INFORMATION TECHNOLOGY
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Original Theoretical Research

UDC 004.032.26
https://doi.org/10.23947/2587-8999-2024-8-2-68-79

Application of Neural Networks to Solve the Dirichlet Problem for Areas of Complex Shape

Alexander V. Galaburdin
Don State Technical University, Rostov-on-Don, Russian Federation
Galaburdin@mail.ru

Abstract

Introduction. Many mathematical problems are reduced to solving partial differential equations (PDEs) in domains of complex shapes. Existing analytical and numerical methods do not always provide efficient solutions for such problems. Recently, neural networks have been successfully applied to solve PDEs, typically addressing boundary value problems for domains with simple shapes. This paper attempts to construct a neural network capable of effectively solving boundary value problems for domains of complex shapes.

Materials and Methods. A method for constructing a neural network to solve the Dirichlet problem for regions of complex shape is proposed. Derivatives of singular solutions of the Laplace equation are accepted as activation functions. Singular points of these solutions are distributed along closed curves encompassing the boundary of the domain. The adjustment of the network weights is reduced to minimizing the root-mean-square error during training.

Results. The results of solving Dirichlet problems for various complex-shaped domains are presented. The results are provided in tables, comparing the exact solution and the solution obtained using the neural network. Figures show the domain shapes and the locations of points where the solutions were determined.

Discussion and Conclusion. The presented results indicate a good agreement between the obtained solution and the exact one. It is noted that this method can be easily applied to various boundary value problems. Methods for enhancing the efficiency of such neural networks are suggested.

Keywords: Dirichlet problem, complex-shaped domain, neural networks

For citation. Galaburdin A.V. Application of Neural Networks to Solve the Dirichlet Problem for Areas of Complex Shape. *Computational Mathematics and Information Technologies*. 2024;8(2):68–79. https://doi.org/10.23947/2587-8999-2024-8-2-68-79

Оригинальное теоретическое исследование

Применение нейронных сетей для решения задачи Дирихле для областей сложной формы

А.В. Галабурдин
Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация
Galaburdin@mail.ru

Аннотация

Введение. Многие задачи в математике сводятся к решению дифференциальных уравнений в частных производных для областей сложной формы. Не всегда существующие аналитические и численные методы позволяют эффективно получить решение подобных задач. В последнее время достаточно успешно для решения дифференциальных уравнений в частных производных применяются нейронные сети. При этом обычно рассматриваются краевые задачи для областей, имеющих простую форму. В данной работе предпринимается попытка построить нейронную сеть, способную эффективно решать краевые задачи для областей сложной формы.

Материалы и методы. Предлагается метод построения нейронной сети для решения задачи Дирихле для областей сложной формы. В качестве активационных функций принимаются производные от сингулярных решений уравнения Лапласа. Сингулярные точки этих решений распределены по замкнутым кривым, охватывающих границу области. Настройка весов сети сводится к минимизации среднеквадратической ошибки обучения.
Результаты исследования. Представлены результаты решения задач Дирихле для различных областей сложной формы. Результаты представлены в виде таблиц, содержащих точное решение и решение, полученное при помощи нейронной сети. На рисунках представлен вид областей и расположение точек, в которых определялось решение.
Обсуждение и заключения. Представленные результаты свидетельствуют о хорошем совпадении полученного решения с точным. Отмечается, что данный метод легко применим к различным краевым задачам. Указываются способы повышения эффективности подобных нейронных сетей.

Ключевые слова: задача Дирихле для области сложной формы, нейронные сети

Для цитирования. Галабурдин А.В. Применение нейронных сетей для решения задачи Дирихле для областей сложной формы. *Computational Mathematics and Information Technologies*. 2024;8(2):68–79. https://doi.org/10.23947/2587-8999-2024-8-2-68-79

Introduction. Differential equations in partial derivatives are often used in modelling various phenomena. The domains in which these differential equations are defined often have sufficiently complex shapes, making it difficult or impossible to apply known methods effectively. The rapid development of computer technology has allowed for the use of various machine learning methods in solving PDEs.

Recently, the neural network method, whose theoretical foundations were laid in the mid-20th century by A.N. Kolmogorov, has been increasingly used to solve such problems. These methods typically use well-studied differential equations that are relatively simple to solve. Many developers apply boundary value problems for the Laplace equation for this purpose.

For example, the work [2] assesses the quality of approximate solutions to the Laplace equation constructed using neural networks. In [3], a neural network is used to solve the problem of membrane deflection. The article [4] discusses the numerical solution of the Poisson equation in a two-dimensional domain using the Galerkin method and the Ritz method with deep neural networks. Various approaches to training radial-basis neural networks for solving the Poisson equation are discussed in [5].

The study [6] proposes a network architecture that allows solving Laplace, Poisson, heat conduction, and wave equations for rectangular domains. Methods for solving PDEs using radial-basis neural networks, feedforward networks, and modified neural networks are considered in [7]. Using a perceptron-type neural network with a single hidden layer, [8] obtains an analytical approximation of solutions for parabolic-type PDEs.

The use of radial-basis functions in implementing the finite element method with neural networks is explored in [9]. Studies [10, 11] vary the parameters of radial-basis functions when training radial-basis neural networks.

The method of physics-informed neural networks is currently gaining popularity for solving PDEs [12]. The study [13] describes algorithms for using physics-informed neural networks to solve classical mechanics problems.

Artificial neural networks were used to solve the Navier-Stokes equations in [14]. The article [15] investigates approaches to solving heat and mass transfer problems based on a perceptron-type neural network.

The examples above illustrate a wide range of problems solved using neural networks and the various approaches to applying neural networks to solve different boundary value problems. Neural networks are more commonly applied to solving boundary value problems for domains of simple shapes. This study aims to propose an approach for using neural networks to solve boundary value problems for complex-shaped domains.

Materials and Methods. Consider the Dirichlet problem for a plane region G, bounded by a smooth closed curve γ. One effective method for solving this problem is the boundary integral equation method. To obtain the corresponding boundary integral equation, Green's formula can be used:

$$u = \frac{1}{2\pi}\int_{\gamma}\frac{\partial u}{\partial n}U\, d\gamma - \frac{1}{2\pi}\int_{\gamma}\frac{\partial u}{\partial n}u\, d\gamma.$$

Here, U is the singular solution of the Laplace equation.

Using a quadrature formula for calculating integrals, this relationship can be repre-sented as:

$$u_i = \frac{1}{2\pi}\sum_{k=1}^{N} C_k \left[\frac{\partial u}{\partial n}\right]_k [U]_{ik} - \frac{1}{2\pi}\sum_{k=1}^{N} C_k [u]_k \left[\frac{\partial U}{\partial n}\right]_{ik}, \quad (1)$$

where u_i is the value of u at the i-th point of the boundary γ, C_k are the coefficients of the quadrature formula.

In this expression, $[U]_{ik}$ and $\left[\frac{\partial U}{\partial n}\right]_{ik}$ can be considered as activation functions, while $C_k\left[\frac{\partial u}{\partial n}\right]_k$ and $C_k[u]_k$ can be considered as weights.

By requiring the fulfillment of the relationship in each point of the boundary for all functions of the training set, a system of equations for determining the weights can be obtained using the least squares method. However, these systems of equations are ill-conditioned. To improve the conditioning of these systems, the singularity of $[U]_{ik}$ and $\left[\frac{\partial U}{\partial n}\right]_{ik}$, can be increased by shifting the integration contour some distance away from the boundary γ.

The Dirichlet problem solution can then be sought in the form:

$$u(x)=\sum_{k=1}^{N} w_k f(s_k) U(x,\sigma_k)+\sum_{k=1}^{N} v_k f(s_k) V(x,\tau_k),$$

where $f(s_k)$ is the value of the unknown function u on the boundary; $U(x, \sigma_k)$ и V (x, τ_k) are activation functions; σ_k and τ_k are points on closed curves γ_1 and γ_2, encompassing the boundary γ; x is a point in the domain G.

The curves γ_1 and γ_2 are similar to the contour γ and are obtained by shifting each point in the direction of the outward normal to the boundary by distances ε_1 and ε_2 respectively.

During network training, weights are adjusted and the values ε_1 and ε_2, are determined by minimizing the error functional:

$$J(w_k, v_k, \varepsilon_1, \varepsilon_2)=\sum_{j=1}^{M}\sum_{i=1}^{N}\left\{\sum_{k=1}^{N} w_k f_k^j U(x_i,\sigma_k)+v_k f_k^j V(x_i,\tau_k)-f_i^j\right\}^2,$$

where x_i is the coordinate of the i-th point of the boundary contour γ; f_i^j is the boundary value of the j-th function in the training set at point x_i.

From the relations $\frac{\partial J}{\partial w_m}=0$ и $\frac{\partial J}{\partial v_m}=0$, m = 1,2,…Na system of linear equations for determining w_m and v_m can be obtained.

The values ε_1 and ε_2 are determined by simple iteration. Assuming $\varepsilon_2 = \varepsilon_1+1$, the values of $\varepsilon_1 = a + hj$, j =1,2,…L are chosen. The value of ε_1, that provides the best result is selected. After that, all neural network parameters are determined and its configuration is completed.

The accuracy of the obtained solution can be assessed by comparing the values of u on the boundary calculated using the neural network:

$$\tilde{u}(s_i)=\sum_{k=1}^{N} w_k f(s_k) U(s_i,\sigma_k)+\sum_{k=1}^{N} v_k f(s_k) V(s_i,\tau_k)$$

with the given boundary conditions $f(s)$.

The defined network parameters do not always ensure the desired accuracy of the neural network solution. In this case, the required accuracy can be achieved by iterative refinement of the obtained result:

$$\Delta u^{\circ}(s_i)=f(s_i),\quad u_t^{\circ}(s_i)=f(s_i),$$

$$\Delta v^{n+1}(s_i)=\sum_{k=1}^{N} w_k \Delta u^n(s_k) U(s_i,\sigma_k)+\sum_{k=1}^{N} v_k \Delta u^n(s_k) V(s_i,\tau_k),$$

$$\Delta u^{n+1}(s_i)=\Delta u^{n+1}(s_i)-\Delta v^{n+1}(s_i),\quad u_t^{n+1}(s_i)=u_t^{n+1}(s_i)+\Delta u^{n+1}(s_i),$$

$u_t^{n+1}(s_i)$ are the values of the refined solution at the boundary of the region.

The refinement process continues until the specified accuracy is achieved, i. e. $\frac{\|\Delta u^{n+1}(s_i)\|}{\|u_t^{n+1}(s_i)\|}<\delta$ or the value $\frac{\|\Delta u^{n+1}(s_i)\|}{\|u_t^{n+1}(s_i)\|}$. starts to increase.

After this, the value of the solution at any point x in the domain G can be computed using the formula:

$$\tilde{u}(x)=\sum_{k=1}^{N} w_k u_k u_t(s_k) U(x,\sigma_k)+\sum_{k=1}^{N} v_k u_t(s_k) V(x,\tau_k),$$

where $u_t(s_k)$ are the refined values of the unknown function on the boundary γ.

The training set used functions that are solutions to the Laplace equation:

$$r^k \cos\left(\text{karccos}\left(\frac{x}{r}\right)\right) + r^k \sin\left(\text{karccos}\left(\frac{x}{r}\right)\right), r = \sqrt{x^2 + y^2},$$

where $k = 0, 1, 2, 3, \ldots M$.

These functions were specified in different coordinate systems, each rotated relative to the others by an angle that is a multiple of $2\pi/5$.

Results. The presented method was utilized to solve the Dirichlet problem for regions whose boundary γ was defined as:

$$\begin{cases} x = a\cos(t) + g\cos(\propto t), \\ y = b\sin(t) + q\sin(\propto t), \end{cases} \quad t \in [0, 2\pi],$$

where a, b, g, q, $\propto$ are variable parameters.

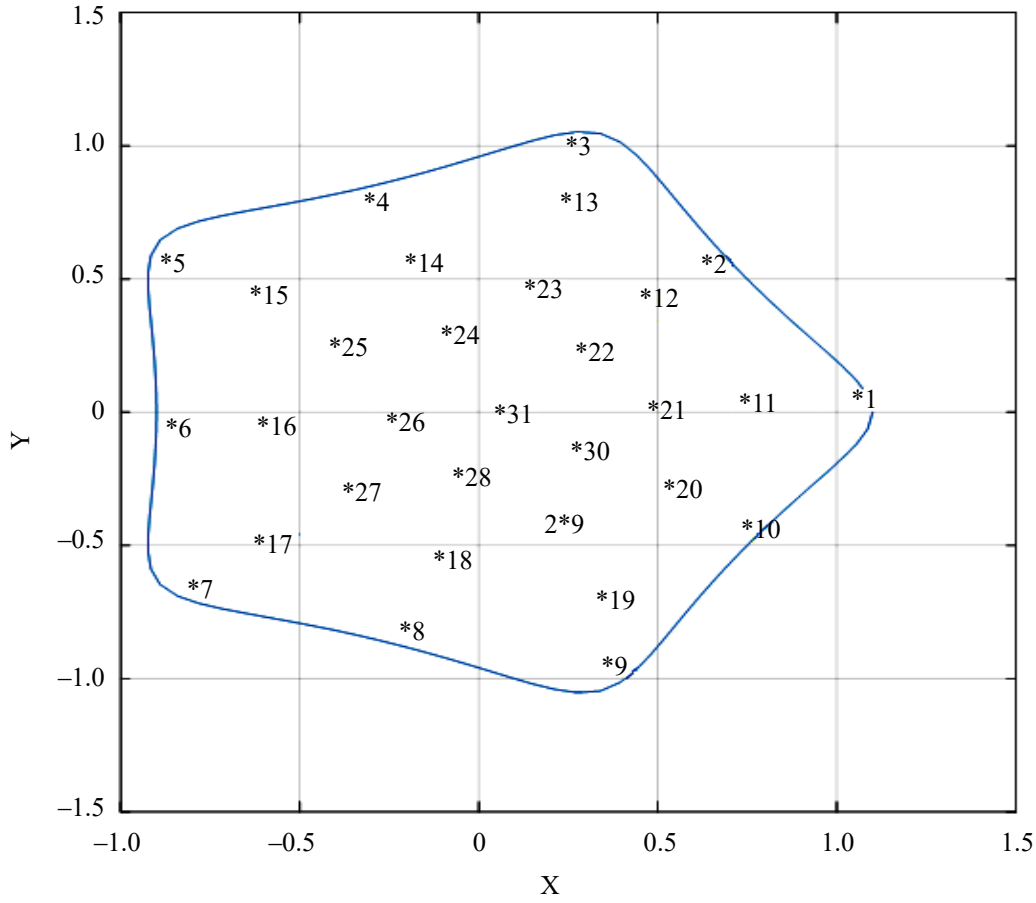

Fig. 1. Region $G1$

Figure 1 shows the region whose boundary corresponds to $a = 1$, $b = 1$, $g = 0.1$, $q = –0.1$, $\propto = 4$. The numbered stars indicate the locations of points in region $G1$ where the exact Dirichlet problem solutions and the values obtained using the neural network with $\varepsilon_1 = 5$ are calculated.

Table 1 presents the calculation results corresponding to the solution

$$u = e^{2.45x} \cos 2.45y. \tag{2}$$

The table includes the point numbers in region $G1$, their coordinates, the exact solution of the Dirichlet problem, and the solution obtained by the neural network.

Table 2 presents the calculation results corresponding to the solution

$$u = \frac{x^3 + xy^2 + x^2 - y^2 + 5x + 5}{(x+1)^2 + y^2}. \tag{3}$$

in region $G1$.

Figure 2 shows the region corresponding to $a = 1$, $b = 1$, $g = 0.1$, $q = 0.1$, $\propto = 5$. Tables 3 and 4 present the calculation results corresponding to solutions (2) and (3) in region $G2$ for $\varepsilon_1 = 6.45$. Figure 3 shows region $G3$, corresponding to $a = 1$, $b = 1$, $g = 0.2$, $q = –0.2$, $\propto = 2$. The calculation results corresponding to solutions (2) and (3) in region $G3$ for (3) $\varepsilon_1 = 6.3$, are presented in Tables 5 and 6.

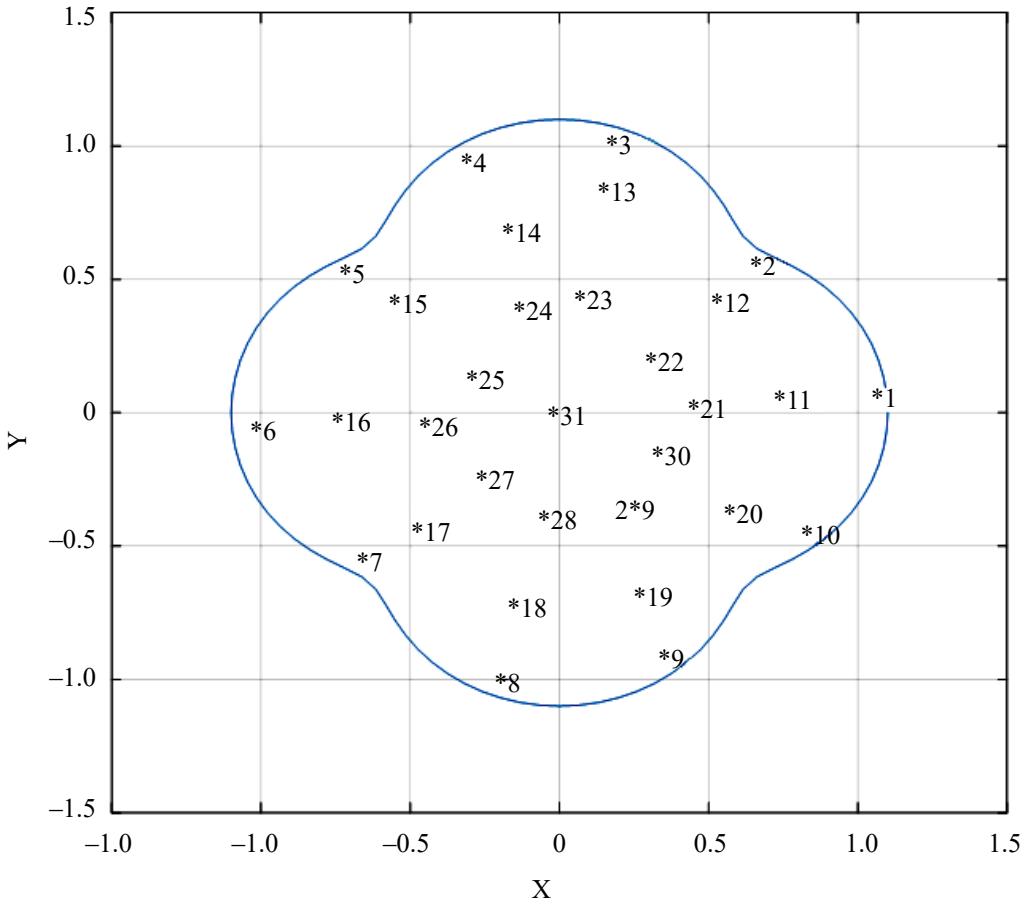

Fig. 2. Region *G*2

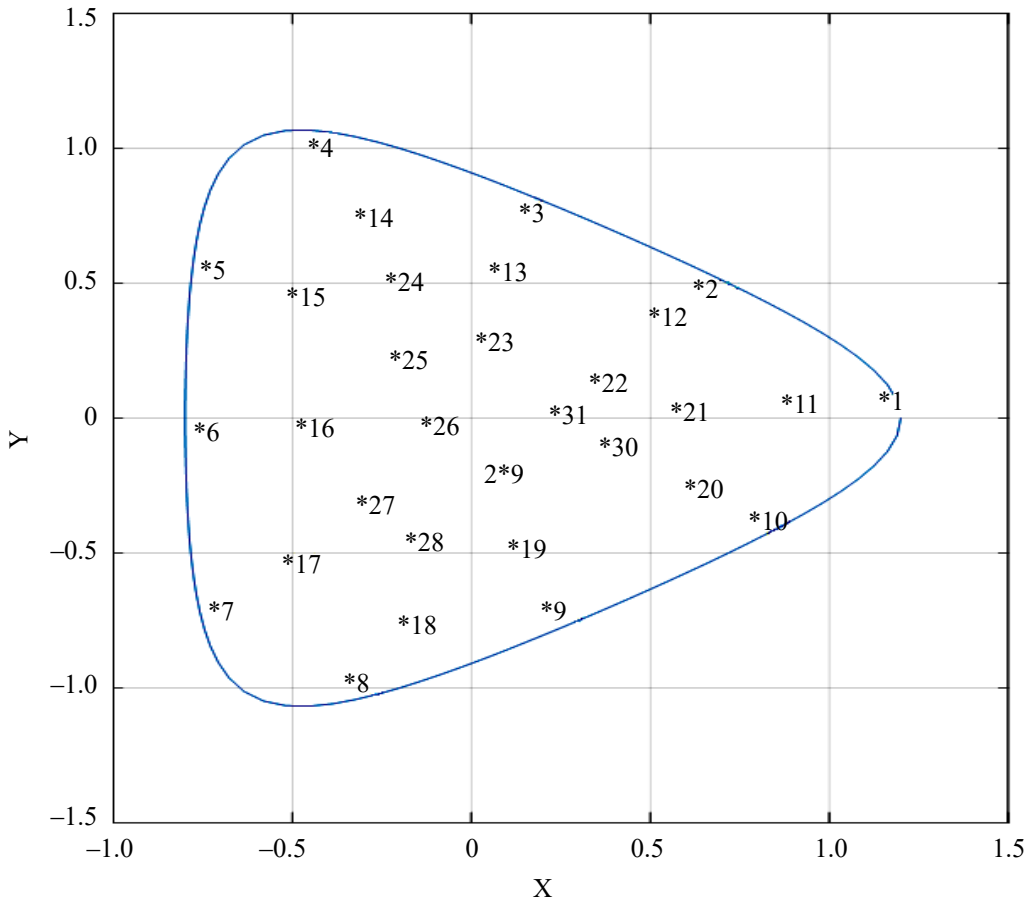

Fig. 3. Region *G*3

Table 1

The calculation results

Point number	1	2	3	4	5	6	7
x	1.0351	0.6510	0.2626	–0.3216	–0.8728	–0.8497	–0.8020
y	0.0602	0.5496	1.0030	0.7890	0.5597	–0.0620	–0.6571
The exact solution	12.4909	1.0954	–1.4745	–0.1611	0.0234	0.1233	–0.0055
The HC solution	12.477	1.1020	–1.4730	–0.1630	0.0240	0.1240	–0.0090
Point number	8	9	10	11	12	13	14
x	–0.2036	0.3771	0.7239	0.7804	0.4546	0.1996	–0.2272
y	–0.8273	–0.9658	–0.4493	0.0437	0.3866	0.7557	0.5518
The exact solution	0.2675	–1.7992	2.6689	6.7272	1.7790	–0.4516	0.1244
The HC solution	0.2736	–1.7670	2.6512	6.7282	1.7803	–0.4501	0.1245
Point number	15	16	17	18	19	20	21
x	–0.6570	–0.5950	–0.6056	–0.1406	0.2827	0.5081	0.4747
y	0.4233	–0.0455	–0.4941	–0.5800	–0.7287	–0.3129	0.0239
The exact solution	0.1017	0.2313	0.0800	0.1058	–0.4254	2.5011	3.1943
The HC solution	0.1023	0.2312	0.0817	0.1020	–0.4318	2.4936	3.1933
Point number	22	23	24	25	26	27	28
x	0.2190	0.1240	–0.1139	–0.3981	–0.2894	–0.3700	–0.0650
y	0.1909	0.4589	0.2673	0.2597	–0.0257	–0.2984	–0.2831
The exact solution	1.5265	0.5851	0.6001	0.3033	0.4912	0.3007	0.6557
The HC solution	1.5259	0.5857	0.5997	0.3032	0.4903	0.3003	0.6527
Point number	29	30	31				
x	0.1694	0.2492	0.0744				
y	–0.4441	–0.1493	0.0000				
The exact solution	0.7030	1.7196	1.1998				
The HC solution	0.6961	1.7160	1.1980				

Table 2

The calculation results

Point number	1	2	3	4	5	6	7
x	1.0351	0.6510	0.2626	–0.3216	–0.8728	–0.8497	–0.8020
y	0.0602	0.5496	1.0030	0.7890	0.5597	–0.0620	–0.6571
The exact solution	3.9985	3.8121	3.1740	4.1404	7.0650	9.9034	5.9459
The HC solution	4.0010	3.8210	3.1740	4.1330	7.0860	9.8590	5.9370
Point number	8	9	10	11	12	13	14
x	–0.2036	0.3771	0.7239	0.7804	0.4546	0.1996	–0.2272
y	–0.8273	–0.9658	–0.4493	0.0437	0.3866	0.7557	0.5518
The exact solution	3.8595	3.2536	3.8810	4.0331	4.0649	3.7186	4.8021
The HC solution	3.8675	3.2293	3.8793	4.0297	4.0655	3.7168	4.7974
Point number	15	16	17	18	19	20	21
x	–0.6570	–0.5950	–0.6056	–0.1406	0.2827	0.5081	0.4747
y	0.4233	–0.0455	–0.4941	–0.5800	–0.7287	–0.3129	0.0239
The exact solution	6.8777	7.6080	6.2579	4.5410	3.7216	4.0829	4.2223
The HC solution	6.8790	7.5937	6.2519	4.5407	3.7254	4.0835	4.2207
Point number	22	23	24	25	26	27	28

Continuation of table 2

x	0.2190	0.1240	–0.1139	–0.3981	–0.2894	–0.3700	–0.0650
y	0.1909	0.4589	0.2673	0.2597	–0.0257	–0.2984	–0.2831
The exact solution	4.4902	4.3447	5.1277	6.1527	5.9413	5.9450	4.9773
The HC solution	4.4883	4.3425	5.1237	6.1467	5.9351	5.9387	4.9750
Point number	29	30	31				
x	0.1694	0.2492	0.0744				
y	–0.4441	–0.1493	0.0000				
The exact solution	4.3023	4.4661	4.8261				
The HC solution	4.3041	4.4653	4.8236				

Table 3

The calculation results

Point number	1	2	3	4	5	6	7
x	1.0403	0.6832	0.2463	–0.3558	–0.7924	–1.0403	–0.6832
y	0.0633	0.5522	0.9914	0.9278	0.4881	–0.0633	–0.5522
The exact solution	12.6374	1.1533	–1.3835	–0.2702	0.0526	0.0772	0.0406
The HC solution	12.5300	1.1510	–1.3230	–0.2480	0.0360	0.1000	0.0290
Point number	8	9	10	11	12	13	14
x	–0.2463	0.3558	0.7924	0.7591	0.4985	0.1798	–0.2596
y	–0.9914	–0.9278	–0.4881	0.0462	0.4029	0.7235	0.6771
The exact solution	–0.4138	–1.5445	2.5519	6.3819	1.8691	–0.3111	–0.0465
The HC solution	–0.4635	–1.4973	2.5558	6.3787	1.8724	–0.3007	–0.0338
Point number	15	16	17	18	19	20	21
x	–0.5782	–0.7591	–0.4985	–0.1798	0.2596	0.5782	0.4217
y	0.3562	–0.0462	–0.4029	–0.7235	–0.6771	–0.3562	0.0257
The exact solution	0.1559	0.1547	0.1625	–0.1290	–0.1660	2.6505	2.8047
The HC solution	0.1554	0.1509	0.1558	–0.1404	–0.1775	2.6392	2.8042
Point number	22	23	24	25	26	27	28
x	0.2770	0.0999	–0.1442	–0.3212	–0.4217	–0.2770	–0.0999
y	0.2238	0.4019	0.3761	0.1979	–0.0257	–0.2238	–0.4019
The exact solution	1.6820	0.7064	0.4246	0.4027	0.3551	0.4329	0.4331
The HC solution	1.6832	0.7097	0.4288	0.4031	0.3514	0.4279	0.4261
Point number	29	30	31				
x	0.1442	0.3212	–0.0282				
y	–0.3761	–0.1979	0.0000				
The exact solution	0.8608	1.9437	0.9332				
The HC solution	0.8512	1.9369	0.9308				

Fig. 4 and Fig. 5 show graphically obtained results of solving the Dirichlet problem in *G*3 for solution (3).

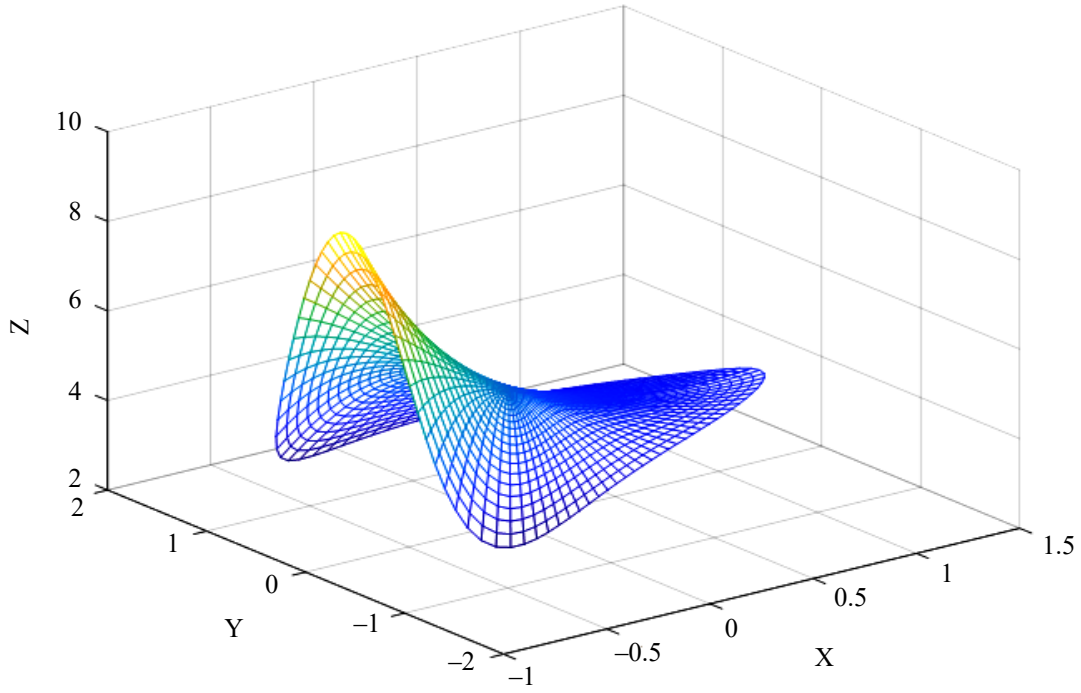

Fig. 4. HC solution in *G*3 corresponding to (3)

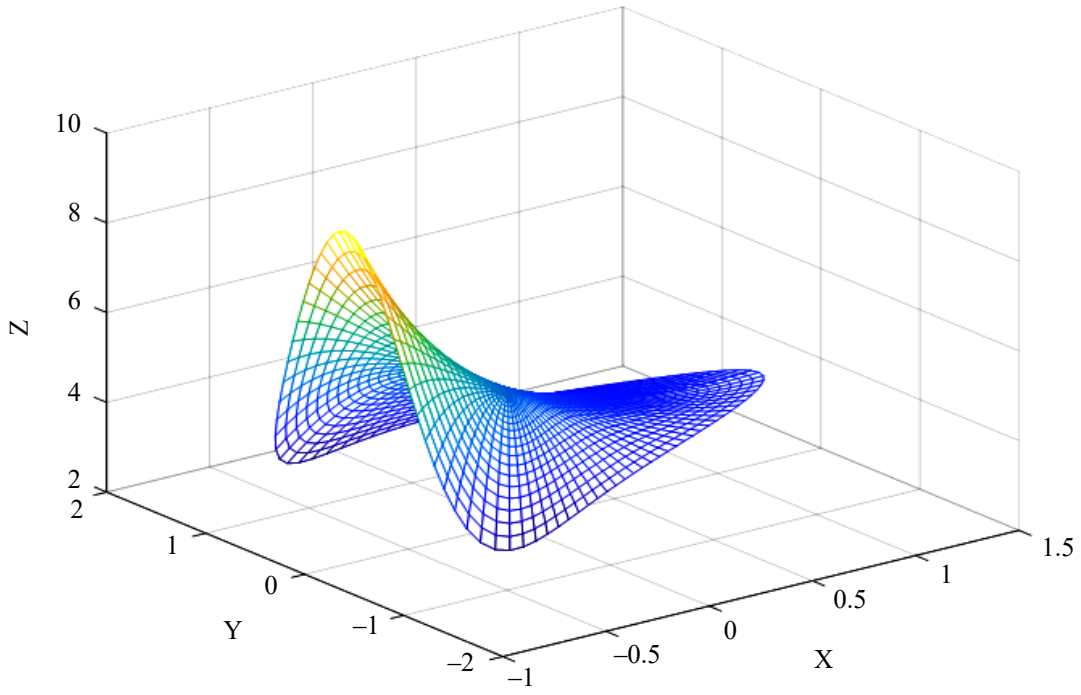

Fig. 5. The exact solution in *G*3 corresponding to (3)

Table 4

The calculation results

Point number	1	2	3	4	5	6	7
x	1.0403	0.6832	0.2463	–0.3558	–0.7924	–1.0403	–0.6832
y	0.0633	0.5522	0.9914	0.9278	0.4881	–0.0633	–0.5522
The exact solution	3.9985	3.8023	3.1994	3.5918	7.2276	12.5939	6.2589
The HC solution	3.9990	3.7940	3.1690	3.6220	7.2120	12.488	6.2450

Continuation of table 4

Point number	8	9	10	11	12	13	14
x	–0.2463	0.3558	0.7924	0.7591	0.4985	0.1798	–0.2596
y	–0.9914	–0.9278	–0.4881	0.0462	0.4029	0.7235	0.6771
The exact solution	3.2856	3.3263	3.8398	4.0402	4.0187	3.7979	4.4638
The HC solution	3.3086	3.3405	3.8291	4.0326	4.0107	3.7853	4.4515
Point number	15	16	17	18	19	20	21
x	–0.5782	–0.7591	–0.4985	–0.1798	0.2596	0.5782	0.4217
y	0.3562	–0.0462	–0.4029	–0.7235	–0.6771	–0.3562	0.0257
The exact solution	6.7562	8.9677	6.1858	4.1803	3.8289	4.0072	4.2753
The HC solution	6.7422	8.9594	6.1844	4.1866	3.8342	4.0038	4.2690
Point number	22	23	24	25	26	27	28
x	0.2770	0.0999	–0.1442	–0.3212	–0.4217	–0.2770	–0.0999
y	0.2238	0.4019	0.3761	0.1979	–0.0257	–0.2238	–0.4019
The exact solution	4.3835	4.4688	5.0283	5.9328	6.5585	5.7193	4.8676
The HC solution	4.3760	4.4594	5.0170	5.9234	6.5537	5.7168	4.8675
Point number	29	30	31				
x	0.1442	0.3212	–0.0282				
y	–0.3761	–0.1979	0.0000				
The exact solution	4.4325	4.3410	5.0723				
The HC solution	4.4321	4.3377	5.0668				

Table 5

The calculation results

Point number	1	2	3	4	5	6	7
x	1.1387	0.6789	0.1404	–0.4339	–0.7355	–0.7488	–0.7213
y	0.0610	0.4697	0.7826	1.0207	0.5615	–0.0620	–0.6858
The exact solution	16.0958	2.1517	–0.4790	–0.2769	0.0320	0.1579	–0.0187
The HC solution	16.055	2.1310	–0.4540	–0.2700	0.0420	0.1780	–0.0090
Point number	8	9	10	11	12	13	14
x	–0.3119	0.2489	0.7847	0.8836	0.4897	0.0792	–0.3416
y	–0.9953	–0.7209	–0.4028	0.0444	0.3136	0.5423	0.7854
The exact solution	–0.3552	–0.3575	3.7699	8.6620	2.3870	0.2911	–0.1498
The HC solution	–0.3695	–0.3003	3.7043	8.6771	2.3820	0.3008	–0.1382
Point number	15	16	17	18	19	20	21
x	–0.5300	–0.4937	–0.5321	–0.2508	0.1566	0.5793	0.5776
y	0.4312	–0.0454	–0.5297	–0.7550	–0.4856	–0.2725	0.0244
The exact solution	0.1343	0.2965	0.0732	–0.1490	0.5459	3.2463	4.109099
The HC solution	0.1459	0.3115	0.0881	–0.1402	0.5636	3.2274	4.1094
Point number	22	23	24	25	26	27	28
x	0.2626	0.0059	–0.2309	–0.2835	–0.1876	–0.3050	–0.1775
y	0.1262	0.2540	0.5029	0.2748	–0.0254	–0.3423	–0.4667
The exact solution	1.8128	0.8243	0.1887	0.3903	0.6302	0.3166	0.2683
The HC solution	1.8150	0.8318	0.2004	0.4015	0.6414	0.3311	0.2834
Point number	29	30	31				
x	0.0459	0.3328	0.1744				
y	–0.2032	–0.1161	0.0000				

Continuation of table 5

The exact solution	0.9832	2.1689	1.5329				
The HC solution	0.9924	2.1689	1.5373				

Table 6

The calculation results

Point number	1	2	3	4	5	6	7
x	1.1387	0.6789	0.1404	–0.4339	–0.7355	–0.7488	–0.7213
y	0.0610	0.4697	0.7826	1.0207	0.5615	–0.0620	–0.6858
The exact solution	4.0101	3.8766	3.6988	3.1990	6.4195	8.8530	5.4894
The HC solution	4.0110	3.8840	3.6990	3.2160	6.4370	8.8080	5.4920
Point number	8	9	10	11	12	13	14
x	–0.3119	0.2489	0.7847	0.8836	0.4897	0.0792	–0.3416
y	–0.9953	–0.7209	–0.4028	0.0444	0.3136	0.5423	0.7854
The exact solution	3.2886	3.7558	3.9008	4.0079	4.0963	4.2640	4.1824
The HC solution	3.2830	3.7587	3.8986	4.0062	4.0998	4.2648	4.1807
Point number	15	16	17	18	19	20	21
x	–0.5300	–0.4937	–0.5321	–0.2508	0.1566	0.5793	0.5776
y	0.4312	–0.0454	–0.5297	–0.7550	–0.4856	–0.2725	0.0244
The exact solution	6.2152	6.9516	5.7855	4.1727	4.2569	4.0602	4.1378
The HC solution	6.2154	6.9378	5.7897	4.1682	4.2566	4.0600	4.1380
Point number	22	23	24	25	26	27	28
x	0.2626	0.0059	–0.2309	–0.2835	–0.1876	–0.3050	–0.1775
y	0.1262	0.2540	0.5029	0.2748	–0.0254	–0.3423	–0.4667
The exact solution	4.4559	4.8437	4.9521	5.6575	5.5545	5.5934	4.9136
The HC solution	4.4563	4.8424	4.9498	5.6532	5.5499	5.5904	4.9112
Point number	29	30	31				
x	0.0459	0.3328	0.1744				
y	–0.2032	–0.1161	0.0000				
The exact solution	4.8040	4.3626	4.6270				
The HC solution	4.8022	4.3624	4.6263				

In all cases, when clarifying the decision, $M = 75$, $\delta = 0.00025$ were taken and the Euclidean norm was used.The following activation functions were taken

$$U(x,y,t,s)=\frac{\partial^6}{\partial^3 t\partial^3 s}Y,\quad V(x,y,t,s)=\frac{\partial^5}{\partial^3 t\partial^2 s}Y-\frac{\partial^5}{\partial^2 t\partial^3 s}Y,$$

$$Y=\ln\frac{1}{R},\quad R=\sqrt{(x-t)^2+(y-s)^2}.$$

Discussion and Conclusion. he presented results convincingly demonstrate that the proposed method for constructing a neural network to solve the Dirichlet problem for regions of complex shapes is highly effective. This method can also be utilized for solving other partial differential equations. It can be easily adapted for solving three-dimensional problems and boundary value problems for multiply connected regions. Its efficiency can be further enhanced by appropriately selecting activation functions (by choosing parameters ε_1 and ε_2), by optimizing the training set selection, and by fine-tuning the weights. All of the above indicates the considerable potential of the proposed method.

References

1. Kolmogorov A.N. On the Representation of Continuous Functions of Several Variables by Superpositions of Continuous Functions of One Variable and Addition. *Doklady Akademii Nauk SSSR*. 1957; 114(5): 953–956. (in Russ.).

2. Varshavchik E.A., Galyautdinova A.R., Sedova Y. S., Tarkhov D.A. Solving Partial Differential Equations for Regions with Constant Boundaries. Artificial Intelligence in Solving Current Social and Economic Problems of the 21st Century. In: *Proceedings of the Third All-Russian Scientific-Practical Conference*. Perm: Publishing House of Perm State National Research University; 2018. pp. 294–303. (in Russ.).

3. Bortkovskaya M.R., Kaverzneva T.T., Semenova D.A., Shishkina I.A., Tarkhov D.A., Udalov P.P. Construction of a Mathematical Model of Membrane Deflection Using the Two-Layer Euler Method Based on a Differential Equation and Experimental Data. Artificial Intelligence in Solving Current Social and Economic Problems of the 21st Century. In: *Proceedings of the Third All-Russian Scientific-Practical Conference*. Perm: Publishing House of Perm State National Research University; 2018. pp. 194–201. (in Russ.).

4. Epifanov A.A. Application of Deep Learning Methods for Solving Partial Differential Equations. *Advances in Cybernetics*. 2020;1(4): 22–28. (In Russ.). https://doi.org/10.51790/2712-9942-2020-1-4-3

5. Gorbatchenko V.I., Artyukhina E.V. Two Approaches to Training Radial Basis Neural Networks for Solving Partial Differential Equations. *Izvestiya Vuzov. Povolzhskiy Region. Technical Sciences. Informatics and Computer Technology.* 2007;2:56–66. (in Russ.).

6. Korsunov N.I., Lomakin A.V. Modelling Processes Described by the Wave Differential Equation Using Cellular Neural Networks. *Scientific Bulletins. Series History. Political Science. Economics. Informatics*. 2014;15(186):103–107. (in Russ.).

7. Kovalenko A.N., Chernomorets A.A., Petina M.A. On the Application of Neural Networks for Solving Partial Differential Equations. *Scientific Bulletins. Series Economics. Informatics*. 2017;258:103–110. (in Russ.).

8. Vershinin V.E., Ponomarev R.Y. Application of Neural Network Modelling Methods for Solving Initial-Boundary Value Problems for Partial Differential Equations. *Bulletin of Tyumen State University. Physical-Mathematical Modelling. Oil, Gas, Energy.* 2017;9(35):132–147. (in Russ.). https://doi.org/10.21684/2411-7978-2023-9-3-132-147

9. Zemskova Y.N. Applicability of Compactly Supported Neural Networks for Solving Partial Differential Equations Using the Finite Element Method. *Izvestiya PGPU im. V.G. Belinskogo*. 2009;13(17):144–148. (in Russ.).

10. Kansa E.J. Motivation for Using Radial Basis Functions to Solve PDEs. URL: http://uahtitan.uah.edu/kansaweb.html (accessed: January 16, 1999).

11. Kansa E.J. Multiquadrics. A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics. II. Solutions to Parabolic, Hyperbolic, and Elliptic Partial Differential Equations. *Comput. Math. Appl.* 1990; 19(8/9):147–161.

12. Raissi M., Perdikaris P., Karniadakis G.E. Physics-Informed Neural Net-works: A Deep Learning Framework for Solving Forward and Inverse Problems In-volving Nonlinear Partial Differential Equations. *Journal of Computational Physics*. 2019;378:686–707.

13. Zrelova D.P., Ulyanov S.V. Models of Physically Informed Classical La-grangian Hamiltonian Neural Networks in Deep Learning. *Modern Information Technologies and IT Education*. 2022;18(2):310–325. (in Russ.). https://doi.org/10.25559/SITITO.18.202202.310-325

14. Chen J., Viquerat J., Hachem E. U-net Architectures for Fast Prediction of Incompressible Laminar Flows. URL: https://arxiv.org/pdf/1999.13532.pdf (accessed: May 17, 1999).

15. Cai S., Wang Z., Wang S., Perdikaris P., Karniadakis G.E. Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*. 2021;143(6):060801. https://doi.org/10.1115/1.4050542

Received 26.04.2024
Revised 14.05.2024
Accepted 15.05.2024

About the Author:
Alexander V. Galaburdin, associate professor of the department Mathematics and informatics, Don State Technical University (1, Gagarin Sq., Rostovon-Don, 344003, RF), Cand.Sci. (Phys. – math.), associate professor, ORCID, Galaburdin@mail.ru

Conflict of interest statement
The author does not have any conflict of interest.

The author has read and approved the final manuscript.

Поступила в редакцию 26.04.2024
Поступила после рецензирования 14.05.2024
Принята к публикации 15.05.2024

Об авторе:
Александр Васильевич Галабурдин, кандидат физико-математических наук, доцент кафедры математики и информатики Донского государственного технического университета (РФ, 344003, г. Ростов-на-Дону, пл. Гагарина, 1), ORCID, Galaburdin@mail.ru

Конфликт интересов
Автор заявляет об отсутствии конфликта интересов

Автор прочитал и одобрил окончательный вариант рукописи.