AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 11(9): 30759–30781.
DOI: 10.3934/math.20261218
Received: 29 July 2026
Revised: 03 September 2026
Accepted: 08 September 2026
Published: 21 September 2026

Research article

Physics-informed neural network method for elasticity interface problems and its error analysis

Bokyu Kim[1], **Dongsik Jo**[2,*], **and Gwanghyun Jo**[1,*]

[1] Department of Mathematical Data Science, Hanyang University ERICA, Ansan-si 15588, Republic of Korea

[2] Department of Electrical, Electronic and Computer Engineering, University of Ulsan , Ulsan 44610, Republic of Korea

* **Correspondence:** Email: gwanghyun@hanyang.ac.kr, dongsikjo@ulsan.ac.kr.

Abstract: The elasticity problem describes the relationship between displacement and stress in an elastic body in response to an external force. However, when a spring-type jump condition is imposed at the interface, it is difficult to numerically approximate the solution because the displacement becomes discontinuous across the interface and the displacement jump is given only implicitly. We propose a physics-informed neural network (PINN) to approximate solutions of elasticity equations with spring-type jump conditions. Since a PINN is represented by a continuous neural network function, it is inefficient for directly approximating discontinuous solutions. To remedy this, we introduce an additional variable that labels the subdomains, thereby representing the discontinuous solution as a continuous function in an augmented higher-dimensional space. This continuous representation enables us to exploit the universal approximation property of neural networks. The loss function of the proposed PINN is constructed to incorporate the governing elasticity equations. We show that the energy-norm error of the neural network approximation can be controlled by the residuals of the loss function and prove that these residuals can be made arbitrarily small. In addition, we conduct numerical experiments for circular- and line-interface problems as well as a driven cavity problem. Comparisons with a piecewise PINN and an immersed finite element method demonstrate the accuracy and efficiency of the proposed method.

Keywords: elasticity interface problem; physics-informed neural network; nonhomogeneous interface; spring-type jump condition; error estimates
Mathematics Subject Classification: 65N12, 65N15, 65N30

1. Introduction

The elasticity equations describe the relationship between displacement and stress in elastic bodies and play an important role in analyzing mechanical behavior in various engineering problems [1, 2]. In particular, in elastic structures composed of two materials, the displacement field may exhibit discontinuities across the interface, with the displacement jump being proportional to the normal traction acting on the interface [3–5]. In such *spring-type* jump conditions, the resulting elasticity equations are difficult to numerically solve, since the displacement jump is only implicitly given.

Several numerical approaches have been developed for solving elasticity problems with spring-type jump conditions. In finite element methods (FEMs), this problem has been addressed by introducing zero-thickness imperfect interface elements [6, 7]. Another approach is the extended finite element method (XFEM) [8, 9], in which specially designed enrichment functions are introduced to capture displacement jumps induced by spring-type imperfect interfaces. More recently, the scaled boundary finite element method (SBFEM) has been used to model elasticity with spring-type jump conditions by incorporating zero-thickness, linearly elastic interface elements in a three-dimensional formulation [10, 11]. However, these methods require specialized interface treatments, and their implementation can become increasingly complex as the complexity of the problem increases [12, 13]. For these reasons, traditional numerical methods face limitations in effectively solving complex interface elasticity problems.

Meanwhile, mesh-free methods for solving partial differential equations (PDEs) have emerged with the extensive development of deep learning methods. One of the main reasons for the success of neural-network-based methods stems from the universal approximation property. In this light, physics-informed neural networks (PINNs), which are neural network-based surrogate methods for solving PDEs, have been successfully employed to solve various types of PDEs [14]. PINNs approximate solutions of PDEs via the inclusion of the governing equations together with interface and boundary conditions in the loss [15]. As a result, PINNs offer greater flexibility than conventional numerical methods for high-dimensional problems and problems involving complex interfaces [13]. In addition, theoretical studies have established error estimates for PINN approximations [16, 17].

However, since the neural network approximation in PINNs is smooth, approximating discontinuous functions is inefficient [18, 19]. This limitation presents a particular difficulty for the elasticity interface problem considered in this study, whose displacement field exhibits a jump across the interface. Recently, a PINN employing axis-augmentation has been introduced [20]. Instead of directly approximating discontinuous solutions, this method expands the input space so that the solution can be represented as a continuous function in each region separated by the interface.

Based on this strategy, we propose a new PINN framework for elasticity interface problems with spring-type jump conditions. Following [21, 22], we first apply a Sobolev extension to each subdomain displacement variable so that it is defined over the entire domain. Then, we construct a continuous global representation of the originally discontinuous solution through a linear combination of the previously obtained extended displacement variables. The advantage of this formulation is that the original discontinuous solution can be readily recovered by restricting the auxiliary variable. Next, we design a PINN to approximate the extended function. The loss function of the PINN is based on the governing equations of elasticity along with the interface and boundary conditions. In addition, we establish the validity of the proposed PINN by providing an error analysis for the proposed loss function. Finally, we present numerical experiments to demonstrate the performance of our proposed PINN.

This paper is organized into six main sections. Section 2 presents the governing equations and notation for the elasticity interface problem with spring-type jump conditions. Section 3 presents the proposed PINN method designed to capture discontinuous solutions. In Section 4, we derive the error estimates for the proposed method. Section 5 validates the accuracy and effectiveness of the proposed framework through a series of numerical experiments, including a comparison with the FEM. Finally, Section 6 summarizes the main findings and discusses directions for future research.

2. Preliminary

In this section, we describe the governing equations for the elasticity interface problem with spring-type jump conditions. Let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain partitioned into two bounded Lipschitz subdomains Ω^+ and Ω^- by the interface

$$\Gamma = \partial\Omega^- \subset \Omega,$$

where $\overline{\Omega^-} \subset \Omega$. The vectors $\mathbf{n}^s$ denote the outer unit normal vectors to Ω^s ($s = +, -$). We let $\mathbf{n}_\Gamma = \mathbf{n}^-$. The displacement $\boldsymbol{u} = (u_1, u_2)$ under the effects of body force $\boldsymbol{f} \in [L^2(\Omega)]^2$ follows the below relations:

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{f}, \qquad \text{in} \quad \Omega^+ \cup \Omega^-, \tag{2.1}$$

$$[\boldsymbol{u}]_\Gamma = -R\boldsymbol{\sigma}(\boldsymbol{u})\mathbf{n}_\Gamma, \qquad \text{on} \quad \Gamma, \tag{2.2}$$

$$[\boldsymbol{\sigma}(\boldsymbol{u})\mathbf{n}_\Gamma]_\Gamma = \mathbf{0}, \qquad \text{on} \quad \Gamma, \tag{2.3}$$

where $R_{ij} := (\alpha\delta_{ij} + (\beta - \alpha)n_i n_j)$ is an elasticity matrix, and the stress $\boldsymbol{\sigma}(\boldsymbol{u})$ is defined as $2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda \operatorname{div} \boldsymbol{u} I$ and $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right)$. Here, we adopt the convention

$$(\nabla\boldsymbol{u})_{ij} = \frac{\partial u_i}{\partial x_j}, \quad i, j \in \{1, 2\}.$$

In (2.1), the divergence of the stress tensor is evaluated separately in Ω^+ and Ω^-, while the interface behavior is prescribed by the jump conditions in (2.2) and (2.3). We assume that $\alpha > 0$ and $\beta > 0$. The eigenvalues of R are α and β. Hence, R is positive definite and invertible with its spectral condition number $\kappa_2(R)$ and the spectral norm of its inverse $\left\|R^{-1}\right\|_2$ given by

$$\kappa_2(R) = \frac{\max(\alpha, \beta)}{\min(\alpha, \beta)}, \quad \left\|R^{-1}\right\|_2 = \frac{1}{\min(\alpha, \beta)}.$$

Consequently, a large disparity between α and β may deteriorate the conditioning of the interface relation, and the constants involving R or R^{-1} may depend on $\max(\alpha, \beta)$ and $1/\min(\alpha, \beta)$. Here, $[\cdot]_\Gamma$ denotes the jump across the interface Γ, i.e.,

$$[\boldsymbol{u}]_\Gamma = \boldsymbol{u}^- - \boldsymbol{u}^+, \qquad [\boldsymbol{\sigma}(\boldsymbol{u})\mathbf{n}_\Gamma]_\Gamma = \boldsymbol{\sigma}(\boldsymbol{u}^-)\mathbf{n}_\Gamma - \boldsymbol{\sigma}(\boldsymbol{u}^+)\mathbf{n}_\Gamma,$$

where the superscripts + and – denote the traces from Ω^+ and Ω^-, respectively. Since the same fixed normal $\mathbf{n}_\Gamma$ is used for both traces, (2.3) is equivalent to $\boldsymbol{\sigma}(\boldsymbol{u}^-)\mathbf{n}_\Gamma = \boldsymbol{\sigma}(\boldsymbol{u}^+)\mathbf{n}_\Gamma$ on Γ. Because the outward normal to Ω^+ satisfies $\mathbf{n}^+ = -\mathbf{n}_\Gamma$ on Γ, the equality means that the physical outward tractions on the two subdomains are equal in magnitude and opposite in direction. Moreover, by (2.3), the traction in (2.2) may be evaluated using either trace. Using the trace from Ω^-, (2.2) can be rewritten as $\boldsymbol{\sigma}(\boldsymbol{u}^-)\mathbf{n}_\Gamma = R^{-1}(\boldsymbol{u}^+ - \boldsymbol{u}^-)$ on Γ, which confirms that the minus sign in (2.2) is consistent with the adopted jump and normal conventions. The Lamé parameters λ and μ are assumed to be constant within each material subdomain. For each $s \in \{+, -\}$, we set

$$\lambda|_{\Omega^s} = \lambda^s, \qquad \mu|_{\Omega^s} = \mu^s,$$

where $\mu^s > 0$ and $\lambda^s > 0$. Thus, the stress tensor is evaluated separately in each subdomain using λ^s and μ^s. The stress tensor may therefore be discontinuous across Γ, even when the displacement is smooth within each subdomain. Equation (2.3) enforces continuity of the normal traction, rather than continuity of the entire stress tensor. To ensure well-posedness of the solution, we impose a Dirichlet boundary condition.

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on} \quad \partial\Omega. \tag{2.4}$$

We believe any other reasonable choice of boundary condition, such as a traction-free condition, would lead to similar PINN methods. To avoid unnecessary complications, we restrict our attention to the compressible regime and consider fixed positive Lamé parameters satisfying

$$\lambda^s > 0, \qquad \mu^s > 0, \qquad s \in \{+, -\}.$$

The constants in the subsequent estimates may depend on the fixed material parameters $\lambda^\pm$ and $\mu^\pm$. We do not claim estimates that are uniform in the nearly incompressible limit

$$\frac{\lambda^s}{\mu^s} \to \infty,$$

and this regime is left for future work. Before closing this section, we introduce some notation. For a domain $D \subset \Omega$, the spaces $W^{k,p}(D)$ and $L^2(D)$ denote the usual Sobolev space and L^2 space on D, respectively, equipped with the norms $\|\cdot\|_{W^{k,p}(D)}$ and $\|\cdot\|_{0,D}$. The notation $(\cdot,\cdot)_D$ denotes the L^2-inner product on D. Also, $H^k(D) := W^{k,2}(D)$ with the norm denoted by $\|\cdot\|_{k,D}$. The subspace $H^1_0(D)$ is the set of functions in $H^1(D)$ with the vanishing trace on ∂D. When $D = \Omega$, we drop D in subscripts for simplicity, e.g, $\|\cdot\|_k$ and $(\cdot,\cdot)$. For vector-valued functions, we use the notation $[\cdot]^d$, for instance, $[H^k(\Omega)]^d$. We now define piecewise Sobolev spaces:

$$[W^{k,p}(\Omega^\pm)]^d = \{\boldsymbol{v} \in [L^2(\Omega)]^d \ : \ \boldsymbol{v}|_{\Omega^s} \in [W^{k,p}(\Omega^s)]^d, s \in \{-, +\}\}, \quad k \geq 1,$$
$$[H^k(\Omega^\pm)]^d = \{\boldsymbol{v} \in [L^2(\Omega)]^d \ : \ \boldsymbol{v}|_{\Omega^s} \in [H^k(\Omega^s)]^d, s \in \{-, +\}\}, \quad k \geq 1,$$

equipped with the norms

$$\|\boldsymbol{v}\|^p_{W^{k,p}(\Omega^\pm)} = \|\boldsymbol{v}\|^p_{W^{k,p}(\Omega^-)} + \|\boldsymbol{v}\|^p_{W^{k,p}(\Omega^+)},$$
$$\|\boldsymbol{v}\|^2_{k,\Omega^\pm} = \|\boldsymbol{v}\|^2_{k,\Omega^-} + \|\boldsymbol{v}\|^2_{k,\Omega^+}.$$

The piecewise C^1 space is defined similarly:

$$[C^1(\Omega^\pm)]^2 = \{\boldsymbol{v} \in [L^2(\Omega)]^2 \ : \ \boldsymbol{v}|_{\Omega^s} \in [C^1(\Omega^s)]^2, s \in \{-, +\}\},$$

with the norm

$$\|\boldsymbol{v}\|_{C^1(\Omega^\pm)} = \|\boldsymbol{v}|_{\Omega^-}\|_{C^1(\Omega^-)} + \|\boldsymbol{v}|_{\Omega^+}\|_{C^1(\Omega^+)}.$$

Throughout this paper, C is a generic constant not necessarily the same at each occurrence.

3. Method

In this section, we construct a PINN for the elasticity interface problem described in Section 2. Since the exact solution exhibits a displacement jump across the interface, directly approximating the solution using a standard PINN is inefficient because the neural network is smooth. To remedy this, we employ an axis-augmentation strategy [20–22] that transforms the discontinuous solution into a continuous function in an augmented space. Based on this continuous representation, we construct a PINN that approximates the elasticity solution while preserving the interface jump conditions.

The construction of the proposed method consists of three steps. First, the displacement variables defined separately in the two subdomains are extended to the entire domain. Second, the extended displacement variables are combined into a continuous function in an augmented space. Finally, an augmented neural network is introduced to approximate this continuous representation, and its parameters are determined by minimizing the physics-informed loss function.

First, $\boldsymbol{u}^s|_{\Omega^s} \in \left[H^2(\Omega^s)\right]^2$ is extended to $(\bar{\boldsymbol{u}})^s \in \left[H^2(\Omega)\right]^2$ by the Sobolev extension theorem [23], where $s = \pm$. Based on these extensions, we define the function

$$\boldsymbol{u}_{\text{aug}}(\mathbf{x}, z) = \frac{1+z}{2}(\bar{\boldsymbol{u}})^+(\mathbf{x}) + \frac{1-z}{2}(\bar{\boldsymbol{u}})^-(\mathbf{x}), \tag{3.1}$$

where z is an auxiliary variable that distinguishes the two subdomains. In particular,

$$\boldsymbol{u}_{\text{aug}}(\mathbf{x}, 1) = (\bar{\boldsymbol{u}})^+(\mathbf{x}) \qquad \text{and} \qquad \boldsymbol{u}_{\text{aug}}(\mathbf{x}, -1) = (\bar{\boldsymbol{u}})^-(\mathbf{x}).$$

Only the two values $z = \pm 1$ are used during training and prediction in the physical domain, as shown explicitly in the residual evaluations below; therefore, the intermediate values in $(-1, 1)$ are not evaluated during training. The main advantage of this formulation is that the discontinuous solution becomes continuous in the augmented space. Thus, instead of approximating the discontinuous displacement field directly in the physical domain, we approximate a continuous representation defined on the augmented input space. Since the universal approximation property of artificial neural networks holds for continuous functions on compact subsets of $\mathbb{R}^3$, we can construct a PINN that approximates $\boldsymbol{u}_{\text{aug}}$. To optimize the physics-informed objective function, we train an artificial neural network with $(\mathbf{x}, z)$ as the input to approximate $\boldsymbol{u}_{\text{aug}}$.

We now introduce the neural network used to approximate the augmented displacement function. For each i, let L_i be an affine mapping, and let $\sigma = \tanh$ denote the hyperbolic tangent activation function applied componentwise. We define the augmented neural network function $\boldsymbol{\phi}_\theta^{\text{aug}} : \mathbb{R}^3 \to \mathbb{R}^2$ with ℓ hidden layers by

$$\boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}, z) = (L_\ell \circ \sigma \circ L_{\ell-1} \circ \cdots \circ \sigma \circ L_0)(\mathbf{x}, z), \tag{3.2}$$

where θ denotes the set of training parameters. We denote by $\mathcal{N}^{\text{aug}}$ the set of all such augmented neural network functions.

Although the neural networks are defined in the augmented space $(\mathbf{x}, z)$, the actual approximation of the discontinuous solution is performed in the physical domain by restricting the auxiliary variable z to each subdomain. Specifically, a single augmented neural network induces a piecewise smooth function given by

$$\boldsymbol{\phi}_\theta(\mathbf{x}) = \boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}, 1)\chi_{\Omega^+}(\mathbf{x}) + \boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}, -1)\chi_{\Omega^-}(\mathbf{x}). \tag{3.3}$$

More precisely, (3.3) implies that

$$\boldsymbol{\phi}_\theta|_{\Omega^+} = \boldsymbol{\phi}_\theta^{\text{aug}}(\cdot, 1), \qquad \boldsymbol{\phi}_\theta|_{\Omega^-} = \boldsymbol{\phi}_\theta^{\text{aug}}(\cdot, -1).$$

Since the tanh activation function is C^∞, both restrictions are C^∞ in their respective subdomains, although their traces on Γ need not coincide. Therefore, the displacement jump can be represented using a single augmented neural network without introducing two independent neural networks. Accordingly, we perform the approximation in the function space

$$\mathcal{N}^\pm := \left\{\boldsymbol{\phi}_\theta : \boldsymbol{\phi}_\theta \text{ is generated from some } \boldsymbol{\phi}_\theta^{\text{aug}} \in \mathcal{N}^{\text{aug}}\right\}. \tag{3.4}$$

We remark that the space $\mathcal{N}^\pm$ is a subset of $[C^\infty(\Omega^\pm)]^2$.

Once the approximation space has been constructed, the network parameters are determined by enforcing the governing equation, the spring-type jump conditions, and the boundary condition at a finite number of training points. To describe the training process, we first define these training points. Let $\{\mathbf{x}_i^s\}_{i=1}^{N^s}$ denote the set of training points in Ω^s, where $s \in \{+, -\}$. Similarly, let $\{\mathbf{x}_i^\Gamma\}_{i=1}^{N^\Gamma}$ denote the set of training points on Γ, with N^Γ being the number of such points. Finally, let $\{\mathbf{x}_i^\partial\}_{i=1}^{N^\partial}$ denote the set of training points on $\partial\Omega$, with N^∂ being the corresponding number of points.

Using these training points, we define three loss components corresponding to the elasticity equation, the interface conditions, and the boundary condition. For any piecewise smooth function $\boldsymbol{\phi}_\theta \in \mathcal{N}^\pm$, the loss functionals are defined as follows:

$$\begin{aligned}
\mathcal{L}^\Omega(\boldsymbol{\phi}_\theta) &:= \sum_{s=+,-} \frac{1}{N^s} \sum_{i=1}^{N^s} \left|\boldsymbol{f}(\mathbf{x}_i^s) + \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\phi}_\theta(\mathbf{x}_i^s))\right|^2, \\
\mathcal{L}^\Gamma(\boldsymbol{\phi}_\theta) &:= \frac{1}{N^\Gamma} \sum_{i=1}^{N^\Gamma} \left(\left|[\boldsymbol{\phi}_\theta]_\Gamma(\mathbf{x}_i^\Gamma) + R(\boldsymbol{\sigma}(\boldsymbol{\phi}_\theta)|_{\Omega^-})(\mathbf{x}_i^\Gamma)\mathbf{n}_\Gamma\right|^2 + \left|[\boldsymbol{\sigma}(\boldsymbol{\phi}_\theta)\mathbf{n}_\Gamma]_\Gamma(\mathbf{x}_i^\Gamma)\right|^2 \right), \\
\mathcal{L}^\partial(\boldsymbol{\phi}_\theta) &:= \frac{1}{N^\partial} \sum_{i=1}^{N^\partial} \left|\boldsymbol{\phi}_\theta(\mathbf{x}_i^\partial) - \boldsymbol{g}(\mathbf{x}_i^\partial)\right|^2, \\
\mathcal{L}(\boldsymbol{\phi}_\theta) &:= \nu_0 \mathcal{L}^\Omega(\boldsymbol{\phi}_\theta) + \nu_1 \mathcal{L}^\Gamma(\boldsymbol{\phi}_\theta) + \nu_2 \mathcal{L}^\partial(\boldsymbol{\phi}_\theta).
\end{aligned} \tag{3.5}$$

Here, $\nu_0, \nu_1, \nu_2 > 0$ are the weights of the elasticity loss, the interface loss, and the boundary loss, respectively. Since multiplying the total loss by a positive constant does not change its minimizer, we normalize $\nu_0 = 1$. Dimensionally consistent positive weights, including normalization by the measures of the domain, interface, and boundary, may also be employed without altering the subsequent analysis, except for the constants involved.

Although the loss function is written in terms of the piecewise function $\boldsymbol{\phi}_\theta$, its residuals are evaluated using the augmented neural network $\boldsymbol{\phi}_\theta^{\text{aug}}$ during the implementation stage. More precisely,

$$\begin{aligned}
\nabla \cdot \boldsymbol{\sigma}\left(\boldsymbol{\phi}_\theta(\mathbf{x}_i^+)\right) &= \nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}\left(\boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}_i^+, z = 1)\right), \\
\nabla \cdot \boldsymbol{\sigma}\left(\boldsymbol{\phi}_\theta(\mathbf{x}_i^-)\right) &= \nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}\left(\boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}_i^-, z = -1)\right), \\
[\boldsymbol{\phi}_\theta]_\Gamma(\mathbf{x}_i^\Gamma) &= \boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}_i^\Gamma, z = -1) - \boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}_i^\Gamma, z = 1), \\
[\boldsymbol{\sigma}(\boldsymbol{\phi}_\theta)\mathbf{n}_\Gamma]_\Gamma(\mathbf{x}_i^\Gamma) &= \boldsymbol{\sigma}\left(\boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}_i^\Gamma, z = -1)\right)\mathbf{n}_\Gamma - \boldsymbol{\sigma}\left(\boldsymbol{\phi}_\theta^{\text{aug}}(\mathbf{x}_i^\Gamma, z = 1)\right)\mathbf{n}_\Gamma.
\end{aligned}$$

Here,

$$\nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}(\boldsymbol{\phi}_\theta^{\text{aug}}) = \left(\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2}, \frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} \right).$$

Therefore, the proposed method approximates the solution through the augmented neural network, where the network parameters are obtained by solving the minimization problem

$$\mathbf{u}_\theta = \arg \min_{\boldsymbol{\phi}_\theta \in \mathcal{N}^\pm} \mathcal{L}(\boldsymbol{\phi}_\theta). \tag{3.6}$$

By minimizing the proposed loss functional, the neural network approximately satisfies the elasticity equation together with the spring-type jump conditions on the interface and the boundary conditions in a unified manner. Furthermore, the proposed axis-augmented framework enables efficient approximation of discontinuous solutions using a single neural network without requiring mesh generation or explicit interface reconstruction procedures.

4. Error estimates

In this section, we derive error estimates for the proposed method.

4.1. Lemmas and preliminaries

We state the second Korn inequality (see Chapter 11.2 of [24]).

Lemma 4.1 (Second Korn inequality)**.** *Let D be a bounded connected Lipschitz domain. There exists a constant $C = C(D) > 0$ such that*

$$\|\boldsymbol{v}\|_{1,D} \le C \left(\|\boldsymbol{\varepsilon}(\boldsymbol{v})\|_{0,D} + \|\boldsymbol{v}\|_{0,D} \right), \quad \forall \boldsymbol{v} \in [H^1(D)]^2.$$

Next, we need a Poincaré-Friedrichs-type inequality for the broken space $[H^1(\Omega^\pm)]^2$.

Lemma 4.2. *It holds that*

$$\|\mathbf{v}\|_{0,\Omega} \le C(|\mathbf{v}|_{1,\Omega^\pm} + \|[\mathbf{v}]_\Gamma\|_{0,\Gamma} + \|\mathbf{v}\|_{0,\partial\Omega}), \quad \forall \mathbf{v} \in [H^1(\Omega^\pm)]^2,$$

where C is a positive constant depending only on Ω and Γ.

Proof. Let $\mathbf{v}^\pm = \mathbf{v}|_{\Omega^\pm}$. Since $\mathbf{v}^\pm \in \left[H^1(\Omega^\pm)\right]^2$, their traces on Γ are well defined. Under the standing geometric assumptions, $\Omega^\pm$ are bounded Lipschitz domains, and the boundary portions $\partial\Omega \subset \partial\Omega^+$ and $\Gamma = \partial\Omega^-$ have positive boundary measure. Applying the Poincaré–Friedrichs inequality on Ω^+ with the boundary term on $\partial\Omega$, we obtain

$$\left\|\mathbf{v}^+\right\|_{0,\Omega^+} \le C \left(|\mathbf{v}^+|_{1,\Omega^+} + \|\mathbf{v}\|_{0,\partial\Omega} \right). \tag{4.1}$$

By the trace theorem on Ω^+ and (4.1),

$$\left\|\mathbf{v}^+\right\|_{0,\Gamma} \le C \left\|\mathbf{v}^+\right\|_{1,\Omega^+} \le C \left(|\mathbf{v}^+|_{1,\Omega^+} + \|\mathbf{v}\|_{0,\partial\Omega} \right). \tag{4.2}$$

Applying the Poincaré–Friedrichs inequality on Ω^- with the boundary term on Γ, and using $\mathbf{v}^- = [\mathbf{v}]_\Gamma + \mathbf{v}^+$ on Γ, we have

$$\begin{aligned}\left\|\mathbf{v}^-\right\|_{0,\Omega^-} &\le C\left(|\mathbf{v}^-|_{1,\Omega^-} + \left\|\mathbf{v}^-\right\|_{0,\Gamma}\right)\\ &\le C\left(|\mathbf{v}^-|_{1,\Omega^-} + \|[\mathbf{v}]_\Gamma\|_{0,\Gamma} + \left\|\mathbf{v}^+\right\|_{0,\Gamma}\right)\\ &\le C\left(|\mathbf{v}|_{1,\Omega^\pm} + \|[\mathbf{v}]_\Gamma\|_{0,\Gamma} + \|\mathbf{v}\|_{0,\partial\Omega}\right).\end{aligned} \tag{4.3}$$

Combining (4.1) and (4.3) yields

$$\begin{aligned}\|\mathbf{v}\|_{0,\Omega} &\le \left\|\mathbf{v}^-\right\|_{0,\Omega^-} + \left\|\mathbf{v}^+\right\|_{0,\Omega^+}\\ &\le C\left(|\mathbf{v}|_{1,\Omega^\pm} + \|[\mathbf{v}]_\Gamma\|_{0,\Gamma} + \|\mathbf{v}\|_{0,\partial\Omega}\right),\end{aligned} \tag{4.4}$$

which completes the proof. □

We invoke the approximation theorem for the neural network function [25].

Lemma 4.3. *Suppose that $D \subset \mathbb{R}^3$ is a bounded Lipschitz domain and $\boldsymbol{\phi} \in \left[W^{3,\infty}(D)\right]^2$. Then, for any $\epsilon > 0$, there exists a tanh neural network $\boldsymbol{\phi}_\theta \in \mathcal{N}^{\text{aug}}$ with two hidden layers such that*

$$\left\|\boldsymbol{\phi} - \boldsymbol{\phi}_\theta\right\|_{W^{2,\infty}(D)} \le \epsilon.$$

We employ two hidden layers because the approximation theorem in [25], invoked in Lemma 4.3, is established for this architecture.

Before closing this subsection, we define the residuals $\mathcal{R}^\Omega(\mathbf{v})$, $\mathcal{R}^\Gamma(\mathbf{v})$, $\mathcal{R}^\partial(\mathbf{v})$, and $\mathcal{R}(\mathbf{v})$, which play fundamental roles in the analysis:

$$\mathcal{R}^\Omega(\mathbf{v}) = \sum_{s\in\{+,-\}} \|-\nabla\cdot(\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^s}) - \boldsymbol{f}|_{\Omega^s}\|^2_{0,\Omega^s}, \tag{4.5}$$

$$\mathcal{R}^\Gamma(\mathbf{v}) = \|[\mathbf{v}]_\Gamma + R(\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^-})\mathbf{n}_\Gamma\|^2_{0,\Gamma} + \|[\boldsymbol{\sigma}(\mathbf{v})\mathbf{n}_\Gamma]_\Gamma\|^2_{0,\Gamma}, \tag{4.6}$$

$$\mathcal{R}^\partial(\mathbf{v}) = \|\mathbf{v} - \boldsymbol{g}\|^2_{\partial\Omega}, \tag{4.7}$$

$$\mathcal{R}(\mathbf{v}) = \mathcal{R}^\Omega(\mathbf{v}) + \mathcal{R}^\Gamma(\mathbf{v}) + \mathcal{R}^\partial(\mathbf{v}). \tag{4.8}$$

Here and below, $\|\cdot\|_{0,\partial\Omega}$ denotes the $L^2(\partial\Omega)$ norm. Since $\partial\Omega \subset \partial\Omega^+$ and Ω^+ is a bounded Lipschitz domain, the trace theorem gives

$$\|\mathbf{v}\|_{0,\partial\Omega} \le C\,\|\mathbf{v}\|_{1,\Omega^+}.$$

To relate the residual analysis to the weighted training objective, define

$$\mathcal{R}_\nu(\mathbf{v}) = \mathcal{R}^\Omega(\mathbf{v}) + \nu_1\mathcal{R}^\Gamma(\mathbf{v}) + \nu_2\mathcal{R}^\partial(\mathbf{v}).$$

For fixed $\nu_1, \nu_2 > 0$, the weighted and unweighted residuals satisfy

$$m_\nu\mathcal{R}(\mathbf{v}) \le \mathcal{R}_\nu(\mathbf{v}) \le M_\nu\mathcal{R}(\mathbf{v}),$$

where $m_\nu = \min\{1, \nu_1, \nu_2\}$ and $M_\nu = \max\{1, \nu_1, \nu_2\}$. Consequently, the residual-based stability estimate remains valid in terms of $\mathcal{R}_\nu$, with a constant depending on the fixed weights. In particular, this constant may deteriorate if one of the weights approaches zero.

4.2. Energy-norm error estimates

In Theorem 4.4, we show that the difference between the exact solution and the neural network solution can be bounded by the residuals in (4.5)–(4.8). Next, in Theorem 4.7, we show that these residuals can be made small by increasing the number of parameters. Throughout this section, we assume that the optimization procedure reaches a sufficiently small value of the loss function within the considered neural network space. Accordingly, $\boldsymbol{u}_\theta$ denotes the neural network-based approximation obtained by solving the minimization problem (3.6). We define the energy norm as

$$|\!|\!|\boldsymbol{v}|\!|\!|^2 := \|\boldsymbol{v}\|^2_{1,\Omega^\pm} + \|[\boldsymbol{v}]_\Gamma\|^2_{0,\Gamma}\,.$$

Theorem 4.4. *Let $\boldsymbol{u}$ be the solution of (2.1)–(2.4). Suppose that $\boldsymbol{u} \in [H^2(\Omega^\pm)]^2 \cap [C^1(\Omega^\pm)]^2$. Then, the following estimate holds:*

$$|\!|\!|\boldsymbol{u} - \boldsymbol{u}_\theta|\!|\!|^2 \le C\left(\mathcal{R}(\boldsymbol{u}_\theta) + \left(\mathcal{R}^\Gamma(\boldsymbol{u}_\theta)\right)^{\frac{1}{2}} + \left(\mathcal{R}^\partial(\boldsymbol{u}_\theta)\right)^{\frac{1}{2}}\right),$$

where C is a positive constant depending only on Ω, Γ, R, $\|\boldsymbol{u}\|_{C^1(\Omega^\pm)}$, and $\|\boldsymbol{u}_\theta\|_{C^1(\Omega^\pm)}$.

Proof. Let $\mathbf{v} := \boldsymbol{u} - \boldsymbol{u}_\theta \in C^1(\Omega^\pm)$. By Lemmas 4.1 and 4.2, we have

$$\|\mathbf{v}\|^2_{1,\Omega^\pm} + \|[\mathbf{v}]_\Gamma\|^2_{0,\Gamma} \le \widetilde{C}\left(\sum_{s\in\{+,-\}}\int_{\Omega^s} 2\mu^s|\boldsymbol{\varepsilon}(\mathbf{v})|^2 d\mathbf{x} + \sum_{s\in\{+,-\}}\int_{\Omega^s}\lambda^s|\operatorname{div}\mathbf{v}|^2 d\mathbf{x} + \left\|R^{-1/2}\,[\mathbf{v}]_\Gamma\right\|^2_{0,\Gamma} + \|\mathbf{v}\|^2_{0,\partial\Omega}\right). \tag{4.9}$$

We remark that $\tilde{C}$ depends on Ω, Γ, μ, and λ. The last term is $\mathcal{R}^\partial(\boldsymbol{u}_\theta)$, so it suffices to bound the first three terms on the right-hand side, which we denote by A. Using the definition of the stress tensor $\boldsymbol{\sigma}(\cdot)$, we obtain

$$\begin{aligned} A &= \sum_{s\in\{+,-\}}\int_{\Omega^s}\boldsymbol{\sigma}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{v}) + \left\|\left[R^{-1/2}\mathbf{v}\right]_\Gamma\right\|^2_{0,\Gamma} \\ &= \sum_{s\in\{+,-\}}\int_{\Omega^s}\boldsymbol{\sigma}(\mathbf{v}) : \nabla\mathbf{v} + \left\|\left[R^{-1/2}\mathbf{v}\right]_\Gamma\right\|^2_{0,\Gamma}, \end{aligned}$$

where the last equality comes from the symmetry of $\boldsymbol{\sigma}$. Applying integration by parts separately on each subdomain, and noting that $\mathbf{n}^- = \mathbf{n}_\Gamma$ and $\mathbf{n}^+ = -\mathbf{n}_\Gamma$ on Γ, while $\mathbf{n}^+ = \mathbf{n}_\Omega$ on $\partial\Omega$, we obtain

$$\begin{aligned} A &= \sum_{s=+,-}\left(\int_{\partial\Omega^s}(\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^s}\mathbf{n}^s)\cdot\mathbf{v}|_{\Omega^s}\,ds - \int_{\Omega^s}\nabla\cdot(\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^s})\cdot\mathbf{v}|_{\Omega^s}\,d\mathbf{x}\right) + \left\|\left[R^{-1/2}\mathbf{v}\right]_\Gamma\right\|^2_{0,\Gamma} \\ &= -(\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^+}\,\mathbf{n}_\Gamma, \mathbf{v}|_{\Omega^+})_{0,\Gamma} + (\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^-}\,\mathbf{n}_\Gamma, \mathbf{v}|_{\Omega^-})_{0,\Gamma} + \left\|\left[R^{-1/2}\mathbf{v}\right]_\Gamma\right\|^2_{0,\Gamma} \\ &\quad - \sum_{s\in\{-,+\}}(\nabla\cdot\boldsymbol{\sigma}(\mathbf{v}), \mathbf{v})_{0,\Omega^s} + (\boldsymbol{\sigma}(\mathbf{v})\mathbf{n}_\Omega, \mathbf{v})_{0,\partial\Omega}. \end{aligned} \tag{4.10}$$

For a piecewise smooth vector field $\mathbf{v}$, we define the interface averages by

$$\{\boldsymbol{\sigma}(\mathbf{v})\mathbf{n}_\Gamma\}_\Gamma := \frac{1}{2}\left[(\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^-})\mathbf{n}_\Gamma + (\boldsymbol{\sigma}(\mathbf{v})|_{\Omega^+})\mathbf{n}_\Gamma\right], \quad \{\mathbf{v}\}_\Gamma := \frac{1}{2}(\mathbf{v}^- + \mathbf{v}^+).$$

Using these definitions, the identity

$$a^- b^- - a^+ b^+ = \left(\frac{a^- + a^+}{2}\right)(b^- - b^+) + (a^- - a^+)\left(\frac{b^- + b^+}{2}\right),$$

and the interface conditions (2.2) and (2.3), we have:

$$\begin{aligned}
&(\{\boldsymbol{\sigma}(\mathbf{v})\, \mathbf{n}_\Gamma\}_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} + ([\boldsymbol{\sigma}(\mathbf{v})\, \mathbf{n}_\Gamma]_\Gamma, \{\mathbf{v}\}_\Gamma)_{0,\Gamma} + (R^{-1}\, [\mathbf{v}]_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} \\
&= (\{\boldsymbol{\sigma}(\mathbf{v})\, \mathbf{n}_\Gamma\}_\Gamma + R^{-1}\, [\mathbf{v}]_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} + ([\boldsymbol{\sigma}(\mathbf{v})\, \mathbf{n}_\Gamma]_\Gamma, \{\mathbf{v}\}_\Gamma)_{0,\Gamma} \\
&= (\{\boldsymbol{\sigma}(\boldsymbol{u} - \boldsymbol{u}_\theta)\mathbf{n}_\Gamma\}_\Gamma + R^{-1}\, [\boldsymbol{u} - \boldsymbol{u}_\theta]_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} + ([\boldsymbol{\sigma}(\boldsymbol{u} - \boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma, \{\mathbf{v}\}_\Gamma)_{0,\Gamma} \\
&= -(\{\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma\}_\Gamma + R^{-1}[\boldsymbol{u}_\theta]_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} - ([\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma, \{\mathbf{v}\}_\Gamma)_{0,\Gamma}.
\end{aligned} \tag{4.11}$$

Meanwhile, we rewrite the last three terms in (4.10) using conditions (2.1) and (2.4):

$$\sum_{s\in\{-,+\}} (\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}_\theta) + \boldsymbol{f}, \mathbf{v})_{0,\Omega^s} + (\boldsymbol{\sigma}(\mathbf{v})\mathbf{n}_\Omega, \boldsymbol{g} - \boldsymbol{u}_\theta)_{0,\partial\Omega}.$$

Based on the above terms and (4.10)–(4.11), we write A as

$$\begin{aligned}
A &= \sum_{s\in\{-,+\}} (\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}_\theta) + \boldsymbol{f}, \mathbf{v})_{0,\Omega^s} + (\boldsymbol{\sigma}(\mathbf{v})\mathbf{n}_\Omega, \mathbf{v})_{0,\partial\Omega} \\
&\quad - (\{\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma\}_\Gamma + R^{-1}[\boldsymbol{u}_\theta]_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} - ([\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma, \{\mathbf{v}\}_\Gamma)_{0,\Gamma} \\
&=: I_1 + I_2 + I_3 + I_4.
\end{aligned} \tag{4.12}$$

We now bound each I_i. First, we bound I_1 using the Cauchy–Schwarz inequality and Young's inequality (with $\delta_1 > 0$) :

$$\begin{aligned}
|I_1| &\le \sum_{s\in\{-,+\}} \|\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}_\theta) + \boldsymbol{f}\|_{0,\Omega^s}\, \|\mathbf{v}\|_{0,\Omega^s} \\
&\le \sum_{s\in\{-,+\}} \left(\frac{\delta_1}{2}\, \|\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}_\theta) + \boldsymbol{f}\|^2_{0,\Omega^s} + \frac{1}{2\delta_1}\, \|\mathbf{v}\|^2_{0,\Omega^s}\right) \\
&\le \frac{\delta_1}{2}\mathcal{R}^\Omega(\boldsymbol{u}_\theta) + \frac{1}{2\delta_1}\, \|\mathbf{v}\|^2_{1,\Omega^\pm}\, .
\end{aligned} \tag{4.13}$$

Now, we turn to I_2. By the Cauchy–Schwarz inequality and the fact that $\boldsymbol{v} \in C^1(\Omega^\pm)$,

$$|I_2| \le \|\boldsymbol{\sigma}(\mathbf{v})\mathbf{n}_\Omega\|_{0,\partial\Omega}\, \|\boldsymbol{u} - \boldsymbol{u}_\theta\|_{0,\partial\Omega} \le C(\mathcal{R}^\partial(\boldsymbol{u}_\theta))^{\frac{1}{2}}. \tag{4.14}$$

To derive a bound for I_3, we first rearrange terms.

$$I_3 = -(\boldsymbol{\sigma}(\boldsymbol{u}_\theta)^- \mathbf{n}_\Gamma + R^{-1}[\boldsymbol{u}_\theta]_\Gamma, [\mathbf{v}]_\Gamma)_{0,\Gamma} + \frac{1}{2}([\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma, [\mathbf{v}_\Gamma])_{0,\Gamma}.$$

By the triangle and Cauchy-Schwarz inequalities,

$$|I_3| \le \left\|\boldsymbol{\sigma}(\boldsymbol{u}_\theta)^- \mathbf{n}_\Gamma + R^{-1}[\boldsymbol{u}_\theta]_\Gamma\right\|_{0,\Gamma} \|[\mathbf{v}]_\Gamma\|_{0,\Gamma} + \frac{1}{2}\, \|[\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma\|_{0,\Gamma}\, \|[\mathbf{v}]\|_{0,\Gamma}\, .$$

By Young's inequality with parameter $\delta_2, \delta_3 > 0$, we obtain

$$
\begin{aligned}
|I_3| &\leq \frac{\delta_2}{2} \left\| R^{-1}(R\boldsymbol{\sigma}(\boldsymbol{u}_\theta)^{-}\mathbf{n}_\Gamma + [\boldsymbol{u}_\theta]_\Gamma) \right\|_{0,\Gamma}^2 + \frac{1}{2\delta_2} \|[\mathbf{v}]_\Gamma\|_{0,\Gamma}^2 + \frac{\delta_3}{2} \|[\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma\|_{0,\Gamma}^2 + \frac{1}{8\delta_3} \|[\mathbf{v}]\|_{0,\Gamma}^2 \\
&\leq \max\left(\frac{\delta_2}{2\min(\alpha,\beta)^2}, \frac{\delta_3}{2} \right) \mathcal{R}^\Gamma(\boldsymbol{u}_\theta) + \left(\frac{1}{2\delta_2} + \frac{1}{8\delta_3} \right) \|[\mathbf{v}]\|_{0,\Gamma}^2 .
\end{aligned} \tag{4.15}
$$

In the last inequality, we used the bound $\left\| R^{-1} \right\|_2 = 1/\min(\alpha,\beta)$.

We derive the estimate for I_4 similarly. The trace inequalities on both subdomains yield

$$
\|\{\mathbf{v}\}_\Gamma\|_{0,\Gamma} \leq \frac{1}{2}(\left\|\mathbf{v}^-\right\|_{0,\Gamma} + \left\|\mathbf{v}^+\right\|_{0,\Gamma}) \leq C(\left\|\mathbf{v}^-\right\|_{1,\Omega^-} + \left\|\mathbf{v}^+\right\|_{1,\Omega^+}) \leq C \|\mathbf{v}\|_{1,\Omega^\pm} .
$$

Since $\mathbf{v} = \boldsymbol{u} - \boldsymbol{u}_\theta$, we also have $\|\mathbf{v}\|_{1,\Omega^\pm} \leq C(\|\boldsymbol{u}\|_{C^1(\Omega^\pm)} + \|\boldsymbol{u}_\theta\|_{C^1(\Omega^\pm)})$.

$$
\begin{aligned}
|I_4| &= |([\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma, \{\mathbf{v}\}_\Gamma)_{0,\Gamma}| \leq \|[\boldsymbol{\sigma}(\boldsymbol{u}_\theta)\mathbf{n}_\Gamma]_\Gamma\|_{0,\Gamma} \cdot \|\{\mathbf{v}\}_\Gamma\|_{0,\Gamma} \\
&= C(\mathcal{R}^\Gamma(\boldsymbol{u}_\theta))^{\frac{1}{2}},
\end{aligned} \tag{4.16}
$$

where C has the same dependence as in (4.14).

Finally, combining Eqs (4.9), (4.12), and (4.13)–(4.16), we obtain

$$
\|\mathbf{v}\|_{1,\Omega^\pm}^2 + \|[\mathbf{v}]_\Gamma\|_{0,\Gamma}^2 \leq \widetilde{C} \left[\mathcal{R}(\boldsymbol{u}_\theta) + C(\mathcal{R}^\partial(\boldsymbol{u}_\theta))^{\frac{1}{2}} + C(\mathcal{R}^\Gamma(\boldsymbol{u}_\theta))^{\frac{1}{2}} + \frac{1}{2\delta_1} \|\mathbf{v}\|_{1,\Omega^\pm}^2 + \left(\frac{1}{2\delta_2} + \frac{1}{8\delta_3} \right) \|[\mathbf{v}]\|_{0,\Gamma}^2 \right].
$$

We choose $\delta_1 = \tilde{C}, \delta_2 = 2\tilde{C}$, and $\delta_3 = \tilde{C}/2$. Then

$$
\frac{\tilde{C}}{2\delta_1} = \frac{1}{2}, \quad \tilde{C}\left(\frac{1}{2\delta_2} + \frac{1}{8\delta_3} \right) = \frac{1}{2}.
$$

Hence, the corresponding terms can be absorbed into the left-hand side. After adjusting the generic constant C, we obtain

$$
\|\mathbf{v}\|_{1,\Omega^\pm}^2 + \|[\mathbf{v}]_\Gamma\|_{0,\Gamma}^2 \leq C \left[\mathcal{R}(\boldsymbol{u}_\theta) + (\mathcal{R}^\partial(\boldsymbol{u}_\theta))^{\frac{1}{2}} + (\mathcal{R}^\Gamma(\boldsymbol{u}_\theta))^{\frac{1}{2}} \right].
$$

This completes the proof. □

Remark 4.5. *The constant C in the above proof may depend on the piecewise C^1-norm of $\boldsymbol{u}_\theta$. However, provided that $\|\boldsymbol{u}_\theta\|_{C^1(\Omega^\pm)}$ remains bounded, the result shows that the approximation error decreases as the PINN residual decreases. A similar dependence is observed in [26], where a small PINN loss is shown to imply a small L^2-error, provided that the C^1-norm of the neural network does not blow up. In practice, following the approach in [16], an L^2-regularization term for the network parameters may be incorporated into the loss functional to help keep $\|\boldsymbol{u}_\theta\|_{C^1(\Omega^\pm)}$ bounded.*

It remains to prove that the residuals can be made arbitrarily small. We first establish an approximation result for the piecewise neural network space.

Lemma 4.6. *Let $\boldsymbol{u}$ be the solution of* (2.1)–(2.4)*. We assume that $\partial\Omega$ and the interface Γ are Lipschitz so that both subdomains Ω^+ and Ω^- are bounded Lipschitz domains. Suppose that*

$$
\boldsymbol{u} \in [W^{3,\infty}(\Omega^\pm)]^2.
$$

Then, for every $\varepsilon > 0$, there exists $\boldsymbol{\phi}_\theta \in \mathcal{N}^\pm$ such that

$$
\|\boldsymbol{u} - \boldsymbol{\phi}_\theta\|_{W^{2,\infty}(\Omega^\pm)} \leq \varepsilon.
$$

Proof. Each Ω^s, $s \in \{+, -\}$, admits a bounded linear extension operator

$$E^s : [W^{3,\infty}(\Omega^s)]^2 \to [W^{3,\infty}(\mathbb{R}^2)]^2. \tag{4.17}$$

We define $\bar{\boldsymbol{u}}^S := E^s(\boldsymbol{u}|_{\Omega^s})|_{\Omega}$. Then

$$\bar{\boldsymbol{u}}^s = \boldsymbol{u} \text{ in } \Omega^s, \quad \bar{\boldsymbol{u}}^s \in [W^{3,\infty}(\Omega)]^2,$$

and

$$\|\bar{\boldsymbol{u}}^s\|_{W^{3,\infty}(\Omega)} \leq C \|\boldsymbol{u}\|_{W^{3,\infty}(\Omega^s)}, \quad s \in \{+, -\},$$

where $C > 0$ depends only on the Lipschitz geometry of $\Omega^{\pm}$. Define

$$\boldsymbol{u}^{\text{aug}}(x, z) = \frac{1+z}{2}\bar{\boldsymbol{u}}^+(x) + \frac{1-z}{2}\bar{\boldsymbol{u}}^-(x).$$

Then

$$\boldsymbol{u}^{\text{aug}} \in [W^{3,\infty}(\Omega \times [-1, 1])]^2.$$

By Lemma 4.3, there exists $\boldsymbol{\phi}_\theta^{\text{aug}} \in \mathcal{N}^{\text{aug}}$ such that

$$\|\boldsymbol{u}^{\text{aug}} - \boldsymbol{\phi}_\theta^{\text{aug}}\|_{W^{2,\infty}(\Omega\times[-1,1])} \leq \varepsilon.$$

Restricting this inequality to the flat slices $z = -1$ and $z = 1$ gives

$$\|\boldsymbol{u} - \boldsymbol{\phi}_\theta^{\text{aug}}(\cdot, -1)\|_{W^{2,\infty}(\Omega^-)} + \|\boldsymbol{u} - \boldsymbol{\phi}_\theta^{\text{aug}}(\cdot, 1)\|_{W^{2,\infty}(\Omega^+)} \leq C\varepsilon.$$

By the definition of $\mathcal{N}^{\pm}$, the left-hand side is

$$\|\boldsymbol{u} - \boldsymbol{\phi}_\theta\|_{W^{2,\infty}(\Omega^{\pm})}.$$

This completes the proof. □

Theorem 4.7. *Let $\boldsymbol{u}$ be the solution of (2.1)–(2.4). Suppose that $\boldsymbol{u} \in [W^{3,\infty}(\Omega^{\pm})]^2$. For any $\epsilon > 0$, there exists a function $\boldsymbol{u}_\theta \in \mathcal{N}^{\pm}$ with two hidden layers such that*

$$\mathcal{R}(\boldsymbol{u}_\theta) \leq \epsilon.$$

Proof. Let $\delta > 0$. By Lemma 4.6, there exists $u_\theta \in \mathcal{N}^{\pm}$ such that

$$\|\boldsymbol{u} - \boldsymbol{u}_\theta\|_{W^{2,\infty}(\Omega^{\pm})} \leq \delta.$$

Since $\boldsymbol{u}$ satisfies (2.1) and the Lamé parameters λ^s and μ^s are fixed bounded constants in each subdomain Ω^s, $s \in \{+, -\}$, the constitutive relation yields

$$\mathcal{R}^{\Omega}(\boldsymbol{u}_\theta) = \sum_{s\in\{+,-\}} \|\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u} - \boldsymbol{u}_\theta)\|_{0,\Omega^s}^2 \leq C \|\boldsymbol{u} - \boldsymbol{u}_\theta\|_{W^{2,\infty}(\Omega^{\pm})}^2 \leq C\delta^2,$$

where C depends on the domain and the Lamé parameters.

Moreover, by the interface and boundary conditions,

$$\mathcal{R}_\Gamma(\boldsymbol{u}_\theta) \leq C\delta^2, \qquad \mathcal{R}_\partial(\boldsymbol{u}_\theta) \leq C\delta^2.$$

Therefore,

$$\mathcal{R}(\boldsymbol{u}_\theta) \leq C\delta^2.$$

Choosing $\delta = (\varepsilon/C)^{1/2}$ completes the proof. □

Let us close this section with the following remark.

Remark 4.8. *The above analysis concerns the approximation capability of the proposed neural network space and assumes that the optimization procedure reaches a sufficiently small value of the physics-informed loss. In particular, Theorem 4.7 guarantees the existence of neural network functions with arbitrarily small residuals, but does not guarantee that a specific optimization algorithm reaches such functions. In practice, optimization algorithms such as Adam or limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [27] have been widely used for PINNs and have been reported to effectively reduce physics-informed losses; see [28] as the references therein. In this work, we do not analyze the convergence of the optimization process itself. Instead, the numerical experiments demonstrate that the adopted optimization procedure yields accurate approximations.*

5. Results

In this section, we present the numerical results to validate the performance of the proposed method. Experiments are conducted on the domain $\Omega = (-1,1)^2$, which is separated by a level-set function $L(\boldsymbol{x})$, i.e., $\Omega^- = \{\boldsymbol{x}|L(\boldsymbol{x}) < 0\}$ and $\Omega^+ = \{\boldsymbol{x}|L(\boldsymbol{x}) > 0\}$. For the neural network functions in (3.2), we use two hidden layers ($\ell = 2$) with uniform width w. The training samples consist of collocation points drawn from the interior of the domain, the boundary, and the interface. Specifically, we use r^2 interior points based on Chebyshev–Gauss nodes [29], $4r$ boundary points, and $4r$ interface points. For the optimization, the L-BFGS algorithm is employed with a learning rate of 0.1 and 10,000 L-BFGS closure evaluations. The proposed PINN is implemented using PyTorch and trained in a CPU-only environment using an Intel Core i9-14900K processor.

We consider four examples motivated by [30]. In Example 1, we consider a circular interface. We determine suitable hyperparameters for the proposed PINN, including ν_i, w, and r. Next, Example 2 considers a line-interface problem and reports the numerical results for the proposed PINN. Example 3 presents a comparison between the proposed PINN and the piecewise PINN [31] for the same elasticity interface problem. Finally, in Example 4, we consider a driven cavity problem without an exact solution and evaluate the proposed PINN by comparing its results with the immersed finite element method [32] (IFEM) solution.

Example 1 (Hyperparameter sensitivity for a circular interface)**.** *We consider the following exact solution:*

$$u_1 = \begin{cases} \frac{x^2-y^2}{\mu^+} + \left(\frac{1}{\mu^+} - \frac{1}{\mu^-}\right)\cdot\frac{r_0^2}{2} & \text{for } L(x,y) \geq 0, \\ \frac{x^2-y^2}{\mu^-} & \text{for } L(x,y) < 0, \end{cases}$$

$$u_2 = \begin{cases} \frac{x^2-y^2}{\mu^+} - \left(\frac{1}{\mu^+} - \frac{1}{\mu^-}\right)\cdot\frac{r_0^2}{2} & \text{for } L(x,y) \geq 0, \\ \frac{x^2-y^2}{\mu^-} & \text{for } L(x,y) < 0, \end{cases}$$

with level set function $L(x,y) = x^2 + y^2 - r_0^2$. Here, the parameters are

$$\mu^+ = 1, \mu^- = 10, \lambda^+ = 1, \lambda^- = 10, r_0 = 0.46, \alpha = \beta = 0.1035.$$

We vary the loss weights ν_1 and ν_2 in (3.5) and compare the resulting L^∞, L^2 errors and computational time in Table 1 with fixed $r = 16$ and $w = 8$. The results suggest that the approximation accuracy is sensitive to the choice of the loss weights, with a larger boundary loss weight generally leading to smaller approximation errors. Based on these results, we fix $\nu_1 = 1$ and $\nu_2 = 100$.

Table 1. The L^∞ and L^2 errors obtained for different choices of ν_1 and ν_2, with $\nu_0 = 1$, in Example 1.

ν_1	ν_2	L^∞ error	L^2 error	Computational time (s)
1	1	$3.75 \cdot 10^{-3}$	$5.85 \cdot 10^{-3}$	96.71
	10	$1.61 \cdot 10^{-3}$	$3.50 \cdot 10^{-3}$	97.13
	100	$\mathbf{1.03 \cdot 10^{-4}}$	$\mathbf{1.75 \cdot 10^{-4}}$	96.76
10	1	$1.04 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$	96.88
	10	$7.60 \cdot 10^{-4}$	$1.25 \cdot 10^{-3}$	96.92
	100	$7.40 \cdot 10^{-4}$	$9.29 \cdot 10^{-4}$	96.66
100	1	$6.21 \cdot 10^{-1}$	$1.34 \cdot 10^{+0}$	96.93
	10	$3.66 \cdot 10^{-3}$	$5.30 \cdot 10^{-3}$	96.95
	100	$4.91 \cdot 10^{-4}$	$8.75 \cdot 10^{-4}$	96.87

Next, we examine suitable choices of the parameters w and r. Table 2 reports the resulting L^∞ and L^2 errors together with the computational times. As w and r increase, the computational time increases, whereas the approximation accuracy does not show a consistent improvement. The smallest errors are achieved with $r = 16$ and $w = 8$, which are therefore adopted in the remaining experiments.

Table 2. The L^∞ and L^2 errors obtained for different choices of r and w, with $\nu_0 = \nu_1 = 1$ and $\nu_2 = 100$.

r	w	L^∞ error	L^2 error	Computational time (s)
8	8	$1.50 \cdot 10^{-4}$	$2.35 \cdot 10^{-4}$	92.80
	16	$3.05 \cdot 10^{-4}$	$3.53 \cdot 10^{-4}$	94.55
	32	$2.43 \cdot 10^{-4}$	$2.63 \cdot 10^{-4}$	99.34
16	8	$\mathbf{1.03 \cdot 10^{-4}}$	$\mathbf{1.75 \cdot 10^{-4}}$	96.76
	16	$2.28 \cdot 10^{-4}$	$3.08 \cdot 10^{-4}$	143.51
	32	$1.42 \cdot 10^{-4}$	$2.37 \cdot 10^{-4}$	216.17
32	8	$2.20 \cdot 10^{-4}$	$3.13 \cdot 10^{-4}$	192.33
	16	$4.36 \cdot 10^{-4}$	$4.14 \cdot 10^{-4}$	248.39
	32	$2.03 \cdot 10^{-4}$	$3.36 \cdot 10^{-4}$	298.24

The exact solution $\boldsymbol{u} = (u_1, u_2)$, the PINN-generated solution $\boldsymbol{u}_\theta = (u_{1,\theta}, u_{2,\theta})$, and the absolute error $|\boldsymbol{u} - \boldsymbol{u}_\theta| = (|u_1 - u_{1,\theta}|, |u_2 - u_{2,\theta}|)$ are shown in Figure 1. The proposed PINN accurately reproduces the exact solution, with small absolute errors observed over the domain.

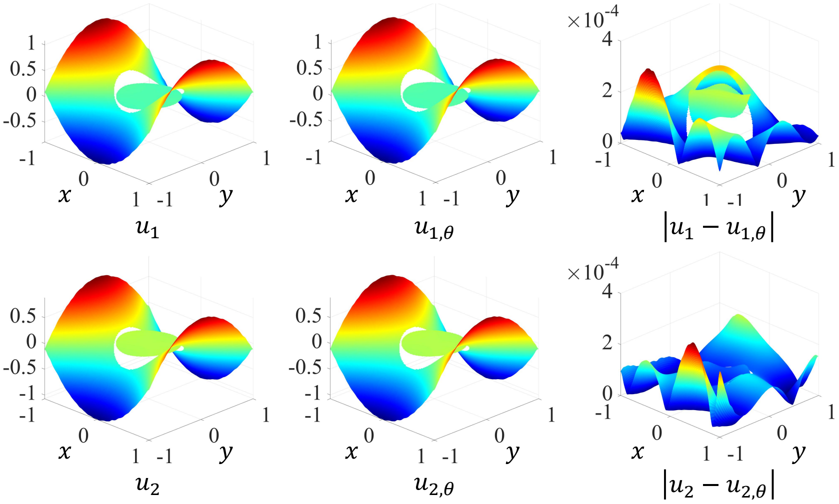


Figure 1. Graphs of $\boldsymbol{u}$ (left), $\boldsymbol{u}_\theta$ (middle), and the absolute difference $|\boldsymbol{u} - \boldsymbol{u}_\theta|$ (right) in Example 1. The first row corresponds to the first displacement component (u_1), and the second row corresponds to the second displacement component (u_2).

Example 2 (Line interface problems with varying regularity)**.** *We consider the line interface with level set function $L(x, y) = x - y - c$. To construct exact solutions satisfying the spring-type jump conditions, we define*

$$c^* = \frac{\sqrt{2}\alpha}{1/\mu^+ - 1/\mu^-} \quad \text{and} \quad P(x, y) = x^2 - y^2 + (c^* - c)(x + y).$$

The exact solution is

$$u_1 = \begin{cases} \frac{P(x,y) + \sqrt{2}c^*\beta\lambda^+}{\mu^+} + \frac{\varepsilon}{\mu^+}\left|\frac{L(x,y)}{\sqrt{2}}\right|^\gamma & \textit{for } L(x, y) \geq 0, \\ \frac{P(x,y)}{\mu^-} + \frac{\varepsilon}{\mu^-}\left|\frac{L(x,y)}{\sqrt{2}}\right|^\gamma & \textit{for } L(x, y) < 0, \end{cases}$$

$$u_2 = \begin{cases} \frac{P(x,y) - \sqrt{2}c^*\beta\lambda^+}{\mu^+} - \frac{\varepsilon}{\mu^+}\left|\frac{L(x,y)}{\sqrt{2}}\right|^\gamma & \textit{for } L(x, y) \geq 0, \\ \frac{P(x,y)}{\mu^-} - \frac{\varepsilon}{\mu^-}\left|\frac{L(x,y)}{\sqrt{2}}\right|^\gamma & \textit{for } L(x, y) < 0. \end{cases}$$

To evaluate the performance of solution regularity on approximation performance, we consider three cases with different piecewise regularities. Case 1 is the high regularity case with $\varepsilon = 0$; the perturbation term vanishes, and the solution belongs to $[W^{3,\infty}(\Omega^\pm)]^2$. Case 2 is the reduced-regularity case with $\varepsilon = 1$ and $\gamma = 2.5$; the solution belongs to $[H^2(\Omega^\pm)]^2$ but not to $[W^{3,\infty}(\Omega^\pm)]^2$. Case 3 is the low-regularity case with $\varepsilon = 1$ and $\gamma = 1.5$; the solution belongs to $[H^1(\Omega^\pm)]^2$ but not to $[H^2(\Omega^\pm)]^2$. A detailed justification of these regularity properties is provided in Appendix A.

For all cases, the parameters are fixed to

$$c = 0.06,\ \mu^+ = 1,\ \mu^- = 10,\ \lambda^+ = 4,\ \lambda^- = 40,\ \alpha = \frac{c}{\sqrt{2}}\left(\frac{1}{\mu^+} - \frac{1}{\mu^-}\right),\ \beta = 0.08.$$

Table 3 reports the L^∞ and L^2 errors for the three cases with different piecewise regularities. The results show that both errors increase as the piecewise regularity decreases. Figure 2 shows the *componentwise absolute errors and confirms the progressive deterioration in accuracy as the regularity is reduced. These results demonstrate that the regularity assumptions are important for the reliable performance of the proposed PINN and that the method is not uniformly robust outside the regularity regime covered by the present theoretical analysis.*

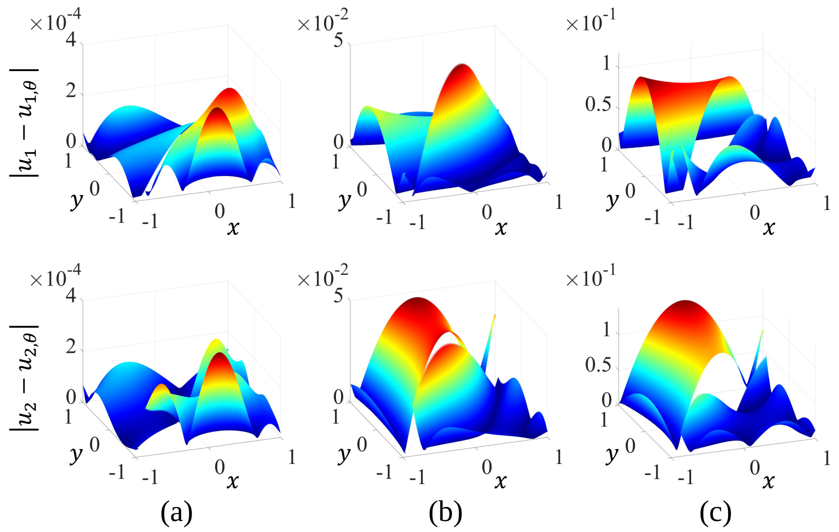


Figure 2. Graphs of the componentwise absolute errors for the three regularity cases in Example 2: (a) $\boldsymbol{u} \in [W^{3,\infty}(\Omega^{\pm})]^2$, (b) $\boldsymbol{u} \in [H^2(\Omega^{\pm})]^2 \setminus [W^{3,\infty}(\Omega^{\pm})]^2$, and (c) $\boldsymbol{u} \in [H^1(\Omega^{\pm})]^2 \setminus [H^2(\Omega^{\pm})]^2$. The first and second rows correspond to the first and second displacement components, respectively.

Table 3. The L^∞ and L^2 errors of the proposed PINN for the three regularity cases in Example 2. The regularity statements are understood componentwise on $\Omega^{\pm}$.

	Piecewise Regularity	L^∞ error	L^2 error
Case 1	$[W^{3,\infty}(\Omega^{\pm})]^2$	$4.86 \cdot 10^{-4}$	$1.51 \cdot 10^{-4}$
Case 2	$[H^2(\Omega^{\pm})]^2$, not $[W^{3,\infty}(\Omega^{\pm})]^2$	$7.35 \cdot 10^{-2}$	$2.95 \cdot 10^{-2}$
Case 3	$[H^1(\Omega^{\pm})]^2$, not $[H^2(\Omega^{\pm})]^2$	$1.41 \cdot 10^{-1}$	$7.94 \cdot 10^{-2}$

Example 3 (Comparison with the piecewise PINN)**.** *We consider the following level set function and exact solution:*

$$L(x,y) = \frac{x}{2} - y - c,$$

$$u_1 = \begin{cases} \frac{\left(\frac{x}{2}-y\right)^2}{\mu^+} - \left(\frac{1}{\mu^+} - \frac{1}{\mu^-}\right)c^2 + \alpha c \frac{6}{\sqrt{5}} & \text{for } L(x,y) \geq 0, \\ \frac{\left(\frac{x}{2}-y\right)^2}{\mu^-} & \text{for } L(x,y) < 0, \end{cases}$$

$$u_2 = \begin{cases} \frac{\left(\frac{x}{2}-y\right)^2}{\mu^+} - \left(\frac{1}{\mu^+} - \frac{1}{\mu^-}\right)c^2 + \alpha c \frac{3}{\sqrt{5}} & \text{for } L(x,y) \geq 0, \\ \frac{\left(\frac{x}{2}-y\right)^2}{\mu^-} & \text{for } L(x,y) < 0. \end{cases}$$

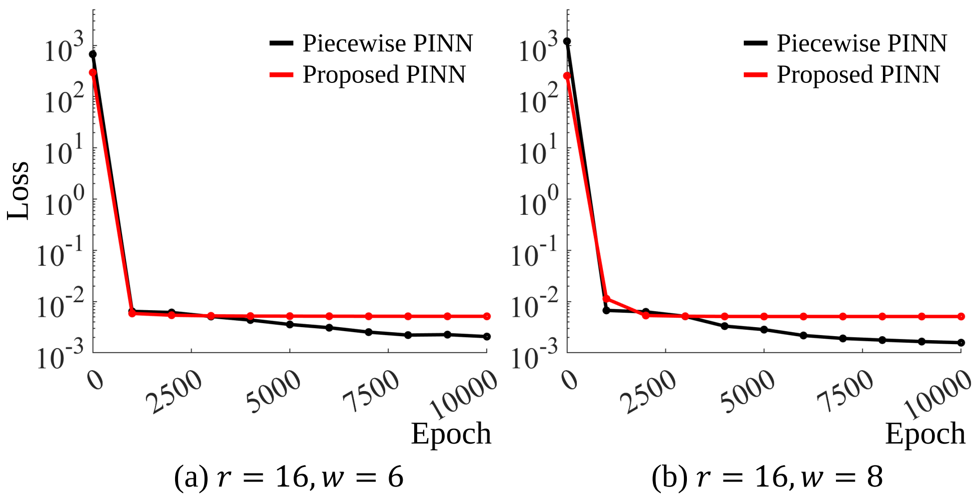
Here, the parameters are $\alpha = 0.6, \beta = 0.05, \mu^+ = \lambda^+ = 1, \mu^- = \lambda^- = 100$, *and* $c = 0.79$.

The proposed PINN is compared with the piecewise PINN [31] under the same sampling and training settings. The piecewise PINN employs two independent subnetworks, one defined on each subdomain. We use $r = 16$ *and* $w = 6, 8$ *for both methods. Table 4 reports the number of trainable parameters, relative errors, peak process resident set size (RSS), and computational time. The two methods show similar memory usage, whereas the piecewise PINN requires more computational time for both values of* w*. The proposed PINN also yields smaller* L^∞ *and* L^2 *errors.*

Table 4. Scaling study of the piecewise PINN [31] and proposed PINN for Example 3 with $r = 16$ and $w \in \{6, 8\}$. The relative L^∞ and L^2 errors, number of trainable parameters, peak process RSS, and computational time are reported.

Method	w	# of Param.	L^∞ error	L^2 error	Peak RSS (MiB)	Computational time (s)
piecewise PINN [31]	6	148	$3.06 \cdot 10^{-2}$	$2.85 \cdot 10^{-2}$	660.48	166.84
	8	228	$3.27 \cdot 10^{-2}$	$3.21 \cdot 10^{-2}$	660.48	168.23
proposed PINN	6	80	$4.08 \cdot 10^{-3}$	$6.37 \cdot 10^{-3}$	657.91	98.39
	8	122	$3.95 \cdot 10^{-3}$	$6.55 \cdot 10^{-3}$	657.94	99.33

Figure 3 presents the loss histories over 10,000 *L-BFGS closure evaluations. Black and red lines represent the losses of the piecewise and proposed PINNs, respectively. The losses decrease rapidly for both methods; however, the proposed PINN reaches a stable plateau earlier, whereas the piecewise PINN exhibits a more gradual decrease. Finally, Figure 4 compares the exact solution with the proposed PINN solution for* $r = 16$ *and* $w = 8$*, together with the corresponding pointwise absolute errors. The results demonstrate good agreement between the exact and predicted solutions.*



(a) $r = 16, w = 6$ (b) $r = 16, w = 8$

Figure 3. Graphs of loss histories of the piecewise and proposed PINNs in Example 3 for (a) $r = 16$ and $w = 6$ and (b) $r = 16$ and $w = 8$).

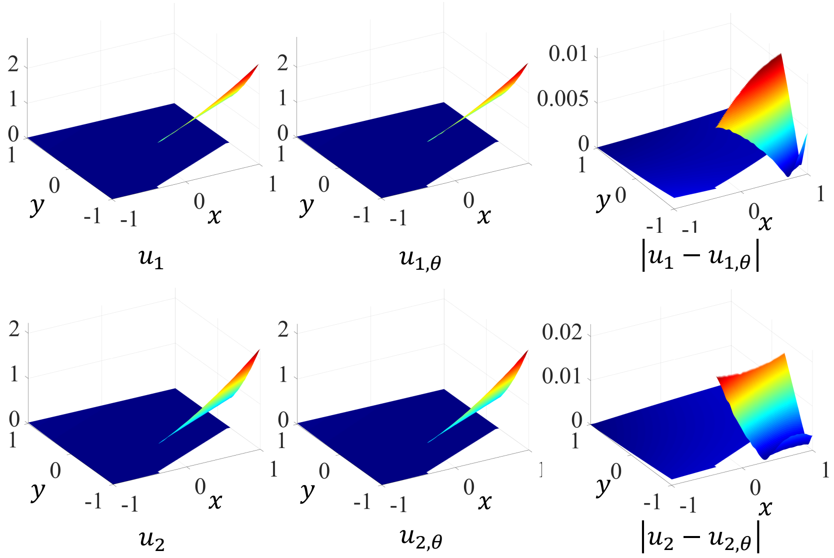


Figure 4. Graphs of $\boldsymbol{u}$ (left), $\boldsymbol{u}_\theta$ (middle), and the absolute difference $|\boldsymbol{u} - \boldsymbol{u}_\theta|$ (right) in Example 3 for $r = 16$ and $w = 8$. The first row corresponds to the first displacement component (u_1), and the second row corresponds to the second displacement component (u_2).

Example 4 (Driven-cavity problem with complex interfaces)**.** *Finally, we consider a driven-cavity problem for which an exact solution is not available. The elasticity interface problem is defined by the body force* $\boldsymbol{f} = (1, 0)^T$ *and the boundary condition* $\mathbf{g} = (g_1,\ g_2)^T$ *given by*

$$g_1 = \begin{cases} -(x-1)(x+1), & y = 1 \\ 0, & o.w. \end{cases} \quad \textit{and} \quad g_2 = 0,$$

where $\mu^+ = \lambda^+ = 1$, $\mu^- = \lambda^- = 100$, *and* $\alpha = \beta = 0.05$. *The interface is represented by the level set function*

$$L(x, y) = \sqrt{x^2 + y^2} - 0.4\left(1 + \frac{1}{7}\sin 6\theta\right), \quad \theta = \mathrm{atan2}(y, x).$$

Since the analytical solution is unavailable, the IFEM solution is used as a numerical reference. For this interface geometry, we compute the reference solution using the piecewise-linear Crouzeix–Raviart-type IFEM described in [30, 32], for which stability and optimal-order convergence properties have been reported. A uniform interface-unfitted triangulation with $\Delta x = \Delta y = 1/31$ *is employed, consisting of* 7,688 *triangular elements and* 23,312 *displacement degrees of freedom. To accommodate the geometric complexity of the interface, the proposed PINN was trained using* $r = 32$ *and* $w = 16$.

The L^2 *difference between the proposed PINN and the IFEM solution is* $\|\boldsymbol{u}_\theta - \boldsymbol{u}_{\mathrm{IFEM}}\|_{L^2(\Omega)} = 7.76 \times 10^{-3}$. *This result numerically supports the applicability of the proposed framework to a driven-cavity problem involving a geometrically complex interface.*

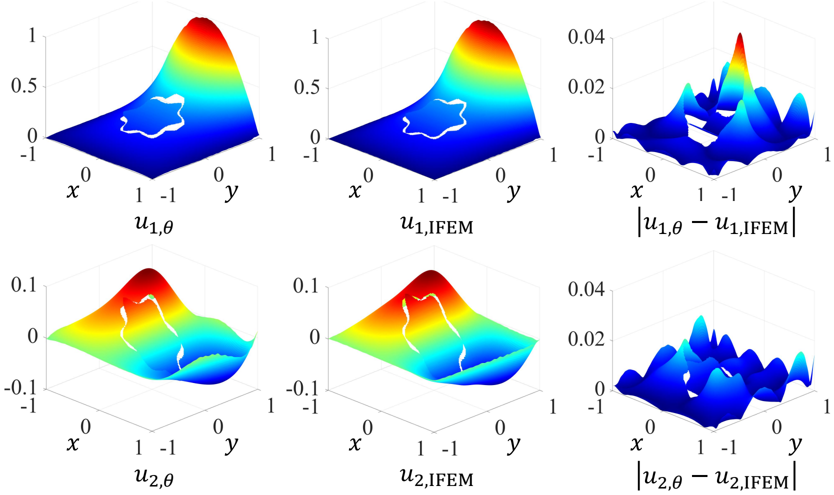


Figure 5. Graphs of $\boldsymbol{u}_\theta$ (left), $\boldsymbol{u}_{\mathrm{IFEM}}$ (middle), and the absolute difference $|\boldsymbol{u}_\theta - \boldsymbol{u}_{\mathrm{IFEM}}|$ (right) for the case with the six-lobed interface in Example 4. The first row corresponds to the first displacement component (u_1), and the second row corresponds to the second displacement component (u_2).

6. Conclusions and discussion

In this paper, we proposed an axis-augmented physics-informed neural network for elasticity interface problems with spring-type jump conditions. By introducing Sobolev extensions and an augmented representation, the discontinuous displacement field was transformed into a continuous function, allowing a single neural network to approximate the interface solution. We also established an error estimate for the proposed loss function and demonstrated through several numerical examples that the proposed method provides accurate approximations for elasticity interface problems. The present analysis is restricted to the compressible case under sufficient regularity assumptions. Another limitation is that the convergence of the neural network optimization process is not analyzed, and theoretical guarantees for reaching the small-residual regime remain for future investigation. In particular, the additional interface residuals may affect the optimization landscape compared with standard PINNs, and whether they introduce additional difficulties in reaching a small-residual regime remains to be investigated. Theoretical analysis of the optimization process for such interface problems will be considered in future work.

Author contributions

Bokyu Kim: Formal analysis, Methodology, Validation, Writing–original draft; Dongsik Jo: Validation, Writing–review and editing; Gwanghyun Jo: Conceptualization, Methodology, Writing–review and editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

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

Acknowledgments

The second author (D. Jo) was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program - Shipbuilding and Offshore Industry Core Technology Development Program) (RS-2024-00508239, Development of a welding simulator for shipyard worker training based on eXtended reality) funded by the Ministry of Trade, Industry & Energy (MOTIE, Republic of Korea). The third author (G. Jo) was supported by an NRF grant funded by MSIT (No. RS-2025-16068758).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. I. N. Sneddon, D. S. Berry, The classical theory of elasticity, In: *Elasticity and Plasticity/Elastizität und Plastizität*, Berlin: Springer, 1958, 1–126. https://doi.org/10.1007/978-3-642-45887-3_1
2. Z. Hashin, The spherical inclusion with imperfect interface, *J. Appl. Mech.*, **58** (1991), 444–449. https://doi.org/10.1115/1.2897205
3. J. M. Chen, J. Z. Tong, Y. B. Chen, Elastic properties of particulate composites with imperfect interface, *J. Wuhan Univ. Technol.-Mater. Sci. Ed.*, **17** (2002), 78–82. https://doi.org/10.1007/BF02832629
4. Z. Hashin, Thin interphase/imperfect interface in elasticity with application to coated fiber composites, *J. Mech. Phys. Solids*, **50** (2002), 2509–2537. https://doi.org/10.1016/S0022-5096(02)00050-9
5. Y. Benveniste, A general interface model for a three-dimensional curved thin anisotropic interphase between two anisotropic media, *J. Mech. Phys. Solids*, **54** (2006), 708–734. https://doi.org/10.1016/j.jmps.2005.10.009
6. A. Hansbo, P. Hansbo, A finite element method for the simulation of strong and weak discontinuities in solid mechanics, *Comput. Methods Appl. Mech. Engrg.*, **193** (2004), 3523–3540. https://doi.org/10.1016/j.cma.2003.12.041
7. J. A. Nairn, Numerical implementation of imperfect interfaces, *Comput. Mater. Sci.*, **40** (2007), 525–536. https://doi.org/10.1016/j.commatsci.2007.02.010
8. Q. Z. Zhu, S. T. Gu, J. Yvonnet, J. F. Shao, Q. C. He, Three-dimensional numerical modelling by XFEM of spring-layer imperfect curved interfaces with applications to linearly elastic composite materials, *Int. J. Numer. Methods Eng.*, **88** (2011), 307–328. https://doi.org/10.1002/nme.3175

9. E. Benvenuti, G. Ventura, N. Ponara, A. Tralli, Variationally consistent eXtended FE model for 3D planar and curved imperfect interfaces, *Comput. Methods Appl. Mech. Engrg.*, **267** (2013), 434–457. https://doi.org/10.1016/j.cma.2013.08.013

10. N. Garg, N. D. Chakladar, B. G. Prusty, C. Song, A. W. Phillips, Modelling of laminated composite plates with weakly bonded interfaces using scaled boundary finite element method, *Int. J. Mech. Sci.*, **170** (2020), 105349. https://doi.org/10.1016/j.ijmecsci.2019.105349

11. N. Garg, M. J. Donough, C. Song, A. W. Phillips, B. G. Prusty, Three-dimensional scaled boundary finite element method to study interfacial imperfections in thick laminated composite plates undergoing bi-axial bending, *Eng. Anal. Bound. Elem.*, **146** (2023), 34–43. https://doi.org/10.1016/j.enganabound.2022.10.003

12. T. P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, *Int. J. Numer. Methods Eng.*, **84** (2010), 253–304. https://doi.org/10.1002/nme.2914

13. J. Hao, D. Zhou, Q. Li, D. Huang, Beyond hard constraint: Unified knowledge-embedding physics-informed neural networks for multi-domain system, *Neural Netw.*, **197** (2026), 108499. https://doi.org/10.1016/j.neunet.2025.108499

14. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

15. X. Zhu, X. Hu, P. Sun, Physics-informed neural networks for solving dynamic two-phase interface problems, *SIAM J. Sci. Comput.*, **45** (2023), A2912–A2944. https://doi.org/10.1137/22M1517081

16. S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating PDEs, *IMA J. Numer. Anal.*, **43** (2023), 1–43. https://doi.org/10.1093/imanum/drab093

17. M. Zeinhofer, R. Masri, K. A. Mardal, A unified framework for the error analysis of physics-informed neural networks, *IMA J. Numer. Anal.*, **45** (2025), 2988–3025. https://doi.org/10.1093/imanum/drae081

18. A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M. W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 26548–26560.

19. S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, *J. Comput. Phys.*, **449** (2022), 110768. https://doi.org/10.1016/j.jcp.2021.110768

20. W. F. Hu, T. S. Lin, M. C. Lai, A discontinuity capturing shallow neural network for elliptic interface problems, *J. Comput. Phys.*, **469** (2022), 111576. https://doi.org/10.1016/j.jcp.2022.111576

21. H. Park, G. Jo, A physics-informed neural network based method for the nonlinear Poisson–Boltzmann equation and its error analysis, *J. Comput. Phys.*, **522** (2025), 113579. https://doi.org/10.1016/j.jcp.2024.113579

22. H. Oh, G. Jo, Physics-informed neural network for the heat equation under imperfect contact conditions and its error analysis, *AIMS Math.*, **10** (2025), 7920–7940. https://doi.org/10.3934/math.2025364

23. L. C. Evans, *Partial Differential Equations*, 2 Eds, Providence, RI: American Mathematical Society, 2022.

24. S. C. Brenner, L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 3 Eds, New York: Springer, 2008. https://doi.org/10.1007/978-0-387-75934-0

25. T. De Ryck, S. Lanthaler, S. Mishra, On the approximation of functions by tanh neural networks, *Neural Netw.*, **143** (2021), 732–750. https://doi.org/10.1016/j.neunet.2021.08.015

26. T. De Ryck, A. D. Jagtap, S. Mishra, Error estimates for physics-informed neural networks approximating the Navier–Stokes equations, *IMA J. Numer. Anal.*, **44** (2024), 83–119. https://doi.org/10.1093/imanum/drac085

27. C. Zhu, R. H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, *ACM Trans. Math. Softw.*, **23** (1997), 550–560. https://doi.org/10.1145/279232.279236

28. F. V. Difonzo, L. Lopez, S. F. Pellergrino, Physics informed neural networks for learning the horizon size in bond-based peridynamic models, *Comput. Methods Appl. Mech. Eng.*, **436** (2025), 117727. https://doi.org/10.1016/j.cma.2024.117727

29. J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2 Eds, Mineola: Dover Publications, 2001.

30. D. Kyeong, D. Y. Kwak, An immersed finite element method for the elasticity problems with displacement jump, *Adv. Appl. Math. Mech.*, **9** (2017), 407–428. https://doi.org/10.4208/aamm.2016.m1427

31. C. He, X. Hu, L. Mu, A mesh-free method using piecewise deep neural network for elliptic interface problems, *J. Comput. Appl. Math.*, **412** (2022), 114358. https://doi.org/10.1016/j.cam.2022.114358

32. G. Jo, D. Y. Kwak, Recent development of immersed FEM for elliptic and elastic interface problems, *J. Korean Soc. Ind. Appl. Math.*, **23** (2019), 65–92.

A. Piecewise regularity of the exact solutions in Example 2

This appendix provides the justification for the piecewise regularity statements used in Example 2. Let $\boldsymbol{u}_{\varepsilon,\gamma} = (u_1, u_2)^T$ and $\boldsymbol{u}^s_{\varepsilon,\gamma} = \boldsymbol{u}_{\varepsilon,\gamma}|_{\Omega^s}$. On the two subdomains, the exact solution in Example 2 can be written as

$$\begin{aligned}
\boldsymbol{u}^+_{\varepsilon,\gamma} &= \frac{P(x,y)}{\mu^+}\begin{pmatrix}1\\1\end{pmatrix} + \frac{\sqrt{2}c^*\beta\lambda^+}{\mu^+}\begin{pmatrix}1\\-1\end{pmatrix} + \varepsilon|t|^\gamma\begin{pmatrix}1\\-1\end{pmatrix},\\
\boldsymbol{u}^-_{\varepsilon,\gamma} &= \frac{P(x,y)}{\mu^-}\begin{pmatrix}1\\1\end{pmatrix} + \varepsilon|t|^\gamma\begin{pmatrix}1\\-1\end{pmatrix},
\end{aligned}$$

where $t = L(x,y)/\sqrt{2}$. Since $P(x,y)$ is a polynomial, all terms except the perturbation $\varepsilon|t|^\gamma(1,-1)^T$ are smooth on the corresponding subdomains. Therefore, the piecewise regularity of $\boldsymbol{u}_{\varepsilon,\gamma}$ is determined by $|t|^\gamma$ near the interface Γ.

In Case 1, $\varepsilon = 0$, so the singular perturbation vanishes. Consequently,

$$\boldsymbol{u}_{0,\gamma} \in [W^{3,\infty}(\Omega^{\pm})]^2.$$

In Case 2, $\varepsilon = 1$ and $\gamma = 2.5$. For $t \neq 0$, we have

$$\frac{d}{dt}|t|^{5/2} = \frac{5}{2}\,\mathrm{sgn}(t)|t|^{3/2}, \qquad \frac{d^2}{dt^2}|t|^{5/2} = \frac{15}{4}|t|^{1/2}.$$

Hence, $|t|^{5/2}$ and its derivatives up to order two are bounded near $t = 0$. It follows that

$$\boldsymbol{u}_{1,2.5} \in [W^{2,\infty}(\Omega^{\pm})]^2 \subset [H^2(\Omega^{\pm})]^2.$$

However, the third derivative satisfies

$$\frac{d^3}{dt^3}|t|^{5/2} = \frac{15}{8}\,\mathrm{sgn}(t)|t|^{-1/2},$$

which is unbounded as $t \to 0$. Therefore,

$$\boldsymbol{u}_{1,2.5} \notin [W^{3,\infty}(\Omega^{\pm})]^2.$$

In Case 3, $\varepsilon = 1$ and $\gamma = 1.5$. Similarly,

$$\frac{d}{dt}|t|^{3/2} = \frac{3}{2}\,\mathrm{sgn}(t)|t|^{1/2}, \qquad \frac{d^2}{dt^2}|t|^{3/2} = \frac{3}{4}|t|^{-1/2}.$$

The first derivative is bounded near $t = 0$, and hence

$$\boldsymbol{u}_{1,1.5} \in [W^{1,\infty}(\Omega^{\pm})]^2 \subset [H^1(\Omega^{\pm})]^2.$$

However, the second derivative is not square integrable because

$$\int_0^{\delta} |t|^{-1}\,dt = \infty.$$

Therefore,

$$\boldsymbol{u}_{1,1.5} \notin [H^2(\Omega^{\pm})]^2.$$

These observations justify the piecewise regularity statements used in Example 2.