Solving Direct and Inverse Boundary Value Problems for Piecewise Homogeneous Media on Radial Basis Functions Networks

V. I. Gorbachenko, D. A. Stenkin

Federal State Budgetary Educational Institution of Higher Education "Penza State University", Penza, Russia

Abstract. The application of physics-informed radial basis function networks for solving boundary value problems describing piecewise homogeneous media is considered. A meshless algorithm for solving boundary value problems for piecewise homogeneous media is proposed, using the solution of individual problems for each region with different properties of the medium, and the conditions for the conjugation of media. To solve the coefficient inverse problem of determining the properties of a piecewise inhomogeneous medium, a parametric optimization algorithm is proposed that uses separate networks to approximate the properties of the medium and solve the direct problem. To train networks, a fast algorithm developed by the authors based on the Levenberg – Marquardt method was applied. The work of the proposed algorithms is demonstrated on model problems.

Keywords: partial differential equations, piecewise homogeneous medium, inverse problems, physics informed neural-networks, radial basis function networks, neural network learning, Levenberg-Marquardt method.

DOI 10.14357/20718632210409

Introduction

The solving of boundary value problems for partial differential equations on neural networks has been known for a long time (see, for example, [1–2]). This approach is based on a number of theorems, first of all, on the universal approximation theorem (the Cybenko G. theorem) [3]. The theorem states that a feedforward network with one hidden layer, containing neurons with sigmoidal activation functions in the hidden layer and a linear activation function in the output layer, allows an arbitrary continuous function to be approximated with any accuracy. It only requires a sufficiently large number of neurons in the hidden layer. In [4], the theorem was generalized to arbitrary activation functions in a hidden layer. However, a network with one hidden layer requires a very large number of hidden layer neurons, which makes the network almost impossible to train. The problem is solved by using feedforward fully connected neural networks containing a large number of layers of relatively small width. In [5], it was proved that a network with two hidden layers is a universal approximator. Finally, it was proved in [6] that deep architecture networks with the ReLU activation function popular in such networks are universal approximators. All of the above theorems are not constructive, that is, they do not allow determining the structure of the network and choosing a learning algorithm.

At present, interest in solving direct and inverse boundary value problems on deep neural networks has sharply increased [7]. Networks for solving

boundary value problems are called "Physics-informed neural networks", which emphasizes that such networks are trained not by examples, but by minimizing residuals in sets of test points inside and at the boundary of the solution domain. The popularity of physics-informed neural networks for solving boundary value problems is largely due to the use of automatic differentiation [8], implemented in popular deep learning libraries, primarily TensorFlow. But deep learning libraries are focused mainly on the implementation of convolutional neural networks and do not support second-order gradient learning algorithms. The use of first-order gradient algorithms does not allow obtaining a high accuracy of the solution, while the networks are prone to overfitting [9]. To solve boundary value problems, we use neural networks of a different architecture - radial basis function networks (RBFNN) [10]. This paper presents the results of solving direct and inverse boundary value problems on RBFNN for piecewise homogeneous media.

1. Solving of Boundary Value Problems on Radial Basis Function Networks

Radial basis functions (RBF) [11] are functions of the distance between some point in space and a parameter of the function, called the center of the function. In this paper, consider the Gaussian

$$\varphi(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2a^2}\right),$$

where $\mathbf{c}$ is the vector of coordinates of the center of the function, a is the shape parameter (width).

In this paper, the Gauss function is used, since when solving boundary value problems described by second-order partial differential equations, it is necessary to calculate the second derivatives of RBF. The domain of the Gaussian function is comparable to the domain of its derivatives, which simplifies calculations.

RBFs are used in projection methods for solving boundary value problems [12–13]. Traditionally, we consider the stationary boundary value problem in the operator form

$$Lu(\mathbf{x}) = f(\mathbf{x}),\ \mathbf{x}\in\Omega,\quad Bu(\mathbf{x}) = p(\mathbf{x}),\ \mathbf{x}\in\partial\Omega, \quad (1)$$

where u is the solution to the problem; L — linear differential operator; B — linear operator of boundary conditions; Ω — solution area; $\partial\Omega$ — the border of the area; f and p are known functions. In projection methods, the solution is represented as a weighted sum of basis functions

$$u_{RBF}(\mathbf{x}) = \sum_{j=1}^{n_{RBF}} w_j \varphi_j(\mathbf{x}),\quad \mathbf{x}\in\bar{\Omega}=\Omega\cup\partial\Omega, \quad (2)$$

where φ_j is RBF; w_j — weight, n_{RBF} — number of RBFs.

The RBF parameters are selected a priori, and the weights are calculated as a solution to a system of linear algebraic equations obtained from the equality of residuals to zero at test points inside and on the boundary of the solution domain. When solving non-stationary problems, time can be considered as one of the coordinates, which can significantly complicate the task. It is easier to replace the time derivative with a finite difference and solve the stationary problem at each time step. The disadvantage of projection methods using RBF is the informal choice of RBF parameters. RBFNN [10] are free from this drawback, in the process of training which it is possible to adjust both the linear parameters of the network - weights, and the RBF parameters included in the basis functions nonlinearly.

The RBFNN structure is shown in Fig. 1. The network output is described by expression (2).

When solving problem (1) in the process of training the network, the loss function is minimized

$$I(\mathbf{w},\mathbf{p}) = \sum_{i=1}^{N} r_i^2 + \lambda \sum_{i=N+1}^{N+K} r_j^2, \quad (3)$$

$r_j = Bu_{RBF}(\mathbf{x}_j;\mathbf{w},\mathbf{p}) - p(\mathbf{x}_j)$, $\mathbf{x}_j\in\partial\Omega$ is the residual at the sample points on the boundary of the solution area, N and K is the number of test points in the solution area and on the boundary of the solution area, respectively.

As a result of network training, vectors of weights and RBF parameters are found.

In the well-known works [2, 14–15], the first-order gradient method - the gradient descent algorithm - is used for RBFNN training. This algorithm does not provide a high solution accuracy. In [16], a fast RBFNN learning algorithm based on the trust region method was developed. The method is complex, since it requires solving a conditional optimization problem at each step of training the network. In [17], an RBFNN learning algorithm

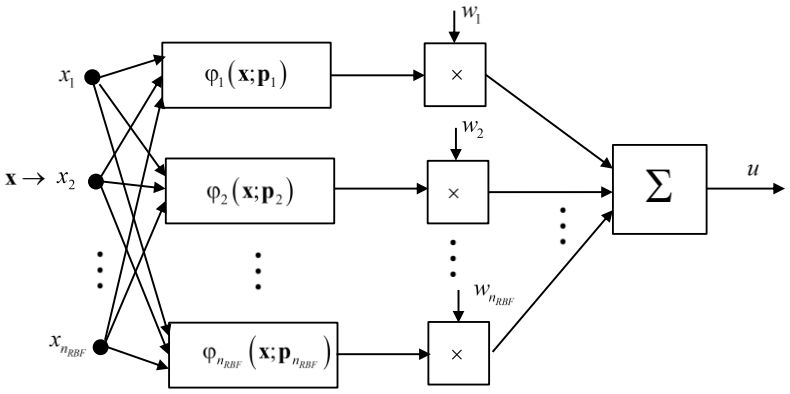


Fig. 1. RBFNN structure

based on the Levenberg – Marquardt optimization method is proposed. It is known [18] that the Levenberg – Marquardt method is equivalent to the confidence region method, but does not require solving constrained optimization problems.

Let us consider the algorithm of the Levenberg – Marquardt method using the example of solving the two-dimensional stationary problem (1). RBFNN parameters can be represented as a vector

$$\mathbf{\Theta} = \left[\mathbf{w}, \mathbf{c}_1, \mathbf{c}_1, \mathbf{a}\right]^{\mathrm{T}}, \qquad (4)$$

where $\mathbf{w}, \mathbf{c}_1, \mathbf{c}_1, \mathbf{a}$ are the vectors of the weights, the first and second coordinates of the centers, and the RBF network width, respectively.

At the *k*th iteration of RBFNN training, the vector of parameters (4) is corrected by the formula $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \Delta\boldsymbol{\theta}^{(k)}$, in which the vector of correction of parameters $\Delta\boldsymbol{\theta}^{(k)}$ is calculated as a result of solving the system of linear algebraic equations

$$\left(\mathbf{J}_{k-1}^{\mathrm{T}}\mathbf{J}_{k-1} + \mu_k\mathbf{E}\right)\Delta\boldsymbol{\theta}^{(k)} = -\mathbf{J}^{\mathrm{T}}\mathbf{r}, \qquad (5)$$

where $\mathbf{J}_{k-1}$ is the Jacobi matrix calculated from the values of the network parameters in the $(k-1)$th iteration, $\mathbf{E}$ is the unit matrix, μ_k is the regularization parameter, $\mathbf{r} = \left[r_1\ r_2 \ldots r_{N+K}\right]^{\mathrm{T}}$ is the vector of residuals at sample points in the solution domain and at the boundary of the solution domain.

We represent the Jacobi matrix in block form $\mathbf{J} = \left[\mathbf{J}_{\mathbf{w}} \mid \mathbf{J}_{\mathbf{c}_1} \mid \mathbf{J}_{\mathbf{c}_2} \mid \mathbf{J}_{\mathbf{a}}\right]$, where blocks are matrices of residual derivatives with respect to the network parameters. For example, the matrix $\mathbf{J}_{\mathbf{w}}$ has the form

$$\mathbf{J}_{\mathbf{w}} = \begin{bmatrix} \frac{\partial r_1}{\partial w_1} & \frac{\partial r_1}{\partial w_2} & \cdots & \frac{\partial r_1}{\partial w_{n_{RBF}}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial r_N}{\partial w_1} & \frac{\partial r_N}{\partial w_2} & \cdots & \frac{\partial r_N}{\partial w_{n_{RBF}}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial r_{N+K}}{\partial w_1} & \frac{\partial r_{N+K}}{\partial w_2} & \cdots & \frac{\partial r_{N+K}}{\partial w_{n_{RBF}}} \end{bmatrix}.$$

The elements of the Jacobi matrix are easy to calculate analytically [17]. The regularization parameter μ should decrease as the network learns, for example, linearly [18]. If the loss function decreases, then the current value of the regularization parameter is divided by the coefficient $\nu > 1$. If the loss function increases, then the current value of the regularization parameter is multiplied by ν. The ν coefficient must be selected.

2. Solution of Direct Boundary Value Problems for Piecewise Homogeneous Media

Many diffusion problems, for example, problems of modeling oil fields [19], thermal conductivity [20], and modeling groundwater [21] require consideration of piecewise homogeneous medium. The presence of a piecewise homogeneous medium means that the properties of the medium are constant in some regions, and the conjugation conditions are satisfied at the boundaries of the media. As a rule, the conditions of ideal conjugation should be fulfilled at the interface between the me-

dia [22]: equality of solutions and fluxes at the interface between the media. Equality of flows means a discontinuity of the derivative of the solution at the interface between the media.

When solving problems describing processes in piecewise homogeneous media, two restrictions arise on RBFNN: the impossibility of simulating the discontinuity of the derivative of the solution at the interfaces between the media and the nonlocal nature of the influence of RBF - RBF with an unbounded domain of definition, for example, Gaussian, affect the solution in all domains. The first limitation is removed by using a finite difference approximation [23]. But with this approach, an additional error of the difference approximation is introduced and the advantages of the gridless method are lost. The second limitation can be overcome by applying in each region RBFs with compactly supported functions, when RBFs of each region affect the solution only in its own region, and by introducing additional conjugation conditions at the boundaries of regions [24].

In [25], an approach to solving boundary value problems for piecewise homogeneous media is proposed, in which separate problems are solved for each domain with different properties of the medium, related by the conjugation conditions, which makes it possible to apply RBF with unbounded domains of definition. For RBFNN training, the fast algorithm of the Levenberg-Marquardt method [17] is used.

Let us consider the algorithm for solving the problem using the example of a problem for two subdomains described by the equation

$$\frac{\partial}{\partial x}\left(\sigma_i(x,y)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\sigma_i(x,y)\frac{\partial u}{\partial y}\right)=f(x,y),\quad (x,y)\in\Omega,\quad i=1,2, \tag{6}$$

$$u(x,y)=p(x,y),\qquad (x,y)\in\partial\Omega, \tag{7}$$

where Ω is the computational area; $\partial\Omega$ — the boundary of the computational domain; σ_i — a function that describes the properties of the environment.

The conjugation conditions must be met at the interface between the media

$$u_1\big|_S=u_2\big|_S,\ \sigma_1\frac{du_1}{dx}\bigg|_S=\sigma_2\frac{du_2}{dx}\bigg|_S. \tag{8}$$

Two problems (6)–(7) are solved for regions 1 and 2, taking into account the conjugation conditions (8). The loss function for problem 1 has the form (the function for problem 2 has a similar form)

$$J_1=\sum_{i=1}^{N_1}\left[L_1u_1(\mathbf{x}_{1i})-f_1(\mathbf{x}_{1i})\right]^2+\lambda_1\sum_{j=1}^{K_1}\left[B_1u_1(\mathbf{x}_{1j})-p_1(\mathbf{x}_{1j})\right]^2+\lambda_3\sum_{j=1}^{T}\left[u_1\left(\mathbf{x}_{Sj}\right)-u_2\left(\mathbf{x}_{Sj}\right)\right]^2+\lambda_4\sum_{j=1}^{T}\left[\sigma_1\frac{\partial u_1\left(\mathbf{x}_{Sj}\right)}{\partial x}-\sigma_2\frac{\partial u_2\left(\mathbf{x}_{Sj}\right)}{\partial x}\right]^2, \tag{9}$$

where L_1 is the differential operator, B_1 is the operator of boundary conditions, N_1 is the number of sample points inside area 1, K_1 is the number of sample points, on the boundary of area 1, T is the number of sample points at the interface between the media, $\lambda_1,\lambda_2,\lambda_3,\lambda_4$ are penalty factors $\mathbf{x}_{1i}$, $\mathbf{x}_{1j}$, and $\mathbf{x}_{Sj}$ are the coordinates of the test points inside, on the border of the region and at the interface of the media.

The solution process is iterative and uses two RBFNNs. At each iteration using the first RBFNN, one step is taken to minimize the loss function (9) for region 1, using the solution at the interface between the media and the flow $\sigma_2\partial u_2\left(\mathbf{x}_{Sj}\right)\big/\partial x$ of region 2 from the previous iteration. Then the step of training the second network for region 2 is performed in a similar way, using the obtained values of the solution at the interface between the media and the flow for region 1. For RBFNN training, it is proposed to use the Levenberg – Marquardt algorithm. Jacobi matrices differ from the one considered in Section 1 by the presence of derivatives with respect to network parameters from residuals with respect to conjugation conditions. Analytical expressions are obtained to calculate the elements of Jacobi matrices. The iterations of training the networks continue until the loss functions for regions 1 and 2 become small.

3. Solution of Inverse Boundary Value Problems for Piecewise Homogeneous Media

Let us consider the coefficient inverse problem of determining the unknown properties of a piecewise homogeneous medium from the results of

measuring the solution at several points. Coefficient inverse problems arise when determining the properties of materials, detecting anomalies in a certain environment, detecting objects, and many others. Solving such problems is very difficult. The problem was not solved in the gridless setting. In [26], using a neural network, the unknown diffusion coefficient was determined by the finite difference method. For a piecewise homogeneous medium, the problem was not solved.

To solve the problem in a meshless setting, we assume that the properties of the medium are approximately described by a continuous differentiable function. The problem is described by the equation

$$\frac{\partial}{\partial x_1}\left(k(\mathbf{x})\frac{\partial u}{\partial x_1}\right)+\frac{\partial}{\partial x_2}\left(k(\mathbf{x})\frac{\partial u}{\partial x_2}\right)=f(\mathbf{x}),\mathbf{x}\in\Omega \quad (10)$$

with boundary conditions

$$Bu(\mathbf{x})=p(\mathbf{x}),\quad \mathbf{x}\in\partial\Omega, \quad (11)$$

where $k(\mathbf{x})$ is a continuous unknown function describing the medium, $\mathbf{x}$ is the coordinate vector, Ω is the solution area, $\partial\Omega$ is the boundary of the area.

It is necessary to find the solution u and the function $k(\mathbf{x})$ by the solution, approximately known at some set of points

$$u(\mathbf{z})=\psi(\mathbf{z}),\quad \mathbf{z}\in Z,\quad Z\subset\Omega\cup\partial\Omega. \quad (12)$$

To solve the inverse problem, we will apply an approach known as parametric optimization [27]. For this, the unknown function $k(\mathbf{x})$ is approximated by RBFNN

$$k_{RBF}(\mathbf{x})=\sum_{m=1}^{M_k}w_m^k\varphi_m^k(\mathbf{x},\mathbf{p}_m^k), \quad (13)$$

where M_k is the number of radial basis functions, w_m^k, $\mathbf{p}_m^k$ are the weights and parameters of radial basis functions, φ_m^k is RBF.

The solution u of problem (10)–(11), in which the function $k(\mathbf{x})$ is approximated by the network (13), we approximate the second RBFNN

$$u_{RBF}(\mathbf{x})=\sum_{m=1}^{M_u}w_m^u\varphi^u(\mathbf{x};\mathbf{p}_m^u).$$

Let us construct the loss function in the form, taking into account the fulfillment of conditions (12)

$$I=\sum_{i=1}^{N}\left[Lu(\mathbf{x}_i,k(\mathbf{x}_i))-f(\mathbf{x}_i)\right]^2+$$
$$+\lambda_B\sum_{j=1}^{K}\left[Bu(\mathbf{x}_j)-p(\mathbf{x}_j)\right]^2+\lambda_D\sum_{m=1}^{S}\left[u(\mathbf{x}_m)-\psi(\mathbf{x}_m)\right]^2, \quad (14)$$

where $Lu(\mathbf{x}_i,k(\mathbf{x}_i))$ is the differential operator of Eq. (10), $Bu(\mathbf{x}_j)$ is the operator of boundary conditions (11), N is the number of sample points inside the region, K is the number of sample points on the boundary of the region, S is the number of points of additional conditions, λ_B, λ_D are penalty factors, $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_m$ — coordinates of test points inside, on the border of the area and coordinates of points of additional conditions.

As a regularizer of an ill-posed inverse problem, we use the iterative regularization method (Morozov's condition) [28]: the training of the network continues until

$$\sum_{m=1}^{S}\left[u(\mathbf{x}_m)-\psi(\mathbf{x}_m)\right]^2>S\delta^2,$$

where δ is the absolute measurement error of the solution at the points of conditions (12).

Let us take boundary conditions of the first kind $u(\mathbf{x})=p(\mathbf{x})$, $\mathbf{x}\in\partial\Omega$, and use the Gaussian function as radial basis functions. Then the differential operator of equation (10) takes the form

$$Lu(\mathbf{x}_i,k(\mathbf{x}_i))=S_1(\mathbf{x}_i)S_2(\mathbf{x}_i)+S_3(\mathbf{x}_i)S_4(\mathbf{x}_i)+$$
$$+S_5(\mathbf{x}_i)S_6(\mathbf{x}_i),$$

where

$$S_1(\mathbf{x}_i)=\sum_{m=1}^{M_k}\left[\frac{w_m^k}{\left(a_m^k\right)^2}\varphi_m^k(\mathbf{x}_i)\cdot\left(x_{i1}-c_{m1}^k\right)\right],$$

$$S_2(\mathbf{x}_i)=\sum_{m=1}^{M_u}\left[\frac{w_m^u}{\left(a_m^u\right)^2}\varphi_m^u(\mathbf{x}_i)\cdot\left(x_{i1}-c_{m1}^u\right)\right],$$

$$S_3(\mathbf{x}_i)=\sum_{m=1}^{M_k}\left[\frac{w_m^k}{\left(a_m^k\right)^2}\varphi_m^k(\mathbf{x}_i)\cdot\left(x_{i2}-c_{m2}^k\right)\right],$$

$$S_4(\mathbf{x}_i)=\sum_{m=1}^{M_u}\left[\frac{w_m^u}{\left(a_m^u\right)^2}\varphi_m^u(\mathbf{x}_i)\cdot\left(x_{i2}-c_{m2}^u\right)\right],$$

$$S_5 = \sum_{m=1}^{M_k} w_m^k \varphi_m^k\left(\mathbf{x}_i\right),$$

$$S_6\left(\mathbf{x}_i\right) = \sum_{m=1}^{M_u}\left[w_m^u \varphi_m^u\left(\mathbf{x}_i\right) \cdot \left(\frac{\left\|\mathbf{x}_i - \mathbf{c}_m^u\right\|^2 - 2\left(a_m^u\right)^2}{\left(a_m^u\right)^4} \right)\right].$$

To train the network, the algorithm of the Levenberg–Marquardt method [17] was used. To apply this algorithm, a single vector of parameters of the u_{RBF} and k_{RBF} networks is formed

$$\mathbf{\Theta} = \left[w_1^u \ldots w_{n_{RBF}^u}^u c_{11}^u \ldots c_{n_{RBF}^u 1}^u \ldots a_1^u \ldots a_{n_{RBF}^u}^u \ldots \right.$$
$$\left. \ldots w_1^k \ldots w_{n_{RBF}^k}^k c_{11}^k \ldots c_{n_{RBF}^k 1}^k \ldots a_1^k \ldots a_{n_{RBF}^k}^k \right]$$

where the superscript means the network belongs to u_{RBF} or k_{RBF}, n_{RBF}^u and n_{RBF}^k are the number of basis functions (neurons) in the networks.

Residuals at sample points: $1, 2, \ldots, N, N+1, \ldots, N+K, N+K+1, \ldots, N+K+S$ are numbered sequentially, and the Jacobi matrix is represented in block form

$$\mathbf{J} = \left[\mathbf{J}_{\mathbf{w}}^u \mid \mathbf{J}_{c_1}^u \mid \mathbf{J}_{c_2}^u \mid \mathbf{J}_a^u \mid \mathbf{J}_{\mathbf{w}}^k \mid \mathbf{J}_{c_1}^k \mid \mathbf{J}_{c_2}^k \mid \mathbf{J}_a^k \right]. \quad (15)$$

The elements of the Jacobi matrix (15) were calculated analytically. For example, the elements of the $\mathbf{J}_{\mathbf{w}}^u$ matrix for internal sample points are calculated by the formula

$$\frac{\partial r_i}{\partial w_p^u} = S_1\left(\mathbf{x}_i\right) \cdot \frac{\varphi_p^u\left(\mathbf{x}_i\right) \cdot \left(x_{i1} - c_{p1}^u\right)}{\left(a_p^u\right)^2} +$$
$$+ S_3\left(\mathbf{x}_i\right) \cdot \frac{\varphi_p^u\left(\mathbf{x}_i\right) \cdot \left(x_{i2} - c_{p2}^u\right)}{\left(a_p^u\right)^2} +$$
$$+ S_5\left(\mathbf{x}_i\right) \cdot \varphi_p^u\left(\mathbf{x}_i\right) \cdot \frac{\left\|\mathbf{x}_i - \mathbf{c}_p^u\right\|^2 - 2\left(a_p^u\right)^2}{\left(a_p^u\right)^4},$$
$$i = 1, 2, \ldots, N,$$

where $\varphi_p^u\left(\mathbf{x}_i\right)$ is the value of the p-th basic function of the network u_{RBF} at sample point $\mathbf{x}_i$, $\mathbf{c}_p^u$ and a_p^u are the parameters of the p-th basic function of the network ccc at the sample point $\mathbf{x}_i$. For boundary sample points, calculations are performed using the formula $\frac{\partial r_j}{\partial w_p^u} = \varphi_p^u\left(\mathbf{x}_j\right), \quad j = 1, 2, \ldots, K$. For the points of the additional condition, the elements of the matrix have the form

$$\frac{\partial r_m}{\partial w_p^u} = \varphi_p^u\left(\mathbf{x}_m\right), \quad m = 1, 2, \ldots, S.$$

The elements of the $\mathbf{J}_{c_1}^u$ matrix for internal sample points have the form

$$\frac{\partial r_i}{\partial c_{p1}^u} = \frac{w_p^u \cdot \varphi_p^u\left(\mathbf{x}_i\right) \cdot \left(x_{i1} - c_{p1}^u\right)}{\left(a_p^u\right)^4} \times$$
$$\times \left\{ S_1\left(\mathbf{x}_i\right) \left[\left(x_{i1} - c_{p1}^u\right) - \frac{\left(a_p^u\right)^2}{x_{i1} - c_{p1}^u} \right] + \right.$$
$$\left. + S_3\left(\mathbf{x}_i\right) \cdot \left(x_{i2} - c_{p2}^u\right) + S_5\left(\mathbf{x}_i\right) \cdot \frac{\left\|\mathbf{x}_i - \mathbf{c}_p^u\right\|^2 - 4\left(a_p^u\right)^2}{\left(a_p^u\right)^2} \right\}.$$

For boundary sample points, the elements of the $\mathbf{J}_{c_1}$ matrix are written in the form

$$\frac{\partial r_j}{\partial c_{p1}^u} = w_p^u \varphi_p^u\left(\mathbf{x}_j\right) \cdot \frac{\left(x_{j1} - c_{p1}^u\right)}{a_p^2}, \qquad j = 1, 2, \ldots, K.$$

For the points of the additional condition, the elements of the $\mathbf{J}_{c_1}$ matrix have the form

$$\frac{\partial r_m}{\partial c_{p1}^u} = w_p^u \varphi_p^u\left(\mathbf{x}_m\right) \cdot \frac{\left(x_{m1} - c_{p1}^u\right)}{a_p^2}, \quad m = 1, 2, \ldots, S.$$

Other elements of the Jacobi matrix are calculated similarly.

4. Experimental Study

The direct problem for a piecewise homogeneous medium (6)–(8) was solved for the region 1×1, divided at $x = 0.5$ by a vertical line into two subdomains with constant values of the function describing the medium $k_1 = 2$ and $k_2 = 5$. The Gaussian function is used as RBF. For each area, the number of RBFs is 64. The number of internal sample points for each area is N = 80. The number of boundary sampling points for the two regions is K = 80.

The number of sample points on the media interface is 20. RBFNN was initialized with zero weights and constant widths of 0.2. The network was trained by the developed algorithm of the Levenberg – Marquardt method to the root mean square of the loss function equal to 10^{-12}. Regularization parameters kk are selected for each region: $\mu_1 = 20,\ \mu_2 = 40$. The specified value of the loss function was reached in 1500 iterations. The result of the solution is shown in Fig. 2.

The centers, width, and weights before and after training the network for the two regions are shown in Fig. 3. The fill intensity of the circles shows the values of the weights. The radius of the circles is proportional to the width of the RBF. A significant change in RBF parameters after network training shows the importance of adjusting RBF parameters.

The inverse problem (10)–(11) was solved under the same conditions as the direct problem. In the domain of solving the direct problem, 40 points located on the grid were taken, the solution in which was taken as the values at the points of additional conditions (12). Moreover, the points were not located on the interface line of the media. RBFNN, modeling the environment function and the solution of the direct problem, each contained 64 RBF, the number of test points inside the solution area, on the boundary and additional conditions, respectively, is equal to $N = K = 80,\ S = 40.$ Penalty coefficients are equal $\lambda_B = \lambda_D = 10.$

The network parameters were the same as for solving the direct problem. Both networks were trained by the Levenberg-Marquardt algorithm up to the mean square value of the loss function (14), taking into account the Morozov condition equal to 10^{-3}. Although the properties of the medium are approximated by a continuous function, from the results of solving the inverse problem it is possible to approximately determine the position of the boundary of the media (Fig. 4) and to quite accurately reconstruct the values of the medium function in different regions.

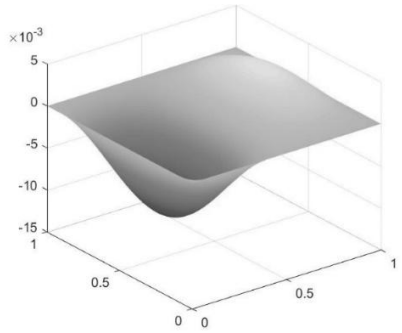
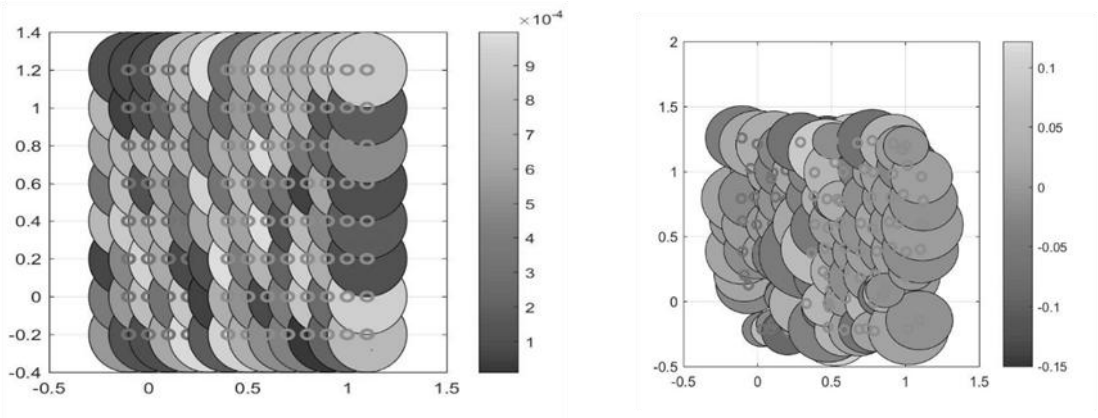


Fig. 2. The graph of the solution of the problem for a piecewise homogeneous medium



а) б)

Fig. 3. Centers, width and weights (a - before training the network, b - after training the network)

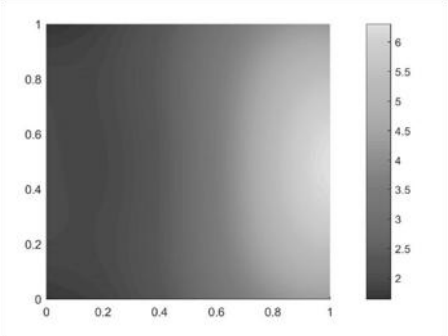


Fig. 4. Calculated values of the environment function

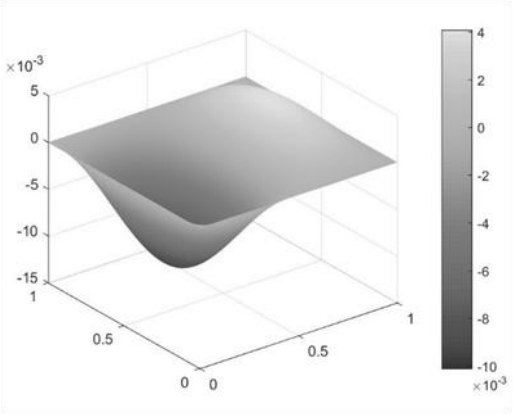


Fig. 5. Reconstructed solution of the direct problem

As shown in Fig. 5, the solution to the direct problem coincides with the solution obtained earlier.

Conclusion

For the first time, a meshless algorithm for solving inverse coefficient boundary value problems for piecewise homogeneous media on radial basis functions networks is developed. The algorithm for solving the inverse problem is based on the approximation of the properties of the environment by a radial basis functions networks and joint training of networks that approximate the solution of the direct problem and the properties of the environment. To train radial basis functions networks, we used an algorithm previously developed by the authors and based on the Levenberg-Marquardt method. Analytical expressions are obtained for elements of Jacobi matrices used for training networks. Additional conditions for solving the inverse problem are obtained as a result of solving the direct problem using the algorithm previously proposed by the authors on radial basis functions network. Despite the approximation of a piecewise homogeneous medium by a smooth function, the proposed algorithm made it possible to approximately identify the properties of the medium and restore the solution of the direct problem with high accuracy.

In the future, experiments with various inhomogeneities of the medium are planned. It is planned to improve the algorithm for solving the inverse problem in order to determine the position of the the border of areas. To reduce preparatory work and simplify programming, it is proposed to use machine learning libraries, in particular, TensorFlow, which implement automatic differentiation. This will require the development of custom library extensions.

References

1. Lagaris, I.E., A. Likas, D.I. Fotiadis. 1998. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks. 9(5): 987–1000. doi: 10.1109/72.712178.
2. Yadav, N., Yadav, A., Kumar, M. 2015. An Introduction to Neural Network Methods for Differential Equations. Dordrecht: Springer. 115 p.
3. Cybenko, G. 1989. Approximation by Superposition of a Sigmoidal Function. Mathematics of Control, Signals and Systems. 2: 303–314. doi: 0.1007/BF02551274.
4. Hornik, K., Stinchcombe, M., White, H. 1989. Multilayer feedforward networks are universal approximators. Neural networks. 2(5): 359–366. doi: 10.1016/0893-6080(89)90020-8.
5. Kurkova, V. 1992. Kolmogorov's theorem and multilayer neural networks. Neural networks. 5(3): 501-506. doi: 0893-6080(92)90012-8
6. Hanin, B. 2019. Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations. Mathematics. 7(10), 992: 1–9. doi: 10.3390/math7100992.
7. Raissi, M., Perdikaris, P., Karniadakis, G.E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. 378: 686–707. doi: 10.1016/j.jcp.2018.10.045.
8. Bavdin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M. 2018. Automatic Differentiation in Machine Learning: a Survey. Journal of Machine Learning Research. 18 (1): 1–43. Available at: https://jmlr.org/papers/volume18/17-468/17-468.pdf (accessed July 08, 2021).
9. Avrutskiy, V.I. 2021. Enhancing Function Approximation Abilities of Neural Networks by Training Derivatives.

IEEE Transactions on Neural Networks and Learning Systems. 32(2): 916 – 924. doi: 10.1109/TNNLS.2020.2979706.
10. Aggarwal, C.C. 2018. Neural Networks and Deep Learning. Cham: Springer. 520 p.
11. Buhmann, M.D. 2004. Radial Basis Functions: Theory and Implementations. Cambridge Cambridge: University Press. 259 p.
12. Kansa, E.J. 1990. Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — I surface approximations and partial derivative estimates. Computers&Mathematics with Applications. 19(8–9): 127–145. doi: 10.1016/0898-1221(90)90270-T.
13. Kansa, E.J. 1990. Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — II solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers&Mathematics with Applications. 19(8–9): 147–161. doi: 10.1016/0898-1221(90)90271-K.
14. Jianyu, L., Siwei, L., Yingjian, Q. 2003. Numerical solution of elliptic partial differential equation by growing radial basis function neural networks. Neural Networks. 16(5–6): 729–734. doi: 10.1016/S0893-6080(03)00083-2.
15. Chen, W., Fu, Z.J. 2014. Recent Advances in Radial Basis Function Collocation Methods. Cham: Springer. 90 p.
16. Gorbachenko, V.I., Zhukov, M.V. 2017. Solving boundary value problems of mathematical physics using radial basis function networks. Computational Mathematics and Mathematical Physics. 57(1): 145–155. doi: 10.1134/S0965542517010079.
17. Gorbachenko, V.I., Alqezweeni, M.M. 2019. Learning Radial Basis Functions Networks in Solving Boundary Value Problems.: 2019 International Russian Automation Conference. Sochi, Russia September 8-14, 2019: 1–6.
18. Marquardt, D.W. 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics. 11(2): 431–441. doi: 10.1137/0111030.
19. Islam, M.R., Abou-Kassem, J.H., Farouq-Ali, S.M. 2020. Petroleum Reservoir Simulation: The Engineering Approach. Houston: Gulf Professional Publishing. 526 p.
20. Carslaw, H.S. 2018. Introduction to the Mathematical Theory of the Conduction of Heat in Solids. Blacweel: Franklin Classic. 286 p.
21. Anderson, M.P., Woessner, W.W., Hunt, R.J. 2015. Applied Groundwater Modeling: Simulation of Flow and Advective Transport. Cambridge: Academic Press. 564 p.
22. Samarskii, A.A., Vabishchevich, P.N. 1996. The Finite Difference Methodology, Volume 2, Computational Heat Transfer. New York: Wiley. 432 p.
23. Ngo-Cong, D., Tien, C.M.T., Nguyen-Ky, T., An-Vo, D.-A., Mai-Duy, N., Strunin, D.V., Tran-Cong, T. 2017. A generalised finite difference scheme based on compact integrated radial basis function for flow in heterogeneous soils. International Journal for Numerical Methods in Fluids. 85(7): 404-429. doi: 10.1002/fld.4386.
24. Chen, J.-S., Wang, L., Hu, H.-Y., Chi, S.-W. 2009. Subdomain radial basis collocation method for heterogeneous media. International journal for numerical methods in engineering. 80(2): 163–190. doi: 10.1002/nme.2624.
25. Stenkin, D.A., Gorbachenko, V.I. 2021. Solving Equations Describing Processes in a Piecewise Homogeneous Medium on Radial Basis Functions Networks. Kryzhanovsky, B., Dunin-Barkowski, W., Redko, V., Tiumentsev, Y. (eds) Advances in Neural Computation, Machine Learning, and Cognitive Research IV. NEUROINFORMATICS 2020. Studies in Computational Intelligence. Cham: Springer. 925: 412–419. doi: 0.1007/978-3-030-60577-3_49.
26. Pakravan, S., Mistani, P.A., Aragon-Calvo, M.A., Gibou, F. Solving inverse-PDE problems with physics-aware neural networks. Available at: https://arxiv.org/abs/2001.03608 (accessed July 08, 2021).
27. Samarskii, A.A., Vabishchevich, P.N. 2007. Numerical Methods for Solving Inverse Problems of Mathematical Physics. Berlin: Walter de Gruyter. 454 p.
28. Morozov, V.A. 1984. Methods for Solving Incorrectly Posed Problems. New York: Springe. 280 p.

Gorbachenko V. I. Federal State Budgetary Educational Institution of Higher Education "Penza State University", Penza, Russia. Professor, Doctor of Technical Sciences, Head of the Department of Computer Technologies. Number of publications: 540 (including 4 monographs). Research interests: neural networks, machine learning, neural network methods for solving boundary value problems in mathematical physics, neural network methods of diagnostics and forecasting in medicine, computer modeling, computational mathematics. E-mail: gorvi@mail.ru

Stenkin D. A. Federal State Budgetary Educational Institution of Higher Education "Penza State University", Penza, Russia. Master student of the direction "Applied Mathematics and Informatics". Number of publications: 10. Research interests: neural networks, neural network methods for solving boundary value problems of mathematical physics. E-mail: stynukin@mail.ru