



Research article

Application of neural networks to inverse elastic scattering problems with near-field measurements

Yao Sun*, Lijuan He and Bo Chen*

College of Science, Civil Aviation University of China, Tianjin, China

* **Correspondence:** Email: sunyao10@mails.jlu.edu.cn, charliecb@163.com.

Abstract: This paper is concerned with the application of a machine learning approach to inverse elastic scattering problems via neural networks. In the forward problem, the displacements are approximated by linear combinations of the fundamental tensors of the Cauchy-Navier equations of elasticity, which are expressed in terms of sources placed inside the elastic solid. From the near-field measurement data, a two-layer neural network method consisting of a gated recurrent unit to gate recurrent unit has been used to reconstruct the shape of an unknown elastic body. Moreover, the convergence of the method is proved. Finally, the feasibility and effectiveness of the presented method are examined through numerical examples.

Keywords: partial differential equations; inverse scattering problem; linear elasticity

1. Introduction

Direct and inverse time-harmonic elastic wave scattering by a bounded body [1] has a wide range of applications, such as radar/sonar, geophysical exploration, medical imaging, seismology and materials sciences [2–5]. The forward problem is to determine and approximate the diffracted field given an incident wave and the elastic body. We refer the reader to, e.g., [6, 7] for the existence and uniqueness of forward solutions via the variational approach. The inverse scattering problem (see e.g., [8–10]) is to determine the physical or geometric information of the scatterer based on knowledge of the scattered field. This paper is concerned with both direct and inverse problems.

In recent years, numerical approaches for position and shape reconstruction in many elastic scattering problems have drawn the attention of a large number of scholars. In [11], Arens proved that in two dimensions the shape of an elastically rigid body can be determined from the information of compressional and shear far-field patterns for all incident elastic plane waves with distinct directions at a fixed frequency. In [7], Bao et al. investigated the direct and inverse problems for a penetrable anisotropic elastic body embedded in a homogeneous and isotropic background medium. They show the unique-

ness and existence of weak solutions by using the Fredholm alternative together with properties of the Dirichlet-to-Neumann map in both two and three dimensions. Moreover, an iterative method was designed to recover the interface from the multi-frequency near-field data. Charalambopoulos et al. [12] evaluated the inverse elastic scattering problem by using penetrable isotropic bodies embedded in an isotropic host environment via the factorization method. Hu et al. [13] investigated some uniqueness results for a ball or a convex Lipschitz polyhedron and adapted the factorization method to recover the shape of a rigid body from the scattered S-waves or P-waves corresponding to all incident plane shear or pressure waves. Recently, a reverse time migration method was proposed by Chen and Huang [14]. In [15], Li et al. presented an iterative method to solve the inverse elastic scattering problem. The domain derivative expresses the rate of change with respect to the surface displacement. A continuation method using multiple frequency data was established for the inverse problem.

The method of fundamental solutions (MFS) is an effective method for solving both direct and inverse problems governed by partial differential equations (PDEs). Since it is very simple and easy to program, it has been attracting an increasing amount of attention in recent years. Various engineering problems, including the acoustics scattering problems [16, 17] and the acoustics-elastic scattering problem [18], can be solved by the MFS. The MFS can also be used to solve the inverse acoustic scattering problems (see e.g., [19–21]). Alves et al. [22] used the MFS to solve non-homogeneous elastic wave equations.

In recent years, neural networks, one of the most popular schemes in machine learning, have been applied in many research studies, including inverse scattering problems [23, 24]. The seq2seq neural network has been applied for natural language processing and machine translation in the literature [25, 26]. In [27], a new SwitchNet with a sparse neural network structure was proposed to solve the wave equation-based backscattering problems. Neural networks are especially suitable for dealing with inaccurate and vague information processing problems involving many parameters and constraints, and they can fully approximate complex nonlinear maps. Yin et al. [23] proposed a two-layer neural network method to recover the location and shape of an obstacle by using phaseless far-field data.

In this paper, we consider inverse elastic scattering problems related to recovering the shape of a rigid body and a cavity by applying a two-layer gated controlled recurrent unit (GRU) neural network. The convergence of the presented method will be verified. The contributions of this paper are summarized as follows.

- (1) We propose a fast numerical scheme for solving the direct problem;
- (2) We prove the convergence of a two-layer GRU neural network;
- (3) We test the accuracy of the reconstruction method by numerical examples.

The rest of the paper is arranged as follows. In Section 2, we formulate the elastic scattering problem and describe the numerical scheme for forward solutions. In Section 3 we introduce the architecture of the neural network. In Section 4, we perform theoretical analysis for the convergence of the network. In Section 5, numerical experiments are presented to verify the efficiency and effectiveness of the neural network.

2. Formulation and solution method

Let $D \subset \Omega$ be a two-dimensional elastic body with the smooth boundary ∂D , where $\Omega \subset \mathbb{R}^2$ is a subdomain. The exterior domain $D^c := \mathbb{R}^2 \setminus \overline{D}$ is supposed to be connected and filled with a

homogeneous and isotropic elastic medium. The displacement $\mathbf{u}(\mathbf{x}, t)$ satisfies

$$\mu \Delta \mathbf{u}(\mathbf{x}, t) + (\lambda + \mu) \nabla \nabla \cdot \mathbf{u}(\mathbf{x}, t) - \rho \frac{\partial^2 \mathbf{u}(\mathbf{x}, t)}{\partial t^2} = 0.$$

In general, the displacement $\mathbf{u}(\mathbf{x}, t)$ is time harmonic and given by $\mathbf{u}(\mathbf{x}, t) = e^{i\omega t} \mathbf{v}(\mathbf{x})$ (where i is the imaginary unit); thus, the displacement $\mathbf{v}(\mathbf{x})$ satisfies the Navier equation

$$\mu \Delta \mathbf{v} + (\lambda + \mu) \nabla \nabla \cdot \mathbf{v} + \rho \omega^2 \mathbf{v} = 0 \quad \text{in } \mathbb{R}^2 \setminus \overline{D}, \quad (2.1)$$

where $\omega > 0$ is the angular frequency, λ, μ are the Lamé constants satisfying that $\mu > 0$ and $\lambda + \mu > 0$, and ρ is the density function, which is assumed to be constant.

The forward problem considered in this paper is to give an incident elastic plane wave $\mathbf{u}^{incident} = \mathbf{d} e^{ik_p \mathbf{d} \cdot \mathbf{x}} + \mathbf{d}^\perp e^{ik_s \mathbf{d} \cdot \mathbf{x}}$ with $\mathbf{d} = (\cos \vartheta, \sin \vartheta)$ and $\mathbf{d}^\perp = (-\sin \vartheta, \cos \vartheta)$, $\vartheta \in [0, 2\pi)$, as well as to find the scattered field $\mathbf{v} = (v_1, v_2)^\top$ that satisfies the Navier equation.

By the well-known Hodge decomposition, any solution \mathbf{v} to (2.1) can be decomposed into

$$\mathbf{v} = \mathbf{v}_p + \mathbf{v}_s,$$

where $\mathbf{v}_p := (-1/k_p^2) \text{grad div } \mathbf{v}$ is known as the compressional part of \mathbf{v} and $\mathbf{v}_s := (-1/k_s^2) \text{grad}^\perp \text{div}^\perp \mathbf{v}$, with $\text{grad}^\perp = (-\partial_2, \partial_1)^\perp$ and $\text{div}^\perp \mathbf{v} = -\partial_2 v_1 + \partial_1 v_2$ as the shear part. Here, $k_s = \omega \sqrt{\frac{\rho}{\mu}}$ is the shear wave number and $k_p = \omega \sqrt{\frac{\rho}{\lambda + 2\mu}}$ is the compressional wave number.

For a rigid body D , the scattered field \mathbf{v} satisfies the first (Dirichlet) kind of boundary condition, i.e., it holds that

$$\mathbf{v} = -\mathbf{u}^{incident} \quad \text{on } \partial D \quad (2.2)$$

so that the total displacement is zero.

If D is a cavity, the following second (Neumann) kind of boundary condition holds:

$$\mathbf{t}_n \mathbf{v} = -\mathbf{t} \quad \text{on } \partial D, \quad (2.3)$$

with $\mathbf{t} := \mathbf{t}_n \mathbf{u}^{incident}$. Here, the traction operator \mathbf{t}_n is defined by

$$\begin{aligned} (\mathbf{t}_n \mathbf{v})(\mathbf{x}) &= \left\{ \left[(\lambda + 2\mu) \frac{\partial v_1}{\partial x_1} + \lambda \frac{\partial v_2}{\partial x_2} \right] n_1 + \mu \left[\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} \right] n_2 \right\} \hat{\mathbf{e}}_1 \\ &+ \left\{ \mu \left[\frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2} \right] n_1 + \left[\lambda \frac{\partial v_1}{\partial x_1} + (\lambda + 2\mu) \frac{\partial v_2}{\partial x_2} \right] n_2 \right\} \hat{\mathbf{e}}_2, \end{aligned} \quad (2.4)$$

where $\hat{\mathbf{e}}_i$, $i = 1, 2$, designates the unit vector along the x_i axis and $\mathbf{n} = (n_1, n_2)$ denotes the unit normal direction point into D^c .

To ensure uniqueness, we require the scattered field \mathbf{v} to satisfy the two-dimensional Kupradze-Sommerfeld radiation condition at infinity, that is,

$$\lim_{r \rightarrow \infty} r^{1/2} (\partial_r \mathbf{v}_p - ik_p \mathbf{v}_p) = 0, \quad \lim_{r \rightarrow \infty} r^{1/2} (\partial_r \mathbf{v}_s - ik_s \mathbf{v}_s) = 0, \quad r := |\mathbf{x}|.$$

In this paper, we will employ the MFS (see [28] for the acoustic scattering problem) to calculate the forward solution \mathbf{v} . The main idea of the MFS is to approximate the scattered field in the solution

domain D^c by using a linear combination of fundamental solutions [18,29] to the Navier equation (2.1) with N source points $\mathbf{y}_j \in D$, i.e.,

$$\mathbf{v}(\mathbf{x}) \approx \mathbf{v}_N(\mathbf{x}) = \sum_{j=1}^N \mathbf{U}(\mathbf{x}, \mathbf{y}_j) \begin{bmatrix} a_j \\ b_j \end{bmatrix}, \quad \mathbf{x} \in D^c. \quad (2.5)$$

Here, a_j, b_j are the coefficients to be determined by the boundary condition on ∂D and $\mathbf{U}(\mathbf{x}, \mathbf{y})$ is Green's tensor of the two-dimensional Lamé system of the form

$$\mathbf{U}(\mathbf{x}, \mathbf{y}) = \frac{\Phi(k_s|\mathbf{x} - \mathbf{y}|)}{\mu} \mathbf{I} + \frac{1}{\rho\omega^2} \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} & \frac{\partial^2}{\partial x_1 \partial x_2} \\ \frac{\partial^2}{\partial x_2 \partial x_1} & \frac{\partial^2}{\partial x_2^2} \end{pmatrix} \left(\Phi(k_s|\mathbf{x} - \mathbf{y}|) - \Phi(k_p|\mathbf{x} - \mathbf{y}|) \right),$$

where \mathbf{I} is the 2-by-2 identity matrix and

$$\Phi(k|\mathbf{x} - \mathbf{y}|) = -\frac{i}{4} H_0^{(1)}(k|\mathbf{x} - \mathbf{y}|), \quad k = k_p, k_s$$

is the fundamental solution of the Helmholtz equation associated with the wavenumber k . Here, $H_0^{(1)}$ is the Hankel function of the first kind of order zero.

Remark 2.1. Following the idea in [29], the solution $\mathbf{v}(\mathbf{x})$ of this problem can be approximated by the single-layer potential function, with $\psi \in (L^2(\partial B))^2$, as follows:

$$S\psi(\mathbf{x}) = \int_{\partial B} \mathbf{U}(\mathbf{x}, \mathbf{y}) \psi(\mathbf{y}) ds_{\mathbf{y}}, \quad \mathbf{x} \in D^c, \quad (2.6)$$

where $\bar{B} \subset D$. If we discretize the integral operator S , we can get a linear combination of the fundamental solutions with respect to some source points located on ∂B with the unknown density function $\psi(\mathbf{y}_j)$. It can be seen that (2.5) is a certain discrete form of (2.6) with some weight coefficients.

We respectively define the compressional and shear parts of $\mathbf{U}(\mathbf{x}, \mathbf{y})$ as follows:

$$\mathbf{U}_p(\mathbf{x}, \mathbf{y}) = -\frac{1}{\rho\omega^2} \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} & \frac{\partial^2}{\partial x_1 \partial x_2} \\ \frac{\partial^2}{\partial x_2 \partial x_1} & \frac{\partial^2}{\partial x_2^2} \end{pmatrix} \Phi(k_p|\mathbf{x} - \mathbf{y}|)$$

and

$$\mathbf{U}_s(\mathbf{x}, \mathbf{y}) = \frac{\Phi(k_s|\mathbf{x} - \mathbf{y}|)}{\mu} \mathbf{I} + \frac{1}{\rho\omega^2} \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} & \frac{\partial^2}{\partial x_1 \partial x_2} \\ \frac{\partial^2}{\partial x_2 \partial x_1} & \frac{\partial^2}{\partial x_2^2} \end{pmatrix} \Phi(k_s|\mathbf{x} - \mathbf{y}|).$$

Then $\mathbf{U}(\mathbf{x}, \mathbf{y}) = \mathbf{U}_p(\mathbf{x}, \mathbf{y}) + \mathbf{U}_s(\mathbf{x}, \mathbf{y})$. Thus we have

$$\mathbf{v}_p(\mathbf{x}) = \sum_{j=1}^N \mathbf{U}_p(\mathbf{x}, \mathbf{y}_j) \begin{bmatrix} a_j \\ b_j \end{bmatrix}, \quad \mathbf{x} \in D^c, \quad (2.7)$$

and

$$\mathbf{v}_s(\mathbf{x}) = \sum_{j=1}^N \mathbf{U}_s(\mathbf{x}, \mathbf{y}_j) \begin{bmatrix} a_j \\ b_j \end{bmatrix}, \quad \mathbf{x} \in D^c. \quad (2.8)$$

Recalling the definitions of the traction vector (2.4) and the fundamental solution to the Navier equation, one can approximate the traction vector on the boundary ∂D as follows:

$$\mathbf{t}_n \mathbf{v}(\mathbf{x}) \approx \mathbf{t}_n \mathbf{v}_N(\mathbf{x}) = \sum_{j=1}^N \mathbf{T}(\mathbf{x}, \mathbf{y}_j) \begin{bmatrix} a_j \\ b_j \end{bmatrix}. \quad (2.9)$$

Here, the matrix $\mathbf{T}(\mathbf{x}, \mathbf{y}_j)$ is defined by

$$\mathbf{T}(\mathbf{x}, \mathbf{y}_j) = \begin{pmatrix} \mathbf{T}_{11}(\mathbf{x}, \mathbf{y}_j) & \mathbf{T}_{12}(\mathbf{x}, \mathbf{y}_j) \\ \mathbf{T}_{21}(\mathbf{x}, \mathbf{y}_j) & \mathbf{T}_{22}(\mathbf{x}, \mathbf{y}_j) \end{pmatrix}$$

with the entries

$$\mathbf{T}_{11}(\mathbf{x}, \mathbf{y}_j) = \left[(\lambda + 2\mu) \frac{\partial \mathbf{U}_{11}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} + \lambda \frac{\partial \mathbf{U}_{21}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} \right] n_1 + \mu \left(\frac{\partial \mathbf{U}_{11}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} + \frac{\partial \mathbf{U}_{21}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} \right) n_2,$$

$$\mathbf{T}_{12}(\mathbf{x}, \mathbf{y}_j) = \left[(\lambda + 2\mu) \frac{\partial \mathbf{U}_{12}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} + \lambda \frac{\partial \mathbf{U}_{22}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} \right] n_1 + \mu \left(\frac{\partial \mathbf{U}_{12}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} + \frac{\partial \mathbf{U}_{22}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} \right) n_2,$$

$$\mathbf{T}_{21}(\mathbf{x}, \mathbf{y}_j) = \mu \left(\frac{\partial \mathbf{U}_{11}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} + \frac{\partial \mathbf{U}_{21}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} \right) n_1 + \left[\lambda \frac{\partial \mathbf{U}_{11}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} + (\lambda + 2\mu) \frac{\partial \mathbf{U}_{21}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} \right] n_2,$$

$$\mathbf{T}_{22}(\mathbf{x}, \mathbf{y}_j) = \mu \left(\frac{\partial \mathbf{U}_{12}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} + \frac{\partial \mathbf{U}_{22}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} \right) n_1 + \left[\lambda \frac{\partial \mathbf{U}_{12}(\mathbf{x}, \mathbf{y}_j)}{\partial x_1} + (\lambda + 2\mu) \frac{\partial \mathbf{U}_{22}(\mathbf{x}, \mathbf{y}_j)}{\partial x_2} \right] n_2.$$

To determine the coefficients $a_j, b_j, j = 1, 2, \dots, N$, we employ the collocation method by choosing M collocation points $\{\mathbf{x}_j\}_{j=1}^M \subset \partial D$ and impose the boundary condition on ∂D . Set $\mathbf{d}' := (a_1, \dots, a_N, b_1, \dots, b_N)^\top$ to be the vector of $2N$ unknown parameters. It can be calculated by solving a linear system with $2M$ equations of the form

$$\mathcal{A} \mathbf{d}' = \mathbf{h}', \quad (2.10)$$

where the matrix $\mathcal{A} \in \mathbb{C}^{2M \times 2N}$ on the left-hand side and the vector $\mathbf{h}' \in \mathbb{C}^{2M \times 1}$ on the right-hand side are defined as follows depending on the boundary condition on ∂D .

- Dirichlet boundary condition (rigid body)

$$\mathcal{A}_{i,j} = \mathbf{U}_{11}(\mathbf{x}_i, \mathbf{y}_j), \mathcal{A}_{i,j+N} = \mathbf{U}_{12}(\mathbf{x}_i, \mathbf{y}_j), \mathcal{A}_{i+M,j} = \mathbf{U}_{21}(\mathbf{x}_i, \mathbf{y}_j), \mathcal{A}_{i+M,j+N} = \mathbf{U}_{22}(\mathbf{x}_i, \mathbf{y}_j),$$

$$\mathbf{h}'_i = -u_1^{\text{incident}}(\mathbf{x}_i), \mathbf{h}'_{i+M} = -u_2^{\text{incident}}(\mathbf{x}_i), i = 1, \dots, M, \quad j = 1, \dots, N.$$

- Neumann boundary condition (cavity)

$$\mathcal{A}_{i,j} = \mathbf{T}_{11}(\mathbf{x}_i, \mathbf{y}_j), \mathcal{A}_{i,j+N} = \mathbf{T}_{12}(\mathbf{x}_i, \mathbf{y}_j), \mathcal{A}_{i+M,j} = \mathbf{T}_{21}(\mathbf{x}_i, \mathbf{y}_j), \mathcal{A}_{i+M,j+N} = \mathbf{T}_{22}(\mathbf{x}_i, \mathbf{y}_j),$$

$$\mathbf{h}'_i = -t_1(\mathbf{x}_i), \mathbf{h}'_{i+M} = -t_2(\mathbf{x}_i), i = 1, \dots, M, \quad j = 1, \dots, N.$$

In our numerics, we choose the number $M = N$ in all examples. The algebraic linear system (2.10) is solved directly by using Matlab with the command $\mathbf{d}' = \mathcal{A}\mathbf{h}'$. Then we can get the scattered near-field data \mathbf{v} on $\partial\Omega$ from (2.5). One can also get the shear part $\mathbf{v}_s|_{\partial\Omega}$ and compressional part $\mathbf{v}_p|_{\partial\Omega}$ of \mathbf{v} by replacing Green's tensor \mathbf{U} by its shear and compressional parts in (2.7) and (2.8), respectively. These synthetic data will be used as the measurement data in our inverse problem. In our examples below, we suppose, for simplicity that the boundary $\partial\Omega$ can be parameterized as follows:

$$\partial\Omega = \{c(\theta) = R(\cos \theta, \sin \theta) + (a, b) : \theta \in (0, 2\pi]\}, \quad R, a, b \in \mathbb{R}.$$

This means that the measurement data are taken on the circle centered at $(a, b) \in \mathbb{R}^2$ with the radius $R > 0$.

Definition 2.1. Let $\hat{\mathbf{d}}_m := R(\cos \theta_m, \sin \theta_m) + (a, b)$, $m = 1, 2, \dots, \mathcal{N}_1$; indicate the measurement positions where $\mathcal{N}_1 \in \mathbb{N}$ and $\theta_m \in (0, 2\pi]$. Given the obstacle D and the compressional and shear wave-numbers k_p, k_s , respectively, we define the measurement set $\mathbf{I} = (I(1), I(2), \dots, I(T))$ with $T = 2\mathcal{N}_1$ as follows:

$$I(n) = \mathbf{v}_p(\hat{\mathbf{d}}_n, D), \quad I(n + \mathcal{N}_1) = \mathbf{v}_s(\hat{\mathbf{d}}_n, D), \quad n = 1, 2, \dots, \mathcal{N}_1.$$

The inverse problem in this paper is to recover the shape of the obstacle or the cavity, namely ∂D , by applying knowledge of the near-field data on $\partial\Omega$ associated with a fixed incident field:

$$(\mathbf{v}_p(\hat{\mathbf{d}}_n, \partial\Omega), \mathbf{v}_s(\hat{\mathbf{d}}_n, \partial\Omega)) \longrightarrow \partial D.$$

Assumption 2.1. The boundary of the obstacle D is assumed to take the form

$$\partial D : \mathbf{x}(\theta) = p(\theta) (\cos \theta, \sin \theta), \quad 0 \leq \theta \leq 2\pi,$$

where $p(\theta)$ permits the following (truncated) Fourier representation:

$$p(\theta) = \frac{a_0}{2} + \sum_{m=1}^Q a_m \cos(m\theta) + \sum_{m=1}^Q b_m \sin(m\theta), \quad (2.11)$$

where $Q \in \mathbb{N}$. Below we define the set of the Fourier coefficients a_0, a_m and b_m , $m = 1, 2, \dots, Q$, by $\mathbf{y} = \{y(1), y(2), \dots, y(N)\}$ with $N = 2Q + 1$. In the remaining part of this paper, we shall consider the inverse problem of recovering \mathbf{y} from our measurement data set.

3. Architecture of neural networks

In this section, we introduce the main components of the neural network first. Then how to construct the neural network to deal with the inverse elastic scattering problem is given.

Figure 1 gives a simple map of the exhibition of its working status. We introduce the GRU cells (see e.g., [23, 30]), i.e., the principal element of the update network. This network consists of many GRU neurons.

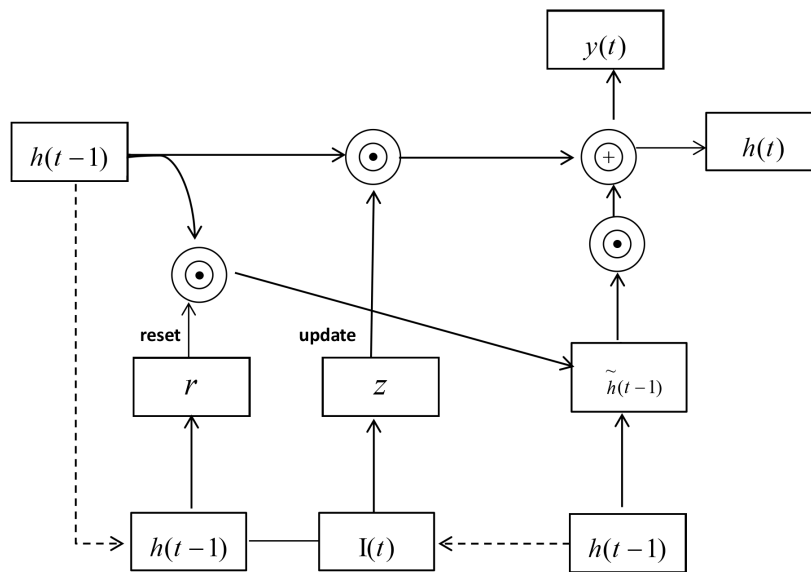


Figure 1. Internal structure of GRU neuron.

- (1) The cell receives a stimulus, i.e., an input $I(t)$ ($t = 1, 2, \dots, T$). Then the cell will respond to the stimulus. We describe this response as an activation function acting on the previous output $h(t-1)$ and the present stimulus $I(t)$ to produce a reset gate $r(t)$, which is to control candidate status $\tilde{h}(t)$ based on whether it depends on the prior status of learning.

$$r(t) = \sigma(\mathbf{w}^r \cdot [I(t), h(t-1)]) \quad (3.1)$$

$$\tilde{h}(t-1) = h(t-1) \odot r(t) \quad (3.2)$$

$$\tilde{h}(t) = \tanh(\mathbf{w}^{\tilde{h}} \cdot [I(t), \tilde{h}(t-1)]) \quad (3.3)$$

- (2) It preserves historical output $h(t-1)$ and presents input $I(t)$ to obtain update gate $z(t)$, which can be used to choose and forget memory. In other words, it can forget some information passed down in $h(t-1)$ and select and memorize the information in $\tilde{h}(t)$.

$$z(t) = \sigma(\mathbf{w}^z \cdot [I(t), h(t-1)]) \quad (3.4)$$

$$h(t) = (1 - z(t)) \odot h(t-1) \oplus z(t) \odot \tilde{h}(t). \quad (3.5)$$

In order to see the notations clearly, we list the notations used in the neural network in Table 1.

3.1. Construction of the network

The whole model consists of three parts, namely, the encoding end, the middle status and the decoding end. The role of the encoding end is to extract data features, and the intermediate state serves to connect the encoding and decoding ends. The function of the decoder is to decode the information obtained by the encoder to obtain the output sequence. Next, the relevant details of the network structure are illustrated as below.

Table 1. The notations used in GRU neurons.

Notations	
$r(t)$	The reset gate
$z(t)$	The update gate
w^r	The reset gate weight
$w^{\tilde{h}}$	The candidate status weight
w^z	The update gate weight
$\sigma(x) = \frac{1}{1+e^{-x}}$	A sigmoid activation function
$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	The hyperbolic tangent activation function
$h(t)$	The GRU output
\odot	The tensor product of matrices
\oplus	Plus

Step 1: The scattered data features are extracted by using GRU neurons on the encoding end, i.e.,

$$h^1(t) := \zeta_t(I(t) \ h^1(t-1)), \quad t = 1, 2, \dots, T,$$

where ζ_t signifies the t^{th} GRU cells in the course of the work. That is, it is equivalent to extracting the input data features by using (3.1)–(3.5). Then the first layer extracts the scattered data as $\mathbf{h}^1 = (h^1(1), h^1(2), \dots, h^1(T))$.

Step 2: Extracting features again.

Take the features extracted for the first time, i.e., $\mathbf{h}^1 = (h^1(1), h^1(2), \dots, h^1(T))$ as input to extract the features again by means of (3.1)–(3.5):

$$h^2(t) := \zeta_t(h^1(t) \ h^2(t-1)), \quad t = 1, 2, \dots, T.$$

Step 3: First decoding process on the decoding terminal.

The decoder side will use N GRU neurons to decode the second layer output information $h^2(T)$ and the obstacle boundary Fourier expansion coefficients $\mathbf{y} = (y(1), y(2), \dots, y(N))$:

$$h^3(1) = \zeta_1(y(1) \ h^2(T)),$$

$$h^3(n) = \zeta_n(y(n) \ h^3(n-1)), \quad n = 2, \dots, N,$$

where ζ_n signifies operation of the n^{th} GRU. It plays the same role as ζ_t in Step 1.

Step 4: Second decoding process at the decoder end.

Get the output \mathbf{h}^4 for the fourth layer by decoding the output $\mathbf{h}^3 = (h^3(1), h^3(2), \dots, h^3(N))$ of the third layer. That is, this layer will preserve profitable information and discard unprofitable information

$$h^4(1) = \zeta_1(h^3(1) \ h^2(0)),$$

$$h^4(n) := \zeta_n(h^3(n) \ h^4(n-1)), \quad n = 2, \dots, N.$$

Finally, the latest output from the decoding side will be converted into a predictive output by the fully connected layer. That is to say, we will calculate $\hat{\mathbf{y}} = (\hat{y}(1), \hat{y}(2), \dots, \hat{y}(N))$, i.e., the predicted output of the network by using the following formula

$$\hat{y}(n) = \mathbf{w}_o \cdot h^4(n).$$

4. Theoretical analysis of convergence

This section is devoted to the convergence analysis of the neural network.

Definition 4.1. Let $\{y(n)\}_{n=1}^N$ be the initial values of the boundary curve and $\{\hat{y}(n)\}_{n=1}^N$ be the network prediction output. Define the common mean absolute value loss (error) function as

$$L(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N |y(n) - \hat{y}(n)|. \quad (4.1)$$

When $y(n) - \hat{y}(n) \neq 0$, the derivative of the loss function (4.1) with respect to the weight \mathbf{w} in (3.1)–(3.5) is given by

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = -\frac{1}{N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial \mathbf{w}} \right)^\top. \quad (4.2)$$

For the sake of convenience, we introduce some symbols. The input weights and the internal weights of every layer are gathered in the weight matrices \mathbf{W}_I and \mathbf{W}_L . We denote $\mathbf{W} = (\mathbf{W}_I, \mathbf{W}_L)$. The specific forms are as follows

$$\mathbf{W}_I = (\mathbf{W}_I^1, \mathbf{W}_I^2, \mathbf{W}_I^3, \mathbf{W}_I^4),$$

$$\mathbf{W}_L = (\mathbf{W}_L^1, \mathbf{W}_L^2, \mathbf{W}_L^3, \mathbf{W}_L^4)^\top = \begin{pmatrix} w_1^r & w_1^{\tilde{h}} & w_1^z \\ w_2^r & w_2^{\tilde{h}} & w_2^z \\ w_3^r & w_3^{\tilde{h}} & w_3^z \\ w_4^r & w_4^{\tilde{h}} & w_4^z \end{pmatrix}.$$

Here, \mathbf{W}_I has the same form as \mathbf{W}_L .

Next, we write the input of every layer as

$$\mathbf{u}^1(t) = \begin{pmatrix} I(t) \\ h^1(t-1) \end{pmatrix}, \quad \mathbf{u}^2(t) = \begin{pmatrix} h^1(t) \\ h^2(t-1) \end{pmatrix},$$

$$\mathbf{u}^3(n) = \begin{pmatrix} y(n) \\ h^3(n-1) \end{pmatrix}, \quad \mathbf{u}^4(n) = \begin{pmatrix} h^3(n) \\ h^4(n-1) \end{pmatrix}.$$

The input vectors for each layer of the encoding and decoding ends are given as follows:

$$\begin{pmatrix} \mathbf{v}^1(t) \\ \mathbf{v}^2(t) \\ \mathbf{v}^3(n) \\ \mathbf{v}^4(n) \end{pmatrix} = \begin{pmatrix} \mathbf{W}_I^1 \mathbf{u}^1(t) \\ \mathbf{W}_I^2 \mathbf{u}^2(t) \\ \mathbf{W}_I^3 \mathbf{u}^3(n) \\ \mathbf{W}_I^4 \mathbf{u}^4(n) \end{pmatrix} = \begin{pmatrix} \mathbf{W}_I^1 I(t) + \mathbf{W}_L^1 h^1(t-1) \\ \mathbf{W}_I^2 h^1(t) + \mathbf{W}_L^2 h^2(t-1) \\ \mathbf{W}_I^3 y(n) + \mathbf{W}_L^3 h^3(t-1) \\ \mathbf{W}_I^4 h^3(n) + \mathbf{W}_L^4 h^4(n-1) \end{pmatrix}.$$

Definition 4.2. Given a vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$, the activation function acts on a vector as follows:

$$\sigma(\boldsymbol{\alpha}) = (\sigma(\alpha_1), \sigma(\alpha_2), \dots, \sigma(\alpha_n)).$$

The derivatives of the activation function acts similarly

$$\sigma'(\boldsymbol{\alpha}) = (\sigma'(\alpha_1), \sigma'(\alpha_2), \dots, \sigma'(\alpha_n)),$$

$$\sigma''(\boldsymbol{\alpha}) = (\sigma''(\alpha_1), \sigma''(\alpha_2), \dots, \sigma''(\alpha_n)).$$

In other words, the activation function acts on a vector componentwise.

The weight gradient of the fourth-layer neural network update gate can be calculated directly:

$$\begin{aligned} \frac{\partial \hat{y}(n)}{\partial w_4^z} &= \frac{\partial \hat{y}(n)}{\partial h^4(n)} \cdot ((1 - z_4'(n)) \odot h^4(n-1) \oplus z_4'(n) \odot \tilde{h}(n)) \sigma'(v^4(n)) \\ &\times \left(\mathbf{u}^4(n) + \mathbf{W}_L^4 \frac{\partial h^4(n-1)}{\partial w_4^z} \right). \end{aligned} \quad (4.3)$$

From the initial condition constant $\hat{y}(0) \equiv 0$, we have

$$\frac{\partial \hat{y}(0)}{\partial \mathbf{w}} = 0. \quad (4.4)$$

For any given initial weight \mathbf{w}^0 , the weight sequence $\{\mathbf{w}^k\}$ can be generated from an iterative formula:

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \Delta \mathbf{w}^k, \quad (4.5)$$

where

$$\Delta \mathbf{w}^k = -\gamma \frac{\partial L(\mathbf{w}^k)}{\partial \mathbf{w}} \quad (4.6)$$

and γ is the learning rate.

Assumption 4.1. ([31, 32]) $|\sigma(a)|, |\sigma'(a)|, |\sigma''(a)|$ and $|\tanh(a)|, |\tanh'(a)|, |\tanh''(a)|$ are uniformly bounded for $a \in \mathbb{R}$.

Assumption 4.2. ([31, 32]) $\|\mathbf{W}_L^k\|_{(k=1,2,\dots)}$ is bounded in the learning course described by (4.5).

In order to check the convergence of the neural network, we claim the following theorem.

Theorem 4.1. *The loss function is defined as the formula (4.1), and the weight sequence $\{\mathbf{w}^k\}$ can be generated from the iterative formula (4.5). If Assumptions 4.1 and 4.2 are valid, then we have the following:*

- (1) $L(\mathbf{w}^{k+1}) \leq L(\mathbf{w}^k)$;
- (2) $\exists L^* \geq 0$ s.t. $\lim_{k \rightarrow \infty} L(\mathbf{w}^k) = L^*$.

Proof. First, we can also use the method employed in [32], which adopts the induction argument to calculate $\Delta v^4(\mathbf{w}^k, n)$ as follows:

$$\Delta v^4(\mathbf{w}^k, n) = \left(\mathbf{u}^4(n) + \mathbf{W}_L^4 \frac{\partial h^4(n-1)}{\partial w_4^z} \right) \Delta \mathbf{w}^k + \delta(\mathbf{w}^k, n), \quad (4.7)$$

$$\|\Delta v(\mathbf{w}^k, n)\| \leq C \|\Delta \mathbf{w}^k\|,$$

where $\delta(\mathbf{w}^k, n)$ will be given in the following remark. Then, apply the Taylor expansion to $\mathbf{v}^4(\mathbf{w}^k, n)$; we have

$$\begin{aligned} L(\mathbf{w}^{k+1}) - L(\mathbf{w}^k) = & -\frac{1}{N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z'_4(n)) \odot \\ & h^4(n-1) \oplus z'_4(n) \odot \tilde{h}(n)) \sigma'(\mathbf{v}^4(n)) \Delta \mathbf{v}^4(\mathbf{w}^k, n) \\ & - \frac{1}{2N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z''_4(n)) \odot h^4(n-1) \\ & \oplus z''_4(n) \odot \tilde{h}(n)) (\sigma'(\tau(\mathbf{w}^k, n)))^2 (\Delta \mathbf{v}^4(\mathbf{w}^k, n))^2 \\ & - \frac{1}{2N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z'_4(n)) \odot h^4(n-1) \\ & \oplus z'_4(n) \odot \tilde{h}(n)) \sigma''(\tau(\mathbf{w}^k, n)) (\Delta \mathbf{v}^4(\mathbf{w}^k, n))^2. \end{aligned}$$

Next, substituting (4.7) into the first order term of $\Delta \mathbf{v}^4(\mathbf{w}^k, n)$ of the above equation, we have

$$\begin{aligned} L(\mathbf{w}^{k+1}) - L(\mathbf{w}^k) = & -\frac{1}{N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z'_4(n)) \odot h^4(n-1) \\ & \oplus z'_4(n) \odot \tilde{h}(n)) \sigma'(\mathbf{v}^4(n)) (\mathbf{u}^4(n) + \mathbf{W}_L^4 \frac{\partial h^4(n-1)}{\partial \mathbf{w}_4^z}) \Delta \mathbf{w}^k + \delta(\mathbf{w}^k, n) \\ & - \frac{1}{2N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z''_4(n)) \odot h^4(n-1) \\ & \oplus z''_4(n) \odot \tilde{h}(n)) (\sigma'(\tau(\mathbf{w}^k, n)))^2 (\Delta \mathbf{v}^4(\mathbf{w}^k, n))^2 \\ & - \frac{1}{2N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z'_4(n)) \odot h^4(n-1) \\ & \oplus z'_4(n) \odot \tilde{h}(n)) \sigma''(\tau(\mathbf{w}^k, n)) (\Delta \mathbf{v}^4(\mathbf{w}^k, n))^2. \end{aligned}$$

Then, combining (4.2), (4.3) and (4.6), we will get

$$\begin{aligned} L(\mathbf{w}^{k+1}) - L(\mathbf{w}^k) = & -\frac{1}{\gamma} \|\Delta \mathbf{w}^k\|^2 - \frac{1}{N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z'_4(n)) \odot h^4(n-1) \\ & \oplus z'_4(n) \odot \tilde{h}(n)) \sigma'(\mathbf{v}^4(n)) \delta(\mathbf{w}^k, n) - \frac{1}{2N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) \\ & ((1 - z''_4(n)) \odot h^4(n-1) \oplus z''_4(n) \odot \tilde{h}(n)) (\sigma'(\tau(\mathbf{w}^k, n)))^2 (\Delta \mathbf{v}^4(\mathbf{w}^k, n))^2 \\ & - \frac{1}{2N} \sum_{n=1}^N \frac{y(n) - \hat{y}(n)}{|y(n) - \hat{y}(n)|} \left(\frac{\partial \hat{y}(n)}{\partial h^4(n)} \right) ((1 - z'_4(n)) \odot h^4(n-1) \\ & \oplus z'_4(n) \odot \tilde{h}(n)) \sigma''(\tau(\mathbf{w}^k, n)) (\Delta \mathbf{v}^4(\mathbf{w}^k, n))^2 \\ = & -\frac{1}{\gamma} \|\Delta \mathbf{w}^k\|^2 + \varphi(\mathbf{w}^k). \end{aligned}$$

By Assumptions 4.1 and 4.2, together with the Cauchy-Schwarz inequality, we obtain that $|\varphi(\mathbf{w}^k)| \leq C\|\Delta\mathbf{w}^k\|^2$. Hence $L(\mathbf{w}^{k+1}) - L(\mathbf{w}^k) \leq (-\frac{1}{\gamma} + C)\|\Delta\mathbf{w}^k\|^2$. If the learning rate is sufficiently small and satisfies that $0 < \gamma < 1/C$ (C is a constant), then the first conclusion of Theorem 4.1 is established.

Since $L(\mathbf{w}^k) \geq 0$ is monotonically bounded, there exists an L^* such that $\lim_{k \rightarrow \infty} L(\mathbf{w}^k) = L^*$. The proof is completed.

Remark 4.1. In (4.7), we have

$$\begin{aligned} \delta(\mathbf{w}^k, n) &= \sum_{d=1}^{n-1} \prod_{l=n-d+1}^n \mathbf{W}_L^k((1 - z'_4(l)) \odot h^4(l-1) \oplus z'_4(l) \odot \tilde{h}(l)) \sigma'(v^4(l)) \Delta \mathbf{W}_L^k \\ &\quad \times ((1 - z'_4(n-d)) \odot h^4(n-d-1) \oplus z'_4(n-d) \odot \tilde{h}(n-d)) \\ &\quad \sigma'(v^4(n-d)) \Delta \mathbf{v}(\mathbf{w}^k, n-d) + \frac{1}{2} \sum_{d=0}^{n-1} \prod_{l=n-d+1}^n \mathbf{W}_L^k((1 - z'_4(l)) \\ &\quad \odot h^4(l-1) \oplus z'_4(l) \odot \tilde{h}(l)) \sigma'(v^4(l)) \times \Delta \mathbf{W}_L^k \times (((1 - z'_4(n-d)) \\ &\quad \odot h^4(n-d-1) \oplus z'_4(n-d) \odot \tilde{h}(n-d)) (\sigma'(\tau(n-d)))^2 + ((1 - z'_4(n-d)) \\ &\quad \odot h^4(n-d-1) \oplus z'_4(n-d) \odot \tilde{h}(n-d)) \sigma''(\tau(n-d))) (\Delta \mathbf{v}(\mathbf{w}^k, n-d))^2, \end{aligned}$$

with

$$\prod_{l=n-d+1}^n \mathbf{W}_L^k((1 - z'_4(l)) \odot h^4(l-1) \oplus z'_4(l) \odot \tilde{h}(l)) \sigma'(v^4(l)) = 1.$$

when $d = 1$. Here, $\tau(n-d)$ is the intermediate quantity for $\mathbf{v}(\mathbf{w}^{k+1}, n)$ and $\mathbf{v}(\mathbf{w}^k, n)$.

From (4.3) and (4.4), we have

$$\begin{aligned} \frac{\partial h^4(n)}{\partial w_4^z} &= \sum_{d=0}^{n-1} \prod_{l=n-d+1}^n ((1 - z'_4(l)) \odot h^4(l-1) \oplus z'_4(l) \odot \tilde{h}(l)) \sigma'(v^4(l)) \times \\ &\quad \mathbf{W}_L^4((1 - z'_4(n-d)) \odot h^4(n-d-1) \oplus z'_4(n-d) \\ &\quad \odot \tilde{h}(n-d)) \times \sigma'(v^4(n-d)) u(n-d). \end{aligned}$$

When $d = 0$, $\prod_{l=n-d+1}^n \mathbf{W}_L^4((1 - z'_4(l)) \odot h^4(l-1) \oplus z'_4(l) \odot \tilde{h}(l)) \sigma'(v^4(l)) = 1$.

5. Numerical examples

In this section, we will give some numerical examples to show the effectiveness of the presented method.

The scattered data described by $\mathbf{I}_D = (I(1), I(2), \dots, I(T))$ corresponding to the obstacle D are given by Definition 2.1. The shape parameters $\mathbf{y}_D = (y(1), y(2), \dots, y(N))$ are defined as in Assumption 2.1. We denote the training data set by

$$T' = \{(\mathbf{I}_D, \mathbf{y}_D), D \in \mathcal{D}\},$$

where \mathcal{D} denotes the set of shape parameters. For the elements in T' , \mathbf{I}_D is also updated as the shape D changes, and we randomly selected 2% of the scattered data as the testing set to get the final prediction results.

The inverse problem can be solved in two steps. First, for a given shape D , the elastic wave equation (2.1) is solved by using the MFS described in Section 2. Second, the network was trained to identify the obstacle from the scattered data, i.e., the network was trained to approximate the Fourier coefficients of the obstacle.

To test the sensitivity to the noise, we suppose that noisy data is generated by adding Gaussian random noise pointwise, i.e.,

$$I^\delta(i) = I(i) + \delta \cdot I(i)\xi(i),$$

where $\delta \geq 0$ denotes the noise level and the random variable $\xi(i)$ is the standard Gaussian distribution.

In all numerical examples, the parameters are given as follows.

- (1) We shall use the Lamé constants $\lambda = 2$ and $\mu = 1$ and the frequency $\omega = 3$, $\rho = 1$.
- (2) The symbol “-” signifies the exact boundary of an obstacle, and “- -” signifies the forecast boundary of an obstacle.
- (3) The number of neurons of the GRU neural network is generally set as 2^p ($p \in \mathbb{R}^+$), and the network parameters were determined through a large number of numerical experiments. The number of GRU neurons was set as 128. Dropout was set as 0.5. Batch processing was set as 500. And the number of iterations is 40.

Remark 5.1. The scattering data are complex numbers and the current neural network method can only deal with real numbers; thus, we took out the real and imaginary parts of the scattering data as the scattering data information in the training data set. In addition, we need to use a random function to generate random numbers as the shape parameters in the training data set. This is the preparation of the training set. After the training set is prepared, the network model is trained by using the data in the training set through the two-layer network which has been built. Finally the output of the network, i.e., the predicted values of the shape parameters, is obtained by using the trained model to use the test data set.

Example 5.1 (Dirichlet boundary condition). First, the scattered data used in the training network are noise free, and the scattered data are collected on a circle with the radius $R = 3$ centered at the origin. We chose to investigate the numerical recovery results by changing the number of measurements T . The $\mathbf{d} = (1, 0)$ is the incident direction. The boundaries of the scatterers used in this example are parameterized as follows.

The first case is a heart-shaped domain parameterized by

$$\mathbf{x}(\theta) = \left(1.1 + 1.6 \cos^2\left(\frac{\theta}{2}\right) \sin\left(\frac{\theta}{2}\right)\right) (\cos \theta, \sin \theta).$$

The second case is a round-square domain parameterized by

$$\mathbf{x}(\theta) = (\cos^{10} \theta + \sin^{10} \theta)^{-\frac{1}{10}} (\cos \theta, \sin \theta).$$

The polar diameter has the following (truncated) Fourier representation: $\frac{a_0}{2} + \sum_{m=1}^Q a_m \cos(m\theta)$. We take $Q = 6$, and the idea is to reconstruct the Fourier coefficients a_m . The coefficients used in the training process, i.e., $a_0 \in [0.5, 4]$, $a_1, a_2 \in [-0.5, 0.5]$ and $a_m \in [-0.05, 0.05]$, $m = 3, 4, 5, 6$, are uniformly distributed random numbers.

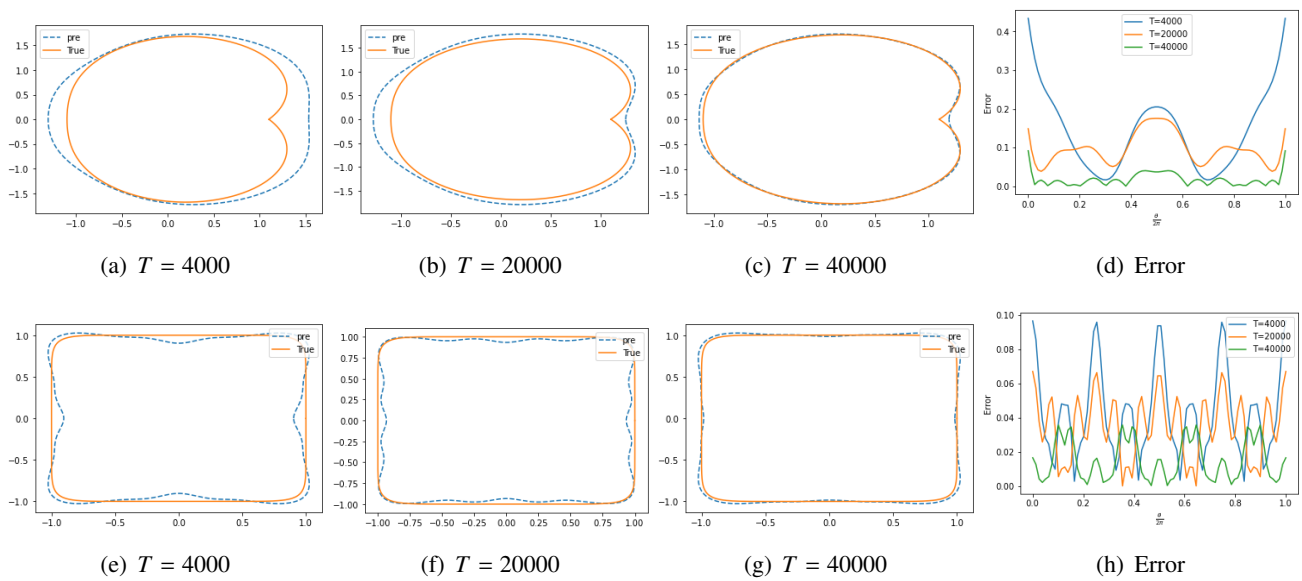


Figure 2. Numerical results for recovering the heart-shaped and round-square domain, obtained with noise free data and different numbers $T \in \{4000, 20,000, 40,000\}$ for Example 5.1.

Figure 2 presents the recovery results for a fixed number $T \in \{4000, 20,000, 40,000\}$. From the numerical recovery results in Figure 2, we can see that the accuracy in the numerical results will be improved by increasing the number T .

Second, in addition to the usual rounding errors, we also contaminated the forward scattered data by adding random noise. We chose to investigate the stability of the inversion schemes by changing the noise level δ . Figure 3 presents the recovery results for the noise level $\delta \in \{5\%, 10\%, 15\%\}$. From the fourth column in Figure 3, we can see that the errors are becoming smaller as the noise level decreases. We can also see that the numerical reconstruction results are poorest where the body has high curvature. This meets our expectations, since more expansion coefficients are required at the corner.

Example 5.2 (Neumann boundary condition). For comparison, in this example we consider the scattering problem by cavities, i.e., the Neumann boundary condition is imposed. In this example, we will consider a heart-shaped domain parameterized by

$$\mathbf{x}(\theta) = \left(1.1 + 1.6 \cos^2\left(\frac{\theta}{2}\right) \sin\left(\frac{\theta}{2}\right)\right) (\cos \theta, \sin \theta).$$

The training sets are the same as in Example 5.1.

We chose the direction $\mathbf{d} = (1, 0)$. The scattered data were collected on a circle with the radius $R = 3$ centered at the origin. The numerical recovery results for the noise level $\delta \in \{0\%, 5\%, 10\%, 15\%\}$ with a fixed number $T = 40,000$ are shown in Figure 4. From this, we observe that the locations and shapes are still well captured. From the fourth row in Figure 4, we can see that the errors are becoming smaller as the noise level decreases. We can also see that the numerical reconstruction results are poorest where the body has high curvature. This further shows that our methods are independent of the physical properties of the underlying scatterers.

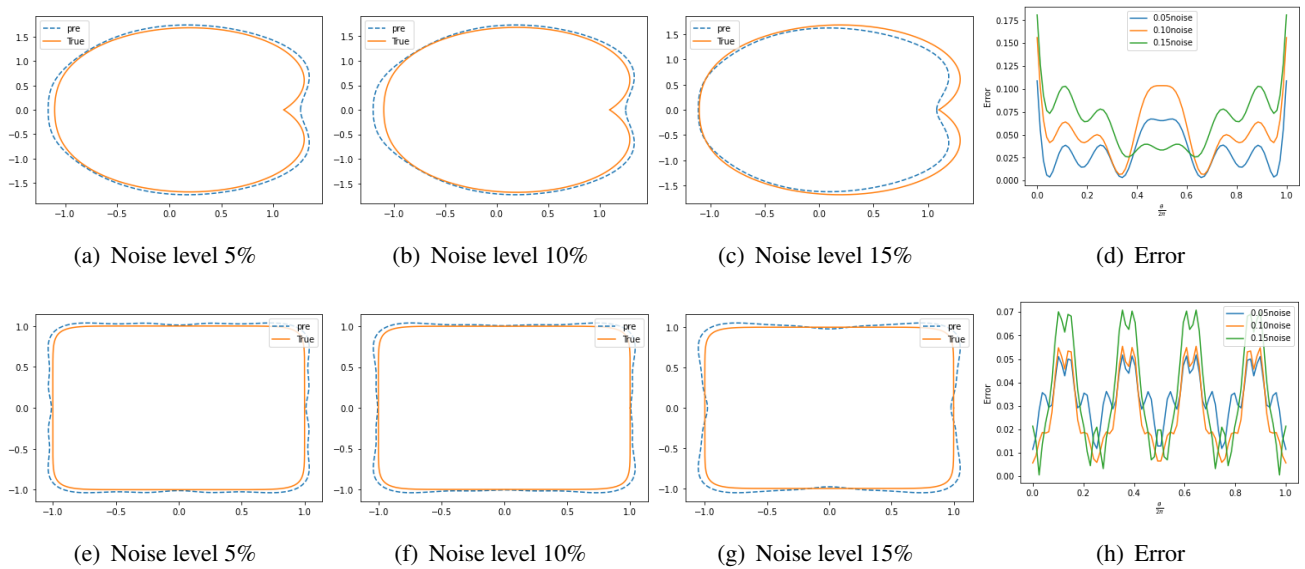


Figure 3. Numerical results for recovering the heart-shaped and round-square domain, obtained with different noise levels $\delta \in \{5\%, 10\%, 15\%\}$ and $T = 40,000$ for Example 5.1.

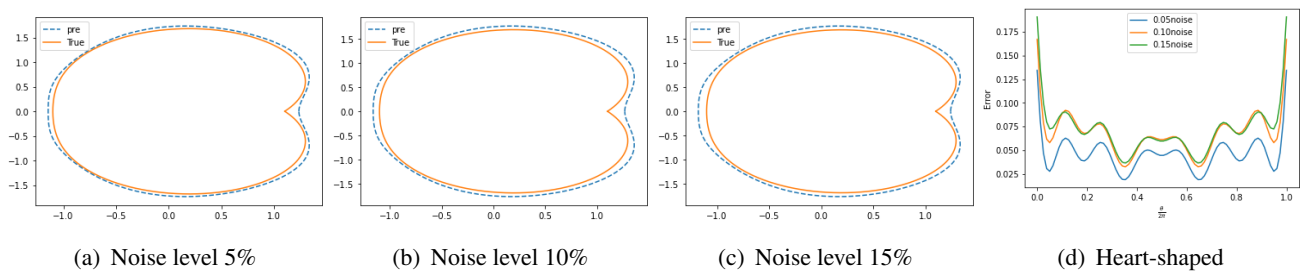


Figure 4. Numerical results for recovering the heart-shaped domain, obtained with different noise levels $\delta \in \{0\%, 5\%, 10\%, 15\%\}$ and $T = 40,000$ for Example 5.2.

Example 5.3 (Neumann boundary condition). In this example, we will consider the case of an apple-shaped domain parameterized by

$$\mathbf{x}(\theta) = \left(1 + \frac{\sin 2\theta}{5 + 4 \cos \theta} \right) (\cos \theta, \sin \theta).$$

The polar diameter has the following (truncated) Fourier representation $\frac{a_0}{2} + \sum_{m=1}^Q b_m \sin(m\theta)$. We take $Q = 6$, and the idea is to reconstruct the Fourier coefficients a_0 and b_m . The coefficients used in the training process, i.e., $a_0 \in [0.5, 3]$, $b_1, b_2, b_3, \in [-0.5, 0.5]$ and $b_4, b_5, b_6, \in [-0.1, 0.1]$, are uniformly distributed random numbers.

We chose the direction $\mathbf{d} = (1, 0)$. The scattered data were collected on a circle with the radius $R = 3$ centered at the origin. The numerical recovery results for the noise level $\delta \in \{5\%, 10\%, 15\%\}$

with a fixed number $T = 40,000$ are shown in Figure 5. The error for all testing sets obtained with different noise levels $\delta \in \{5\%, 10\%, 15\%\}$ and $T = 40,000$ is demonstrated by the fourth column in Figure 5. From this, we observe that the locations and shapes are still well captured.

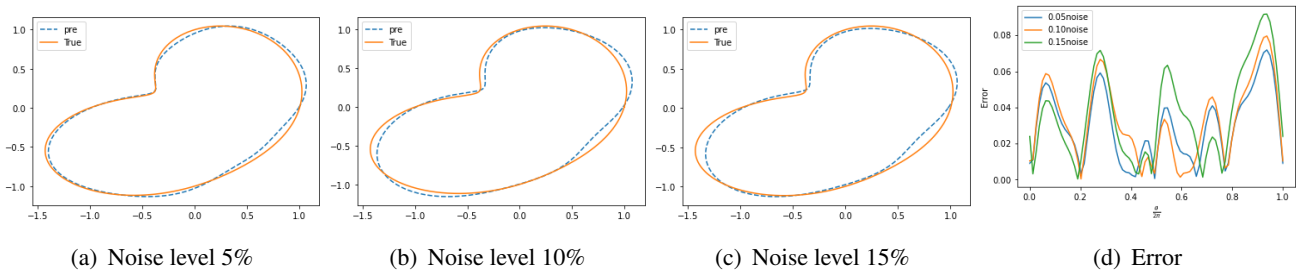


Figure 5. Numerical results for recovering the apple-shaped domain obtained with different noise levels $\delta \in \{5\%, 10\%, 15\%\}$ and $T = 40,000$ for Example 5.3.

The above examples give the effectiveness of the presented method for the case of a single obstacle or cavity. The following example will give the case of two disjoint components. Meanwhile, we should know that D is the case of two disjoint components as the a priori information. The Fourier coefficients were tracked as two independent groups.

Example 5.4 (Mixed boundary conditions). As shown in Figure 6, we consider a scatterer with two disjoint components. We set the scatterer to be a combination of a triangle-shaped domain and a peanut-shaped domain. Suppose that D_1 is a big triangle-shaped domain of the Dirichlet kind, parameterized by

$$\mathbf{x}(\theta) = 0.8(2 + 0.3 \cos \theta)(\cos \theta, \sin \theta).$$

D_2 is a mini peanut-shaped cavity. The boundary ∂D_2 is parameterized by

$$\mathbf{x}(\theta) = 0.05 \sqrt{3 \cos^2 \theta + 1}(\cos \theta, \sin \theta) + (2, -1).$$

We chose the direction $\mathbf{d} = (\frac{\sqrt{3}}{2}, \frac{1}{2})$ in this example. The scattered data were collected on a circle with the radius $R = 3$ centered at the origin. Figure 6 shows the numerical results obtained from the knowledge of the compressional near-field measurements, the shear near-field measurements and the full near-field measurements with $\delta = 10\%$. From these subfigures, we observe that the locations and shapes can be captured. This further shows that the presented method is independent of the physical properties of the underlying scatterers, even for two disconnected obstacles with different boundary conditions. We also note that the numerical error is worse than that in the above examples.

Example 5.5 (Two obstacles). As shown in Figures 7 and 8, the underlying scatterer D is given as the union of two disjoint obstacles $D = D_1 \cup D_2$. We consider two cases in this example.

- **Dirichlet boundary condition:** We chose the direction $\mathbf{d} = (1, 0)$. D_1 is a peanut-shaped domain, and the boundary ∂D_1 is parameterized by

$$\mathbf{x}(t) = 0.5 \sqrt{3 \cos^2 t + 1}(\cos t, \sin t) + (0, 3).$$

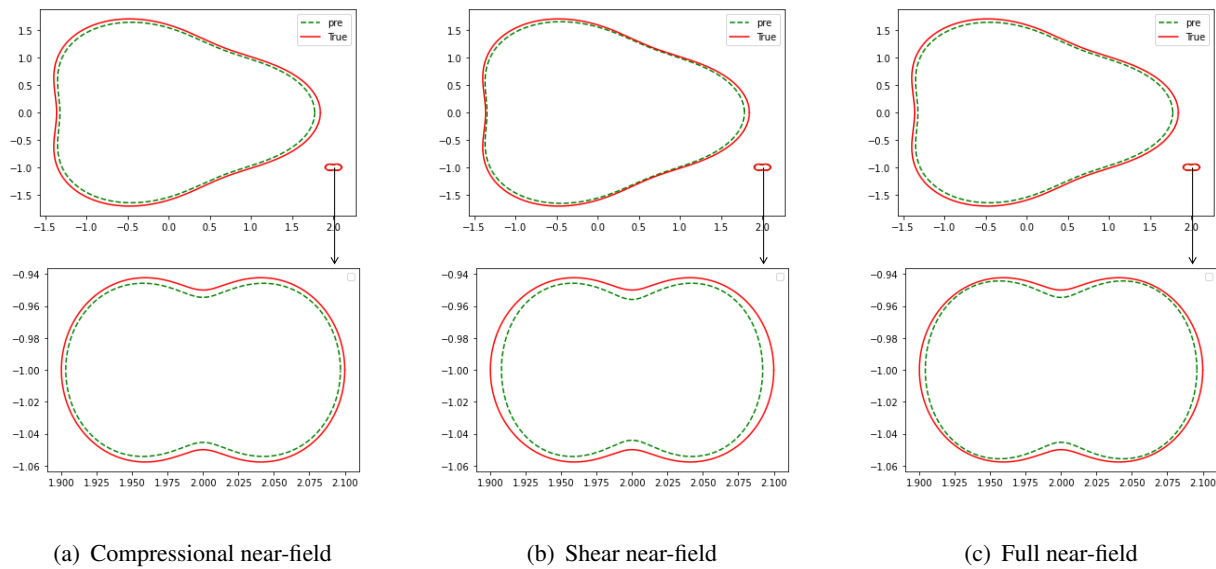


Figure 6. Numerical results for recovering a Dirichlet-type boundary ∂D_1 and a Neumann-type boundary ∂D_2 shown in Example 5.4 with $\delta = 10\%$, $T = 40,000$.

D_2 is a heart-shaped domain, and the boundary ∂D_2 is parameterized by

$$\mathbf{x}(\theta) = 0.5 \left(1.1 + 1.6 \cos^2 \left(\frac{\theta}{2} \right) \sin \left(\frac{\theta}{2} \right) \right) (\cos \theta, \sin \theta) + (3, 0).$$

- **Neumann boundary condition:** We chose the direction $\mathbf{d} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)$. D_1 is a round-square domain, and the boundary ∂D_1 is parameterized by

$$\partial D_1 = 0.5(\cos^{10} \theta + \sin^{10} \theta)^{-\frac{1}{10}} (\cos \theta, \sin \theta).$$

D_2 is a kite-shaped domain, and the boundary ∂D_2 is parameterized by

$$\partial D_2 = 0.4(\cos \theta + 0.65 \cos 2\theta - 0.65, 1.5 \sin \theta) + (1.3, 0).$$

The numerical recovery results for the noise level $\delta \in \{0\%, 10\%\}$ with a fixed number $T = 40,000$ are shown in Figures 7 and 8. From this, we observe that the locations and shapes are still well captured.

Example 5.6 (Dirichlet boundary condition). The last example considered here is a different option of the physical parameters. The Lamé constants are given as $\lambda = 3.88, \mu = 2.56$ and the frequency $\omega = \pi$. The scattered data were collected on a circle with the radius $R = 3$ centered at the origin. The incident direction is $\mathbf{d} = \left(-\frac{1}{2}, \frac{3}{2} \right)$. The boundary of the scatterer is a heart-shaped domain parameterized by

$$\mathbf{x}(\theta) = 0.5 \left(1.1 + 1.6 \cos^2 \left(\frac{\theta}{2} \right) \sin \left(\frac{\theta}{2} \right) \right) (\cos \theta, \sin \theta).$$

Figure 9 presents the recovery results for a fixed number $T = 40,000$ with the noise level $\delta \in \{5\%, 10\%, 15\%\}$. It can be seen that the recovery results are independent of the physical parameters.

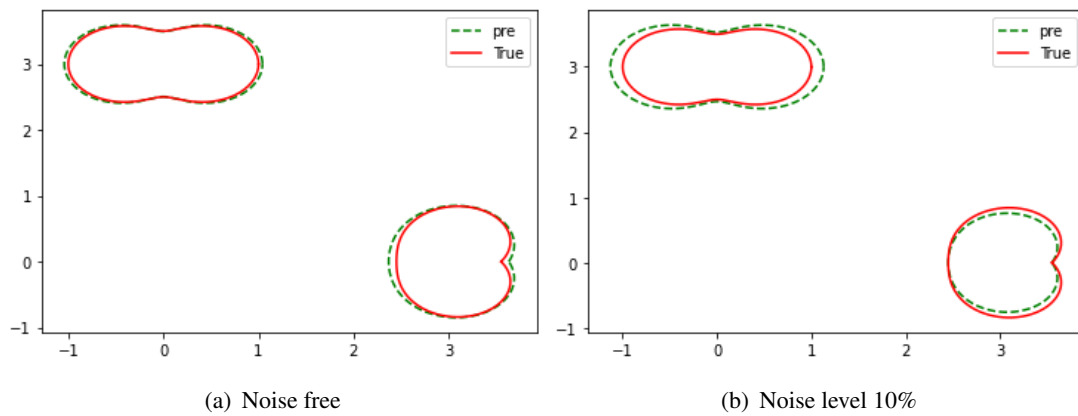


Figure 7. Numerical recovery results for Dirichlet boundary condition on ∂D for Example 5.5 with $T = 40,000$.

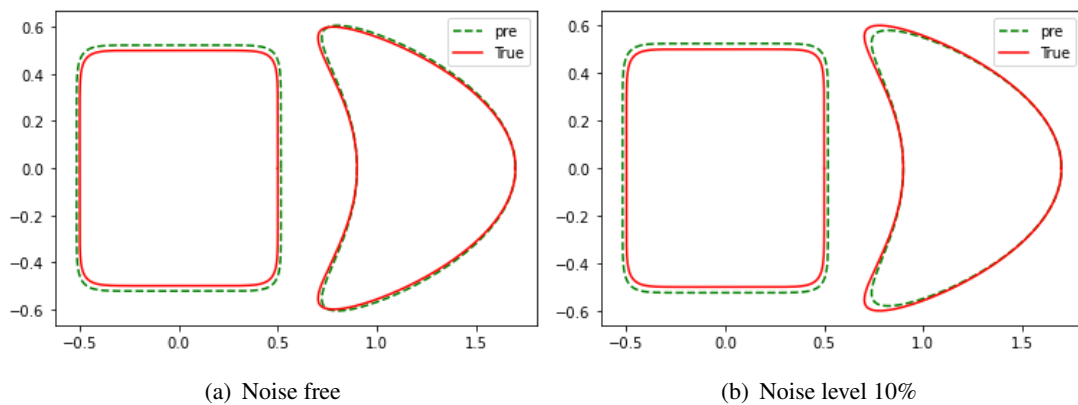


Figure 8. Numerical recovery results for Neumann boundary condition on ∂D for Example 5.5 with $T = 40,000$.

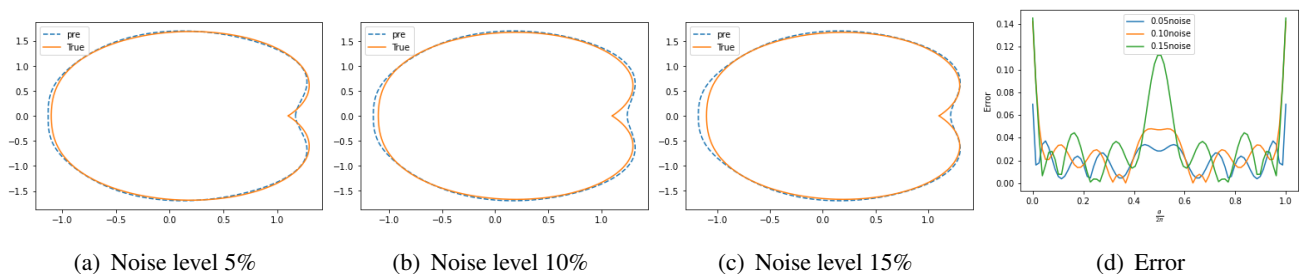


Figure 9. Numerical results for recovering the heart-shaped domain obtained with different noise levels $\delta \in \{5\%, 10\%, 15\%\}$ and $T = 40,000$ for Example 5.6.

6. Conclusions

In this paper, we applied a neural network trained on solutions from the MFS to solve inverse elastic wave scattering problems. Both convex and non-convex geometries with smooth boundaries have been investigated. The numerical results confirm the effectiveness of the presented method. From the presented examples, we can see that the presented method is independent of the physical properties of the underlying scatterers. When the boundary consists of two parts, this method also gives a good approximation of the exact boundary curves.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

We would like to thank the reviewers for their valuable comments that helped to improve the quality of the submitted manuscript. We also thank professor G. Hu for the valuable comments on this paper. The research was supported by the Natural Science Foundation of China (Nos. 12101603, 11501566), the Fundamental Research Funds for the Central Universities (No. 3122019159) and Tianjin Education Commission Research Project (No. 2022KJ072).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. G. Bao, T. Yin, Recent progress on the study of direct and inverse elastic scattering problems (in Chinese), *Sci. Sin. Math.*, **47** (2017), 1103–1118. <https://doi.org/10.1360/N012016-00198>
2. D. Colton, R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, Springer Nature, Berlin, 2019. <https://doi.org/10.1007/978-3-030-30351-8>
3. F. Cakoni, D. Colton, *A Qualitative Approach to Inverse Scattering Theory*, Springer, US, 2014. <https://doi.org/10.1007/978-1-4614-8827-9>
4. F. Cakoni, D. Colton, *Qualitative Methods in Inverse Scattering Theory*, Springer, Vienna, 2006. <https://doi.org/10.1007/3-540-31230-7>
5. H. Ammari, E. Bretin, J. Garnier, H. Kang, H. Lee, A. Wahab, *Mathematical Methods in Elasticity Imaging*, Princeton University Press, New Jersey, 2015.
6. J. H. Bramble, J. E. Pasciak, A note on the existence and uniqueness of solutions of frequency domain elastic wave problems: a priori estimates in H^1 , *J. Math. Anal. Appl.*, **345** (2008), 396–404. <https://doi.org/10.1016/j.jmaa.2008.04.028>

7. G. Bao, G. Hu, J. Sun, T. Yin, Direct and inverse elastic scattering from anisotropic media, *J. Math. Pures Appl.*, **117** (2018), 263–301. <https://doi.org/10.1016/j.matpur.2018.01.007>
8. H. Liu, M. Petrini, L. Rondi, J. Xiao, Stable determination of sound-hard polyhedral scatterers by a minimal number of scattering measurements, *J. Differ. Equations*, **262** (2017), 1631–1670. <https://doi.org/10.1016/j.jde.2016.10.021>
9. J. Li, H. Liu, Y. Wang, Recovering an electromagnetic obstacle by a few phaseless backscattering measurements, *Inverse Probl.*, **333** (2017), 035011. <https://doi.org/10.1088/1361-6420/aa5bf3>
10. J. Li, H. Liu, Q. Wang, Enhanced multilevel linear sampling methods for inverse scattering problems, *J. Comput. Phys.*, **257** (2014), 554–571. <https://doi.org/10.1016/j.jcp.2013.09.048>
11. T. Arens, Linear sampling methods for 2D inverse elastic wave scattering, *Inverse Probl.*, **17** (2001), 1445–1464. <https://doi.org/10.1088/0266-5611/17/5/314>
12. A. Charalambopoulos, A. Kirsch, K. Anagnostopoulos, D. Gintides, K. Kiriaki, The factorization method in inverse elastic scattering from penetrable bodies, *Inverse Probl.*, **23** (2007), 27–51. <https://doi.org/10.1088/0266-5611/23/1/002>
13. G. Hu, A. Kirsch, M. Sini, Some inverse problems arising from elastic scattering by rigid obstacles, *Inverse Probl.*, **29** (2013), 015009. <https://doi.org/10.1088/0266-5611/29/1/015009>
14. Z. Chen, G. Huang, Reverse time migration for extended obstacles: elastic waves (in Chinese), *Sci. Sin. Math.*, **45** (2015), 1103–1114. <https://doi.org/10.1360/N012014-00097>
15. P. Li, Y. Wang, Z. Wang, Y. Zhao, Inverse obstacle scattering for elastic waves, *Inverse Probl.*, **32** (2016), 115018. <https://doi.org/10.1088/0266-5611/32/11/115018>
16. J. António, A. Tadeu, L. Godinho, A three-dimensional acoustics model using the method of fundamental solutions, *Eng. Anal. Boundary Elem.*, **32** (2008), 525–531. <https://doi.org/10.1016/j.enganabound.2007.10.008>
17. A. Karageorghis, B. T. Johansson, D. Lesnic, The method of fundamental solutions for the identification of a sound-soft obstacle in inverse acoustic scattering, *Appl. Numer. Math.*, **62** (2012), 1767–1780. <https://doi.org/10.1016/j.apnum.2012.05.011>
18. Y. Sun, X. Lu, B. Chen, The method of fundamental solutions for the high frequency acoustic-elastic problem and its relationship to a pure acoustic problem, *Eng. Anal. Boundary Elem.*, **156** (2023), 299–310. <https://doi.org/10.1016/j.enganabound.2023.08.010>
19. A. Karageorghis, D. Lesnic, L. Marin, The MFS for the identification of a sound-soft interior acoustic scatterer, *Eng. Anal. Boundary Elem.*, **83** (2017), 107–112. <https://doi.org/10.1016/j.enganabound.2017.07.021>
20. A. Karageorghis, D. Lesnic, L. Marin, The method of fundamental solutions for the identification of a scatterer with impedance boundary condition in interior inverse acoustic scattering, *Eng. Anal. Boundary Elem.*, **92** (2018), 218–224. <https://doi.org/10.1016/j.enganabound.2017.07.005>
21. D. Colton, R. Kress, Using fundamental solutions in inverse scattering, *Inverse Probl.*, **22** (2006), R49. <https://doi.org/10.1088/0266-5611/22/3/R01>
22. C. J. S. Alves, N. F. M. Martins, S. S. Valtchev, Extending the method of fundamental solutions to non-homogeneous elastic wave problems, *Appl. Numer. Math.*, **115** (2017), 299–313. <https://doi.org/10.1016/j.apnum.2016.06.002>

23. W. Yin, W. Yang, H. Liu, A neural network scheme for recovering scattering obstacles with limited phaseless far-field data, *J. Comput. Phys.*, **417** (2020), 109594. <https://doi.org/10.1016/j.jcp.2020.109594>
24. Z. Wei, X. Chen, Deep-learning schemes for full-wave nonlinear inverse scattering problems, *IEEE Trans. Geosci. Remote Sens.*, **57** (2019), 1849–1860. <https://doi.org/10.1109/TGRS.2018.2869221>
25. E. Chung, J. G. Park, Sentence-chain based Seq2seq model for corpus expansion, *ETRI J.*, **39** (2017), 455–466. <https://doi.org/10.4218/etrij.17.0116.0074>
26. J. Torres, C. Vaca, L. Terán, C. L. Abad, Seq2Seq models for recommending short text conversations, *Expert Syst. Appl.*, **150** (2020), 113270. <https://doi.org/10.1016/j.eswa.2020.113270>
27. Y. Khoo, L. Ying, SwitchNet: a neural network model for forward and inverse scattering problems, *SIAM J. Sci. Comput.*, **41** (2019), A3182–A3201. <https://doi.org/10.1137/18M1222399>
28. G. Fairweather, A. Karageorghis, P. A. Martin, The method of fundamental solutions for scattering and radiation problems, *Eng. Anal. Boundary Elem.*, **27** (2003), 759–769. [https://doi.org/10.1016/S0955-7997\(03\)00017-1](https://doi.org/10.1016/S0955-7997(03)00017-1)
29. Y. Sun, F. Ma, A meshless method for the Cauchy problem in linear elastodynamics, *Appl. Anal.*, **93** (2014), 2647–2667. <https://doi.org/10.1080/00036811.2014.882913>
30. P. Zhang, P. Meng, W. Yin, H. Liu, A neural network method for time-dependent inverse source problem with limited-aperture data, *J. Comput. Appl. Math.*, **421** (2023), 114842. <https://doi.org/10.1016/j.cam.2022.114842>
31. D. Xu, Z. Li, W. Wu, X. Ding, D. Qu, Convergence of gradient method descent algorithm for recurrent neuron, *Int. Symp. Neural Network*, **4493** (2007), 117–122. https://doi.org/10.1007/978-3-540-72395-0_16
32. D. Xu, Z. Li, W. Wu, Convergence of gradient method for fully recurrent neural network, *Soft Comput.*, **14** (2010), 245–250. <https://doi.org/10.1007/s00500-009-0398-0>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)