

Applications of Topological Derivatives and Neural Networks for Inverse Problems

Marta Grzanek and Katarzyna Szulc

Abstract Numerical method of identification for small circular openings in the domain of integration of an elliptic equation is presented. The method combines the asymptotic analysis of PDE's with an application of neural networks. The asymptotic analysis is performed in singularly perturbed geometrical domains with the imperfections in form of small voids and results in the form of the so-called topological derivatives of observation functionals for the inverse problem under study. Neural networks are used in order to find the mapping which associates to the observation shape functionals the conditional expectation of the size and location of the imperfections. The observation is given by a finite number of shape functionals. The approximation of the shape functionals by using the topological derivatives is used to prepare the training data for the learning process of an artificial neural network. Numerical results of the computations are presented and the probabilistic error analysis of such an identification method of the holes by neural network is performed.

1 Introduction

The paper describes a new numerical method which can be used for numerical identification of the conditional expectation of *small imperfections* in geometrical domains. The numerical method is implemented in neural networks, therefore, it requires the learning data set. Such a set can be constructed in general with a high cost of the computational effort. We propose the specific method of construction

Marta Grzanek
University of Łódź, Faculty of Mathematics and Computer Science, Banacha 22, 90-238 Łódź,
Poland, e-mail: marta@math.uni.lodz.pl

Katarzyna Szulc
Université Henri Poincaré Nancy 1, Institut Elie Cartan, B.P. 239, 54506 Vandœuvre-Lès-Nancy
Cedex, France, e-mail: katarzyna.szulc@iecn.u-nancy.fr

which reduces the cost by careful asymptotic analysis of the mathematical model which describes the observation of the real object. Such an observation is given by a finite number of functionals with their values depending on the imperfections to be identified. The simplest applied problem could be:

Identify a microcrack in the elastic body Ω from the finite number of elastic modes $\Lambda = (\lambda_1, \dots, \lambda_M)$.

The microcrack should be characterized by a finite number of parameters, say $\Phi = (\ell_1, \dots, \ell_N)$, and we should have in hand the relation between the elastic modes Λ and the form of microcrack Φ . This relation can be formally denoted by

$$\mathcal{F} : \Phi \mapsto \Lambda$$

and the learning set for neural network includes the set of the pointwise values $(\Phi_k, \Lambda_k), k = 1, \dots$ obtained from mathematical model of such a relation in order to model the inverse relation $(\Lambda_k, \Phi_k), k = 1, \dots$. The inverse relation is approximated by a neural network after the appropriate training procedure. Then, this approximation in hand, we can try to answer the question:

Given real observation of elastic modes of an elastic body, determine the size, the shape and the location of a finite number of microcracks in the body if the set $\tilde{\Lambda}$ of elastic modes is measured for the specific body.

If we want to find one microcrack, the formal description of our problem is just

$$\tilde{\Phi} = \mathcal{F}^{-1}(\tilde{\Lambda}) . \quad (1)$$

This problem is an inverse problem which is, it seems, quite important for applications. There are difficult questions associated with such a problem:

- what is the meaning of the inverse mapping
- how for given value of observation $\tilde{\Lambda}$ the required value $\tilde{\Phi}$ of parameters which characterize the unknown defect or imperfection in the form of a microcrack can be computed in the way that the proposed method is robust
- if the method is convergent and in what sense

The inverse problem is a subject of the research, in the paper we restrict ourselves to a simplified variant which shows that the proposed methodology is promising and can be possibly developed for some real life inverse problems. The framework of our method includes the PDE model of our object and its asymptotic analysis in singularly perturbed geometrical domains, finite number of parameters which model the imperfections to be identified, finite number of observation functionals which can be evaluated from the PDE model and measured from the real life object, and finally the neural network which model the inverse mapping according to the standard rules in such approach. We refer to [3] for an introduction to the methodology of numerical solutions to inverse problems with the asymptotic analysis and neural networks. For voids of arbitrary shape, the topological derivatives are determined in function of the so-called polarization tensors and virtual mass tensors, we refer the

reader to [6] for the detailed description of the results for spectral problems and the energy functional for Laplacian.

In [3], a particular problem is considered that guarantees existence of the inverse mapping. In this paper we don't know whether the inverse mapping exists. Based on probability theory [12] we obtain only the conditional expectation of the inverse mapping. So in our meaning **the “inverse mapping” is the conditional expectation of multifunction that for the vector of observation calculates the location and the size of imperfections.**

2 Preliminaries

We propose numerical method for identification of a finite number of imperfections in geometrical domain $\Omega_0 \subset \mathbb{R}^2$. The imperfections $\mathcal{B}_j(y_j)$, $j = 1, \dots, k$ are small voids or holes of radii ρ_j , $j = 1, \dots, k$, included in Ω_0 , so the geometrical domain with the given number k of imperfections is denoted by Ω_k (see Section 3.1 for details). We point out that k holes $\mathcal{B}_j(y_j)$, $j = 1, \dots, k$ are included in the domain Ω_k .

The identification procedure is based on the knowledge of N shape functionals $\mathcal{J}_i(\Omega_k)$, $i = 1, \dots, N$, which are evaluated for the solutions $u(\Omega_k)$ of a PDE defined in Ω_k . Since $\Omega_k = \Omega_0 \setminus \bigcup_{j=1}^k \overline{\mathcal{B}_j(y_j)}$, and $\mathcal{B}_j(y_j) = \{x : |x - y_j| < \rho_j\}$, it follows that $\mathcal{J}_i(\Omega_k)$ depends on $3k$ parameters, i.e., $y_j \in \mathbb{R}^2$ and $\rho_j > 0$, $j = 1, \dots, k$.

The numerical procedure means, e.g., the *minimization* with respect to $3k$ parameters of the gap between the values of observation functionals which are given for the real object, and the values which are obtained from the mathematical model. The values from the mathematical model are determined in the form of the function

$$\mathcal{G}_k : \mathbb{R}^{3k} \ni (y_1, \rho_1, \dots, y_k, \rho_k) \mapsto (\mathcal{J}_1(\Omega_k), \dots, \mathcal{J}_N(\Omega_k)) \in \mathbb{R}^N \quad (2)$$

Therefore, the inverse problem can be defined by e.g., the minimization of the following goal functional

$$\min_{(Y, \rho)} \text{dist}(\mathcal{G}_k(Y, \rho), \mathcal{J}(\Omega_k)) \quad (3)$$

where $(Y, \rho) = (y_1, \rho_1, \dots, y_k, \rho_k)$ and $\mathcal{J}(\Omega_k) = (\mathcal{J}_1, \dots, \mathcal{J}_N) \in \mathbb{R}^N$ are the values determined for the real object.

In order to simplify the numerical procedure, the mapping $\mathcal{G}_k(Y, \rho)$ is replaced in (3) by its approximation given in terms of the so-called topological derivatives. In Section 3 the inverse problem is introduced. The observation shape functionals (4) are defined in terms of solutions by (5). The proposed approximation of the observation operator (6) is given by formula (7).

In Section 3 some observation shape functionals are introduced in (4) and the form of the topological derivatives for such functionals is presented in Theorem 1.

In Subsection 3.2 the observation shape functionals, used in Section 4, depend on the solution to the Laplace equation defined in Ω_0 and in each domain Ω_k , $k = 3$, with the homogeneous Neumann boundary conditions on the boundaries γ_j of the holes \mathcal{B}_j , respectively.

In Subsection 4.3 the numerical realization of the identification procedure based on the application of artificial neural networks is described in detail. The computations are performed for one hole, the results of computations are given in the last part of the paper.

The proposed numerical procedure can be characterized by the following features. The learning data sets for the neural networks are constructed using the asymptotic analysis of the observation shape functionals. Such a method of construction is very useful from the numerical point of view but unfortunately restricts the validity of the proposed approach only for small radii of imperfections. We assume also that the given data for the inverse problem are exact in the sense that there are some unknown imperfections in the form of circular holes which furnish the prescribed values of the observation shape functionals. Therefore, the inverse mapping $\mathcal{G}_k^{-1}(\mathcal{J}(\Omega_k)) = (Y, \rho)$ can be defined and it is given by the neural network. In fact, the situation is more complicated, since we can only model the conditional expectation, we refer to Section 4 for details. So instead of values of inverse mapping we calculate only conditional expectation of these values. The positive conclusion for our approach is the ‘‘probabilistic’’ convergence of the numerical procedure which, to the best of our knowledge, is an original contribution to the field of inverse problems.

3 Shape Optimization

In this paper we present the identification method for finite number of holes in a given geometrical domain $\Omega_0 \subset \mathbb{R}^2$. Actually, we want to identify the locations $y_j \in \Omega_0$ and the radii ρ_j , $j = 1, \dots, k$ of k small holes $\mathcal{B}_j(y_j) = \{x : |x - y_j| < \rho_j\}$ included in Ω_0 .

The procedure of identification is based on the values of specific shape functionals $\mathcal{J}_i(\Omega_k)$, $i = 1, \dots, N$, which depend on the solutions $u(x)$, $x \in \Omega_k$, for the Laplace equation in the domain Ω_k . Here we denote by Ω_0 , the domain without hole, $\Omega_1 = \Omega_0 \setminus \overline{\mathcal{B}_1(y_1)}$, for $k = 1$, is the domain with one hole and for $k \geq 2$, Ω_k is the domain with k holes. For our numerical examples we use $k = 1$ or $k = 3$. By the proposed identification procedure we can compute the coordinates $y_j \in \Omega_0$ of centers of the holes and the radii ρ_j such that given values of observation from physical model denoted by $\mathcal{J}_1, \dots, \mathcal{J}_N$, $j = 1, \dots, k$ coincide approximatively with the values $\mathcal{J}_i(\Omega_k)$, $i = 1, \dots, N$, evaluated from the mathematical model (see Fig. 1).

Let us consider the following inverse problem. We assume that the shape functionals take the form

$$\mathcal{J}_i(\Omega_k) = \int_{\Omega_k} \mathcal{F}_i(x, u(x), \nabla u(x)) dx \quad i = 1, \dots, N, \quad (4)$$

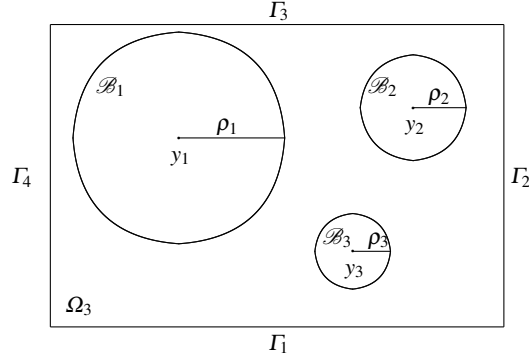


Fig. 1 Domain Ω_3 with imperfections $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$

where $\Omega_k = \Omega_0 \setminus \bigcup_{j=1}^k \overline{\mathcal{B}_j(y_j)}$, $\Omega_k \subset \mathbb{R}^2$, are given domains. We assume that $\mathcal{F}_i(\cdot, \cdot, \cdot)$ are given smooth functions. The partial differential equations for $u \in H^1(\Omega_k)$ are defined as follows:

$$\begin{cases} \Delta u = f & \text{in } \Omega_k, \\ u = g & \text{on } \Gamma_1 \cup \Gamma_3, \\ \frac{\partial u}{\partial n} = h & \text{on } \Gamma_2 \cup \Gamma_4, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \bigcup_{j=1}^k \gamma_j, \end{cases} \quad (5)$$

where f, g, h are given continuous functions in $\overline{\Omega_0}$ and $H^1(\Omega)$ stands for the standard Sobolev space of functions in $L^2(\Omega)$ along with the first order derivatives. The identification procedure for the location and size of several holes by using neural networks is presented in the present paper. Therefore, we are interested from numerical point of view, in the generalized inverse of the mapping

$$\mathcal{G}_k : \mathbb{R}^{3k} \ni (y_j, \rho_j, \dots, y_k, \rho_k) \mapsto (\mathcal{J}_1(\Omega_k), \dots, \mathcal{J}_N(\Omega_k)) \in \mathbb{R}^N \quad (6)$$

where (y_j, ρ_j) represents the center and radius of the hole \mathcal{B}_j . To this end we construct its approximation in the form

$$\mathcal{G}_{ki} \cong \mathcal{J}_i(\Omega_k) + \sum_{j=1}^k \frac{\rho_j^2}{2} \mathcal{T}_{\Omega_0} \mathcal{J}_i(y_j) \quad (7)$$

and determine an inverse of this approximation, instead of the inverses of mapping \mathcal{G}_k . From the mathematical point of view, the inverse of the mapping \mathcal{G}_k is difficult to evaluate. In this case we can use artificial neural networks to construct an approximation of the inverse of mapping (6) taking into account its approximation (7).

3.1 Analysis of Inverse Problems with Multiple Holes

In the present section we investigate some theoretical aspects of the inverse problem. To begin with, let us consider the domain Ω_k with k small holes $\mathcal{B}_j(y_j)$, $j = 1, \dots, k$. It is well known ([3]) that we can identify the hole $\mathcal{B}_1(y_1)$ by means of the measurements performed inside of Ω_0 , as well as, on the exterior boundary of Ω_0 . We propose to extend this method of identification to the case of several holes.

In such a case we consider the following boundary value problem defined in Ω_k . Find $u \in H^1(\Omega_k)$ such that

$$\begin{cases} \Delta u = 0 & \text{in } \Omega_k, \\ u = 1 & \text{on } \Gamma_1, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \Gamma_2 \cup \Gamma_4 \cup \bigcup_{j=1}^k \gamma_j, \\ u = 0 & \text{on } \Gamma_3, \end{cases} \quad (8)$$

where γ_j denotes the boundary of the hole $\mathcal{B}_j(y_j)$. We denote by $u = u(\Omega_k)$ the solution to (8) in Ω_k .

The asymptotic expansion of the shape functional (3) is given by

$$\mathcal{J}_i(\Omega_k) = \mathcal{J}_i(\Omega_0) + \sum_{j=1}^k \frac{\rho_j^2}{2} (\mathcal{T}_{\Omega_0} \mathcal{J}_i)(y_j) + o(\rho^2), \quad \text{with } \rho = (\rho_1, \dots, \rho_k). \quad (9)$$

where $\mathcal{T}_{\Omega_0} \mathcal{J}_i$ is the function defined in Ω_0 , it is the topological derivative of the shape functional $\mathcal{J}_i(\Omega)$ evaluated at y_j in the domain Ω_0 . For simplicity, we assume that

$$\mathcal{F}_i(x, u, q) = F_i(u) + G_i(q). \quad (10)$$

Hence, the shape functional $\mathcal{J}_i(\Omega_k)$ takes the form

$$\mathcal{J}_i(\Omega_k) = \int_{\Omega_k} [F_i(u(\Omega_k)) + G_i(\nabla u(\Omega_k))] dx \quad (11)$$

The formula for topological derivative of the shape functional [10, 11] defined by (11) is given in the following theorem [3].

Theorem 1. *The topological derivative of functional (11) at a point $y \in \Omega_0$ is given by the following formula*

$$\mathcal{T}_{\Omega_0} \mathcal{J}_i(y) = -\frac{1}{2\pi} [2\pi F_i(u(y)) + g_i(\nabla u(y)) + 2\pi f(y)v(y) + 4\pi \nabla u(y) \cdot \nabla v(y)] \quad (12)$$

where $\nabla u(y) = (a, b)^\top$ and the function g_i depending on the gradient of solution at the point $y \in \Omega_0$ takes the form

$$g_i(\nabla u(y)) = \frac{1}{2\pi} \int_0^{2\pi} G_i(a \sin^2 \vartheta - b \sin \vartheta \cos \vartheta, -a \sin \vartheta \cos \vartheta + b \cos^2 \vartheta) d\vartheta. \quad (13)$$

The adjoint state $v \in H_{\Gamma_1}^1(\Omega_0) = \{\phi \in H^1(\Omega_0) \mid \phi = 0 \text{ on } \Gamma_1\}$ solves the boundary value problem

$$-\int_{\Omega_0} \nabla v \cdot \nabla \phi dx = -\int_{\Omega_0} [F'_i(u)\phi + G_{iq}(\nabla u) \cdot \nabla \phi] dx, \quad \forall \phi \in H_{\Gamma_1}^1(\Omega_0). \quad (14)$$

3.2 Numerical Example of Observation Shape Functionals

Let Ω_0 be the square $(0, 1) \times (0, 1)$. We consider three cases, the domain Ω_1 with one hole, and the domains Ω_2, Ω_3 with two and three holes, respectively. In each case the asymptotic approximations of solutions in $\Omega_1, \Omega_2, \Omega_3$ can be computed in the fixed domain Ω_0 . Therefore, we consider four boundary value problems defined in Ω_0 for harmonic functions $\Delta u = 0$ in Ω_0 with the different boundary conditions.

For the first boundary value problem the following boundary conditions are prescribed for the solution $u = u_1$ with $i = 1$,

$$\begin{aligned} u_1 &= 1 - x_1 \text{ on } (0, 1) \times \{0\}, \\ u_1 &= 0 \quad \text{on } \{1\} \times (0, 1), \\ u_1 &= 0 \quad \text{on } (0, 1) \times \{1\}, \\ u_1 &= 1 - x_2 \text{ on } \{0\} \times (0, 1). \end{aligned} \quad (15)$$

The other three cases $i = 2, 3, 4$ of boundary conditions are obtained from the above conditions by applying the successive rotation by the angle $\frac{\pi}{2}$. The observation shape functionals $\mathcal{J}_i = \mathcal{J}_i(\Omega_0) = \int_{\Omega_0} [F_i(u) + G_i(\nabla u)] dx$ are defined as follows: for $i = 1, 2, 3, 4$

$$\begin{aligned} \mathcal{J}_{[1+3(i-1)]} &= \int_{\Omega} u_i^2 dx, \quad \mathcal{J}_{[2+3(i-1)]} = \int_{\Omega} \left(\frac{\partial u_i}{\partial x_1} \right)^2 dx, \\ \mathcal{J}_{[3+3(i-1)]} &= \int_{\Omega} \left(\frac{\partial u_i}{\partial x_2} \right)^2 dx \end{aligned} \quad (16)$$

In the perforated domains $\Omega_k = \Omega_0 \setminus \bigcup_{j=1}^k \overline{\mathcal{B}_j(y_j)}$, $y_j = (y_j^1, y_j^2)$, we prescribe for the solution $u = u(\Omega_k)$ the homogenous Neumann boundary conditions on the boundaries γ_j of the balls $\mathcal{B}_j(y_j)$.

Topological derivatives of the shape functionals are obtained from Theorem 1 by direct computation of the function g_i . For the functionals in the form $\mathcal{J}_{[1+3(i-1)]}$ the topological derivative is given by

$$\mathcal{T}_{\Omega_0} \mathcal{J}_1(y_j) = -[u(y_j)]^2 + 2\nabla u(y_j) \nabla v_1(y_j), \quad (17)$$

where $v = v_i$ is the adjoint state, defined for $i = 1, 2, 3$ by a solution to the boundary value problem

$$\begin{cases} \Delta v = -F'_i(u) + \operatorname{div}(G_{iq}(\nabla u)) & \text{in } \Omega_0, \\ v = 0 & \text{on } \Gamma_1 \cup \Gamma_3, \\ \frac{\partial v}{\partial n} = 0 & \text{on } \Gamma_2 \cup \Gamma_4, \end{cases} \quad (18)$$

For the functionals in the form $\mathcal{J}_{[2+3(i-1)]}$ and $\mathcal{J}_{[3+3(i-1)]}$, we observe that

$$\frac{\partial u}{\partial \tau} = \cos \vartheta \cdot \frac{\partial u}{\partial x_1} + \sin \vartheta \cdot \frac{\partial u}{\partial x_2} \quad (19)$$

hence we can use Theorem 1 with

$$G(\nabla u) = (\sin \vartheta \frac{\partial u}{\partial \tau})^2 = (-\sin^2 \vartheta \frac{\partial u}{\partial x_1} + \sin \vartheta \cos \vartheta \frac{\partial u}{\partial x_2})^2 \quad (20)$$

As a result we obtain

$$\mathcal{T}_{\Omega_0} \mathcal{J}_2(y_j) = -\pi \left[\frac{3}{2} \left(\frac{\partial u}{\partial x_1} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial x_2} \right)^2 + 4(\nabla u \cdot \nabla v_2) \right] (y_j) \quad (21)$$

$$\mathcal{T}_{\Omega_0} \mathcal{J}_3(y_j) = -\pi \left[\frac{1}{2} \left(\frac{\partial u}{\partial x_1} \right)^2 + \frac{3}{2} \left(\frac{\partial u}{\partial x_2} \right)^2 + 4(\nabla u \cdot \nabla v_3) \right] (y_j) \quad (22)$$

where u is the harmonic function in Ω_0 with the boundary conditions determined for $i = 1$ by (15), v_1 is the adjoint state for $\mathcal{J}_1(\Omega_0) = \int_{\Omega} u_1^2 dx$, v_2 is the adjoint state for $\mathcal{J}_2(\Omega_0) = \int_{\Omega_0} (\frac{\partial u_1}{\partial x_1})^2 dx$ and v_3 is the adjoint state for $\mathcal{J}_3(\Omega_0) = \int_{\Omega_0} (\frac{\partial u_1}{\partial x_2})^2 dx$. The remaining formulae for topological derivatives are obtained in the same way from (17) and (14), (18).

4 Neural Networks

We are going to present the method which is successfully tested for numerical solution of the inverse problem under considerations. We assume that the distance between the observations from real object and from the mathematical model equals zero, it means that there is (Y, ρ) such that

$$\mathcal{G}_k(Y, \rho) = \mathcal{J}(\Omega_k) \in \mathbb{R}^N. \quad (23)$$

Thus, it makes sense to consider the inverse mapping \mathcal{G}_k^{-1} . Basing on probability theory [12] we obtain only the conditional expectation of the inverse mapping. So in our meaning the ‘‘inverse mapping’’ is the conditional expectation of multifunction that for the vector of observation calculates the location and the size of

imperfections. It is general case for the problem that is considered in [3]. Moreover, Theorem 4.1 in [3] shows, under appropriate conditions, the existence of the inverse mapping. Therefore, the neural network is constructed to approximate the inverse mapping \mathcal{G}_k^{-1} . In practice, using probability theory once more, we can only approximate the conditional expectation of the location and radii of k holes instead of the inverse mapping \mathcal{G}_k^{-1} . Our procedure uses formula (7) to construct data set for learning of the network. We use this asymptotic formula to calculate the shape functionals $\mathcal{J}_j(\Omega_k)$, $i = 1, \dots, N$. k is a fixed integer, and $k = 3$ for this part of the paper. In other words we are going to determine the size $\rho_j > 0$ and the location $y_j \in \mathbb{R}^2$ of centers of three holes. The numerical result, however, is the conditional expectation of unknown values.

For the vector of actual observations $\mathcal{J}(\Omega_3) = (\mathcal{J}_1(\Omega_3), \dots, \mathcal{J}_{12}(\Omega_3)) \in \mathbb{R}^{12}$ which describes the unknown properties of imperfections, we want to find $(Y, \rho) = (y_1, \rho_1, y_2, \rho_2, y_3, \rho_3) \in \mathbb{R}^9$ such that the vector $\mathcal{G}_k(Y, \rho) \in \mathbb{R}^{12}$ evaluated from the mathematical model coincides with the given vector of actual observations.

In the learning process [2], for random distribution [1] of (Y, ρ) the observation vectors $\mathcal{G}_k(Y, \rho)$ are computed from the mathematical model of the body $\Omega_3 \subset \mathbb{R}^2$ by an application of the asymptotics obtained for $\rho_j \rightarrow 0$ with $j = 1, 2, 3$. Therefore, a systematic error is introduced to the learning procedure by taking values of topological derivatives in order to compute the approximate values of $\mathcal{G}_k(Y, \rho)$.

We describe briefly the neural networks. Fundamental element of artificial neural networks is an artificial neuron. An artificial neuron has many inputs and a single output. Each of inputs has weight. Output signal is calculated inside neuron based on input information. Each artificial neuron has activation function. Additionally, every neuron has one extra input named bias. This input is always equal to one and has its own weight. Let n be the number of neuron's inputs, and denote input vector by $x = [x_0, x_1, \dots, x_n]$, where $x_0 = 1$, weight vector by $w = [w_0, w_1, \dots, w_n]$, as well as activation function by f . The model of this neuron is described in Fig. 2.

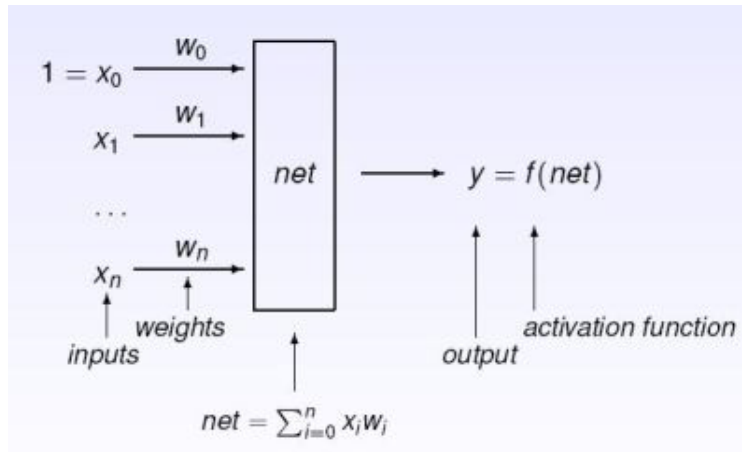


Fig. 2 Model of artificial neuron

A multilayer feedforward neural network is a set of neurons that are divided in separate groups named layers. Input vector represents input vector for all neurons in the first layer. Each connection between input and neuron has its own weight. Each neuron of the first layer generates output signal. Vector of output signals from the first layer represents input vector to the second layer. An example of typical multilayer feedforward network is shown in Fig. 3.

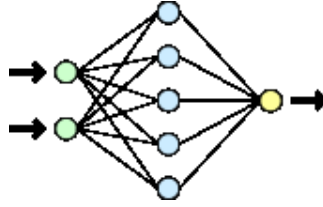


Fig. 3 Model of multilayer feedforward neural network

4.1 Inverse Problem

We consider a particular case of the mapping \mathcal{G} for $k = 3$ and $N = 12$ (see Fig. 4). It means that we consider the domain Ω_3 as a square $(0, 1) \times (0, 1)$ with three holes $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$.

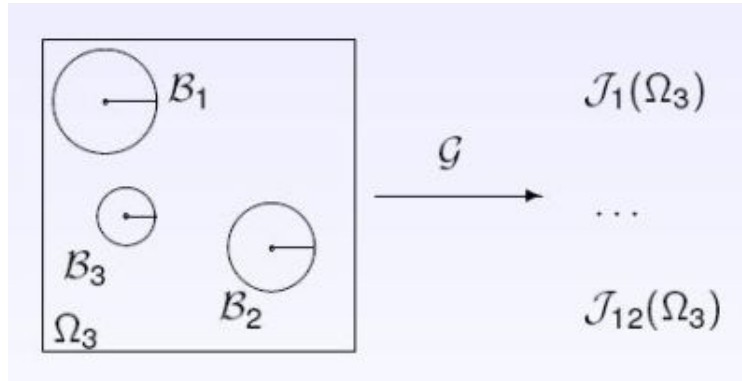


Fig. 4 Mapping \mathcal{G}

Mapping $\mathcal{G} : \mathbb{R}^9 \rightarrow \mathbb{R}^{12}$ is defined by

$$\mathcal{G} : (y_1, \rho_1, y_2, \rho_2, y_3, \rho_3) \rightarrow (\mathcal{J}_1(\Omega_3), \dots, \mathcal{J}_{12}(\Omega_3)), \quad (24)$$

where

(y_1, ρ_1) – center and radius of first hole,

(y_2, ρ_2) – center and radius of second hole,

(y_3, ρ_3) – center and radius of third hole,

$\mathcal{J}_1(\Omega_3), \dots, \mathcal{J}_{12}(\Omega_3)$ – shape functionals.

For domain with three holes mapping (24) calculates vector of shape functionals (Fig. 4).

Let $\mathcal{G}^{-1} : \mathbb{R}^{12} \rightarrow \mathbb{R}^9$ be **the inverse mapping** defined by

$$\mathcal{G}^{-1} : (\mathcal{J}_1(\Omega_3), \dots, \mathcal{J}_{12}(\Omega_3)) \rightarrow (y_1, \rho_1, y_2, \rho_2, y_3, \rho_3). \quad (25)$$

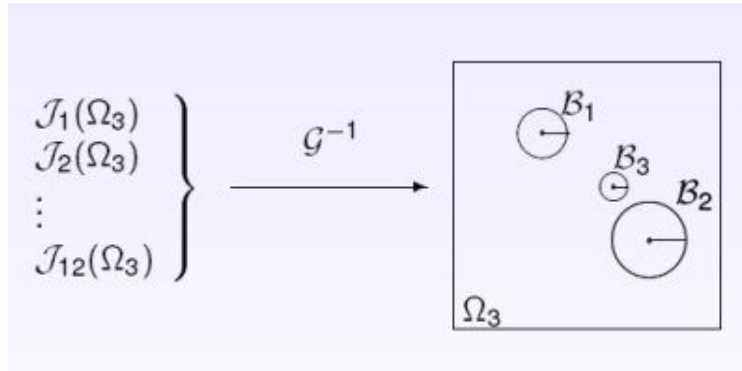


Fig. 5 Mapping \mathcal{G}^{-1}

The inverse mapping \mathcal{G}^{-1} for vector of shape functionals calculates location and size of three holes in our domain (Fig. 5). Any analytical formula for mapping \mathcal{G}^{-1} is unknown. Our aim is construction of an approximation of this mapping using artificial neural networks.

Multilayer feedforward neural network is capable of arbitrarily accurate approximations to arbitrary mapping [12]. To approximate mapping \mathcal{G}^{-1} by neural network we have to construct network and learning set. Appropriate neural network consists of 4 hidden layers. Let q_1, q_2, q_3 and q_4 denote the numbers of neurons in hidden layers. ϕ_1, ϕ_2, ϕ_3 and ϕ_4 denote sigmoidal activation functions for each of hidden layers. Furthermore, our network has one output layer with 9 neurons and linear activation function. Input vector has 12 components and represents the vector of shape functionals. Numbers of neurons in hidden layers are not fixed, and increase to infinity when the size of learning set increases to infinity. Output vector represents vector of data for three holes. The first and the second components represent the center of the first hole, the third entry represents the radius of the first hole. In the same way we denote all entries.

To construct learning set we use probability theory. We consider a sequence $\{Z^m\}$, $m = 1, 2, \dots$ of vectors, where $Z^m = (Y^m, X^m)$. $X^m = (X_1^m, \dots, X_9^m)$ describes the three holes. Its coordinates denote:

- (X_1^m, X_2^m) – the center of first hole,
- X_3^m – the radius of first hole,
- (X_4^m, X_5^m) – the center of second hole,
- X_6^m – the radius of second hole,
- (X_7^m, X_8^m) – the center of third hole,
- X_9^m – the radius of third hole.

We generate vector X^m randomly by choosing three holes in a domain Ω_0 . $Y^m = (Y_1^m, \dots, Y_9^m)$ – describes unknown value of function \mathcal{G}^{-1} . Using notation of previous section we have that $(Y_1^m, Y_2^m) = y_1$, $Y_3^m = \rho_1$, $(Y_4^m, Y_5^m) = y_2$, $Y_6^m = \rho_2$, $(Y_7^m, Y_8^m) = y_3$, $Y_9^m = \rho_3$.

We generate vectors X^m , $m = 1, 2, \dots$, in a random way. For $m \rightarrow \infty$ the whole square $(0, 1) \times (0, 1)$ can be “covered”.

We generate vectors X^m , $m = 1, 2, \dots$, then we calculate, using the direct mapping \mathcal{G} , shape functionals $(\mathcal{J}_1^m, \dots, \mathcal{J}_{12}^m) = \mathcal{G}(X^m)$, $m = 1, 2, \dots$. It is our information about the graph of the inverse mapping \mathcal{G}^{-1} which is used for the construction of the learning set. Values of mapping \mathcal{G}^{-1} are not exact because we have used asymptotic formula (7) so our network approximates only **the conditional expectation** defined by following formula

$$\theta_o(X^m) = E(Y^m | X^m). \quad (26)$$

It is the conditional expectation of Y^m provided that X^m is known. We use an artificial neural network as an approximator of θ_o .

Let M be the size of the learning set. The learning set is composed of the following vectors

$$(\mathcal{J}_1^m, \dots, \mathcal{J}_{12}^m, X_1^m, \dots, X_9^m), \quad m = 1, \dots, M$$

where $(\mathcal{J}_1^m, \dots, \mathcal{J}_{12}^m)$ is the input vector for neural network, and (X_1^m, \dots, X_9^m) is the output vector (required).

Furthermore, we have dependence

$$(\mathcal{J}_1^m, \dots, \mathcal{J}_{12}^m) = \mathcal{G}(X_1^m, \dots, X_9^m), \quad m = 1, \dots, M.$$

To solve our problem we use artificial neural networks. Each of networks can be represented by some mapping $f^q : \mathbb{R}^{12} \rightarrow \mathbb{R}^6$. Parameter q describes number of all neurons in hidden layers. This value depends on the numbers of neurons in all hidden layers. We construct a sequence of networks as a sequence of approximators to θ_o . We let networks where q_1, q_2, q_3, q_4 and M grows [12]. For a given M the learning network provides an approximation to the unknown regression function θ_o .

Set Θ as a function space containing θ . $\theta(\cdot) := f^q(\cdot, \delta^q)$ where δ^q is a set parameters of networks. The function space Θ contains θ_o . We construct a sequence of “sieves” $\{\Theta_M\}$ for $M = 1, 2, \dots$ where Θ_M is a function space containing networks learned by M – *elements* of the learning sets.

The ‘‘connectionist sieve estimator’’ $\tilde{\theta}_M$ is defined as a solution to the least squares problem (appropriate for learning $E(Y^M | X^M)$)

$$\min_{\theta \in \Theta_M} M^{-1} \sum_{m=1}^M [Y^m - \theta(X^m)]^2, \quad (27)$$

for $M = 1, 2, \dots$.

We can apply the following result of White [12] to obtain a convergence result for our method.

Theorem 2. *Under some technical assumptions there exists sieve estimator $\tilde{\theta}_M$ such that*

$$\frac{1}{M} \sum_{m=1}^M [Y^m - \tilde{\theta}_M(X^m)]^2 = \min_{\theta \in \Theta_M} \frac{1}{M} \sum_{m=1}^M [Y^m - \theta(X^m)]^2, \quad (28)$$

for $M = 1, 2, \dots$. Furthermore, $d(\tilde{\theta}_M, \theta_0) \xrightarrow{P} 0$ (i.e., for all $\varepsilon > 0$ $P[\omega \in \text{domain} : d(\tilde{\theta}_M(\omega), \theta_0) > \varepsilon] \rightarrow 0$ as $M \rightarrow \infty$), where d measures the distance between functions and the convergence is in measure.

4.2 Conclusions

In this paper we use artificial neural networks as an approximator of unknown mapping which, for a given vector of shape functionals, calculates locations and radii of three holes. In [3] the authors consider the same problem for one hole. Here, we consider more complicated problem with some additional features. Our learning data set is not exact because asymptotic formula (7) is used to calculate the set. Therefore the obtained network is an approximator of the conditional expectation θ_0 .

Our aim is to determine a solution of problem (27). This problem is defined for exact unknown values Y^m however network is learned based on inexact data. In Theorem 2 we present a result on the existence of a solution $\tilde{\theta}_M$ to problem (27) as well as on the convergence of the method. It follows that $d(\tilde{\theta}_M, \theta_0) \xrightarrow{P} 0$. It means that distance between: $\tilde{\theta}_M$, the solution of problem (27) and θ_0 , the conditional expectation tends to zero in probability measure.

The numerical results presented below show that the method is very efficient and robust, however there are still many open problems in the mathematical analysis of the proposed method.

4.3 Numerical Example for One Hole

We consider an example with one hole instead of three holes. The example with one hole needs less complicated network and less time to perform the learning process

as well as it has the same rules during the creation and learning process. We used Matlab 6.1 to generate learning data, testing data and to create and train neural network. Our network has two hidden layers: first with 24 neurons and second with 12 neurons. In both layers we use sigmoidal activation function. Output layer is composed of three neurons. First and second output are the conditional expectation of location of hole and third output is the radius of hole. Input vector has 12 components and describes 12-element vector of shape functionals. The number of shape functionals is fixed for numerical example (more details [3]). Based on vector of shape functionals network calculates output of network.

We prepared two (different) sets, a learning set and a testing set. The learning set contains 1000 elements, and the testing set contains 10 elements. The number of elements in both sets is fixed for numerical example. The method of constructing learning set and testing set is the same. In the domain Ω_0 we generate randomly distributed hole. We calculate using mapping \mathcal{G} the vector of shape functionals in domain with holes. So we have pair of vectors: the first vector describes holes and the second vector includes the of shape functionals. We repeat this procedure for a finite fixed number of iterations. To train the network we use vectors from the testing set. The vectors of shape functionals are the inputs for network, the vectors which describe holes are the required output from the network.

We assume tolerated learning error: 10^{-5} and number of learning epochs: 700. Consequently, we get following results:

- error on learning data is 0.0083 ($< 1\%$),
- error on testing data is 0.0081 ($< 1\%$).

In the following figure (Fig. 6) we present the result that we obtained for the testing set. We prepared, in the same way as the learning set, the testing set. The testing set contains 10 pairs of vectors: describes holes and vectors of topological derivatives. We put 10 holes in the figure, drawn in bold. These are the real holes from the testing set. Each of the vectors of topological derivatives was put to the neural network. On the output we obtain 10 vectors. Each of them describes one hole. It is the answer of network for the vector of observations. In the figure, 10 holes drawn in solid lines are the outputs of the network. Finally, in the figure we present differences between results for 10 pairs of holes: from testing set (bold) and outputs of network (solid).

References

1. A. R. Baron: Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory* **39** (1993) pp. 930–945
2. M. Hagan, M. Menhaj: Training feedforward networks with the Marquardt algorithm. *IEEE Trans. on Neural Networks* **5**(6) (1994) pp. 989–993
3. L. Jackowska-Strumillo, J. Sokołowski, A. Żochowski, A. Henrot: On Numerical Solution of Shape Inverse Problems. *Computational Optimization and Applications* **23** (2002) pp. 231–255
4. W. G. Mazja, S. A. Nazarov, B. A. Plamenevskii: *Asymptotische Theorie elliptischer Randwertaufgaben in singular gestörten Gebieten*. vol. 1, Berlin: Akademie Verlag (1991) p. 432

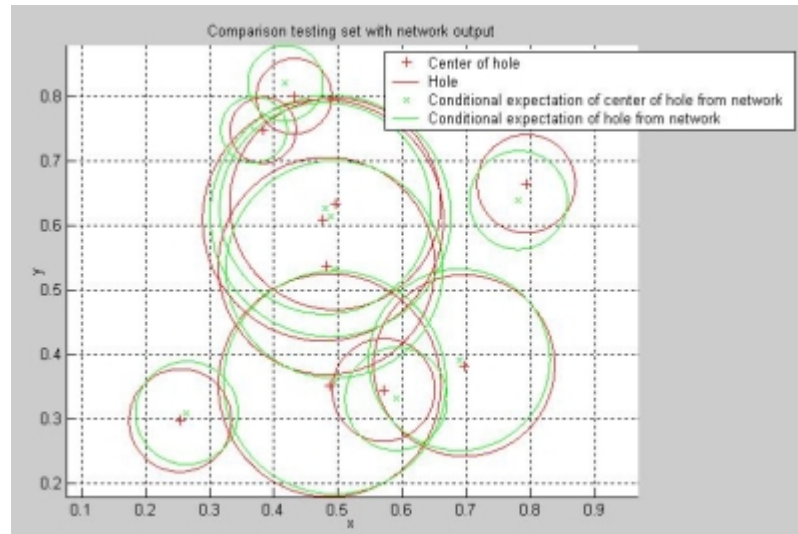


Fig. 6 Numerical result

(English transl.: *Asymptotic Theory of Elliptic Boundary Value Problems in Singularly Perturbed Domains*, vol. 1, Basel: Birkhäuser Verlag (2000) p. 435)

5. S. A. Nazarov, B. A. Plamenevsky: Elliptic Problems in Domains with Piecewise Smooth Boundaries. *De Gruyter Exposition in Mathematics* **13** (1994) Walter de Gruyter
6. S. A. Nazarov, J. Sokołowski: Spectral problems in the shape optimisation. Singular boundary perturbations. *Asymptotic Analysis* **56**(3-4) (2008) pp. 159–204
7. J. R. Roche, J. Sokołowski: Numerical methods for shape identification problems. *Special issue of Control and Cybernetics: Shape Optimization and Scientific Computations* **5** (1996) pp. 867-894
8. A. Shumacher: *Topologieoptimierung von Bauteilstrukturen unter Verwendung von Lochpositionierungskriterien*. Ph.D. Thesis, Universität-Gesamthochschule-Siegen, Siegen (1995)
9. J. Sokołowski, J.-P. Zolesio: *Introduction to Shape Optimization. Shape Sensitivity Analysis*. Springer Verlag, Berlin (1992)
10. J. Sokołowski, A. Żochowski: Modeling of topological derivatives for contact problems. *Numerische Mathematik* **102**(1) (2005) pp. 145–179
11. J. Sokołowski, A. Żochowski: On topological derivative in shape optimization. *SIAM Journal on Control and Optimization* **37**(4) (1999) pp. 1251–1272
12. H. White: Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings. *Neural Networks* **3** (1990) pp. 535–549