A deep Nitsche method for elliptic interface problem with implicit jump condition

Shuxuan Liang[a], Hyeokjoo Park[b], Gwanghyun Jo[a,*]

[a]*Department of Mathematical Data Science, Hanyang University ERICA, 55, Hanyangdaehak-ro, Sangnok-gu, Ansan-si, 15588, Gyeonggi-do, Republic of Korea*
[b]*School of Mathematics and Computing (Computational Science and Engineering), Yonsei University, 50, Yonsei-ro, Seodaemoon-gu, Seoul, 03722, Republic of Korea*

Abstract

In this paper, we propose a new deep Ritz method for elliptic interface problem with implicit jump condition. The parameters are defined by solving minimizing problem of energy problem where boundary condition is treated by Nitsche technique.

Keywords: Deep Nitsche method, neural networks, elliptic interface problem

1. Introduction

Elliptic interface problems arise in a wide range of engineering and scientific applications, including heat conduction in composite materials, Poisson-type equations in fluid and porous media [14, 27, 16]. For example, imperfectly bonded composite interfaces can produce jumps in the temperature field. Analogously, in fluid and porous media, discontinuities in material coefficients such as conductivity or permeability across interfaces lead to jumps in the solution and its normal flux for Poisson-type equations [15]. These features underscore the need for accurate and efficient numerical methods to enable reliable multiphysics predictions.

Let us categorize the two types of jump conditions. When the explicit jump conditions are known, a variety of numerical techniques have been developed to address elliptic interface problems in finite element/difference method (FEM/FDM) communities. The Immersed Interface Method (IIM) [6] and Immersed Finite Element Method (IFE) [10] directly incorporate interface conditions into discretizations, enabling accurate flux computation across irregular geometries. More recently, high-order finite element methods combined with penalty terms have been proposed to handle elliptic problems with nonhomogeneous jump conditions [4, 1]. Despite these advances, most existing techniques

*Corresponding Author.
Email addresses: liang.shuxuan2024@gmail.com (Shuxuan Liang), hyeokjoopark@yonsei.ac.kr (Hyeokjoo Park), gwanghyun@hanyang.ac.kr (Gwanghyun Jo)

rely heavily on body-fitted meshes or explicit knowledge of interface conditions, which limits their efficiency and generality when facing complex geometries or partially unknown interfacial behaviors.

However, developing numerical methods become more complicated when the interface conditions are implicit. These problems can be found in medical imaging of cancer cells using magnetic resonance electrical impedance tomography (MREIT) [2, 3], in electrochemotherapy [21] wgere voltage jumps across the cell membrane are observed, in elastic bodies with spring-type jumps associated with stress [18, 20], and in heat conduction across imperfectly matched material interfaces [17, 11].When (numerically) solve problems with implicit jump conditions, there are several challenges. One of the main difficulties comes from the fact that explicit jump profile is not known. Therefore, some people use iteration methods where jump profile is treated as independent unknowns updated at each iteration step. This difficulties may be handled by judiciously defined bilinear form where jump conditions are implicitly imposed, see .

The recent development of deep learning has motivated alternative strategies for solving PDEs. Physics-Informed Neural Networks (PINNs) [26] incorporate residuals of PDE and boundary conditions into loss functions, thereby requiring no mesh generation. The Deep Ritz method [9] formulates PDEs as energy minimization problems, providing a variational perspective. The Deep Ritz method represents the trial function in a variational formulation (e.g., minimizing the energy functional for Poisson-type PDEs) by a deep neural network and seeks to minimize the resulting Ritz functional where the boundary conditions are enforced by adding penalty terms. Recently, Liao and Ming introduced the Deep Nitsche method [22], which incorporates Nitsche's technique into the Deep Ritz framework to enforce Dirichlet conditions weakly and robustly.

While these neural approaches demonstrate flexibility for high-dimensional PDEs, they remain largely unexplored for interface problems, and particularly inadequate for implicit jump conditions where the coupling of unknowns across interfaces cannot be handled by standard formulations. This gap motivates the present study.

To bridge the above gap, we propose a Deep Nitsche framework for elliptic interface problems with implicit jump conditions. Our main contributions are: 1) Extension of Nitsche's method to implicit interfaces: we develop a weak enforcement strategy that avoids explicit jump corrections and circumvents additional degrees of freedom; 2) Integration with the Deep Ritz variational formulation: the approach leverages the approximation power of neural networks, enabling adaptation to irregular geometries and high-dimensional domains without meshing; 3) Stability through interface penalty and symmetrization terms: the proposed energy functional incorporates stabilization mechanisms that ensure accuracy and consistency even in the presence of discontinuous solutions.

We implement the proposed method within a neural network-based framework and validate its performance on benchmark problems, demonstrating its ability to accurately capture implicit and nonlinear jump conditions. Beyond elliptic models, this framework provides a potential foundation for addressing more general multiphysics coupling problems that remain challenging for exist-

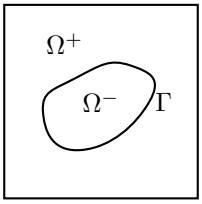


Figure 1: A domain Ω with interface Γ.

ing deep learning-based solvers.

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 2.3. In Section 3, we do some experiments and get some numerical results. The conclusion follows in Section 4.

2. Preliminaries

We follow the usual definition and notation of the Sobolev spaces (cf. []). Given a geometric object G and $s \geq 0$, let $\|\cdot\|_{s,G}$ (resp. $|\cdot|_{s,G}$) denote the usual norm (resp. seminorm) of the Sobolev space $H^s(G)$. Let $(\cdot,\cdot)_G$ denote the usual L^2-inner product over G. Note that $H^0(G) = L^2(G)$ and $\|\cdot\|_{0,G}^2 = (\cdot,\cdot)_G$.

Let Ω be a bounded domain in $\mathbb{R}^d$ with $d = 2, 3$, divided into two subdomains Ω^+ and Ω^- by a C^2-interface Γ. For simplicity, we assume that the closure of Ω^- is contained in the interior of Ω and $\partial\Omega^- = \Gamma$; see Figure 1. For $s = \pm$, let $\boldsymbol{n}^s$ denote the unit outward normal vector to $\partial\Omega^s$. The governing equations are given by

$$\begin{cases} -\nabla\cdot\beta\nabla u = f & \text{in } \Omega^+\cup\Omega^-, \\ u = g & \text{on } \partial\Omega, \\ [u]_\Gamma = \alpha\nabla u|_{\Omega^+}\cdot\boldsymbol{n}^+ & \text{on } \Gamma, \\ [\beta\nabla u\cdot\boldsymbol{n}]_\Gamma = 0 & \text{on } \Gamma, \end{cases} \tag{1}$$

where $f \in L^2(\Omega)$ and $g \in H^{3/2}(\partial\Omega)$, α is a positive constant, β is a positive piecewise constant, i.e., $\beta = \beta^+$ in Ω^+ and $\beta = \beta^-$ in Ω^- for some positive constants β^+ and β^-, and $[\cdot]_\Gamma$ denotes the jumps along the interface Γ given by

$$[u]_\Gamma = u|_{\Omega^+} - u|_{\Omega^-}, \quad [\beta\nabla u\cdot\boldsymbol{n}]_\Gamma = \beta^+\nabla u|_{\Omega^+}\cdot\boldsymbol{n}^+ + \beta^-\nabla u|_{\Omega^-}\cdot\boldsymbol{n}^-.$$

Let us next present a weak formulation of the problem (1). To do this, we introduce a broken Sobolev space

$$H^r(\Omega^\pm) = \{v \in L^2(\Omega) : v|_{\Omega^s} \in H^r(\Omega^s),\ \forall s = \pm\},$$

equipped with the norm $\|\cdot\|_{r,\Omega^\pm}$ and seminorm $|\cdot|_{r,\Omega^\pm}$ given by

$$|v|_{r,\Omega^\pm}^2 = |v|_{r,\Omega^+}^2 + |v|_{r,\Omega^-}^2, \quad \|v\|_{r,\Omega^\pm}^2 = \|v\|_{0,\Omega}^2 + |v|_{r,\Omega^\pm}^2.$$

We also define the following subspace of $H^1(\Omega^\pm)$:

$$H_0^1(\Omega^\pm) = \{v \in H^1(\Omega^\pm) : v|_{\partial\Omega} = 0\}.$$

Then the weak formulation of the problem (1) is written as follows: Find $u \in H^1(\Omega^\pm)$ such that $u|_{\partial\Omega} = g$ and

$$(\beta\nabla u, \nabla v)_{\Omega^+\cup\Omega^-} + \frac{\beta^+}{\alpha}([u]_\Gamma, [v]_\Gamma)_\Gamma = (f, v)_\Omega, \quad \forall v \in H_0^1(\Omega^\pm). \tag{2}$$

The well-posedness and regularity result for the problem above has been established in [5].

Theorem 2.1. *The problem (2) has a unique solution u. Furthermore, if Ω is convex or has C^2-smooth boundary $\partial\Omega$, then $u \in H^2(\Omega^\pm)$ and*

$$\|u\|_{2,\Omega^\pm} \le C(\|f\|_{0,\Omega} + \|g\|_{3/2,\partial\Omega}),$$

where $C > 0$ is a constant depending only on Ω, Γ, α and β.

3. Method

In this section, we describe the proposed method for the problem (2) and its error analysis. We describe the deep Nitsche PINN method for the governing equation. The overall process is illustrated in Figure **??**.

3.1. Neural network structure for piecewise smooth functions

In order to approximate functions having discontinuities across the interface Γ, we adopt the idea in [13, 25, 24], that is, we will approximate an extension $\widetilde{v}$ of $v \in H^1(\Omega^\pm)$ to an H^1-function on a higher dimensional domain $\Omega \times [-1, 1]$:

$$\widetilde{v}(\boldsymbol{x}, z) \coloneqq \frac{1+z}{2}v^+(\boldsymbol{x}) + \frac{1-z}{2}v^-(\boldsymbol{x}), \quad (\boldsymbol{x}, z) \in \Omega \times [-1, 1],$$

where, for each $s = \pm$, v^s is a Sobolev extension of $v|_{\Omega^s}$ to Ω (cf. [28, 8]).

Given a positive integer n, an augmented neural network having n hidden layers on $\Omega \times [-1, 1]$ is defined by a real-valued function $\widetilde{\varphi}$ on $\Omega \times [-1, 1]$ such that

$$\widetilde{\varphi}(\boldsymbol{x}, z) = (A_n \circ \sigma \circ \cdots \sigma \circ A_1 \circ \sigma \circ A_0)(\boldsymbol{x}), \quad (\boldsymbol{x}, z) \in \Omega \times [-1, 1],$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth activation function applied componentwisely, and A_i for each $i = 0, 1, \cdots, n$ is an affine mapping from $\mathbb{R}^{k_i}$ into $\mathbb{R}^{k_{i+1}}$ with $k_0 = d + 1$ and $k_{n+1} = 1$:

$$A_i(\boldsymbol{z}) = \mathbf{W}_i\boldsymbol{z} + \boldsymbol{b}_i, \quad \mathbf{W}_i \in \mathbb{R}^{k_{i+1}\times k_i}, \ \boldsymbol{b}_i \in \mathbb{R}^{k_{i+1}}.$$

Here, the matrices $\mathbf{W}_0, \cdots, \mathbf{W}_n$ are called the weights, and the vectors $\boldsymbol{b}_0, \cdots, \boldsymbol{b}_n$ is called the bias. The parameter θ of the neural network φ is then given by $\theta = (\mathbf{W}_0, \cdots, \mathbf{W}_n, \boldsymbol{b}_0, \cdots, \boldsymbol{b}_n)$ and we may write $\widetilde{\varphi} = \widetilde{\varphi}_\theta$.

We then let $\widetilde{\mathcal{N}}_n$ denote the set of all augmented neural networks having n hidden layers on $\Omega \times [-1,1]$ as given above. We also let $\mathcal{N}_n^{\pm}$ denote the set of all piecewise neural network functions φ_θ on Ω given by

$$\varphi_\theta(\boldsymbol{x}) = \widetilde{\varphi}_\theta(\boldsymbol{x},1)\chi_{\Omega^+}(\boldsymbol{x}) + \widetilde{\varphi}_\theta(\boldsymbol{x},-1)\chi_{\Omega^-}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$

for $\widetilde{\varphi}_\theta \in \widetilde{\mathcal{N}}_n$. Throughout the paper, we assume that the piecewise neural networks in $\mathcal{N}_n^{\pm}$ satisfy the following inverse-type inequality: There exists a constant $C_{\mathrm{inv}} > 0$ such that

$$\|\nabla\varphi_\theta\|_{0,\partial\Omega} \le C_{\mathrm{inv}}\|\nabla\varphi_\theta\|_{0,\Omega^+}, \quad \forall \varphi_\theta \in \mathcal{N}_n^{\pm}. \tag{3}$$

3.2. Nitsche formulation

We first derive the Nitsche formulation for (1). Let $u \in H^{3/2+\delta}(\Omega^{\pm})$ be the solution of (2). Then, for any $v \in H^1(\Omega^{\pm})$,

$$(\beta\nabla u, \nabla v)_{\Omega^+\cup\Omega^-} + \frac{\beta^+}{\alpha}([u]_\Gamma,[v]_\Gamma)_\Gamma = (f,v)_\Omega + (\beta\nabla u\cdot\boldsymbol{n}_{\partial\Omega}, v)_{\partial\Omega},$$

where $\boldsymbol{n}_{\partial\Omega}$ denotes the outward normal vector to $\partial\Omega$. Since $u = g$ on $\partial\Omega$,

$$\begin{aligned}&(\beta\nabla u, \nabla v)_{\Omega^+\cup\Omega^-} + \frac{\beta^+}{\alpha}([u]_\Gamma,[v]_\Gamma)_\Gamma \\ &\quad = (f,v)_\Omega + (\beta\nabla u\cdot\boldsymbol{n}_{\partial\Omega}, v)_{\partial\Omega} + (\beta\nabla v\cdot\boldsymbol{n}_{\partial\Omega}, u-g)_{\partial\Omega} + \eta(u-g,v)_{\partial\Omega},\end{aligned} \tag{4}$$

for any $v \in H^1(\Omega^{\pm})$ with $(\nabla v)|_{\partial\Omega} \in L^2(\partial\Omega)$ and any constant $\eta > 0$.

Motivated by this derivation, we propose the deep Nitsche method for (1) as follows: Find $u \in \mathcal{N}_n^{\pm}$ such that

$$u_\theta = \underset{v_\theta\in\mathcal{N}_n^{\pm}}{\operatorname{argmin}} \left(\frac{1}{2}A(v_\theta,v_\theta) - F(v_\theta)\right), \tag{5}$$

where $A(\cdot,\cdot)$ and $F(\cdot)$ are the bilinear form and the linear functional given by

$$\begin{aligned}A(u,v) &= (\beta\nabla u,\nabla v)_{\Omega^+\cup\Omega^-} + \frac{\beta^+}{\alpha}([u]_\Gamma,[v]_\Gamma)_\Gamma \\ &\quad - (\beta\nabla u\cdot\boldsymbol{n}_{\partial\Omega},v)_{\partial\Omega} - (\beta\nabla v\cdot\boldsymbol{n}_{\partial\Omega},u)_{\partial\Omega} + \eta\beta^+(u,v)_{\partial\Omega},\\ F(v) &= (f,v_\theta)_\Omega - (\beta\nabla v_\theta\cdot\boldsymbol{n}_{\partial\Omega},g)_{\partial\Omega} + \eta\beta^+(g,v_\theta)_{\partial\Omega},\end{aligned}$$

with a sufficiently large user-defined parameter $\eta > 0$, respectively.

Let $|||\cdot|||$ be the energy norm on $H^1(\Omega^{\pm})$ given by

$$|||v|||^2 = \|\beta^{1/2}\nabla v\|_{0,\Omega^+\cup\Omega^-}^2 + \frac{\beta^+}{\alpha}\|[v]_\Gamma\|_{0,\Gamma}^2 + \beta^+\|v\|_{0,\partial\Omega}^2.$$

Note that it is indeed a norm due to [25, Lemma 1].

The existence and uniqueness of the minimizer for (5) is a direct consequence of the following lemma.

Lemma 3.1. *For any $\eta > 0$, the bilinear form $A(\cdot,\cdot)$ is continuous on $\mathcal{N}_n^{\pm}$ with respect to the norm $|||\cdot|||$:*

$$|A(v_\theta, w_\theta)| \le \max\{1,\eta\}\max\{1, C_{\rm inv}\}|||v_\theta|||\,|||w_\theta|||, \quad \forall v_\theta, w_\theta \in \mathcal{N}_n^{\pm}. \tag{6}$$

Furthermore, if $\eta \ge 5C_{\rm inv}/2$ then $A(\cdot,\cdot)$ is coercive on $\mathcal{N}_n^{\pm}$ with respect to the norm $|||\cdot|||$:

$$\frac{1}{2}\min\{1, C_{\rm inv}\}|||v_\theta|||^2 \le A(v_\theta, v_\theta), \quad \forall v_\theta \in \mathcal{N}_n^{\pm}. \tag{7}$$

Proof. Since $\beta = \beta^+$ on $\partial\Omega$, the inverse-type inequality (3) implies that

$$|(\beta\nabla v_\theta \cdot \boldsymbol{n}_{\partial\Omega}, w_\theta)_{\partial\Omega}| \le C_{\rm inv}\|\beta^{1/2}\nabla v_\theta\|_{\Omega^+}(\beta^+)^{1/2}\|w_\theta\|_{0,\partial\Omega},$$

for any $v_\theta, w_\theta \in \mathcal{N}_n^{\pm}$. This implies (6). In addition, by Young's inequality with modulus $t > 0$, for any $v_\theta \in \mathcal{N}_n^{\pm}$,

$$(\beta\nabla v_\theta \cdot \boldsymbol{n}_{\partial\Omega}, v_\theta)_{\partial\Omega} \ge -\frac{C_{\rm inv}t}{2}\|\beta^{1/2}\nabla v_\theta\|_{\Omega^+}^2 - \frac{\beta^+}{2t}\|v_\theta\|_{0,\partial\Omega}^2.$$

Thus, if $\eta \ge 5C_{\rm inv}/2$ and $t = (2C_{\rm inv})^{-1}$ then

$$\begin{aligned} A(v_\theta, v_\theta) &\ge \frac{1}{2}\|\beta^{1/2}\nabla v_\theta\|_{0,\Omega^+\cup\Omega^-}^2 + \frac{\beta^+}{\alpha}\|[v_\theta]_\Gamma\|_{0,\Gamma}^2 + \left(\eta - \frac{1}{t}\right)\beta^+\|v_\theta\|_{0,\partial\Omega}^2 \\ &\ge \frac{1}{2}\min\{1, C_{\rm inv}\}|||v_\theta|||^2. \end{aligned}$$

This completes the proof of the lemma. □

3.3. Error analysis

We present an error analysis for the proposed method (5). Proceeding as in the proof of [22, Lemma 2.1], together with the consistency (4), the following lemma can be proved.

Lemma 3.2. *Let $u \in H^{3/2+\delta}(\Omega^{\pm})$ with $\delta > 0$ be the solution of (2), and u_θ the solution of (5). Then we have*

$$L(u - u_\theta) = \min_{v_\theta \in \mathcal{N}_n^{\pm}} L(u - v_\theta),$$

where $L(v) = A(v, v)$.

We are now ready to derive an error estimate for (5) as follows.

Theorem 3.3. *Let $u \in H^{3/2+\delta}(\Omega^{\pm})$ with $\delta > 0$ be the solution of (2), and u_θ the solution of (5). Then we have*

$$|||u - u_\theta|||^2 \le C \inf_{v_\theta \in \mathcal{N}_n^{\pm}} \left(|||u - v_\theta|||^2 + \beta^+\|\nabla(u - v_\theta)\cdot\boldsymbol{n}\|_{0,\partial\Omega}^2\right),$$

where $C > 0$ is a constant depending only on $C_{\rm inv}$ and η.

Proof. Let $v_\theta \in \mathcal{N}_n^\pm$ be given arbitrarily. Let $L(\cdot) = A(\cdot,\cdot)$. Then, by the triangle inequality and the coercivity (7),

$$\begin{aligned}
|||u - u_\theta||| &\le |||u - v_\theta||| + |||v_\theta - u_\theta||| \\
&\le |||u - v_\theta||| + C(L(v_\theta - u_\theta))^{1/2} \\
&\le |||u - v_\theta||| + C(L(v_\theta - u))^{1/2} + C(L(u - u_\theta))^{1/2}.
\end{aligned}$$

Note that $L(v_\theta - u) = L(u - v_\theta)$, and $L(u - u_\theta) \le L(u - v_\theta)$ by Lemma 3.2. This yields

$$|||u - u_\theta||| \le |||u - v_\theta||| + 2(L(u - v_\theta))^{1/2}.$$

By the definition of $L(\cdot)$,

$$L(u - v_\theta) \le \max\{\eta, 2\}|||u - v_\theta|||^2 + \beta^+ \|\nabla(u - v_\theta)\cdot \boldsymbol{n}\|^2_{0,\partial\Omega}.$$

Now the conclusion follows by combining the last two inequalities and the assumption that $v_\theta \in \mathcal{N}_n^\pm$ is chosen arbitrarily. □

Remark 3.4. Following the argument in [25], if the solution u is sufficiently smooth on each of the subdomain Ω^+ and Ω^-, then for any $\epsilon > 0$ one can find $v_\theta \in \mathcal{N}_n^\pm$ with $n = 2$ such that

$$|||u - v_\theta|||^2 + \beta^+ \|\nabla(u - v_\theta)\cdot \boldsymbol{n}\|^2_{0,\partial\Omega} \le \epsilon.$$

4. Numerical results

In this subsection, 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 [12], with biases set to zero.

We present three samples, each featuring a different interface shape: circle, perturbed circle, and three disjoint circles. In Example 4.1-4.3, the L^2- and L^∞-errors against the exact solutions are computed, where we observe reasonable convergence as the parameters increase. In Example 4.2, since the analysitc solution is unknown in this case, we validate the Nitsche-predicted solution by comparing it with the FEM solution.

4.1. Choices of optimizers and training sample sets

In this section, 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 Limited-memory Broyden-Fletcher-Goldfard-Shanno (L-BFGS) optimizer [23] with 2,000 epochs, a learning rate of 0.1, and a strong Wolfe line search option. The second is Adam optimizer [19] with 2000 epochs, a learning rate of 0.001.

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 by Gauss-Legendre points [7]. For example, Gauss-Legendre points of order m for *inner* Ω^+ are defined by

$$(\beta\nabla u_\theta, \nabla u_\theta)_{\Omega^+} \approx \sum_{(x_i,y_j)\in\Omega^+} \beta^+ w_i w_j \left|\nabla u_\theta(x_i, y_j)\right|^2 .$$

For boundary points ($4m$-number), we use one-dimensional Gauss–Legendre nodes along each edge to impose Dirichlet conditions smoothly and stably. On the interface ($16m$-number), uniform points with equal angular spacing in polar coordinates are sampled to accurately enforce jump conditions during training.

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

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

where $r_0 = 0.5$ and $\alpha = 1$. We consider two different thermal conductivity contrasts: $(k^+, k^-) = (1, 10)$ or $(k^+, k^-) = (10, 1)$. We set the penalty parameter $\eta = 1000$. We have tallied the absolute errors in L^∞ norm and L^2 norm for the numerical results, and the results are summarized in Table 1 and Table 2, respectively.

We compare loss curves with increasing training epochs for the L-BFGS optimizer in Figure 2. The loss function decreases to the lower level by the L-BFGS optimizer. Consequently, we adopted the L-BFGS optimizer for all experiments in this study. Next, we compare the L^2 and L^∞ errors computed as the number of penalty parameter η increasing. We report errors in Figure 3. Overall, the errors start to stabilize when η was greater than or equal to 1000. Therefore, we set the penalty parameter $\eta = 1000$.

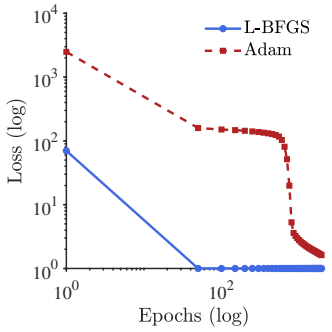


Figure 2: Evolution of loss function (log-scale) with increasing number of epochs for the L-BFGS and Adam optimizers.

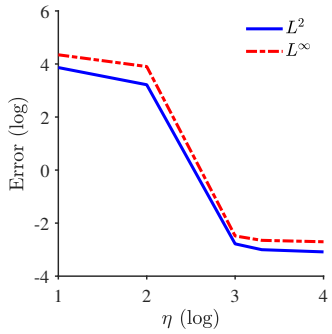


Figure 3: The absolute errors in L^2 and L^∞ norm with respect to different penalty coefficients $\eta = 10, 10^2, 10^3$ and 10^4.

The main idea of the Deep Ritz method is to use a deep neural network as a trial function in the classical Ritz framework to approximate u_θ by minimizing the energy functional

$$\min_{u_\theta \in \mathcal{N}_\theta^{\pm}(\Omega)} J(u_\theta) = \frac{1}{2} A^*(u_\theta, u_\theta) - F^*(u_\theta) + \eta \, \|u - g\|^2_{L^2(\partial\Omega)}$$

where

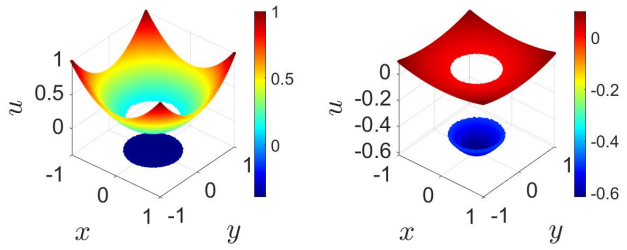
$$\begin{aligned} A^*(u, v) &= (\beta \nabla u, \nabla v)_{\Omega^+ \cup \Omega^-} + \frac{\beta^+}{\alpha} ([u]_\Gamma, [v]_\Gamma)_\Gamma \\ F^*(v) &= (f, v)_\Omega. \end{aligned}$$

The method works well for smooth regions and simple boundary conditions, but its accuracy and stability often get worse when there are discontinuities or complex interface conditions. The proposed Deep Nitsche method extends this idea by adding the weak form of interface conditions into the neural network framework. This improvement allows the method to achieve consistent and stable coupling at the interface without explicitly enforcing boundary conditions.

As shown in Figure 4 and Table 1 and 2, the numerical results shows that the proposed method is better than the Deep Ritz method in both accuracy and stability. For the parameter choice $(N, m) = (80, 64)$, the L^2 error stays below 1.19×10^{-3}, and the corresponding CPU time are 499 (s) and 243 (s), showing very good computational efficiency. In general, the deep Nitsche method gives much higher accuracy than the deep Ritz method, especially for elliptic interface problems with implicit jump condition.

Example 4.2 (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

$$\begin{cases} u = 1, & \text{when } x = -1, \\ u = 0, & \text{when } x = 1, \\ u = \frac{1}{2} - \frac{1}{2}x, & \text{when } y = 1 \text{ or } y = -1 \end{cases}$$



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

Figure 4: Graphs of the solutions obtained by deep Nitsche in Example 4.1 with thermal conductivity $(k^+, k^-) = (1, 10)$ or $(10, 1)$

N	m	Deep Nitsche		
		$\|T_\theta - T\|_{L^\infty(\Omega)}$	$\|T_\theta - T\|_{L^2(\Omega)}$	CPU time (s)
20	32	1.23E-02	5.05E-03	398.64
	64	3.00E-03	1.13E-03	253.96
	128	1.31E-03	5.72E-04	530.61
40	32	1.28E-02	5.25E-03	562.7
	64	2.72E-03	1.07E-03	574.2
	128	1.98E-03	9.44E-04	332.16
80	32	1.38E-02	5.23E-03	555.5
	64	3.38E-03	1.55E-03	342.94
	128	1.19E-03	6.05E-04	578.62

Table 1: The L^2 and L^∞ errors and CPU time of deep Nitsche method in Example 4.1 where $(k^+, k^-) = (1, 10)$ and $\eta = 1000$.

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

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

Example 4.3 (Three circles shaped interface). (...)

5. Conclusion

(...)

Acknowledgements

This work is supported by National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (No. 2020R1C1C1A01005396 and No. RS-2025-00513210).

N	m	Deep Nitsche		
		$\|T_\theta - T\|_{L^\infty(\Omega)}$	$\|T_\theta - T\|_{L^2(\Omega)}$	CPU time (s)
20	32	1.15E-02	3.31E-03	277.79
	64	3.53E-03	8.34E-04	212.81
	128	1.98E-03	4.07E-04	372.72
40	32	1.22E-02	3.38E-03	424.93
	64	3.64E-03	8.70E-04	494.49
	128	1.55E-03	4.12E-04	403.74
80	32	1.27E-02	3.48E-03	292.94
	64	3.84E-03	8.40E-04	384.91
	128	2.78E-03	5.83E-04	315.39

Table 2: The L^2 and L^∞ errors and CPU time of deep Nitsche method in Example 4.1 where $(k^+, k^-) = (10, 1)$ and $\eta = 1000$.

Data availability

No data was used for the research described in the article.

References

[1] S. Adjerid, I. Babuška, R. Guo, and T. Lin, *An enriched immersed finite element method for interface problems with nonhomogeneous jump conditions*, Computer Methods in Applied Mechanics and Engineering, 404 (2023), p. 115770.

[2] H. Ammari, J. Garnier, L. Giovangigli, W. Jing, and J.-K. Seo, *Spectroscopic imaging of a dilute cell suspension*, Journal de Mathématiques Pures et Appliquées, 105 (2016), pp. 603–661.

[3] H. Ammari, J. Garnier, H. Kang, M. Lim, and S. Yu, *Generalized polarization tensors for shape description*, Numerische Mathematik, 126 (2014), pp. 199–224.

[4] C. Attanayake, S.-H. Chou, and Q. Deng, *High-order enriched finite element methods for elliptic interface problems with discontinuous solutions*, arXiv preprint arXiv:2204.07665, (2022).

[5] F. Ben Belgacem, C. Bernardi, F. Jelassi, and M. Mint Brahim, *Finite element methods for the temperature in composite media with contact resistance*, J. Sci. Comput., 63 (2015), pp. 478–501.

[6] H. Cho and M. Kang, *Fully implicit and accurate treatment of jump conditions for two-phase incompressible navier–stokes equations*, Journal of Computational Physics, 445 (2021), p. 110587.

[7] P. J. Davis and P. Rabinowitz, *Methods of numerical integration*, Courier Corporation, 2007.

[8] L. C. Evans, *Partial differential equations*, vol. 19, American Mathematical Society, 2022.

[9] Y. B. EW, *The deep ritz method: A deep learning-based numerical algorithm for solving variational problems*, Communications in Mathematics and Statistics, 6 (2018), pp. 1–12.

[10] Y. Gong, B. Li, and Z. Li, *Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 472–495.

[11] D. W. Hahn and M. N. Özisik, *Heat conduction*, John Wiley & Sons, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[13] W.-F. Hu, T.-S. Lin, and M.-C. Lai, *A discontinuity capturing shallow neural network for elliptic interface problems*, J. Comput. Phys., 469 (2022), pp. Paper No. 111576, 11.

[14] W. Jäger and A. Mikelić, *On the boundary conditions at the contact interface between a porous medium and a free fluid*, Annali della Scuola Normale Superiore di Pisa-Classe di Scienze, 23 (1996), pp. 403–465.

[15] G. Jo and D. Y. Kwak, *Enriched p1-conforming methods for elliptic interface problems with implicit jump conditions*, Advances in Mathematical Physics, 2018 (2018), p. 9891281.

[16] C. Jourdana and P. Pietra, *An interface formulation for the poisson equation in the presence of a semiconducting single-layer material*, ESAIM: Mathematical Modelling and Numerical Analysis, 58 (2024), pp. 833–856.

[17] J. Kačur and R. Van Keer, *Numerical method for a class of parabolic problems in composite media*, Numerical Methods for Partial Differential Equations, 9 (1993), pp. 711–731.

[18] S. Kim, J. K. Seo, and T. Ha, *A nondestructive evaluation method for concrete voids: frequency differential electrical impedance scanning*, SIAM Journal on Applied Mathematics, 69 (2009), pp. 1759–1771.

[19] D. P. Kingma, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[20] D. Kyeong et al., *An immersed finite element method for the elasticity problems with displacement jump*, Advances in Applied Mathematics and Mechanics, 9 (2017), pp. 407–428.

[21] I. Lackovic, R. Magjarević, and D. Miklavcic, *Analysis of tissue heating during electroporation based therapy: A 3d fem model for a pair of needle electrodes*, in 11th Mediterranean Conference on Medical and Biomedical Engineering and Computing 2007: MEDICON 2007, 26-30 June 2007, Ljubljana, Slovenia, Springer, 2007, pp. 631–634.

[22] Y. Liao and P. Ming, *Deep Nitsche method: deep Ritz method with essential boundary conditions*, Commun. Comput. Phys., 29 (2021), pp. 1365–1384.

[23] D. C. Liu and J. Nocedal, *On the limited memory bfgs method for large scale optimization*, Mathematical programming, 45 (1989), pp. 503–528.

[24] H. Oh and G. Jo, *Physics-informed neural network for the heat equation under imperfect contact conditions and its error analysis*, AIMS Math., 10 (2025), pp. 7920–7940.

[25] H. Park and G. Jo, *A physics-informed neural network based method for the nonlinear Poisson-Boltzmann equation and its error analysis*, J. Comput. Phys., 522 (2025), pp. Paper No. 113579, 14.

[26] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational physics, 378 (2019), pp. 686–707.

[27] Z. M. Seyidmamedov and E. Ozbilge, *A mathematical model and numerical solution of interface problems for steady state heat conduction*, Mathematical Problems in Engineering, 2006 (2006), p. 020898.

[28] J. Wloka, *Partial differential equations*, Cambridge University Press, Cambridge, 1987. Translated from the German by C. B. Thomas and M. J. Thomas.