Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

A Deep Uzawa-Lagrange Multiplier Approach for Boundary Conditions in PINNs and Deep Ritz Methods

Charalambos G. Makridakis * [1,2], Aaron Pim † [3], and Tristan Pryer ‡ [3]

[1]DMAM, University of Crete / Institute of Applied and Computational Mathematics, FORTH, Heraklion 700 13, Crete, Greece.
[2] MPS, University of Sussex, Brighton BN1 9QH, UK.
[3]Department of Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, UK.

Abstract. We introduce a deep learning-based framework for weakly enforcing boundary conditions in the numerical approximation of partial differential equations. Building on existing physics-informed neural network and deep Ritz methods, we propose the Deep Uzawa algorithm, which incorporates Lagrange multipliers to handle boundary conditions effectively. This modification requires only a minor computational adjustment but ensures enhanced convergence properties and provably accurate enforcement of boundary conditions, even for singularly perturbed problems. We provide a comprehensive mathematical analysis demonstrating the convergence of the scheme and validate the effectiveness of the Deep Uzawa algorithm through numerical experiments, including high dimensional, singularly perturbed problems and those posed over non-convex domains.

Keywords:
Physics-informed neural networks,
Deep Ritz method,
Uzawa algorithm,
Lagrange multipliers,
Boundary condition enforcement.

Article Info.:
Volume: 4
Number: 3
Pages: 166 - 191
Date: September/2025
doi.org/10.4208/jml.250107

Article History:
Received: 07/01/2025
Accepted: 09/07/2025

Communicated by:
Zhi-Qin Xu

1 Introduction

The numerical approximation of partial differential equations (PDEs) using artificial neural networks (ANNs) has gained significant attention in recent years [4, 9, 14, 18, 20]. This surge is largely attributed to the success of deep learning in various complex tasks [10, 13]. In the context of solving PDEs, neural network-based methods such as the deep Ritz approach [4], which approximates solutions by minimising the Dirichlet energy, and the physics-informed neural networks (PINNs) [18], which minimise the L^2-norm of the residuals are prominent examples.

Despite their success, a challenge in these methods lies in the enforcement of boundary conditions. While classical numerical methods also face difficulties in this regard, the non-standard nature of neural network approximation spaces makes this issue particularly pronounced. Standard penalty approaches often require large penalty weights to enforce boundary conditions accurately, resulting in ill-conditioned optimisation problems that are difficult to tune and can lead to suboptimal solutions. This issue is particularly

*C.G.Makridakis@iacm.forth.gr
†A.R.Pim@bath.ac.uk
‡Corresponding author. tmp38@bath.ac.uk

severe in problems involving singular perturbations or complex domains. Additionally, the practical implementation of boundary conditions in ANN-based methods poses challenges, as accurately capturing boundary data within the neural network's architecture often proves difficult.

To address these challenges, we propose extending the Lagrange multiplier framework from finite elements, as introduced by Babuška [1], to neural network-based PDE solvers. Our work develops a class of algorithms termed Deep Uzawa algorithms, which iteratively solve the resulting saddle point problems to weakly impose boundary conditions. The key innovation lies in adapting Uzawa's algorithm [22] to this context, allowing for efficient iterative approximation of PDEs where boundary conditions are enforced using an augmented Lagrangian formulation. This approach ensures that the algorithmic framework remains stable and accurate due to the coercivity of the energies involved.

The Deep Uzawa methods, Ritz-Uzawa (RitUz) and PINNs-Uzawa (PINNUz), extend existing deep Ritz and PINN frameworks with minimal modifications, making them highly practical for integration into current computational workflows. The theoretical analysis provided includes convergence proofs that demonstrate the iterative schemes' stability at the PDE level. These theoretical guarantees offer a robust foundation for the implementation of the Deep Uzawa algorithms and provide insight into their convergence behaviour. We compare the behaviour of these approaches to the vanilla methods and show that the Deep Uzawa approach achieves comparable or superior performance without relying on tuning penalty parameters.

Numerous methods have been explored for weakly imposing boundary conditions within ANN-based PDE solvers. The deep Ritz method [4] and PINNs [18] form the foundational basis for many current approaches, but both rely on penalty terms for boundary enforcement, which can make optimisation challenging, particularly when large penalties are needed. To address these shortcomings, [14] proposed an adaptation using Nitsche's method [17] from finite element analysis to weakly impose boundary conditions, mitigating conformity issues highlighted in [4]. Similarly, [23] compared traditional Ritz-Galerkin methods with ANN-based approaches, noting the implicit regularisation properties provided by neural networks. Other advancements, such as the penalty-free neural network strategy in [19], have targeted second-order boundary value problems in complex geometries. In high-dimensional settings, [11] explored deep learning approaches for elliptic PDEs with non-trivial boundary conditions, showcasing the flexibility of neural networks in handling such cases.

Our Deep Uzawa method builds on these developments by leveraging a consistent saddle point framework to address the boundary enforcement problem, offering a minor tweak computationally that provably enhances stability and accuracy. The application of Uzawa-type iterations in neural network contexts, as presented in [16], serves as a foundation for our iterative scheme. Our approach provides a structured way to balance the competing objectives of PDE accuracy and boundary condition enforcement, demonstrating improved stability in various numerical experiments, including problems on non-convex domains and high-dimensional geometries.

The rest of the paper is organized as follows: In Section 2, we introduce the notation and fundamental concepts related to Sobolev spaces, which form the basis for the functional framework of our analysis. Section 3 presents the development of a Deep Ritz-Uzawa method, including a proof of convergence in suitable Sobolev spaces. In Section 4, we extend this approach to the PINNs-Uzawa scheme, demonstrating convergence within an appropriate space for least-squares minimisation. The construction of neural network approximations and their integration within the Deep Uzawa framework are discussed in Section 5. Numerical results showcasing the effectiveness of our methods for boundary layer problems, those in complex geometries and high dimension are provided in Section 6.

2 Notation and problem setup

We will use a standard notation for Sobolev spaces [6]. For $\Omega \subseteq \mathbb{R}^d$, we denote by $\|\cdot\|_{\mathrm{L}^p(\Omega)}$ the $\mathrm{L}^2(\Omega)$-norm with associated inner product $\langle \cdot, \cdot \rangle_{\mathrm{L}^2(\Omega)}$. For $s \geq 0, p \in (1, \infty)$, we denote by $\|\cdot\|_{\mathrm{H}^p(\Omega)}$ ($|\cdot|_{\mathrm{H}^p(\Omega)}$) the norm (semi-norm) in the Hilbert space $\mathrm{H}^p(\Omega)$. We will now provide a short summary of fractional and negative Sobolev spaces and their associated norms and inner products.

2.1 Fractional and negative Sobolev spaces

For $m \in \mathbb{N}$, we define the fractional Sobolev space $\mathrm{H}^{m-1/2}(\partial\Omega)$ as the set of traces of functions in $\mathrm{H}^m(\Omega)$,

$$\mathrm{H}^{m-\frac{1}{2}}(\partial\Omega) := \{g : \partial\Omega \to \mathbb{R} : u = g \text{ a.e. on } \partial\Omega,\ \forall\, u \in \mathrm{H}^m(\Omega)\}. \tag{2.1}$$

In general, fractional Sobolev spaces are defined for all exponents and powers in terms of Gagliardo semi-norms. However, for the purposes of this paper, the above definition is sufficient as we consider the case when $m = 1, 2$, which are particularly relevant for analysis of the methods discussed.

For $s > 0$, the negative Sobolev space $\mathrm{H}^{-s}(\partial\Omega)$ is defined as the dual of the space $\mathrm{H}^s(\partial\Omega)$ and has an associated norm given by

$$\|G\|_{\mathrm{H}^{-s}(\partial\Omega)} := \sup_{v \in \mathrm{H}^s(\partial\Omega)} \frac{\langle G \mid v \rangle_{\mathrm{H}^{-s}(\partial\Omega) \times \mathrm{H}^s(\partial\Omega)}}{\|v\|_{\mathrm{H}^s(\partial\Omega)}}, \tag{2.2}$$

where $\langle G \mid v \rangle_{\mathrm{H}^{-s}(\partial\Omega) \times \mathrm{H}^s(\partial\Omega)}$ represents the duality pairing, i.e. the action of the linear functional G applied to the function v.

To facilitate later analysis, we recall the following trace theorem, which will be used to relate functions defined on Ω to their behaviour on $\partial\Omega$.

Theorem 2.1 ([7, Theorem 1]). *Let $s > 1/2$. Then, for $v \in \mathrm{H}^s(\Omega)$, there exists a constant $C_{\mathrm{tr}} > 0$ such that*

$$\|v\|_{\mathrm{L}^2(\partial\Omega)} \leq C_{\mathrm{tr}} \|v\|_{\mathrm{H}^s(\Omega)}. \tag{2.3}$$

Remark 2.1 (Computability of C_{tr})**.** The trace constant C_{tr} can, in many cases, be estimated, particularly for standard geometries such as cubes and spheres [3]. For a domain Ω that is star-shaped with respect to a point $\boldsymbol{x}_0 \in \Omega$ and satisfies $(\boldsymbol{x} - \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \partial\Omega$. Then we have that the following inequality holds:

$$C_{\mathrm{tr}} \le \frac{1}{2 \min_{\boldsymbol{x} \in \partial\Omega} (\boldsymbol{x} - \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x})} \left(d + \sqrt{d^2 + 4 \max_{\boldsymbol{x} \in \partial\Omega} |\boldsymbol{x} - \boldsymbol{x}_0|^2} \right). \tag{2.4}$$

3 Dirichlet energy minimisation

In this section, we introduce the model problem and provide some background on the Lagrange multiplier method, highlighting its connection to the Uzawa algorithm when the Dirichlet energy defines the loss function as in Ritz-based neural network methods.

To that end, we consider a self-adjoint elliptic problem with a strictly positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $f \in \mathrm{L}^2(\Omega)$, and $g \in \mathrm{H}^{1/2}(\partial\Omega)$. To set up the problem, we define the function space

$$\mathrm{H}^1_g(\Omega) := \{\phi \in \mathrm{H}^1(\Omega) \,:\, \phi|_{\partial\Omega} = g\}. \tag{3.1}$$

We then seek a solution $u \in \mathrm{H}^1_g(\Omega)$ such that

$$\mathscr{L}u := -\mathrm{div}\,(\boldsymbol{A}\nabla u) + u = f \quad \text{in } \Omega, \tag{3.2}$$

in the weak sense, that is u satisfies

$$\langle \boldsymbol{A}\nabla u, \nabla v\rangle_{\mathrm{L}^2(\Omega)} + \langle u, v\rangle_{\mathrm{L}^2(\Omega)} = \langle f, v\rangle_{\mathrm{L}^2(\Omega)}, \quad \forall\, v \in \mathrm{H}^1_0(\Omega). \tag{3.3}$$

We observe that Eq. (3.2) is the Euler-Lagrange equation corresponding to the minimisation of the convex quadratic Dirichlet functional

$$J_D(u) := \frac{1}{2}\|\boldsymbol{A}^{\frac{1}{2}}\nabla u\|^2_{\mathrm{L}^2(\Omega)} + \frac{1}{2}\|u\|^2_{\mathrm{L}^2(\Omega)} - \langle f, u\rangle_{\mathrm{L}^2(\Omega)}. \tag{3.4}$$

Furthermore, the weak solution to (3.3) also minimises the Dirichlet energy, that is

$$u = \underset{\phi \in \mathrm{H}^1_g(\Omega)}{\arg\min} J_D(\phi). \tag{3.5}$$

This variational formulation serves as the foundation for deep Ritz neural network methods, where neural networks are utilised to approximate the minimiser of $J_D(u)$.

3.1 Penalty methods

In practice, it is often beneficial to pose the minimisation problem over a larger space, such as $\mathrm{H}^1(\Omega)$, and enforce the boundary condition as a constraint within the formulation. This can be accomplished by extending the definition of $J_D : \mathrm{H}^1(\Omega) \to \mathbb{R}$ through the addition of a penalty term

$$J_D(u) := \frac{1}{2}\|\boldsymbol{A}^{\frac{1}{2}}\nabla u\|^2_{\mathrm{L}^2(\Omega)} + \frac{1}{2}\|u\|^2_{\mathrm{L}^2(\Omega)} - \langle f, u\rangle_{\mathrm{L}^2(\Omega)} + \frac{\gamma}{2}\|u - g\|^2_X \tag{3.6}$$

for some $\gamma \geq 0$ and an appropriate normed space X. A natural choice from an analytic point of view for X is $\mathrm{H}^{1/2}(\partial\Omega)$.

Remark 3.1 (Finite Element Approaches)**.** In the context of finite element methods, the penalty term is often formulated using a mesh-weighted L^2 norm. Let h denote the finite element discretisation parameter then one can consider

$$J_{D,h}(u_h) := \frac{1}{2}\|\boldsymbol{A}^{\frac{1}{2}}\nabla u_h\|^2_{\mathrm{L}^2(\Omega)} + \frac{1}{2}\|u_h\|^2_{\mathrm{L}^2(\Omega)} - \langle f, u_h\rangle_{\mathrm{L}^2(\Omega)} + \frac{\gamma}{h}\|u_h - g\|^2_{\mathrm{L}^2(\partial\Omega)}. \tag{3.7}$$

It can then be shown that, by choosing γ sufficiently large, we have

$$\lim_{h\to 0}\|u - u_h\|_{\mathrm{H}^1(\Omega)} = 0. \tag{3.8}$$

This argument relies on applying an inverse estimate, however, such estimates are not available in the context of neural network approximations.

Definition 3.1 (Dirichlet Loss Functional)**.** *To maintain compatibility with neural network-based methodologies, we use* $X = \mathrm{L}^2(\partial\Omega)$*-penalty term, i.e.*

$$J_D(u) := \frac{1}{2}\|\boldsymbol{A}^{\frac{1}{2}}\nabla u\|^2_{\mathrm{L}^2(\Omega)} + \frac{1}{2}\|u\|^2_{\mathrm{L}^2(\Omega)} - \langle f, u\rangle_{\mathrm{L}^2(\Omega)} + \frac{\gamma}{2}\|u - g\|^2_{\mathrm{L}^2(\partial\Omega)}. \tag{3.9}$$

Remark 3.2 (Limitations of Penalty Only Approaches)**.** In neural network approaches, the penalty term in (3.9) is often insufficient to guarantee the solution well approximates both the PDE and the boundary condition. This is as the resulting Euler-Lagrange equations corresponding to (3.9) are

$$\begin{aligned} \mathscr{L}u - f &= 0 && \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{A}\nabla u + \gamma(u - g) &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{3.10}$$

Thus, the minimiser of J_D satisfies a Robin boundary condition with parameter γ, rather than a true Dirichlet boundary condition. This observation motivates the exploration of alternative methods, such as those involving Lagrange multipliers [1].

The first core idea of our approach is the introduction of the Lagrangian.

Definition 3.2 (Dirichlet Energy Lagrangian)**.** *For a given* $g \in \mathrm{H}^{1/2}(\partial\Omega)$ *and* $f \in \mathrm{L}^2(\Omega)$*, let the Lagrangian* $L_D : \mathrm{H}^1(\Omega) \times \mathrm{H}^{-1/2}(\partial\Omega) \to \mathbb{R}$ *be given by*

$$L_D(u, \lambda) := J_D(u) - \langle \lambda \mid u - g\rangle_{\mathrm{H}^{-1/2}(\partial\Omega)\times\mathrm{H}^{1/2}(\partial\Omega)}. \tag{3.11}$$

This Lagrangian formulation allows us to incorporate the boundary condition $u = g$ *weakly by introducing a Lagrange multiplier* λ*.*

We then seek the saddle points of the Lagrangian L_D, that satisfy $(u^*, \lambda^*) \in \mathrm{H}^1(\Omega) \times \mathrm{H}^{-1/2}(\partial\Omega)$ such that

$$(u^*, \lambda^*) = \operatorname*{arg\,min}_{u \in \mathrm{H}^1(\Omega)} \operatorname*{arg\,max}_{\lambda \in \mathrm{H}^{-1/2}(\partial\Omega)} L_D(u, \lambda). \tag{3.12}$$

This problem leads to the Euler-Lagrange equations

$$\begin{aligned} \mathscr{L}u^* - f &= 0 && \text{in } \Omega, \\ u^* - g &= 0 && \text{on } \partial\Omega, \\ -\lambda^* + \boldsymbol{n} \cdot \boldsymbol{A}\nabla u^* &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{3.13}$$

To solve the saddle point problem (3.12), we propose an iterative method through an Uzawa algorithm at the continuum level.

Definition 3.3 (Dirichlet Energy Update Scheme). *For a given initial guess $\lambda^0 \in \mathrm{H}^{-1/2}(\partial\Omega)$ and a step size $\rho > 0$, we define the sequence of functions $\{\lambda^k\}_{k=0}^{\infty} \subset \mathrm{H}^{-1/2}(\partial\Omega)$ and $\{u^k\}_{k=0}^{\infty} \subset \mathrm{H}^1(\Omega)$ by the following iterative scheme:*

$$u^k = \operatorname*{arg\,min}_{u \in \mathrm{H}^1(\Omega)} L_D(u, \lambda^k), \tag{3.14a}$$

$$\langle \lambda^{k+1} - \lambda^k \,|\, \phi \rangle_{\mathrm{H}^{-1/2}(\partial\Omega) \times \mathrm{H}^{1/2}(\partial\Omega)} = -\rho \langle u^k - g, \phi \rangle_{\mathrm{L}^2(\partial\Omega)}, \quad \forall \phi \in \mathrm{H}^{\frac{1}{2}}(\partial\Omega). \tag{3.14b}$$

Remark 3.3 (Simplifications in the Neural Network Framework). Our algorithm relies on approximating the Uzawa iterates (3.14) within an appropriate neural network framework. Due to the inherent smoothness of neural network functions, their traces belong to $L^2(\partial\Omega)$. Similarly, the discrete Lagrange multipliers λ_ℓ^k are also functions in $L^2(\partial\Omega)$. As a result, duality pairings simplify to basic $L^2(\partial\Omega)$ integrals

$$\langle \lambda_\ell \,|\, u_\ell - g \rangle_{\mathrm{H}^{-1/2}(\partial\Omega) \times \mathrm{H}^{1/2}(\partial\Omega)} = \int_{\partial\Omega} (u_\ell - g)\, \lambda_\ell \, ds. \tag{3.15}$$

Additionally, updating the Lagrange multiplier in the Eq. (3.14b) simplifies to a straightforward function evaluation on $\partial\Omega$. For more details, see Section 5.3. These observations result in a particularly simple and efficient implementation.

Theorem 3.1. *Let the sequences $\{u^k\}$ and $\{\lambda^k\}$ denote the Uzawa iterates given in Definition 3.3, and (u^*, λ^*) denote the saddle point of (3.12). Let $\mathrm{C_{tr}} > 0$ is the trace constant associated with Ω and assume that $\boldsymbol{A}$ is a strictly positive definite matrix with the smallest eigenvalue $\sigma_{\min} > 0$. Suppose further that the parameters $\gamma \geq 0$ and $\rho > 0$ satisfy*

$$\rho - 2\gamma < \frac{2 \min\{\sigma_{\min}, 1\}}{\mathrm{C}_{\mathrm{tr}}^2}. \tag{3.16}$$

Then we have

$$u^k \;\to\; u^* \quad \textit{in } \mathrm{H}^1(\Omega) \quad \textit{as} \quad k \;\to\; \infty. \tag{3.17}$$

Remark 3.4 (Convergence Regimes)**.** There are two regimes in which the inequality in Eq. (3.16) holds. The first is when γ is large relative to ρ, corresponding to $2\gamma \geq \rho$. In this regime, the inequality is trivially satisfied, and the bound does not depend on the matrix $\boldsymbol{A}$. The second is when γ is small relative to ρ, including the case when $\gamma = 0$, corresponding to $2\gamma < \rho$. Here, the Uzawa update step-size ρ must be sufficiently small. These regimes illustrate how the balance between γ and ρ impacts convergence.

To prove Theorem 3.1, we will prove or state a series of definitions and technical lemmata that will be used in the proof. To begin, we recall the classical Riesz representation theorem and some of its consequences, which play a important role in the convergence analysis.

Theorem 3.2 (Riesz Representation)**.** *For every $s \in \mathbb{N}$ and $z \in \mathrm{H}^{-s/2}(\partial\Omega)$, there exists a unique element $R_s[z] \in \mathrm{H}^{s/2}(\partial\Omega)$, known as the Riesz representor of z such that for all $\phi \in \mathrm{H}^{s/2}(\partial\Omega)$, the following holds:*

$$\langle R_s[z], \phi\rangle_{\mathrm{H}^{s/2}(\partial\Omega)} := \langle z \,|\, \phi\rangle_{\mathrm{H}^{-s/2}(\partial\Omega)\times\mathrm{H}^{s/2}(\partial\Omega)}, \quad \forall \phi \in \mathrm{H}^{\frac{s}{2}}(\partial\Omega). \tag{3.18}$$

Moreover, the mapping $R_s : \mathrm{H}^{-s/2}(\partial\Omega) \to \mathrm{H}^{s/2}(\partial\Omega)$ is an isometric isomorphism, i.e. it preserves the norm structure

$$\|R_s[z]\|_{\mathrm{H}^{s/2}(\partial\Omega)} = \|z\|_{\mathrm{H}^{-s/2}(\partial\Omega)}. \tag{3.19}$$

Furthermore, as a consequence of this definition, for any $z \in \mathrm{H}^{-s/2}(\partial\Omega)$, we have

$$\|R_s[z]\|^2_{\mathrm{H}^{s/2}(\partial\Omega)} = \langle z \,|\, R_s[z]\rangle_{\mathrm{H}^{-s/2}(\partial\Omega)\times\mathrm{H}^{s/2}(\partial\Omega)}. \tag{3.20}$$

To prove Theorem 3.1, we first establish the following lemma, which provides bounds on the Uzawa iterates u^k and λ^k in relation to the saddle points u^* and λ^*.

Lemma 3.1 (Bounds on the Dirichlet Lagrangian)**.** *Let the sequences $\{u^k\}, \{\lambda^k\}$ denote the Uzawa iterates given in Definition* 3.3 *and u^*, λ^* denote the saddle points of L_D* (3.12)*. Then we have*

$$\begin{aligned} &\| \boldsymbol{A}^{\frac{1}{2}}\nabla(u^k - u^*)\|^2_{\mathrm{L}^2(\Omega)} + \|u^k - u^*\|^2_{\mathrm{L}^2(\Omega)} + \gamma\|u^k - u^*\|^2_{\mathrm{L}^2(\partial\Omega)} \\ &= \langle \lambda^k - \lambda^* \,|\, u^k - u^*\rangle_{\mathrm{H}^{-1/2}(\partial\Omega)\times\mathrm{H}^{1/2}(\partial\Omega)}. \end{aligned} \tag{3.21}$$

Proof. Since u^* and λ^* are the saddle points of the Lagrangian L_D and λ^k, u^k satisfy Definition 3.3, we infer the first-order optimality conditions

$$\begin{aligned} &\langle \boldsymbol{A}\nabla u^*, \nabla\phi\rangle_{\mathrm{L}^2(\Omega)} + \langle u^*, \phi\rangle_{\mathrm{L}^2(\Omega)} - \langle f, \phi\rangle_{\mathrm{L}^2(\Omega)} \\ &\quad + \gamma\langle u^* - g, \phi\rangle_{\mathrm{L}^2(\partial\Omega)} - \langle \lambda^* \,|\, \phi\rangle_{\mathrm{H}^{-1/2}(\partial\Omega)\times\mathrm{H}^{1/2}(\partial\Omega)} = 0, \\ &\langle \boldsymbol{A}\nabla u^k, \nabla\phi\rangle_{\mathrm{L}^2(\Omega)} + \langle u^k, \phi\rangle_{\mathrm{L}^2(\Omega)} - \langle f, \phi\rangle_{\mathrm{L}^2(\Omega)} \\ &\quad + \gamma\langle u^k - g, \phi\rangle_{\mathrm{L}^2(\partial\Omega)} - \langle \lambda^k \,|\, \phi\rangle_{\mathrm{H}^{-1/2}(\partial\Omega)\times\mathrm{H}^{1/2}(\partial\Omega)} = 0 \end{aligned} \tag{3.22}$$

for all $\phi \in \mathrm{H}^1(\Omega)$.

Taking the difference of the above expressions and setting $\phi = u^k - u^*$ gives the following error equation:

$$\begin{aligned} &\|A^{\frac{1}{2}}\nabla(u^k - u^*)\|^2_{L^2(\Omega)} + \|u^k - u^*\|^2_{L^2(\Omega)} + \gamma\|u^k - u^*\|^2_{L^2(\partial\Omega)} \\ &= \langle \lambda^k - \lambda^* \mid u^k - u^* \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)}, \end{aligned} \tag{3.23}$$

concluding the proof. □

Lemma 3.2. *Let* $\{u^k\}, \{\lambda^k\}$ *be given by* (3.14)*, and* u^*, λ^* *denote the saddle points of* (3.12)*. Then*

$$\begin{aligned} &\|\lambda^{k+1} - \lambda^*\|^2_{H^{-1/2}(\partial\Omega)} \\ &= \|\lambda^k - \lambda^*\|^2_{H^{-1/2}(\partial\Omega)} + \rho^2\|u^k - u^*\|^2_{L^2(\partial\Omega)} \\ &\quad - 2\rho\langle \lambda^k - \lambda^* \mid u^k - u^* \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)}. \end{aligned} \tag{3.24}$$

Proof. We begin by recalling the first-order optimality conditions for the saddle points and iterates of L_D. By these conditions, we have

$$\begin{aligned} &\langle \lambda^* \mid \psi \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)} \\ &= \langle \lambda^* \mid \psi \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)} \\ &\quad - \rho\langle u^* - g, \psi \rangle_{L^2(\partial\Omega)}, \quad \forall\, \psi \in H^{\frac{1}{2}}(\partial\Omega). \end{aligned} \tag{3.25}$$

Taking the difference between Eq. (3.25) and the update condition in (3.14), we obtain

$$\begin{aligned} &\langle \lambda^{k+1} - \lambda^* \mid \psi \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)} \\ &= \langle \lambda^k - \lambda^* \mid \psi \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)} \\ &\quad - \rho\langle u^k - u^*, \psi \rangle_{L^2(\partial\Omega)}, \quad \forall\, \psi \in H^{\frac{1}{2}}(\partial\Omega). \end{aligned} \tag{3.26}$$

Now, let $\psi = R_1[\lambda^{k+1} - \lambda^*]$. Since R_1 is an isometric isomorphism, we have

$$\begin{aligned} &\|\lambda^{k+1} - \lambda^*\|^2_{H^{-1/2}(\partial\Omega)} \\ &= \langle \lambda^k - \lambda^* \mid R_1[\lambda^{k+1} - \lambda^*] \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)} \\ &\quad - \rho\langle u^k - u^*, R_1[\lambda^{k+1} - \lambda^*] \rangle_{L^2(\partial\Omega)}. \end{aligned} \tag{3.27}$$

The duality pairing is symmetric with respect to the Riesz representor

$$\begin{aligned} &\langle w \mid R_1[z] \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)} \\ &= \langle z \mid R_1[w] \rangle_{H^{-1/2}(\partial\Omega)\times H^{1/2}(\partial\Omega)}, \quad \forall\, w, z \in H^{-\frac{1}{2}}(\partial\Omega). \end{aligned} \tag{3.28}$$

Taking $\psi = R_1[\lambda^k - \lambda^*]$ in Eq. (3.26) and substituting into (3.27) gives

$$\begin{aligned}&\|\lambda^{k+1} - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)}\\ &= \|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{1/2}(\partial\Omega)} - \rho\langle u^k - u^*, R_1[\lambda^k - \lambda^*]\rangle_{\mathrm{L}^2(\partial\Omega)}\\ &\quad - \rho\langle u^k - u^*, R_1[\lambda^{k+1} - \lambda^*]\rangle_{\mathrm{L}^2(\partial\Omega)}.\end{aligned} \tag{3.29}$$

Finally, taking $\psi = u^k - u^*$ in Eq. (3.26) and substituting into (3.29), we have

$$\begin{aligned}&\|\lambda^{k+1} - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)}\\ &= \|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{1/2}(\partial\Omega)} - 2\rho\langle u^k - u^*, R_1[\lambda^k - \lambda^*]\rangle_{\mathrm{L}^2(\partial\Omega)}\\ &\quad + \rho^2\|u^k - u^*\|^2_{\mathrm{L}^2(\partial\Omega)},\end{aligned} \tag{3.30}$$

completing the proof. □

3.2 Proof of Theorem 3.1

Proof. Let $e^k := u^k - u^*$ and $\beta := 2\gamma - \rho$. By applying Lemma 3.1 to Lemma 3.2, we obtain

$$\begin{aligned}&\|\lambda^{k+1} - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)}\\ &= \|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - 2\rho\left\|\boldsymbol{A}^{\frac{1}{2}}\nabla e^k\right\|^2_{\mathrm{L}^2(\Omega)}\\ &\quad - 2\rho\|e^k\|^2_{\mathrm{L}^2(\Omega)} - \rho\beta\|e^k\|^2_{\mathrm{L}^2(\partial\Omega)}.\end{aligned} \tag{3.31}$$

Without loss of generality, we assume $e^k \neq 0$.

Case 1. $\beta \geq 0$ **(Non-Negative Penalty Term).** If $\beta \geq 0$, then

$$\begin{aligned}&\|\lambda^{k+1} - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)}\\ &\leq \|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - 2\rho\left\|\boldsymbol{A}^{\frac{1}{2}}\nabla e^k\right\|^2_{\mathrm{L}^2(\Omega)} - 2\rho\|e^k\|^2_{\mathrm{L}^2(\Omega)}\\ &\leq \|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - 2\rho\min\{\sigma_{\min}, 1\}\|e^k\|^2_{\mathrm{H}^1(\Omega)}.\end{aligned} \tag{3.32}$$

Assuming $\rho > 0$ and using the positivity of $\vec{A}$, we conclude that $\|\lambda^k - \lambda^*\|_{\mathrm{H}^{-1/2}(\partial\Omega)}$ is a strictly decreasing sequence. Thus,

$$0 \leq \lim_{k\to\infty}\|e^k\|^2_{\mathrm{H}^1(\Omega)} \leq \lim_{k\to\infty}\frac{\|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - \|\lambda^{k+1} - \lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)}}{2\rho\min\{\sigma_{\min}, 1\}} = 0. \tag{3.33}$$

Case 2. $\beta < 0$ **(Negative Penalty Term).** If $\beta < 0$, then by Theorem 2.1

$$\begin{aligned}
&\|\lambda^{k+1}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} \\
&= \|\lambda^{k}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - 2\rho\|\boldsymbol{A}^{\frac{1}{2}}\nabla e^k\|^2_{\mathrm{L}^2(\Omega)} \\
&\quad - 2\rho\|e^k\|^2_{\mathrm{L}^2(\Omega)} + \rho|\beta|\|e^k\|^2_{\mathrm{L}^2(\partial\Omega)} \\
&\le \|\lambda^{k}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - 2\rho\|\boldsymbol{A}^{\frac{1}{2}}\nabla e^k\|^2_{\mathrm{L}^2(\Omega)} \\
&\quad - 2\rho\|e^k\|^2_{\mathrm{L}^2(\Omega)} + \rho|\beta|C_{\mathrm{tr}}^2\|e^k\|^2_{\mathrm{H}^1(\Omega)}.
\end{aligned} \tag{3.34}$$

Given that $\boldsymbol{A}$ is positive definite with the smallest eigenvalue $\sigma_{\min} > 0$, we obtain

$$\begin{aligned}
&\|\lambda^{k+1}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} \\
&\le \|\lambda^{k}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - 2\rho\sigma_{\min}\|\nabla e^k\|^2_{\mathrm{L}^2(\Omega)} \\
&\quad - 2\rho\|e^k\|^2_{\mathrm{L}^2(\Omega)} + \rho|\beta|C_{\mathrm{tr}}^2\|e^k\|^2_{\mathrm{H}^1(\Omega)} \\
&\le \|\lambda^{k}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - \alpha\|e^k\|^2_{\mathrm{H}^1(\Omega)}, \\
&\quad \alpha := \rho\big(2\min\{\sigma_{\min},1\} - |\beta|C_{\mathrm{tr}}^2\big).
\end{aligned} \tag{3.35}$$

By Eq. (3.16), $\|\lambda^k-\lambda^*\|_{\mathrm{H}^{-1/2}(\partial\Omega)}$ is a strictly decreasing sequence. Hence,

$$\lim_{k\to\infty}\|e^k\|^2_{\mathrm{H}^1(\Omega)} \le \frac{1}{\alpha}\lim_{k\to\infty}\|\lambda^k-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} - \|\lambda^{k+1}-\lambda^*\|^2_{\mathrm{H}^{-1/2}(\partial\Omega)} \ \to\ 0, \tag{3.36}$$

completing the proof of Theorem 3.1. □

4 Least squares minimisation

In the previous section, we developed an iterative scheme based on the Euler-Lagrange equation of the cost functional J_D. However, it is important to note that J_D is not the only choice of cost functional. An alternative approach is to consider a least squares minimisation of the residual of the PDE. This method forms the foundation of many physics-informed neural network algorithms.

Let us assume Ω is a convex domain and consider a more regular boundary function $g \in \mathrm{H}^{3/2}(\partial\Omega)$, defining the cost functional

$$J_R(u) := \frac{1}{2}\|\mathscr{L}u - f\|^2_{\mathrm{L}^2(\Omega)} \tag{4.1}$$

over the constrained space

$$\mathrm{H}^2_g(\Omega) := \{u \in \mathrm{H}^2(\Omega) : u|_{\partial\Omega} = g\}. \tag{4.2}$$

This space ensures that the solution adheres to the boundary conditions and is suitable for PINN implementations. However, similar to the Dirichlet energy minimisation approach often it is practical to extend the functional to include a penalty term for $\gamma \geq 0$,

$$J_R(u) := \frac{1}{2}\|\mathscr{L}u - f\|^2_{\mathrm{L}^2(\Omega)} + \frac{\gamma}{2}\|u - g\|^2_{\mathrm{L}^2(\partial\Omega)}. \tag{4.3}$$

A drawback of this approach is that the boundary penalty $\|u - g\|^2_{\mathrm{L}^2(\Omega)}$ is weak, making the resulting loss $J_R(u)$ unbalanced. Consequently, boundary errors often dominate PINN algorithm inaccuracies, especially in singularly perturbed problems. To address this, we consider the critical points of an equivalent functional over $\mathrm{H}^2(\Omega)$.

Remark 4.1 (Non-Convex Domains)**.** The analysis presented here assumes Ω is convex to leverage full elliptic regularity of the critical points. Extending this analysis to non-convex domains introduces geometric singularities, which can be addressed by considering the space

$$\mathrm{H}^{\mathscr{L}}_g(\Omega) := \{u \in \mathrm{H}^1_g(\Omega) : \mathscr{L}u \in \mathrm{L}^2(\Omega)\}. \tag{4.4}$$

However, to avoid excessive notation and complexity, we do not present this analysis here. For illustration, we demonstrate the method applied to a non-convex domain in Section 6.2.

Definition 4.1 (PINNs Energy Lagrangian)**.** *For a given $g \in \mathrm{H}^{3/2}(\partial\Omega)$ and $f \in \mathrm{L}^2(\Omega)$, let the Lagrangian $L_R : \mathrm{H}^2(\Omega) \times \mathrm{H}^{-3/2}(\partial\Omega) \to \mathbb{R}$ be defined by*

$$L_R(u, \lambda) := J_R(u) - \langle \lambda \,|\, u - g\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)}. \tag{4.5}$$

We seek the saddle points of the Lagrangian L_R, denoted by $(u^*, \lambda^*) \in \mathrm{H}^2(\Omega) \times \mathrm{H}^{-3/2}(\partial\Omega)$ such that

$$(u^*, \lambda^*) = \operatorname*{arg\,min}_{u\in\mathrm{H}^2(\Omega)} \operatorname*{arg\,max}_{\lambda\in\mathrm{H}^{-3/2}(\partial\Omega)} L_R(u, \lambda). \tag{4.6}$$

This can be achieved through an Uzawa iteration scheme.

Definition 4.2 (PINNs Energy Update Scheme)**.** *For a given initial guess $\lambda^0 \in \mathrm{H}^{-3/2}(\partial\Omega)$, a step size $\rho > 0$ and a penalty parameter $\gamma > 0$, we define the sequences $\{\lambda^k\}_{k=1}^\infty \subset \mathrm{H}^{-3/2}(\partial\Omega)$ and $\{u^k\}_{k=1}^\infty \subset \mathrm{H}^2(\Omega)$ by the following iterative scheme:*

$$u^k = \operatorname*{arg\,min}_{u\in\mathrm{H}^2(\Omega)} L_R(u, \lambda^k), \tag{4.7a}$$

$$\langle \lambda^{k+1} - \lambda^k \,|\, \psi\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)} = -\rho\langle u^k - g, \psi\rangle_{\mathrm{L}^2(\partial\Omega)}, \quad \forall\, \psi \in \mathrm{H}^{\frac{3}{2}}(\partial\Omega). \tag{4.7b}$$

As in the case of the Dirichlet energy approach, our algorithm relies on approximating (4.7) within an appropriate neural network framework. Due to the inherent smoothness of neural network functions, the duality pairings again simplify to $L^2(\partial\Omega)$ integrals

$$\langle \lambda_\ell \,|\, u_\ell - g\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)} = \int_{\partial\Omega} (u_\ell - g)\, \lambda_\ell \, dS. \tag{4.8}$$

Furthermore, updating the Lagrange multiplier in the Eq. (4.7b) simplifies to a function evaluation on $\partial\Omega$. For more details, see Section 5.2.

To distinguish this method from the previous section, we shall refer to the Deep Ritz Uzawa scheme as RitUz and the PINNs Uzawa scheme as PINNUz. This terminology helps to clearly identify the different iterative schemes discussed. From this scheme, we can demonstrate the convergence of the sequence u^k in an appropriate norm.

Theorem 4.1. *Let the sequence u^k, λ^k denote the Uzawa iterates from Definition 4.2, and let u^*, λ^* denote the saddle points (4.6). Assume that $\boldsymbol{A}$ is a strictly positive definite matrix, and that the Uzawa constant $\rho > 0$ and the stabilisation constant γ satisfy the following inequality:*

$$2\gamma - \rho > 2. \tag{4.9}$$

Then, for all $s \in [0, 1/2)$, we have

$$u^k \quad \to \quad u^* \quad \textit{in } \mathrm{H}^s(\Omega) \textit{ and } \mathrm{L}^2(\partial\Omega) \quad \textit{as} \quad k \quad \to \quad \infty. \tag{4.10}$$

Remark 4.2 (Weaker Convergence for Least Squares)**.** Comparing Theorems 3.1 and 4.1, we observe several key differences.

Firstly, in the PINNUz scheme, u^k converges in H^s for $s \in [0, 1/2)$, whereas in the RitUz scheme, u^k converges in H^1. This difference arises because the Dirichlet energy functional is coercive over all of $\mathrm{H}^1(\Omega)$, which facilitates stronger convergence properties in the RitUz scheme. On the other hand, the least squares energy functional used in the PINNUz scheme is not coercive over $\mathrm{H}^2(\Omega)$; instead, it only exhibits coercivity over a weaker space. This reduced coercivity leads to the need for weaker regularity conditions for convergence in the PINNUz scheme.

Secondly, the inequality in (4.9) does not depend on the matrix $\boldsymbol{A}$, unlike the corresponding inequality in the RitUz scheme (3.16). This difference can be understood by recalling the two regimes in the RitUz scheme: when $\beta \leq 0$ and when $\beta > 0$. Dependence on the matrix $\boldsymbol{A}$ was only required in the proof for the case $\beta \leq 0$. In contrast, the PINNUz scheme operates solely in a regime where $\beta > 0$, corresponding to sufficiently large γ. Specifically, γ must be strictly greater than 1 in the PINNUz scheme, whereas the RitUz scheme could accommodate cases where $\gamma = 0$.

Similar to the proof of Theorem 3.1, we will begin by stating a series of lemmata that will be used in the proof. Some of these lemmata are analogous to those in the RitUz scheme but are adapted to highlight the main differences in the PINNUz scheme.

Lemma 4.1 (Bounds on the PINNs Lagrangian)**.** *Let u^k, λ^k be as in Definition 4.2, and u^*, λ^* be as in (4.6). Then*

$$\|\mathscr{L}(u^k - u^*)\|^2_{\mathrm{L}^2(\Omega)} + \gamma \|u^k - u^*\|^2_{\mathrm{L}^2(\partial\Omega)} = \langle \lambda^k - \lambda^* \mid u^k - u^* \rangle_{\mathrm{H}^{-3/2}(\partial\Omega) \times \mathrm{H}^{3/2}(\partial\Omega)}. \tag{4.11}$$

Proof. Similar to the proof of Lemma 3.1, we compute the Euler-Lagrange equations for u^* and u^k to deduce

$$\begin{aligned} \langle \mathscr{L}u^* - f, \mathscr{L}\phi \rangle_{\mathrm{L}^2(\Omega)} + \gamma \langle u^* - g, \phi \rangle_{\mathrm{L}^2(\partial\Omega)} &= \langle \lambda^* \mid \phi \rangle_{\mathrm{H}^{-3/2}(\partial\Omega) \times \mathrm{H}^{3/2}(\partial\Omega)}, \\ \langle \mathscr{L}u^k - f, \mathscr{L}\phi \rangle_{\mathrm{L}^2(\Omega)} + \gamma \langle u^k - g, \phi \rangle_{\mathrm{L}^2(\partial\Omega)} &= \langle \lambda^k \mid \phi \rangle_{\mathrm{H}^{-3/2}(\partial\Omega) \times \mathrm{H}^{3/2}(\partial\Omega)}. \end{aligned} \tag{4.12}$$

Considering the difference between these two equalities and setting $\phi = u^k - u^*$ yields the desired result. □

Lemma 4.2. *Let $\{u^k, \lambda^k\}$ be as given in* (4.7)*, and $\{u^*, \lambda^*\}$ be as given in* (4.6)*. Then*

$$\begin{aligned}&\|\lambda^{k+1} - \lambda^*\|^2_{\mathrm{H}^{-3/2}(\partial\Omega)}\\ &= \|\lambda^k - \lambda^*\|^2_{\mathrm{H}^{-3/2}(\partial\Omega)} - 2\rho\langle\lambda^k - \lambda^* \,|\, u^k - u^*\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)}\\ &\quad + \rho^2\|u^k - u^*\|^2_{\mathrm{L}^2(\partial\Omega)}.\end{aligned} \tag{4.13}$$

Proof. The proof follows a similar structure to Lemma 3.2, but instead utilises the representor $R_3 : \mathrm{H}^{-3/2}(\partial\Omega) \to \mathrm{H}^{3/2}(\partial\Omega)$ in place of R_1. □

Lemma 4.3. *Let $\{u^k, \lambda^k\}$ be given by* (4.7)*, and $\{u^*, \lambda^*\}$ be given by* (4.6)*. Then the sequence of functions defined by $e^k := u^k - u^*$ weakly satisfies the PDE*

$$\mathscr{L}e^k = 0 \quad \text{on } \Omega, \quad \forall\, k > 0, \tag{4.14}$$

subject to non-trivial boundary conditions. Additionally, for all $s \in [0, 1/2)$, there exists a constant $C_{\mathrm{reg}} > 0$ dependent on $\mathscr{L}$ such that

$$\|e^k\|_{\mathrm{H}^s(\Omega)} \le C_{\mathrm{reg}}\big(\|e^k\|_{\mathrm{L}^2(\partial\Omega)} + \|\mathscr{L}e^k\|_{\mathrm{L}^2(\Omega)}\big). \tag{4.15}$$

Proof. The initial analysis is similar to that found in [8]. The functions u^k and u^* satisfy the following Euler-Lagrange equations:

$$\begin{aligned}&\langle\mathscr{L}u^* - f, \mathscr{L}\phi\rangle_{\mathrm{L}^2(\Omega)} + \gamma\langle u^* - g, \phi\rangle_{\mathrm{L}^2(\partial\Omega)} = \langle\lambda^* \,|\, \phi\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)},\\ &\langle\mathscr{L}u^k - f, \mathscr{L}\phi\rangle_{\mathrm{L}^2(\Omega)} + \gamma\langle u^k - g, \phi\rangle_{\mathrm{L}^2(\partial\Omega)} = \langle\lambda^k \,|\, \phi\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)}\end{aligned} \tag{4.16}$$

for all $\phi \in \mathrm{H}^2 \cap \mathrm{H}^1_0(\Omega)$. Thus, for $e^k := u^k - u^*$, we have

$$\langle\mathscr{L}e^k, \mathscr{L}\phi\rangle_{\mathrm{L}^2(\Omega)} + \gamma\langle e^k, \phi\rangle_{\mathrm{L}^2(\partial\Omega)} = \langle\lambda^k - \lambda^* \,|\, \phi\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)}. \tag{4.17}$$

Let $w \in \mathrm{L}^2(\Omega)$ be arbitrary, and consider ϕ as the solution of $\mathscr{L}\phi = w$ with zero boundary conditions. Then

$$\begin{aligned}&\langle\mathscr{L}e^k, \mathscr{L}\phi\rangle_{\mathrm{L}^2(\Omega)} + \gamma\langle e^k, \phi\rangle_{\mathrm{L}^2(\partial\Omega)} - \langle\lambda^k - \lambda^* \,|\, \phi\rangle_{\mathrm{H}^{-3/2}(\partial\Omega)\times\mathrm{H}^{3/2}(\partial\Omega)}\\ &= \langle\mathscr{L}e^k, w\rangle_{\mathrm{L}^2(\Omega)} = 0, \quad \forall\, w \in \mathrm{L}^2(\Omega).\end{aligned} \tag{4.18}$$

Thus, e^k weakly satisfies $\mathscr{L}e^k = 0$ on Ω. Using the elliptic regularity estimate from [2], we deduce the desired result. □

4.1 Proof of Theorem 4.1

To prove Theorem 4.1, we use the lemmata developed earlier and a similar approach to the proof of Theorem 3.1. With $\beta := 2\gamma - \rho$ and $e^k := u^k - u^*$, by Lemmas 4.1 and 4.2, we have

$$\begin{aligned}&\|\lambda^{k+1}-\lambda^*\|^2_{\mathrm{H}^{-3/2}(\partial\Omega)}\\&=\|\lambda^k-\lambda^*\|^2_{\mathrm{H}^{-3/2}(\partial\Omega)}-\rho\beta\|e^k\|^2_{\mathrm{L}^2(\partial\Omega)}-2\rho\|\mathscr{L}e^k\|^2_{\mathrm{L}^2(\Omega)}.\end{aligned}\tag{4.19}$$

Applying Lemma 4.3 along with Young's inequality, we obtain

$$\begin{aligned}&\|\lambda^{k+1}-\lambda^*\|^2_{\mathrm{H}^{-3/2}(\partial\Omega)}-\|\lambda^k-\lambda^*\|^2_{\mathrm{H}^{-3/2}(\partial\Omega)}\\&\le-\rho(\beta-2)\|e^k\|^2_{\mathrm{L}^2(\partial\Omega)}-\frac{\rho}{C^2_{\mathrm{reg}}}\|e^k\|^2_{\mathrm{H}^s(\Omega)}.\end{aligned}\tag{4.20}$$

Without loss of generality, assuming $e^k\neq 0$, we see that $\|\lambda^k-\lambda^*\|_{\mathrm{H}^{-3/2}(\partial\Omega)}$ is a strictly decreasing sequence. By using (4.9) and the assumption that $\rho>0$, the rest of the proof follows similarly to the proof of Theorem 3.1.

Remark 4.3 (Impact of Stronger Penalties on Convergence)**.** By increasing the strength of the boundary penalty term to higher-order Sobolev norms, such as $\mathrm{H}^1(\partial\Omega)$, it is possible to establish convergence of u^k in stronger norms, such as $\mathrm{H}^{3/2}(\Omega)$. Implementing such an $\mathrm{H}^1(\Omega)$ penalty can be practically realised as it requires incorporating tangential derivatives, which can be efficiently computed and accounted for in many neural network frameworks.

5 Neural networks and Deep Uzawa

This section outlines the methodology for constructing neural network approximation schemes and their integration within the Deep Uzawa framework, illustrated through examples.

5.1 Neural network function approximation

Consider functions u_θ approximated by neural networks. A deep neural network maps each point $x\in\Omega$ to a value $w_\theta(x)\in\mathbb{R}$ through the process

$$w_\theta(x):=\mathcal{C}_L\circ\sigma\circ\mathcal{C}_{L-1}\circ\cdots\circ\sigma\circ\mathcal{C}_1(x),\tag{5.1}$$

where $\mathcal{C}_k$ represents affine transformations defined by

$$\mathcal{C}_k y=W_k y+b_k,\quad W_k\in\mathbb{R}^{d_{k+1}\times d_k},\quad b_k\in\mathbb{R}^{d_{k+1}}.\tag{5.2}$$

An illustration of this network structure may be seen in Fig. 5.1.

The parameters $\theta=\{W_k,b_k\}_{k=1}^L$ collectively define the network $\mathcal{C}_L$. The set of all such networks is denoted by $\mathcal{N}$, with the space of functions represented by

$$\mathcal{V}_N=\{u_\theta:\Omega\to\mathbb{R}\mid u_\theta(x)=\mathcal{C}_L(x)\text{ for some }\mathcal{C}_L\in\mathcal{N}\}.\tag{5.3}$$

Note that $\mathcal{V}_N$ is not a linear space, although $\Theta=\{\theta\mid u_\theta\in\mathcal{V}_N\}$ is a linear subspace of $\mathbb{R}^{\dim\mathcal{N}}$.

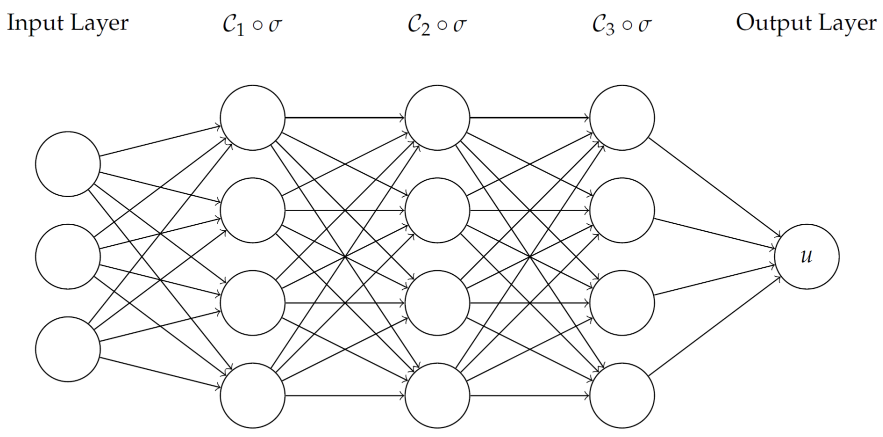
Input Layer
$\mathcal{C}_1 \circ \sigma$
$\mathcal{C}_2 \circ \sigma$
$\mathcal{C}_3 \circ \sigma$
Output Layer
u

Figure 5.1: Illustration of w_θ.

5.2 Training within the Deep Uzawa framework

To implement the Deep Uzawa iteration, we require discrete versions of the energy functionals L_D and L_R. These can be approximated using quadrature rules for integrals over Ω and $\partial\Omega$. Let $\mathcal{K}_h$ and $\partial\mathcal{K}_h$ represent sets of discrete points in Ω and on $\partial\Omega$, respectively, with weights w_y and w_b. We approximate the RitzUz functional as

$$\sum_{y \in \mathcal{K}_h} w_y g(y) \approx \int_\Omega g(x)\,dx, \qquad \sum_{b \in \partial\mathcal{K}_h} w_b g(b) \approx \int_{\partial\Omega} g(x)\,ds. \tag{5.4}$$

That is

$$\begin{aligned} L_D(u,\lambda) \approx \sum_{y \in \mathcal{K}_h} w_y \left(\frac{1}{2} \left| \boldsymbol{A}^{\frac{1}{2}} \nabla u \right|^2 (y) + \frac{1}{2} u^2(y) - u(y) f(y) \right) \\ + \sum_{b \in \partial\mathcal{K}_h} w_b \left(\frac{\gamma}{2} (u-g)^2(b) - \big(u(b) - g(b)\big) \lambda(b) \right). \end{aligned} \tag{5.5}$$

For the PINNUz iteration, we define the discrete functional similarly through

$$\begin{aligned} L_R(u,\lambda) \approx \frac{1}{2} \sum_{y \in \mathcal{K}_h} w_y |\mathscr{L} u - f|^2 (y) \\ + \sum_{b \in \partial\mathcal{K}_h} w_b \left(\frac{\gamma}{2} (u-g)^2(b) - \big(u(b) - g(b)\big) \lambda(b) \right). \end{aligned} \tag{5.6}$$

5.3 The Deep Uzawa iteration

The Deep Uzawa algorithm integrates the Uzawa iteration with neural network-based minimisation, alternating between minimising the discrete energy functional and updating the Lagrange multiplier. The procedure is detailed in Algorithm 1.

The total number of training epochs is $N_{\mathrm{SGD}} \times N_{\mathrm{Uz}}$.

Algorithm 1 Deep Uzawa Iteration.

```
Require: Initial guess $\lambda^0$, Uzawa step size $\rho > 0$, number of Uzawa steps $N_{\mathrm{Uz}}$, number of
    SGD iterations $N_{\mathrm{SGD}}$, learning rate $\eta$.
 1: $k \leftarrow 0$.
 2: Initialise neural network parameters $\theta^0$.
 3: for $k = 1$ to $N_{\mathrm{Uz}}$ do
 4:   for $m = 0$ to $N_{\mathrm{SGD}} - 1$ do
 5:     Compute stochastic gradient $\nabla_\theta L_{Q,\gamma}(u_\theta^m, \lambda^k)$.
 6:     Update parameters: $\theta^{m+1} \leftarrow \theta^m - \eta \nabla_\theta L_{Q,\gamma}(u_\theta^m, \lambda^k)$.
 7:   end for
 8:   $u^k \leftarrow u_\theta^{N_{\mathrm{SGD}}}$.
 9:   Update Lagrange multiplier: $\lambda^{k+1}(b) \leftarrow \lambda^k(b) + \rho(u^k(b) - g(b)), \forall\, b \in \partial\mathcal{K}_h$.
10:   $k \leftarrow k + 1$.
11: end for
```

6 Numerical results

In this section, we present numerical experiments to evaluate the performance of our proposed methodologies, focusing on the effectiveness of the Deep Uzawa approaches (RitUz and PINNUz) for singularly perturbed boundary value problems.

For these experiments, we use the PyTorch Adam optimiser [12] with a learning rate of $\eta = 10^{-3}$. Unless stated otherwise, we set $N_{\mathrm{SGD}} = 40$ and $N_{\mathrm{Uz}} = 500$, and adopt the sigmoid-weighted linear unit (SiLU) activation function [5]

$$\sigma(x) := \frac{x}{1 + e^{-x}}. \tag{6.1}$$

The choice of $N_{\mathrm{SGD}} = 40$ was to ensure that the convergence is achieved. In Fig. 6.10, we explore how the error varies with respect to N_{SGD} for a specific problem. In this case we observe diminishing returns, with respect to the error, as N_{SGD} increases. The optimal choice of N_{SGD} will depend on the functional L and the dimension of the problem. The choice of $N_{\mathrm{Uz}} = 500$, was to illustrate the impact that poor choices of ρ and γ have on the rate of convergence of the error. In practical application, this value may be significantly lower.

6.1 Example: Two-sided 1D boundary layer

As a benchmark, we consider a singularly perturbed problem on the domain $\Omega = (0, 1)$ with a small parameter $\epsilon > 0$. The exact solution exhibits boundary layers, presenting a challenge for standard penalty-based neural network approaches that often require careful tuning of penalty weights for accurate results. This example allows us to examine how varying the parameters ρ, γ, and ϵ influences the performance of RitUz and PINNUz.

We seek $u^* \in \mathrm{H}^1(\Omega)$ that satisfies

$$-\epsilon\Delta u^* + u^* = 1 \quad \text{in } \Omega, \quad u^*(0) = u^*(1) = 0. \tag{6.2}$$

The exact solution is given by

$$u^*(x) = 1 - \frac{e^{(1-x)/\sqrt{\epsilon}} + e^{x/\sqrt{\epsilon}}}{e^{1/\sqrt{\epsilon}} + 1}, \quad x \in \Omega. \tag{6.3}$$

This setup provides a test case to evaluate how RitUz and PINNUz schemes perform under varying conditions, demonstrating their capability to handle singularly perturbed problems without extensive tuning. The neural networks used in these experiments have depth $L = 5$ and width $h = 40$, parameterised by θ.

6.1.1 RitUz

Figs. 6.1 and 6.2 show the results of applying the RitUz algorithm to the problem defined in Example 6.1. In these experiments, we fix the PDE parameter $\epsilon \in \{10^{-1}, 10^{-3}\}$ and the boundary parameter $\gamma \in \{0, 2\}$. We vary the Uzawa parameter $\rho \in [0.01, 20]$ to study its impact on convergence. Note that comparison to vanilla penalty methods is given in Section 6.3.

For both large and small values of ϵ, we observe that the rate of convergence of the L^2-error with respect to the update number increases as ρ increases. This trend holds until ρ exceeds the bound specified by condition (3.16), after which u^k fails to converge to u^* as the update number increases.

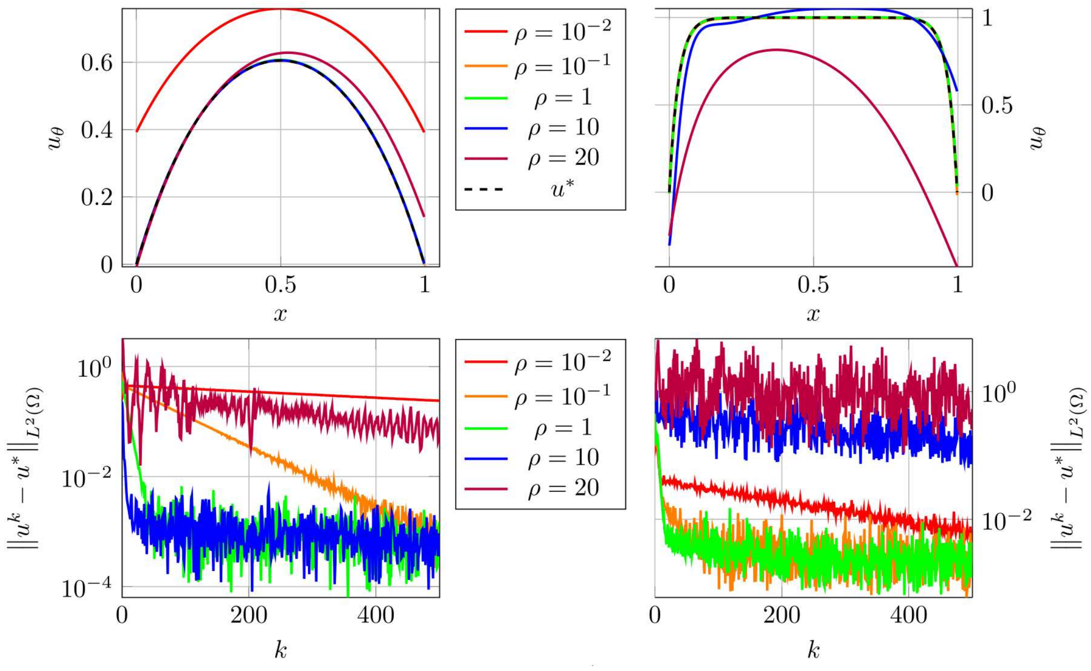

Figure 6.1: Results for Example 6.1 using the RitUz scheme with $\gamma = 2$ and $\epsilon = 10^{-1}$ (left) and $\epsilon = 10^{-3}$ (right). The plots show the state u_θ (top), and the L^2-error $\|u^k - u^*\|_{L^2(\Omega)}$ (bottom) as ρ varies in $[0.01, 20]$.

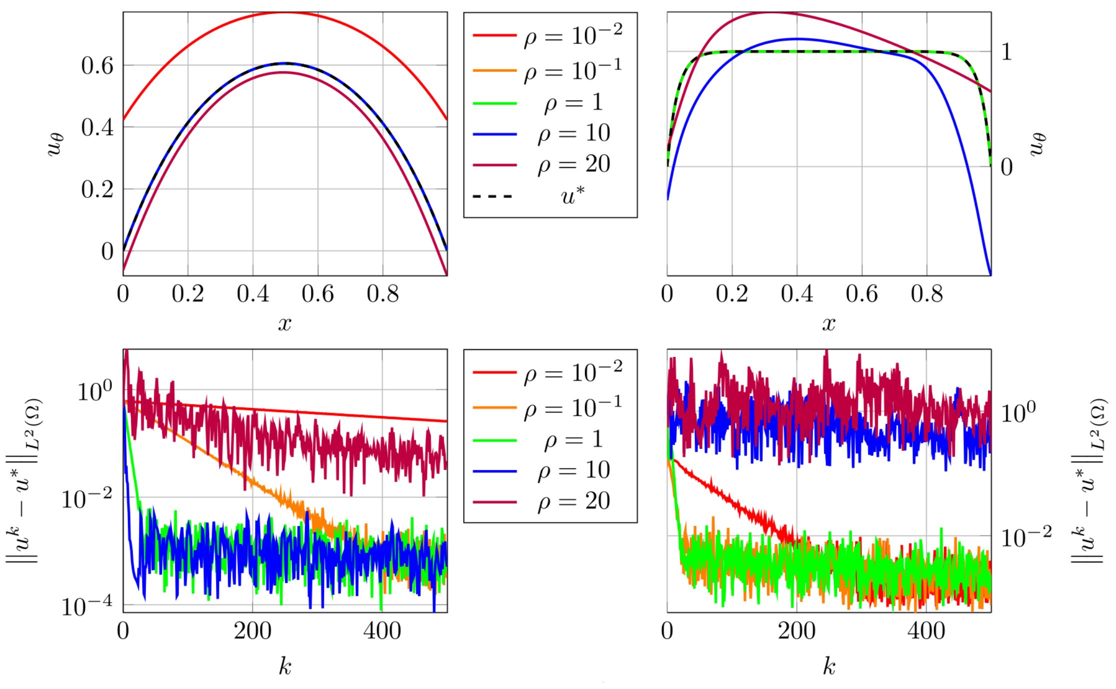

Figure 6.2: Results for Example 6.1 using the RitUz scheme with $\gamma = 0$ and $\epsilon = 10^{-1}$ (left) and $\epsilon = 10^{-3}$ (right). The plots show the state u_θ (top), and the L^2-error $\|u^k - u^*\|_{L^2(\Omega)}$ (bottom) as ρ varies in $[0.01, 20]$.

6.1.2 PINNUz

Fig. 6.3 presents the results of the PINNUz algorithm applied to the problem from Example 6.1. In these experiments, we fix the PDE parameter $\epsilon \in \{10^{-1}, 10^{-3}\}$ and set the boundary Lagrangian parameter $\gamma = 2$. The Uzawa parameter ρ is varied in the range $[10^{-4}, 5]$ to examine its effect on convergence.

As with the RitUz scheme, we observe that the L^2-error convergence rate with respect to the update number improves as ρ increases. This holds until ρ violates the condition in Eq. (4.9), causing u^k to fail to converge to u^* as the update number grows.

6.2 Example: 2D L-shaped domain

As highlighted in Remark 4.1, extending our methodology to non-convex domains introduces additional challenges due to geometric singularities. To demonstrate the robustness of the PINNUz scheme in such settings, we consider a non-convex, two-dimensional L-shaped domain $\Omega := (-1,1)^2 \setminus ([0,1) \times (-1,0])$. In this example, the solution $u^* \in H^1(\Omega)$ satisfies the reaction-diffusion equation

$$-\epsilon \Delta u^* + u^* = f \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega. \tag{6.4}$$

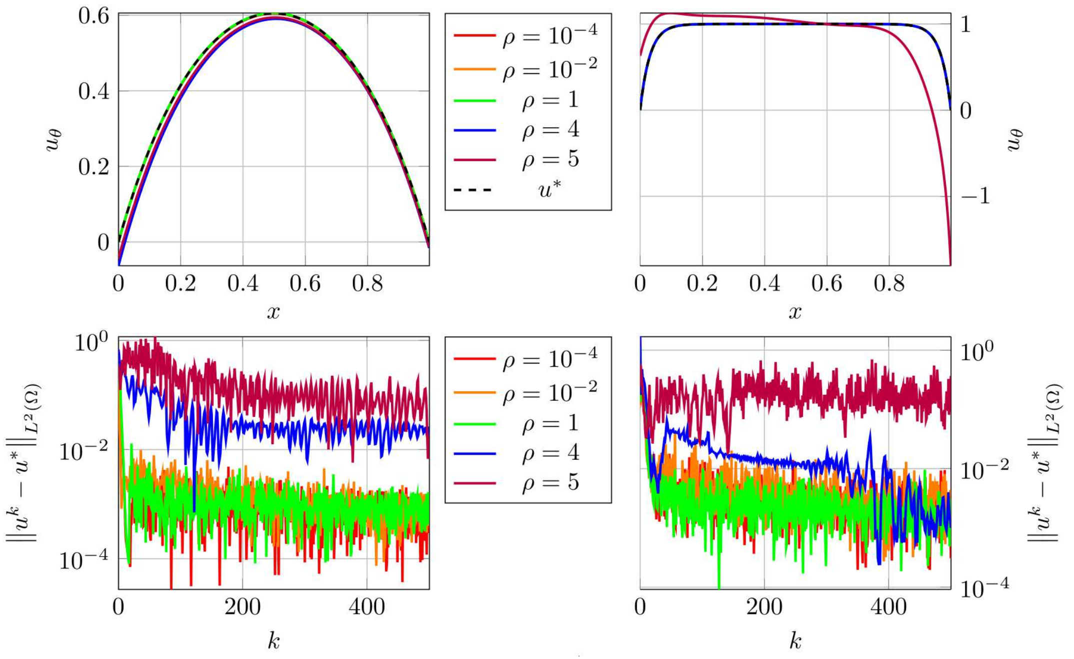

Figure 6.3: Results for Example 6.1 using the PINNUz scheme with $\gamma = 2$ and $\epsilon = 10^{-1}$ (left) and $\epsilon = 10^{-3}$ (right). The plots show the state u_θ (top), and the L^2-error $\|u^k - u^*\|_{\mathrm{L}^2(\Omega)}$ (bottom) for $\rho \in [10^{-4}, 5]$.

We choose f and g so that the exact solution is

$$u^*(x,y) = \left(x^2+y^2\right)^{\frac{2}{3}} \sin\left(\frac{2}{3}[\operatorname{atan2}(y,-x) - \pi]\right) \times \left((x+1)^2 + (y-1)^2\right)^{\frac{2}{3}} \sin\left(2 \cdot \operatorname{atan2}(y-1, x+1)\right). \tag{6.5}$$

It is chosen to have the correct asymptotic behaviour at the origin whilst also non-trivial boundary conditions elsewhere. An illustration of this solution is shown in Fig. 6.4(a).

For this experiment, we use $\eta = 10^{-4}$. We study the behaviour of the networks for fixed values of ϵ and γ, while varying ρ. The network architecture differs slightly from the one-dimensional example, in that $L = 10$ and $h = 40$.

Figs. 6.4(b)-6.4(c) displays the state u_θ for different values of ρ: two cases where the solution converges in the L^2-norm and one where it diverges. In the convergent cases, we observe that the approximation struggles to match the boundary data near the re-entrant corner at $(0,0)$, a common challenge in non-convex domains due to reduced regularity. When ρ exceeds the threshold defined in Eq. (4.9), the scheme fails to converge globally.

Similar to the one-dimensional examples, in Fig. 6.5 we observe convergence of the L^2-error, with respect to update number, for small values of ρ and non-convergent behaviour as ρ exceeds the previously specified bound. Thus demonstrating that the analytical result is applicable to higher dimensional problems, in non-convex geometry.

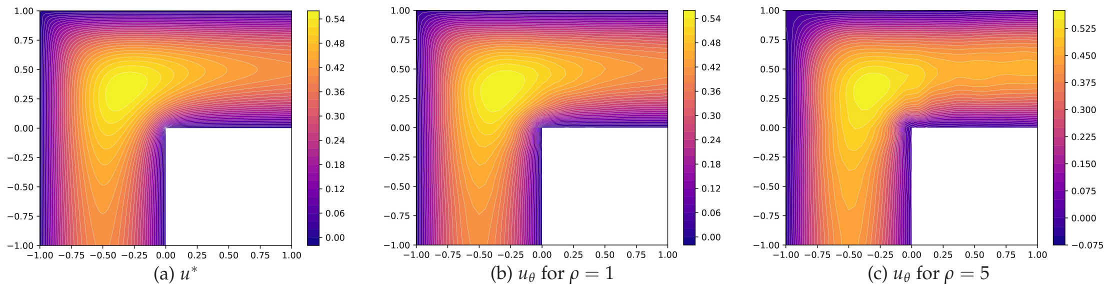

Figure 6.4: Example 6.2: (a) Illustration of the exact solution u^* on the L-shaped domain as defined in Eq. (6.5). (b) and (c) show the state u_θ for $\gamma = 2$, $\epsilon = 10^{-3}$, and $\rho = 1$ and $\rho = 5$, respectively.

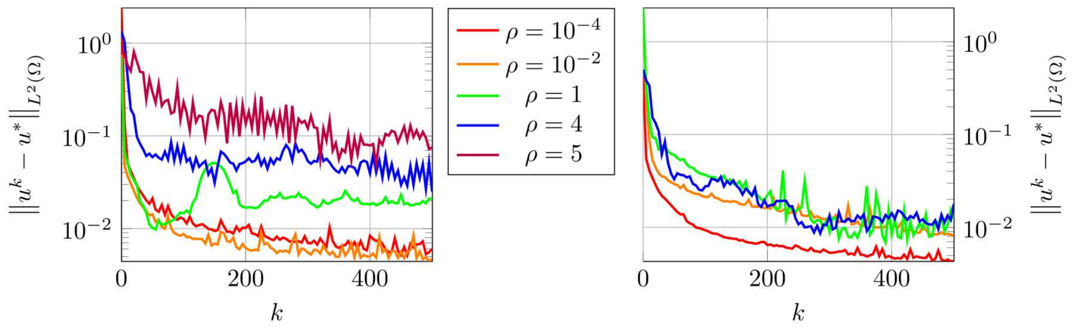

Figure 6.5: Example 6.2: The L^2-error $\|u^k - u^*\|_{\mathrm{L}^2(\Omega)}$ for $\gamma = 2$, $\epsilon = 10^{-1}$ (left) and $\epsilon = 10^{-3}$ (right) with $\rho \in [10^{-4}, 5]$.

6.3 Validating against penalty-based boundary

Penalty-based methods impose boundary conditions by minimising

$$u^D_{\theta,\gamma} := \min_{u_\theta \in \mathcal{V}_N} J_D(u_\theta), \quad u^R_{\theta,\gamma} := \min_{u_\theta \in \mathcal{V}_N} J_R(u_\theta), \tag{6.6}$$

where J_D and J_R are defined in Eqs. (3.9) and (4.3), respectively. The penalty parameter γ controls the weight of the boundary condition enforcement.

Penalty-based methods are flexible and straightforward to implement but require careful tuning of γ. High values of γ can lead to better adherence to boundary conditions but may also cause numerical stiffness and ill-conditioning, making optimisation more difficult.

6.3.1 RitUz and PINNUz

Figs. 6.6 and 6.7 illustrate the performance of penalty-based methods for different γ values. For moderate γ, the solution approximates the true solution, but larger γ can lead to issues with stiffness. The RitUz and PINNUz schemes, implemented with moderate ρ values and reduced γ, show that these methods can achieve reliable convergence without extensive parameter tuning.

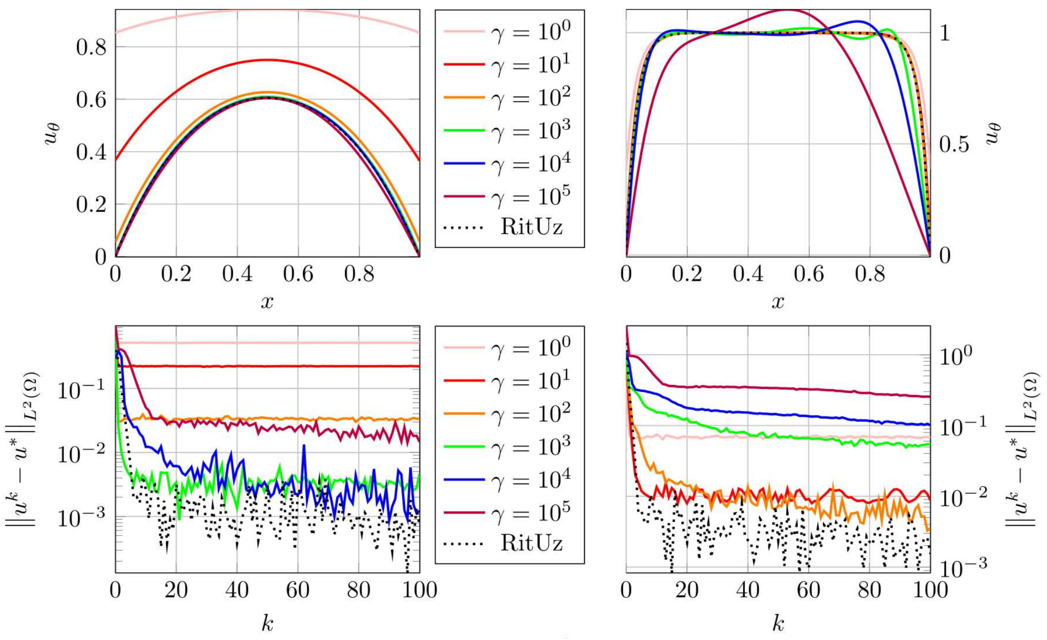

Figure 6.6: Example 6.1: For $\epsilon = 10^{-1}$ (left) and $\epsilon = 10^{-3}$ (right), results from minimising J_D over $\mathcal{V}_N$ for $\gamma \in [1, 10^5]$. The plots show the state (top) and L^2-error vs. update number (bottom). Outputs of the RitUz scheme for $\gamma = 0$ and $\rho = 1$ are also shown.

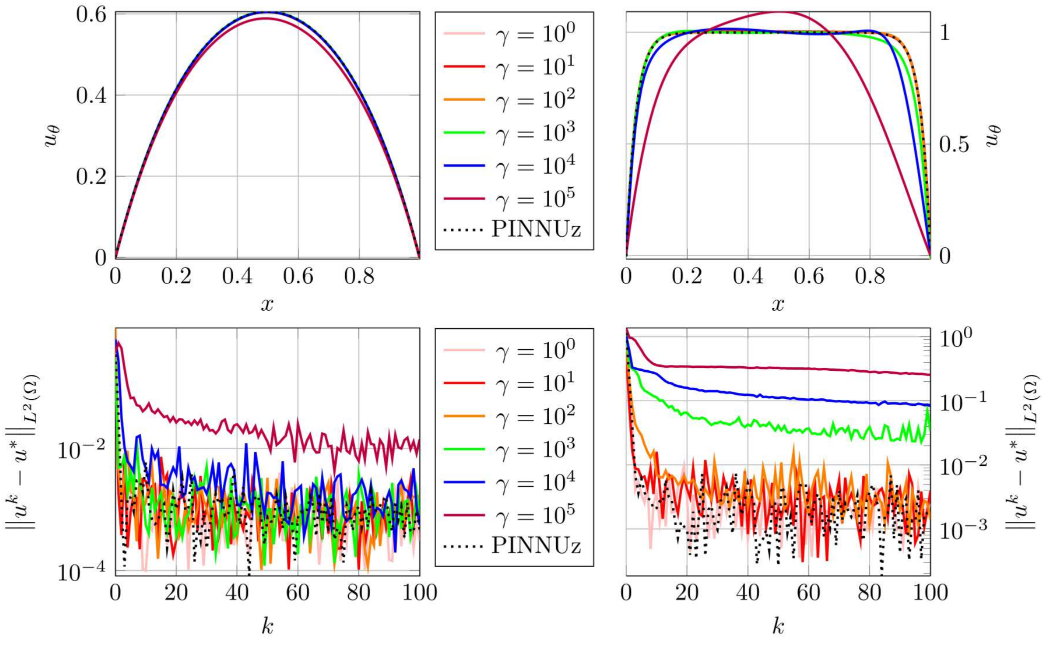

Figure 6.7: Example 6.1: For $\epsilon = 10^{-1}$ (left) and $\epsilon = 10^{-3}$ (right), results from minimising J_R over $\mathcal{V}_N$ for $\gamma \in [1, 10^5]$. The plots show the state (top) and L^2-error vs. update number (bottom). Outputs from the PINNUz scheme for $\gamma = 2$ and $\rho = 1$ are also plotted.

6.4 Example: Higher-dimensional problems

In this section, we consider solving Laplace's equation on the $2d$-dimensional unit sphere, extending the problem to higher dimensions. The harmonic function $u^* \in \mathrm{H}^1(\Omega)$ satisfies the Dirichlet boundary condition

$$u^*(\boldsymbol{x}) = \sum_{i=1}^{d} x_{2i-1}x_{2i}, \quad \forall \boldsymbol{x} \in \mathbb{S}^{2d-1}. \tag{6.7}$$

A common method for enforcing boundary conditions directly is through hard imposition, which modifies the neural network's output to inherently satisfy the boundary conditions [14,15,21].

6.4.1 RitUz

We compare three methodologies for this problem: hard boundary conditions (cRitz) [14, 21], the Ritz penalty method, and the RitUz scheme.

The cRitz method conditions the neural network output as follows:

$$(1-|\boldsymbol{x}|^2)u_\theta(\boldsymbol{x}) + |\boldsymbol{x}|^2 u^*(\boldsymbol{x}) \quad \mapsto \quad u_\theta(\boldsymbol{x}). \tag{6.8}$$

The penalty method includes a standard $\mathrm{L}^2(\partial\Omega)$ penalty, and the RitUz scheme follows the iterative approach previously described.

For these experiments, collocation points are uniformly sampled within the domain every 10 epochs, with a batch size of 1024 for the Dirichlet energy computation. Boundary points are sampled with a batch size of 2048, updated every 10 epochs. In the Uzawa scheme, an initial fixed set of 2048 boundary points is used for approximating λ. We employ a network depth of $L = 5$ and width $h = 40$.

Fig. 6.8 shows the L^2-error of the Ritz methods per epoch, indicating that the RitUz method with $\gamma = 10$ achieves significantly lower errors compared to the penalty method and performs similarly to the hard boundary condition approach. The increase in error with dimension suggests that ρ may need to be adjusted due to the dimensional dependency of C_{tr}, as discussed in Remark 2.1.

Fig. 6.9 shows the L^2-error after $50,000$ epochs for each of the three Ritz methods, plotted against the dimension of the problem. We observe similar performance for low-dimensional systems between the cRitz and RitUz methods, but a rising trend in the error of the RitUz method as the dimension increases.

6.4.2 PINNUz

We extend the experiment to PINNs, comparing hard boundary conditions (hPINNs) [15], penalty PINNs, and the PINNUz scheme. The network architecture and sampling methodology are as in the previous section.

Fig. 6.10 shows the error variation with respect to N_{SGD} for the PINNUz scheme. Higher dimensions require more epochs per Uzawa step to maintain accurate updates. For

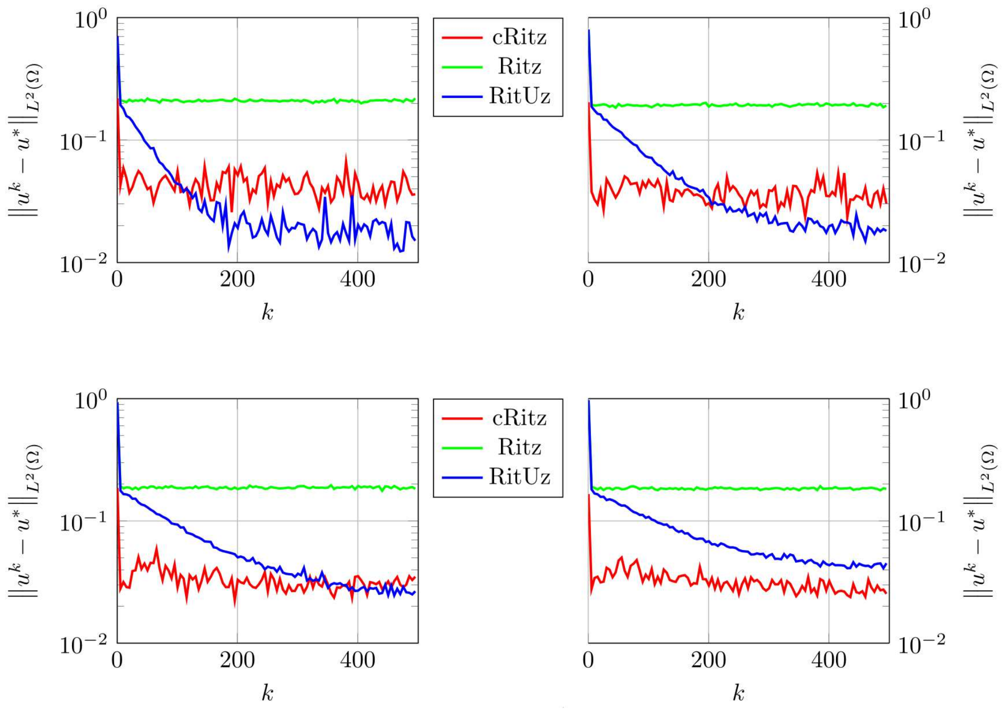

Figure 6.8: Example 6.4: Comparison of L^2-errors for the cRitz, penalty, and RitUz methods in dimensions 4 (top left), 6 (top right), 8 (bottom left), and 10 (bottom right) with $\gamma = 10$ and $\rho = 0.1$.

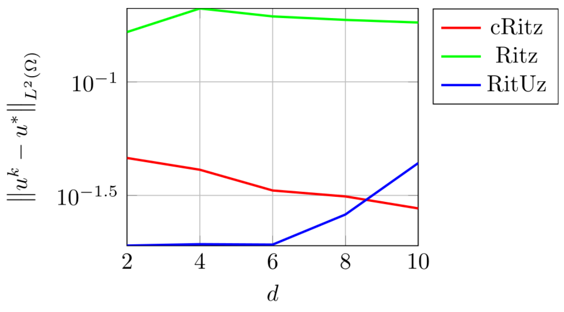

Figure 6.9: Example 6.4: Final L^2-errors for the cRitz, penalty, and RitUz methods after $50,000$ epochs for dimension $d = 2, 4, 6, 8, 10$.

$N_{\mathrm{SGD}} = 500$, performance matches the hard boundary condition for 2D and 4D problems, but errors increase for higher dimensions, suggesting that N_{SGD} may need adjustment.

Fig. 6.11 presents the final L^2-errors for hPINNs, PINNs and the PINNUz scheme across dimension. The figure highlights that for lower-dimensional problems (2D and 4D), the PINNUz scheme performs comparably to hard boundary condition methods, achieving similar error levels.

Fig. 6.12 shows that the computation time scales approximately linearly with problem dimension across hPINNs, PINNs, and PINNUz.

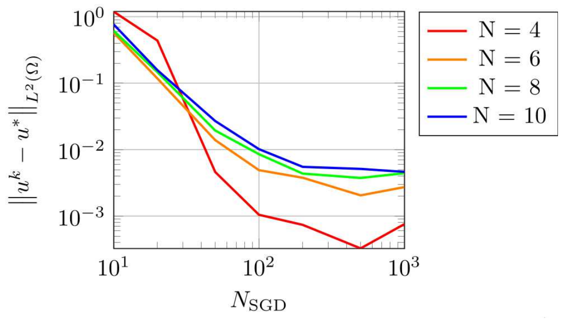

Figure 6.10: Example 6.4.2: Error of the PINNUz scheme versus N_{SGD} for $\gamma = 2$, $\rho = 0.1$, and various dimensions.

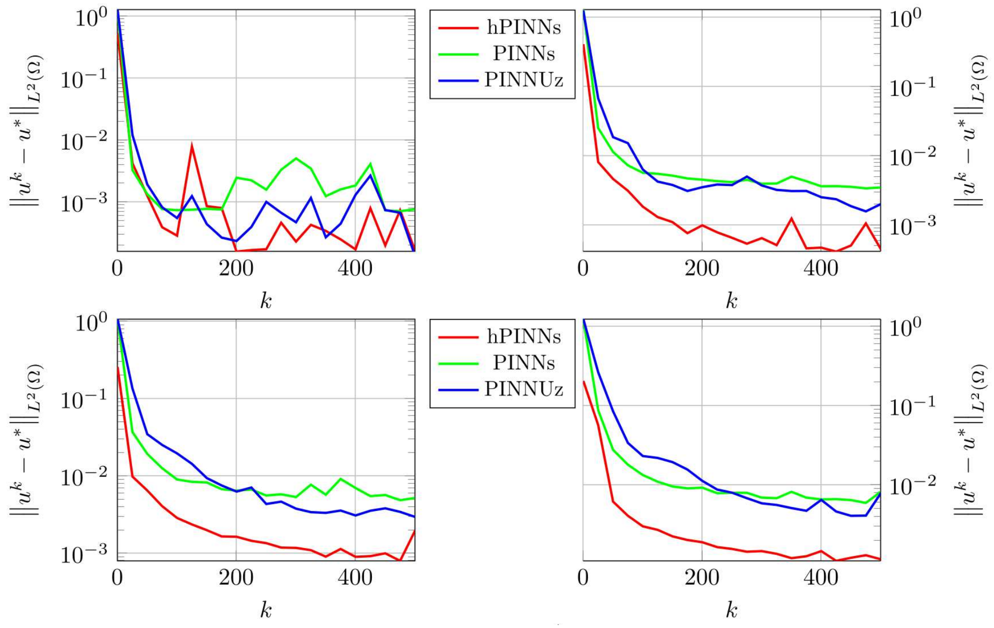

Figure 6.11: Example 6.4.2: L^2-errors for hard boundary conditions, penalty methods, and the PINNUz scheme for dimensions $2, 4, 6$, and 8, averaged over three trials.

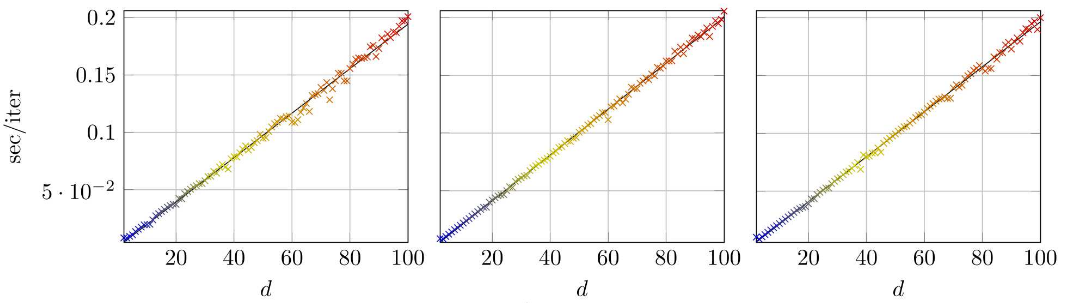

Figure 6.12: Example 6.4.2: Average time per iteration against dimension $d = 2, \ldots, 100$; for hard boundary conditions (left), penalty methods (centre), and the Uzawa method (right).

Acknowledgements

This work was initiated under a during a hosted visit at ITE, Crete. A. Pim and T. Pryer received support from the EPSRC Programme Grant EP/W026899/1. T. Pryer was also supported by the Leverhulme RPG-2021-238 and by the EPSRC Grant EP/X030067/1.

References

[1] I. Babuška, The finite element method with Lagrangian multipliers, *Numer. Math.*, 20(3):179–192, 1973.

[2] M. Berggren, Approximations of very weak solutions to boundary-value problems, *SIAM J. Numer. Anal.*, 42(2):860–877, 2004.

[3] A. Cangiani, Z. Dong, and E. Georgoulis, hp-Version discontinuous Galerkin methods on essentially arbitrarily-shaped elements, *Math. Comp.*, 91(333):1–35, 2022.

[4] W. E and B. Yu, The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.*, 6(1):1–12, 2018.

[5] S. Elfwing, E. Uchibe, and K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Networks*, 107:3–11, 2018.

[6] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, in: *Applied Mathematical Sciences*, Vol. 159, Springer, 2004.

[7] E. Gagliardo, Caratterizzazioni delle tracce sulla frontiera relative ad alcune classi di funzioni in n variabili, *Rendiconti del Seminario Matematico della Università di Padova*, 27:284–305, 1957.

[8] D. Gazoulis, I. Gkanis, and C. G. Makridakis, On the stability and convergence of physics informed neural networks, *arXiv:2308.05423*, 2023.

[9] E. H. Georgoulis, M. Loulakis, and A. Tsiourvas, Discrete gradient flow approximations of high dimensional evolution partial differential equations via deep neural networks, *Commun. Nonlinear Sci. Numer. Simul.*, 117:106893, 2023.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[11] P. Grohs and L. Herrmann, Deep neural network approximation for high-dimensional elliptic PDEs with boundary conditions, *IMA J. Numer. Anal.*, 42(3):2055–2082, 2022.

[12] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *CoRR*, abs/1412.6980, 2014.

[13] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, 521(7553):436–444, 2015.

[14] Y. Liao and P. Ming, Deep Nitsche method: Deep Ritz method with essential boundary conditions, *Commun. Comput. Phys.*, 29(5):1365–1384, 2021.

[15] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, Physics-informed neural networks with hard constraints for inverse design, *SIAM J. Sci. Comput.*, 43(6), B1105–B1132, 2021.

[16] C. G. Makridakis, A. Pim, and T. Pryer, Deep Uzawa for PDE constrained optimisation, *arXiv:2410.17359*, 2024.

[17] J. Nitsche, Über ein variationsprinzip zur Lösung von Dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, in: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, Vol. 36, Springer, 9–15, 1971.

[18] 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, *J. Comput. Phys.*, 378:686–707, 2019.

[19] H. Sheng and C. Yang, PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries, *J. Comput. Phys.*, 428:110085, 2021.

[20] J. Sirignano and K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, 375:1339–1364, 2018.

[21] N. Sukumar and A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, *Comput. Methods Appl. Mech. Engrg.*, 389:114333, 2022.

[22] H. Uzawa, Iterative methods for concave programming, in: *Studies in Linear and Nonlinear Programming*, Stanford University Press, 154–165, 1958.

[23] J. Wang, Z.-Q. J. Xu, J. Zhang, and Y. Zhang, Implicit bias with Ritz-Galerkin method in understanding deep learning for solving PDEs, *arXiv:2002.07989*, 2020.