Available online at www.sciencedirect.com
ScienceDirect
St. Petersburg Polytechnical University Journal: Physics and Mathematics 1 (2015) 192–198
www.elsevier.com/locate/spjpm

Fresh approaches to the construction of parameterized neural network solutions of a stiff differential equation

Tatiana V. Lazovskaya*, Dmitry A. Tarkhov

Peter the Great St. Petersburg Polytechnic University, 29 Politekhnicheskaya St., St. Petersburg 195251, Russian Federation

Available online 14 August 2015

Abstract

A number of new fundamental problems expanding Vasiliev's and Tarkhov's methodology worked out for neural network models constructed on the basis of differential equations and other data has been stated and solved in this paper. The possibility of extending the parameter range in the same neural network model without loss of accuracy was studied. The influence of the new approach to choosing test points and using heterogeneous complementary data on the solution accuracy was analyzed.

The additional conditions in equation form derived from the asymptotic decomposition were used apart from the point data. The classical and non-classical definitions of the problem were compared by entering a parameter into the complementary data. A new sampling scheme of test point choice at different stages of minimization (the procedure of test point regeneration) under various initial conditions was investigated. A way of combining two approaches (classical and neural network) based on the Adams PECE method was considered.

Keywords: Neural network; Differential equation; Parameterized mathematical model; Hybrid algorithm.

1. Introduction

A methodology of designing neural network models from differential equations or other data (boundary conditions, measurements, etc.) developed by the St. Petersburg Polytechnic University professors Vasiliev and Tarkhov [3] allows solving complex and ill-posed problems of mathematical physics [4–7]. Those showing the most promise are the parameterized neural network models including one or several problem parameters as input variables [6–8] and allowing to simultaneously solve a family of problems with common parameters.

This paper raises and solves some new fundamental questions using a simple modeling task as an example.

First, we studied the possibility of extending the parameter variation range within a single neural network model without loss of accuracy, i.e. without increasing the pool of simultaneously solved tasks.

Second, we investigated how the new approach to choosing test points that we called a special test point regeneration influences solution accuracy.

Third, we continued the study in ref. [3] aimed at refining the solution through the use of heterogeneous complementary data. This is point data of the sought-for

* Corresponding author.
E-mail addresses: tatianala@list.ru (T.V. Lazovskaya), dtarkhov@gmail.com (D.A. Tarkhov).

http://dx.doi.org/10.1016/j.spjpm.2015.07.005

(Peer review under responsibility of St. Petersburg Polytechnic University).

function, including the inaccurate data, which is often the case with real models.

The novel nature of the approach we have adopted is, compared with previous studies [3], that the above-mentioned point data was obtained by an intentionally inaccurate numerical method. Additionally, complementary conditions that are equations obtained through asymptotic decompositions are used along with the point data.

To answer the questions listed above, there was a good reason to primarily consider a simplest modeling problem with an analytical solution that the constructed approximate solutions could be compared to, and then objectively estimate the obtained results.

For this modeling task we chose a stiff first-order differential equation [1]. Studies [2–8] give reason to assume that the conclusions from the comparative analysis of the studied methods and algorithms remain valid for more complex tasks, including the problems of mathematical physics; so taking such a simple problem is justified.

Introducing a parameter into the complementary data of the problem (expressed through an equation) allows, in particular, to compare the classical and the non-classical statements of the problem. In the latter case, the conditions are imposed on the sought-for function outside the domain chosen for the solution. The natural asymptotic behavior of the studied problem is used as a starting point for such a condition. An approximate solution of the problem obtained through one of the classical methods serves as the inaccurate complementary point data.

A neural network consolidates the information both in data and equation forms using the minimizing functional reflecting the quality of a model. Additionally, in this paper we studied a new system of choosing test points at different stages of minimization (the test point regeneration procedure) for different types of input conditions.

2. Neural network models with complementary data

The problems that are commonly difficult to solve by classical explicit methods or require a lot of iterations are particularly interesting. Among the ordinary differential equations (DEs) these are stiff ones [1].

Ref. [1] deals with a classical example of a stiff equation

$$y' = -50(y - \cos x) \tag{1}$$

with an initial condition $y(0) = 0$.

When this problem is solved by the explicit Euler method, a critical value of the grid step equal to 2/50 occurs, above which the approximate solution becomes unstable with large variations (Fig. 1, *a*). At the same time, the error appears to be too large for a smaller step.

We shall focus on a generalized parameterized problem

$$y' = -\alpha(y - \cos x), \tag{2}$$

$$y(0) = 0,$$

where $\alpha \in [5, 50]$ or$\alpha \in [0.5, 50]$, $x \in [0, 1]$.

The problem is stiff for the variable x in the vicinity of 0, which governs the choice of the proper intervals. Test runs showed that the quality of the neural network solution is also preserved for wider intervals. The problem is solved for all examined values of the parameter α. Notice that these intervals of parameter variation are sufficiently wider than those discussed in refs. [6,8].

An approximate solution is sought in the form of an output of an artificial neural network of the given architecture:

$$y(x) = \sum_{i=1}^{n} c_i v(x, \alpha, \mathbf{a}_i),$$

whose weights $\{c_i, \mathbf{a}_i\}_{i=1}^{n}$ are determined when minimizing the error functional

$$\sum_{j=1}^{m} (y'(\xi_j) - F(\xi_j, y(\xi_j), \alpha_j))^2 + \delta y^2(0),$$

and for our case, $F(x, y, \alpha) = -\alpha(y - \cos x)$.

Test points (ξ_j, α_j) are chosen to be random and distributed uniformly over the examined intervals of variation of the value x and the parameter α; their choice is repeated after several (3–5) iterations of the optimization algorithm. We shall define a new random choice of test points at some step as test point regeneration.

The quality of the obtained solution is assessed from the exact analytical solution of Eq. (2) with an initial condition $y(0) = 0$, which takes the form

$$y(x, \alpha) = \frac{\alpha^2(\cos x - \exp(-\alpha x)) + \alpha \sin x}{\alpha^2 + 1}. \tag{3}$$

In the present work, we have examined two types of models corresponding to various basic functions with the varying number of neurons in the network. The first case involved choosing universal sigmoids in the form

$$\mathrm{th}[a(x - d)]\mathrm{th}[a_1(\alpha - d_1)],$$

and the second one asymmetric Gaussians in the form

$$x \exp[-a(x - d)^2] \exp[-a_1(\alpha - d_1)^2]$$

that were known to satisfy the initial condition.

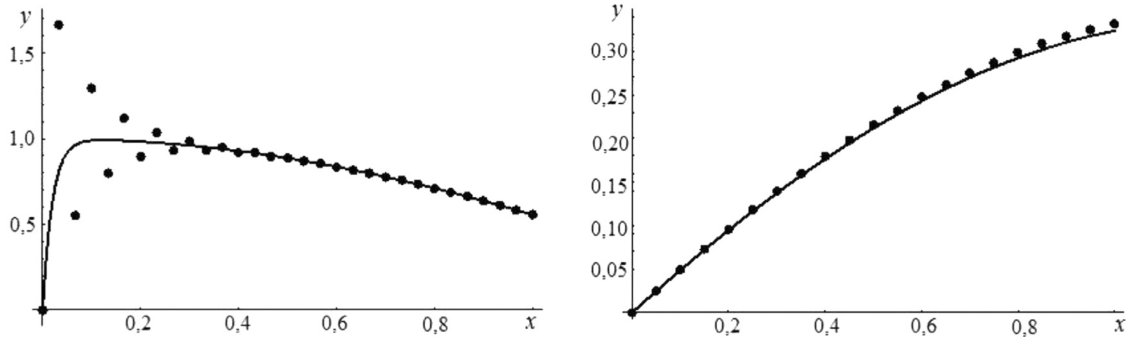

Fig. 1. The solutions of a stiff differential Eq. (2) by the explicit Euler method for $\alpha = 50$ (*a*) and 5 (*b*). The symbols show the pointwise solutions by the Euler method, the lines show the true solutions.

Table 1
Values of the error functional with various data sets for two types of neural networks.

n	Basic model ($\delta_1 = 0$)		1st modification ($\alpha = 50$)		2nd modification ($\alpha = 5$; $\alpha = 50$)	
	Sigmoid	Gaussian	Sigmoid	Gaussian	Sigmoid	Gaussian
5	4.078	1.503	2.176	3.746	1.561	3.376
20	4.312	0.932	2.781	1.226	1.673	2.074
50	8.811	1.787	4.482	1.587	1.26	1.556

Notations: n is the number of neurons, α is a parameter; δ_1 is a weight of the complementary data; for nonzero values of δ_1, the complementary data for various values of α was used. The number of iterations is 200.

The error functional was optimized according to the algorithm combining RPop and the cloud method [2]; the points were randomly regenerated in each three steps, and the cloud consisted of three particles.

It is worth mentioning that the optimization process is complex because the optimized functional changes after each test point regeneration; thus we avoid getting a good approximation for a fixed set of points and a bad one for other points of the examined interval, which may happen when applying the collocation method.

We should stress that the calculation procedure used is doubly stochastic, i.e., the initial weights of the neural network are chosen as random in addition to the aforementioned random test point regeneration.

Additionally, we studied model construction algorithms using the complementary data on the sought-for solution, estimated the effect of such a refinement for various types of basis functions and various numbers of neurons in the network. Matches of the sought-for solutions and the ones already found by the explicit Euler method for the values of the parameter α equal to 5 and 50 were treated as such data. Apparently, for $\alpha = 5$, the equation is no longer stiff and has a sufficiently accurate solution (see Fig. 1, *b*). A 'bad' solution for $\alpha = 50$ allows examining how the model reacts to inaccurate data.

New information is introduced into the model by adding the following complementary summand to the minimizing functional:

$$\delta_1 \sum_{j=1}^{m} (f(x_j) - y(x_j))^2,$$

where $f(x_j)$ is a pointwise Euler solution; a weight δ_1 may be varied, and accuracy of all acquired data or any special conditions should be taken into account.

The above described data for $\alpha = 50$ was used in the first modification of the model, while the data for $\alpha = 5$ was also taken into account for the second modification. Some results of the simulation experiments of the error functional for two types of neural networks are listed in Table 1. We examined networks with a varying number of neurons and with the number of iterations equaling 200.

Evidently, with no complementary data, the option with the basis functions satisfying the initial condition, i.e. the Gaussians, proved to be preferable. Introducing the complementary data increases the accuracy only when universal basis functions, i.e. sigmoids, are used, and vice versa, for the network with the functions adjusted to the initial condition, there is, generally, an increase in error. The effect of using the data for networks with a great number of neurons ($n = 50$) is particularly pronounced.

As for the great error in the column $\delta_1 = 0$ (which means there is no complementary data), for the perceptron network it is explained by the the functional being characteristically sensitive to a steep rise in solution. In this case, increasing the number of neurons does not change the situation. With no or little data, a great number of neurons just slows down the training and impairs the result. Importantly, the results are improved with a sufficient increase in the number of iterations.

Therefore, refining a neural-network solution is possible even if inaccurate data (e.g. approximate numerical solutions obtained through classical methods, including the weak ones like the explicit Euler method) is used as complementary information.

3. Modification of a neural network model using special test point regeneration

Let us continue to refine the model with the complementary data using a new test point regeneration procedure which is selecting the test points at each iteration by a specific rule. Let us introduce the parameter d_t taking the values 0, 0.3, 0.5, 0.7, 1.0 (and generally speaking, any between 0 and 1) and reflecting the proportion of points fixed from one iteration to another. For example, $d_t = 0$ means a complete regeneration, i.e. all points are randomly selected anew (are evenly distributed over the examined interval) before each iteration, $d_t = 1$ means that the points are fixed at the first iteration and do not change. For intermediate values of the parameter the following rule is used: $d_t \cdot m$ points of the total m test points with the highest values of the error functional are fixed, while the rest are regenerated randomly. In all cases, at the first iteration the points are selected to be randomly and evenly distributed over the examined interval.

A perceptron network that had previously proved to be the most sensitive to new complementary data about the model was used in the experiments. The number of neurons in the network was chosen to equal 20 ($n = 20$); the data used was on the correspondence between the sought-for solution and the approximate one, obtained by the explicit Euler method for $\alpha = 5$ and 50. The number of iterations was 300.

Let us introduce the following measure to objectively estimate the obtained results. Since there is a solution in explicit form (3) for our equation, we may compare the solution constructed using the neural networks with the true one. A root-mean-square deviation c found in 10^6 points (α, x) is used as such a measure with α and x evenly distributed over the respective intervals. The selected number of points gives reason to assume that the estimate is stable relative to various samples.

Table 2
Root-mean-square estimate of the quality of the constructed neural-network model.

Regeneration parameter	$d_t = 0$	$d_t = 0.3$	$d_t = 0.5$	$d_t = 0.7$	$d_t = 1.0$
E_c, 10^{-2}	6.39	6.45	6.80	6.62	6.85
s^2, 10^{-2}	0.96	0.59	0.48	0.92	1.25

Notations: E_c, s^2 are the mean and the root-mean-square deviations, d_t is the parameter introduced to reflect the proportion of the points fixed from one iteration to another.
The number of iterations is 300; the number of test points $m = 20$.

A series of tests was held for various values of parameter d_t. The quality of the solutions constructed by the neural network was determined using the above-mentioned root-mean-square estimate c. Table 2 lists the results of the experiments as mean (E_c) and root-mean-square (s^2) deviations of the obtained sample for c. As seen from the table data, the mean error values differ only in the second digit. We may thus postulate that in this case the regeneration by the above-described rule for $d_t = 0.3$ and $d_t = 0.5$ ensures a more stable neural network modeling result.

Evidently, the effect should be enhanced with an increase in dimensionality, i.e. with a transition to partial differential equations. We shall continue analyzing the influence that point regeneration has on the result in the next section.

4. Refinement of the neural network model using an asymptotic condition

Let us note that the relation $y \cong \alpha \sin x$ is true for sufficiently small values of α. Let us regard the thus obtained asymptotic condition as the complementary data for the model. Additionally, let us take into account the correspondence between the sought-for solution and the approximate one obtained by the Euler method for $\alpha = 50$; in other words, let us study how to use the heterogeneous data in the model. Let us also continue to examine the effect of test point regeneration described in the previous section.

We shall take into account the asymptotic condition by adding a summand

$$\delta_2 \sum_{k=1}^{m} \left[y\left(x_k, \frac{\alpha_k}{M}\right) - \frac{\alpha_k}{M} \sin x_k \right]^2$$

to the minimizing functional in the model, where M is a sufficiently large fixed positive number, and the variables x_k and α_k are regenerated in the same way as earlier in the examined intervals $\alpha \in [0.5, 50]$, $x \in [0, 1]$.

Let us give the explicit form of the obtained solution for $m = 20$ and $M = 50$:

$$u(x,\alpha) = 0.095 + 0.103\,\mathrm{th}[4.294\cdot 10^{11}(3.941\cdot 10^{11} + x)]\mathrm{th}[0.086(-49.788+\alpha)] + 0.147\,\mathrm{th}[6.398(-0.37+x)]\mathrm{th} \times [0.112(-45.994+\alpha)] + 0.235\,\mathrm{th}[4.21(-0.76+x) \times \mathrm{th}[0.025(-45.33+\alpha)] -0.787\,\mathrm{th}[1.88(-0.305+x)]\mathrm{th}[0.046(-34.378+\alpha)] + 0.083\,\mathrm{th}[4.474(-0.643+x)\mathrm{th}[0.149(-31.034 +\alpha)] + 0.579\,\mathrm{th}[1.523(-0.503+x)\mathrm{th}[0.166(-26.922+\alpha)] - 0.934\,\mathrm{th}[0.083(-0.054+x)]\mathrm{th} \times[0.565(-26.823+\alpha)] - 0.163\,\mathrm{th}[1.818(-0.533+x)]\mathrm{th}[0.246(-25.466+\alpha)] - 0.819\,\mathrm{th} \times[2.017(-0.88+x)]\mathrm{th}[0.14(-23.668+\alpha)] + 0.577\,\mathrm{th}[0.938(-0.831+x)]\mathrm{th}[0.237(-23.39+\alpha)] -0.012\,\mathrm{th}[27.835(-0.284+x)\mathrm{th}[0.407(-22.059+\alpha)] - 0.276\,\mathrm{th}[1.099\cdot 10^{62}(2.145\cdot 10^{65}+x)]\mathrm{th} \times[0.091(-11.042+\alpha)] - 0.317\,\mathrm{th}[1.464(-0.569+x)]\mathrm{th}[0.244(-9.012+\alpha)] - 0.105\,\mathrm{th} \times[5.303(-0.489+x)]\mathrm{th}[0.576(-3.596+\alpha)] + 0.893\,\mathrm{th}[1.588(-0.798+x)]\mathrm{th}[0.105(-3.546+\alpha)] + 0.23\,\mathrm{th}[1.061(-0.581+x)]\mathrm{th}[0.341(-2.658+\alpha)] + 0.445\,\mathrm{th}[62.244(-0.025+x)]\mathrm{th} \times[(0.084(-1.548+\alpha)] - 1.009\,\mathrm{th}[1.812(-0.976+x)]\mathrm{th}[(0.121(-1.166+\alpha)] - 0.026\,\mathrm{th}[1.222 \times(-0.375+x)]\mathrm{th}[(16.58(-0.559+\alpha)] + 0.849\,\mathrm{th}[0.595(-0.056+x)]\mathrm{th}[2.416(-0.55+\alpha)].$$

The above-described approach was investigated for a network with 20 basis functions ($n = 20$) and 20 and 50 test points ($m = 20$ and 50). The neuron network was used with the asymptotic condition for $M = 50$, 100 and 200. Apparently, the value $1/M$ for these cases will be beyond the interval in which the problem (2) is supposed to be executed, and so we can say with reasonable confidence that the non-classical problem is solved.

A series of tests was held for each set of parameters. The quality of the solutions constructed by the network was determined using the above-mentioned root-mean-square estimate. The results of the experiments are listed in Table 3. Only the case of using the asymptotic condition for $M = 50$ is listed for a model with 50 test points, as no discernible differences in the results have been found for other values of M. Obviously, for high values of the parameter M, the problem is largely non-classical, which means the results deteriorate at full regeneration for $m = 20$. The method with the partial test point regeneration turns up better results than the collocation method that produces the greatest error. Significantly, it is for the model with the fixed points when $m = 50$ that we obtain a result as good as the one for complete regeneration with $m = 20$. The training time highly increases as it linearly depends on the number of test points. Thus, other things equal, the complete regeneration makes for decreasing the algorithm's running time by reducing the number of control points while retaining the accuracy of the result.

Error increase at the complete regeneration of 50 test points suggests that there is some network retraining, i.e. the method and the number of points should be chosen correctly in accordance with the conditions of the problem. Partial regeneration may be successfully applied for a non-classical problem, thus allowing to modify the model at the 'complex' points.

It is particularly interesting to compare neural network solutions constructed using various types of complementary data at specific values of the root-mean-square error. Fig. 2 shows neural network approximations for the models with the parameters $m = 20$, $d_t = 0$ without applying the asymptotic condition, and with

Table 3
Mean-root-square estimate of the quality of neural network models designed taking into account the asymptotic condition at various values of the parameter M and the regeneration parameter d_t.

M	Regeneration	$d_t = 0$	$d_t = 0.3$	$d_t = 0.5$	$d_t = 0.7$	$d_t = 1.0$
$m = 20$						
50	E_c, 10^{-2}	5.51	6.83	8.38	7.36	7.54
	s^2, 10^{-2}	0.41	0.99	0.32	0.83	1.24
100	E_c, 10^{-2}	7.03	6.84	8.13	6.87	9.12
	s^2, 10^{-2}	0.68	1.25	1.70	1.10	1.35
200	E_c, 10^{-2}	7.71	7.09	6.96	6.68	7.63
	s^2, 10^{-2}	1.49	0.96	1.27	1.01	1.95
$m = 50$						
50	E_c, 10^{-2}	6.13	6.05	6.23	6.96	5.62
	s^2, 10^{-2}	0.95	0.46	1.18	0.71	0.27

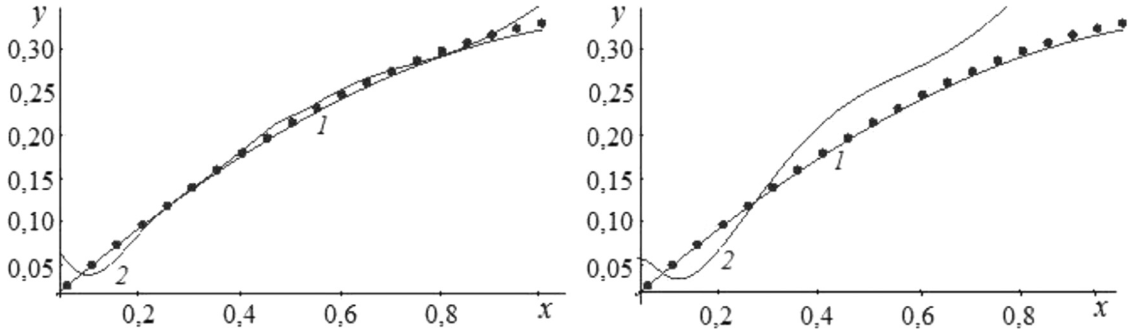

Fig. 2. Neural network approximations of the solution obtained using point data (*a*) and the asymptotic condition (*b*) for $\alpha = 0.5$ and at the same root-mean-square error of 0.027. The parameters of the models $m = 20$, $d_t = 0$. The symbols show the pointwise solutions by the Euler method, the lines *1* and *2* show the true and the neural network solutions, respectively.

applying it at $\alpha = 0.5$ and the same root-mean-square error. The results in Fig. 2 demonstrate that the neural network model designed using the asymptotic condition solves a non-stiff problem with less accuracy. At the same time, its mean error estimate over the entire interval of parameter variation is less (see Tables 2 and 3), which suggests a more uniform approximation to the true solution over the whole examined range.

Notably, this effect remains as well for the asymptotic model at $M = 200$ (non-classical problem), where the approximation for small values of α is even cruder for the equal values of the root-mean-square estimate.

Thus, the neural network successfully solves non-classical problems, while the asymptotic model produces a more uniform distribution over the whole interval of parameter variation.

5. The way of combining the neural network and the classical methods

It is possible to take one more hybrid approach that uses the neural network approximation $u(x, \alpha)$ to solve the problem (2) while modifying the implicit Adams method of second-order accuracy for the solution of the differential equation $y' = f(x, y)$:

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, y_{i+1}))$$

With the classical approach, the variable y_{i+1} is given implicitly and requires applying some method at each step in order to solve this non-linear equation. The predictor–corrector method involving two-stage calculations at each step is most commonly used in this case. The Euler method is used as the first stage, and

$$\hat{y}_{i+1} = y_i + hf(x_i, y_i)$$

is calculated, while the Adams method formula

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, \hat{y}_{i+1}))$$

is used as the second.

Let us use the neural network approximation $u(x, \alpha)$ to replace two formulae with one:

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, u(x_{i+1}, \alpha)))$$

This approach has been used and analyzed for neural network models with the perceptron network and the basis Gaussian functions for various numbers of neurons, and also with and without the complementary data. Let us now describe the most interesting results.

As expected, without the complementary data, applying the neural network with Gaussians yielded the best result for the hybrid method in question. The number of neurons $n = 5$ is apparently insufficient, and there is an overtraining effect at $n = 50$. At $n = 20$ there is a significant decrease in error for $\alpha = 50$, and thus the hybrid method allows to improve the result of the predictor–corrector method for the stiff case.

As for the perceptron network, the effect of applying the hybrid method occurs in the model using the complementary data at values of $\alpha = 5$ and 50 (see Section 2). Even for a network with a small number of neurons, $n = 5$ improves the result for a stiff equation at $\alpha = 50$, compared to the neural network and the classical methods, while this is not observed for small values of α. It appears that the reason for this is the lack of neurons in the network, as for $n = 20$ and 50 the best result for the stiff case is retained and the accuracy of the classical methods is achieved for the equation for the small values of parameter α.

6. Conclusion

The conducted study established that applying the neural network approach when designing a mathematical model from heterogeneous data including differential equations allows, by modifying the error functional, to take into account various types of complementary conditions without substantially altering the algorithm.

Neural networks provide a way of solving nonclassical problems and model construction problems from inaccurate data, which is often the case with actual applications. For the examined problem it turned out that a perceptron network with basis sigmoid functions (also known as universal approximators) was the most responsive to complementary data.

The algorithm of test point regeneration suggested in the paper ensures saving the network running time in simple problems and refining the model if there is any complex complementary data.

A hybrid algorithm using approximated neural network solutions in classical implicit methods is effective for neural networks producing sufficiently accurate approximations. In this case the algorithm may serve to considerably increase the accuracy of an approximate solution in a discrete point set.

The methodology presented in book [3] offers the possibility of naturally generalizing of the approach introduced in the present paper to systems of ordinary differential equations, equations of higher order, and partial derivative equations.

Acknowledgment

The study was conducted with the financial support of the Russian Foundation for Basic Research (grant no. 14-01-00660).

References

[1] E. Hairer, G. Wanner, Reshenie obyknovennykh differentsial'nykh uravnenij. Zhestkie i differentsial'no-algebraicheskie zadachi [The solution of ordinary differential equations. Stiff and differential algebraic problems], Mir, Moscow, 1999.

[2] D.A. Tarkhov, Nejrosetevye modeli i algoritmy [Neural network models and algorithms], Radiotekhnika, Moscow, 2014.

[3] A.N. Vasiliev, D.A. Tarkhov, Nejrosetevoe modelirovanie. Printsipy. Algoritmy. Prilozheniya [Neural network modeling. Principles. Algorithms. Applications], Polytechnical University Publishing House, St. Petersburg, 2009.

[4] A.N. Vasiliev, D.A. Tarkhov, Postroenie priblizhennykh nejrosetevykh modelej po raznorodnym dannym [Construction of approximate neural network models according to heterogeneous data], Math. Model. Comput. Simul. 19 (12) (2007) 43–51.

[5] A.N. Vasiliev, V.P. Osipov, D.A. Tarkhov, Unified process of modeling of physicotechnical objects with distributed parameters, St. Petersburg State Polytechnical University J.: Phys. Math. 3 (104) (2010) 39–52.

[6] A.N. Vasiliev, D.A. Tarkhov, Parametrical neural network models of classical and nonclassical problems for heat conduction equation, St. Petersburg State Polytechnical University J.: Phys. Math. 3 (153) (2012) 136–144.

[7] D.A. Tarkhov, A.N. Vasiliev, Mathematical models of complex systems on the basis of artificial neural networks, Nonlin. Phenom. Complex Syst. 17 (3) (2014) 327–335.

[8] A.N. Vasiliev, D.A. Tarkhov, Mathematical models of systems with interval representation of parameters on the basis of heterogeneous neural networks, Porous catalyst, Neurocomput.: Design Appl. 5 (2014) 3–7.