AIMS Mathematics, 10(4): 7920–7940.
DOI: 10.3934/math.2025364
Received: 10 February 2025
Revised: 24 March 2025
Accepted: 01 April 2025
Published: 07 April 2025

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

Research article

Physics-informed neural network for the heat equation under imperfect contact conditions and its error analysis

Hansaem Oh and Gwanghyun Jo*

Department of Mathematical Data Science, Hanyang University ERICA, 55 Hanyangdaehak-ro, Sangnok-gu, Ansan-si, Gyeonggi-do, Republic of Korea

* **Correspondence:** Email: gwanghyun@hanyang.ac.kr.

Abstract: We propose a physics-informed neural network (PINN)-based method to solve the heat transfer equation under imperfect contact conditions. A major challenge arises from the discontinuity of the solution across the interface, where the exact jump is unknown and implicitly determined by the Kapitza thermal resistance condition. Since the neural network function is smooth on the entire domain, conventional PINN could be inefficient to capture such discontinuities without certain modifications. One remedy is to extend a piecewise continuous function on $\mathbb{R}^d$ to a continuous function on $\mathbb{R}^{d+1}$. This is achieved by applying a Sobolev extension for the solution within each subdomain and introducing, additional coordinate variable that labels the subdomains. This formulation enables the design of neural network functions in the augmented variable, which retains the universal approximation property. We define the PINN in an augmented variable by the minimizer of the loss functional, which includes the implicit interface conditions. Once the loss functional is minimized, the solution obtained by the axis-augmented PINN satisfies the implicit jump conditions. In this way, our method offers a user-friendly way to solve heat transfer equations with imperfect contact conditions. Another advantage of using a continuous representation of solutions in augmented variables is that it allows error analysis in the space of smooth functions. We provide an error analysis of the proposed method, demonstrating that the difference between the exact solution and the predicted solution is bounded by the physics-informed loss functional. Furthermore, the loss functional can be made small by increasing the parameters in the neural network such as the number of nodes in the hidden layers.

Keywords: physics-informed neural network; imperfect contact; heat transfer; error analysis; Sobolev extension
Mathematics Subject Classification: 65N12, 65N15, 65N30

1. Introduction

Composite media often contain *imperfect* contacts at the material interfaces, resulting in discontinuous temperatures across the interface. Since the heat transfer and energy dissipation through imperfectly matched interfaces can significantly influence the temperature profile in internal domains [1–4], neglecting the effects of thermal resistance occurring from imperfect contact in modeling can deteriorate the predictions of thermal behavior in the domain of interest. Kapitza's thermal resistance is one of the widely adopted heat transfer models in material science [5–7], where the amount of jump in the temperature variable along the interface is proportional to the normal heat flux.

One of the difficulties in predicting the thermal profile in imperfectly contacted materials is that the temperature jump across interfaces is implicitly given. In [8, 9], the jump is treated as independent variables; the jump profiles are updated in an iterative way, where in each iteration the elliptic interface problem is solved with the approximate jump. On the other hand, some people applied finite element method (FEM)-based algorithms, where interface conditions are implicitly contained in bilinear forms [10, 11]. In addition, there are extended approaches based on the finite element method (XFEM) to solve the imperfect contact problem where additional degrees of freedom appear near the interface to fit implicit jump conditions [7,12–14]. However, the numerical methods discussed above are inherently complex to implement, as they require additional degrees of freedom in the case of XFEM [13, 14] or involve intricate modifications of the bilinear form [10, 11] or demand iterative updates for the jump profile [8, 9]. Therefore, there arises a need for convenient and efficient methods for the imperfect contact heat transfer equation.

Recently, neural network-based methods have emerged as successful alternatives for solving partial differential equations (PDEs) in the scientific computing community. In the physics-informed neural network (PINN) method, neural network functions serve as a surrogate model for solving PDEs, with its parameters optimized by minimizing objective functions that include the strong form of the governing equations. For an excellent review of PINNs, we refer to [15]. With the advancement of highly efficient automatic differentiation techniques such as those provided by PyTorch [16], PINNs have been successfully applied to a wide range of problems, including Peridynamics [17], thermal-fluid dynamics [18, 19], electromagnetism [20], and Poisson-Boltzmann [22].

In this work, we develop a new PINN-based method for the heat equation under the imperfect contact condition. One of the restrictions of conventional PINN in solving PDE lies in the fact that neural networks belong to a continuous family of functions. Therefore, when PDE contains some interface with possibly discontinuous solutions along the interface, it is hard to expect an accurate reconstruction of solutions by PINN. One remedy is to extend the piecewise continuous function to a continuous function of higher dimension. In [21, 22], the axis augmentation techniques are proposed, where an additional variable labels the locations of subdomains. Following the way of [21,22], we first employ continuous Sobolev extensions for the piecewise H^1-temperature variable. Then, together with the auxiliary variable that labels the locations of subregions, we develop a continuous representation of the heat variable in an augmented variable. Finally, we define the physics-informed neural network in an augmented variable by the minimizer of the loss functional, which includes the implicit jump condition across the interface. In this way, our method offers a user-friendly way to solve heat transfer equations with imperfect contact conditions, without the need for mesh generation or iterative methods to determine jump profiles across the interface.

Aside from the ease of implementation, another advantage of using a continuous representation of solutions in augmented variables is that it allows error analysis in the space of smooth functions. We provide an error analysis of the proposed method. Our main theorem (see Theorem 3.3) states that the difference between the exact solution and the neural network approximation in an energy-like norm is bounded by the physics-informed loss functional. Here, the loss functional can be made arbitrarily small by increasing the parameters in the neural network, such as the number of nodes and number of training samples (see Theorem 3.6). The numerical tests in the result section support the error analysis. To the best of the authors' knowledge, the proposed method is the first successful application of PINN for solving interface problems with implicit jump conditions, accompanied by an error analysis. Let us summarize the advantages of the proposed work: 1) by establishing the continuous representation of solutions in the augmented variable, our method benefits from the universal approximation property of the neural networks; 2) since our methods are formulated on smooth function space, we were able to derive the error estimates for the proposed PINN; 3) our method does not require mesh generation or linearization procedures, in contrast to the conventional FEM-based method.

The rest of the paper is organized as follows. In the next section, we describe the governing equation and develop our version of PINN. The error estimates are carried out in Section 3. In Section 4, we document the performance of the proposed algorithm. The conclusion follows in Section 5.

2. Method

In this section, we develop a PINN method for a heat equation involving imperfect contact. We employ continuous Sobolev extensions for the temperature variable to represent the solution as a continuous function in an augmented variable. The model equation is described in Subsection 2.1, and our version of the PINN method is proposed in Subsection 2.2.

We introduce some definitions. Suppose G is any domain in $\mathbb{R}^d$. For integer $d \geq 1$, $s \geq 0$, and $1 \leq p \leq \infty$, we denote by $W^{s,p}(G)$ the usual Sobolev space associated with norm $\|\cdot\|_{W^{s,p}(G)}$ and seminorm $|\cdot|_{W^{s,p}(G)}$. In particular, if $p = 2$, we then write $W^{s,2}(G) = H^s(G)$, with $\|\cdot\|_{W^{s,2}(G)} = \|\cdot\|_{s,G}$ and $|\cdot|_{W^{s,2}(G)} = |\cdot|_{s,G}$. We also write $W^{0,p}(G) = L^p(G)$ and $\|\cdot\|_{W^{0,p}(G)} = \|\cdot\|_{L^p(G)}$. For the inner product on G, we use the notation $(\cdot,\cdot)_G$. Given an integer $k \geq 0$, let $C^k(G)$ denote the space of all C^k functions in the closure of G. Given a real-valued function ψ, its support is denoted by $\mathrm{supp}(\psi)$. We denote by χ_G the indicator function of G.

2.1. Heat equation under imperfect contact condition

We consider a (bounded) composite material domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) divided by material interface Γ, i.e., $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$ (see Figure 1). The heat flux variable is defined as $\mathbf{q} = -k\nabla T$, where T and k are the temperature variable and thermal conductivity constant, respectively. In an imperfect contact situation, the temperature jump across the material interface is proportional to the heat flux in the normal direction of the interface:

$$T|_{\Omega^+} - T|_{\Omega^-} = -\alpha \mathbf{q}|_{\Omega^-} \cdot \mathbf{n}_\Gamma \quad \text{on } \Gamma,$$

where α the is Kapitza constant [5–7]. Now, let us summarize the steady-state heat equation on Ω under the imperfect condition as follows:

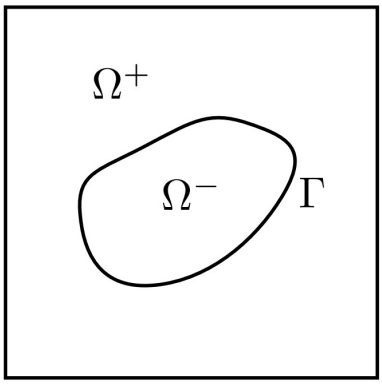


Figure 1. Illustration of a domain Ω. The subregions Ω^+ and $\Omega^-$are separated by Γ.

$$
\begin{aligned}
-\mathrm{div}(k\nabla T) &= f && \text{in } \Omega^+ \cup \Omega^-, && (2.1a)\\
[T]_\Gamma &= \alpha k^- \nabla T|_{\Omega^-} \cdot \mathbf{n}_\Gamma && \text{on } \Gamma, && (2.1b)\\
[k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma &= 0 && \text{on } \Gamma, && (2.1c)\\
T &= g && \text{on } \partial\Omega, && (2.1d)
\end{aligned}
$$

where f is the source term and $\mathbf{n}_\Gamma = \mathbf{n}_{\Omega^-}$. Also, $[\cdot]_\Gamma$ denotes a jump operator across the interface, i.e.,

$$[T]_\Gamma := T|_{\Omega^+} - T|_{\Omega^-}, \quad [k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma = (k\nabla T)|_{\Omega^+} \cdot \mathbf{n}_\Gamma - (k\nabla T)|_{\Omega^-} \cdot \mathbf{n}_\Gamma.$$

Here, we assume that k is piecewise constant, that is, $k = k^+\chi_{\Omega^+} + k^-\chi_{\Omega^-}$ for some positive constants k^+ and k^-. The problem (2.1) has a unique solution [23, 24].

2.2. *Physics-informed neural network method*

In this subsection, we describe the PINN method for the governing equation. The overall process is illustrated in Figure 2.

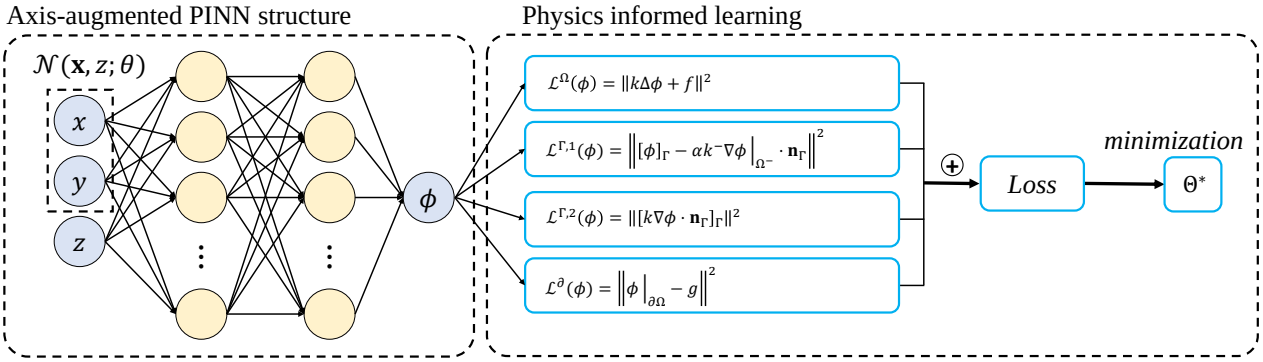


Figure 2. Overall process of the proposed PINN method. The left box illustrates the structure of the neural network with augmented input $(\mathbf{x}, z)$ and the right box details the physics-informed loss functionals.

One challenge is that the solution and its derivative are discontinuous across Γ, whereas conventional neural network functions are inherently smooth on the entire domain. As a result, traditional neural network functions may be inefficient in capturing the discontinuities in the governing equation. To remedy this, we adopt the axis augmentation technique introduced in [21]. The key idea is to introduce an augmented variable z to extend the discontinuous function T to an augmented function T^{aug} that is continuous over $\Omega \times [-1, 1]$. Let us assume that $T|_{\Omega^-}$ and $T|_{\Omega^+}$ are sufficiently smooth. We apply the Sobolev extension theorem (see, e.g., [25, 26]), which guarantees the existence of smooth extensions T^s, $(s = \pm)$, satisfying

$$T^s = T \quad \text{on} \quad \Omega^s, \quad s = \pm.$$

With the additional variable z, we define axis-augmented function

$$T^{aug}(\mathbf{x}, z) := \frac{1+z}{2} T^+(\mathbf{x}) + \frac{1-z}{2} T^-(\mathbf{x}), \qquad \forall (\mathbf{x}, z) \in \Omega \times [-1, 1]. \tag{2.2}$$

Then, it follows that

$$\begin{aligned} T^{aug}(\mathbf{x}, z = 1) &= T(\mathbf{x}), \quad \mathbf{x} \in \Omega^+, \\ T^{aug}(\mathbf{x}, z = -1) &= T(\mathbf{x}), \quad \mathbf{x} \in \Omega^-, \end{aligned}$$

In this way, we have derived a continuous function T^{aug} in an augmented variable, where the variable z labels the locations of subregions, i.e., $z = 1$ for Ω^+ and $z = -1$ for Ω^-.

The advantage of introducing such a continuous representation is that it enables the design of neural network surrogate functions in the augmented variable space, which retains the universal approximation property [27]. The neural network functions in the augmented variable are defined as follows: Given integer $n \geq 1$, we introduce the set $\mathcal{N}_n^{aug}$ of all augmented neural network functions ϕ_θ^{aug} having n hidden layers:

$$\mathcal{N}_n^{aug} := \{\phi_\theta^{aug} : \phi_\theta^{aug}(\mathbf{x}, z) = (L_n \circ \sigma \circ \cdots \circ \sigma \circ L_0)(\mathbf{x}, z)\}, \tag{2.3}$$

where L_i's are linear affine functions and σ is a tanh activation function. Here, θ denotes the collection of all training parameters. The advantage of defining such an augmented neural network function is that, by restricting z to subdomains, we can obtain the piecewise continuous function, i.e.,

$$\phi_\theta(\mathbf{x}) = \phi_\theta^{aug}(\mathbf{x}, 1)\chi_{\Omega^+}(\mathbf{x}) + \phi_\theta^{aug}(\mathbf{x}, -1)\chi_{\Omega^-}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2.4}$$

which can efficiently approximate the desired piecewise continuous solution T. It remains to impose loss functionals for the functions in

$$\mathcal{N}_n^{\pm} := \left\{\phi_\theta : \phi_\theta \text{ satisfies (2.4) for some } \phi_\theta^{aug} \in \mathcal{N}_n^{aug}\right\}.$$

Since the residual of the governing equation can be evaluated on discrete points, we need to define the collection of training points in the variable $(\mathbf{x}, z)$. Let $\{(\mathbf{x}_i^+, 1)\}_{i=1}^{M^+}$ and $\{(\mathbf{x}_i^-, -1)\}_{i=1}^{M^-}$ be the collections of training points in $\Omega^+ \times \{+1\}$ and $\Omega^- \times \{-1\}$, respectively. Let $\{\mathbf{x}_i^\Gamma\}_{i=1}^{M^\Gamma}$ be the collection of training points on Γ. We also let $\{\mathbf{x}_i^\partial\}_{i=1}^{M^\partial}$ be the collection of points on $\partial\Omega$. Some choices of the training points can be

found in [21, 22]. Motivated by the governing Eqs. (2.1a)–(2.1d), the loss functionals are defined as follows: For any piecewise smooth function $\phi_\theta \in \mathcal{N}_n^\pm$,

$$\mathcal{L}^\Omega(\phi_\theta) := \frac{1}{M^+}\sum_{i=1}^{M^+}\left|k^+\Delta\phi_\theta(\mathbf{x}_i^+) + f(\mathbf{x}_i)\right|^2 + \frac{1}{M^-}\sum_{i=1}^{M^-}\left|k^-\Delta\phi_\theta(\mathbf{x}_i^-) + f(\mathbf{x}_i)\right|^2, \tag{2.5}$$

$$\mathcal{L}^{\Gamma,1}(\phi_\theta) := \frac{1}{M^\Gamma}\sum_{i=1}^{M^\Gamma}\left|[\phi_\theta]_\Gamma(\mathbf{x}_i^\Gamma) - \alpha k^-\nabla\phi_\theta|_{\Omega^-}(\mathbf{x}_i^\Gamma)\cdot\mathbf{n}_\Gamma\right|^2, \tag{2.6}$$

$$\mathcal{L}^{\Gamma,2}(\phi_\theta) := \frac{1}{M^\Gamma}\sum_{i=1}^{M^\Gamma}\left|[k\nabla\phi_\theta\cdot\mathbf{n}_\Gamma]_\Gamma(\mathbf{x}_i^\Gamma)\right|^2, \tag{2.7}$$

$$\mathcal{L}^{\partial}(\phi_\theta) := \frac{1}{M^\partial}\sum_{i=1}^{M^\partial}\left|\phi_\theta(\mathbf{x}_i^\partial) - g(\mathbf{x}_i^\partial)\right|^2, \tag{2.8}$$

$$\mathcal{L}(\phi_\theta) := \mathcal{L}^\Omega(\phi_\theta) + \mathcal{L}^{\Gamma,1}(\phi_\theta) + \mathcal{L}^{\Gamma,2}(\phi_\theta) + \mathcal{L}^{\partial}(\phi_\theta). \tag{2.9}$$

The detailed calculations of loss functionals are provided in Appendix A. Finally, our version of the PINN method for (2.1) is formulated as follows: Find $T_\theta \in \mathcal{N}_n^\pm$ such that

$$T_\theta = \underset{\phi_\theta\in\mathcal{N}_n^\pm}{\arg\min}\,\mathcal{L}(\phi_\theta). \tag{2.10}$$

Once T_θ is obtained, the function inherently satisfies the implicit jump conditions (2.1b)–(2.1c). Consequently, we introduce an efficient and user-friendly method to solve the imperfect contact problem without requiring mesh generation or iterative procedures to determine the jump profile.

3. Error estimates

In this section, we derive the error estimates of the proposed PINN method. We define the residuals $\mathcal{R}^\Omega, \mathcal{R}^\Gamma$, and $\mathcal{R}^\partial$ as follows: Given a real-valued function ϕ on Ω,

$$\begin{aligned}
\mathcal{R}^\Omega(\phi) &:= \|k^+\Delta\phi + f\|_{0,\Omega^+}^2 + \|k^-\Delta\phi + f\|_{0,\Omega^-}^2,\\
\mathcal{R}^\Gamma(\phi) &:= \|[\phi]_\Gamma - \alpha k^-\nabla\phi|_{\Omega^-}\cdot\mathbf{n}_\Gamma\|_{0,\Gamma}^2 + \|[k\nabla\phi\cdot\mathbf{n}]_\Gamma\|_{0,\Gamma}^2,\\
\mathcal{R}^\partial(\phi) &:= \|\phi - g\|_{0,\partial\Omega}^2.
\end{aligned}$$

Also, let $\mathcal{R} := \mathcal{R}^\Omega + \mathcal{R}^\Gamma + \mathcal{R}^\partial$. Then the main result in this section is summarized as follows.

- (Theorem 3.3) If T is the solution of (2.1) which is sufficiently smooth in Ω^+ and Ω^-, then for any $T_\theta \in \mathcal{N}_n^\pm$, the piecewise H^1-error between T and T_θ can be bounded above by the residual $\mathcal{R}(T_\theta)$.
- (Theorem 3.6) If T is sufficiently smooth in Ω^+ and Ω^-, then the residual can be made arbitrarily small. For any $\varepsilon > 0$ there exists $T_\theta \in \mathcal{N}_n^\pm$ such that $\mathcal{R}(T_\theta) < \varepsilon$.

Note that, for a sufficiently large number of training points in the definition of the loss functional $\mathcal{L}$, the errors between the loss functional $\mathcal{L}$ and the residual $\mathcal{R}$ can be made arbitrarily small (see, e.g., [28–31]). Therefore, we assume that the following holds.

Assumption 3.1. It holds that $\mathcal{L}^s(\phi) \approx \mathcal{R}^s(\phi)$ for $s = \Omega, \Gamma, \partial$.

This assumption, together with the results in Theorem 3.3 and Theorem 3.6, implies that the error between the exact solution and the neural network solution can be made arbitrarily small by controlling the parameters in the neural network.

Finally, we introduce some broken function spaces

$$H^k(\Omega^\pm) = \{v \in L^2(\Omega) : v|_{\Omega^s} \in H^k(\Omega^s),\ s = \pm\}, \quad k \geq 1,$$
$$C^1(\Omega^\pm) = \{v \in L^2(\Omega) : v|_{\Omega^s} \in C^1(\Omega^s),\ s = \pm\},$$

and associated norms (respectively)

$$\|v\|^2_{k,\Omega^\pm} = \sum_{s=+,-} \|v\|^2_{k,\Omega^s}, \quad v \in H^k(\Omega^\pm),$$
$$\|v\|_{C^1(\Omega^\pm)} = \|v|_{\Omega^+}\|_{C^1(\Omega^+)} + \|v|_{\Omega^-}\|_{C^1(\Omega^-)}, \quad v \in C^1(\Omega^\pm).$$

3.1. Error bound by residuals

To prove Theorem 3.3, we need the Poincaŕe inequality for the piecewise H^1-functions which can be found in [22, Lemma 1].

Lemma 3.2. *It holds that*

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

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

Theorem 3.3. *Suppose that the solution T of (2.1) satisfies $T \in H^2(\Omega^\pm) \cap C^1(\Omega^\pm)$. Then we have*

$$\|T - T_\theta\|^2_{1,\Omega^\pm} + \|[T - T_\theta]_\Gamma\|^2_{0,\Gamma} \leq C\left(\mathcal{R}(T_\theta) + (\mathcal{R}^\Gamma(T_\theta))^{\frac{1}{2}} + (\mathcal{R}^\partial(T_\theta))^{\frac{1}{2}}\right), \quad \forall T_\theta \in \mathcal{N}_n^\pm,$$

where C is a positive constant depending only on Ω, Γ, k, α, $\|T\|_{C^1(\Omega^\pm)}$ and $\|T_\theta\|_{C^1(\Omega^\pm)}$.

Proof. Let $v = T - T_\theta$. By Lemma 3.2 and by the fact that k^-, k^+, and α are positive constants, we have

$$\begin{aligned}
&\|v\|^2_{1,\Omega^\pm} + \|[v]_\Gamma\|^2_{0,\Gamma} \\
&\leq C_0(|v|^2_{1,\Omega^\pm} + \|[v]_\Gamma\|^2_{0,\Gamma} + \|v\|^2_{0,\partial\Omega}) \\
&\leq \tilde{C}\left(|(k^+)^{1/2}\nabla v|^2_{1,\Omega^+} + |(k^-)^{1/2}\nabla v|^2_{1,\Omega^-} + \|\alpha^{-1/2}[v]_\Gamma\|^2_{0,\Gamma} + \|v\|^2_{0,\partial\Omega}\right),
\end{aligned} \tag{3.1}$$

where $\widetilde{C} = C_0\max\{(k^-)^{-1}, (k^+)^{-1}, \alpha^{-1}\}$. By the boundary condition (2.1d), we have

$$\|v\|^2_{0,\partial\Omega} = \|T_\theta - g\|^2_{0,\partial\Omega} = \mathcal{R}(T_\theta).$$

Using integration by parts and the fact that $-k\Delta T = f$ on each $\Omega^- \cup \Omega^+$, we have

$$\begin{aligned}
&|(k^+)^{1/2}\nabla v|^2_{1,\Omega^+} + |(k^-)^{1/2}\nabla v|^2_{1,\Omega^-} + \|\alpha^{-1/2}[v]_\Gamma\|^2_{0,\Gamma} \\
=&(k^+\nabla(T - T_\theta), \nabla v)_{0,\Omega^+} + (k^-\nabla(T - T_\theta), \nabla v)_{0,\Omega^-} + (\alpha^{-1}[T - T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma} \\
=&(k^+\Delta T_\theta + f, v)_{0,\Omega^+} + (k^-\Delta T_\theta + f, v)_{0,\Omega^-} + (k\nabla(T - T_\theta)\cdot\mathbf{n}_\Omega, T - T_\theta)_{0,\partial\Omega} \\
&+ (k^+\nabla(T - T_\theta)|_{\Omega^+}\cdot\mathbf{n}_{\Omega^+}, v|_{\Omega^+})_{0,\Gamma} + (k^-\nabla(T - T_\theta)|_{\Omega^-}\cdot\mathbf{n}_{\Omega^-}, v|_{\Omega^-})_{0,\Gamma} + (\alpha^{-1}[T - T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma}
\end{aligned} \tag{3.2}$$

We remark that $\mathbf{n}_\Gamma = \mathbf{n}_{\Omega^-} = -\mathbf{n}_{\Omega^+}$. Using the identity that

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

we can rewrite the last three terms of (3.2) as

$$\begin{aligned}
&-(k^+\nabla(T-T_\theta)|_{\Omega^+}\cdot\mathbf{n}_\Gamma, v|_{\Omega^+})_{0,\Gamma} + (k^-\nabla(T-T_\theta)|_{\Omega^-}\cdot\mathbf{n}_\Gamma, v|_{\Omega^-})_{0,\Gamma} + (\alpha^{-1}[T-T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma}\\
&= -(\{k\nabla(T-T_\theta)\cdot\mathbf{n}_\Gamma\}_\Gamma, [v]_\Gamma)_{0,\Gamma} - ([k\nabla(T-T_\theta)\cdot\mathbf{n}_\Gamma]_\Gamma, \{v\}_\Gamma)_{0,\Gamma} + \alpha^{-1}[T-T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma}\\
&= -(\{k\nabla(T-T_\theta)\cdot\mathbf{n}_\Gamma\}_\Gamma - \alpha^{-1}[T-T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma} - ([k\nabla(T-T_\theta)\cdot\mathbf{n}_\Gamma]_\Gamma, \{v\}_\Gamma)_{0,\Gamma}. \qquad (3.3)
\end{aligned}$$

Here, by the interface condition (2.1b)–(2.1c), we have

$$\{k\nabla T\cdot\mathbf{n}_\Gamma\}_\Gamma - \alpha^{-1}[T]_\Gamma = 0, \quad [k\nabla T\cdot\mathbf{n}_\Gamma]_\Gamma = 0.$$

Therefore, (3.3) becomes

$$\begin{aligned}
&(\{k\nabla T_\theta\cdot\mathbf{n}_\Gamma\}_\Gamma - \alpha^{-1}[T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma} + ([k\nabla T_\theta\cdot\mathbf{n}_\Gamma]_\Gamma, \{v\}_\Gamma)_{0,\Gamma} \qquad (3.4)\\
&= (k^-\nabla T_\theta|_{\Omega^-}\cdot\mathbf{n}_\Gamma - \alpha^{-1}[T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma} + \frac{1}{2}([k\nabla T_\theta\cdot\mathbf{n}_\Gamma]_\Gamma, [v]_\Gamma)_{0,\Gamma} + ([k\nabla T_\theta\cdot\mathbf{n}_\Gamma]_\Gamma, \{v\}_\Gamma)_{0,\Gamma}
\end{aligned}$$

By (3.2)–(3.5), we obtain

$$\begin{aligned}
&|(k^+)^{1/2}\nabla v|^2_{1,\Omega^+} + |(k^-)^{1/2}\nabla v|^2_{1,\Omega^-} + \|\alpha^{-1/2}[v]_\Gamma\|^2_{0,\Gamma} \qquad (3.5)\\
&= (k^+\Delta T_\theta + f, v)_{0,\Omega^+} + (k^-\Delta T_\theta + f, v)_{0,\Omega^-} + (k\nabla(T-T_\theta)\cdot\mathbf{n}_\Omega, v)_{0,\partial\Omega}\\
&\quad + (k^-\nabla T_\theta|_{\Omega^-}\cdot\mathbf{n}_\Gamma - \alpha^{-1}[T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma} + \frac{1}{2}([k\nabla T_\theta\cdot\mathbf{n}_\Gamma]_\Gamma, [v]_\Gamma)_{0,\Gamma} + ([k\nabla T_\theta\cdot\mathbf{n}_\Gamma]_\Gamma, \{v\}_\Gamma)_{0,\Gamma}\\
&=: I_1 + \cdots + I_6. \qquad (3.6)
\end{aligned}$$

We estimate each of the terms. By the Cauchy-Schwarz and Young's inequalities, we have for I_1, I_2,

$$\begin{aligned}
|I_1| + |I_2| &= |(k^-\Delta T_\theta + f, v)_{0,\Omega^+}| + |(k^+\Delta T_\theta + f, v)_{0,\Omega^-}|\\
&\le \|k^-\Delta T_\theta + f\|_{0,\Omega^+}\|v\|_{0,\Omega^+} + \|k^+\Delta T_\theta + f\|_{0,\Omega^-}\|v\|_{0,\Omega^-}\\
&\le \frac{\delta_1}{2}\|k^-\Delta T_\theta + f\|^2_{0,\Omega^+} + \frac{1}{2\delta_1}\|v\|^2_{0,\Omega^+} + \frac{\delta_2}{2}\|k^+\Delta T_\theta + f\|^2_{0,\Omega^-} + \frac{1}{2\delta_2}\|v\|^2_{0,\Omega^-}\\
&= \frac{1}{2}\max(\delta_1,\delta_2)\mathcal{R}^\Omega(T_\theta) + \left(\frac{1}{2\delta_1} + \frac{1}{2\delta_2}\right)\|v\|^2_{1,\Omega^\pm}.
\end{aligned}$$

Here, δ_1 and δ_2 are positive constants related to the Young's inequality, which will be determined later.

Next, let us estimate I_3. By the fact that $T, T_\theta \in C^1(\Omega^\pm)$ and $T = g$ on $\partial\Omega$, we have

$$|I_3| \le \|k\nabla(T-T_\theta)\cdot\mathbf{n}\|_{0,\partial\Omega}\|T_\theta - g\|_{0,\partial\Omega} \le C(\mathcal{R}^\partial(T_\theta))^{\frac{1}{2}}.$$

By the Cauchy-Schwarz and Young's inequalities (with δ_3, $\delta_4 > 0$), we have, for I_4 and I_5,

$$|I_4| + |I_5| = |(k^-\nabla T_\theta|_{\Omega^-}\cdot\mathbf{n}_\Gamma - \alpha^{-1}[T_\theta]_\Gamma, [v]_\Gamma)_{0,\Gamma}| + \frac{1}{2}|([k\nabla T_\theta\cdot\mathbf{n}]_\Gamma, [v]_\Gamma)_{0,\Gamma}|$$

$$\leq \|k^-\nabla T_\theta|_{\Omega^-}\cdot\mathbf{n}_\Gamma-\alpha^{-1}[T_\theta]_\Gamma\|_{0,\Gamma}\cdot\|[v]_\Gamma\|_{0,\Gamma}+\frac{1}{2}\|[k\nabla T_\theta\cdot\mathbf{n}]_\Gamma\|_{0,\Gamma}\cdot\|[v]_\Gamma\|_{0,\Gamma}$$
$$\leq \frac{\delta_3}{2}\|k^-\nabla T_\theta|_{\Omega^-}\cdot\mathbf{n}_\Gamma-\alpha^{-1}[T_\theta]_\Gamma\|^2_{0,\Gamma}+\frac{1}{2\delta_3}\|[v]_\Gamma\|^2_{0,\Gamma}+\frac{\delta_4}{2}\|[k\nabla T_\theta\cdot\mathbf{n}]_\Gamma\|^2_{0,\Gamma}+\frac{1}{8\delta_4}\|[v]_\Gamma\|^2_{0,\Gamma}$$
$$\leq \max\left\{\frac{\delta_3}{2\alpha^2},\frac{\delta_4}{2}\right\}\mathcal{R}^\Gamma(T_\theta)+\left(\frac{1}{2\delta_3}+\frac{1}{8\delta_4}\right)\|[v]_\Gamma\|^2_{0,\Gamma}.$$

Since $T, T_\theta\in C^1(\Omega^\pm)$ and $[k\nabla\cdot\mathbf{n}_\Gamma]=0$, we have, for I_6,

$$|I_6|\leq\|[k\nabla T_\theta\cdot\mathbf{n}]_\Gamma\|_{0,\Gamma}\|\{T-T_\theta\}\|_{0,\Gamma}\leq C(\mathcal{R}^\Gamma(T_\theta))^{\frac{1}{2}}.$$

Then, plugging the estimates of $I_1,\cdots,I_6$ into (3.6), together with (3.1), we have

$$\left[1-\tilde{C}\left(\frac{1}{2\delta_1}+\frac{1}{2\delta_2}\right)\right]\|v\|^2_{1,\Omega^\pm}+\left(1-\frac{\tilde{C}}{2\delta_3}-\frac{\tilde{C}}{8\delta_4}\right)\|[v]_\Gamma\|^2_{0,\Gamma}$$
$$\leq\tilde{C}\left[\frac{1}{2}\max(\delta_1,\delta_2)\mathcal{R}^\Omega(T_\theta)+\mathcal{R}^\partial(T_\theta)+C(\mathcal{R}^\partial(T_\theta))^{\frac{1}{2}}+\max\left\{\frac{\delta_3}{2\alpha^2},\frac{\delta_4}{2}\right\}\mathcal{R}^\Gamma(T_\theta)+C(\mathcal{R}^\Gamma(T_\theta))^{\frac{1}{2}}\right]$$

Now, by the choice

$$\delta_1=\delta_2=2\tilde{C},\quad \delta_3=2\tilde{C},\quad \delta_4=\frac{\tilde{C}}{2},$$

we obtain the desired inequality. □

Remark 3.4. As stated in Theorem 3.3, the constant C may depend on the piecewise C^1-norm of the neural network T_θ. In practice, to prevent this norm from becoming excessively large, we can include an L^2-regularization term in the loss functional $\mathcal{L}$, following the approach in [29, 31].

3.2. Estimates on the residual and the training error

As proved in the previous subsection, the piecewise H^1-error can be controlled by the residuals $\mathcal{R}^\Omega$, $\mathcal{R}^\partial$ and $\mathcal{R}^\Gamma$. Given some regularity assumption, we will prove that if $n>1$, then the residuals can be made arbitrarily small, i.e., for any $\varepsilon>0$, there exists $T_\theta\in\mathcal{N}_n^\pm$ such that $\mathcal{R}(T_\theta)<\varepsilon$.

Lemma 3.5. *Suppose that $n>1$ and the solution T of (2.1) satisfies $T|_{\Omega^s}\in W^{3,\infty}(\Omega^s)$ for $s=\pm$. For any $\varepsilon>0$, there exists a neural network $T_\theta^\pm\in\mathcal{N}_n^\pm$ such that*

$$\|T-T_\theta^\pm\|_{W^{2,\infty}(\Omega^+)}+\|T-T_\theta^\pm\|_{W^{2,\infty}(\Omega^-)}\leq\varepsilon.$$

Proof. By the argument given in the beginning of Subsection 2.2, for sufficiently large $R>0$, one can construct an extension $T^{aug}\in W^{3,\infty}([-R,R]^{d+1})$ of T such that $\Omega\times[-1,1]\subset[-R,R]^{d+1}$ and

$$T^{aug}(\mathbf{x},1)=T(\mathbf{x}),\quad\forall\mathbf{x}\in\Omega^+,\qquad T^{aug}(\mathbf{x},-1)=T(\mathbf{x})\quad\forall\mathbf{x}\in\Omega^-.$$

Then, according to the approximation theorem of neural network functions with the tanh activation function [32], there exists $T_\theta^{aug}\in\mathcal{N}_n^{aug}$ such that

$$\|T^{aug}-T_\theta^{aug}\|_{W^{3,\infty}([-R,R]^{d+1})}\leq\varepsilon.$$

Now the conclusion follows from the Sobolev embedding theorem (cf. [26]). □

Using the lemma above, we obtain the desired theorem.

Theorem 3.6. *Suppose that $n > 1$ and the solution T of (2.1) satisfies $T|_{\Omega^s} \in W^{3,\infty}(\Omega^s)$ for $s = \pm$. For any $\varepsilon > 0$, there exists $T_\theta \in \mathcal{N}_n^\pm$ such that*

$$\mathcal{R}(T_\theta) \leq C\varepsilon,$$

where C is a positive constant depending only on Ω, Γ, k, α and T.

Proof. Let $\varepsilon > 0$ be chosen arbitrarily. By Lemma 3.5, there exists $T_\theta \in \mathcal{N}_n^\pm$ such that

$$\|T - T_\theta^\pm\|_{W^{2,\infty}(\Omega^+)} + \|T - T_\theta^\pm\|_{W^{2,\infty}(\Omega^-)} \leq \varepsilon.$$

Therefore, we have

$$\begin{aligned} \|T - T_\theta\|_{2,\Omega^+}^2 + \|T - T_\theta\|_{2,\Omega^-}^2 &\leq |\Omega^+| \, \|T - T_\theta^\pm\|_{W^{2,\infty}(\Omega^+)} + |\Omega^-| \, \|T - T_\theta^\pm\|_{W^{2,\infty}(\Omega^-)} \\ &\leq |\Omega|\varepsilon. \end{aligned} \tag{3.7}$$

Since T is the solution of (2.1), we obtain from (3.7) and the trace inequality that

$$\begin{gathered} \mathcal{R}^\Omega(T_\theta) = \|k^+\Delta(T - T_\theta)\|_{0,\Omega^+}^2 + \|k^-\Delta(T - T_\theta)\|_{0,\Omega^-}^2 \leq \max\{(k^-)^2, (k^+)^2\}|\Omega|\varepsilon, \\ \mathcal{R}^\Gamma(T_\theta) \leq C(\|T - T_\theta\|_{2,\Omega^+}^2 + \|T - T_\theta\|_{2,\Omega^-}^2) \leq C|\Omega|\varepsilon, \\ \mathcal{R}^\partial(T_\theta) = \|T - T_\theta\|_{0,\partial\Omega}^2 \leq C\varepsilon. \end{gathered}$$

This completes the proof. □

Remark 3.7. In the proof of Lemma 3.5, the approximation property of the tanh neural network functions [32] was employed. This validates our choice of tanh activation functions in (2.3).

4. Numerical results

In this section, we present some examples. The domain $\Omega = [-1, 1]^2$ is separated by zeros of a level set function $L(x, y)$, i.e., $\Omega^- = \{(x, y) \in \Omega : L(x, y) < 0\}$ and $\Omega^+ = \{(x, y) \in \Omega : L(x, y) > 0\}$. For all examples, we assume that $n = 2$ (that is, the number of hidden layers is two) and the tangent hyperbolic function is adopted for the activation function. In addition, we assign N-number of nodes for each hidden layer. The weights in the neural network are initialized using the Kaiming uniform method [33], with biases set to zero.

We present three examples, each featuring a different interface shape: circle, line, and perturbed circle. In Example 4.1–4.2, the L^2- and L^∞-errors against the exact solutions are computed, where we observe reasonable convergence as the parameters increase. In Example 4.3, a unidirectional heat flux generated by some boundary condition is considered. Since the analytic solution is unknown in this case, we validate the PINN-predicted solution by comparing it with the FEM solution. All examples are performed on a single core of an Intel(R) Core(TM) i7-13700 CPU.

4.1. Choices of optimizers and training sample sets

In this subsection, we justify our choice of optimizer and the training sampling strategy. We compare the results obtained by two optimizers in Example 4.1. The first is the Levenberg–Marquardt (LM) optimizer [34] with 3,000 epochs. The LM optimizer is configured with an initial damping parameter of 10^5, which is updated at every iteration. The damping parameter is divided by 1.3 when the process is successful and is multiplied by 3 otherwise. The second is the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimizer [35] with 10,000 epochs, a learning rate of 0.1, and a strong Wolfe line search option.

To describe the training points, let us introduce the order-related parameter m, which determines the total number of training points. The *inner* points (m^2-number) are determined either by Chebyshev–Gauss points, Gauss–Legendre points, or uniformly spaced grid points. For example, Chebyshev–Gauss points of order m is defined by

$$\mathbf{x}^{ij} = \left(\cos\left((2i+1)\pi/2m\right), \cos\left((2j+1)\pi/2m\right)\right), \quad 0 \le i, j < m$$

The *interface* ($4m$-number) and *boundary* points ($4m$-number) are uniformly sampled at the interface and boundary, respectively. In this way, the total number of training points corresponding to the parameter m is $m^2 + 8m$. We compare the results obtained by different types of training points in Example 4.1.

Example 4.1 (Circle shaped interface)**.** In this example, the level set function is $L(x, y) = x^2 + y^2 - r_0^2$ and the exact solution is

$$T = \begin{cases} \dfrac{x^2+y^2}{2k^-} - r_0\alpha + \left(\dfrac{1}{2k^+} - \dfrac{1}{2k^-}\right) r_0^2, & \text{if } (x, y) \in \Omega^-, \\ \dfrac{x^2+y^2}{2k^+}, & \text{if } (x, y) \in \Omega^+, \end{cases}$$

where $r_0 = 0.5$ and $\alpha = 1$. We consider two different thermal conductivity contrasts: $(k^-, k^+) = (1, 10)$, or $(100, 1)$.

We compare loss curves with increasing training epochs for the LM and L-BFGS optimizer in Figure 3. The loss function decreases to the lower level by the LM optimizer. Consequently, we adopted the LM optimizer for all experiments in this study. Next, we compare the L^2 and L^∞ errors computed using Chebyshev–Gauss, Gauss–Legendre, and uniform grids type training points. We report errors in Figure 4. Overall, the smallest errors are achieved with the choice of Chebyshev–Gauss points and with $N = 20$. Therefore, we use Chebysev–Gauss type training points for the rest of the Section.

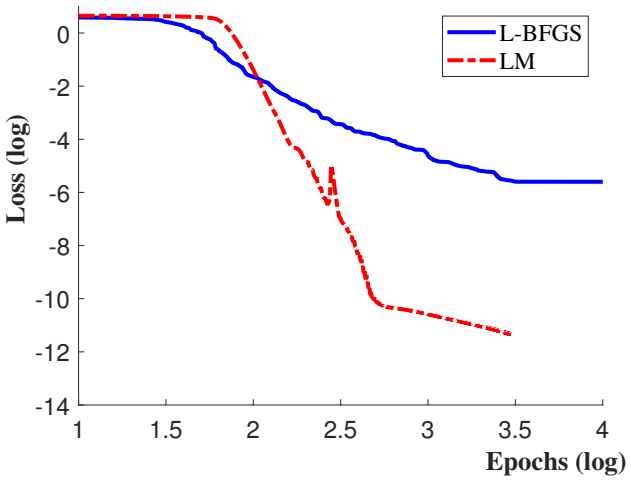
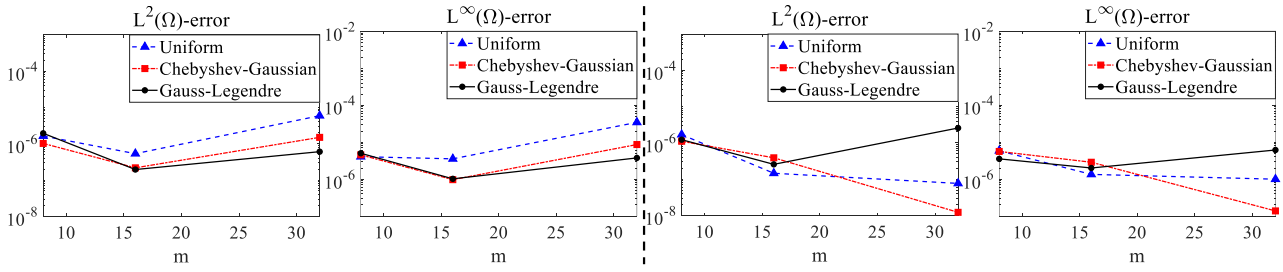


Figure 3. Evolution of loss function (log-scale) with increasing number of epochs for the LM and L-BFGS optimizers.

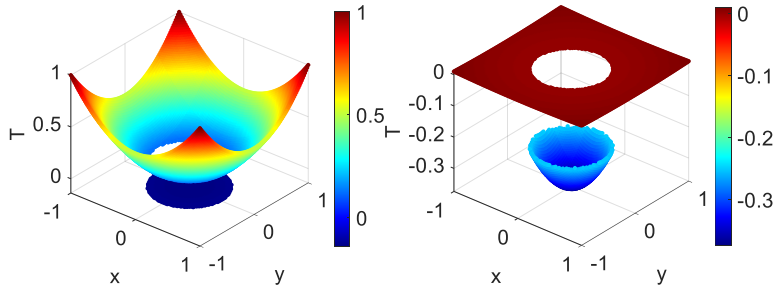


(a) $N = 10$ (b) $N = 20$

Figure 4. Comparisons of L^2 and L^∞ errors obtained by different choices of training sampling points (uniform grids, Chebyshev–Guass, and Gauss–Legendre). The left box (a) corresponds to the case with $N = 10$, and the right box (b) corresponds to the case when $N = 20$.

Now, let us present numerical results by the proposed PINN method. Figure 5 presents the numerical solutions for the two conductivity cases. Notably, no spurious oscillation appears near the interface in any case. We report the errors L^2 and L^∞ that are computed at 10,000 uniformly sampled test points. The errors and CPU time for the two conductivity contrasts, that is, $(k^+, k^-) = (1, 10)$ or $(100, 1)$, are documented in Tables 1 and 2, respectively.

Let us discuss the computational complexities with respect to N and m. One of the bottlenecks of the LM optimizer is the inversion of the Hessian matrix associated with the loss functional. In our case, the number of entries in the Hessian matrix is $O(N^2m^2)$. As a results, the CPU time may dramatically increases when N and m increase. Fortunately, the smallest errors were achieved at relatively small parameters, specifically $N = 20$ and $m = 32$. In particular, with this parameter choice, the L^2 errors remain below $4 \cdot 10^{-7}$ in Tables 1 and 2 with corresponding CPU times of 261 (s) and 274 (s), respectively.



(a)$(k^+, k^-) = (1, 10)$ (b)$(k^+, k^-) = (100, 1)$

Figure 5. Graphs of the solutions obtained by PINN in Example 4.1 with thermal conductivity $(k^+, k^-) = (1, 10)$ or $(100, 1)$.

N	m	$\|T - T_\theta\|_{L^\infty}$	$\|T - T_\theta\|_{L^2}$	CPU time
10	8	4.55E−6	1.02E−6	75.40
	16	9.70E−7	2.17E−7	90.16
	32	4.82E−6	1.08E−6	156.61
20	8	5.63E−6	1.11E−6	114.57
	16	2.87E−6	3.89E−7	157.11
	32	3.70E−7	1.44E−7	261.38

Table 1. The L^2 and L^∞ errors and CPU time of PINN in Example 4.1 where $(k^+, k^-) = (1, 10)$.

N	m	$\|T - T_\theta\|_{L^\infty}$	$\|T - T_\theta\|_{L^2}$	CPU time
10	8	8.99E−4	2.54E−4	75.90
	16	9.58E−6	1.98E−6	88.90
	32	4.46E−6	6.29E−7	157.78
20	8	1.79E−3	6.74E−4	113.07
	16	3.53E−5	3.94E−6	157.24
	32	3.02E−6	3.61E−7	273.93

Table 2. The L^2 and L^∞ errors and CPU time of PINN in Example 4.1 where $(k^+, k^-) = (100, 1)$.

Before closing this example, we compare the proposed method with the FEM-type method. One of the authors has the experience of implementing FEM for elliptic interface problems with implicit jump conditions, as described in [10], where the L^2 error is of $O(h^{-2})$. To achieve comparable accuracy, the FEM approach requires solving an algebraic system with over 10^5 unknowns, resulting in substantial

computational cost (see e.g., Table 2 in [10]). Although it is difficult to compare directly, our method achieves a comparable accuracy with less CPU time.

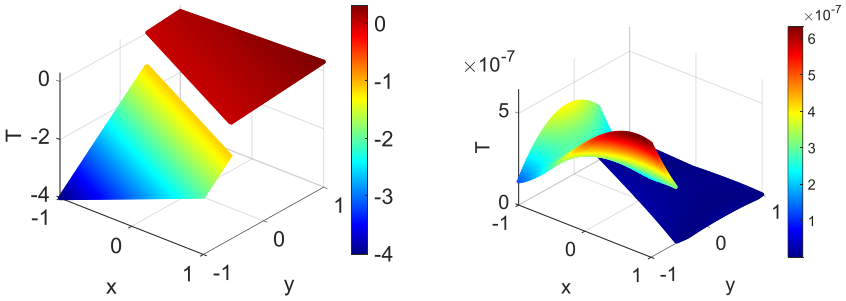
4.2. Comparison with other methods

In this subsection, we compare the proposed PINN with other methods. In Example 4.2, we compare our method with the piecewise PINN method [36], while in Example 4.3, we compare it with a FEM-type method [11].

Example 4.2 (Line interface)**.** The level set function is $L(x, y) = x + 2y + r_0$ and the exact solution is

$$T = \begin{cases} \dfrac{x + 2y + r_0}{k^-} - \sqrt{5}\alpha, & \text{if } (x, y) \in \Omega^-, \\ \dfrac{x + 2y + r_0}{k^+}, & \text{if } (x, y) \in \Omega^+, \end{cases}$$

where $\alpha = 0.5$ and $r_0 = 0.1$. The thermal conductivity is $(k^-, k^+) = (1, 10)$. The graphs of T_θ and $|T - T_\theta|$ are presented in Figure 6. The L^2 and L^∞ errors are reported on the left side of Table 3. Notably, the L^2 errors are below 2×10^{-7} when $N = 10$ and $m = 16$. Therefore, the proposed PINN successfully approximates the exact solution.



(a) T_θ (b) $|T - T_\theta|$

Figure 6. Graphs of the T_θ and $|T - T_\theta|$ in Example 4.2.

N	m	Proposed PINN			Piecewise PINN [36]		
		$\|T_\theta - T\|_{L_\infty}$	$\|T_\theta - T\|_{L_2}$	CPU time	$\|T_\theta - T\|_{L_\infty}$	$\|T_\theta - T\|_{L_2}$	CPU time
10	8	3.98E−5	1.09E−5	77.76	7.38E−5	2.29E−5	288.85
	16	6.32E−7	2.34E−7	96.07	1.10E−6	2.72E−7	345.40
	32	2.88E−6	6.54E−7	136.73	1.33E−5	1.76E−6	400.98
20	8	2.14E−4	5.93E−5	173.40	9.65E−4	1.20E−4	778.21
	16	1.56E−6	4.21E−7	261.18	3.11E−5	3.70E−6	858.13
	32	2.38E−5	2.91E−6	405.33	2.32E−5	3.06E−6	1149.51

Table 3. The L^2 and L^∞ errors and CPU time of the proposed PINN (left) and piecewise PINN [36] (right) in Example 4.2.

We compare our version of PINN with piecewise PINN introduced in [36]. In [36], the authors propose two separate neural network functions for each subdomain. As a result, the total number of

parameters is approximately doubled when the same number of nodes is used in each hidden layer. We report the L^2 and L^∞ errors and CPU time on the right side of Table 3. We observe that L^2 and L^∞ errors obtained by the proposed PINN and piecewise PINN are similar. However, in terms of CPU time, our PINN is more efficient, which is due to the increased number of parameters in piecewise PINN.

Example 4.3 (Perturbed-circle shaped interface)**.** In this example, we consider the unidirectional heat flow given by the homogeneous outer source ($f = 0$) and the boundary condition that

$$\begin{aligned} T &= 1, && \text{when } x = -1, \\ T &= 0, && \text{when } x = 1, \\ \nabla T \cdot \mathbf{n}_\Omega &= 0, && \text{when } y = -1 \text{ or } y = 1, \end{aligned}$$

where $(k^-, k^+) = (1, 2)$ and $\alpha = 0.5$. We consider the perturbed circle shape interface:

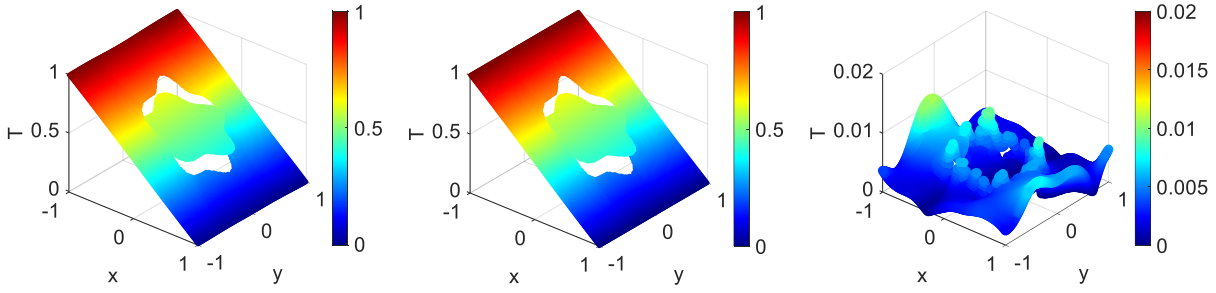
$$r = \frac{1}{2}\left(1 + \frac{1}{7}\sin 5\theta\right).$$

Since the exact solution is unknown, we compare the solution obtained by PINN with that by FEM [11], denoted as T_{FEM}. In Appendix B, we provide a brief description of FEM for the completeness of the presentation.

Figure 7 presents the comparison between T_{FEM} and T_θ, showing the overall agreement between the two solutions. We also calculate L^2 and L^∞ differences between T_θ and T_{FEM}:

$$\|T_\theta - T_{FEM}\|_{L^2(\Omega)} = 3.31 \times 10^{-3}, \qquad \|T_\theta - T_{FEM}\|_{L^\infty(\Omega)} = 1.08 \times 10^{-2},$$

which are reasonably small. Thus, we conclude that the solutions obtained by PINN are comparable to those from the well-established FEM. However, in terms of implementation, the proposed method is simpler than FEM type method, without the needs for the mesh generation or iteration techniques to determine jump profiles.



(a) T_θ (b) T_{FEM} (c) $|T_\theta - T_{\text{FEM}}|$

Figure 7. Comparisons of the solutions T_θ and T_{FEM} in Example 4.3.

4.3. Discussion

In this subsection, we provide some discussion regarding the numerical experiments. We begin by examining the relation between the choice of the parameter N (number of nodes in a hidden layer) and its influences on the accuracy. In Figure 8, we plot the L^2 errors of the reconstructed solutions

obtained by our methods with varying number of N, with $m = 16$ fixed both for the Example 4.1 and 4.2. Here, we do not observe clear decaying trends in errors as N increases. This could result from difficulties in finding the global minimizer using optimization algorithms, as discussed in [37,38]. The convergence of the optimizer, closly related to the optimization landscape, depends on various factors including the interface shapes, PDE-related parameters (e.g., $k^{\pm}$), etc. Notably, in Example 4.2, where both the interface shape and the solutions are relatively simple, the smallest L^2 error (2.3E − 7) was observed with just $N = 10$. A similar phenomenon is observed for the m parameter, which determines the number of training points. In Figure 4, we do not observe the clear decay of errors as m increases. Based on our parameter study, we suggest selecting moderate values such as $N = 10, 20$ and $m = 16$ rather than significantly large values.

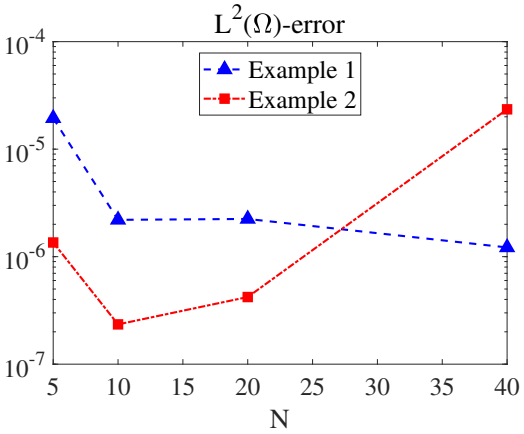


Figure 8. The L^2 errors when $N = 5, 10, 20, 40$ with fixed $m = 16$ for Example 4.1 and 4.2.

We now turn to the discussion of the stability of the optimization process. Through our experiments, we observed that the optimization process becomes unstable when there is a large contrast of k^+ and k^- or the parameters N and m are too large. In particular, when the damping parameter in LM becomes close to zero, the ill-conditioned Hessian matrix may cause the loss function to diverge. To remedy this, we modified the LM optimizer slightly by introducing an additional rule: if current loss exceeds twice the previous loss, increase the damping parameter by a factor of 10^5. This adjustment improved the robustness of the optimization process.

Finally, let us discuss the regularity assumptions used in our error estimates. In the statements of Lemma 3.5 and Theorem 3.6, we assumed that $T|_{\Omega^s} \in W^{3,\infty}(\Omega^s)$ for $s = \pm$, which is a strong regularity condition. However, this assumption may not hold in practical applications, especially when material interfaces induce singularities. In such cases, the errors may not be controlled by increasing the number of parameters since the universal approximation fails under low regularity conditions.

5. Conclusions

In this work, we developed a new PINN method to solve heat transfer equations involving imperfect contact conditions. There are mainly two difficulties in solving imperfect contact problems: 1) the solutions are discontinuous across the interfaces, and 2) the jumps are implicitly determined. To remedy these, we adopted the axis-augmentation techniques introduced in [21] to establish a

continuous representation of the solution in the augmented variable. Then, we defined the neural network function in an augmented variable with the physics-informed loss functional, which includes the implicit jump conditions. After the loss function is minimized, the resulting solutions obtained by PINN naturally satisfy the desired jump conditions. One of the advantages of such a continuous representation formulation was the availability of error analysis. We carried out the error analysis; our main theorem states that given a sufficient number of parameters, the loss between exact and PINN solutions can be made arbitrarily small. The numerical experiments that support our analysis are reported in the result section. Let us discuss the limitations of the proposed method. First, the number of parameters in the neural network increases with the introduction of a new axis. Second, while our analysis assumes strong regularity conditions, these conditions may not always hold in the real-world applications.

Appendix

A. Calculation of loss functional

We provide details regarding the calculation of loss functionals presented in (2.5)–(2.9). Recall first that $\phi_\theta(\mathbf{x}) = \phi_\theta^{aug}(\mathbf{x}, 1)\chi_{\Omega^+}(\mathbf{x}) + \phi_\theta^{aug}(\mathbf{x}, -1)\chi_{\Omega^-}(\mathbf{x})$. While the loss functionals are written in the function ϕ_θ, the augmented neural network function $\phi_\theta^{aug}(\mathbf{x}, z)$ is employed during the implementation stage. Specifically, the parameter $z \in \{1, -1\}$, indicates whether the sampling points belong to Ω^- or Ω^+, respectively. Below, we summarize how operators such as ∇, ∇, and $[\cdot]$ acting on ϕ_θ are calculated using the augmented neural network functions:

$$
\begin{aligned}
&\Delta\phi_\theta(\mathbf{x}_i^\pm) = \Delta_{\mathbf{x}}\phi_\theta^{aug}(\mathbf{x}_i^\pm, z = \pm 1),\\
&[\phi_\theta]_\Gamma(\mathbf{x}_i^\Gamma) = \phi_\theta^{aug}(\mathbf{x}_i^\Gamma, z = 1) - \phi_\theta^{aug}(\mathbf{x}_i^\Gamma, z = -1),\\
&[k\nabla\phi_\theta \cdot \mathbf{n}_\Gamma]_\Gamma(\mathbf{x}_i^\Gamma) = k^+\nabla_{\mathbf{x}}\phi_\theta^{aug}(\mathbf{x}_i^\Gamma, z = 1) \cdot \mathbf{n}_\Gamma - k^-\nabla_{\mathbf{x}}\phi_\theta^{aug}(\mathbf{x}_i^\Gamma, z = -1) \cdot \mathbf{n}_\Gamma,
\end{aligned}
$$

where

$$
\begin{aligned}
\nabla_{\mathbf{x}}\phi_\theta^{aug} &= \left(\frac{\partial\phi_\theta^{aug}}{\partial x}, \frac{\partial\phi_\theta^{aug}}{\partial y}\right)\\
\Delta_{\mathbf{x}}\phi_\theta^{aug} &= \frac{\partial^2\phi_\theta^{aug}}{\partial x^2} + \frac{\partial^2\phi_\theta^{aug}}{\partial y^2}.
\end{aligned}
$$

B. FEM for heat equation involving imperfect contact condition

We briefly describe the FEM introduced in [11], where the so-called immersed finite element (IFE) method was employed. In IFE methodology, uniform grids are employed for the interface problem, allowing the interface to cut through the element, which contrasts with the usage of interface-fitted grids. Instead, the basis function is modified so that the they satisfy the interface conditions.

Let $\mathcal{T}_h$ be a triangulation of the domain Ω. When $E \in \mathcal{T}_h$ is not cut through by the interface, the local space is a standard linear space, denoted by $S_h(E)$. Suppose E is cut through by the interface; the basis function is defined as a piecewise polynomial of the form:

$$\phi(\mathbf{x}) = \begin{cases} \phi^+(\mathbf{x}) = a^+x + b^+y + c^+, & \text{when } \mathbf{x} \in E \cap \Omega^+ \\ \phi^-(\mathbf{x}) = a^-x + b^-y + c^-, & \text{when } \mathbf{x} \in E \cap \Omega^- \end{cases} \tag{B.1}$$

Here, the coefficients in (B.1) are determined by the interface conditions (2.1b)-(2.1c), i.e.,

$$\begin{cases} \phi^+ - \phi^- = \alpha k^- \nabla\phi^- \cdot \mathbf{n}_\Gamma \\ k^+\nabla\phi^+ \cdot \mathbf{n}_\Gamma - k^-\nabla\phi^- \cdot \mathbf{n}_\Gamma = 0. \end{cases}$$

In this way, the local space $\widehat{S}_h(E)$ is defined with nodes and degrees of freedom. The global space $\widehat{S}_h(\Omega)$ is defined by patching local spaces, imposing the continuity at the nodes. Let us remark that the dimension of $\widehat{S}_h(\Omega)$ corresponds to the number of nodes in $\mathcal{T}_h$. Finally, FEM solution is obtained by solving the weak problem : find $T_h \in \widehat{S}_h$ such that satisfies

$$a_h(T_h, \phi) = (f, \phi), \quad \forall \phi \in \widehat{S}_h.$$

where

$$a_h(T_h, \phi) = \sum_{E \in \mathcal{T}_h} \int_E k\nabla T_h \cdot \nabla\phi \, dx + \frac{1}{\alpha} \int_\Gamma [T_h]_\Gamma [\phi]_\Gamma \, ds.$$

The detailed implementation can be found in [11].

Author contributions

Hansaem Oh: Formal analysis, Methodology, Writing-original draft; Gwanghyun Jo: Conceptualization, Methodology, Writing-review & editing.

Acknowledgements

The first author (H. Oh) is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NO. RS-2024-00463063). We thank Dr. Hyeokjoo Park for his valuable advice on the error analysis.

Use of Generative-AI tools declaration

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

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. L. J. Challis, Kapitza resistance and acoustic transmission across boundaries at high frequencies, *J. Phys. C: Solid State Phys.,* **7** (1974), 481. https://doi.org/10.1088/0022-3719/7/3/005
2. E. T. Swartz, R. O. Pohl, Thermal resistance at interfaces, *Appl. Phys. Lett.,* **51** (1987), 2200–2202. https://doi.org/10.1063/1.98939
3. R. J. Stoner, H. J. Maris, Kapitza conductance and heat flow between solids at temperatures from 50 to 300 K, *Phys. Rev. B,* **48** (1993), 16373–16387. https://doi.org/10.1103/PhysRevB.48.16373
4. D. G. Cahill, W. K. Ford, K. E. Goodson, G. D. Mahan, A. Majumdar, H. J. Maris, et al., Nanoscale thermal transport, *J. Appl. Phys.,* **93** (2003), 793–818. https://doi.org/10.1063/1.1524305
5. P. Kapitza, Heat transfer and superfluidity of helium II, *Phys. Rev.,* **60** (1941), 354–355. https://doi.org/10.1103/PhysRev.60.354
6. C. W. Nan, R. Birringer, Determining the kapitza resistance and the thermal, *Phys. Rev. B,* **57** (1998), 8264–8268. https://doi.org/10.1103/PhysRevB.57.8264
7. J. Yvonnet, Q. C. He, Q. Z. Zhu, J. F. Shao, A general and efficient computational procedure for modelling the Kapitza thermal resistance based on XFEM, *Comput. Mater. Sci.,* **50** (2011), 1220–1224. https://doi.org/10.1016/j.commatsci.2010.02.040
8. J. Weisz, On an iterative method for the solution of discretized elliptic problems with imperfect contact condition, *J. Comput. Appl. Math.,* **72** (1996), 319–333. https://doi.org/10.1016/0377-0427(96)00003-9
9. S. Hüeber, B. I. Wohlmuth, Thermo-mechanical contact problems on non-matching meshes, *Comput. Method. Appl. M.,* **198** (2009), 1338–1350. https://doi.org/10.1016/j.cma.2008.11.022
10. G. H. Jo, D. Y. Kwak, Enriched P1-Conforming Methods for Elliptic Interface Problems with Implicit Jump Conditions, *Adv. Math. Phys.,* **2018** (2018), 9891281. https://doi.org/10.1155/2018/9891281
11. D. Y. Kwak, S. Lee, Y. Hyon, A new finite element for interface problems having robin type jump, *Int. J. Numer. Anal. Model.,* **14** (2017), 532–549.
12. J. Yvonnet, Q. C. He, C. Toulemonde, Numerical modelling of the effective conductivities of composites with arbitrarily shaped inclusions and highly conducting interface, *Compos. Sci. Technol.,* **68** (2008), 2818–2825. https://doi.org/10.1016/j.compscitech.2008.06.008
13. J. T. Liu, S. T. Gu, E. Monteiro, Q. C. He, A versatile interface model for thermal conduction phenomena and its numerical implementation by XFEM, *Comput. Mech.,* **53** (2014), 825–843. https://doi.org/10.1007/s00466-013-0933-9
14. T. Xue, X. Zhang, K. K. Tamma, Generalized heat conduction model involving imperfect thermal contact surface: Application of the GSSSS-1 differential-algebraic equation time integration, *Int. J. Heat Mass Tran.,* **116** (2018), 889–896. https://doi.org/10.1016/j.ijheatmasstransfer.2017.09.081
15. 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

16. *Automatic Differentiation in PyTorch*, *Adv. Neural Inf. Process. Syst.*, 2017. Available from: https://openreview.net/forum?id=BJJsrmfCZ

17. E. Haghighat, A. C. Bekar, E. Madenci, R. Juanes, A nonlocal physics-informed deep learning framework using the peridynamic differential operator, *Comput. Method. Appl. Mech. Eng.,* **385** (2021), 114012. https://doi.org/10.1016/j.cma.2021.114012

18. H. Bararnia, M. Esmaeilpour, On the application of physics informed neural networks (PINN) to solve boundary layer thermal-fluid problems, *Int. Commun. Heat Mass Tran.,* **132** (2022), 105890. https://doi.org/10.1016/j.icheatmasstransfer.2022.105890

19. L. Ning, Z. Cai, H. Dong, Y. Liu, W. Wang, Physics-informed neural network frameworks for crack simulation based on minimized peridynamic potential energy, *Comput. Method. Appl. Mech. Eng.,* **417** (2023), 116430. https://doi.org/10.1016/j.cma.2023.116430

20. M. Baldan, P. D. Barba, D. A. Lowther, Physics-Informed Neural Networks for Inverse Electromagnetic Problems, *IEEE T. Magn.,* **59** (2023), 1–5. https://doi.org/10.1109/TMAG.2023.3247023

21. 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

22. H. J. Park, G. H. 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

23. P. Hansbo, M. G. Larson, Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method, *Comput. Method. Appl. Mech. Eng.,* **191** (2002), 1895–1908. https://doi.org/10.1016/S0045-7825(01)00358-9

24. D. Leguillon, E. Sanchez-Palencia, *Computation of singular solutions in elliptic problems and elasticity*, New York: John Wiley & Sons, Inc., 1987. http://doi.org/10.5555/39281

25. Evans, C. Lawrence, *Partial differential equations*, 2 Eds., Providence, RI: Am.Math.Soc., 2010. http://doi.org/10.1090/gsm/019

26. J. Wloka, *Partial differential equations*, Cambridge: Cambridge Univ.Press, 1987. http://doi.org/10.1017/CBO9781139171755

27. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.,* **2** (1989), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

28. R. E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, *Acta Numer.,* **7** (1998), 1–49. https://doi.org/10.1017/S0962492900002804

29. S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating PDEs, *IMA J. Numer. Anal.,* **43** (2022), 1–43. https://doi.org/10.1093/imanum/drab093

30. S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, *IMA J. Numer. Anal.,* **42** (2021), 981–1022. https://doi.org/10.1093/imanum/drab032

31. T. D. Ryck, A. D. Jagtap, S. Mishra, Error estimates for physics-informed neural networks approximating the Navier-Stokes equations, *IMA J. Numer. Anal.,* **44** (2023), 83–119. https://doi.org/10.1093/imanum/drac085

32. T. D. 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

33. K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, *Proc. IEEE Int. Conf. Comput. Vis.,* (2015), 1026–1034. https://doi.org/10.1109/ICCV.2015.123

34. D. W. Marquardt, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *J. Soc. Ind. Appl. Math.,* **11** (1963), 431–441. https://doi.org/10.1137/0111030

35. D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Program.,* **45** (1989), 503–528. https://doi.org/10.1007/BF01589116

36. 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

37. J. F. Urbán, P. Stefanou, J. A. Pons, Unveiling the optimization process of Physics Informed Neural Networks: How accurate and competitive can PINNs be?, *J. Comput. Phys.,* **523** (2025), 113656. https://doi.org/10.1016/j.jcp.2024.113656

38. C. Liu, L. Zhu, M. Belkin, Loss landscapes and optimization in over-parameterized non-linear systems and neural networks, *Appl. Comput. Harmon. Anal.,* **59** (2022), 85–116. https://doi.org/10.1016/j.acha.2021.12.009

AIMS Press