a short description of the software that was used for generation and analysis of the model. In Section 5 we describe the search for the optimal ANN model for this example. In the scope of this, we have trained several ANN on different training data sets, which was done by using the IGLib's generic training modules. Section 5 is dedicated to validation of the results and estimation of errors. Section 7 and 8 contains a detailed depiction of results of the developed ANN model. These two parts are the most interesting for practical use and for industrial people involved in the modeled process.

2 ANN MODELING SOFTWARE

A software for construction and use of ANN-based models has been developed in the scope of this work. The software was designed as to match the challenges and requirements met when solving this kind of problems. In particular, it must provide good flexibility in designing training strategies, filtering training data, verification of results, testing different network layouts, integration with other software, etc. This is crucial when approximating behavior of steel processing systems with large number of processing parameters. Data obtained from such systems is often inaccurate or even corrupted due to practical limitations in acquisition procedures. Response sampling can not be planned in advance but is accommodated to production schedules in the factory, therefore information available may be deficient in some regions of parameter space in order to obtain good response approximation and therefore verification of results plays an important role.

The software is based on *IGLib* [14], a specialized framework for efficient development of demanding technical applications, and inherits the majority of design paradigms from this framework. Some of these concepts historically follow from the experience gained in development of general purpose optimization software [9]-[13], which is subjected to similar requirements regarding the modularity of software and sufficient flexibility of its user interfaces to accommodate to dynamic environment of multidisciplinary research and development. Among the others, this historical concept is the origin of postulation that a suitable interface for defining and solving complex optimization problems should include a full interpreter capability, through which the built-in functionality, packed in modular libraries, is accessed. Two-level interpreter system developed in IGLib on basis of this experience is utilized in our software and enables rapid reactions to changing demands as new knowledge is gained form the results. IGLib also incorporates design of approximation-based modeling utilities emerging from the development of optimization libraries [11]-[13]. In the scope of the framework, the elaborate object oriented design was supplemented by a number of advanced concepts directed towards rapid application development. These include adherence the concepts of .NET frameworks and extensive use of generic programming. Yet another design feature brought by IGLib is typical arrangement of the framework libraries in a number of levels according to criteria such as platform dependency or license restrictions. The software relies on a number of third party libraries that are mainly distributed under permissive open source licenses.

The Aforge.Net library is used as ANN framework [15]. A convenient characteristics of neural networks is that approximation can be performed in two separate stages (Figure 1). In the training stage, the network is trained by using the sampled response (either measured or calculated by a numerical model). In the approximation stage, trained network is used for all subsequent calculations of approximated response at arbitrary values of input parameters. This gives neural networks an important advantage over other modeling techniques, since the second stage if very fast as compared to the first stage. The software takes full advantage of this feature by separating these stages. This is especially important when performing extensive analyses of the considered process on basis of the developed ANN models, or when incorporating the models in automatic optimization procedures [17],[18].

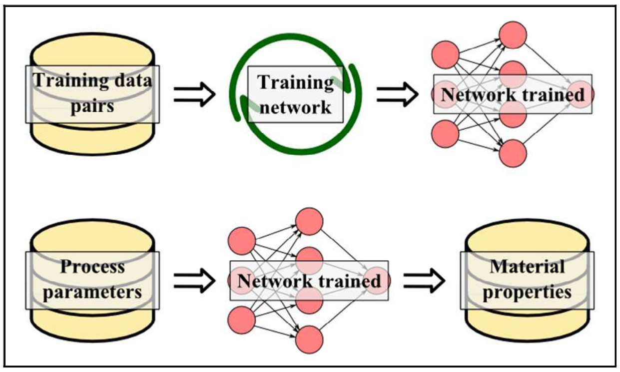


Figure 1: Approximation with neural networks: training a network with presented data pairs (top) and calculation of approximated response with trained network (bottom).

3 NATURAL CONVECTION BASICS

Natural convection is an important physical phenomenon, which can be found in nature and industrial applications, such as furnaces, electronics cooling, materials processing. It is a type of heat transport, in which the fluid motion is not generated by any external source (like a pump or fan) but only by density differences in the fluid occurring due to temperature gradients.

Computational domain is a closed square cavity with fixed height H =1 and fixed width L =1. The cavity is heated with different temperature on vertical walls and isolated on horizontal walls as shown in Figure 2. No slip boundary conditions for the velocity **u** = 0 are applied at the walls. To enhance the accuracy in the boundary layer, the arrangements are refined near the walls with a refinement level *b* =1.2 from the center to the walls. Square cavity is filled with some media which is represented with Prandtl number Pr [1]-[4].

This example is very well established in science and engineering for developing and testing various numerical algorithms and to compare the numerical results. A huge amount of articles were written on this topic [4],[5].

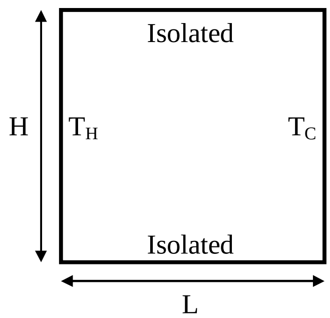


Figure 2: Scheme of the natural convection in a square.

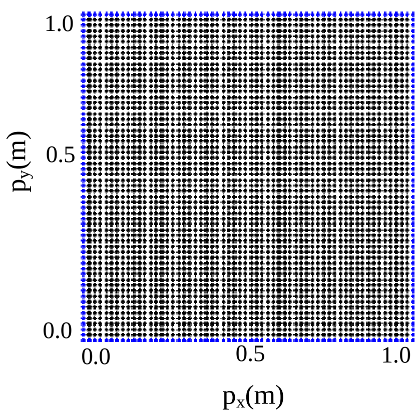


Figure 3: 61x61 node arrangement. Black dots represent domain nodes, blue dots represents boundary nodes.

Process parameters are shown in Table 1 and output values are shown in Table 2 for the current example.

Table 1: Process parameters (input parameters).

VALUES	**NUMBER**
Rayleigh number - Ra	1
Prandtl number - Pr	1
TOTAL	**2**

Table 2: Output values.

VALUES	**NUMBER**
Nusselt number at the hot wall - Nu	1
Maximum velocity in x direction on center line - V_x	1
Maximum velocity in y direction on center line - V_y	1
TOTAL	**3**

4 GENERATING DATA WITH PHYSICAL MODEL

We used a physical simulator for natural convection in a square cavity by dr. Robert Vertnik [1],[2]. The parameter limits for physical simulator are defined in Table 2. On the output side we observed Nusselt number at the hot wall. We ran the simulator with different parameters within the limits from Table 2 many times and save parameters and calculated results in a dedicated file as shown in Figure 13. In total we generated 3000 sets, 2000 sets for training the ANN models and 1000 sets for verifying these models. All ANN models have been verified with the same verification sets.

Table 3: Parameter limits for physical model.

Description	Min Value	Max Value
Pr - Prandtl number	0.7	10
Ra - Rayleigh number	10^3	10^6
Node arrangement (fixed)	61x61	

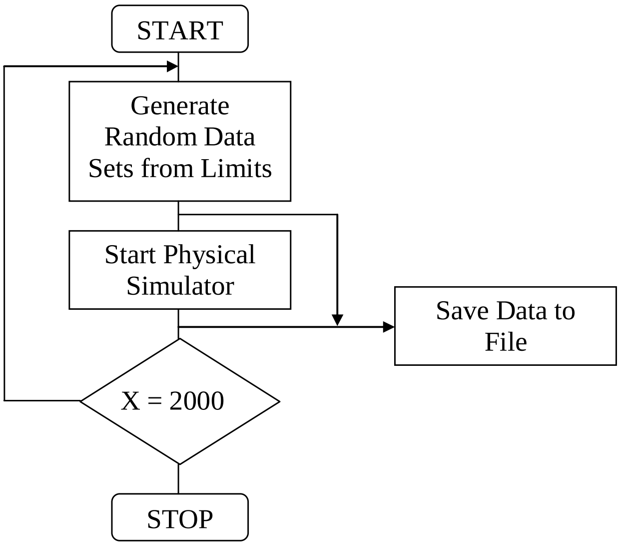


Figure 4: Generating data sets with physical based simulator algorithm.

Sets dedicated for training the ANNs were separated in two groups. 1000 sets were equally randomly distributed in space, other 1000 sets were logarithmically randomly distributed in space as shown in Figure 5.

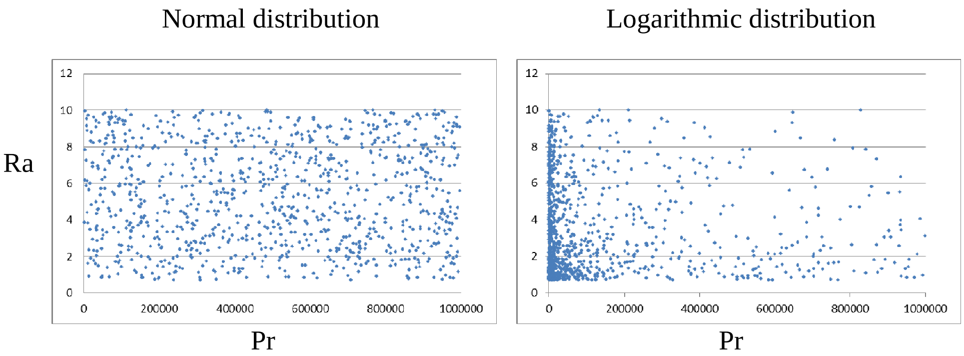


Figure 5: Training data distribution. Left: Normal distribution. Right: Logarithmic distribution

Equal data distribution is usually used for these kind of tests. In this test we generate also logarithmically distributed data, because we notice that small changes on small Prandtl numbers significantly change the Nu number and big changes on big Prandtl number have small effects on Nu number as shown in Figure 6. Therefore in normal distribution we would get oversampling at big Prandtl numbers and undersampling at small Prandtl numbers, while in logarithmically distribution we would resolve this problem.

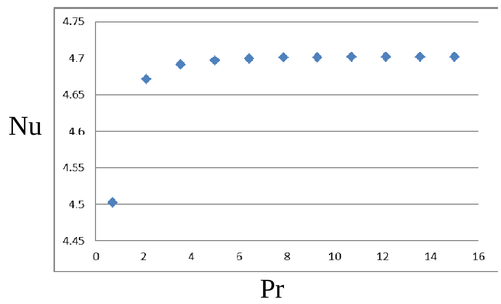


Figure 6: Influence in changing the Prandtl number (Pr) on Nusselt number (Nu).

5 ANNs Trainings and Error Estimation

Before we trained the ANNs we prepared 4 different training packages with 100, 200, 500 and 1000 training sets for normal and logarithmic data distribution. With these we can determine the amount of data sets we need to sufficiently describe the problem. Then we trained ANNs on each of these training sets. The settings we use for setting the ANN are listed in Table 4 and were the same in all tests. All trainings were performed on HP workstation HPDL380G7 with 12 Intel Xenon 2.0 GHz processors.

Table 4: ANN training and architecture settings.

Training parameters	
ANN Type	Feedforward with backpropagation
Activation functions	Sigmoid
Learning rate	0.3
Momentum	0.6
Alpha value	2.0
Number of epochs	150000
Architecture	
Neurons in input layer	2
Neurons in 1st hidden layer	30
Neurons in output layer	3

5.1 Maximum Errors on Different Training Sets

All results for maximum relative errors after trainings were finished are represented in Table 5 - Table 8. Results for errors on training points are represented as:

- Table 5 shows maximum errors on training points on data-sets that are normally distributed in data space.
- Table 6 shows maximum errors on verification points on data-sets that are normally distributed in data space.
- Figure 7 shows maximum errors on training points on normally and logarithmically distributed data-sets.

Table 5: Maximum errors on training points on different normally distributed data-sets.

	100 TP	**200 TP**	**500 TP**	**1000 TP**
Nu	0.0073	0.0130	0.0150	0.0234
V_x	0.0139	0.0146	0.0333	0.0271

V_y	0.0063	0.0074	0.0186	0.0227

Table 6: Maximum errors on training points on different logarithmically data-sets.

	100 TP	**200 TP**	**500 TP**	**1000 TP**
Nu	0.0035	0.0023	0.0015	0.0014
V_x	0.0025	0.0027	0.0053	0.0034
V_y	0.0062	0.0040	0.0060	0.0058

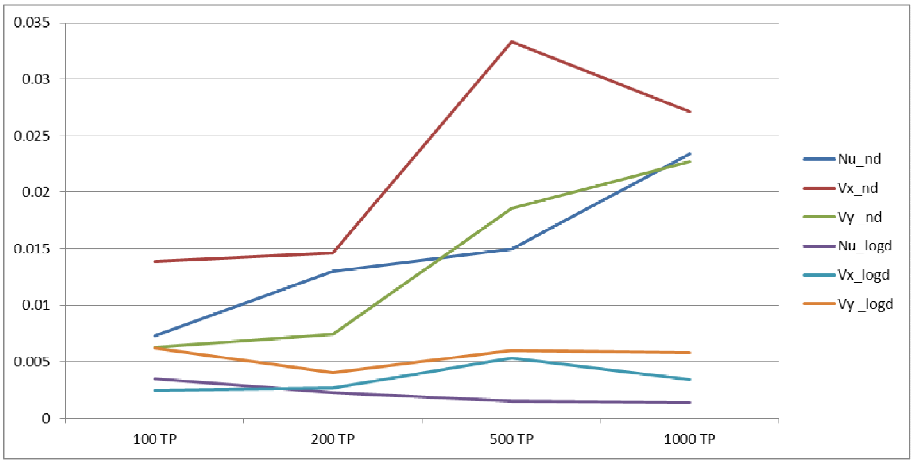


Figure 7: Maximum errors on training points on different data-sets.

Results for errors on verification points are represented as:

- Table 7 shows maximum errors on training points on data-sets that are logarithmically distributed in data space.
- Table 8 shows maximum errors on verification points on data-sets that are logarithmically distributed in data space.
- Figure 8 shows maximum errors on verification points on normally and logarithmically distributed data-sets.

Table 7: Maximum error on verification points on different normally distributed data-sets.

	100 TP	**200 TP**	**500 TP**	**1000 TP**
Nu	0.1338	0.0694	0.0287	0.0222
V_x	0.1866	0.0834	0.0381	0.0223
V_y	0.0637	0.0514	0.0346	0.0252

Table 8: Maximum error on verification points on different logarithmically distributed data-sets.

	100 TP	**200 TP**	**500 TP**	**1000 TP**
Nu	0.0109	0.0050	0.0024	0.0015
V_x	0.0177	0.0130	0.0087	0.0074
V_y	0.0174	0.0129	0.0146	0.0093

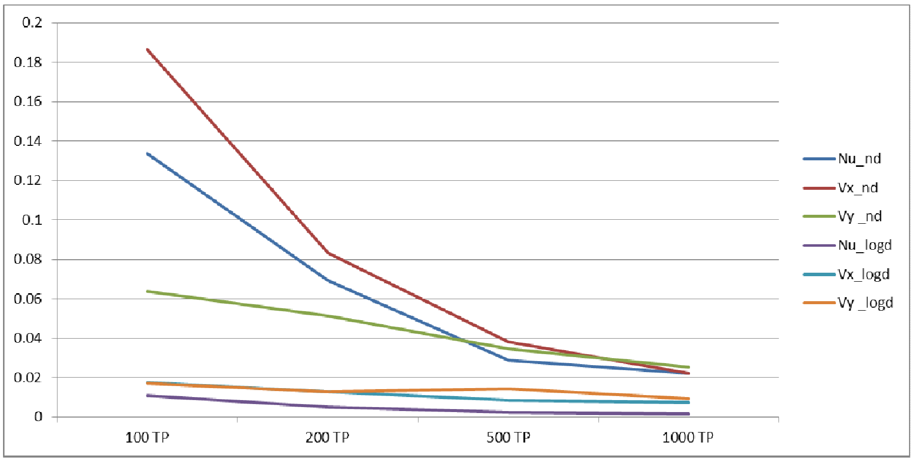


Figure 8: Maximum errors on verification points on different data-sets.

6 ERROR ESTIMATION

After training of the neural network was done, errors of the approximated outputs in training and verification sets were calculated. With these tests we try to determine the accuracy of the ANN.
Relative errors in all training points represents how the ANN is approximating on these points, while relative errors in all verification points represent how the ANN is approximating in the space between training points. These errors are defined as follows:

$$\delta v_i = \left| \frac{v^{(m)}(\mathbf{p}_i) - v(\mathbf{p}_i)}{\max\limits_{j \in I_T}\left(v^{(m)}(\mathbf{p}_j)\right) - \min\limits_{j \in I_T}\left(v^{(m)}(\mathbf{p}_j)\right)} \right|; i \in I_V, , \tag{1}$$

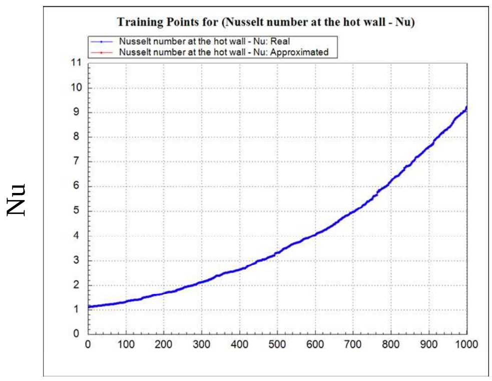
where $v^{(m)}(\mathbf{p}_i)$ is the actual (measured) value of the output quantity v at the vector of input parameters $\mathbf{p}_i$, $v(\mathbf{p}_i)$ is the approximated value of this quantity at the same vector of parameters.

6.1 *Error Estimation on ANN Trained with 1000 Training Sets*

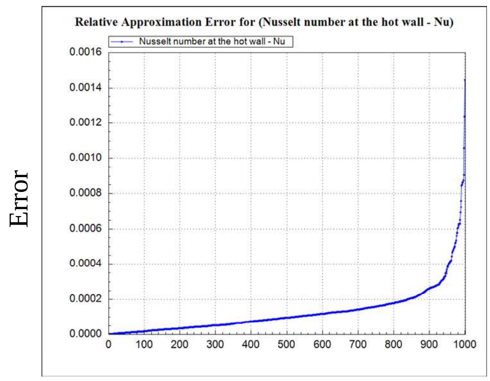
6.1.1 Nusselt Number (*Nu*)

Relative errors for Nusselt numbers (*Nu*) are represented in Figure 9 for training points and Figure 10 for verification points.

6.1.1.1 Errors on Training Points



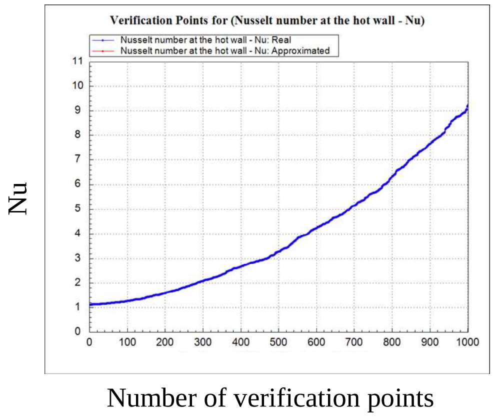
Number of training points



Number of training points

Figure 9: Approximation for Nusselt number (*Nu*) in training points. Training points are represented by dots. Left: real values. Right: relative error in training points.

6.1.1.2 Errors on Verification Points



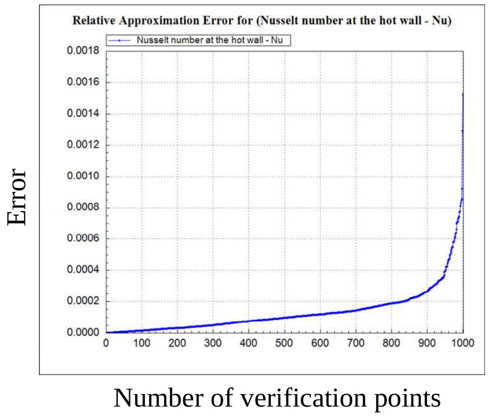


Figure 10: Approximation for Nusselt number (*Nu*) in verification points. Verification points are represented by dots. Left: real values. Right: relative error in verification points.

6.1.2 Maximum Velocity in x Direction on Center Line (V_x)

Relative errors for maximum velocity in x direction on center line (V_x) are represented in Figure 11 for training points and Figure 12 for verification points.

6.1.2.1 Errors on Training Points

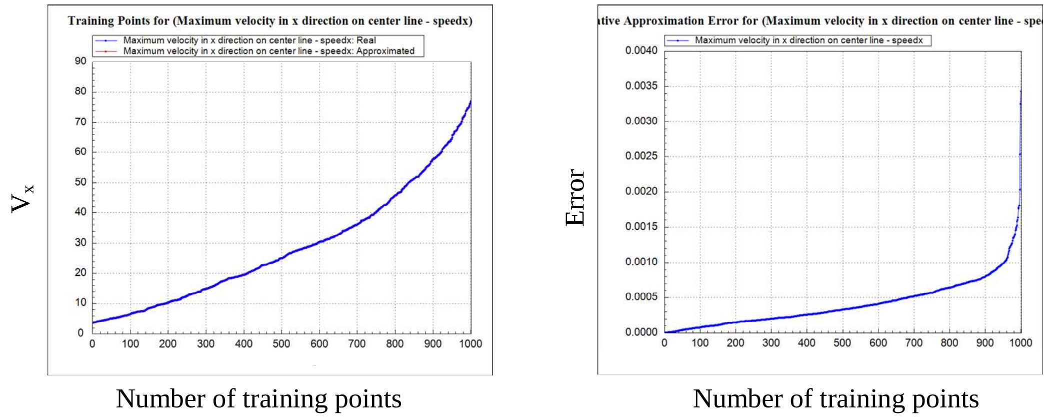


Figure 11: Approximation for maximum velocity in x direction on center line (V_x) in training points. Training points are represented by dots. Left: real values. Right: relative error in training points.

6.1.2.2 Errors on Verification Points

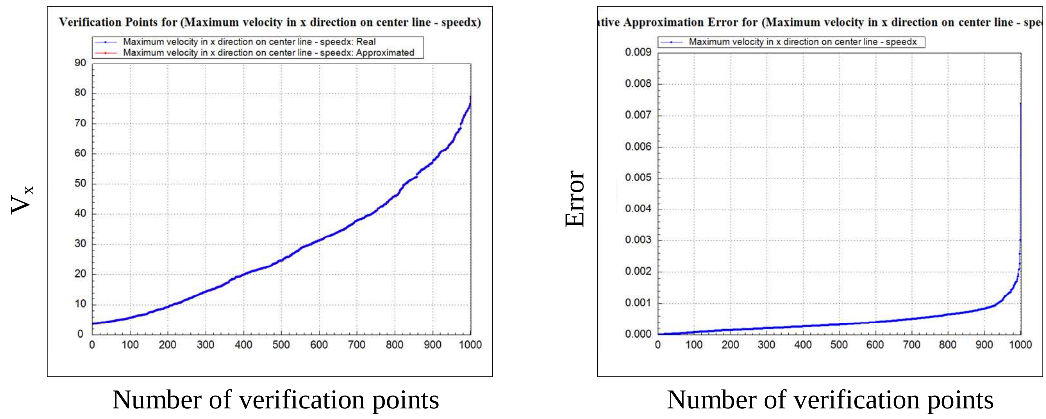


Figure 12: Approximation for maximum velocity in x direction on center line (V_x) in verification points. Verification points are represented by dots. Left: real values. Right: relative error in verification points.

6.1.3 Maximum Velocity in x Direction on Center Line (V_y)

Relative errors for maximum velocity in y direction on center line (V_y) are represented in Figure 13 for training points and Figure 14 for verification points.

6.1.3.1 Errors on Training Points

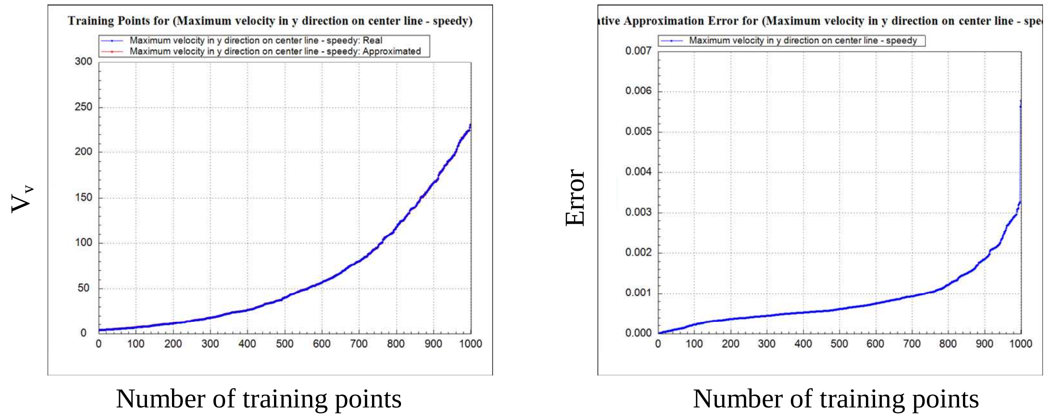


Figure 13: Approximation for maximum velocity in y direction on center line (V_y) in training points. Training points are represented by dots. Left: real values. Right: relative error in training points.

6.1.3.2 Errors on Verification Points

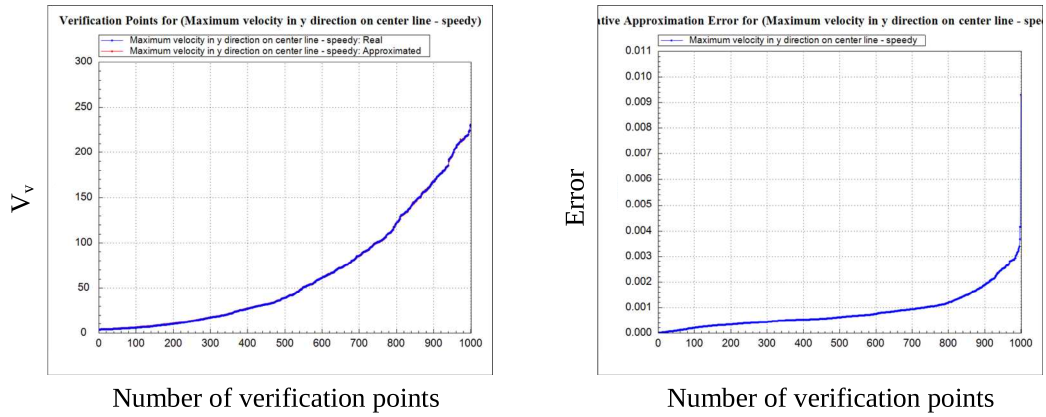


Figure 14: Approximation maximum velocity in y direction on center line (V_y) in verification points. Verification points are represented by dots. Left: real values. Right: relative error in verification points.

7 ONE DIMENSIONAL PARAMETRIC STUDIES

After performing error estimation tests, some parametric studies were performed. With these parametric tests we try to determine the accuracy of the ANN and also verify dependences between parameters.

7.1.1 Center Point

In this study we calculated one center point from training data set and one point from verification data set. Center point for training point was defined as:

$$\mathbf{r}_T = \frac{\sum_{i \in I_T} (\mathbf{p}_i)}{N_T} , \tag{2}$$

while center point for verification set was defined as,

$$\mathbf{r}_V = \frac{\sum_{i \in I_V} (\mathbf{p}_i)}{N_V} . \tag{3}$$

In each chosen point we varied one parameter, while other parameters were fixed. Parameter was varied within the range defined by the minimum and maximum value of that parameter over all dataset used in training. These kind of tests help us find out how the change of one parameter, influences on output values. The influences are shown from Figure 15 to Figure 20.

7.1.1.1 Nusselt Number (*Nu*)

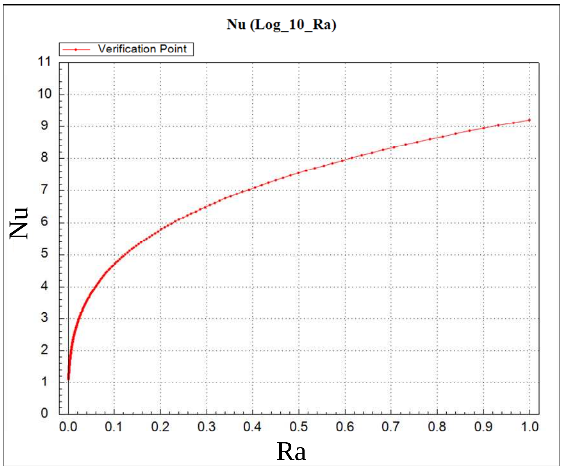


Figure 15: Nusselt number (*Nu*) as a function of the Rayleigh number (*Ra*), calculated by the ANN model on centered verification point.

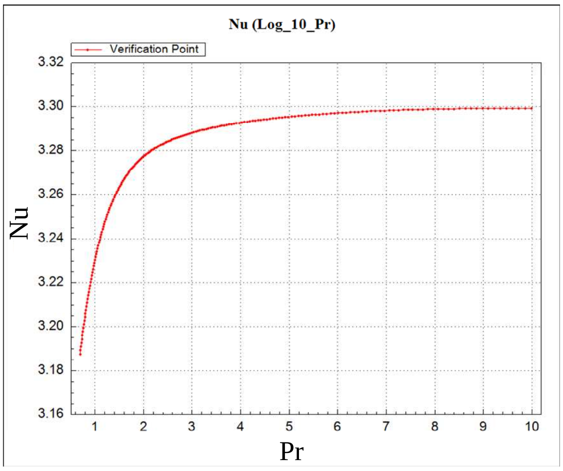


Figure 16: Nusselt number (*Nu*) as a function of the Prandtl number (*Pr*), calculated by the ANN model on centered verification point.

7.1.1.2 Maximum Velocity in x Direction on Center Line (V_x)

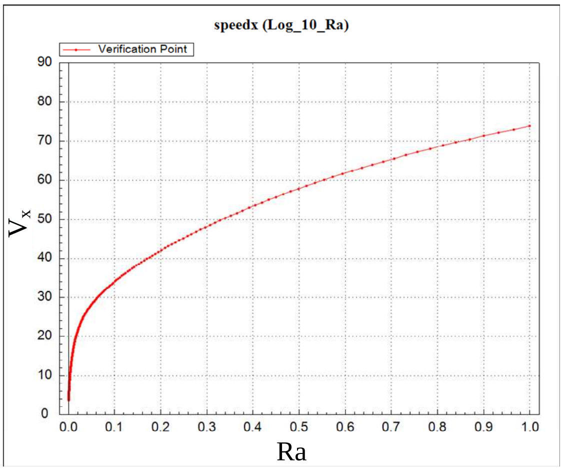


Figure 17: Maximum velocity in x direction on center line (V_x) as a function of the Rayleigh number (*Ra*), calculated by the ANN model on centered verification point.

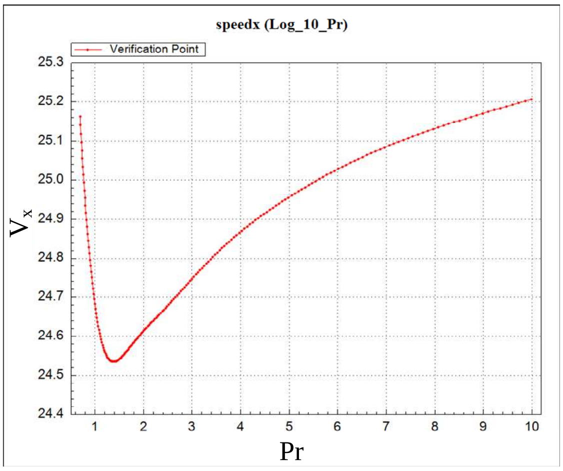


Figure 18: Maximum velocity in x direction on center line (V_x) as a function of the Prandtl number (*Pr*), calculated by the ANN model on centered verification point.

7.1.1.3 Maximum Velocity in y Direction on Center Line (V_y)

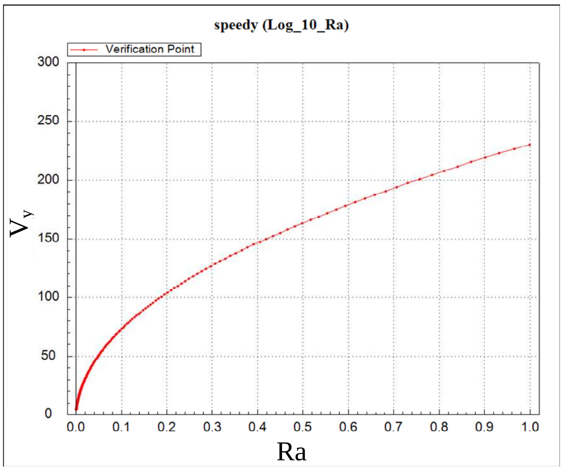


Figure 19: Maximum velocity in y direction on center line (V_y) as a function of the Rayleigh number (*Ra*), calculated by the ANN model on centered verification point.

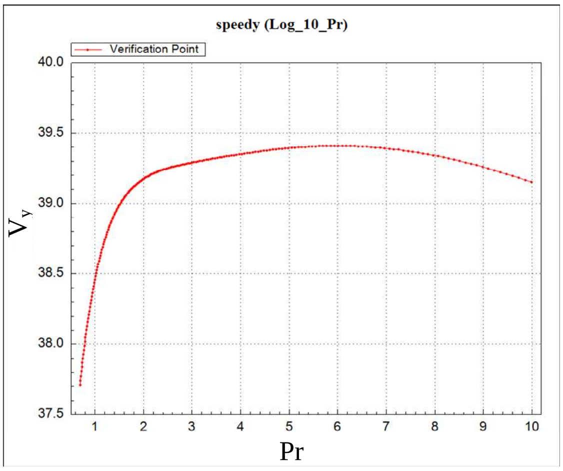


Figure 20: Maximum velocity in y direction on center line (V_y) as a function of the Prandtl number (*Pr*), calculated by the ANN model on centered verification point.

7.1.2 Points on Line

We chose two points $(\mathbf{p}_I, \mathbf{p}_F)$ from the training data-set and from verification data-set. $\mathbf{p}_I$ represents initial point with minimum parameters from elected data-set, and $\mathbf{p}_F$ represents final point, with maximum parameters from that data-set. Then we took a certain number of equally spaced points on the line segment between these two points (including the chosen points). The effects of variation of certain parameters were done for all points and are represented from Figure 21 to Figure 26. The intermediate points $\mathbf{p}_j$ were calculated according to

$$\mathbf{p}_j = \mathbf{p}_I + (\mathbf{p}_F - \mathbf{p}_I)\frac{j}{n+1}; \; j = 0, 1, 2, \ldots, n+1, \tag{4}$$

where *j* is the number of the intermediate points.

7.1.2.1 Nusselt number (*Nu*)

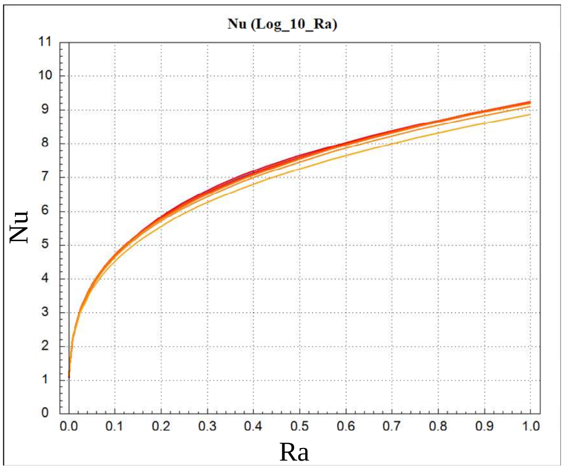


Figure 21: Nusselt number (*Nu*) as a function of the Rayleigh number (*Ra*), calculated by the ANN model in two points from the data set, and for 18 other points on the line between them.

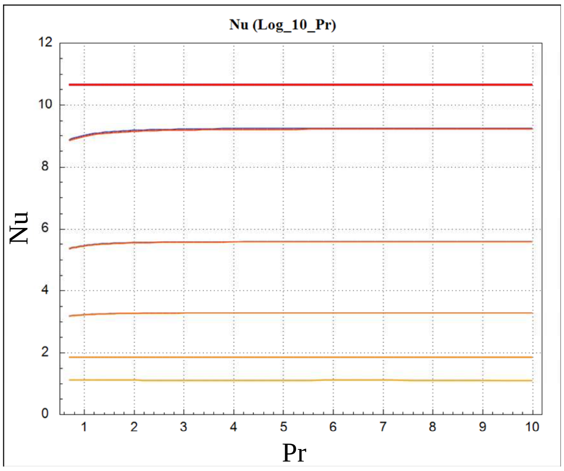


Figure 22: Nusselt number (*Nu*) as a function of the Prandtl number (*Pr*), calculated by the ANN model in two points from the data set, and for 18 other points on the line between them.

7.1.2.2 Maximum Velocity in x Direction on Center Line (V_x)

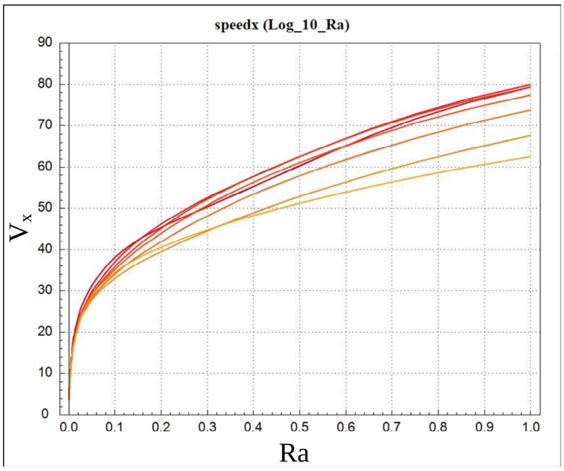


Figure 23: Maximum velocity in x direction on center line (V_x) as a function of the Rayleigh number (*Ra*), calculated by the ANN model in two points from the data set, and for 18 other points on the line between them.

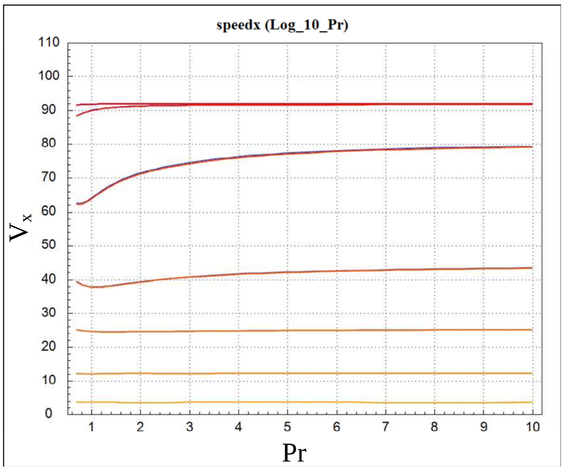


Figure 24: Maximum velocity in x direction on center line (V_x) as a function of the Prandtl number (*Pr*), calculated by the ANN model in two points from the data set, and for 18 other points on the line between them.

7.1.2.3 Maximum Velocity in y Direction on Center Line (V_y)

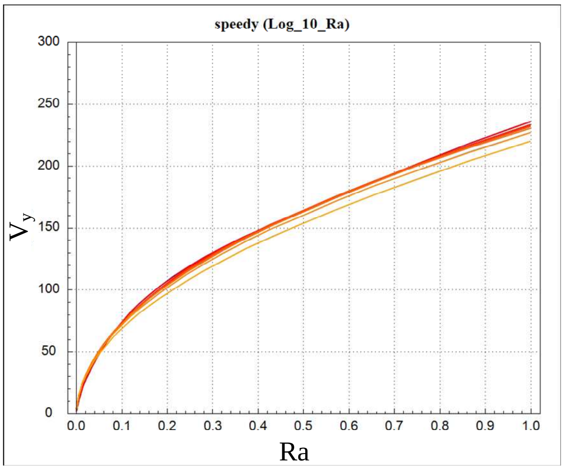


Figure 25: Maximum velocity in y direction on center line (V_y) as a function of the Rayleigh number (*Ra*), calculated by the ANN model in two points from the data set, and for 18 other points on the line between them.

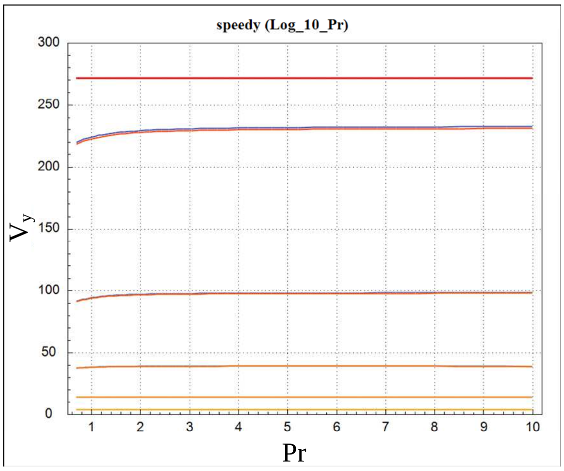


Figure 26: Maximum velocity in y direction on center line (V_y) as a function of the Prandtl number (*Pr*), calculated by the ANN model in two points from the data set, and for 18 other points on the line between them.

8 TWO DIMENSIONAL PARAMETRIC STUDIES

Here we took one central data set from the entire verification data set. In each chosen set we varied both input parameters (*Ra* and *Pr*). The parameters were varied within the range defined by the minimum and the maximum value of each parameter over all data sets used in the training. These kinds of tests help us find out how the changes of parameters influence the output quantities of interest such as the Nusselt number at the hot wall. The influence of the Prandtl number and Rayleigh number on output quantities are shown in figure Figure 27 - Figure 29. All the data are scaled to the [0,1] region.

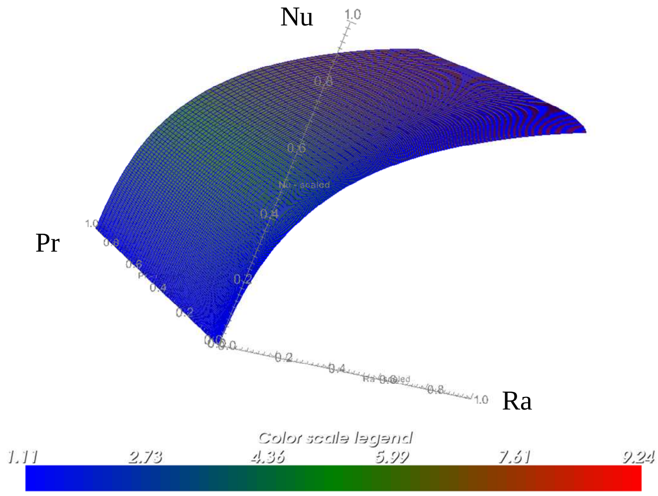


Figure 27: Nusselt number (*Nu*) at the hot wall as a function of the Prandtl number and Rayleigh number (*Ra*) at the hot wall, calculated by the ANN model.

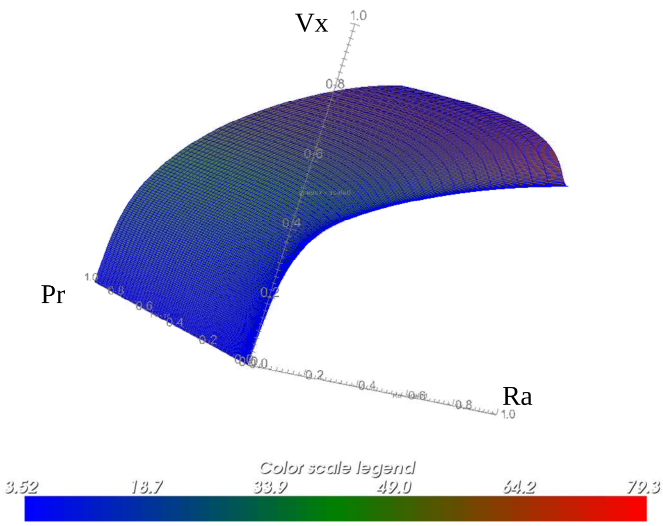


Figure 28: Maximum velocity in x direction on center line (V_x) as a function of the Prandtl ($_{Pr}$) number and Rayleigh number (*Ra*), calculated by the ANN model.

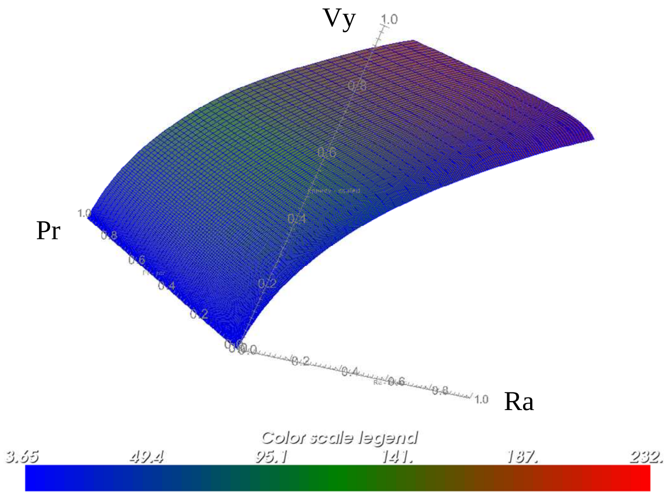


Figure 29: Maximum velocity in y direction on center line (V_y) as a function of the Prandtl number (*Pr*) and Rayleigh number (*Ra*), calculated by the ANN model.

References:

[1] Vertnik, R. (2006) Local collocation method for phase-change problems. Ms.C. thesis, University of Nova Gorica, Slovenia.

[2] Vertnik, R. (2010). Heat and fluid flow simulation of the continuous casting of steel by a meshless method. PhD Thesis. University of Nova Gorica, Slovenia.

[3] Kosec, G. and Šarler, B. (2012). Solution of a low Prandtl number natural convection benchmark by a local meshless method. International Journal of Numerical Methods for Heat & Fluid Flow, vol. 23, pp. 189-204.

[4] Mramor, K.; Vertnik, R. and Šarler, B. (2012). Simulation of natural convection influenced by magnetic field with explicit local radial basis function collocation method. ICCES MM 2012, Budva, Montenegro, September 2-6.

[5] Šarler, B.; Gobin, D.; Goyeau, B.; Perko, J. and Power, H. (2000). Natural convection in porous media—dual reciprocity boundary element method solution of the Darcy model. International Journal for Numerical Methods in Fluids, vol. 33, pp. 279-312.

[6] Grešovnik, I.; Kodelja, T.; Vertnik, R.; Šarler, B. (2012): Application of artificial neural networks to improve steel production process. Bruzzone, A. G.; Hamza, M. H. Proceedings of the 15th International Conference on Artificial Intelligence and Soft Computing, Napoli, Italy, pp. 249-255.

[7] Grešovnik, I.; Kodelja, T.; Vertnik, R.; Senčič. B.; Kovačič, M.; Šarler, B. (2012): Application of artificial neural networks in design of steel production path. Computers, Materials and Continua, vol. 30, pp. 19-38.

[8] Kodelja, T.; Grešovnik, I.; Vertnik, R.; Šarler, B. (2012): Topmost steel production design by using artificial neural networks through process modeling, 20^{th} jubilee conference on materials and technology, Portorož, Slovenia.

[9] Gresovnik, I. (2000): A General Purpose Computational Shell for Solving Inverse and Optimisation Problems - Applications to Metal Forming Processes, Ph. D. thesis, University of Wales Swansea, U.K..

[10] Grešovnik, I. (2012): Optimization shell Inverse, http://www2.arnes.si/~ljc3m2/inverse/.

[11] Grešovnik, I. (2007): The use of moving least squares for a smooth approximation of sampled data. Journal of Mechanical Engineering, vol. 9, pp. 582-598.

[12] Grešovnik, I. (2009): Simplex Algorithms for Nonlinear Constrained Optimization Problems, technical report. Available at: http://www2.arnes.si/~ljc3m2/igor/doc/rep/optalgsimplex.pdf.

[13] Grešovnik, I. (2012): IOptLib – library for solving inverse and optimization problems. Available at: http://www2.arnes.si/~ljc3m2/igor/ioptlib/ .

[14] Grešovnik, I. (2012): IGLib.net - investigative generic library. Available at: http://www2.arnes.si/~ljc3m2/igor/iglib/.

[15] Aforge.Net, artificial intelligence library (2012). Available at: http://www.aforgenet.com/.

[16] Grešovnik, I.; Kodelja, T.; Vertnik, R.; Šarler, B. (2012): A software framework for optimization parameters in material production. Applied Mechanics and Materials, vol. 101, pp. 838-841.

[17] Grešovnik, I.; Kodelja, T.; Vertnik, R.; Šarler, B. (2012): A software Framework for Optimization Parameters in Material Production. Applied Mechanics and Materialds, vol. 101-102, pp. 838-841. Trans Tech Publications, Switzerland.

[18] Grešovnik, I.; Kodelja, T.; Vertnik, R.; Šarler, B. (2012): Application of artificial neural networks to improve steel production process. Bruzzone, A. G.; Hamza, M. H. Proceedings of the 15th International Conference on Artificial Intelligence and Soft Computing, Napoli, Italy, pp. 249-255.