

# How to use neural network and web technologies in modeling complex technical systems

M G Semenenko<sup>1</sup>, I V Kniazeva<sup>2</sup>, L S Beckel<sup>3</sup>, V N Rutskiy<sup>4</sup>, R Yu Tsarev<sup>4</sup>, T N Yamskikh<sup>4</sup> and I N Kartsan<sup>5</sup>

<sup>1</sup> RANEPa, Kaluga branch, 4, Okruzhnaya street, Kaluga, 248016, Russia

<sup>2</sup> Financial University under the Government of the Russian Federation, Kaluga branch, 17, Chizhevsky street, Kaluga, 248016, Russia

<sup>3</sup> Bauman Moscow State Technical University, Kaluga branch, 2, Bazhenova street, Kaluga, 248000, Russia

<sup>4</sup> Siberian Federal University, 79, Svobodny pr., Krasnoyarsk, 660041, Russia

<sup>5</sup> Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy Ave., Krasnoyarsk, 660037, Russia

E-mail: [tsarev.sfu@mail.ru](mailto:tsarev.sfu@mail.ru)

**Abstract.** This paper discusses the problem of integrating modern methods of forecasting and modeling complex technical objects into the learning process. As an example, the problem of solving a system of ordinary differential equations is considered, which has significant practical application. In particular, solving a system of differential equations can be an essential part of patents. The neural network method to solve this problem by using Matlab simulation software and visual modeling tool Simulink is considered. Efficient cloud-based solution to ordinary differential equations is presented.

## 1. Introduction

Modern technologies and methods used to control complex technical objects and assess their state imply application of modern mathematical tools, which should be integrated into the curriculum of graduate degree programs. The experienced teachers point out that there is an urgent necessity for the revision of standard teaching methods. Now each course syllabus including classical mathematical courses should integrate capabilities of modern computational mathematics. This applies not only to engineering studies, but also to social and humanitarian studies in universities.

Due to the development of modern control methods, universities should provide graduates with knowledge and skills of applying neural network and cloud technologies to complex technical objects modeling by using modern computational mathematics.

In this paper, the construction of a neural network for solving systems of ordinary differential equations using Matlab simulation software and visual modeling tool Simulink is considered. The practical aspects of this problem are examined, in particular, in relation to patents. To solve the systems of ordinary differential equations, it is proposed to use the cloud service Mathematica WolframAlpha for some modeling tasks.

## 2. Solving differential equations with neural networks

Artificial neural networks (ANNs) are widely used to solve both engineering and scientific problems. Since they have proven to be a very effective tool for processing information, there are numerous attempts to expand the scope of their application or to find new principles for their construction and operation. Therefore, it is extremely important for engineering graduates to know both the theoretical foundations of building ANNs and the possibility of their practical implementation.

The fundamental processing elements of ANN are known as formal neurons. As a rule, a formal neuron receives several input signals  $x_i$  and performs a weighed sum followed by a nonlinear transformation. Connections between neurons are characterized by weights  $\omega_{ij}$ . Neurons can also receive external inputs  $X_i$  with weights  $\omega'_{ik}$ . Thus, a typical operation performed by a neuron in a network is a transformation of the form

$$y = f\left(\sum_j \omega_{ij} x_j + \sum_j \omega'_{ij} X_k\right).$$

The ANN sends  $y$  as an output signal to other neurons or devices. The function  $f$  is called the activation function. The key advantage of ANN is the ability to learn from examples and that they can be trained. ANN training refers to determining the best set of weights  $\omega_{ij}$  in order to achieve the desired result.

As an example of ANN application we consider the solution to ordinary differential equation in the neural network basis which can be of great practical importance. In particular, solving a system of differential equations can be an essential part of patents describing optimum dynamic parameters of a system [1, 2].

The neural network basis is a non-linear transformation of a weighted sum, i.e. weighted summation of several input signals with the subsequent transformation of this sum by activation function of a formal neuron, which can be both linear and nonlinear.

Given a system of  $n$  differential equations of the 1st order, which can be represented in the matrix form as follows

$$Y(x) = A Y(x), \quad (1)$$

where  $A$  – is a square constant coefficient matrix of size  $n \times n$ ;  $Y$  – is an  $n$ -dimensional vector of the sought-for function  $x$ .

For the 1st order Runge-Kutta method (The Rectangle Method), we can represent the solution to the system (1) as follows:

$$Y_{t+1} = Y_t + h A Y_t = (E + h A) Y_t = B_t Y_t, \quad (2)$$

where  $E$  – is the identity matrix of size  $n \times n$ ;  $h$  – is the integration step, which determines the accuracy of the solution. The elements of the matrix  $B$  are the first two terms of expanding the matrix exponent in a power series.

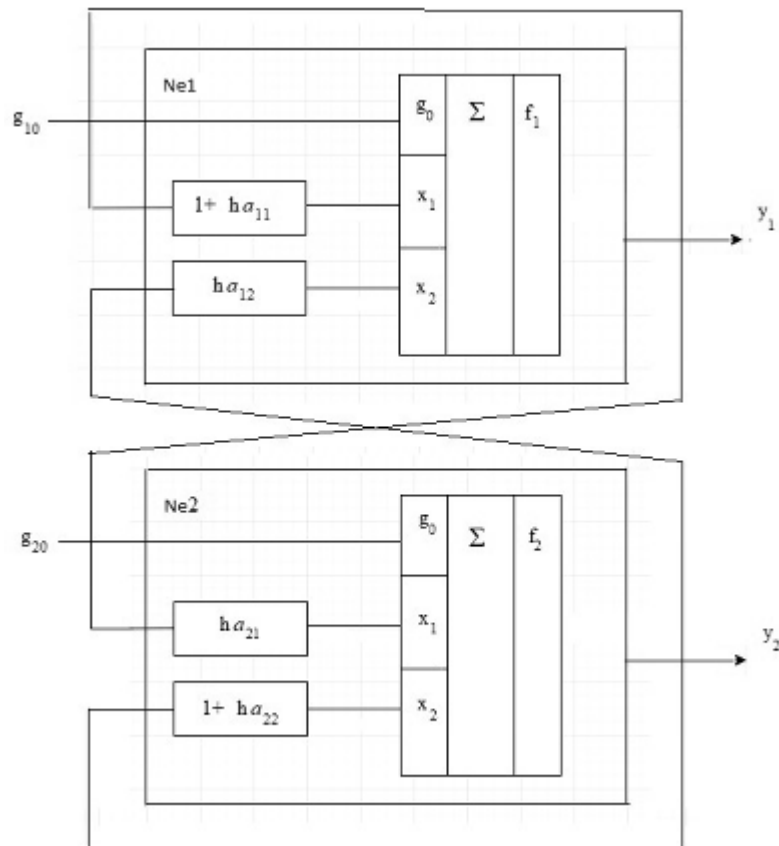
For a system of two equations, expression (2) takes the form:

$$\begin{aligned} y_{1t+1} &= b_{11} y_{1t} + b_{12} y_{2t} = (1 + h a_{11}) y_{1t} + h a_{12} y_{2t}, \\ y_{2t+1} &= b_{21} y_{1t} + b_{22} y_{2t} = h a_{21} y_{1t} + (1 + h a_{22}) y_{2t}. \end{aligned}$$

The neuron connection scheme that implements the solution to the system is shown in Figure 1. Here, Ne1 and Ne2 are the neurons involved in the integration process. Signals  $y_1$  and  $y_2$  are the network output. The input signals  $g_{10}$  and  $g_{20}$  set the initial conditions for solving the system of equations. The activation function of both neurons is a symmetric linear function.

Visual modeling systems, for example, Simulink (Matlab) are suitable to implement the schemes similar to those shown in Figure 1. The main advantage of such systems is that there is no need to write any programming code, since it is generated automatically. Moreover, after creating a workflow, you can get its C code with Simulink Coder and implement it in device controllers.

We should note that there is a free alternative to Simulink. It is an open source, cross-platform numerical computational tool Scilab.



**Figure 1.** A scheme for solving systems of ordinary differential equations in the neural network basis.

Now we consider constructing an ANN that can approximate the solution to the system of equations as an example,

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= -y_1. \end{aligned}$$

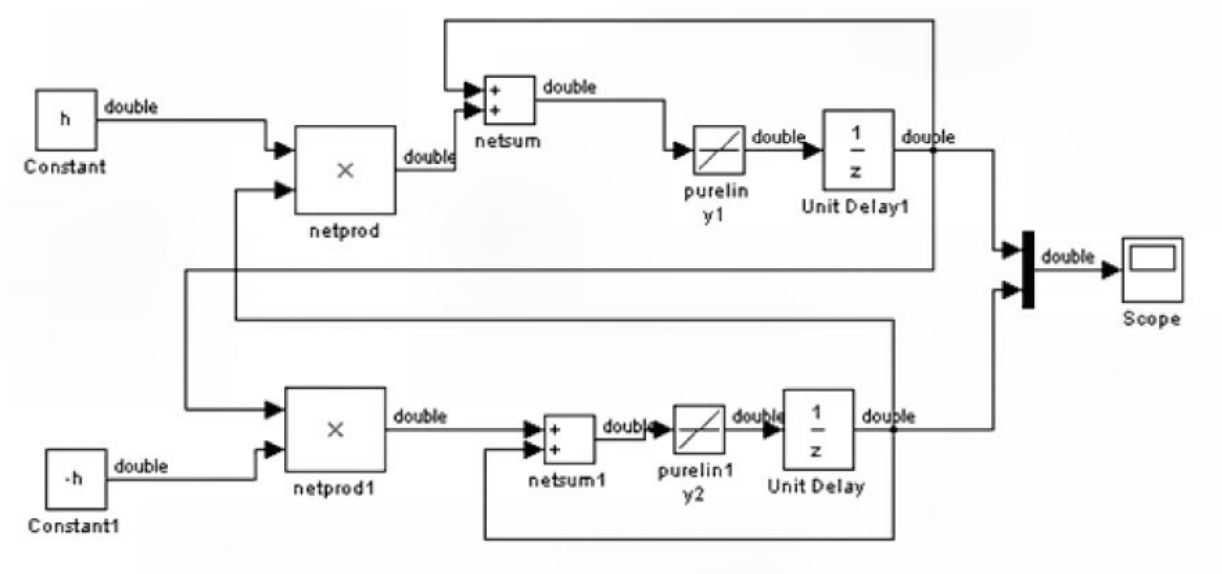
with initial conditions:  $y_1(0) = -1, y_2(0) = 0, t = 0 \dots 2\pi$ .

A block diagram of solution process in the neural network basis created in the Simulink environment is shown in Figure 2.

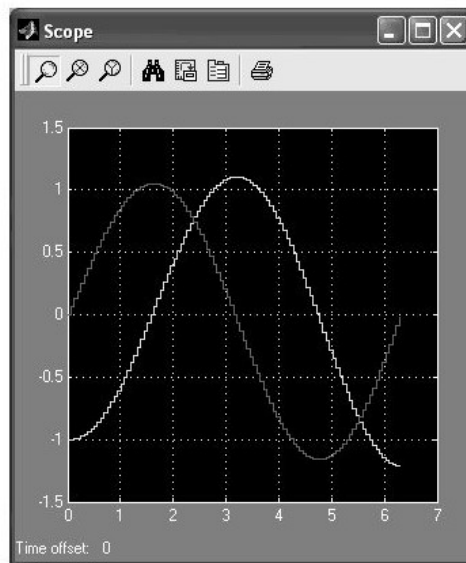
To build a scheme we use blocks from the Neural Network Blockset menu: netprod, netsum (Net Input Functions), purelin (Transfer Functions). We use the Scope block (Sinks) to visualize the results. Constant block (Sources) is used to specify the integration step. There is Unit Delay block (Discrete) available for setting the initial values. In the command window of the Unit Delay block we set the Initial value -1, and specify a Sample Time parameter  $h$ . To plot two graphs in the Scope block, the Mux block (Signals & Systems) is used.

When you open a Scope block, you see the graphs of equations (Figure 3).

An algorithm for solving differential equations based on the neural network approach was described in [3]. Calculations have demonstrated their time efficiency compared to standard algorithms.



**Figure 2.** Artificial Neural Network for solving ordinary differential equations in Simulink.



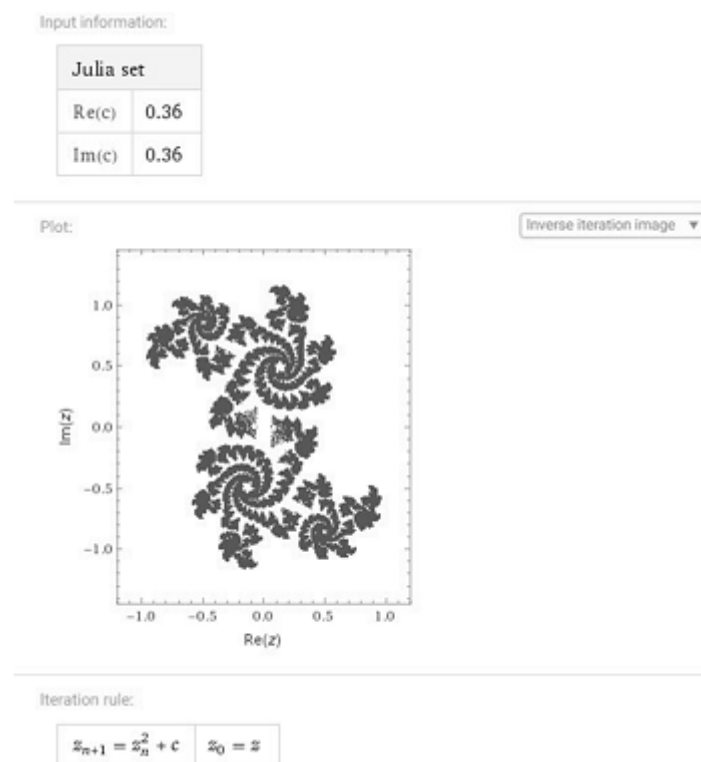
**Figure 3.** A graph on the oscilloscope screen (Scope block).

### 3. Cloud computing for solving differential equations

A modern engineer needs to be able to use the capabilities of modern computing systems, which in the last decade have presented a number of free services. Various authors reported the convenience of cloud service Wolframalpha/Mathematica [4]. The basic version of the service is free and does not require installation of additional software, which guards against illegal software use. The paid version of the system is an affordable resource which possesses enhanced features. For example, it provides the Manipulate option, which demonstrates clearly the dependence of solution on the parameters.

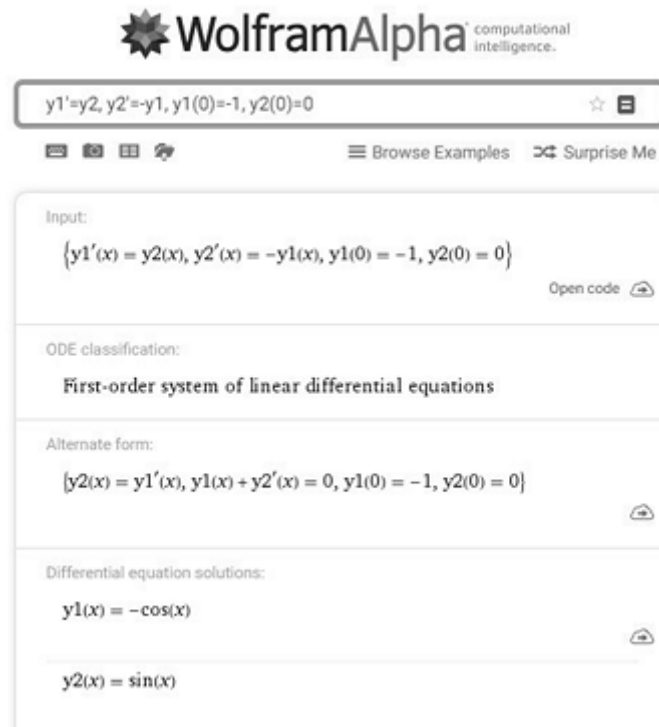
The appearance of modern computational systems made it possible to demonstrate the research results on the screen of a personal computer and thus share them with students. This is especially important for interdisciplinary and multidisciplinary research.

For example, various fractal theories are developed in the fields of engineering, natural, social and humanitarian sciences. Thus, in [5] a fractal model of dendrites with fractal parameters in the dynamic equation to simulate equilibrium phenomena in porous media was developed, in [6] a fractal model was designed to simulate evolution of trends in concentration of economic activities in the national economy, and in [7] a microscopic model for a non-Debye (nonexponential) dielectric relaxation in complex systems was constructed based on self-similarity theory. For example, the equation of the dielectric constant can be used to interpret the dielectric spectra in disordered dielectrics. Students can study the concepts related to fractal in the WolframAlpha system on the example of constructing the Julia set (Figure 4). When we type on the command line Julia set and specify the fractal parameters values in the corresponding cells, we obtain an image of quaternion filled-in Julia set and an iterative rule for its construction.



**Figure 4.** Julia set.

We also consider the solution to the system of equations from the previous section (Figure 5).



**Figure 5.** Solution to ordinary differential equation.

It is obvious that solution obtained corresponds to the graphs in Figure 3.

#### 4. Conclusion

The results of this research show that it is possible to enhance learning and teaching efficiency using modern computing systems, including cloud applications. Methods of fractal analysis seem to have very promising set of techniques useful for research activities and applied modelling in natural, social and humanitarian studies.

The approaches considered can also be used to address the issues of research, analytic and engineering practice.

#### References

- [1] Vinokurov V V, Vorob'ev A V, Zalesskii S E, Kachanov B O and Kulikov V E 2005 RusPatent no. 2264598. 2005.
- [2] Lazovskaya T V, Tarkhov D A and Vasilyev A N 2017 Parametric neural network modeling in engineering *Recent Patents on Engineering* **11** 10-15
- [3] Semenenko M G 2009 Visual modelling electrochemical processes *J. Power Sources* **191** 91-96
- [4] WolframAlpha.com
- [5] Nagiev A G, Mamedov D I and Gulieva N A 2014 Modelling of non-stationary processes of transfer of substance and adsorption in the porous environment on the basis of the fractal "dendrite" *Russ. J. of Chem. and Chem. Technology* **57 (8)** 80-84
- [6] Andronache I C, Peptenatu D, Ciobotaru A M, Gruia A K and Gropoșilă N M 2016 Using Fractal Analysis in Modeling Trends in the National Economy. *Procedia Environmental Sciences* **32** 344-351
- [7] Khamzin A A, Nigmatullin R R and Popov I I 2012 Microscopic model of a non-debye dielectric relaxation: the cole-cole law and its generalization *Theor. and Math. Phys.* **173 (2)** 314-332