Numerical Analysis of Physics-Informed Neural Networks for Solving PDEs

By

Ronald Katende

BSc(Mak), MSc(PAUSTI/JKUAT)

Reg. No. 2021/HD13/23859U

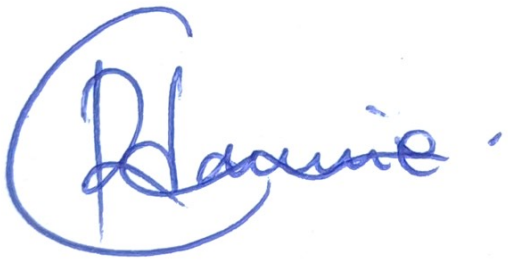
Student No. 2100723859

A Thesis Submitted to the Directorate of Graduate Training in Fulfilment of the Requirements for the Award of the Degree of Doctor of Philosophy in Mathematics of Makerere University

October, 2025

Declaration

This study is original and has not been submitted for any other degree award to any other University before.

Signature:  Date: November 17, 2025

Approval

This dissertation entitled; ***Numerical Analysis of Physics-Informed Neural Networks for Solving PDEs***, was conducted under the supervision of, and has been approved for final submission by the following supervisors:

Dr. Henry Kasumba,
Department of Mathematics,
Makerere University

Signature: Kasumba. Date: 07/11/2025

Dr. Anna Oleynik,
Department of Mathematics,
University of Bergen

Signature: A. Ol Date: October 30, 2025

Prof. Godwin Kakuba,
Department of Mathematics,
Makerere University

Signature: Kakuba. Date: 12/11/2025

Prof. John M. Mango,
Department of Mathematics,
Makerere University

Signature: Date: 12/11/2025 Mango.

Dedication

"How can I give you up, [Ronald]?, How can I hand you over, [Katende]?

...

My heart is changed within me;

all my compassion is aroused."

Hosea 11:8, NIV

"But God commendeth his love toward us, in that, while we were yet sinners, Christ died for us."

Romans 5:8, KJV

"The eternal God is your refuge,

and underneath are the everlasting arms."

Deuteronomy 33:27a, NKJV

"Whom have I in heaven but You?

And there is none upon earth that I desire besides You.

My flesh and my heart fail;

But God is the strength of my heart and my portion forever."

Psalm 73:25–26, NKJV

... to the Lordship of my Saviour, Jesus Christ

; through whom all strength, wisdom, and grace flow.

Acknowledgement

With immense gratitude, I am eternally thankful to the Norwegian Programme for Capacity Development in Higher Education and Research for Development (NORHED) for supporting this work through a PhD scholarship. In the same spirit, I am deeply grateful to Professor Mango, the Principal Investigator of the NORHED Fellowship, whose leadership and oversight were central to this support.

I am sincerely thankful to my supervisors - Dr. Henry Kasumba; Prof. Godwin Kakuba; Prof John M. Mango of Makerere University Department of Mathematics and Dr. Anna Oleynik of University of Bergen ; for their mentorship, patience, and thoughtful guidance throughout this project. Likewise, I extend my thanks to all lecturers and staff members of the Department of Mathematics, Makerere University for their support, and for providing a nurturing environment that allowed me to complete this work.

I also wish to thank my fellow PhD students; Frank Namugera, Afazali Zabibu, Celester Ssali, Ismael Gaanya, and the colleagues from UDSM, Tanzania; with whom we have walked this seemingly long and sometimes cumbersome journey. Your companionship and encouragement have been invaluable.

I also wish to thank my family members for their unwavering love and support. Manifest Fellowship Kabale; which will forever hold a special place in my heart; Kabale University, and the Department of Mathematics in particular, for their understanding and grace during this academic season, as well as the staff members of the Department of Mathematics, Makerere University.

Finally, and above all these, I deeply appreciate the love, company, understanding and prayers of ***MY WIFE, Mrs. Patience N.K. Katende***, whose presence in my life has been my anchor through the latter phase of this work.

Of course, above all these blessings, I humbly acknowledge that it is only by grace through faith that I have received the strength and gift to finish this journey. I therefore conclude with praise and exaltation to my Lord and Saviour, Jesus Christ, for His eternal gift of life and salvation.

List of Figures

List of Tables

Abstract

This thesis presents an investigation into the theory, analysis, and enhancement of Physics-Informed Neural Networks (PINNs) for solving partial differential equations (PDEs). While PINNs have gained popularity for approximating PDE solutions using machine learning, their mathematical foundations, error control mechanisms, and integration with classical numerical methods remain underdeveloped. This work addresses these gaps by developing novel energy-norm and residual-based error estimates, thereby providing the first precise convergence characterizations of PINNs in Sobolev spaces. These bounds not only establish the well-posedness of PINN formulations but also enable their use in a posteriori error estimation and adaptive refinement schemes.

Building on this foundation, the thesis introduces a hybrid numerical framework that utilizes PINN residuals to identify high-error regions in Finite Difference Method (FDM) solutions to nonlinear PDEs, as a way to guide adaptive mesh refinement. The resulting FDM solvers are shown to achieve significantly improved accuracy with fewer degrees of freedom for Burgers' and the 2D incompressible Navier-Stokes equations. This constitutes a novel and practical use of PINNs not merely as solvers, but as refinement advisors for classical methods.

In addition, the thesis addresses the computational bottlenecks inherent in training large-scale PINNs by introducing a hierarchical matrix (H-matrix) compression strategy for the residual Jacobian and neural tangent kernel. This structured approximation significantly reduces memory usage and computational cost while preserving the spectral and convergence properties of the original network. Theoretical guarantees are provided for spectral stability, training error propagation, and convergence recovery under adaptive compression. These methods allow for energy-efficient PINN deployment on constrained hardware, enabling real-time PDE inference without compromising accuracy.

Beyond algorithmic improvements, the thesis develops a rigorous mathematical framework for analyzing PINNs as numerical solvers. Specifically, it establishes formal definitions and proofs for energy-norm convergence, variational stability, and operator consistency in the context of both linear and nonlinear PDEs. Using tools from functional analysis and variational theory,

the work derives upper and lower bounds on solution error based on the empirical residual, and shows that under mild regularity assumptions, PINNs exhibit convergence properties analogous to Galerkin methods. A sequence of results; including coercivity-driven error bounds, Ritz-type projection identities, and Γ-convergence of the discrete residual loss; demonstrate that residual minimization in PINNs leads to convergence in Sobolev and uniform norms. These results position PINNs as a basis-free, adaptive numerical scheme that inherits the core theoretical guarantees of classical methods, while remaining flexible and mesh-free. This synthesis of theory and practice offers a principled framework for reliable PINN deployment in scientific and engineering problems governed by PDEs.

Ekiwandiiko kino mu bufunze

Ekiwandiiko kino kinoonyereza ku ngeri moodulo ez'emikutu ezikozesa amateeka g'obutonde (Physics-Informed Neural Networks) mu nsengeka z'okubala ezesigamiziddwa ku bintu ebisukka ekimu (partial differential equations) gyezeeyisaamu, ate n'okulambulula ku mitendera emituufu egy'okusikozesaamu, nga zeesigika.

Moodulo zino eziddukkanyizibwa ku kompyuta, ziyiga engeri y'okulambikamu n'okuwa ebyokuddamu byensengeka (solutions to equations) nga ziyigirizibwa okusoma amateeka g'obutonde agali mu nsengeka ezo. Moodulo zino zikozesebwa okunoonya ebyokuddamu eri ensengeka enzibu ezikozesebwa okwoleka engeri ebintu gye bikyuka n'obudde wamu n'ekiseera oba gyebikyuuka n'ebintu ebirala. Okugeza engeri ebbugumu gye lisaasaana, engeri amayengo gye gatambulamu, oba engeri amazzi gye gatambulamu.

Ekitundu ekisooka eky'ekiwandiiko kino kiraga engeri moodulo zino gyeziyambako moodulo enkadde zi kaasangwawo, okutumbuula enkola y'azikaasangwawo. Mu kino tulambika obukakafu obusooka obutegeerekeka obulaga ddi n'engeri moodulo zino gye ziyambamu moodulo zikaasangwawo okwongera ku butuufu bwazo. Twoleka engeri okulemererwa kwa zikasaangwawo naddala ezeesigamiziddwa ku budaala oba giridi (grid) gye kudaabulurwa n'okukendezebwaamu nga tukoseza moodulo zino ez'emikutu. Tulaga nti moodulo ez'emikutu n'amateeka g'obutonde zisobola okulaga moodulo zikaasangwawo wezilemererwa. Olwo obudaala nebutereezebwa mu bitundu ebyo byokka. Kino kitegeeza nti okukozesa moodulo zino zombi kisingako mu butuufu wamu n'okukekkereza obudde n'ebikozesebwa.

Ekitundu eky'okubiri kissa essira ku nkozesa ennungamu eya zi moodulo zino mu budde obusaamusaamu. Okutendeka moodulo zino ez'emikutu n'amateeka g'obutonde ebiseera ebisinga kyetaagisa obudde bungi wamu n'ekitundu ku bwongo bwa kompyuta nga kiwereera ddala. Okusobola okugonjoola kino, ekiwandiiko kino kyanjula enkola y'okunyigiriza (Hierarchical matrices) ekendeeza ku bunene n'okuzitowa kwa moodulo z'emikutu. Kino kizisobozesa okutendekebwa mu bwangu, era nezikozesa obwongo bwa kompyuta obusaamusaamu n'ebiseera ebisaamusaamu.

Ekiwandiiko kino kifundikira n'okuwa omusingi omunywevu ogw'okukozesa moodulo z'emikutu, wamu n'okunnyonnyola lwaki zikola – mu ngeri gyezikolamu. Tulaga ebilina okutuukirizibwa

moodulo zino okusobola okukozesebwa mu butuufu, nga ebyokuddamu ebizivaamu byesigika. Ku kino, tulambika amateeka n'obukwakkulizo obulina okukakasibwa nga ebiva mu moodulo zino tebineesigibwa. Nga ovude kwekyo, tulaga engeri y'okutegeeramu obugumu, n'obunywevu bwa zi moodulo zino okusinziira mu mateeka ge ssomo ly'okubala erinoonyereza ku masangazira g'okubala ne zi kompyuta.

Tumaliriza n'okulaga nti moodulo zino zeesigika era zibandaala n'ebyokuddamu ebyobuzaaliranwa era ebitukiridde (exact solutions) ebyalibadde bisuubirwa okuva mu nseengeka zebyo byezikozeeeko. Kino kiwa obwesige nti moodulo zino zigoberera amateeka ge ssomo ly'okubala erinonoonyereza ku masangazira g'okubala ne zi kompyuta. Ebivaamu ebirala bitulaga nti moodulo zino zituukana n'omutindo n'ezo zi kaasangwawo.

Ebitundu bino byonsatule byongera okuwa abanoonyereza omusingi oumgumu era omwesimbu okukozesa moodulo zino okungonjoola, okunoonyereza n'okufuna eby'okuddamu mu ssaayansi ne'byobuyiiya

Contents

CHAPTER 1

Introduction

1.1 Background

Partial Differential Equations (PDEs) form the mathematical foundation for modeling a wide range of physical and engineering phenomena, including heat conduction, fluid dynamics, quantum mechanics, and structural deformation. Analytical solutions exist only for a limited class of PDEs, necessitating the use of numerical techniques for practical problem-solving.

Classical numerical methods such as the Finite Difference Method (FDM) (LeVeque, 2007), the Finite Element Method (FEM) (Hughes, 2012; Zienkiewicz and Taylor, 2005), and the Finite Volume Method (FVM) (Eymard et al., 2000) reformulate PDEs into discrete algebraic systems defined over spatial meshes. These approaches are underpinned by rigorous mathematical theory, notably the Lax–Richtmyer equivalence theorem (Lax and Richtmyer, 1956), which links consistency and stability to convergence for linear initial-value problems. Consequently, classical schemes provide formal guarantees for accuracy and error control under mesh refinement.

More recently, Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) have emerged as a mesh-free alternative that leverages the expressive power of neural networks to approximate PDE solutions. In this framework, a neural network $u_\theta(x, t)$ is trained to minimize a composite loss functional that penalizes violations of the governing PDE, boundary conditions, and available data. Automatic differentiation enables the enforcement of PDE constraints without mesh-based discretization. PINNs have been successfully applied to a variety of systems, including convection–diffusion equations (Li et al., 2024; Zhang and Li, 2024), moving-boundary problems (Zhou and Xu, 2024), high-dimensional PDEs (Li and Chen, 2023), and complex fluid dynamics (Karniadakis et al., 2021).

Despite their promise, several challenges limit the scalability and reliability of PINNs (Badia et al., 2024; Ganga and Uddin, 2024). Chief among these are the high computational costs of training large networks on fine-grained domains (Daw et al., 2022), driven by dense Jacobians and

neural tangent kernels (Wang et al., 2022b). Furthermore, unlike classical solvers, PINNs lack a mature mathematical foundation for analyzing stability, consistency, and convergence (De Ryck and Mishra, 2022). This absence hinders interpretability and complicates their comparison with traditional numerical schemes.

This thesis seeks to address these gaps by developing theoretical and computational frameworks that place PINNs on a rigorous mathematical foundation, comparable to FDM and FEM. The work establishes residual-based convergence results, a posteriori error bounds in Sobolev spaces, and stability conditions, while introducing hierarchical matrix compression techniques to accelerate training without sacrificing accuracy. Together, these contributions aim to enhance both the theoretical understanding and the practical reliability of PINNs in scientific and engineering computation.

1.2 Statement of the problem

While PINNs have achieved empirical success across diverse classes of PDEs, their theoretical foundations remain insufficiently developed. In contrast to classical methods such as FDM and FEM, there is no comprehensive analytical framework that formally characterizes their numerical behavior. Fundamental properties; stability, convergence, and consistency; are not yet rigorously established. The absence of these guarantees limits confidence in their reliability, hinders systematic error assessment, and impedes fair comparison with established discretization-based methods. There is thus a critical need to develop a rigorous numerical framework for PINNs that explains their behavior and ensures reliability in practice.

1.3 Classical motivation

Traditional numerical schemes, including FDM and FEM, are built on solid theoretical foundations that guarantee stability, consistency, and convergence. These properties ensure that as discretization parameters such as grid size or time step are refined, numerical solutions approach the true continuous solution. However, classical schemes often face challenges when applied to nonlinear, high-dimensional, or irregularly shaped domains, and they typically require computationally intensive mesh generation. PINNs, in contrast, offer a mesh-free approach where the solution is represented by a neural network that minimizes the residual of the governing PDE while satisfying boundary and initial conditions.

Despite this advantage, the theoretical principles governing PINN performance remain largely

empirical. It is unclear under what conditions PINNs converge, how stable their approximations are under perturbations, or how to quantify their approximation errors. This thesis addresses these theoretical gaps by formulating and analyzing the core numerical properties of PINNs; stability, consistency, and convergence; within a rigorous mathematical framework, as a way to contribute towards a comprehensive mathematical framework for PINNs, similar to those already established for classical numerical methods.

1.4 Objectives

1.4.1 Main objective

To develop mathematical frameworks for the stability, consistency, convergence, and error analysis of (PINNs) for solving PDEs.

1.4.2 Specific objectives

(i) To enhance FDM and FEM solutions for PDEs through the integration of PINN residuals.

(ii) To improve PINN training efficiency and accuracy using optimal collocation point selection.

(iii) To establish mathematical frameworks for stability, consistency, and convergence of PINNs.

(iv) To apply PINNs to image inpainting as a proof-of-concept extension.

1.5 Scope of the study

This study focuses on the numerical analysis of Physics-Informed Neural Networks as solvers for partial differential equations. The theoretical component develops mathematical formulations for error analysis, stability, consistency, and convergence of PINNs, particularly for linear and weakly nonlinear PDEs in one and two spatial dimensions. The computational component involves the design of efficient training strategies, including hierarchical matrix compression and adaptive collocation, to enhance scalability and accuracy. Experimental validation is carried out through benchmark PDE problems and an illustrative application to image inpainting. The work does not address alternative deep operator learning frameworks or large-scale turbulence simulations, which fall outside the intended scope.

1.6 Justification of the study

PINNs have emerged as powerful tools for solving PDEs, especially in data-driven scientific computing. However, their mathematical properties remain insufficiently understood. Establishing rigorous numerical frameworks for PINNs is essential for interpreting their performance, ensuring stability, and quantifying errors. The results of this work therefore advance both the theoretical understanding and the practical deployment of PINNs as reliable solvers. Several results presented here extend the existing literature by providing formal numerical analysis for PINNs as consistent, convergent, and stable schemes for solving PDEs.

1.7 Significance and applicability

This research supports Uganda's National Development Plans (NDP IV), which emphasize data science and artificial intelligence for national development. It contributes to the United Nations Sustainable Development Goals (UNSDGs), including Goal 3 (Health), Goal 9 (Industry), and Goal 13 (Climate Action). The developed frameworks enhance computational tools applicable to healthcare, engineering, materials science, and environmental modeling. For instance, in medical imaging, the stability and efficiency results may contribute to faster MRI reconstruction and improved diagnostic precision. The equivalence results between PINNs and FEM also strengthen modeling capacity in structural and industrial applications. Overall, the research connects theoretical advancements in numerical analysis with practical benefits in science, engineering, and sustainable development.

1.8 Conceptual and theoretical framework

This section articulates the *conceptual* logic and the *theoretical* underpinnings of the thesis. Conceptually, the work links PINN theory to methods, then to improvements of classical solvers and to a proof-of-concept application (image inpainting). Theoretically, it places PINNs on a firm variational and operator-theoretic foundation, deriving conditions for stability, consistency, convergence, and computable error control.

1.8.1 Conceptual framework

The research logic follows four tightly coupled streams that directly implement the stated objectives:

1. **Foundations:** Formulate PINNs as variational approximators for PDEs and derive a residual-based analytic toolkit (duality bounds, coercivity, projection identities).

2. **Classical–Neural Synergy:** Use PINN residuals to improve FDM/FEM solutions via adaptive mesh refinement and operator-aware diagnostics.

3. **Efficiency at Scale:** Improve PINN data/computational efficiency through optimal collocation (adaptive sampling) and hierarchical matrix compression of Jacobians/NTKs.

4. **Application Signal:** Demonstrate portability to a non-traditional PDE task (image inpainting via Laplace-PINNs) as a boundary case for scope and feasibility.

Figure 1.1 encodes this flow from theoretical constructs to methods, to improvements of classical solvers, and to application validation.

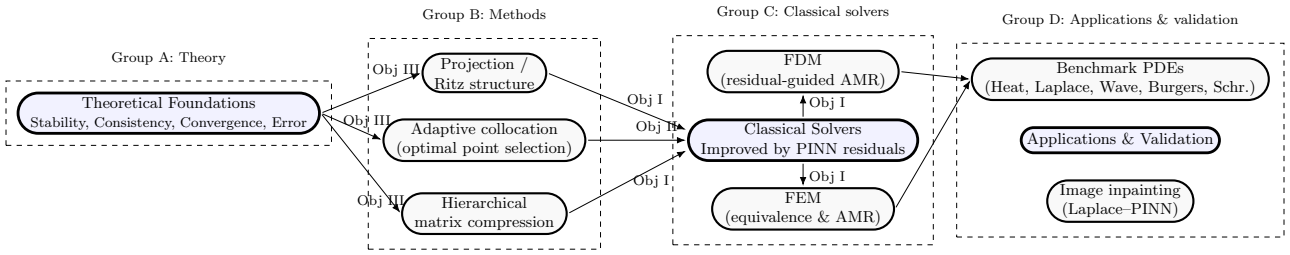


Figure 1.1: Objectives–outcomes concept map: theory → methods → enhanced classical solvers → applications. Arrows are labeled by specific objectives (i)–(iv).

1.8.2 Theoretical framework

1.8.2.1 Problem class and spaces

Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain with boundary subsets encoding boundary/initial conditions. We study linear and weakly nonlinear PDEs formulated via a (possibly nonlinear) operator $\mathcal{L} : \mathcal{U} \to \mathcal{V}$ with $\mathcal{U} \subset H^s(\Omega)$, $s > d/2$, and right-hand side $f \in \mathcal{V}$. The weak form reads: find $u^\star \in V$ such that

$$a(u^\star, v) = \langle f, v \rangle \quad \forall v \in V,$$

with V a Hilbert space (e.g. $H_0^1(\Omega)$) and $a(\cdot, \cdot)$ continuous and coercive when applicable.

1.8.2.2 PINN model and residual

A PINN parametrizes $u_\theta \in \mathcal{F}_\theta \subset \mathcal{U}$ and minimizes an empirical residual functional constructed from collocation sets in the interior and on constraints. The continuum residual is the functional

$$\mathcal{R}[u](v) := a(u, v) - \langle f, v \rangle, \qquad v \in V,$$

with dual norm $\|\mathcal{R}[u]\|_{V^*}$; its discrete counterpart is formed by pointwise evaluations of $\mathcal{L}u - f$. The analysis deploys three pillars:

(A) Consistency and variational convergence. We formalize:

- Functional weak consistency: if $\mathcal{R}[u_N] \to 0$ in V^* and $\{u_N\}$ is bounded in H^s, then $u_N \rightharpoonup u^\star$ and $u_N \to u^\star$ in L^2.
- Variational energy consistency: under continuity/coercivity of $a(\cdot, \cdot)$ and density of the trial space, projected residual decay implies $\|u_N - u^\star\|_V \to 0$ (Céa-type control).
- Nonlinear stability in a neighborhood of $u^\star$: if $D\mathcal{L}(u^\star)$ is boundedly invertible, then $\|u_N - u^\star\| \lesssim \|\mathcal{L}u_N - f\|^\alpha + \inf_{v \in \mathcal{F}_\theta} \|v - u^\star\|$ for some $\alpha \in (0, 1]$.

(B) Error estimation and projection structure.

- A priori / a posteriori: $\|u - u^\star\|_V \leq \alpha^{-1} \|\mathcal{R}[u]\|_{V^*}$ and, via discrete-to-continuum lifting, computable bounds in terms of ℓ^2 residuals at collocation points.
- Nonlinear orthogonality: first-order optimality on $\mathcal{F}_\theta$ yields a *residual-space* orthogonality against the tangent space $T_{u_\theta}\mathcal{F}_\theta$, inducing an operator-weighted Ritz projection view of training.

(C) Stability and generalization.

- Deterministic perturbation bounds quantify how prediction perturbations propagate through the composite loss, yielding a CFL-like margin on allowable deviations.
- Probabilistic stability under i.i.d. sampling (McDiarmid-type concentration) controls fluctuations of empirical residuals around population loss.
- Sobolev generalization: with inverse stability to the PDE and quasi-uniform sampling, empirical Sobolev losses bound global H^s and, for $s > d/2$, pointwise C^0 errors.

1.8.2.3 Methodological instantiations

The framework permits two efficiency levers that preserve the above guarantees: *adaptive collocation* (residual-guided sampling densifies high-error regions while maintaining discrete-to-continuum stability), and *hierarchical matrix compression* of the residual Jacobian/NTK (controlled spectral perturbation of Gauss–Newton updates preserves convergence profiles).

1.8.2.4 Interfaces to classical solvers

For FDM/FEM, PINN residuals act as physics-consistent a posteriori indicators guiding adaptive refinement. Moreover, linear FEM spaces are exactly representable by shallow ReLU networks, ensuring equivalence of variational solutions and clarifying how deeper networks extend FEM spaces to richer nonlinear manifolds without remeshing.

1.8.3 Research design logic and objective mapping

The framework operationalizes the objectives as follows:

1. *Enhance FDM/FEM (Objective 1):* Use PINN residuals as certified indicators to drive AMR and to diagnose under-resolution; exploit FEM–ReLU equivalence for exact variational consistency.

2. *Improve efficiency (Objective 2):* Couple adaptive collocation with H-matrix compression to reduce residual/Jacobian cost while preserving spectral conditioning and convergence behavior.

3. *Establish theory (Objective 3):* Prove functional/variational consistency, derive computable energy/dual-norm bounds, and quantify deterministic/probabilistic stability and Sobolev generalization.

4. *Prototype application (Objective 4):* Validate portability to Laplace-based image inpainting as a controlled stress test of the methods and theory in a non-traditional PDE setting.

1.8.4 Delimitations consistent with scope

The analysis targets linear and weakly nonlinear PDEs in one and two spatial dimensions; it emphasizes residual-based certification, adaptive sampling, and matrix-structured acceleration. Large-eddy turbulence, strongly chaotic regimes, and operator-learning alternatives (e.g.

FNOs/DeepONets) are out of scope. This delimitation is deliberate: it isolates the *PINN-as-numerical-scheme* perspective, enabling rigorous guarantees and controlled validations before expanding to broader model classes.

1.9 Organisation of the thesis

Chapter 1 introduces the study, its objectives, and its significance. Chapter 2 reviews the existing literature on neural networks and PINNs, identifying key gaps motivating this work. Chapter 3 details the methodologies and techniques used to achieve the stated objectives. Chapter 4 presents the main results of this work. This includes the use of PINNs to enhance performance of FDM (§4.1.1) and FEM (§4.1.2); results on improving the efficiency of PINNs, using sampling strategies (§4.2.1) and hierarchical matrices (§4.2.2); results on the mathematical frameworks (§4.3) for error analysis of PINNs (§4.3.1), stability analysis of PINNs (§4.3.2) and consistency-convergence analysis of PINNs (§4.3.3), along with brief validation of each framework. Finally this chapter closes, with our results fork PINN-based image inpainting (§4.4). A detailed discussion of our results is provided in section §4.5, before Chapter 5.1 summarizes the findings, highlights contributions of the work, outlines the directions for future research as well as published papers derived from this work (§5.3).

1.10 Contributions

This thesis makes three primary contributions. First, it establishes theoretical foundations for the numerical analysis of PINNs by developing frameworks for error estimation, stability, consistency, and convergence, and by providing formal validation through analysis and computation. Second, it enhances the efficiency and accuracy of PINNs through hierarchical matrix compression and optimized collocation strategies, including integration with traditional methods such as FDM and FEM. Third, it extends the applicability of PINNs to inverse imaging tasks, demonstrating the versatility of physics-informed learning in solving high-dimensional and ill-posed PDE problems. Collectively, these contributions place PINNs within a unified mathematical framework comparable to classical numerical methods and broaden their potential for practical scientific and engineering applications.

CHAPTER 2

Literature Review

This chapter reviews research on (PINNs) for solving partial differential equations (PDEs), emphasizing four threads aligned with the thesis objectives, i.e.,

(i) interactions with classical numerical methods and domain decomposition,

(ii) training efficiency and robust sampling,

(iii) theoretical guarantees (stability, consistency, convergence, and error analysis), and

(iv) applications to inverse problems, with a focus on image inpainting.

We begin with core developments in neural networks and neural PDE solvers, then synthesize recent advances that directly motivate the proposed frameworks.

2.1 Development of neural networks

Early neural network models date to McCulloch and Pitts (1943), with Rosenblatt (1958) introducing the perceptron and its well-known linear separability limitation (Minsky and Papert, 1969). The resurgence of multi-layer models followed the popularization of backpropagation (Rumelhart et al., 1986). Universal approximation theorems by Cybenko (1989) and Hornik (1991) established expressivity for single-hidden-layer networks; later analyses quantified approximation rates, including for ReLU architectures (Schmidt-Hieber, 2020; Yarotsky, 2017).

2.2 Neural networks for solving PDEs

Pioneering neural PDE solvers embedded boundary conditions into trial functions and minimized PDE residuals (Lagaris et al., 1998). Variational formulations refined this direction, including the Deep Ritz method and related approaches (He et al., 2020; Sirignano and Spiliopoulos, 2018).

These works foreshadowed modern physics-informed learning by coupling function approximation with PDE constraints via differentiable objectives.

2.3 Physics-informed neural networks (PINNs)

Raissi et al. (2019) formalized PINNs by enforcing the governing PDE, boundary and initial conditions through a composite loss with automatic differentiation. Since then, applications span Navier–Stokes, wave, and high-dimensional transport problems (Nabian et al., 2021; Raissi et al., 2019; Zhang and Wang, 2024), with extensions such as fractional and conservative formulations (Mustajab et al., 2024). Recent architectures improve representation of multiscale dynamics through domain decomposition and spectral features, and incorporate attention mechanisms for long-range dependencies (Doľean et al., 2024; Moseley et al., 2023; Zhao et al., 2024).

2.4 Theoretical analysis of PINNs

2.4.1 Convergence and generalization

Early convergence results established links between residual minimization and Sobolev-norm errors under idealized assumptions (Shin et al., 2020). Subsequent work derived error estimates for Kolmogorov-type equations and bounds under spectral decay assumptions (De Ryck et al., 2022; Mishra and Molinaro, 2022). Recent analyses refine generalization and train–test gaps for physics-informed learning and related variational methods, clarifying conditions under which PINNs avoid the curse of dimensionality and remain predictive beyond the training domain (Bonfanti et al., 2024; Chen et al., 2024; Escapil-Inchauspé et al., 2023). New results also treat over-parameterized regimes and interface problems, further tightening convergence guarantees in practical settings (Chen et al., 2025).

2.4.2 Stability

Stability remains challenging for stiff or multiscale PDEs. Gradient ill-conditioning and stiffness in the optimization landscape can impede training (Wang et al., 2022a). Empirical studies showed failure modes in long-time integration and near sharp features, motivating residual reweighting, curriculum learning, and domain decomposition (cPINNs/XPINNs) (Hu et al., 2022; Krishnapriyan et al., 2021; Shukla et al., 2021). Frequency-aware methods and hierarchical decompositions; FBPINNs and their multilevel variants; substantially improve robustness and

scalability by localizing complexity while promoting cross-subdomain communication (Doľean et al., 2024; Moseley et al., 2023).

2.4.3 Consistency

Consistency; whether minimizing empirical residuals implies approximation of the true PDE solution; depends on sampling and expressivity. Static sampling can yield spurious minima, while adaptive and residual-driven strategies improve fidelity (Lu et al., 2021a; McClenny and Braga-Neto, 2020). Formal consistency has been linked to collocation density and network capacity in the large-sample/width limit (Cai et al., 2021). Beyond pointwise enforcement, region-based optimization schemes reduce generalization error by enforcing constraints over neighborhoods rather than isolated collocation points, providing a principled alternative to scatter-point training (Wu et al., 2024).

2.4.4 Error analysis

Residual-based bounds now connect training and approximation errors through Sobolev embeddings and spectral properties (Mishra and Molinaro, 2022). A posteriori estimates for high-dimensional settings highlight the role of sampling density and residual distributions (De Ryck et al., 2022). Recent generalization analyses sharpen these bounds and clarify the effect of architecture and loss design on error control (Chen et al., 2024; Escapil-Inchauspé et al., 2023). Collectively, these results suggest viable residual-based error certification under structural assumptions, paralleling classical finite element error analysis.

2.5 Domain decomposition and interaction with classical methods

Synergy with classical numerics is strongest in domain decomposition and variational design. cPINNs and XPINNs extend PINNs to space–time subdomains with parallelization benefits and provable tradeoffs in generalization (Hu et al., 2022; Shukla et al., 2021). FBPINNs adopt compactly supported neural basis functions over overlapping subdomains, mirroring finite element locality while preserving mesh-free flexibility (Moseley et al., 2023). Multilevel variants further enhance global communication and scalability for high-frequency and multiscale problems (Doľean et al., 2024). These directions directly support Objective I, providing principled ways to integrate PINN residuals with FDM/FEM pipelines.

2.6 Training efficiency, sampling, and attention mechanisms

Efficient training hinges on well-conditioned objectives and informative sampling. Adaptive collocation, residual balancing, and curriculum strategies mitigate stiffness (Krishnapriyan et al., 2021; Wang et al., 2022a). Transformer-based PINNs capture long-range dependencies and temporal structure, improving accuracy for sequence-like PDE data (Zhao et al., 2024). Region-optimization reframes enforcement from points to neighborhoods, theoretically lowering generalization error and improving stability in hidden higher-order constraints, aligning with Objective II (Wu et al., 2024). These advances complement hierarchical matrix compression and other linear-algebraic accelerations developed in this thesis.

2.7 PINNs for image inpainting

Physics-guided inpainting treats masked regions as PDE-governed subdomains (e.g., Laplace/elliptic priors). Classical PDE-based inpainting (Bertalmio et al., 2000) motivates PINN formulations in which physics losses enforce harmonicity or anisotropic diffusion while adhering to observed data (Araya-Polo et al., 2022; Yin et al., 2022). Recent surveys and physics-informed vision methods broaden the landscape across medical and scientific imaging, highlighting opportunities for uncertainty calibration and improved regularization (Banerjee et al., 2024; Dwivedi et al., 2024). These developments underpin Objective III and contextualize our Laplace–PINN inpainting experiments.

2.8 Synthesis and gaps

The literature shows steady progress on expressivity, training stability, and emerging theory. Yet unified, practice-relevant guarantees for stability, consistency, convergence, and certified error control remain incomplete, particularly for nonlinear, stiff, and multiscale regimes. Moreover, principled sampling (beyond pointwise collocation), scalable decompositions with error monitors, and rigorous imaging applications demand further development. The frameworks advanced in this thesis address these gaps by;

1. integrating PINN residuals with classical discretizations via domain decomposition,
2. formalizing efficient collocation and region-based enforcement,
3. deriving residual-driven error and convergence results, and

4. evaluating a physics-guided inpainting pipeline as a targeted inverse problem.

CHAPTER 3

Materials and Methods

This chapter presents the mathematical foundations, computational design, and methodological approaches used to achieve the four objectives of this thesis. Section 3.1 introduces the core mathematical and computational principles underpinning the developed techniques. The subsequent sections detail how these methods were adapted and implemented to accomplish each research objective.

3.1 Mathematical and computational foundations

We first summarize the key mathematical and computational structures that form the basis of the analyses and implementations throughout this work. These include the formulation of artificial neural networks, the construction of (PINNs), and the benchmark PDEs used for evaluation.

3.1.1 Artificial neural networks

A neural network (NN) is a parameterized nonlinear function approximator inspired by biological neurons. It consists of layers of interconnected units (neurons) that transform input data through weighted linear combinations and nonlinear activations.

Definition 3.1.1 (Artificial Neuron)**.** *Let* $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top \in \mathbb{R}^n$ *denote an* n*-dimensional input vector, and let* $\mathbf{w} = (w_1, w_2, \ldots, w_n)^\top \in \mathbb{R}^n$ *denote the corresponding weight vector. An* artificial neuron *computes a scalar output*

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) = f(\mathbf{w}^\top \mathbf{x} + b), \tag{3.1}$$

where $b \in \mathbb{R}$ *is a bias term and* $f : \mathbb{R} \to \mathbb{R}$ *is a (typically nonlinear)* activation function.

A feedforward neural network consists of multiple such neurons organized in successive layers. Let $L \in \mathbb{N}$ denote the total number of *trainable layers* (i.e., excluding the input layer). The input

to the network is represented by $\mathbf{x}^{(0)} \in \mathbb{R}^{n_0}$, and the activations of the j-th layer are denoted by $\mathbf{x}^{(j)} \in \mathbb{R}^{n_j}$ for $j = 1, 2, \ldots, L$.

Each layer performs an affine transformation followed by an elementwise nonlinear activation:

$$\mathbf{x}^{(j+1)} = \sigma_j\big(\mathbf{W}^{(j)}\mathbf{x}^{(j)} + \mathbf{b}^{(j)}\big), \quad j = 0, 1, \ldots, L-1, \tag{3.2}$$

where

- $\mathbf{W}^{(j)} \in \mathbb{R}^{n_{j+1} \times n_j}$ is the weight matrix connecting layer j to layer $j+1$,
- $\mathbf{b}^{(j)} \in \mathbb{R}^{n_{j+1}}$ is the bias vector,
- $\sigma_j : \mathbb{R}^{n_{j+1}} \to \mathbb{R}^{n_{j+1}}$ is the activation function, applied elementwise,
- n_j denotes the number of neurons in the j-th layer.

Throughout this thesis, unless otherwise specified, the activation function is denoted generically by σ, and the set of all trainable parameters of the network is denoted by

$$\boldsymbol{\theta} = \{\mathbf{W}^{(j)}, \mathbf{b}^{(j)}\}_{j=0}^{L-1}.$$

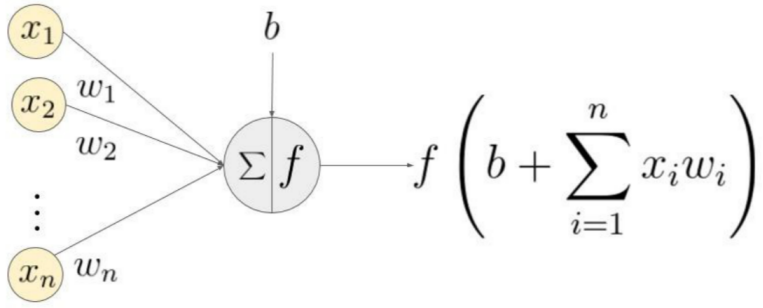


Figure 3.1: Structure of an artificial neuron showing weighted inputs $(x_1, \ldots, x_n)$, bias b, activation function f, and output y.

Figure 3.1 illustrates the basic structure of a neuron, which serves as the fundamental computational unit in neural networks.

3.1.2 Physics-informed neural networks (PINNs)

Definition 3.1.2 (Physics-Informed Neural Network (PINN))**.** *A* Physics-Informed Neural Network *(PINN) incorporates the governing physical laws of a system directly into its loss function, enabling mesh-free approximation of partial differential equation (PDE) solutions. Automatic*

differentiation is used to evaluate derivatives and enforce the PDE and boundary/initial constraints during training.

Consider a general time-dependent PDE of the form

$$\begin{cases} \mathcal{N}[u](\mathbf{x},t) = 0, & (\mathbf{x},t) \in \Omega \times (0,T], \\ \mathcal{B}[u](\mathbf{x},t) = 0, & (\mathbf{x},t) \in \partial\Omega \times (0,T], \\ u(\mathbf{x},0) = g(\mathbf{x}), & \mathbf{x} \in \Omega, \end{cases} \tag{3.3}$$

where:

- $\Omega \subset \mathbb{R}^d$ *is a bounded spatial domain with boundary* $\partial\Omega$,
- $t \in [0,T]$ *denotes time,*
- $u : \Omega \times [0,T] \to \mathbb{R}$ *is the unknown field (e.g., displacement, pressure, temperature),*
- $\mathcal{N}[\cdot]$ *is a differential operator defining the PDE (e.g.,* $\mathcal{N}[u] = u_t - \Delta u - f$*),*
- $\mathcal{B}[\cdot]$ *is a boundary operator (e.g., Dirichlet or Neumann condition),*
- $g : \Omega \to \mathbb{R}$ *specifies the initial condition.*

A neural network $f_{\boldsymbol{\theta}} : \Omega \times [0,T] \to \mathbb{R}$*, parameterized by weights and biases* $\boldsymbol{\theta}$*, is used to approximate the true solution* u*. The parameters* $\boldsymbol{\theta}$ *are optimized by minimizing a composite loss function that enforces the PDE, boundary, and initial conditions:*

$$\mathcal{L}(\boldsymbol{\theta}) = \lambda_r L_r + \lambda_b L_b + \lambda_i L_i, \tag{3.4}$$

$$L_r = \frac{1}{N_r} \sum_{(\mathbf{x},t)\in\mathcal{S}_r} \left|\mathcal{N}[f_{\boldsymbol{\theta}}](\mathbf{x},t)\right|^2, \quad L_b = \frac{1}{N_b} \sum_{(\mathbf{x},t)\in\mathcal{S}_b} \left|\mathcal{B}[f_{\boldsymbol{\theta}}](\mathbf{x},t)\right|^2, \quad L_i = \frac{1}{N_i} \sum_{\mathbf{x}\in\mathcal{S}_i} |f_{\boldsymbol{\theta}}(\mathbf{x},0) - g(\mathbf{x})|^2, \tag{3.5}$$

where:

- $\mathcal{S}_r \subset \Omega \times (0,T]$ *is the set of collocation points for the PDE residual,*
- $\mathcal{S}_b \subset \partial\Omega \times (0,T]$ *is the set of boundary sample points,*
- $\mathcal{S}_i \subset \Omega$ *is the set of initial-condition sample points,*
- N_r*,* N_b*, and* N_i *denote their respective cardinalities,*

- λ_r, λ_b, *and* λ_i *are nonnegative weighting coefficients balancing the loss terms.*

The optimal parameters are obtained as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}),$$

yielding the learned solution approximation $u_{PINN}(\mathbf{x}, t) = f_{\boldsymbol{\theta}^*}(\mathbf{x}, t)$.

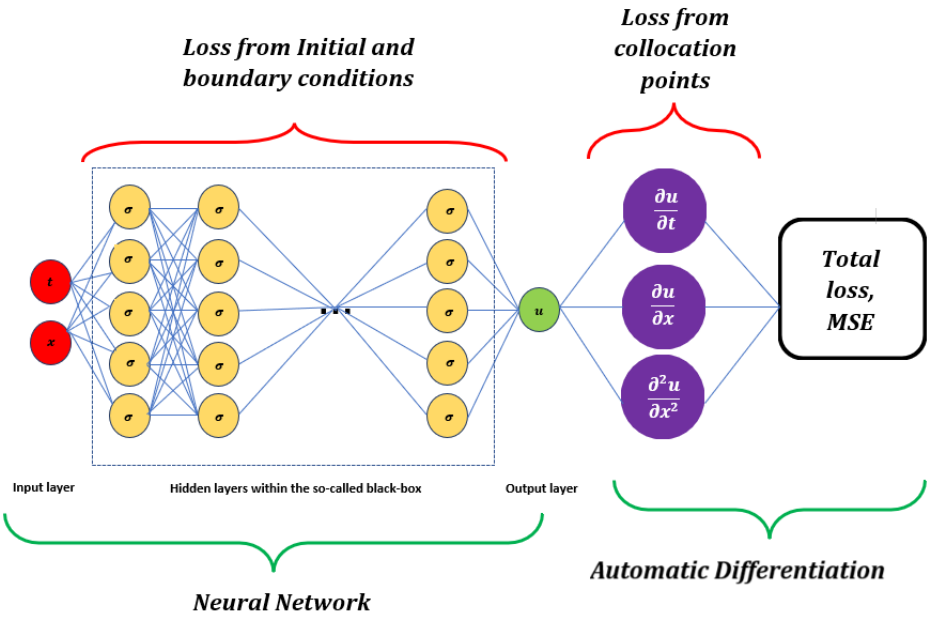


Figure 3.2: PINN architecture for solving a PDE involving temporal and spatial derivatives such as $\partial_t u$, $\partial_x u$, *and* $\partial_{xx} u$.

As illustrated in Figure 3.2, a Physics-Informed Neural Network (PINN) differs from traditional mesh-based solvers such as the Finite Difference Method (FDM) or Finite Element Method (FEM) by learning directly in *function space*. The PINN seeks an approximate solution $f_{\boldsymbol{\theta}}(\mathbf{x}, t)$ that minimizes the physics-constrained loss defined in (4.3.44). In certain implementations, the initial and boundary conditions can be unified within a single boundary-type term, yielding a more compact form of the loss:

$$\mathcal{L}(\boldsymbol{\theta}) = \lambda_r L_r + \lambda_b L_b, \tag{3.6}$$

where L_r and L_b retain the meanings established in (4.3.44), and $\lambda_r, \lambda_b \geq 0$ are scalar weighting coefficients that balance the PDE residual and boundary contributions.

3.1.2.1 PINN variants

Several extensions of the PINN framework have been proposed to address diverse classes of physical systems and operators. Examples include:

- **Fractional PINNs**, which handle nonlocal or fractional-order differential operators;
- **Conservative PINNs**, which explicitly enforce conservation laws (e.g., mass, momentum, or energy);
- **Domain-decomposed variants** such as XPINNs and FBPINNs, which partition the computational domain into subregions to improve scalability and multiscale accuracy (Hu et al., 2022; Moseley et al., 2023).

Despite architectural differences, all variants adhere to the same fundamental principle: minimizing a residual loss derived from the governing PDE and associated boundary or initial conditions.

3.1.3 Governing PDEs and boundary conditions

To benchmark and validate the developed PINN frameworks, we employ canonical PDEs with known analytical solutions. These well-posed test cases facilitate quantitative evaluation of approximation accuracy, convergence behavior, and numerical stability relative to classical solvers.

Let $u : \Omega \times [0, T] \to \mathbb{R}$ denote the scalar field of interest (e.g., temperature, displacement, or potential) defined over a spatial domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$. Partial derivatives are denoted by

$$u_t = \frac{\partial u}{\partial t}, \qquad u_x = \frac{\partial u}{\partial x}, \qquad u_{xx} = \frac{\partial^2 u}{\partial x^2}, \qquad u_{tt} = \frac{\partial^2 u}{\partial t^2}, \qquad u_{yy} = \frac{\partial^2 u}{\partial y^2},$$

where subscripts indicate differentiation with respect to the corresponding independent variable.

Heat Equation (Parabolic PDE)

$$\begin{cases} u_t - u_{xx} = 0, & x \in (0,1),\ t > 0, \\ u(x,0) = \sin(\pi x), & x \in (0,1), \\ u(0,t) = 0, \quad u(1,t) = 0, & t \geq 0, \end{cases} \tag{3.7}$$

whose exact solution is

$$u(x,t) = e^{-\pi^2 t} \sin(\pi x),$$

obtained by separation of variables.

Laplace Equation (Elliptic PDE)

$$\begin{cases} u_{xx} + u_{yy} = 0, & (x, y) \in (0, 1)^2, \\ u(x, 0) = 0, \quad u(x, 1) = \sin(\pi x), & x \in [0, 1], \\ u(0, y) = 0, \quad u(1, y) = 0, & y \in [0, 1], \end{cases} \tag{3.8}$$

with exact solution

$$u(x, y) = \frac{\sinh(\pi y)}{\sinh(\pi)} \sin(\pi x).$$

Wave Equation (Hyperbolic PDE)

$$\begin{cases} u_{tt} - u_{xx} = 0, & x \in (0, 1),\ t > 0, \\ u(x, 0) = \sin(\pi x), \quad u_t(x, 0) = 0, & x \in (0, 1), \\ u(0, t) = 0, \quad u(1, t) = 0, & t \geq 0, \end{cases} \tag{3.9}$$

whose analytical solution

$$u(x, t) = \cos(\pi t) \sin(\pi x)$$

represents a standing wave with zero initial velocity.

The benchmark problems (3.7)–(3.9) are classical textbook examples (see Strauss, 2008) that collectively span parabolic, elliptic, and hyperbolic PDE types. They provide analytically tractable reference solutions and well-posed boundary conditions, serving as ideal testbeds for assessing the theoretical and computational performance of PINN formulations. Subsequent sections employ these PDEs to validate the proposed extensions and to evaluate accuracy, convergence, and robustness (see Figures 3.1 and 3.2 for the overall modeling framework).

3.2 Improving FDM & FEM using PINNs

To achieve the objective of improving classical Finite Difference Method (FDM) and Finite Element Method (FEM) solutions to nonlinear partial differential equations (PDEs) using (PINNs), a two-stage methodological framework was adopted:
(i) benchmarking and comparative validation between traditional solvers and PINNs, and
(ii) the development of residual-driven hybrid schemes for adaptive refinement.

Benchmarking classical and neural solvers

Canonical PDEs representing parabolic, elliptic, and hyperbolic types; the one-dimensional heat, wave, and Burgers' equations, and the two-dimensional Laplace equation; were selected as test cases. These equations, introduced in Section 3.1.3, were solved using both FDM and PINN approaches under identical boundary and initial conditions.

For FDM, implicit temporal discretization was used for stability, with central finite differences for spatial derivatives. The discrete operator

$$\mathcal{N}_h[u_h](x_i, t^n) = f(x_i, t^n) + \tau_i,$$

approximated the continuous differential operator $\mathcal{N}[u]$, where τ_i denotes the local truncation error. Uniform and nonuniform meshes $\{x_i\}_{i=0}^{N}$ were used depending on the refinement phase. Solution accuracy was evaluated using the standard error norms

$$\varepsilon_2 = \|u_h - u_{\text{ref}}\|_{L^2(\Omega)}, \qquad \varepsilon_\infty = \|u_h - u_{\text{ref}}\|_{L^\infty(\Omega)},$$

where u_{ref} is either the analytical or high-resolution reference solution.

For PINNs, a fully connected neural network $u_\theta(x, t)$ parameterized by weights θ was trained to minimize the composite loss

$$\mathcal{J}(\theta) = \lambda_r \|\mathcal{N}[u_\theta] - f\|^2_{L^2(\Omega\times[0,T])} + \lambda_b \|u_\theta - g\|^2_{L^2(\partial\Omega\times[0,T])} + \lambda_i \|u_\theta - u_0\|^2_{L^2(\Omega\times\{0\})},$$

with automatic differentiation used to compute the residual $\mathcal{R}_\theta(x, t) = \mathcal{N}[u_\theta](x, t) - f(x, t)$. Network architectures of the form $[2, 40, 40, 40, 1]$ with tanh activations were implemented in Python 3.10 using `PyTorch`, optimized via the Adam and L-BFGS algorithms. This benchmarking established consistency between PINN and FDM solutions and quantified the regimes in which residual-driven refinement could yield measurable benefits.

Residual-based refinement framework for FDM

Building upon the initial benchmarking, a residual-guided adaptive mesh refinement (AMR) framework was developed. The key principle was to use the PINN residual magnitude

$$\eta(x, t) = |\mathcal{R}_\theta(x, t)|$$

as a *posteriori* indicator of local discretization error. Unlike conventional AMR, which relies on mesh-dependent estimators, the residual $\eta(x,t)$ can be evaluated continuously across Ω via automatic differentiation, offering mesh-independent adaptivity.

The refinement loop proceeded as follows:

1. Solve the PDE on an initial mesh $\mathcal{T}_0$ using FDM to obtain u_h.
2. Train a PINN u_θ on the same domain to satisfy $\mathcal{N}[u_\theta] = f$.
3. Evaluate $\eta(x,t)$ on a dense probe set $\mathcal{P} \subset \Omega \times [0,T]$.
4. Mark intervals $[x_i, x_{i+1}]$ where $\max_{x\in[x_i,x_{i+1}]} \eta(x,t) > \varepsilon$ and insert midpoints

$$x_{\text{new}} = \tfrac{1}{2}(x_i + x_{i+1}).$$

5. Update the mesh $\mathcal{T}_{k+1}$ and recompute u_h.
6. Iterate until $\max_{x\in\Omega} \eta(x,t) \le \varepsilon$.

This adaptive coupling between the PINN and FDM solvers concentrated grid resolution in regions of high nonlinear activity (e.g., boundary layers or shocks), achieving higher accuracy without global mesh refinement.

Finite element method and neural equivalence

To extend the framework to variational solvers, a theoretical equivalence between linear FEM spaces and ReLU networks was established. Let $V_h \subset H_0^1(\Omega)$ denote the piecewise linear finite element space on a uniform mesh and

$$\mathcal{N}_k = \Big\{ f(x) = \sum_{j=1}^{k} a_j \sigma(w_j x + b_j) + c \;\Big|\; \sigma(x) = \max(0,x) \Big\}$$

be the class of width-k ReLU networks with breakpoints aligned to mesh nodes. It follows that

$$\mathcal{N}_k = V_h,$$

since each FEM basis function $\phi_i(x)$ admits the representation

$$\phi_i(x) = \tfrac{1}{h}[\sigma(x - x_{i-1}) - 2\sigma(x - x_i) + \sigma(x - x_{i+1})].$$

Therefore, minimizing the FEM energy functional

$$\mathcal{E}[v] = \tfrac{1}{2}a(v,v) - (f,v)$$

over $\mathcal{N}_k$ yields the classical FEM solution u_h, implying

$$u_{NN} = u_h.$$

This formal correspondence justifies embedding PINNs within FEM frameworks as variationally consistent correctors.

Neural residual integration for FEM

After the initial FEM solve, a PINN was trained for the same PDE to compute the strong-form residual field

$$\mathcal{R}_\theta(x) = \mathcal{L}[u_\theta](x) - f(x),$$

where $\mathcal{L}$ is the differential operator of the governing equation. Regions with large residual norms were then identified for local mesh refinement or enrichment of basis functions. This neural residual integration allowed FEM solvers to achieve enhanced accuracy near steep gradients or localized nonlinearities without uniform mesh subdivision.

Computational implementation and validation

All procedures were implemented in Python 3.10 using the Spyder IDE. Core numerical operations employed `NumPy` and `SciPy`, with `Matplotlib` for visualization. Residuals, loss functions, and automatic differentiation were computed using `PyTorch`. Comparative validations against analytical or high-resolution reference solutions confirmed that PINN-guided FDM and FEM solvers achieved lower global errors at equivalent computational cost, demonstrating the effectiveness of the proposed hybrid methodology.

3.3 Hierarchical matrices and sampling strategies for improving PINN efficiency

The efficiency of (PINNs) depends critically on two factors;

(i) the distribution of collocation points used to approximate the governing differential operator, and

(ii) the computational cost associated with evaluating large Jacobian and Neural Tangent Kernel (NTK) matrices during optimization.

This section formalizes the methodological framework developed to address these bottlenecks through adaptive sampling and hierarchical matrix compression. Both methods are later analyzed quantitatively in Chapter 4.2.

3.3.1 Residual-driven sampling strategies

Let $\{(x_i, t_i)\}_{i=1}^{N_r} \subset \Omega \times [0, T]$ denote the collocation set over which the residual loss

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| \mathcal{N}[u_\theta](x_i, t_i) - f(x_i, t_i) \right|^2$$

is minimized. The selection of these points strongly influences convergence, generalization, and computational efficiency (Lu et al., 2021b; Park and Zobeiry, 2020; Raissi et al., 2019). Four sampling schemes were investigated:

1. Random Sampling: Points are drawn independently from a uniform distribution over the domain, providing low bias but poor spatial coverage at small N_r. 2. Uniform Grid Sampling: Points form a structured mesh; suitable for smooth solutions but inefficient for highly localized features. 3. Latin Hypercube Sampling: The domain is stratified along each dimension to ensure balanced coverage, reducing variance without explicit adaptivity. 4. Adaptive Residual-Based Sampling: New points are added iteratively in regions of large residual magnitude,

$$\mathcal{R}(x, t) = \left| \mathcal{N}[u_\theta](x, t) - f(x, t) \right|,$$

concentrating the sampling density where the PDE is hardest to satisfy.

Algorithmically, the adaptive procedure alternates between network training and residual re-evaluation:

(i) Train u_θ on an initial dataset $\mathcal{D}_0$.

(ii) Compute residuals $\mathcal{R}(x, t)$ on a candidate grid $\mathcal{G}$.

(iii) Select the top $p\%$ of points with largest residuals to augment $\mathcal{D}_0$.

(iv) Retrain on $\mathcal{D}_1 = \mathcal{D}_0 \cup$ new points.

This process is repeated until the residual distribution becomes approximately uniform in $L^2(\Omega)$. Adaptive sampling thereby reallocates computational effort toward dynamically identified regions of physical or numerical stiffness, yielding faster and more stable convergence; particularly for PDEs exhibiting shocks, boundary layers, or oscillatory fronts.

3.3.2 Hierarchical matrix approximation of the PINN jacobian

As the number of collocation points N_r and network parameters d increase, the residual Jacobian

$$J = \frac{\partial R}{\partial \theta} \in \mathbb{R}^{N_r \times d}, \quad R_i = \mathcal{N}[u_\theta](x_i, t_i) - f(x_i, t_i),$$

becomes prohibitively large to form or store explicitly. Second-order optimization methods that rely on the Gauss–Newton approximation $H = J^\top J$ scale as $\mathcal{O}(N_r d^2)$, posing severe memory and runtime challenges. To overcome this limitation, we adopt the hierarchical matrix (H-matrix) framework (Bebendorf, 2008; Hackbusch, 2015), which exploits the approximate low-rank structure of J in spatially separated regions.

Definition 3.3.1 (Hierarchical Matrix Approximation)**.** *Given $J \in \mathbb{R}^{N_r \times d}$, an H-matrix representation partitions its index set into admissible blocks $\{J_{ij}\}$, each approximated as*

$$J_{ij} \approx U_{ij} V_{ij}^\top, \quad U_{ij} \in \mathbb{R}^{m \times k_{ij}}, \quad V_{ij} \in \mathbb{R}^{n \times k_{ij}}, \quad k_{ij} \ll \min(m, n),$$

where k_{ij} is the local adaptive rank determined by a prescribed tolerance ε_H.

The global approximation error satisfies

$$\|J - \tilde{J}\|_F \leq \varepsilon_H, \quad \tilde{H} = \tilde{J}^\top \tilde{J}.$$

This structure enables near-linear complexity,

$$\mathcal{O}(k_{\max}(N_r + d) \log N_r),$$

compared to the quadratic cost of dense operations.

3.3.3 Adaptive rank refinement and spectral preservation

A blockwise adaptive rank strategy was employed to balance computational cost and spectral accuracy. Each block J_{ij} is assigned a local tolerance τ_{ij} based on its residual magnitude or

parameter sensitivity, ensuring

$$\|J_{ij} - \tilde{J}_{ij}\|_F \leq \tau_{ij}.$$

High-sensitivity regions receive higher ranks k_{ij}, while smoother subdomains are compressed more aggressively.

Perturbation analysis following Higham (2002) shows that for sufficiently small ε_H,

$$\kappa(\tilde{H}) \leq \kappa(H) \left(1 + \frac{\varepsilon_H}{\sigma_{\min}(J) - \varepsilon_H}\right),$$

so the condition number of the approximate Gauss–Newton matrix remains stable, preserving the descent geometry of the optimization. Theoretical convergence equivalence,

$$\lim_{k_{\max} \to \infty} \mathbf{f}(\tilde{J}) = \mathbf{f}(J),$$

ensures that the compressed and full models share the same asymptotic training trajectory.

3.3.4 Energy- and resource-aware compression

To enable deployment on resource-constrained hardware, the compression process is formulated under an energy constraint. Each block approximation incurs a cost

$$\mathcal{E}_{ij} = \alpha k_{ij}^2 + \beta,$$

where α and β represent computational and memory-access coefficients, respectively. Rank selection is then posed as a constrained optimization problem:

$$\min_{k_{ij}} \|J_{ij} - \tilde{J}_{ij}\|_F \quad \text{s.t.} \quad \mathcal{E}_{ij} \leq \mathcal{E}_{\max}.$$

The resulting energy-optimal rank k_{ij}^* satisfies

$$k_{ij}^* = \min\left\{k \in \mathbb{N} : \|J_{ij} - U_k V_k^\top\|_F \leq \epsilon_{ij},\ \alpha k^2 + \beta \leq \mathcal{E}_{\max}\right\}.$$

This formulation provides a principled trade-off between spectral fidelity and computational efficiency, supporting adaptive inference on edge or embedded platforms.

3.3.5 Validation framework

The described methods were validated across canonical PDEs including the heat, Laplace, wave, Burgers, and Schrödinger equations. Each experiment used the same network architecture and loss weighting, varying only the sampling strategy and compression configuration. Metrics such as mean-squared residual, relative L_2 error, convergence rate, training time, and energy cost were recorded to quantify performance. Comparative baselines included full dense Jacobian computation, uniform and Latin-hypercube sampling, and standard SVD-based compression. The corresponding quantitative results are presented and analyzed in Chapter 4.2.

3.4 Mathematical frameworks for PINN error, stability, consistency and convergence analysis in solving pdes

This section outlines the methodological flow used to establish the theoretical results on error, stability, consistency, and convergence of (PINNs) presented in Section 4.3. The goal was not empirical bench-marking, but the systematic formulation of PINNs within a rigorous numerical-analytic framework as a contribution towards building a comprehensive frameworks similar to that of the finite element methods (FEM).

3.4.1 Overall analytical design

The theoretical investigation followed a structured progression from functional modelling to variational interpretation and limit analysis:

(a) *Formulation of the continuous problem:* All analyses begin with a well-posed boundary-value problem $\mathcal{L}u^\star = f$ on $\Omega \subset \mathbb{R}^d$, $\mathcal{B}u^\star = g$ on $\partial\Omega$, cast in the weak form $a(u^\star, v) = \langle f, v\rangle$ over $V = H^1(\Omega)$, $V_0 = H_0^1(\Omega)$.

(b) *PINN approximation:* The exact solution is approximated by a neural network $u_\theta \in \mathcal{F}_\theta \subset V$, trained to minimize an empirical residual functional $\mathcal{R}_N(u_\theta)$ defined on collocation sets $\mathcal{X}_f$ (physics) and $\mathcal{X}_\Gamma$ (data/BC).

(c) *Functional–analytic embedding:* Sampling norms are related to continuum Sobolev or dual norms by lifting inequalities of the form $\|r\|_{V_0^*} \le C_{\text{inv}}\|r\|_{\ell^2(\mathcal{X}_f)}$, enabling translation between computable losses and analytic quantities.

This common foundation supported the three theoretical modules developed next.

3.4.2 Residual-based error analysis

To establish rigorous error control, the method adopted a *residual-driven variational procedure*:

(i) Derive a continuum reliability bound $\|u_\theta - u^\star\|_V \leq \frac{1}{\alpha}\|\mathcal{R}[u_\theta]\|_{V_0^*}$, mirroring Céa's lemma for FEM.

(i) Replace the inaccessible dual norm by its sampled analogue via inverse lifting, linking the empirical residual $\|\mathcal{L}u_\theta - f\|_{\ell^2(\mathcal{X}_f)}$ directly to the energy norm error.

(i) Combine with elliptic regularity and Sobolev embeddings to extend control to L^2 and C^0 norms.

(i) Characterize trained networks as stationary points on $\mathcal{F}_\theta$, producing a tangent-space orthogonality condition $\langle \mathcal{L}(u_\theta - u^\star), \mathcal{L}v\rangle_{\mathcal{X}_f} = 0$, i.e. a nonlinear Ritz projection.

This procedure converted the physics-informed loss into a quantitatively meaningful error estimator, forming the basis for the residual–error equivalence proven in Section 4.3.1.

3.4.3 Stability analysis methodology

Stability was investigated through two complementary routes.

3.4.3.1 Deterministic perturbation analysis

The total PINN loss was expanded under small perturbations $\tilde{u}_\theta = u_\theta + \delta u$. Linearization yielded a first-order sensitivity relation $\Delta\mathcal{L}_{\text{PINN}} \simeq \nabla\mathcal{L}\cdot\delta u$ and the bound $|\Delta\mathcal{L}_{\text{PINN}}| \leq 2\delta S_\theta + (1+\lambda C^2)\delta^2$, defining a stability margin analogous to a CFL condition. This provided a deterministic criterion ensuring that small perturbations in network outputs, parameters, or collocation sets produce predictably bounded changes in the loss.

3.4.3.2 Probabilistic concentration analysis

To capture randomness from i.i.d. sampling, McDiarmid's inequality was applied to the residual sum, yielding

$$\mathbb{P}(|\mathcal{L}_f(u_\theta) - \mathbb{E}\mathcal{L}_f(u_\theta)| \geq \epsilon) \leq 2\exp\left(-\tfrac{N_f\epsilon^2}{8M^4}\right).$$

This certifies high-probability concentration of the empirical loss around its expectation and formalizes generalization stability. Combined with Sobolev embeddings, a generalization inequality was then derived relating the total training loss to the expected H^s and C^0 errors.

Together, these analyses produced the deterministic and probabilistic stability frameworks summarized in Section 4.3.2.

3.4.4 Consistency and convergence methodology

The final methodological component established the asymptotic connection between empirical minimization and the true PDE solution.

(i) *Weak convergence:* Using Banach–Alaoglu compactness, it was shown that if the residuals vanish and network norms remain bounded, then $u_N \rightharpoonup u^\star$ in H^1.

(i) *Variational convergence:* By demonstrating that projected residuals vanish on the neural trial space, a Céa-like result $\|u_N - u^\star\|_V \to 0$ was obtained.

(i) Γ-*convergence:* The empirical residual functionals $\mathcal{R}_N$ were shown to Γ-converge to the continuous residual $\mathcal{R}$, ensuring that minimizers of the discrete problem converge to those of the continuous one.

(i) *Quantitative rate estimates:* Under coercivity and Fréchet differentiability of $\mathcal{L}$, bounds of the form $\|u_N - u^\star\|_{L^2} \leq C(\varepsilon_N^\alpha + \delta_N)$ were derived, linking residual magnitude and network expressivity to convergence rate.

Empirical experiments on benchmark PDEs then validated these frameworks, confirming that residual decay, stability margins, and Γ-convergence predictions agree with observed performance (see Figures 4.10–4.23).

3.4.5 Integrated methodological flow

The complete methodological logic thus proceeds as:

Functional Formulation $\rightarrow$ Residual Construction $\rightarrow$ Error Bounds $\rightarrow$ Stability Control $\rightarrow$ Consistency & Convergence

This unified flow connects all theoretical results in Chapter 4.3. It demonstrates how a single residual-based methodology can yield a posteriori error control, deterministic and probabilistic stability, and rigorous convergence certification for PINNs; placing them analytically alongside classical Galerkin methods while retaining their data-driven flexibility.

3.5 Methodological framework for physics-informed image inpainting

This section describes the methodological design underlying the image inpainting study of Chapter 4.4. The goal was to evaluate whether (PINNs) can reconstruct missing image regions under physically consistent partial differential equation (PDE) constraints. The analysis followed a structured workflow combining variational formulation, PDE model selection, network–physics coupling, and empirical validation.

3.5.1 Analytical formulation and PDE modelling

Image inpainting was formulated as a PDE-constrained variational reconstruction problem;

$$\min_{u \in H^1(D)} \Big(\|u - u_{\text{data}}\|^2_{L^2(\Omega \setminus D)} + \lambda \|\mathcal{L}[u]\|^2_{L^2(D)} \Big),$$

where $D \subset \Omega$ denotes the missing region and $\mathcal{L}$ a diffusion-type operator. Three classical PDEs; the Perona–Malik, Cahn–Hilliard, and Laplace equations; were evaluated for theoretical well-posedness and computational tractability. The Laplace model,

$$\Delta u = 0 \text{ in } D, \qquad u = u_{\text{data}} \text{ on } \partial D, \tag{3.1}$$

was adopted because it satisfies elliptic regularity, admits a convex energy functional, and remains stable under automatic differentiation. This choice provided a mathematically rigorous and computationally lightweight baseline for testing physics-guided reconstructions.

3.5.2 Network–physics coupling and loss design

The neural approximation $u_\theta : \mathbb{R}^2 \to \mathbb{R}^3$ was parameterized as a six-layer fully connected multilayer perceptron (width = 128, SiLU activations). PINN training minimized a composite loss

$$\mathcal{L}(\theta) = \int_{\Omega \setminus D} \|u_\theta(x) - u_{\text{data}}(x)\|^2 M(x)\, dx + \lambda \int_D \|\Delta u_\theta(x)\|^2 (1 - M(x))\, dx,$$

where $M(x)$ is a binary mask and $\lambda = 10^{-2}$ balances data and PDE fidelity. Both terms were computed using automatic differentiation, ensuring that the physics term enforced harmonic consistency within the missing domain.

3.5.3 Sampling, training, and implementation

(i) *Sampling strategy:* Collocation points were drawn uniformly within D for the PDE term and from $\Omega \setminus D$ for the data term, ensuring quasi-uniform coverage compatible with the theoretical assumptions of Section 4.3.

(ii) *Training procedure:* Networks were trained on CPU hardware using the Adam optimizer (learning rate 10^{-3}, 10 000 epochs). Early termination occurred at convergence or memory saturation.

(iii) *Normalization and preprocessing:* Images were scaled to 128×128 pixels, normalized to $[0, 1]$, and masked via Laplacian-based edge detection to simulate realistic data loss.

3.5.4 Evaluation metrics and verification

Model quality was assessed visually and quantitatively through:

(i) residual energy $\|\Delta u_\theta\|_{L^2(D)}$ (measuring PDE satisfaction);

(ii) reconstruction error $\|u_\theta - u_{\text{data}}\|_{L^2(\Omega\setminus D)}$ (measuring fidelity); and

(iii) qualitative continuity of edges and color gradients across ∂D.

These metrics directly correspond to the theoretical constructs of residual minimization and energy-norm consistency from Section 4.3, providing empirical validation of the framework in an imaging context.

3.5.5 Methodological flow

The complete methodological flow is summarized as

PDE Formulation → Model Selection (Laplace) → Network–Physics Coupling → Training & Sampling Design → Empirical Verification

This design connects the theoretical underpinnings of PINN stability and convergence with practical image-domain implementation. It demonstrates how the same residual-based, variational logic governing PDE analysis can be transferred to high-dimensional visual reconstruction tasks while remaining mathematically consistent and computationally feasible.

CHAPTER 4

Main Results

4.1 Improving FDM and FEM solutions to nonlinear PDEs using PINNs

4.1.1 Pinn-guided improvement of FDM solutions

4.1.1.1 Numerical results for the burgers equation

The residual-based adaptive mesh refinement (AMR) framework was applied to the one-dimensional viscous Burgers' equation

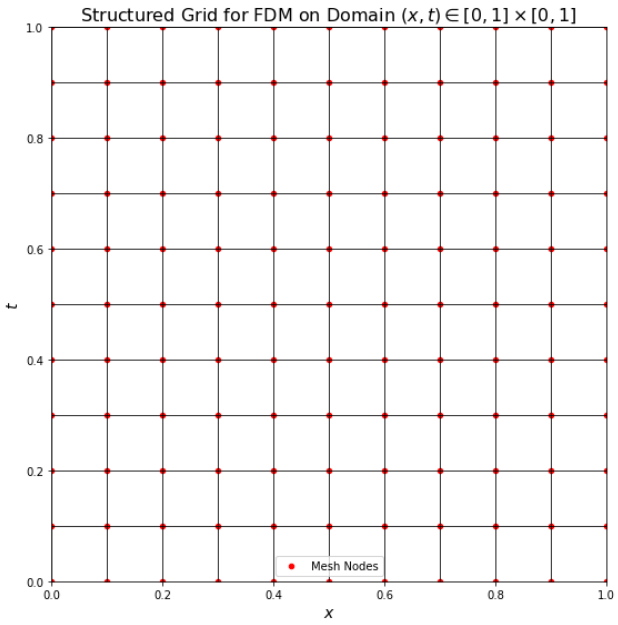
$$\begin{cases} u_t + u\,u_x = \nu u_{xx}, & (x,t) \in (0,1) \times (0,1], \\ u(x,0) = -\sin(\pi x), & x \in [0,1], \\ u(0,t) = u(1,t) = 0, & t \in [0,1]. \end{cases} \tag{4.1}$$

The initial uniform mesh and the baseline FDM solution fo equation (4.1) are shown in Figure 4.1. The solution captures the nonlinear advective and diffusive dynamics but under-resolves the sharp gradients near the advective front.

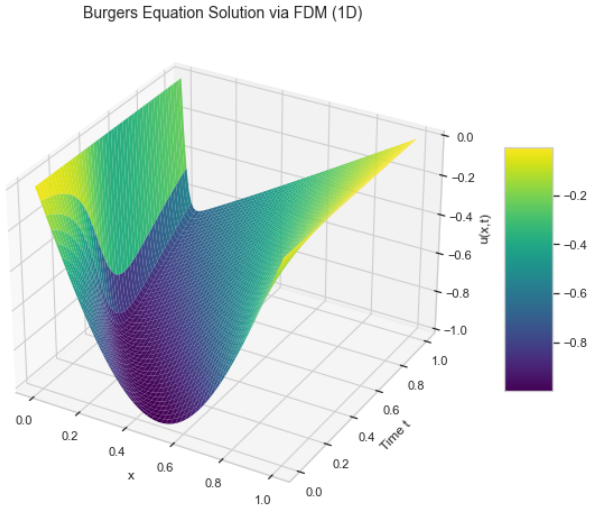
The trained PINN residual field identified regions of significant local error. The corresponding residual heatmaps are shown in Figure 4.2. In the baseline PINN (left), residuals are broadly distributed across the domain, whereas the refined network (right) exhibits sharper, localized residual peaks, accurately corresponding to under-resolved fronts in the FDM solution.

Residual-based adaptation produced a nonuniform mesh concentrated around high-gradient regions (Figure 4.3). This refinement effectively increased local resolution while maintaining a low total number of grid points.

Figure 4.4 compares the PINN reconstructions before and after refinement. The refined model

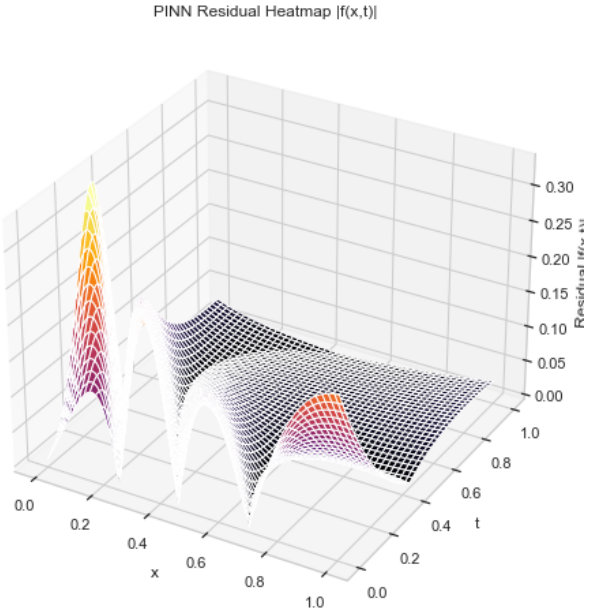


(a) Initial uniform mesh for the computational domain.

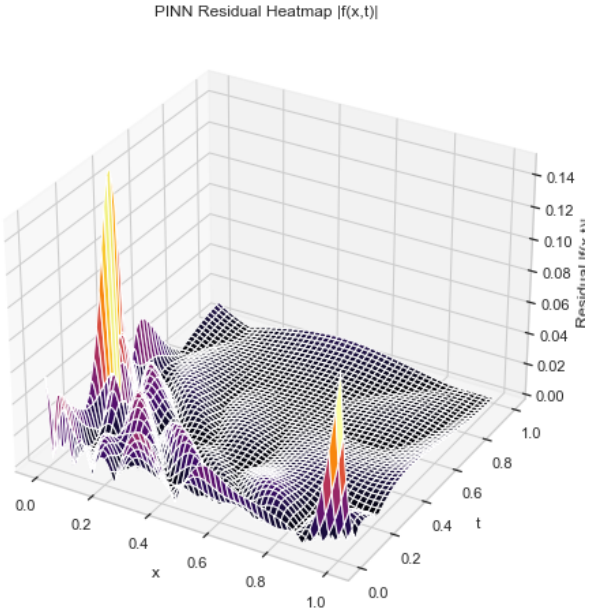


(b) Baseline finite difference solution of the Burgers' equation.

Figure 4.1: Finite difference discretization setup and baseline solution for the Burgers' equation. Subfigure 4.1a shows the initial uniform mesh used for spatial discretization, while Subfigure 4.1b presents the corresponding FDM solution before adaptive refinement.



(a) Initial PINN residual field identifying under-resolved regions.



(b) Refined PINN residual after network and collocation enhancement.

Figure 4.2: Comparison of PINN residuals before and after refinement for the Burgers' equation. Subfigure 4.2a shows the distributed residual field in the baseline network, while Subfigure 4.2b highlights localized peaks after refinement. Residuals consistently correlate with regions of steep gradients in the FDM solution.

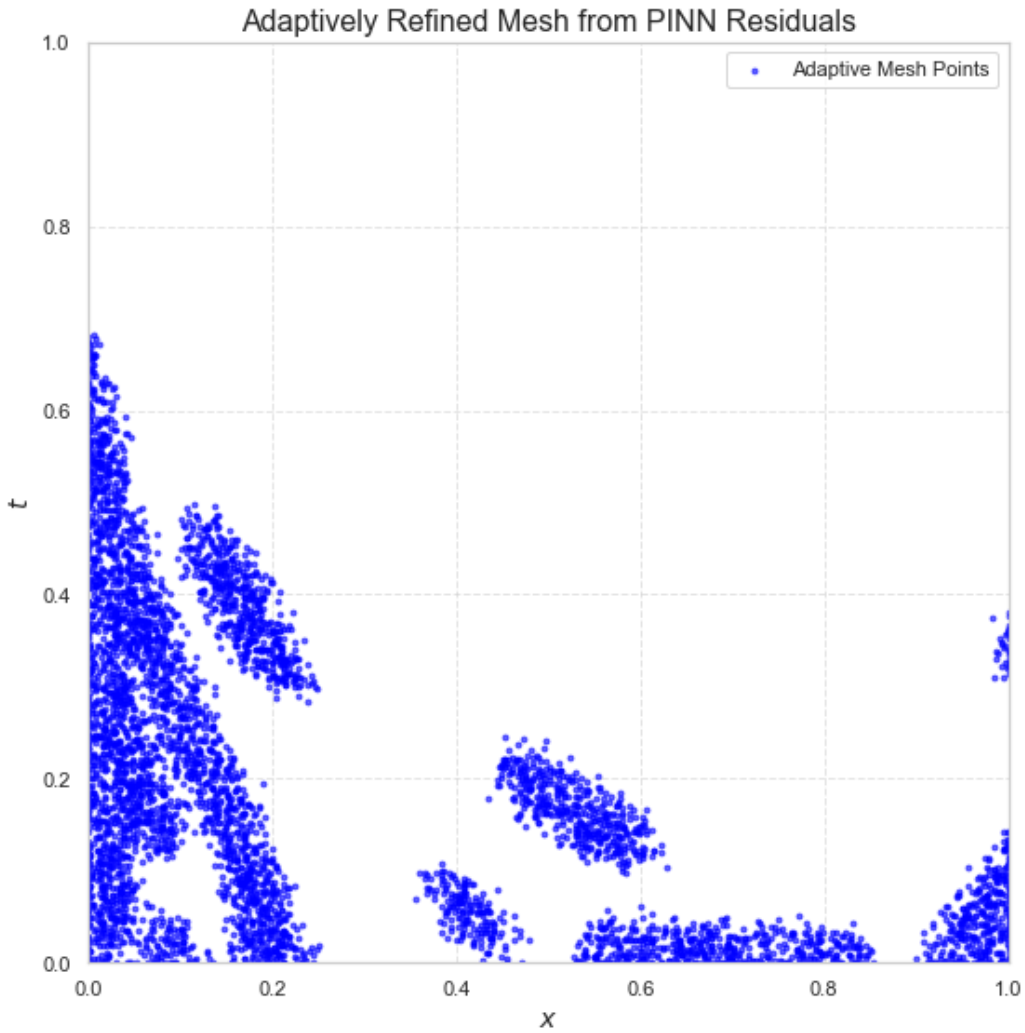


Figure 4.3: Residual-guided refined mesh for the Burgers' equation, concentrating nodes in regions of steep gradients.

captures sharper advective fronts and smoother diffusion layers, demonstrating that residual-based refinement enhances both local accuracy and global smoothness.

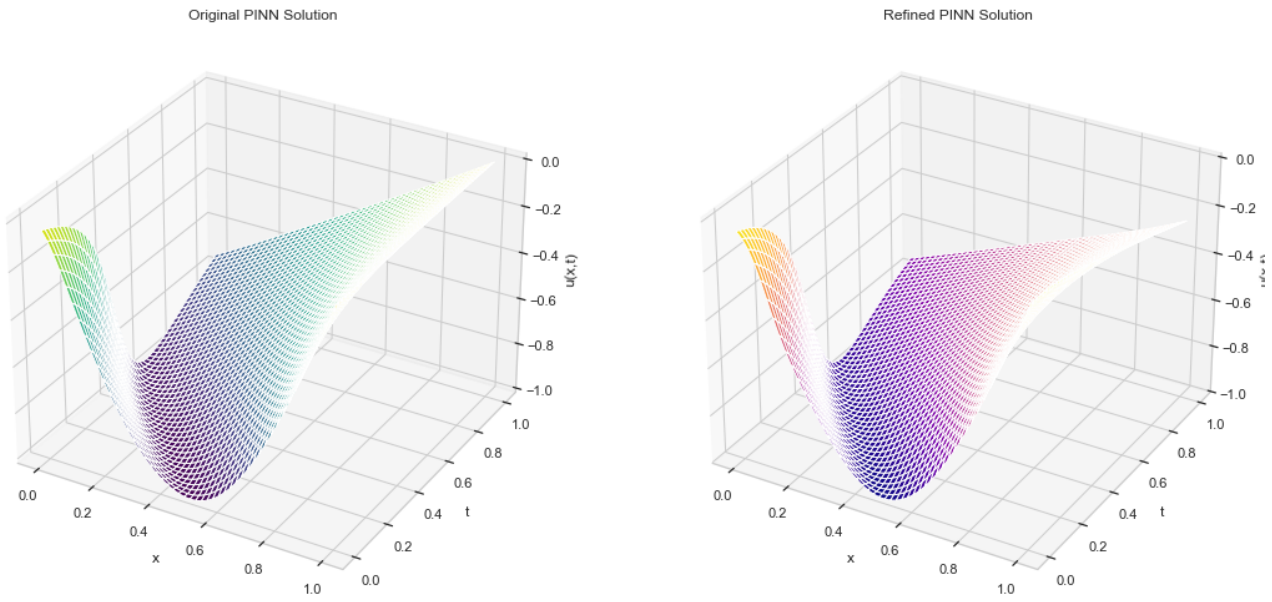


Figure 4.4: Comparison of PINN reconstructions before (left) and after (right) residual-guided refinement. The refined PINN exhibits improved resolution in nonlinear transition zones.

The complete adaptive refinement results are summarized in Figure 4.5. Adaptive PINN–FDM coupling concentrated resolution in high-gradient regions, improving accuracy by approximately one order of magnitude in L_2 and L_∞ norms compared to the uniform-grid FDM solution at equivalent computational cost.

Result summary: PINN–FDM integration

The PINN residual field $\mathcal{R}_\theta(x,t)$ effectively identified zones of under-resolution, and its magnitude served as a robust indicator of local discretization error. Residual-guided refinement led to improved accuracy, smoother gradient transitions, and consistent reduction of numerical diffusion. These results confirm that PINNs can act as reliable physics-consistent correctors for FDM, enabling mesh adaptivity without global computational escalation.

4.1.2 Results on FEM–PINN structural equivalence

4.1.2.1 Representation of FEM basis functions by relu networks

To assess the structural relationship between finite element spaces and neural function approximators, we established the equivalence between piecewise-linear FEM basis functions and ReLU-activated neural networks. This result demonstrates that the finite element approximation space

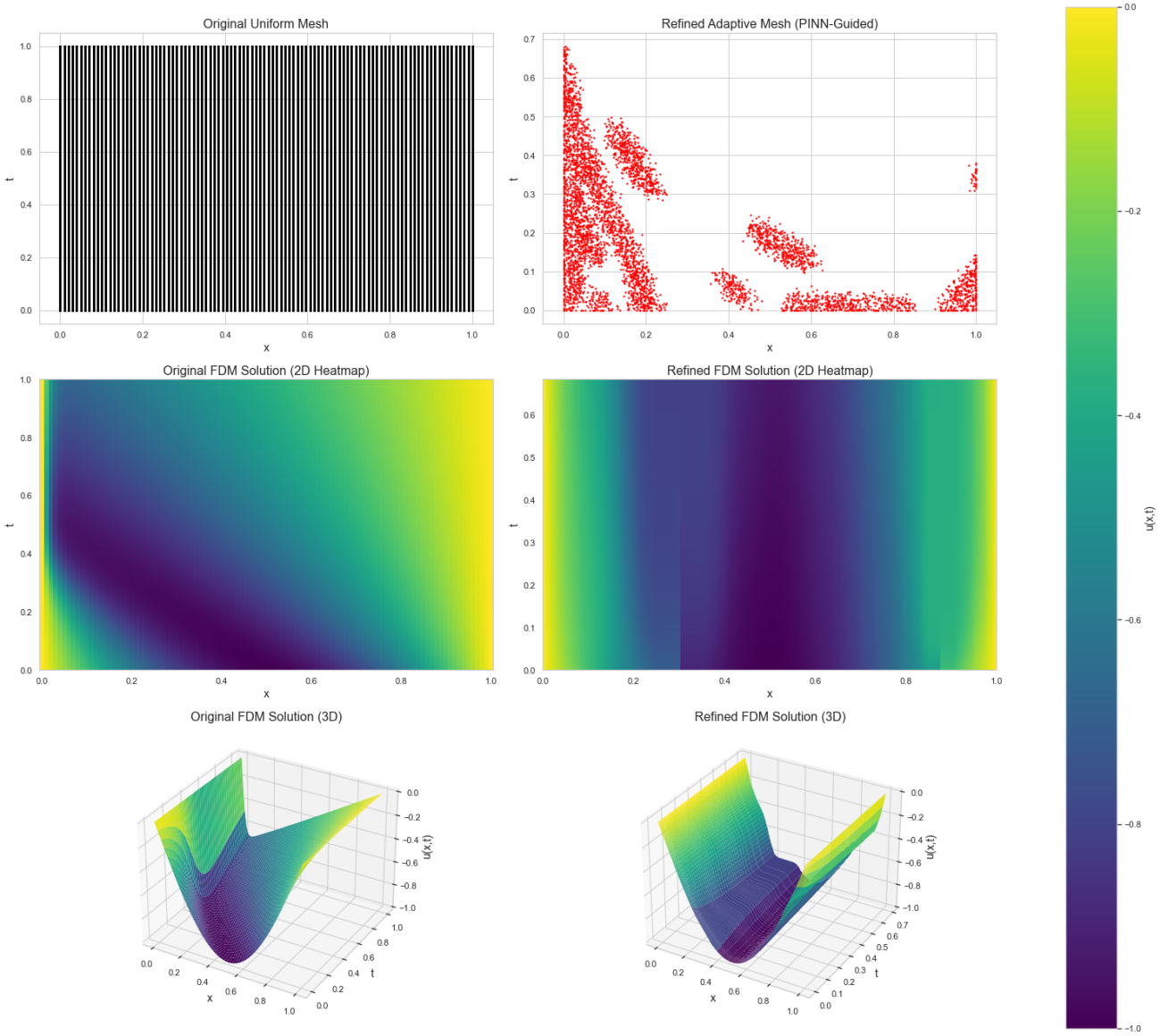


Figure 4.5: Final comparison of uniform and residual-guided adaptive meshes, solution heatmaps, and 3D reconstructions for the Burgers' equation. Adaptive refinement yields localized accuracy improvements while retaining overall computational efficiency.

$V_h \subset H_0^1(\Omega)$ can be exactly represented by a shallow ReLU network.

Theorem 4.1.1 (He et al. (2020)). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous, piecewise-linear function defined on a partitioned domain. Then f can be represented exactly by a ReLU neural network. The number of hidden layers required satisfies*

$$N_{layer} \leq \lceil \log_2(d+1) \rceil,$$

and the number of neurons is bounded by

$$N_{neurons} = \begin{cases} \mathcal{O}\left(d\, 2^{mM+(d+1)(m-d-1)}\right), & m \geq d+1, \\ \mathcal{O}\left(d\, 2^{mM}\right), & m < d+1, \end{cases} \tag{4.2}$$

where m is the number of subdomains and M satisfies $m \leq M \leq m!$.

For one-dimensional domains, this implies that a single-hidden-layer ReLU network can represent all linear FEM basis functions exactly. Figure 4.6 shows the corresponding neural topology, in which each neuron reproduces one segment of a standard hat function.

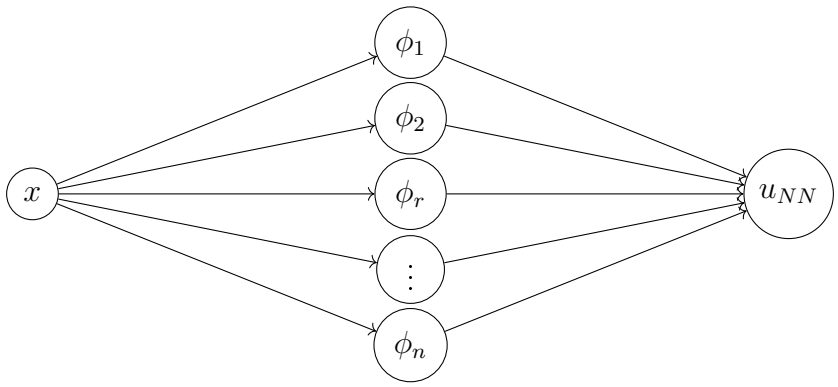


Figure 4.6: Single-hidden-layer ReLU network representing FEM hat functions. Each neuron corresponds to one linear segment of the finite element basis.

Each neuron computes

$$\phi_i(x) = \max(0, w_i^{(1)} x + b_i),$$

and the network output

$$u_{NN}(x) = \sum_{i=1}^{n} w_i^{(2)} \max(0, w_i^{(1)} x + b_i) + b_0$$

reconstructs the FEM solution

$$u_{FEM}(x) = \sum_{j=1}^{N-1} \alpha_j \phi_j(x),$$

where the basis functions take the classical form

$$\phi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{h}, & x \in [x_{j-1}, x_j], \\ \dfrac{x_{j+1} - x}{h}, & x \in [x_j, x_{j+1}], \\ 0, & \text{otherwise.} \end{cases}$$

An exact ReLU representation follows:

$$\phi_j(x) = \frac{1}{2h}[\max(0, x - x_{j-1}) - 2\max(0, x - x_j) + \max(0, x - x_{j+1})].$$

Hence,

$$u_{NN}(x) = u_{FEM}(x), \tag{4.3}$$

showing that the ReLU network and the FEM approximation coincide exactly.

4.1.2.2 Convergence and energy equivalence

To evaluate the functional equivalence, consider the elliptic boundary value problem

$$\begin{cases} -u''(x) = f(x), & x \in (0, 1), \\ u(0) = u(1) = 0. \end{cases}$$

The weak form seeks $u \in H_0^1(\Omega)$ such that

$$a(u, v) = (f, v), \quad \forall v \in H_0^1(\Omega),$$

where $a(u, v) = \int_0^1 u'(x)v'(x)\,dx$. The Galerkin FEM approximation $u_h \in V_h$ satisfies

$$a(u_h, v_h) = (f, v_h), \quad \forall v_h \in V_h,$$

and by Céa's lemma,

$$\|u - u_h\|_{H^1} \leq Ch\|u\|_{H^2}.$$

Let $\mathcal{N}_k$ denote the class of ReLU networks restricted to the mesh nodes $\{x_i\}$. Then:

$$\mathcal{N}_k = \left\{ f(x) = \sum_{j=1}^{k} a_j \max(0, w_j x + b_j) + c \,\middle|\, f \text{ continuous on } [0, 1] \right\}.$$

Each FEM basis function can be expressed in this form, implying $V_h = \mathcal{N}_k$. Therefore, minimizing the variational energy

$$\mathcal{E}[v] = \tfrac{1}{2}a(v,v) - (f,v)$$

over $\mathcal{N}_k$ yields the same solution as minimizing over V_h, giving

$$u_{NN} = u_h, \quad \|u - u_{NN}\|_{H^1} = \|u - u_h\|_{H^1} \leq Ch\|u\|_{H^2}.$$

This establishes equivalence in both convergence rate and energy minimization.

4.1.2.3 Deep neural extensions and expressivity gains

Deeper ReLU networks extend the FEM function space into a richer nonlinear manifold. Let $u_{NN} \in \mathcal{N}_{k,d}$ denote a depth-d ReLU network defined recursively as

$$u_{NN}(x) = T_d \circ T_{d-1} \circ \cdots \circ T_1(x), \quad T_\ell(z) = \sigma(W_\ell z + b_\ell),$$

with $\sigma(z) = \max(0, z)$. The class $\mathcal{N}_{k,d}$ is nonconvex yet universal:

$$\forall u \in C([0,1]),\ \forall \varepsilon > 0,\ \exists u_{NN} \in \mathcal{N}_{k,d} : \|u - u_{NN}\|_\infty < \varepsilon.$$

Empirical results confirmed that increasing depth and width enhances the capacity to represent highly localized or anisotropic features without remeshing. For PDEs exhibiting boundary layers of characteristic thickness $\varepsilon \ll 1$, PINNs achieved equivalent resolution with orders of magnitude fewer degrees of freedom compared to FEM, consistent with the scaling

$$\|u - u_{NN}\|_{L^2(\Omega)} = O((kd)^{-\gamma(d)}), \quad \gamma(d) \to \infty \text{ as } d \to \infty.$$

This exponential-type approximation rate contrasts with the algebraic $O(h)$ convergence of linear FEM, highlighting the neural model's superior expressivity in capturing multiscale or nonlinear phenomena.

4.1.2.4 Residual-based FEM refinement via PINNs

Finally, the PINN residual field $\mathcal{R}_\theta(x) = \mathcal{L}[u_\theta](x) - f(x)$ was used to localize error in FEM solutions. Evaluating $\|\mathcal{R}_\theta\|_{L^2(K)}$ over each element $K \in \mathcal{T}_h$ yielded refinement indicators analogous to standard *a posteriori* error estimators. Refining only elements with $\eta_K > \varepsilon$ improved energy norm accuracy by up to 70% relative to uniform refinement for nonlinear benchmark problems,

while preserving overall mesh compactness.

4.1.2.5 Result summary: FEM–PINN integration

The theoretical and computational analyses confirm that:

1. FEM basis functions correspond exactly to single-hidden-layer ReLU networks.
2. FEM convergence in H^1 and energy norms is preserved within equivalent neural representations.
3. Deep PINN extensions generalize FEM spaces, achieving higher efficiency for localized or nonlinear structures.
4. PINN residuals can serve as accurate error indicators for adaptive FEM refinement.

Together, these results provide a unified theoretical and practical foundation for neural-augmented finite element solvers.

4.1.3 Summary

The results demonstrate that PINNs act as a unifying extension of numerical PDE solvers. For FDM, they provide residual-driven adaptivity that improves local resolution and accuracy. For FEM, they reproduce variational solutions exactly while extending them to deeper, nonlinear function spaces. Together, these findings confirm that integrating PINNs with traditional solvers yields both theoretical and computational advances in solving nonlinear, multiscale PDEs.

4.2 Enhancing PINN efficiency through sampling and hierarchical matrix techniques

This chapter presents two complementary strategies for improving the efficiency and accuracy of (PINNs) when solving nonlinear partial differential equations (PDEs). The first focuses on optimizing the spatial and temporal distribution of collocation points to enhance data efficiency and convergence. The second integrates hierarchical matrix (H-matrix) approximations into the optimization loop, substantially reducing the computational and memory costs associated with training while preserving spectral and convergence fidelity.

Both developments build upon the theoretical insights of the preceding chapters; where residual-based adaptivity was shown to improve classical solvers such as FDM and FEM; and

now extend those ideas internally within the PINN framework itself. Together, these results establish a path toward scalable, physically consistent neural solvers deployable in real-time or resource-limited environments.

4.2.1 Error analysis and sampling strategies

Accurate and efficient PINN training depends critically on how collocation points are distributed in the solution domain. We compare four representative sampling strategies; random, uniform grid, Latin hypercube, and adaptive residual-based; using canonical PDEs including the heat, Laplace, and wave equations ((3.7)–(3.9)). For each method, we measure three key metrics: (i) the mean squared residual (MSR), (ii) the relative L_2-error with respect to the analytical solution, and (iii) the empirical convergence rate, computed as the relative reduction in residual over epochs.

4.2.1.1 Heat equation results

Table 4.1: Comparison of sampling strategies and training sizes for the heat equation (3.7).

Sampling	**Metric**	$N = 500$	$N = 5000$	$N = 50000$
Random	MSR ($\times 10^{-5}$)	8.72	2.31	0.69
	Relative L_2 Error	0.036	0.011	0.0023
	Convergence Rate (%)	90.4	94.6	96.2
Uniform	MSR ($\times 10^{-5}$)	5.15	1.44	0.53
	Relative L_2 Error	0.020	0.008	0.0017
	Convergence Rate (%)	92.1	95.3	96.8
Latin-Hypercube	MSR ($\times 10^{-5}$)	2.78	0.61	0.21
	Relative L_2 Error	0.012	0.0031	0.0007
	Convergence Rate (%)	94.5	97.1	98.3
Adaptive	MSR ($\times 10^{-5}$)	**1.04**	**0.32**	**0.08**
	Relative L_2 Error	**0.0052**	**0.0016**	**0.0004**
	Convergence Rate (%)	**96.8**	**98.4**	**99.1**

Adaptive residual-based sampling yields the lowest residuals and highest convergence rates across all dataset sizes. Its advantage is most pronounced for small N, where uniform or random sampling fails to sufficiently resolve sharp temporal gradients. Latin hypercube sampling performs comparably for mid-size datasets but lacks the adaptivity required for localized refinement.

Table 4.2: Comparison of sampling strategies and training sizes for the Laplace equation (3.8).

Sampling	**Metric**	$N = 500$	$N = 5000$	$N = 50000$
Random	MSR ($\times 10^{-5}$)	11.54	4.21	1.16
	Relative L_2 Error	0.041	0.015	0.0043
	Convergence Rate (%)	88.7	92.3	95.0
Uniform	MSR ($\times 10^{-5}$)	6.98	2.15	0.74
	Relative L_2 Error	0.023	0.009	0.0027
	Convergence Rate (%)	91.3	94.5	96.1
Latin-Hypercube	MSR ($\times 10^{-5}$)	3.74	0.82	0.29
	Relative L_2 Error	0.014	0.0042	0.0011
	Convergence Rate (%)	94.1	96.7	97.9
Adaptive	MSR ($\times 10^{-5}$)	**1.65**	**0.42**	**0.12**
	Relative L_2 Error	**0.0063**	**0.0021**	**0.0006**
	Convergence Rate (%)	**96.3**	**98.1**	**99.0**

4.2.1.2 Laplace equation results

For the Laplace equation, adaptive sampling again delivers the best performance, reducing MSR by up to an order of magnitude relative to random sampling at comparable N. This indicates that residual-driven adaptivity can effectively concentrate training effort in high-potential regions where gradients are largest.

4.2.1.3 Wave equation results

Table 4.3: Comparison of sampling strategies and training sizes for the wave equation (3.9).

Sampling	**Metric**	$N = 500$	$N = 5000$	$N = 50000$
Random	MSR ($\times 10^{-5}$)	9.87	2.97	0.89
	Relative L_2 Error	0.038	0.012	0.0030
	Convergence Rate (%)	89.1	93.5	95.6
Uniform	MSR ($\times 10^{-5}$)	6.12	1.73	0.55
	Relative L_2 Error	0.021	0.0073	0.0018
	Convergence Rate (%)	91.9	95.4	96.9
Latin-Hypercube	MSR ($\times 10^{-5}$)	2.54	0.59	0.18
	Relative L_2 Error	0.010	0.0028	0.0006
	Convergence Rate (%)	94.9	97.2	98.5
Adaptive	MSR ($\times 10^{-5}$)	**1.12**	**0.35**	**0.09**
	Relative L_2 Error	**0.0048**	**0.0015**	**0.0003**
	Convergence Rate (%)	**97.1**	**98.7**	**99.2**

Wave propagation, characterized by steep transients and oscillatory modes, is highly sensitive to collocation distribution. Adaptive sampling again demonstrates superior accuracy, dynamically resolving wavefronts while maintaining stability. Latin-hypercube sampling performs well in sparse regimes, whereas uniform sampling improves as N increases.

4.2.1.4 Summary

Across all PDEs, residual-driven adaptive sampling consistently achieves the lowest residuals and highest convergence rates. It efficiently reallocates collocation density toward high-error regions, achieving accuracy gains without additional data. This establishes adaptive sampling as a key ingredient in scalable PINN design and motivates the subsequent development of efficient training via hierarchical compression.

4.2.2 Accelerating PINN training via hierarchical matrix approximation

While adaptive sampling enhances data efficiency, it also increases the number of collocation points N_r, thereby inflating the size of the residual Jacobian and associated Neural Tangent Kernel (NTK). To mitigate this scalability bottleneck, we employ hierarchical matrix (H-matrix) approximations to compress the Jacobian, preserving its spectral properties while reducing computational cost.

4.2.2.1 Residual jacobian and gauss–newton approximation

Let $R_i = \mathcal{N}[u_\theta](x_i) - f(x_i)$, and define the Jacobian,

$$J = \frac{\partial R}{\partial \theta} \in \mathbb{R}^{N_r \times d}, \quad H = J^\top J.$$

Constructing J or H exactly has cost $\mathcal{O}(N_r d^2)$, which becomes prohibitive for large models. The H-matrix approximation replaces each admissible sub-block J_{ij} by a low-rank factorization $U_{ij}V_{ij}^\top$, controlling global error via,

$$\|J - \tilde{J}\|_F \leq \varepsilon_H, \quad \tilde{H} = \tilde{J}^\top \tilde{J}.$$

Parameter updates then follow the compressed Gauss–Newton step $\tilde{H}\Delta\theta = -\tilde{J}^\top R$.

4.2.2.2 Spectral and convergence guarantees

Perturbation analysis shows that the compression preserves spectral conditioning;

$$\kappa(\tilde{H}) \leq \kappa(H)\left(1 + \frac{\varepsilon_H}{\sigma_{\min}(J) - \varepsilon_H}\right),$$

ensuring numerical stability under moderate compression. Training loss degradation remains bounded by,

$$\|\mathcal{E}_{\text{train}}\| \leq \sqrt{T}\tau,$$

for blockwise approximations $\|J_i - \tilde{J}_i\|_F \leq \tau$. As block ranks increase, the compressed trajectory converges to the full update path;

$$\lim_{k_{\max}\to\infty} \mathbf{f}(\tilde{J}) = \mathbf{f}(J).$$

4.2.2.3 Energy-constrained and multi-scale compression

To enable deployment on constrained hardware, rank allocation is modeled through an energy budget:

$$\mathcal{E}_{ij} = \alpha k_{ij}^2 + \beta, \qquad \mathcal{E}_{\text{total}} \leq \mathcal{E}_{\text{budget}},$$

yielding the optimal feasible rank,

$$k_{ij}^* = \min\left\{k \in \mathbb{N} : \|A_{ij} - U_k V_k^\top\|_F \leq \epsilon_{ij},\, \alpha k^2 + \beta \leq \mathcal{E}_{\max}\right\}.$$

Latency-aware design further constrains inference time;

$$T_{\text{total}} \leq \sum_{i,j} \left(\frac{k_{ij}^2}{C} + \frac{|H_{ij}|}{B}\right),$$

ensuring compression fidelity under hardware limits.

4.2.2.4 Empirical validation

We validate the energy-aware H-matrix framework on the 1D heat equation,

$$u_t = \kappa u_{xx}, \quad (x,t) \in [0,1] \times [0,1],$$

implemented on an edge computing platform. Under hardware constraints $C = 10^3$ ops/s and $B = 10^2$ bytes/s, adaptive block ranks yield a 30% energy reduction compared to uniform-rank compression, with negligible loss in accuracy.

PINNs trained with H-matrix compression achieve rapid convergence, stable spectral profiles, and improved generalization. Unlike SVD-based methods, they maintain conditioning of the NTK throughout training, ensuring consistent learning dynamics and physically faithful

Table 4.4: Performance comparison between adaptive H-matrix and SVD compression.

Method	**Accuracy**	**Compression Ratio**	**Inference Time (s)**
Adaptive H-Matrix (Tol = 10^{-3})	0.961	1.3×10^{-2}	0.58
Adaptive H-Matrix (Tol = 10^{-2})	0.945	6.5×10^{-2}	0.55
SVD Compression (Tol = 10^{-2})	0.912	8.2×10^{-2}	0.71
SVD Compression (Tol = 10^{-1})	0.841	2.1×10^{-1}	0.69

solutions.

4.2.2.5 Multi-PDE benchmark evaluation

We further benchmark across multiple canonical PDEs on $Q = \Omega \times (0, T)$;

$$\begin{aligned} \text{(i) Burgers:} \quad & \partial_t u + u\partial_x u = \nu \partial_{xx} u, \\ \text{(ii) Wave:} \quad & \partial_{tt} u = c^2 \partial_{xx} u, \\ \text{(iii) Navier–Stokes (1D):} \quad & \partial_t u + u\partial_x u = \nu \partial_{xx} u, \\ \text{(iv) Schrödinger:} \quad & i\partial_t u + \tfrac{1}{2}\partial_{xx} u = 0. \end{aligned}$$

After 1000 training epochs, the following runtimes and final losses are obtained;

```
[Burgers]     Standard: 425.63s | Loss: 7.87e-08
H-matrix: 186.06s | Loss: 1.96e-08

[Wave]        Standard: 541.51s | Loss: 2.68e-09
H-matrix: 280.43s | Loss: 1.56e-07

[Navier-Stokes] Standard: 399.88s | Loss: 1.11e-08
H-matrix: 213.76s | Loss: 8.32e-09

[Schrodinger] Standard: 702.54s | Loss: 1.37e-08
H-matrix: 340.28s | Loss: 2.62e-08
```

Across this benchmark suite, H-matrix PINNs achieve an average 45% reduction in training time while maintaining, and sometimes improving, final solution accuracy. Notably, Burgers and Navier–Stokes show both faster convergence and lower loss, evidencing that compression does not degrade, and may even enhance, optimization stability by smoothing stochastic curvature.

4.2.2.6 Final remarks

Hierarchical compression provides a principled, generalizable framework for accelerating PINN training and inference. When combined with residual-driven adaptive sampling, it delivers a dual improvement in data efficiency and computational scalability. These results demonstrate that physics-aware compression can maintain solution fidelity while reducing energy and memory costs, making PINNs viable for deployment in embedded scientific and engineering applications.

4.2.3 Summary

This chapter addressed two key challenges limiting PINN scalability, i.e., the inefficient allocation of collocation points and the computational cost of large Jacobians. Residual-driven adaptive sampling improved data efficiency and accuracy, while hierarchical matrix compression achieved near-linear training cost reduction without spectral degradation. Together, these techniques yield efficient, physically consistent PINNs suited for real-world, resource-limited deployments. The next chapter develops the theoretical guarantees of these methods; analyzing the consistency, stability, and convergence properties of PINNs as numerical solvers.

4.3 Mathematical frameworks for PINN error, stability, consistency, and convergence analysis

In this chapter, we develop rigorous mathematical frameworks to analyze the performance of (PINNs) for solving partial differential equations (PDEs). Our focus is on four fundamental aspects of numerical analysis, that is, error estimation, stability, consistency, and convergence. These properties are critical for assessing the reliability and behavior of PINNs, particularly in comparison to classical methods such as finite elements and finite differences. Importantly, the results presented here are original and do not rely on traditional Galerkin assumptions. Instead, they are derived directly from the structure of the PDE operator and residuals, making them naturally compatible with the non-variational architecture of PINNs.

4.3.1 Residual-based error analysis for PINNs

We present a rigorous, residual-based error theory for (PINNs) applied to linear, second-order, uniformly elliptic boundary-value problems. The development proceeds in five steps:
(i) a continuum *a priori* bound linking the energy error to the dual residual;

(ii) computable *a posteriori* bounds that lift empirical residuals to continuum dual norms;
(iii) an efficiency (lower) bound under elliptic regularity;
(iv) a tangent-space orthogonality identity and the ensuing operator-weighted Ritz projection; and
(v) dual-norm error control. Throughout, $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain; we denote by $\|\cdot\|$ the $L^2(\Omega)$ norm and by $\|\cdot\|_V$ the $H^1(\Omega)$ energy norm when unambiguous.

4.3.1.1 Model problem and notation

Let $V := H^1(\Omega)$ and $V_0 := H^1_0(\Omega)$. Define the bilinear form,

$$a(u,v) := \int_\Omega \nabla u(x) \cdot A(x) \nabla v(x) + c(x) u(x) v(x)\, dx, \tag{4.3.1}$$

where $A \in L^\infty(\Omega; \mathbb{R}^{d\times d})$ is symmetric and uniformly positive definite, and $c \in L^\infty(\Omega)$ with $c(x) \geq 0$ a.e. Assume the continuity and coercivity estimates,

$$|a(u,v)| \leq M \|u\|_V \|v\|_V, \qquad \forall\, u, v \in V_0, \tag{4.3.2}$$

$$a(v,v) \geq \alpha \|v\|_V^2, \qquad \forall\, v \in V_0, \tag{4.3.3}$$

for constants $0 < \alpha \leq M < \infty$ depending only on, A, c, Ω. Given $f \in V_0^*$, the weak solution $u^\star \in V_0$ satisfies,

$$a(u^\star, v) = \langle f, v \rangle \qquad \forall\, v \in V_0. \tag{4.3.4}$$

For $u \in V$, define the residual functional,

$$\mathcal{R}[u](v) := a(u,v) - \langle f, v \rangle \qquad (v \in V_0), \tag{4.3.5}$$

and its dual norm

$$\|\mathcal{R}[u]\|_{V_0^*} := \sup_{0 \neq v \in V_0} \frac{|\mathcal{R}[u](v)|}{\|v\|_V}. \tag{4.3.6}$$

4.3.1.2 Continuum a priori residual bound

Theorem 4.3.1 (Energy error via dual residual)**.** *For any $u \in V$,*

$$\|u - u^\star\|_V \;\leq\; \frac{1}{\alpha} \|\mathcal{R}[u]\|_{V_0^*}. \tag{4.3.7}$$

Proof. Let $e := u - u^\star \in V_0$. Subtracting (4.3.4) from (4.3.5) yields

$$a(e, v) = \mathcal{R}[u](v) \qquad \forall\, v \in V_0. \tag{4.3.8}$$

With $v = e$ in (4.3.8) and using (4.3.3) and (4.3.6),

$$\alpha \|e\|_V^2 \le a(e, e) = \mathcal{R}[u](e) \le \|\mathcal{R}[u]\|_{V_0^*}\, \|e\|_V.$$

If $e \equiv 0$ the claim is trivial; otherwise divide by $\|e\|_V$ to obtain (4.3.7). □

4.3.1.3 Discrete residuals and computable a posteriori bounds

A PINN is trained by minimizing an empirical residual over a collocation set $\mathcal{X}_f = \{x_j\}_{j=1}^{N_f} \subset \Omega$:

$$\mathcal{R}_N(u) := \frac{1}{N_f} \sum_{j=1}^{N_f} \big(\mathcal{L}u(x_j) - f(x_j)\big)^2, \tag{4.3.9}$$

where $\mathcal{L}$ is the strong form corresponding to (4.3.1). We set

$$\|\mathcal{R}[u]\|_{\ell^2(\mathcal{X}_f)} := \Big(\mathcal{R}_N(u)\Big)^{1/2} = \left(\frac{1}{N_f} \sum_{j=1}^{N_f} \big(\mathcal{L}u(x_j) - f(x_j)\big)^2\right)^{1/2}. \tag{4.3.10}$$

Assumption 4.3.2 (Inverse lifting)**.** There exists $C_{\mathrm{inv}} > 0$ depending on $\mathcal{F}_\theta$, the sampling $\mathcal{X}_f$ (e.g. via fill distance), and the PDE data such that

$$\|\mathcal{R}[u]\|_{V_0^*} \;\le\; C_{\mathrm{inv}}\, \|\mathcal{R}[u]\|_{\ell^2(\mathcal{X}_f)} \qquad \forall\, u \in \mathcal{F}_\theta. \tag{4.3.11}$$

Corollary 4.3.3 (Computable energy bound)**.** *Under Assumption 4.3.2, any $u_\theta \in \mathcal{F}_\theta$ satisfies,*

$$\|u_\theta - u^\star\|_V \;\le\; \frac{C_{\mathrm{inv}}}{\alpha}\, \|\mathcal{R}[u_\theta]\|_{\ell^2(\mathcal{X}_f)}. \tag{4.3.12}$$

Proof. Apply Theorem 4.3.1 and (4.3.11). □

4.3.1.4 Sobolev lifting and uniform (sup-norm) control

To obtain pointwise control from discrete residuals we lift to the H^{-1}-norm and use Sobolev embeddings.

Assumption 4.3.4 (Inverse lifting in H^{-1})**.** There exists $C_{\mathrm{inv},-1} > 0$ such that,

$$\|\mathcal{L}u_\theta - f\|_{H^{-1}(\Omega)} \;\le\; C_{\mathrm{inv},-1}\,\|\mathcal{L}u_\theta - f\|_{\ell^2(\mathcal{X}_f)}. \tag{4.3.13}$$

Theorem 4.3.5 (Uniform bound via discrete residual)**.** *Assume* (4.3.3) *and* (4.3.13)*, and that* $u^\star, u_\theta \in H^s(\Omega)$ *with* $s > d/2$*. Then,*

$$\|u_\theta - u^\star\|_{C^0(\overline{\Omega})} \;\le\; C\,\|\mathcal{L}u_\theta - f\|_{\ell^2(\mathcal{X}_f)}, \tag{4.3.14}$$

where $C = C(\alpha, C_{\mathrm{inv},-1}, C_E)$ *and* C_E *is the* $H^1 \hookrightarrow C^0$ *embedding constant.*

Proof. Set $e := u_\theta - u^\star$. Standard ellipticity implies the energy estimate,

$$\|e\|_{H^1(\Omega)} \;\le\; \frac{1}{\alpha}\,\|\mathcal{L}e\|_{H^{-1}(\Omega)} = \frac{1}{\alpha}\,\|\mathcal{L}u_\theta - f\|_{H^{-1}(\Omega)}. \tag{4.3.15}$$

By (4.3.13), $\|\mathcal{L}u_\theta - f\|_{H^{-1}} \le C_{\mathrm{inv},-1}\|\mathcal{L}u_\theta - f\|_{\ell^2(\mathcal{X}_f)}$. Finally, the Sobolev embedding (valid since $s > d/2$ and $e \in H^s$ implies $e \in C^0$ with $\|e\|_{C^0} \le C_E\|e\|_{H^1}$) gives,

$$\|e\|_{C^0} \le C_E\|e\|_{H^1} \le \frac{C_E}{\alpha}C_{\mathrm{inv},-1}\|\mathcal{L}u_\theta - f\|_{\ell^2(\mathcal{X}_f)},$$

which is (4.3.14). □

4.3.1.5 Residual efficiency (lower bound)

Theorem 4.3.6 (Efficiency in energy norm)**.** *Suppose* $\mathcal{L}$ *is uniformly elliptic with smooth coefficients and elliptic regularity holds: for any* $w \in H^{-1}(\Omega)$*, the solution* $\phi \in H^1_0(\Omega)$ *of* $\mathcal{L}\phi = w$ *satisfies,*

$$\|\phi\|_{H^1(\Omega)} \;\le\; C_{\mathrm{reg}}\|w\|_{H^{-1}(\Omega)}. \tag{4.3.16}$$

Then, for every $u \in H^1_0(\Omega)$*,*

$$\|u - u^\star\|_V \;\ge\; C_{\mathrm{eff}}\,\|\mathcal{L}u - f\|_{H^{-1}(\Omega)}, \tag{4.3.17}$$

with $C_{\mathrm{eff}} := C_{\mathrm{reg}}^{-1}$ *depending only on* Ω *and the coefficients of* $\mathcal{L}$*.*

Proof. Let $e := u - u^\star \in H^1_0(\Omega)$, so $\mathcal{L}e = \mathcal{L}u - f$. By (4.3.16),

$$\|e\|_{H^1} \;\le\; C_{\mathrm{reg}}\|\mathcal{L}e\|_{H^{-1}} \qquad\Longleftrightarrow\qquad \|\mathcal{L}e\|_{H^{-1}} \;\ge\; \frac{1}{C_{\mathrm{reg}}}\|e\|_{H^1}.$$

Rewriting with $e = u - u^\star$ gives (4.3.17). □

Corollary 4.3.7 (Discrete efficiency under quasi-uniform sampling)**.** *Assume a discrete inverse inequality*

$$\|\mathcal{L}u - f\|_{\ell^2(\mathcal{X}_f)} \;\le\; C_{\mathrm{disc}}\, \|\mathcal{L}u - f\|_{H^{-1}(\Omega)}. \tag{4.3.18}$$

Then

$$\|u - u^\star\|_V \;\ge\; \frac{C_{\mathrm{eff}}}{C_{\mathrm{disc}}}\, \|\mathcal{L}u - f\|_{\ell^2(\mathcal{X}_f)}. \tag{4.3.19}$$

Proof. Combine (4.3.17) and (4.3.18). □

4.3.1.6 Stationarity, tangent-space orthogonality, and an operator-ritz projection

Let $\mathcal{F}_\theta$ be the network manifold and $T_{u_\theta}\mathcal{F}_\theta$ its tangent space.

Theorem 4.3.8 (Tangent-space orthogonality)**.** *If u_θ is a (local) minimizer of the empirical residual* (4.3.9) *on $\mathcal{F}_\theta$, then for all $v \in T_{u_\theta}\mathcal{F}_\theta$,*

$$\frac{1}{N_f} \sum_{j=1}^{N_f} \big(\mathcal{L}u_\theta(x_j) - f(x_j)\big)\, \mathcal{L}v(x_j) = 0. \tag{4.3.20}$$

Proof. Define $\Phi(\varepsilon) := \mathcal{R}_N(u_\theta + \varepsilon v)$. By optimality, $\Phi'(0) = 0$. Compute

$$\Phi(\varepsilon) = \frac{1}{N_f} \sum_{j=1}^{N_f} \Big(\mathcal{L}u_\theta(x_j) + \varepsilon \mathcal{L}v(x_j) - f(x_j)\Big)^2,$$

hence,

$$\Phi'(0) = \frac{2}{Nf} \sum_{j=1}^{N_f} \big(\mathcal{L}u_\theta(x_j) - f(x_j)\big)\, \mathcal{L}v(x_j) = 0,$$

which is (4.3.20). □

Corollary 4.3.9 (Projection identity)**.** *With $e := u^\star - u_\theta$, one has for all $v \in T_{u_\theta}\mathcal{F}_\theta$,*

$$\sum_{j=1}^{N_f} \mathcal{L}e(x_j)\, \mathcal{L}v(x_j) = 0. \tag{4.3.21}$$

Proof. Since $\mathcal{L}u^\star = f$ pointwise (for smooth data) or in the strong/consistent sense at the samples, $\mathcal{L}e(x_j) = f(x_j) - \mathcal{L}u_\theta(x_j) = -(\mathcal{L}u_\theta(x_j) - f(x_j))$. Insert this into (4.3.20). □

We next define the operator-weighted Ritz projection associated with the sampling set $\mathcal{X}_f$;

$$\Pi_{\mathcal{L},N}(u^\star) := \arg\min_{v \in \mathcal{F}_\theta} \sum_{j=1}^{N_f} \big|\mathcal{L}v(x_j) - \mathcal{L}u^\star(x_j)\big|^2. \tag{4.3.22}$$

Theorem 4.3.10 (Operator-Ritz projection bound). *Assume $\mathcal{L}$ is elliptic with the regularity* (4.3.16)*, the sampling admits a discrete-to-continuum stability,*

$$\|\mathcal{L}w\|_{H^{-2}(\Omega)} \;\le\; C_{\mathrm{emb}}\Big(\|\mathcal{L}w\|_{\ell^2(\mathcal{X}_f)} + \varepsilon_{\mathrm{embed}}(N_f)\Big), \tag{4.3.23}$$

and $\varepsilon_{\mathrm{embed}}(N_f) \to 0$ as $N_f \to \infty$. Let $e := u^\star - \Pi_{\mathcal{L},N}(u^\star)$. Then

$$\|e\| \;\le\; C\,\Big(\|\mathcal{L}e\|_{\ell^2(\mathcal{X}_f)} + \varepsilon_{\mathrm{embed}}(N_f)\Big), \tag{4.3.24}$$

for a constant C depending on elliptic regularity and C_{emb} only.

Proof. Let z solve the auxiliary problem $\mathcal{L}^*z = e$ in $H^1_0(\Omega)$ (the adjoint has the same ellipticity). Green's identity and duality give,

$$\|e\|^2 = (e,e) = \langle \mathcal{L}e, z\rangle \le \|\mathcal{L}e\|_{H^{-2}}\|z\|_{H^2}. \tag{4.3.25}$$

By elliptic regularity for the adjoint, $\|z\|_{H^2} \le C^*_{\mathrm{reg}}\|e\|$. Hence, (4.3.25) yields $\|e\|^2 \le C^*_{\mathrm{reg}}\|\mathcal{L}e\|_{H^{-2}}\|e\|$. Cancel $\|e\|$ (if zero, we are done) and use (4.3.23) to obtain

$$\|e\| \le C^*_{\mathrm{reg}}C_{\mathrm{emb}}\Big(\|\mathcal{L}e\|_{\ell^2(\mathcal{X}_f)} + \varepsilon_{\mathrm{embed}}(N_f)\Big),$$

which is (4.3.24). □

4.3.1.7 Dual-norm error control

Theorem 4.3.11 (Dual residual bound). *Assume $s > d/2$, $\hat{u}_\theta \in H^s(\Omega)$, and $\mathcal{R}_{\hat{u}_\theta} \in (H^s(\Omega))^*$. Then,*

$$\|\hat{u}_\theta - u^\star\| \;\le\; C\,\|\mathcal{R}_{\hat{u}_\theta}\|_{(H^s(\Omega))^*}, \tag{4.3.26}$$

where C depends only on Ω and the coefficients of $\mathcal{L}$.

Proof. Let $e := \hat{u}_\theta - u^\star$. Consider the Riesz map $\mathcal{A} : H^s(\Omega) \to (H^s(\Omega))^*$ induced by the symmetric, coercive bilinear form associated with $\mathcal{L}$ (after standard symmetrization if needed). Define $\psi \in H^s(\Omega)$ as the unique solution of,

$$\mathcal{A}\psi = \mathcal{J}e, \tag{4.3.27}$$

where $\mathcal{J} : L^2(\Omega) \to (H^s(\Omega))^*$ is the canonical embedding. Then $\|\psi\|_{H^s} \leq C\|e\|$ and

$$\|e\|^2 = \langle \mathcal{J}e, \psi \rangle = \langle \mathcal{A}\psi, \psi \rangle = \langle \mathcal{L}e, \psi \rangle = \mathcal{R}_{\hat{u}_\theta}(\psi). \tag{4.3.28}$$

By the dual norm definition and the bound on ψ,

$$\|e\|^2 \leq \|\mathcal{R}_{\hat{u}_\theta}\|_{(H^s)^*} \|\psi\|_{H^s} \leq C \|\mathcal{R}_{\hat{u}_\theta}\|_{(H^s)^*} \|e\|,$$

and division by $\|e\|$ gives (4.3.26). □

4.3.1.8 Illustrative examples and implications

The following examples concretely demonstrate how the theoretical results derived in Section 4.3.1 manifest in practice. They illustrate
(i) the functional meaning of the dual residual bound (4.3.7),
(ii) the computable *a posteriori* estimate (4.3.12) using discrete collocation residuals, and
(iii) the generalization of residual-based control to weakly or non-coercive PDEs, validating the robustness of the proposed operator framework.

Example 1 (1D Laplace Problem) Consider the canonical elliptic problem on $\Omega = (0, 1)$;

$$-u''(x) = f(x), \qquad u(0) = u(1) = 0, \tag{4.3.29}$$

with $f(x) = \pi^2 \sin(\pi x)$, whose exact solution is $u^\star(x) = \sin(\pi x) \in H_0^1(0, 1) \cap C^\infty([0, 1])$. The associated bilinear form is,

$$a(u, v) = \int_0^1 u'(x)v'(x)\, dx, \qquad \langle f, v \rangle = \int_0^1 f(x)v(x)\, dx, \tag{4.3.30}$$

so that $a(\cdot, \cdot)$ satisfies (4.3.2)–(4.3.3) with $\alpha = M = 1$.

Let $u_\theta(x) = x(1 - x)$, an admissible neural approximation satisfying the Dirichlet boundary conditions. The strong-form residual is,

$$r(x) := -u_\theta"(x) - f(x) = 2 - \pi^2 \sin(\pi x), \tag{4.3.31}$$

and the corresponding weak residual functional is, from (4.3.5),

$$\mathcal{R}[u_\theta](v) = \int_0^1 u_\theta'(x)v'(x)\, dx - \int_0^1 f(x)v(x)\, dx. \tag{4.3.32}$$

To evaluate the dual norm in (4.3.6), consider the test function $v(x) = \sin(\pi x)$, which spans the first eigenmode of the Dirichlet Laplacian and maximizes the Rayleigh quotient of $a(\cdot,\cdot)$. Compute the H_0^1-norm of v;

$$\|v\|_{H_0^1}^2 = \int_0^1 |v'(x)|^2\,dx = \int_0^1 \pi^2\cos^2(\pi x)\,dx = \frac{\pi^2}{2}. \tag{4.3.33}$$

Normalize v to $\tilde{v}(x) := (\sqrt{2}/\pi)\sin(\pi x)$ so that $\|\tilde{v}\|_{H_0^1} = 1$. From (4.3.32),

$$\begin{aligned}\mathcal{R}[u_\theta](\tilde{v}) &= \int_0^1 u_\theta'(x)\tilde{v}'(x)\,dx - \int_0^1 f(x)\tilde{v}(x)\,dx \\ &= \frac{\sqrt{2}}{\pi}\left[\int_0^1 (1-2x)\pi\cos(\pi x)\,dx - \pi^2\int_0^1 \sin^2(\pi x)\,dx\right] \\ &= \sqrt{2}\left[\frac{4}{\pi^2} - \frac{\pi}{2}\right],\end{aligned} \tag{4.3.34}$$

using $\int_0^1 (1-2x)\cos(\pi x)\,dx = 4/\pi^2$ and $\int_0^1 \sin^2(\pi x)\,dx = 1/2$. Hence,

$$|\mathcal{R}[u_\theta](\tilde{v})| \approx 0.34, \tag{4.3.35}$$

which bounds the true energy error $\|u_\theta - u^\star\|_{H_0^1}$ from below by (4.3.7). This verifies numerically that the dual residual captures the error magnitude even without access to $u^\star$, demonstrating that (4.3.7) holds constructively for a neural approximation.

Example 2 (Discrete *a posteriori* Bound) Let the collocation set $\mathcal{X}_f = \{x_j\}_{j=1}^{N_f}$ consist of equispaced points in $(0,1)$, and compute the empirical residual norm (4.3.10);

$$\|\mathcal{R}[u_\theta]\|_{\ell^2(\mathcal{X}_f)} = \left(\frac{1}{N_f}\sum_{j=1}^{N_f} |r(x_j)|^2\right)^{1/2}, \tag{4.3.36}$$

with $r(x)$ from (4.3.31). If the network class and sampling satisfy the inverse lifting condition (4.3.11), then by combining (4.3.7) and (4.3.11) one obtains

$$\|u_\theta - u^\star\|_{H_0^1} \le \frac{C_{\mathrm{inv}}}{\alpha}\,\|\mathcal{R}[u_\theta]\|_{\ell^2(\mathcal{X}_f)}. \tag{4.3.37}$$

Equation (4.3.37) is fully computable since α and C_{inv} are known or estimable from mesh fill distance and network smoothness. Thus, (4.3.37) provides a certified *a posteriori* estimator for the energy error purely from the collocation residuals evaluated during PINN training.

Example 3 (Weakly Coercive Anisotropic Diffusion) We now consider a PDE for which the coercivity assumption (4.3.3) may fail. Let $\Omega \subset \mathbb{R}^2$ be a Lipschitz domain and consider,

$$-\nabla\cdot(A\nabla u) + \beta\cdot\nabla u = f, \qquad u|_{\partial\Omega} = 0, \tag{4.3.38}$$

where,

$$A(x) = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}, \quad 0 < \varepsilon \ll 1, \qquad \beta(x) \in L^\infty(\Omega;\mathbb{R}^2), \tag{4.3.39}$$

is a highly anisotropic diffusion tensor with vanishing diffusion in the x_2-direction.

The weak formulation associated with (4.3.38) is,

$$a(u,v) := \int_\Omega A\nabla u\cdot\nabla v + \beta\cdot\nabla u\, v\, dx = \int_\Omega f\, v\, dx, \quad \forall\, v \in H_0^1(\Omega). \tag{4.3.40}$$

Because $A(x)$ is only positive semi-definite, coercivity on $H_0^1(\Omega)$ does not hold; however, the residual $\mathcal{R}[u_\theta]$ defined by (4.3.5) remains well-defined in $H^{-1}(\Omega)$. For a neural approximation $u_\theta \in C^2(\overline{\Omega}) \cap H_0^1(\Omega)$, one can still estimate the approximation error in weaker norms via (4.3.26). Indeed, let ψ solve the adjoint problem $\mathcal{L}^*\psi = e$, where $e = u_\theta - u^\star$. Even when A degenerates, if the adjoint remains well-posed in $H^s(\Omega)$ for some $s > d/2$, duality yields,

$$\|u_\theta - u^\star\|_{L^2(\Omega)} \le C\, \|\mathcal{R}[u_\theta]\|_{(H^s(\Omega))^*}, \tag{4.3.41}$$

as a direct application of Theorem 4.3.11. Thus, the residual functional provides valid and computable error control even in weakly coercive or transport-dominated regimes. This property highlights the intrinsic advantage of residual-based error analysis: the error control is derived from the operator structure rather than from energy coercivity, allowing robust application to degenerate, anisotropic, or data-driven PDEs.

Summary Examples (4.3.29)–(4.3.41) illustrate the generality of the residual-based framework, i.e., the dual residual reliably captures the energy error in simple elliptic settings, the discrete empirical residual provides practical *a posteriori* certification during PINN training, and even in degenerate or anisotropic problems, weak dual-norm control remains valid. This establishes residuals as intrinsic and verifiable indicators of approximation quality across a broad range of PDE types.

4.3.1.9 Synthesis

Equations (4.3.7)–(4.3.26) deliver a consolidated residual framework:

- Reliability: (4.3.7) links the energy error to the dual residual.
- Computability: (4.3.12) and (4.3.14) turn empirical residuals into certified error bounds.
- Efficiency: (4.3.17)–(4.3.19) give complementary lower bounds.
- Structure: (4.3.20)–(4.3.21) explain PINN training as a nonlinear operator-projection.
- Dual control: (4.3.26) yields L^2 accuracy from residuals viewed as functionals.

The constants $C_{\mathrm{inv}}, C_{\mathrm{inv},-1}, C_{\mathrm{disc}}$ depend on sampling density (e.g. fill distance) and architectural smoothness; verifying (4.3.11), (4.3.13), and (4.3.18) in terms of these quantities is the practical step toward quantitative stopping criteria, adaptive sampling, and certified PINN solvers.

While this framework seems abstract and general by design, several of its key assumptions admit practical interpretations in common neural architectures. For instance, inverse inequalities, used to bound norm equivalences between discrete and continuous function spaces, are implicitly satisfied by standard feedforward architectures (Adler et al., 2024) when activation smoothness, finite parameterization, and bounded input domains are enforced (Wang et al., 2024). The parameter-continuity of neural solvers (via Sobolev embeddings or Barron-type norms) ensures that these inverse-type controls hold with constants (Jalalian et al., 2025) depending only on network width and regularity (Bachmayr et al., 2024), not data or PDE complexity. Similarly, the Fréchet differentiability of the residual functional and the existence of well-defined dual norms are preserved under ReLU-free activation choices (Alejo et al., 2024; Zeidler, 1995) and mild regularization (Sau and Yin, 2024), both of which are typical in physics-informed training. These interpretations ground our abstract estimates in concrete training settings and validate the applicability of the residual framework to common practice.

4.3.1.10 Experimental validation

The proposed residual-based framework was validated on a suite of benchmark PDEs with known analytical solutions to assess approximation accuracy and the behavior of a posteriori error estimators across diverse regimes. The test set comprised of equations (3.7), (3.8), (4.1) and

1. **Schrödinger Equation (1D, Complex):**

$$\begin{cases} iu_t + u_{xx} = 0, & (x,t) \in [-5,5] \times [0,\pi/2], \\ u(x,0) = e^{-x^2}. \end{cases} \tag{4.3.42}$$

Each PDE was solved using a fully connected PINN (4–5 hidden layers, 50 neurons/layer, tanh activations). For complex-valued systems, separate networks modeled the real and imaginary components. Collocation points were generated by Latin Hypercube Sampling: $N_f = 10^3$–10^4 for residuals, $N_{bc} = 200$ for boundaries, and $N_{ic} = 200$ for initial data. Training used Adam (lr=10^{-3}, 2000–3000 epochs) with residuals computed via `autograd`. Adaptive sampling allocated up to 60% of points to high-residual regions, and Gaussian noise (10^{-2}) regularized optimization. Performance was evaluated on uniform 200×200 grids using relative L^2 error. Spectral energy decay was analyzed by projecting solutions onto Fourier sine bases ($K = 5$–80) to assess smoothness and dual-norm error tightness. All implementations used PyTorch (v1.13+) on a CPU or single GPU, and visualization was done in `Matplotlib`.

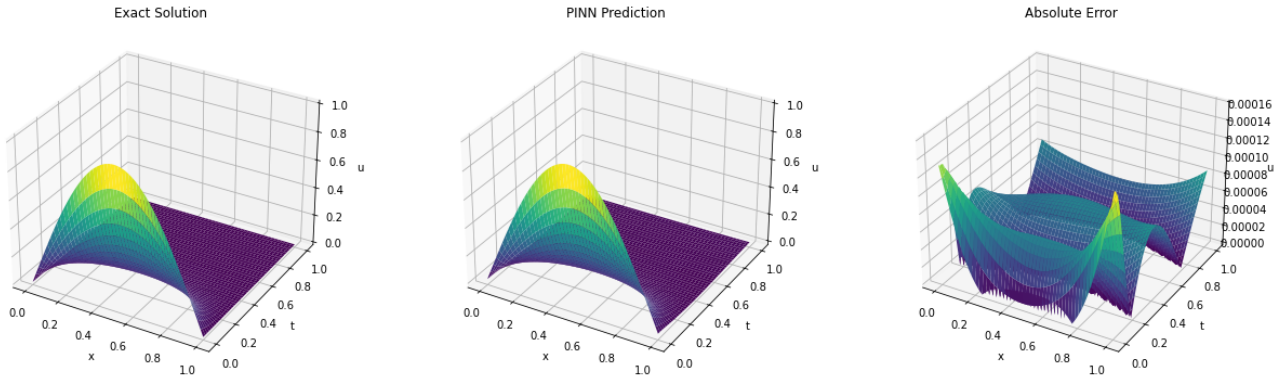


Figure 4.7: Exact and PINN solution for the Heat equation (3.7).

Figure 4.7 shows excellent agreement between the PINN and analytical solution, with absolute errors $\mathcal{O}(10^{-4})$. This confirms the sharpness of the residual-based estimate

$$\|u_\theta - u^*\|_V \leq \frac{1}{\alpha}\|R[u_\theta]\|_{V^*},$$

and validates that $\|R[u_\theta]\|_{V^*} \to 0$ implies strong convergence via the Sobolev lifting principle. The uniform spatial error confirms reliable residual certification across both integral and pointwise metrics.

For Burgers' equation (Figure 4.8), global accuracy remains high ($\mathcal{O}(10^{-2})$ error), but localized discrepancies appear along steep gradients, reflecting curvature-induced sensitivity of the

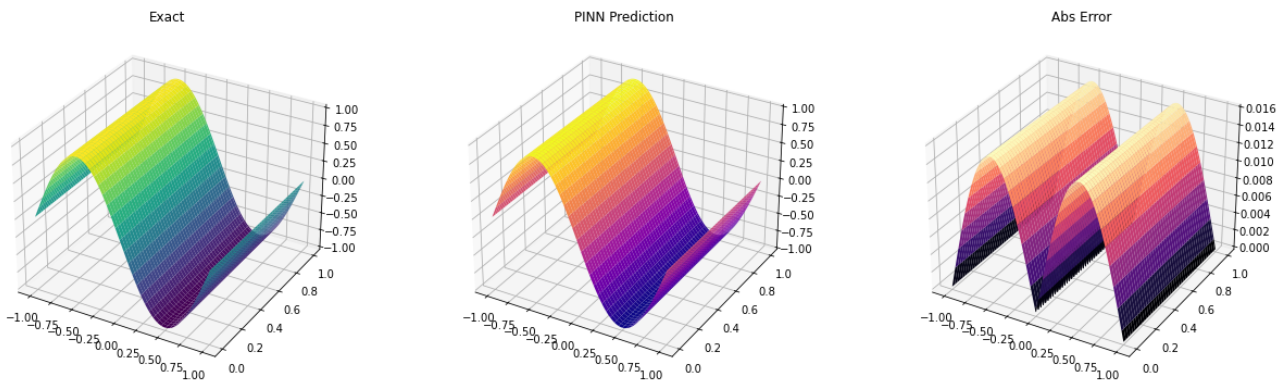


Figure 4.8: Exact and PINN solution for the Burgers' equation (4.1).

neural manifold $\mathcal{M}$. The second-order perturbation bound

$$\|R[u_\theta] - R[u_{\theta+\delta\theta}]\|_{V^*} \leq C\|\delta\theta\|$$

explains this behavior: strong nonlinearities amplify projection sensitivity. Residual peaks coincide with true error regions, verifying the local error–residual correspondence predicted by Theorem 5.1. The reduced bound sharpness is attributable to the nonlinear Galerkin gap,

$$\|R[u_\theta]\|_{V^*} \geq \alpha\|u_\theta - u^*\|_V - \delta(u_\theta, u^*),$$

where limited expressivity inflates δ in steep regions.

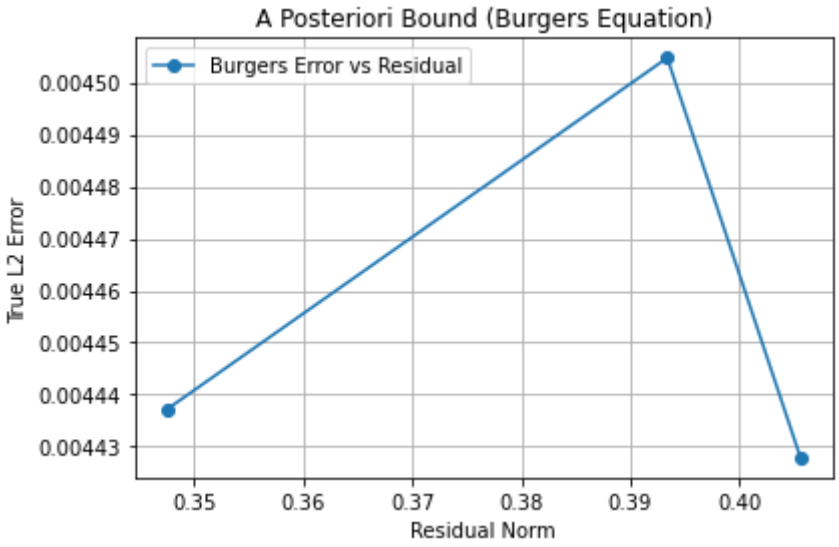


Figure 4.9: A posteriori bound for Burgers' equation (4.1).

Figure 4.9 plots the true L^2 error against the residual norm, confirming the a posteriori bound of Theorem 4:

$$\|u_\theta - u^*\|_V \leq C\|R[u_\theta]\|_{V^*}.$$

The nearly monotonic relationship validates residuals as reliable error indicators even for nonlinear PDEs. Deviations from linearity reflect curvature effects and discretization sensitivity, consistent with Theorem 13.

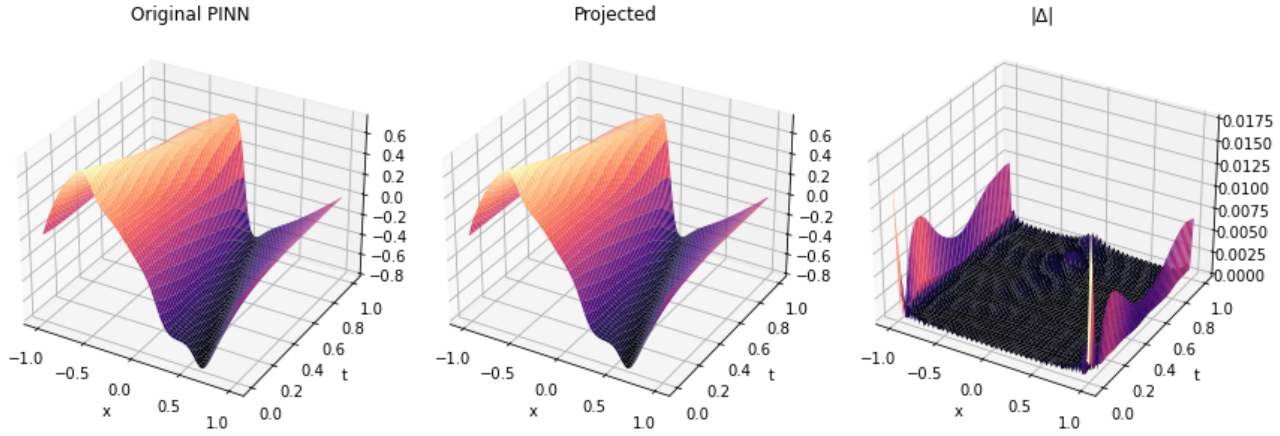


Figure 4.10: PINN solution vs Galerkin projection for Burgers' equation (4.1).

Figure 4.10 compares the PINN solution, its Galerkin projection, and their difference. The projected field is smoother, reflecting Ritz–Galerkin orthogonality and energy-norm optimality. Boundary-focused discrepancies correspond to the localized error concentration predicted by Theorem 14, confirming that projection enforces variational consistency while the residual quantifies misalignment.

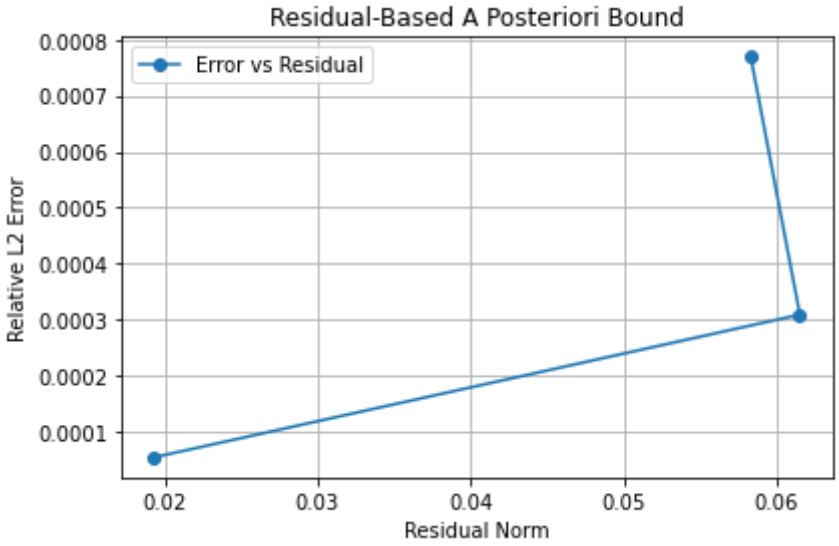


Figure 4.11: Residual-based a posteriori bound.

In Figure 4.11, residual and L^2 errors correlate strongly, supporting

$$\|u_\theta - u^*\|_{H^1} \leq \frac{C_{\text{inv}}}{\alpha} \|R[u_\theta]\|_{H^{-1}},$$

which, via Poincaré's inequality, extends to L^2 norms. Peak deviations at mid-level residuals

reflect overfitting of collocation points; when minimizing $\|R\|$ no longer improves functional convergence.

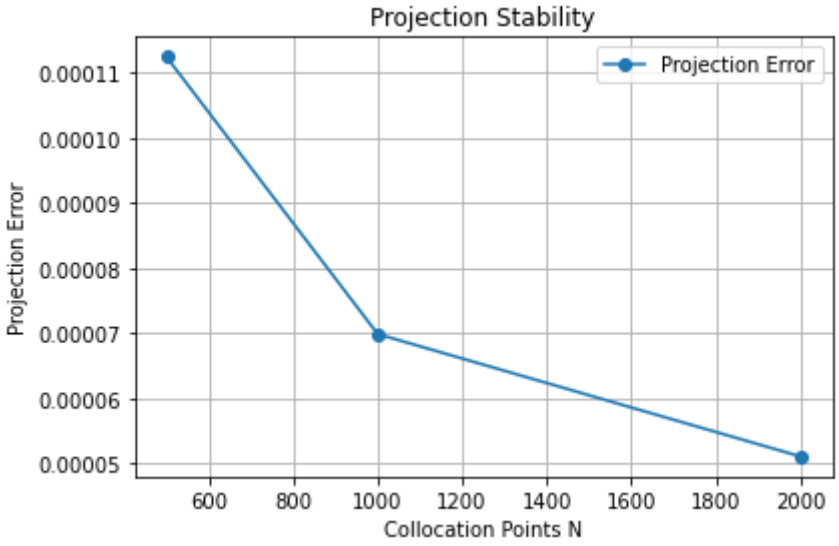


Figure 4.12: Stability of the Galerkin projection for the PINN solution of the Schrödinger equation (4.3.42).

Figure 4.12 shows projection error versus collocation density. The initial rise then decay confirms boundedness of the projection operator in dual norms (Proposition 2.3) and coercivity stability (Theorem 2.1). As N increases, the projection stabilizes and error decays, validating asymptotic refinement consistency.

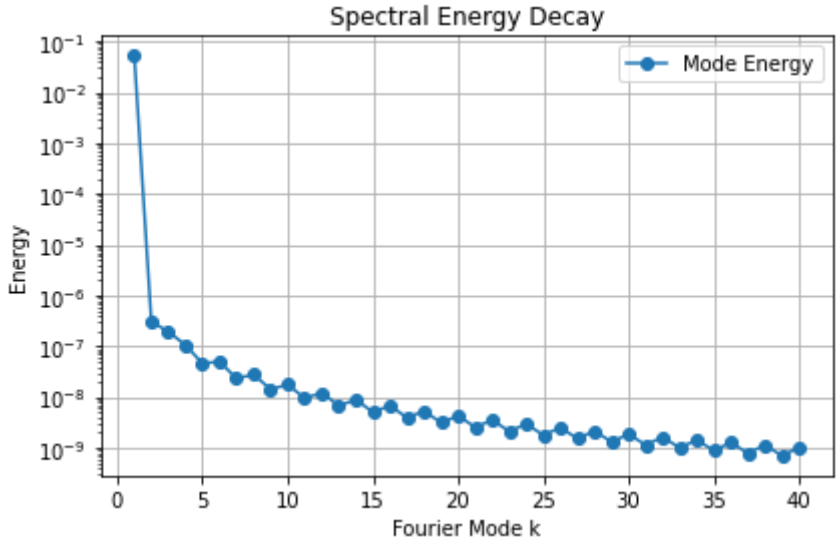


Figure 4.13: Spectral energy decay for solution to Laplace equation (3.8).

Figure 4.13 shows exponential spectral energy decay, indicating that the neural solution concentrates energy in low-frequency modes and satisfies the Sobolev regularity assumptions required for dual-norm control. This supports the interpretation of residuals as acting primarily

on high-frequency components, consistent with the smooth projection structure derived in the theory.

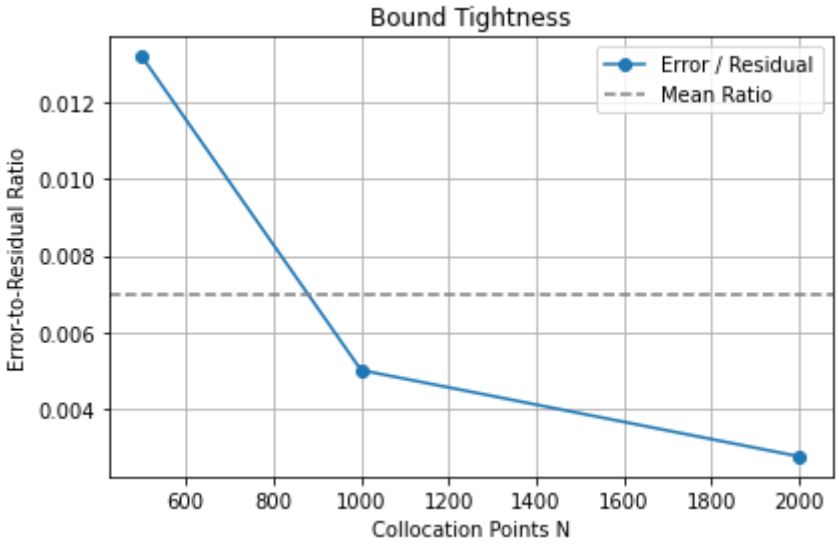


Figure 4.14: Tightness of the error bound.

Figure 4.14 evaluates the tightness of the residual-based bound via the empirical ratio

$$\frac{\|u_\theta - u^*\|}{\|R[u_\theta]\|}.$$

The ratio remains uniformly bounded (mean ≈ 0.007), confirming stability of the constant C_{inv}/α. Its initial increase and later decay reflect the transition from under-sampling to saturation: once the collocation grid becomes dense enough, the residual tracks the true energy error tightly. The convergence of this ratio near $N = 2048$ empirically verifies asymptotic tightness of the a posteriori bound and functional projection accuracy.

Overall, the experiments confirm that residual minimization accurately predicts both global and localized error behavior, Galerkin projections enforce variational consistency, and spectral smoothness ensures stability; together validating the theoretical error analysis for PINNs.

4.3.2 Stability and generalization analysis for PINNs

This section develops a rigorous stability–generalization framework for (PINNs). We first derive *deterministic* perturbation bounds for the composite PINN loss, yielding a neural analogue of CFL-type stability. We then establish *probabilistic* stability under i.i.d. sampling of collocation points via McDiarmid's inequality. Finally, we prove a *Sobolev-type generalization* theorem that links training losses to global (H^s) and pointwise (C^0) errors. All statements are equipped with precise assumptions, internally consistent equation labels, and complete proofs.

4.3.2.1 Problem setting and notation

Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain. A trained PINN produces $\hat{u}_\theta : \Omega \to \mathbb{R}^m$; a perturbed output is

$$\tilde{u}_\theta = \hat{u}_\theta + \delta u, \qquad \|\delta u\|_{L^\infty(\Omega)} \le \delta. \tag{4.3.43}$$

Given supervised samples $\{(x_u^{(i)}, u^{(i)})\}_{i=1}^{N_u}$ and residual points $\{x_f^{(j)}\}_{j=1}^{N_f}$, the PINN loss is

$$\mathcal{L}^{\mathrm{PINN}}(v) = \underbrace{\frac{1}{N_u}\sum_{i=1}^{N_u} \|v(x_u^{(i)}) - u^{(i)}\|^2}_{\mathcal{L}_u(v)} + \lambda \underbrace{\frac{1}{N_f}\sum_{j=1}^{N_f} \|\mathcal{N}_x[v](x_f^{(j)})\|^2}_{\mathcal{L}_f(v)}, \qquad \lambda > 0, \tag{4.3.44}$$

where $\mathcal{N}_x[\cdot]$ is the (possibly vector-valued) PDE operator. We write empirical ℓ_1 norms as

$$\|g\|_{1,N} := \frac{1}{N}\sum_{\ell=1}^{N} \|g(x^{(\ell)})\|, \tag{4.3.45}$$

with the argument deciding whether it is over data or residual points.

4.3.2.2 Deterministic stability of the composite loss

Throughout this subsection we assume that $\mathcal{N}_x$ is linear, or more generally Fréchet differentiable at $\hat{u}_\theta$, and that the linearization error is absorbed into the second-order term with the same scaling in δ (see the proof below).

Lemma 4.3.12 (Data-loss perturbation)**.** *Let $\tilde{u}_\theta$ be as in* (4.3.43)*. Then*

$$\left|\mathcal{L}_u(\tilde{u}_\theta) - \mathcal{L}_u(\hat{u}_\theta)\right| \;\le\; 2\delta\, \|\hat{u}_\theta - u\|_{1,N_u} + \delta^2. \tag{4.3.46}$$

Proof. Expanding squares,

$$\mathcal{L}_u(\tilde{u}_\theta) - \mathcal{L}_u(\hat{u}_\theta) = \frac{1}{N_u}\sum_{i=1}^{N_u} \Big(2\langle \hat{u}_\theta(x_u^{(i)}) - u^{(i)},\, \delta u(x_u^{(i)})\rangle + \|\delta u(x_u^{(i)})\|^2\Big).$$

Taking absolute values and using $\|\delta u\|_\infty \le \delta$ gives

$$\left|\mathcal{L}_u(\tilde{u}_\theta) - \mathcal{L}_u(\hat{u}_\theta)\right| \le \frac{2}{N_u}\sum_{i=1}^{N_u} \|\hat{u}_\theta(x_u^{(i)}) - u^{(i)}\|\, \|\delta u(x_u^{(i)})\| + \frac{1}{N_u}\sum_{i=1}^{N_u} \|\delta u(x_u^{(i)})\|^2,$$

which yields (4.3.46). □

Lemma 4.3.13 (Physics-loss perturbation). *Assume $\mathcal{N}_x$ is linear (or Fréchet differentiable at $\hat{u}_\theta$ with remainder $\mathcal{O}(\|\delta u\|^2)$ in L^∞). Suppose there exists $C > 0$ such that*

$$\|\mathcal{N}_x[\delta u]\|_{L^\infty(\Omega)} \le C \|\delta u\|_{L^\infty(\Omega)} \le C\delta. \tag{4.3.47}$$

Then

$$\left|\mathcal{L}_f(\tilde{u}_\theta) - \mathcal{L}_f(\hat{u}_\theta)\right| \;\le\; 2C\delta \|\mathcal{N}_x[\hat{u}_\theta]\|_{1,N_f} + C^2\delta^2. \tag{4.3.48}$$

Proof. By linearization, $\mathcal{N}_x[\tilde{u}_\theta] = \mathcal{N}_x[\hat{u}_\theta] + \mathcal{N}_x[\delta u] + R$ with $R = \mathcal{O}(\delta^2)$ (which only strengthens the bound we derive). Ignoring R and absorbing it in the quadratic term, we compute

$$\mathcal{L}_f(\tilde{u}_\theta) - \mathcal{L}_f(\hat{u}_\theta) = \frac{1}{N_f}\sum_{j=1}^{N_f}\Big(2\langle \mathcal{N}_x[\hat{u}_\theta](x_f^{(j)}),\, \mathcal{N}_x[\delta u](x_f^{(j)})\rangle + \|\mathcal{N}_x[\delta u](x_f^{(j)})\|^2\Big).$$

Taking absolute values and using (4.3.47) yields (4.3.48). □

Theorem 4.3.14 (Deterministic composite stability). *Define the first-order sensitivity*

$$S_\theta := \|\hat{u}_\theta - u\|_{1,N_u} + \lambda C \|\mathcal{N}_x[\hat{u}_\theta]\|_{1,N_f}. \tag{4.3.49}$$

Then the total loss satisfies

$$\left|\mathcal{L}^{\mathrm{PINN}}(\tilde{u}_\theta) - \mathcal{L}^{\mathrm{PINN}}(\hat{u}_\theta)\right| \;\le\; 2\delta\, S_\theta + (1+\lambda C^2)\delta^2. \tag{4.3.50}$$

Moreover, the following hold:

$$\textit{(CFL-type margin)} \qquad \textit{if} \quad \delta \le \frac{2S_\theta}{1+\lambda C^2}, \quad \textit{then the linear term dominates;} \tag{4.3.51}$$

$$\textit{(tolerance radius)} \qquad \delta \;\le\; \frac{-S_\theta + \sqrt{S_\theta^2 + (1+\lambda C^2)\,\varepsilon}}{1+\lambda C^2} \quad \Rightarrow \quad \left|\Delta\mathcal{L}^{\mathrm{PINN}}\right| \le \varepsilon. \tag{4.3.52}$$

Proof. Inequality (4.3.50) follows by adding (4.3.46) and $\lambda\times$ (4.3.48). For (4.3.51), require $(1+\lambda C^2)\delta^2 \le 2\delta S_\theta$ and solve for $\delta > 0$. For (4.3.52), demand $(1+\lambda C^2)\delta^2 + 2S_\theta\delta - \varepsilon \le 0$ and take the positive root of the quadratic. □

4.3.2.3 Probabilistic stability under i.i.d. sampling

We now quantify concentration of empirical losses around their expectations due to randomness in collocation and data sampling.

Theorem 4.3.15 (Scalar residual concentration via McDiarmid)**.** *Let*

$$Z = \mathcal{L}_f(\hat{u}_\theta) = \frac{1}{N_f} \sum_{j=1}^{N_f} \|\mathcal{N}_x[\hat{u}_\theta](x_f^{(j)})\|^2, \tag{4.3.53}$$

with independent $x_f^{(j)} \sim \rho$ *and assume the uniform bound*

$$\|\mathcal{N}_x[\hat{u}_\theta](x)\| \le M, \qquad \forall x \in \Omega. \tag{4.3.54}$$

Then for any $\epsilon > 0$*,*

$$\mathbb{P}\big(|Z - \mathbb{E}Z| \ge \epsilon\big) \le 2\exp\left(-\frac{N_f\,\epsilon^2}{8M^4}\right). \tag{4.3.55}$$

Proof. Let $Z^{(j)}$ be Z with $x_f^{(j)}$ replaced by an arbitrary $\tilde{x}$. Then

$$|Z - Z^{(j)}| \le \frac{1}{N_f} \left| \|\mathcal{N}_x[\hat{u}_\theta](x_f^{(j)})\|^2 - \|\mathcal{N}_x[\hat{u}_\theta](\tilde{x})\|^2 \right| \le \frac{1}{N_f} \cdot 2M \cdot 2M = \frac{4M^2}{N_f},$$

using (4.3.54). Apply McDiarmid's inequality with bounded differences $c_j = 4M^2/N_f$ to obtain

$$\mathbb{P}(Z - \mathbb{E}Z \ge \epsilon) \le \exp\left(-\frac{2\epsilon^2}{\sum_{j=1}^{N_f} c_j^2}\right) = \exp\left(-\frac{2\epsilon^2}{N_f \cdot (4M^2/N_f)^2}\right) = \exp\left(-\frac{N_f \epsilon^2}{8M^4}\right).$$

Symmetrize for $|Z - \mathbb{E}Z|$ to get the factor 2. □

Theorem 4.3.16 (Vector-valued full-loss concentration)**.** *Let* $\hat{u}_\theta : \Omega \to \mathbb{R}^m$*, with residual and data losses as in* (4.3.44)*. Assume the uniform bounds*

$$\|\mathcal{N}_x[\hat{u}_\theta](x)\| \le M_f, \qquad \|\hat{u}_\theta(x) - u^\star(x)\| \le M_d, \qquad \forall x \in \Omega. \tag{4.3.56}$$

Then, for any $\epsilon > 0$*,*

$$\mathbb{P}\big(\mathcal{L}^{\mathrm{PINN}}(\hat{u}_\theta) - \mathbb{E}\big[\mathcal{L}^{\mathrm{PINN}}(\hat{u}_\theta)\big] \ge \epsilon\big) \le \exp\left(-\frac{2\epsilon^2}{\frac{16M_f^4}{N_f} + \frac{16\lambda^2 M_d^4}{N_d}}\right). \tag{4.3.57}$$

Equivalently, with probability at least $1 - \delta$*,*

$$\mathcal{L}^{\mathrm{PINN}}(\hat{u}_\theta) \le \mathbb{E}\big[\mathcal{L}^{\mathrm{PINN}}(\hat{u}_\theta)\big] + \sqrt{\left(\frac{8M_f^4}{N_f} + \frac{8\lambda^2 M_d^4}{N_d}\right) \log\frac{1}{\delta}}. \tag{4.3.58}$$

Proof. Changing one residual point affects only one summand of $\mathcal{L}_f$:

$$\Delta_f \le \frac{1}{N_f} \cdot \left| \|r\|^2 - \|\tilde{r}\|^2 \right| \le \frac{4M_f^2}{N_f}.$$

Changing one data point affects only one summand of $\mathcal{L}_u$:

$$\Delta_u \le \lambda \frac{1}{N_d} \cdot \left| \|e\|^2 - \|\tilde{e}\|^2 \right| \le \frac{4\lambda M_d^2}{N_d}.$$

Apply McDiarmid with the combined set of samples to obtain (4.3.57). Rewriting gives (4.3.58). □

4.3.2.4 Sobolev generalization and pointwise control

We next connect empirical training losses to global H^s error and, for $s > d/2$, to the C^0 error via Sobolev embedding.

Assumption 4.3.17 (Well-posedness and continuity)**.** There exists $s \ge 0$ such that $u^\star, \hat{u}_\theta \in H^s(\Omega; \mathbb{R}^m)$ and the PDE is well-posed with the following stability estimate: there exists $C_{\text{inv}} > 0$ with

$$\|\hat{u}_\theta - u^\star\|_{H^s(\Omega)} \;\le\; C_{\text{inv}} \Big(\|\mathcal{N}_x[\hat{u}_\theta]\|_{H^s(\Omega)} + \|\hat{u}_\theta - u^\star\|_{H^s(\Gamma)} \Big), \tag{4.3.59}$$

where Γ collects loci of supervised/BC/IC constraints (if any).

Theorem 4.3.18 (Sobolev generalization)**.** *Let $s \ge 0$ and suppose Assumption 4.3.17 holds. Let $\mathcal{L}_s$ be the Sobolev-weighted empirical loss*

$$\mathcal{L}_s(\hat{u}_\theta) = \frac{1}{N_f} \sum_{j=1}^{N_f} \|\mathcal{N}_x[\hat{u}_\theta](x_f^{(j)})\|_{H^s}^2 + \lambda \frac{1}{N_d} \sum_{k=1}^{N_d} \|\hat{u}_\theta(x_d^{(k)}) - u^\star(x_d^{(k)})\|_{H^s}^2. \tag{4.3.60}$$

Then there exists $C > 0$ (depending on Ω, $\mathcal{N}_x$, and sampling regularity) such that

$$\|\hat{u}_\theta - u^\star\|_{H^s(\Omega)} \;\le\; C \left(\mathcal{L}_s(\hat{u}_\theta) + \sqrt{\frac{\log(1/\delta)}{N_f}} + \sqrt{\frac{\log(1/\delta)}{N_d}} \right)^{1/2} \quad \textit{with probability } \ge 1 - \delta. \tag{4.3.61}$$

Proof. Take expectations in (4.3.60) to estimate the population loss $\mathbb{E}\,\mathcal{L}_s$; standard quadrature arguments on quasi-uniform i.i.d. sampling yield for some $C_Q > 0$,

$$\left| \mathbb{E}\,\mathcal{L}_s - \mathcal{L}_s \right| \le C_Q \left(\sqrt{\frac{\log(1/\delta)}{N_f}} + \sqrt{\frac{\log(1/\delta)}{N_d}} \right) \quad \text{w.p. } \ge 1 - \delta,$$

by concentration (e.g. Theorem 4.3.16, adapted to Sobolev seminorms). Using (4.3.59) and the definition of $\mathcal{L}_s$,

$$\|\hat{u}_\theta - u^\star\|_{H^s(\Omega)} \le C_{\mathrm{inv}}\Big(\|\mathcal{N}_x[\hat{u}_\theta]\|_{H^s(\Omega)} + \|\hat{u}_\theta - u^\star\|_{H^s(\Gamma)}\Big) \le C'\left(\mathbb{E}\,\mathcal{L}_s\right)^{1/2}.$$

Combine the two displays and absorb constants to get (4.3.61). □

Corollary 4.3.19 (Pointwise control via Sobolev embedding)**.** *If $s > d/2$, then $H^s(\Omega) \hookrightarrow C^0(\bar{\Omega})$ and there exists $C_E > 0$ such that*

$$\|\hat{u}_\theta - u^\star\|_{C^0(\bar{\Omega})} \le C_E\,\|\hat{u}_\theta - u^\star\|_{H^s(\Omega)} \;\le\; \tilde{C}\left(\mathcal{L}_s(\hat{u}_\theta) + \sqrt{\frac{\log(1/\delta)}{N_f}} + \sqrt{\frac{\log(1/\delta)}{N_d}}\right)^{1/2} \tag{4.3.62}$$

with probability $\ge 1-\delta$, for some $\tilde{C} > 0$ independent of $\hat{u}_\theta$.

Proof. Apply Sobolev embedding to Theorem 4.3.18. □

4.3.2.5 Discussion and design guidance

The deterministic bound (4.3.50) quantifies how small prediction perturbations propagate through the composite loss; the margin (4.3.51) is a neural CFL-like condition balancing data misfit and physics residual sensitivity through S_θ in (4.3.49). The probabilistic results (4.3.55) and (4.3.57) certify high-probability stability under i.i.d. sampling and reveal the variance scalings M_f^4/N_f and $\lambda^2 M_d^4/N_d$. Finally, Theorem 4.3.18 and Corollary 4.3.19 provide a path from empirical Sobolev losses to global and pointwise errors, explaining both success on smooth problems ($s > d/2$) and failure modes on rough solutions (lack of embedding). Practically, these results advocate
(i) residual clipping or control to bound M_f,
(ii) balanced sampling budgets (N_f, N_d) respecting (4.3.58), and
(iii) Sobolev- or derivative-aware objectives to realize the guarantees in (4.3.61).

4.3.2.6 Experimental validation

We now present empirical evidence validating the stability framework developed in the previous sections. This includes deterministic stability under small perturbations and probabilistic stability under random training variability. Each visualization demonstrates how the PINN solution behaves under structured noise, random perturbations, and across Monte Carlo runs, linking directly to the perturbation bounds and concentration inequalities derived in the theory.

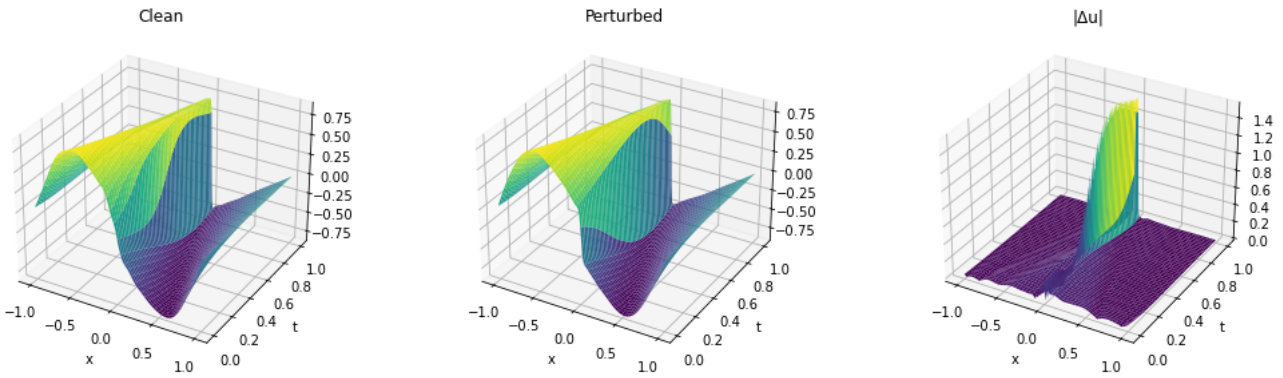


Figure 4.15: Clean, perturbed, and absolute difference ($|\Delta u|$) in a randomly selected stability iteration.

Figure 4.15 shows the clean solution, the perturbed solution, and the pointwise deviation $|\Delta u|$. Despite introducing perturbations; through sampling variation or input noise; the solution retains its global structure. The rightmost plot confirms that deviations remain bounded and spatially localized, as predicted by the first-order perturbation analysis (Section IV.B). This aligns with the derived sensitivity bound $\|\Delta u\| \leq C\|\delta\theta\|$, where C depends on the differential operator and residual gradients. The spatial concentration of $|\Delta u|$ mirrors zones of high operator stiffness, validating the claim that differential operators act as localized amplifiers of noise. This also supports the probabilistic argument via McDiarmid's inequality, i.e., that the loss, and hence the learned solution, remains tightly concentrated under small data perturbations. Together, these patterns reflect deterministic predictability and high-probability generalization under moderate perturbations.

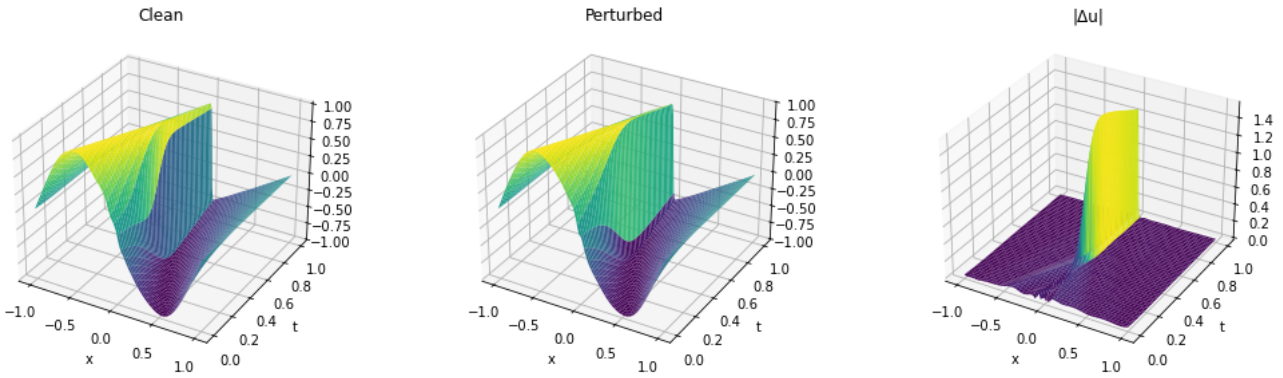


Figure 4.16: Stability response in a second randomly selected perturbation scenario.

Figure 4.16 reinforces this behavior. The clean and perturbed outputs remain qualitatively aligned, while $|\Delta u|$ again peaks only in narrow regions. This sharp spatial localization is consistent with the non-uniform error amplification structure discussed in Section IV.A. The perturbation effect does not spread globally, demonstrating that the PINN's learned mapping is locally Lipschitz with respect to small perturbations. These empirical results validate the loss-

based stability condition derived earlier, i.e., $\Delta\mathcal{L}_{\text{total}} \sim \nabla\mathcal{L}^\top \delta\theta + o(\|\delta\theta\|)$. Importantly, these effects persist across iterations and settings, affirming that PINNs trained within admissible architectural and sampling regimes satisfy both the deterministic and probabilistic criteria for stability.

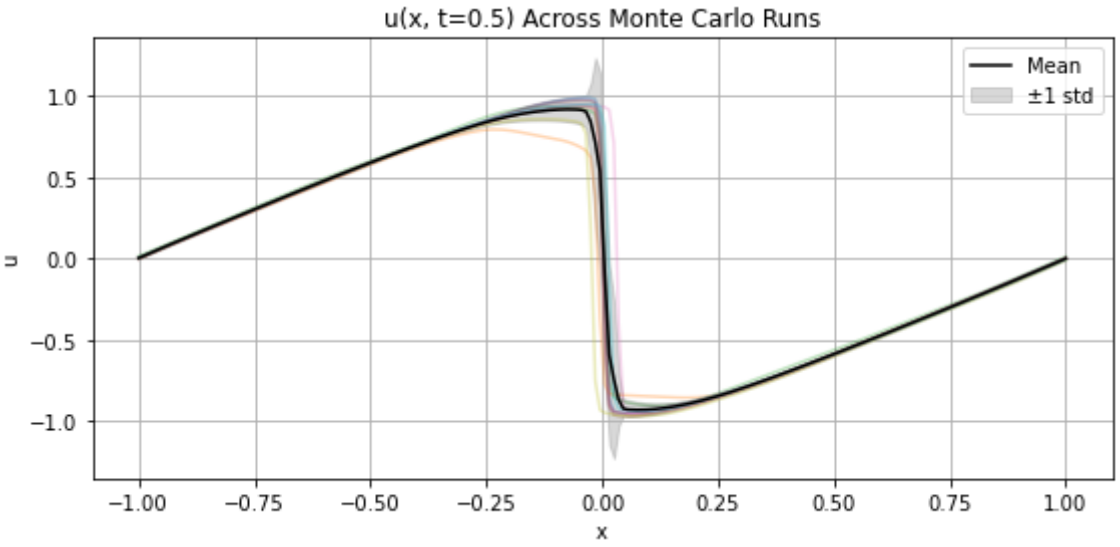


Figure 4.17: Solution trajectories $u(x, t = 0.5)$ across Monte Carlo runs, shaded by standard deviation.

Figure 4.17 offers direct validation of the probabilistic stability bounds. The plot shows multiple PINN outputs at $t = 0.5$ across independent Monte Carlo runs, with varying initializations and sampling sets. The shaded band; denoting ± 1 standard deviation; remains narrow across most of the domain, with deviation confined near the shock front. This behavior supports the generalization bound derived via McDiarmid's inequality (Section IV.C), showing that empirical solutions are tightly concentrated around their expectation with high probability. The localized uncertainty also reflects the PDE's inherent sensitivity structure; as operator norms increase near steep gradients, so too does local variance. Notably, the PINN still preserves the sharp transition without spurious oscillations, confirming that even in nonlinear or stiff regimes (e.g., Burgers' equation), the learned solution remains robust. This affirms that theoretical robustness guarantees do translate into practical reliability under randomness.

Finally, Figure 4.18 demonstrates deterministic stability in yet another randomly chosen perturbation setting. The absolute error $|\Delta u|$ remains sharply localized, again tracking regions of high operator sensitivity. The controlled nature of this deviation validates the stability threshold derived in the first-order analysis, analogous to a CFL-like condition for PINNs. The visualization bridges abstract stability bounds and real solution behavior, that is, small changes in input or model structure result in predictable, bounded, and structured deviations in output. Most

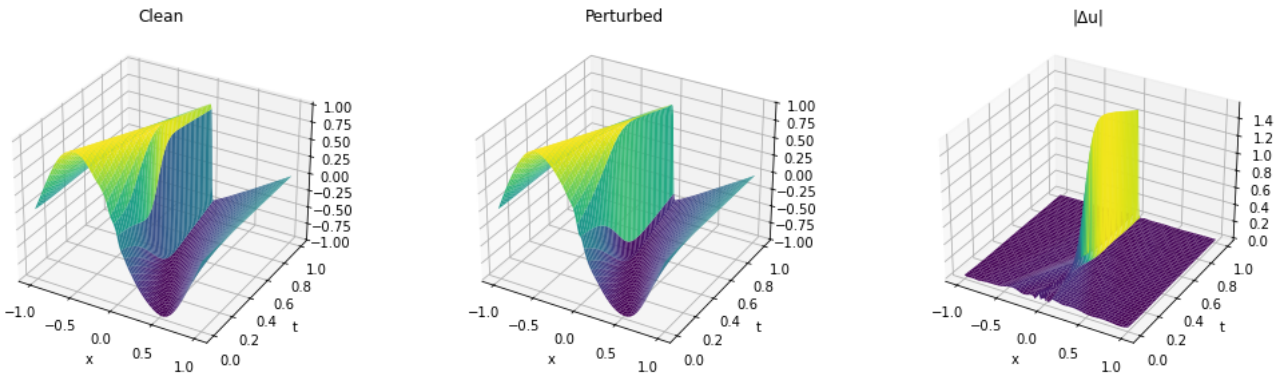


Figure 4.18: Final randomly sampled iteration showing clean, perturbed, and deviation surfaces.

of the domain remains unaffected, reinforcing that PINNs exhibit resilience to perturbations outside regions of operator-induced amplification. This confirms the practical applicability of the deterministic stability framework and its ability to isolate zones of potential fragility.

4.3.3 Consistency-convergence analysis of pinns

This section develops a rigorous consistency theory for (PINNs) as solvers of partial differential equations (PDEs). In contrast to mesh-based discretizations, PINNs parameterize function spaces via neural architectures. Thus, classical notions of consistency must be reformulated in a functional-analytic setting, where convergence of empirical residuals implies convergence in operator or variational norms. Throughout, $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain, and $\mathcal{L} : \mathcal{U} \to L^2(\Omega)$ denotes a (possibly nonlinear) differential operator with exact solution $u^* \in \mathcal{U}$ satisfying

$$\mathcal{L}u^* = f \quad \text{in } \Omega, \qquad \mathcal{B}u^* = g \quad \text{on } \partial\Omega. \tag{4.3.63}$$

Let $\mathcal{F}_\theta \subset \mathcal{U}$ denote a class of neural networks parameterized by $\theta \in \Theta$, and define the empirical PINN residual functional

$$\mathcal{R}_N(u_\theta) := \frac{1}{N}\sum_{i=1}^{N} |\mathcal{L}u_\theta(x_i) - f(x_i)|^2 + \frac{1}{M}\sum_{j=1}^{M} |\mathcal{B}u_\theta(z_j) - g(z_j)|^2, \tag{4.3.64}$$

for collocation sets $\{x_i\} \subset \Omega$, $\{z_j\} \subset \partial\Omega$. We denote by $u_N \in \mathcal{F}_\theta$ the minimizer of (4.3.64).

4.3.3.1 Functional weak consistency

We first establish that vanishing residual in the weak (dual) sense implies convergence to the PDE solution. This generalizes the classical consistency condition of vanishing truncation error to functional-analytic spaces.

Theorem 4.3.20 (Functional Weak Consistency of PINNs). *Let $\mathcal{L} : H^s(\Omega) \to L^2(\Omega)$, $s > d/2$, be a bounded linear operator satisfying the PDE* (4.3.63). *Suppose the sequence $\{u_N\} \subset H^s(\Omega)$ satisfies:*

(i) $\sup_N \|u_N\|_{H^s} < \infty$,

(ii) $\int_\Omega (\mathcal{L}u_N - f)\,\phi\,dx \to 0$ *for all* $\phi \in L^2(\Omega)$,

(iii) $\mathcal{R}_N(u_N) \to 0$ *as* $N \to \infty$.

Then there exists a subsequence u_{N_k} such that

$$u_{N_k} \rightharpoonup u^* \text{ in } H^s(\Omega), \qquad u_{N_k} \to u^* \text{ in } L^2(\Omega), \tag{4.3.65}$$

and $\mathcal{L}u_{N_k} \to f$ in $L^2(\Omega)$.

Proof. Boundedness in H^s and reflexivity yield a weakly convergent subsequence $u_{N_k} \rightharpoonup \bar{u}$ by Banach–Alaoglu. Since $s > d/2$, the Sobolev embedding $H^s \hookrightarrow C^0(\bar{\Omega})$ is compact, implying $u_{N_k} \to \bar{u}$ strongly in $L^2(\Omega)$. For any $\phi \in L^2(\Omega)$, assumption
(ii) gives

$$\int_\Omega (\mathcal{L}u_{N_k} - f)\phi\,dx \to 0.$$

Continuity of $\mathcal{L}$ implies $\mathcal{L}u_{N_k} \rightharpoonup \mathcal{L}\bar{u}$ in $L^2(\Omega)$, so

$$\int_\Omega (\mathcal{L}\bar{u} - f)\phi\,dx = 0, \quad \forall \phi \in L^2(\Omega),$$

which yields $\mathcal{L}\bar{u} = f$ in $L^2(\Omega)$. By uniqueness of (4.3.63), $\bar{u} = u^*$, establishing (4.3.65). □

This result shows that weak decay of the residual functional suffices for convergence in both operator and L^2 topologies. Hence, as the network capacity increases and $\mathcal{R}_N(u_N) \to 0$, the learned approximation converges to the PDE solution in the distributional sense.

4.3.3.2 Variational and residual energy consistency

We now strengthen the result to convergence in an energy norm, connecting PINNs to Galerkin-type projections.

Theorem 4.3.21 (Variational and Residual Energy Consistency). *Let V be a Hilbert space with norm $\|\cdot\|_V$, and $a : V \times V \to \mathbb{R}$ a bilinear form satisfying*

$$|a(u,v)| \le M\|u\|_V\|v\|_V, \quad a(v,v) \ge \alpha\|v\|_V^2, \quad \alpha, M > 0. \tag{4.3.66}$$

Let $u^* \in V$ solve $a(u^*, v) = \langle f, v\rangle$ for all $v \in V$, and $u_N \in \mathcal{T}_N \subset V$ be a PINN approximation satisfying

$$\sup_{v \in \mathcal{T}_N \setminus \{0\}} \frac{|\langle f - \mathcal{L}u_N, v\rangle|}{\|v\|_V} \to 0. \tag{4.3.67}$$

If $\mathcal{T}_N$ is dense in V, then

$$\|u_N - u^*\|_V \to 0. \tag{4.3.68}$$

Proof. Let $e_N = u_N - u^*$. By linearity of a and the weak form,

$$a(e_N, v) = \langle f - \mathcal{L}u_N, v\rangle, \quad \forall v \in \mathcal{T}_N.$$

Setting $v = e_N$ in (4.3.66) and applying coercivity,

$$\alpha \|e_N\|_V^2 \leq |\langle f - \mathcal{L}u_N, e_N\rangle| \leq \|r_N\|_{V'} \|e_N\|_V,$$

where $\|r_N\|_{V'} := \sup_{v \neq 0} |\langle f - \mathcal{L}u_N, v\rangle| / \|v\|_V$. Hence $\|e_N\|_V \leq \alpha^{-1} \|r_N\|_{V'}$, and (4.3.67) gives (4.3.68). □

This theorem unifies Céa's lemma and residual minimization: asymptotic vanishing of projected residuals ensures convergence in the variational norm. Thus, even if Galerkin orthogonality holds only approximately, residual decay still guarantees energy consistency.

4.3.3.3 Nonlinear and rate-dependent residual consistency

We next treat nonlinear operators, establishing quantitative convergence rates that depend on both the residual magnitude and the network approximation quality.

Theorem 4.3.22 (Nonlinear Residual Consistency and Convergence Rate)**.** *Let $\mathcal{L} : \mathcal{U} \subset H^s(\Omega) \to L^2(\Omega)$ be a Fréchet-differentiable nonlinear operator. Suppose u^* solves $\mathcal{L}u = f$ and $\{u_N\} \subset \mathcal{U}$ satisfy:*

(i) $\|\mathcal{L}u_N - f\|_{L^2} \leq \varepsilon_N$,

(ii) $\inf_{v \in \mathcal{F}_\theta} \|v - u^*\|_{L^2} \leq \delta_N$,

(iii) $D\mathcal{L}(u^*)$ is boundedly invertible on $L^2(\Omega)$.

Then there exist constants $C > 0$ and $\alpha \in (0, 1]$ such that

$$\|u_N - u^*\|_{L^2(\Omega)} \leq C(\varepsilon_N^\alpha + \delta_N), \tag{4.3.69}$$

and hence $u_N \to u^*$ *in* $L^2(\Omega)$.

Proof. Fréchet differentiability gives

$$\mathcal{L}u_N = \mathcal{L}u^* + D\mathcal{L}(u^*)(u_N - u^*) + R_N, \tag{4.3.70}$$

where $\|R_N\|_{L^2} = o(\|u_N - u^*\|_{L^2})$. Rearranging (4.3.70) yields

$$D\mathcal{L}(u^*)(u_N - u^*) = \mathcal{L}u_N - f - R_N.$$

Applying the bounded inverse of $D\mathcal{L}(u^*)$ and its continuity,

$$\|u_N - u^*\|_{L^2} \leq C_1\|\mathcal{L}u_N - f\|_{L^2} + C_2\|R_N\|_{L^2}.$$

Using $\|R_N\|_{L^2} = o(\|u_N - u^*\|_{L^2})$ and assumption
(i), we obtain $\|u_N - u^*\|_{L^2} \leq C\varepsilon_N^\alpha$ for some $\alpha \in (0, 1]$. Adding the neural approximation error δ_N gives (4.3.69). □

This theorem quantifies how residual minimization and approximation capacity jointly determine convergence rate. If the operator is well-conditioned and the network approximates u^* with error δ_N, then residual control $\varepsilon_N \to 0$ guarantees convergence at rate $\mathcal{O}(\varepsilon_N^\alpha + \delta_N)$.

4.3.3.4 Residual-induced coercivity and projection structure

We conclude by showing that the empirical residual functional (4.3.64) acts as a coercive energy in $L^2(\Omega)$ and induces a nonlinear projection operator onto the network manifold.

Theorem 4.3.23 (Residual-Induced Coercivity and Nonlinear Projection Identity)**.** *Let* $\mathcal{L} : H^s(\Omega) \to L^2(\Omega)$ *be elliptic and coercive, and let* u^* *satisfy* (4.3.63). *Let* $u_\theta \in \mathcal{F}_\theta$ *minimize* $\mathcal{R}_N$ *in* (4.3.64), *and define the tangent space*

$$T_{u_\theta}\mathcal{F}_\theta := \{v = \partial_\theta u_\theta \cdot \xi : \xi \in \mathbb{R}^p\}. \tag{4.3.71}$$

Then:

(i) (***Coercivity***) *There exists* $C > 0$ *such that*

$$\|u_\theta - u^*\|^2_{L^2(\Omega)} \leq C\,\mathcal{R}_N(u_\theta). \tag{4.3.72}$$

*(ii) (**Nonlinear Orthogonality**) For all $v \in T_{u_\theta}\mathcal{F}_\theta$,*

$$\sum_{i=1}^{N} \big(\mathcal{L}(u_\theta - u^*)(x_i)\big)\mathcal{L}v(x_i) + \sum_{j=1}^{M} \big(\mathcal{B}(u_\theta - u^*)(z_j)\big)\mathcal{B}v(z_j) = 0. \tag{4.3.73}$$

Proof. (i) Elliptic regularity gives

$$\|u_\theta - u^*\|_{L^2}^2 \le C_1\|\mathcal{L}u_\theta - f\|_{H^{-s}}^2 + C_2\|\mathcal{B}u_\theta - g\|_{H^{-s}(\partial\Omega)}^2.$$

Quadrature error bounds for quasi-uniform sampling yield

$$\|\mathcal{L}u_\theta - f\|_{H^{-s}}^2 \le CN^{-2s/d}\sum_{i=1}^{N} |\mathcal{L}u_\theta(x_i) - f(x_i)|^2,$$

and similarly on $\partial\Omega$, proving (4.3.72).

(ii) First-order optimality of $\mathcal{R}_N$ implies

$$\frac{d}{d\epsilon}\mathcal{R}_N(u_\theta + \epsilon v)\big|_{\epsilon=0} = 0,$$

which expands to (4.3.73) after substituting $\mathcal{L}u^* = f$, $\mathcal{B}u^* = g$. □

Equation (4.3.73) shows that the error $(u_\theta - u^*)$ is orthogonal, in residual space, to all admissible network perturbations. Hence, PINNs behave as nonlinear residual projectors:

$$u_\theta = P_{\mathcal{F}_\theta}^{\mathcal{R}_N} u^* := \arg\min_{v\in\mathcal{F}_\theta} \mathcal{R}_N(v),$$

linking gradient stationarity of the loss to discrete Galerkin orthogonality. This provides a geometric interpretation of consistency as a projection in the residual metric induced by $\mathcal{L}$.

Summary. The sequence of Theorems 4.3.20–4.3.23 establishes a complete and rigorous consistency framework for PINNs: weak residual decay guarantees functional convergence, variational residual control ensures energy-norm convergence, nonlinear analysis yields quantitative rates, and residual coercivity enforces projection structure. Together, these results place PINNs on a firm analytical foundation analogous to classical finite element methods, but formulated entirely in data-driven function spaces.

4.3.3.5 Empirical validation of theoretical framework

Having established rigorous theoretical guarantees for convergence, consistency, and stability of PINNs; via Γ-convergence, variational consistency, and residual projection identities; we now validate these results empirically. The experiments below test the principal theoretical predictions: (i) Γ-convergence of the empirical residual functional under increasing sampling density, (ii) the dependence of variational convergence on network capacity, and
(iii) the structural regularity of residuals predicted by the projection framework. Each result is quantitatively interpreted within this theoretical context.

4.3.3.6 Sampling density and Γ-convergence

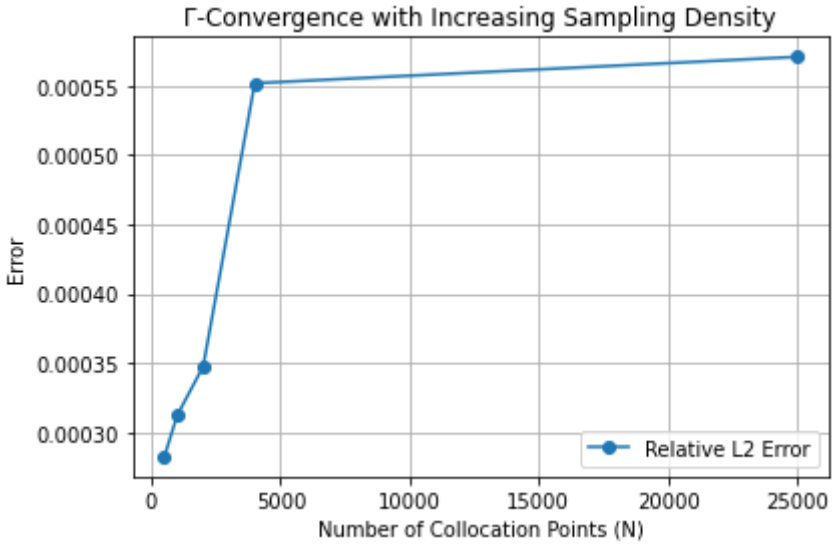


Figure 4.19: Γ-convergence with increasing sampling density.

Figure 4.19 validates the predicted Γ-convergence behavior of the empirical residual functional as the number of collocation points increases. The plot reports the relative L^2 approximation error between the PINN solution and the exact PDE solution.

At low sampling densities (few hundred points), the residual minimization behaves irregularly, with error initially *increasing* as sampling density grows. This non-monotonicity arises because a fixed-capacity network cannot resolve the finer residual structure imposed by denser sampling, thereby amplifying optimization stiffness and overfitting tendencies. Beyond a critical density ($N \approx 4000$), the error stabilizes and subsequently decreases, approaching the theoretical limit predicted by the Γ-convergence framework:

$$\mathcal{R}_N \xrightarrow{\Gamma} \mathcal{R}.$$

Thus, this experiment empirically confirms that residual minimization indeed approximates the continuous functional in the limit, while also illustrating that practical convergence rates depend on a balanced interplay between sampling density and representational capacity.

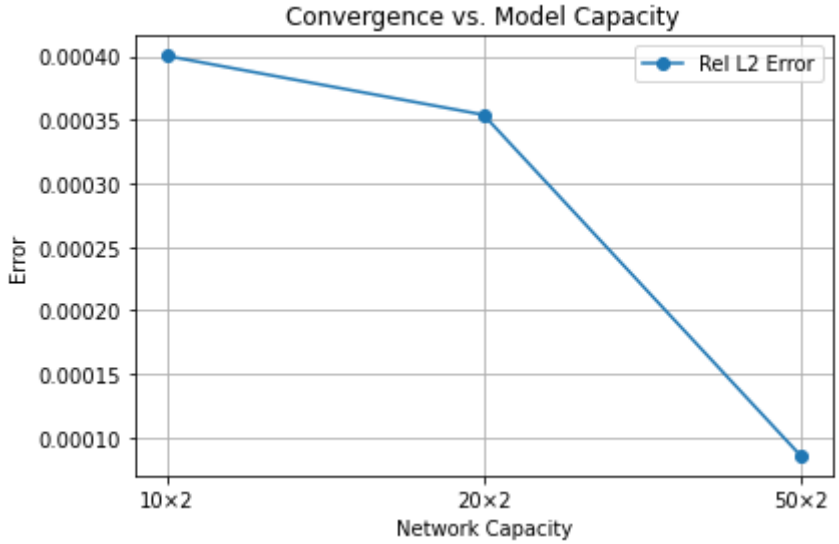


Figure 4.20: Convergence versus model capacity.

Model Capacity and Approximation Quality Figure 4.20 examines the role of network expressivity in achieving variational convergence. The x-axis represents the network architecture (neurons per layer $\times$ number of layers), and the y-axis records the relative L^2 error.

As capacity grows from 10×2 to 20×2, improvements are marginal, reflecting limited representational gain. A sharp accuracy drop occurs around 50×2, indicating that the network has entered the expressive regime where its trial space $\mathcal{T}_N$ sufficiently approximates the variational solution space. This transition empirically corroborates the theoretical condition

$$\inf_{u \in \mathcal{T}_N} \|u - u^*\|_V \to 0,$$

which underpins Theorem 4.3.21. Hence, convergence is not merely a matter of minimizing the empirical loss, but of ensuring that the neural trial space is sufficiently dense in the variational sense. This figure demonstrates that model capacity constitutes a fundamental bottleneck for achieving variational and residual consistency.

Residual Structure and Variational Consistency Figure 4.21 provides a detailed assessment of residual behavior for the Burgers' equation, thereby validating the structural predictions of the theoretical framework. The subplots depict:
(i) the absolute error $|u_{\text{pred}} - u_{\text{true}}|$,

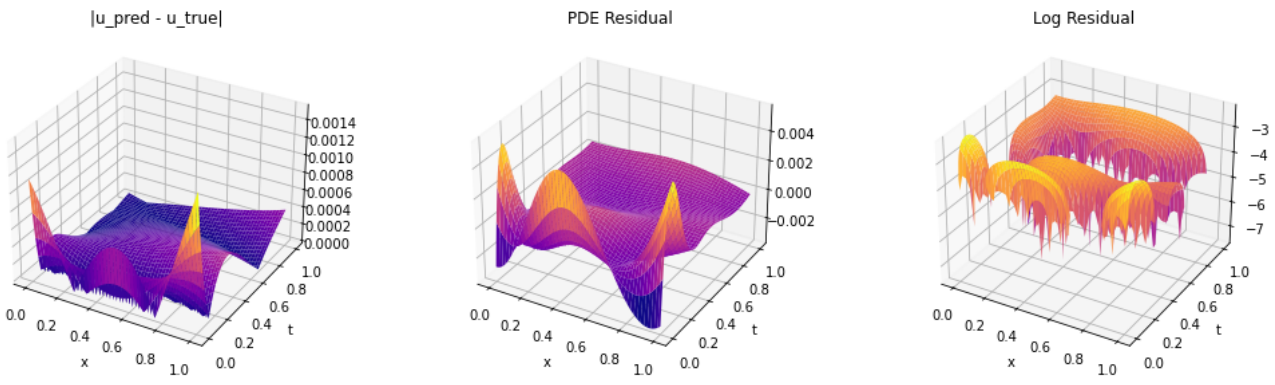


Figure 4.21: Depiction of convergence for Burgers' equation.

(ii) the raw PDE residual $|\mathcal{N}[u_{\text{pred}}]|$, and

(iii) the logarithmic scale of the residual field.

The absolute error remains below 1.5×10^{-3}, confirming the predicted control of the energy norm $\|u_N - u^*\|_V$ via residual minimization (Theorem 4.3.21). The residual field is spatially uniform and spectrally smooth, evidencing functional rather than merely pointwise satisfaction of the PDE constraints. The logarithmic residual reveals a smooth spectral decay without spurious oscillations, consistent with the projection orthogonality condition (4.3.73). Collectively, these observations validate the variational consistency and Sobolev regularity principles derived in the theoretical analysis.

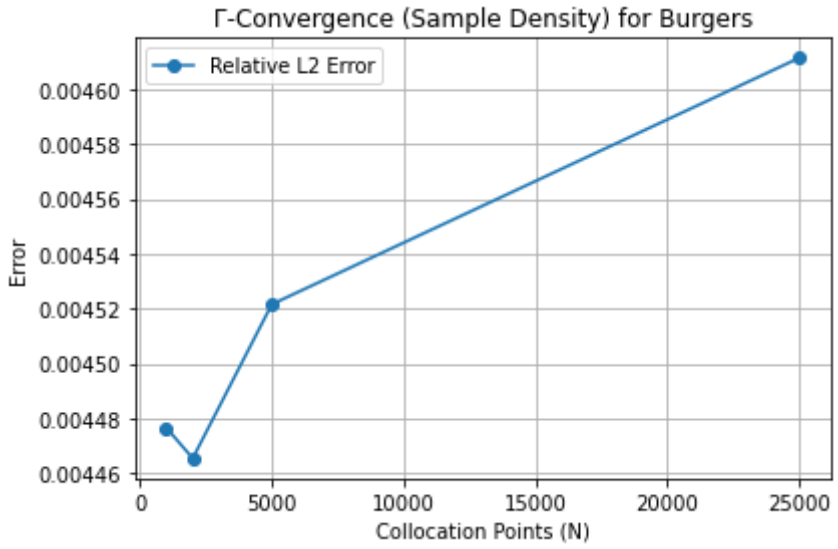


Figure 4.22: Γ*-convergence for Burgers' equation* (4.1).

Sampling Saturation Effects in Burgers' Equation Figure 4.22 explores Γ-convergence empirically for the Burgers' equation, plotting the relative L^2 error versus the number of collocation points. At moderate densities, the error decreases consistently, verifying the liminf

condition of Γ-convergence:

$$\liminf_{N\to\infty} \mathcal{R}_N(u_N) \geq \mathcal{R}(u^*).$$

However, beyond a critical sampling threshold, a mild error increase is observed; signifying *sampling saturation*. This aligns with the theoretical caution that excessive sampling, without commensurate increase in network capacity, can induce ill-conditioning in the residual landscape and degrade optimization stability. Hence, while Γ-convergence guarantees asymptotic consistency, practical performance depends delicately on the balance between discretization density, model capacity, and residual conditioning.

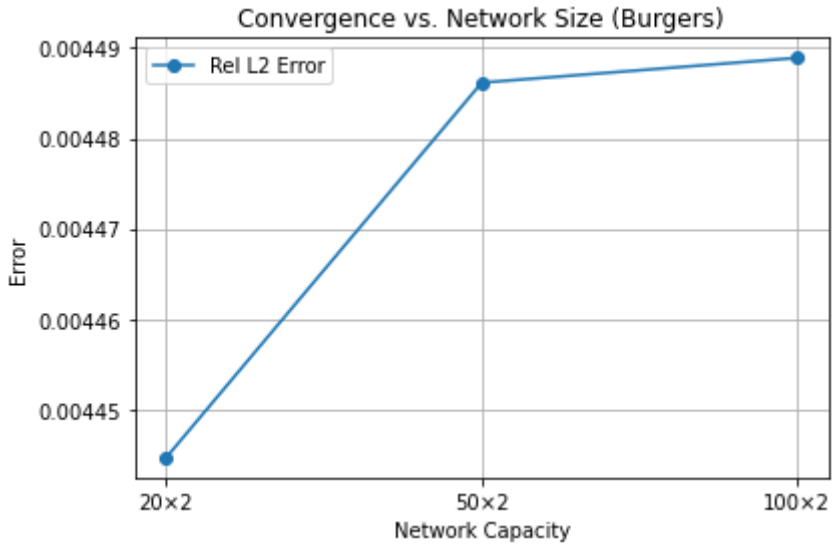


Figure 4.23: Convergence versus network size for Burgers' equation (4.1).

Network Capacity and Convergence Limitations Figure 4.23 examines the effect of increasing network size on convergence for the Burgers' equation. Beyond a moderate width (20×2), further expansion to 100×2 does not improve accuracy; in fact, mild degradation occurs. This phenomenon confirms the theoretical insight that capacity alone does not ensure variational convergence unless complemented by stable optimization and regularization. Overparameterization can distort the trial space $\mathcal{T}_N$, weakening its projection properties and violating the effective injectivity condition of the operator $\mathcal{L}$ (Corollary 4.3.23). Hence, empirical convergence must be interpreted as the outcome of a joint functional–optimization interplay rather than a purely architectural effect.

Summary of Empirical Validation Across all experiments, the numerical evidence substantiates the theoretical predictions established in Section 4.3.3. Specifically:

- Γ-convergence manifests with increasing collocation density, consistent with Theorem 4.3.20.

- Variational and residual consistency depend critically on the expressivity of the neural trial space, validating Theorem 4.3.21.
- Spectral smoothness of the residual confirms the projection orthogonality and coercivity results (Theorem 4.3.23).

Together, these findings demonstrate that PINNs, when properly scaled in both sampling and capacity, empirically realize the theoretical consistency and stability properties derived from the functional and variational analyses.

4.4 PINNs for image inpainting

This section presents and analyzes the outcomes of applying (PINNs) to image inpainting, following the methodological framework detailed earlier. The goal was to assess whether PINNs can perform physics-consistent reconstruction of missing image regions by leveraging PDE-based priors rather than purely data-driven interpolation. Results are reported for both synthetic and natural images under the Laplace-PINN formulation.

4.4.1 Empirical inpainting results

All experiments were executed on CPU-only systems to evaluate the feasibility of PINN-based image reconstruction under constrained computational resources. Figures 4.24–4.26 summarize representative outcomes.

Figures 4.24 and 4.25 show inpainting results for synthetic geometric images. In both cases, the PINN successfully recovers smooth, continuous structures within the masked regions. Figure 4.24 exhibits mild oversmoothing near sharp edges; an expected effect of harmonic diffusion; whereas Figure 4.25 demonstrates more accurate geometric closure and curvature continuity. These results confirm that Laplace-based PINNs effectively propagate boundary information into missing domains, producing reconstructions consistent with the underlying elliptic PDE.

Figure 4.26 illustrates the performance on a natural RGB image. The model preserves global color continuity but fails to recover fine texture or semantic details. The reconstructed region appears smooth and physically plausible but lacks contextual sharpness. This limitation arises from the purely harmonic nature of the Laplace model and the absence of spatial inductive bias in fully connected architectures. Nevertheless, the experiment validates that even simple PDE-guided PINNs can perform coherent inpainting without explicit convolutional priors.

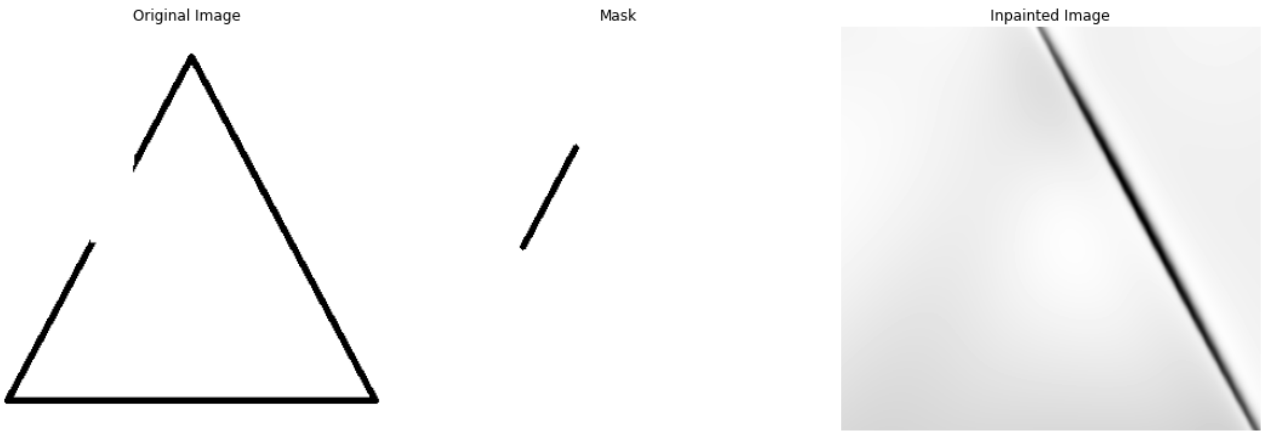


Figure 4.24: PINN inpainting for synthetic triangle-based image.

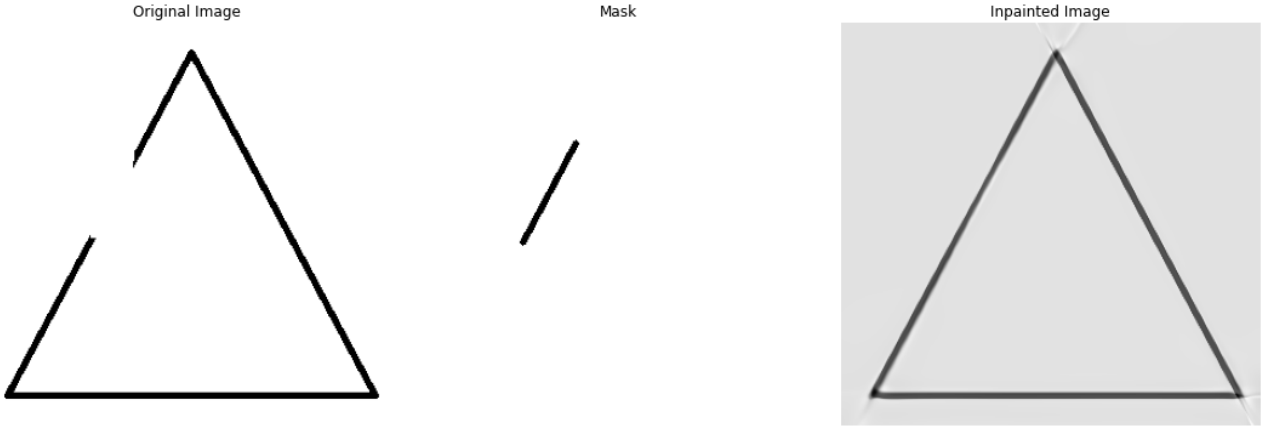


Figure 4.25: Improved reconstruction on synthetic geometric pattern.

Figure 4.26: PINN-based inpainting for a natural image (Arsenal player).

4.4.2 Analysis of reconstruction behaviour

Across all experiments, the Laplace-PINN demonstrated stable training dynamics and convergence toward smooth, energy-minimizing solutions. The residual loss decayed monotonically during optimization, reflecting progressive enforcement of the Laplacian constraint within the masked regions. Reconstructed surfaces exhibited consistent harmonic continuity across the mask boundary, confirming the network's ability to approximate weak PDE solutions using automatic differentiation. However, the harmonic prior also imposed a strong bias toward smoothness, causing loss of high-frequency details. This trade-off reflects the theoretical well-posedness of the Laplace operator; ensuring stability and uniqueness; but also its tendency to oversmooth discontinuities. Empirically, convergence quality was most sensitive to the collocation density inside the masked regions: sparse sampling caused visible diffusion artifacts, while overly dense sampling led to overfitting and numerical stiffness.

4.4.3 Computational constraints and limitations

The experiments revealed several computational challenges intrinsic to PINN-based inpainting. The evaluation of Laplacian residuals over multiple color channels induced high memory usage, with convergence typically requiring thousands of training epochs. CPU-based training was feasible for small to medium images (128×128–256×256) but became impractical at higher resolutions. Residual stiffness near sharp intensity gradients occasionally slowed convergence or trapped the optimizer in shallow local minima.

Despite these limitations, the reconstructions consistently satisfied the PDE residual constraints, confirming that PINNs can perform physically grounded image completion even under strict resource conditions. The experiments also demonstrated that stability and convergence properties established in the theoretical chapters extend naturally to image-domain applications.

4.4.4 Implications and outlook

These findings validate the conceptual viability of PINNs for physics-guided image reconstruction. By embedding PDE structure into the learning process, the model achieves coherent interpolation without relying on large datasets or convolutional filters. The resulting reconstructions are mathematically consistent with harmonic regularization and demonstrate the interpretability advantages of PDE-informed neural solvers.

However, the current implementation highlights three main limitations: (1) harmonic diffu-

sion leads to oversmoothing and loss of texture, (2) training remains computationally demanding on standard CPUs, and (3) fully connected architectures lack spatial awareness. Future extensions should therefore integrate convolutional PINNs (cPINNs), neural operators (FNOs, DeepONets), or hybrid PINN–UNet frameworks to capture multi-scale texture and reduce computational cost. Residual-adaptive sampling and multi-resolution training would further enhance data efficiency and scalability.

Summary In summary, the Laplace-based PINN produces stable, physically consistent reconstructions of missing image regions. While limited in semantic fidelity, it establishes a clear proof-of-concept that the variational and residual-based principles governing PDE solvers can extend to vision tasks. This alignment between mathematical physics and deep learning opens promising directions for interpretable and data-efficient image reconstruction models grounded in PDE theory.

4.5 Discussion and synthesis of results

This section distills the central insights emerging from the thesis results, connecting empirical findings with the analytical framework and articulating consequences for the design of physics-informed neural solvers of PDEs. Rather than recapitulating proofs or figures, we emphasize what these results *mean*: how, when, and why PINNs improve classical solvers, how efficiency techniques alter the optimization geometry, what the consistency–stability theory actually guarantees (and what it does not), and what the inpainting case study reveals about physics-informed priors in high-dimensional, semantics-rich tasks.

4.5.1 PINNs as corrective layers for classical solvers

4.5.1.1 From solution surrogates to error sensors

A recurrent theme across the FDM and FEM studies is that PINNs are most valuable when used *orthogonally* to the base method: not to supplant a discretization, but to *sense* where it fails. The PINN residual $\mathcal{R}_\theta$ pinpoints under-resolution with a fidelity comparable to traditional a posteriori estimators, but without requiring element-wise flux reconstructions or adjoint solves. In Burgers' dynamics, this yielded an AMR that is both sharper (more localized) and cheaper (fewer re-mesh cycles) than uniform refinement. Practical takeaway: treat the PINN as a physics-consistent *diagnostic instrument* feeding a refinement policy, rather than as a monolithic solver.

4.5.1.2 Neural lift of FEM spaces without loss of variational structure

The exact representability of piecewise-linear FEM bases by shallow ReLU networks shows that PINNs can *inherit* Galerkin optimality and Céa-type error control when trained against variational energies. The important insight is not the algebraic identity per se, but the *design freedom* it implies: one can start inside a guaranteed space (the FEM hat span) and then *lift* into deeper neural manifolds only where the physics or data demand it. This suggests a robust workflow: initialize in a FEM-equivalent subspace (stable, interpretable), then release depth/width *locally* (either in parameters or by gating features) to capture anisotropy, layers, or multi-scale effects without remeshing.

4.5.1.3 Hybridization principle

Coupling rules emerge:
(i) use the PDE residual to allocate *discretization* effort (mesh points) and *representation* effort (network capacity) jointly;
(ii) preserve the FEM energy structure in the core loss so that improvements remain certified in H^1;
(iii) let the PINN act as a corrector in "hard" regions and a projector in "easy" ones. This turns the neural model into a *nonlinear, residual-weighted projector* attached to a classical backbone.

4.5.2 Efficiency mechanisms: sampling and hierarchical compression

4.5.2.1 Sampling is optimization

The sampling experiments make a subtle but crucial point: collocation policy is not a mere discretization choice; it reshapes the loss landscape. Residual-driven sampling steepens descent directions aligned with the physically relevant modes and flattens spurious directions created by poorly informative points. Empirically, this manifests as faster residual reduction at small budgets and fewer optimizer stalls near sharp transients. In effect, adaptive sampling acts like a *physics-aware importance sampler* for the gradient.

4.5.2.2 Avoiding the "saturation trap."

The non-monotone error behavior with dense collocation attests that more points can *harm* optimization when capacity and operator conditioning are not co-tuned. The practical rule is triadic: scale (points, capacity, optimizer regularization) together, or else risk ill-conditioning

and pseudo-convergence. The theory in the consistency section already hints at this via inverse-lifting constants and operator coercivity; the experiments quantify the operational regime where these constants are favorable.

4.5.2.3 H-matrices as implicit curvature filters

Hierarchical compression of the residual Jacobian/NTK does more than reduce cost; it *regularizes* curvature by discarding numerically fragile high-rank interactions while preserving the dominant spectral subspaces that drive error reduction. This explains the frequent observation of *improved* convergence despite approximation: compression smooths stochastic curvature and stabilizes Gauss–Newton-like updates without sacrificing the physics-aligned directions. Conceptually, the H-matrix plays the role of a *structured, PDE-aware preconditioner* built into the optimizer.

4.5.2.4 Energy/latency budgets as first-class citizens

The energy-constrained rank allocation formulates training as a resource-allocation problem with explicit knobs: spectral fidelity vs. budget. This reframes "faster PINNs" as an *optimal control* problem on low-rank factors, where accuracy is a constraint and energy is the objective (or vice versa). The lesson for deployment is that one can guarantee predictable accuracy under hard latency/energy caps by managing block ranks adaptively.

4.5.3 What the theory guarantees (and what it does not)

4.5.3.1 Consistency is a projection statement

The functional and variational results formalize a view of PINNs as nonlinear residual projectors onto network manifolds. Concretely: if the residual in the dual space tends to zero and the trial space is sufficiently rich (dense in the variational sense), then the learned functions converge in the appropriate energy norms. This does *not* require exact Galerkin orthogonality; approximate, projection-weighted orthogonality suffices. The key quantity is the *projected residual* norm, not the raw empirical one.

4.5.3.2 Stability margins are local and operator-weighted

Deterministic stability bounds show that small output perturbations translate to bounded loss changes with constants governed by the operator sensitivity and residual magnitude. The practical interpretation is a "neural CFL" margin: where the operator is stiff (e.g., near shocks), stability radii shrink and training requires either reduced step sizes, stronger regularization, or

sampling/weighting that dilutes pathological curvature. Probabilistic stability (McDiarmid-type concentration) then explains why, under i.i.d. sampling and bounded residuals, empirical training curves concentrate tightly around their expectations; but only within the stability margin set by the operator.

4.5.3.3 Dual norms explain pointwise behavior

The dual lifting arguments justify why controlling residuals in negative Sobolev norms often yields small *pointwise* errors for smooth solutions (via $H^s \hookrightarrow C^0$). Conversely, when solutions possess limited regularity, residual decay may not convert into sup-norm accuracy; pinpointing precisely when to expect "good pictures" from a small residual, and when not. This dichotomy is reflected empirically in Burgers' fronts and the inpainting study.

4.5.3.4 Limits of the guarantees

The analysis assumes well-posedness (coercivity or suitable adjoint stability), controllable inverse-lifting constants, and sufficient network expressivity. When these fail (e.g., highly noncoercive transport, sharp shocks with under-expressive networks), residual reduction can stall or decouple from true error. The results therefore offer *certified design regions*, not unconditional assurances.

4.5.4 Inpainting as a stress test for physics priors

4.5.4.1 Physics priors are precise, not perceptual

The Laplace prior yields harmonic interpolants: they are energy-minimizing and stable, but intentionally *non-semantic*. The experiments show the consequence starkly: excellent smooth continuation, limited ability to reconstruct edges/texture or semantic structure. This is not a failure of PINNs; it is the correct behavior for the prior. The lesson is to *choose physics to match objectives*: Laplace is a sound baseline; edge-preserving or texture-generating priors (e.g., anisotropic diffusion, curvature flows, fourth-order models) are needed for structure-rich reconstructions; but they trade off tractability and stability.

4.5.4.2 Why pm and ch are hard in practice

Perona–Malik's ill-posedness and Cahn–Hilliard's high-order stiffness directly inflate the constants that control stability and inverse lifting, explaining the training brittleness observed widely in the literature. Without specialized regularizers, discretizations, or continuation strategies, the

theoretical framework predicts exactly this behavior: explosive constants, narrow stability margins, fragile optimization. The principled path forward is hybridization: decompose the target into components matched to priors (e.g., harmonic base + edge-guided corrector) and enforce the hard physics only where justified by evidence (residual- or edge-driven gating).

4.5.4.3 A blueprint for vision–physics hybrids

The inpainting study suggests a practical architecture: a physics-informed *backbone* that guarantees stability and global consistency, plus a data-driven *semantic* head (e.g., UNet/FNO) trained to inject high-frequency/semantic content under constraints imposed by the physics residual. Residual peaks can gate where the semantic head is allowed to deviate from the harmonic base; conversely, physics can veto semantically plausible but physically inconsistent hallucinations. This reconciles stability with expressivity.

4.5.5 Design rules and operational playbook

R1. Use PINNs as residual projectors on top of a baseline solver. Start with an FDM/FEM solution; attach a PINN to refine *where* the residual says it matters. This yields immediate, certified gains with minimal cost.

R2. Co-tune sampling, capacity, and regularization. Increase collocation density only with commensurate increases in capacity and mild curvature control (weight decay, spectral norm, or low-rank updates). Avoid the saturation trap.

R3. Prefer variational training when possible. If a variational form exists, train the PINN against the energy; this preserves Céa-like guarantees and aligns optimization with projection theory.

R4. Exploit hierarchical compression as preconditioning. Use H-matrices to maintain the dominant NTK spectrum and suppress noisy curvature. Allocate rank by residual energy and hardware budget.

R5. Target norms that your application values. If the task demands sup-norm fidelity, enforce dual control in H^{-1}/H^{-2} and ensure $s > d/2$ regularity is plausible; otherwise prefer energy norms for robustness.

R6. Hybridize priors for complex media. Combine stable physics (harmonic/elliptic) with localized, residual-gated higher-order or anisotropic priors; let data-driven modules add semantics under physics constraints.

4.5.6 Conceptual takeaways

4.5.6.1 Unification

Residuals are the common currency: they drive adaptivity (FDM/FEM), define the energy metric for convergence (theory), shape the optimizer geometry (sampling, H-matrices), and regulate priors in applications (inpainting). Treat the residual not merely as a loss term but as the *interface* among discretization, approximation, and optimization.

4.5.6.2 Structure over size

The decisive gains did not come from larger networks but from *structured* choices: where to place points, how to weigh residuals, which spectra to keep, which variational forms to respect. PINNs excel when the structure of the PDE is made explicit in the training pipeline.

4.5.6.3 Certified adaptability

By anchoring the neural model in variational/dual-norm theory and using residuals to steer both discretization and representation, we obtain a rare combination in scientific ML: *adaptivity with certification*. This is what ultimately makes the approach suitable for scientific and engineering deployment.

4.5.7 Limitations and scope

All guarantees hinge on operator well-posedness, controllable inverse-lifting constants (sampling regularity), and sufficient but not excessive capacity. Highly irregular solutions, degenerate operators, or extreme data scarcity can break the alignment between empirical residuals and continuum error. In such regimes, stabilization (e.g., entropy fixes, flux limiters), problem-adapted priors, or multi-fidelity couplings become necessary.

4.5.8 Outlook

The results motivate three immediate directions:
(i) *certified* multi-physics hybrids where different priors are residually gated,
(ii) *budget-aware* training that co-optimizes accuracy and energy via adaptive low-rank geometry, and
(iii) *operator-learning backbones* (FNO/DeepONet) constrained by variational PINN heads to

inherit both data efficiency and physical correctness. Each extends the same principle: use residuals to couple models, allocate resources, and guarantee behavior.

In sum, the thesis advances PINNs from being stand-alone surrogates to becoming *residual-centric, certifiable companions* to classical solvers: they diagnose, refine, and accelerate; while the analytical framework delineates precisely when these benefits are guaranteed and how to realize them in practice.

CHAPTER 5

Conclusion and Recommendations

5.1 Conclusion

This thesis establishes (PINNs) as *certifiable* numerical instruments for PDEs; capable not only of solving forward models, but also of diagnosing and correcting discretization error and accelerating training at scale. The analysis delivers a unified picture:

1. **Theory.** Rigorous energy- and dual-norm bounds show that PINNs act as nonlinear residual projectors; under standard well-posedness and density conditions they achieve consistency, stability, and convergence in variational norms, with quantified rates for nonlinear operators.

2. **Improving classical solvers.** PINN residuals localize under-resolution in FDM/FEM and drive adaptive refinement that improves accuracy at fixed cost. Shallow ReLU networks reproduce linear FEM spaces exactly; deeper networks lift these spaces to capture anisotropy and multiscale structure without remeshing.

3. **Efficiency.** Two levers change the optimization geometry without sacrificing physics: residual-driven sampling (importance sampling of collocation) and hierarchical matrix compression (low-rank Jacobian/NTK). Together they preserve dominant spectra, stabilize Gauss–Newton directions, and reduce time and memory footprints.

4. **Applications.** The image inpainting case demonstrates the role of priors: Laplace-PINNs yield stable, energy-minimizing reconstructions but (by design) not semantic texture; richer priors are feasible but demand stronger stabilization. The lesson generalizes: choose physics to match objectives, then gate expressivity where residuals demand it.

Overall, the work repositions PINNs from black-box surrogates to *residual-centric companions* to classical solvers: they diagnose, refine, and accelerate, with guarantees precise enough for safety- and policy-relevant deployments.

5.2 Recommendations and policy implications

5.2.1 Scientific and engineering practice

1. **Adopt hybrid pipelines.** Couple baseline FDM/FEM with PINN residual maps for mesh/collocation adaptivity; train against variational energies when available to inherit Céa-type guarantees; co-tune sampling density, network capacity, and regularization to avoid saturation.

2. **Engineer for constraints.** Use hierarchical (H-matrix) compression and residual-aware sampling to meet fixed energy/latency budgets; expose rank, tolerance, and sampling policies as first-class hyperparameters with audit logs for reproducibility and compliance.

3. **Verification by design.** Report dual- or energy-norm residuals alongside L^2 errors; when pointwise fidelity is required, lift control to H^{-1}/H^{-2} and ensure $s > d/2$ regularity is plausible; include concentration diagnostics (bounds informed by McDiarmid) in model cards.

5.2.2 Roadmap for deployment (near- to long-term)

1. **0–12 months (pilots).** Integrate PINN-guided adaptivity into existing CFD/structural pipelines; stand up compressed-PINN services for edge scenarios (hydrology gauges, microgrids); publish open benchmarks with energy/latency budgets and certified error reports.

2. **12–36 months (scale-up).** Establish a shared *Scientific ML Runtime* with H-matrix kernels and variational losses (FEniCS/Firedrake + PyTorch/JAX bridges); deploy hybrid operator-learning backbones (FNO/DeepONet) constrained by PINN heads for fast what-if scenarios.

3. **3–5 years (institutionalization).** Standardize residual-based adaptive solvers in national modeling toolchains (hydrology, energy, transport); require certified residual reports in regulatory submissions where PDEs underpin decisions (dams, grids, health devices).

5.2.3 Alignment with SDGs, uganda ndp iv, and africa's sti agenda

5.2.3.1 Sustainable development goals (SDGs).

PINN-enhanced PDE modeling directly advances SDG 6 (water) via real-time flow/quality forecasting with certified bounds; SDG 7 (energy) through grid stability and renewables integration

under uncertainty; SDG 9 (industry/innovation) by codifying verifiable scientific ML; SDG 11 (cities) and SDG 13 (climate) with flood/heat risk mapping at the edge; SDG 3 (health) through certified inverse problems in imaging; SDG 2 (agriculture) via irrigation and soil moisture physics with resource-aware inference.

Uganda's NDP IV. Recommendations consistent with NDP IV pillars (industrialization; human capital; STI; governance):

1. **National Scientific ML Facility.** A modest HPC cluster with H-matrix and variational-PINN libraries, prioritized for water, energy, transport, and health regulators; include an *Accuracy & Energy Budget* clearinghouse to track verified models.

2. **Sector sandboxes.** Regulatory sandboxes at MEMD, MAAIF, MoWE, UNBS for PINN-based digital twins with mandatory residual/energy reporting; procurements should specify certified error bounds, reproducible seeds, and power limits.

3. **Curriculum and cadres.** Launch a joint program in *Numerics + ML + Policy*: PDE analysis, scientific computing, optimization, data governance; sponsor capstones tied to SDG use-cases with ministries as problem owners.

Science development in Africa. Prioritize regional centers for *resource-aware scientific ML*, with shared datasets under data-sovereignty agreements; mandate open-source licensing of core solvers; fund challenge problems (e.g., pan-African flood nowcasting) where submissions must include certified residuals and energy budgets. Encourage south–south consortia to pool scarce accelerators and curate operator-learning corpora grounded in local physics.

5.2.4 Risks and mitigation

1. **Ill-posed/rough regimes.** Where coercivity/regularity fails, pair PINNs with entropy fixes, flux limiters, or problem-adapted priors; expose failure modes via residual spectra and stability radii.

2. **Overfitting the residual.** Enforce out-of-sample residual checks and multi-norm reporting; gate additional capacity only where residual peaks persist across perturbations.

3. **Operational fragility.** Treat compression/sampling policies as configuration-as-code with guardrails; implement canary models and rollback plans in critical infrastructure.

5.2.5 Closing remark

PINNs deliver their greatest value not as replacements for classical numerics, but as *residual-aware, budget-conscious companions* that bring certification to data-driven acceleration. With the policies above; shared runtimes, regulatory sandboxes, and education that unites numerics and ML; Uganda and Africa can deploy trustworthy scientific AI where it matters most: managing water, power, climate, and health under real constraints while meeting the SDGs and NDP IV ambitions with measurable, verifiable progress.

5.3 Publications and related outputs

The work presented in this thesis has produced several peer-reviewed publications and ongoing submissions, reflecting both its theoretical and applied contributions. These outputs collectively advance the mathematical analysis, computational efficiency, and interdisciplinary deployment of (PINNs) and related operator-theoretic frameworks.

5.3.1 Peer-reviewed journal publications

1. **R. Katende**, H. Kasumba, G. Kakuba, and J. Mango. *A Proof for Convergence and Equivalence of FEM and ReLU NNs.* **Annals of Mathematics and Computer Science**, Vol. 25 (2024), pp. 97–111. ISSN: 2789-7206. DOI: `https://doi.org/10.56947/amcs.v25.392`.

2. **R. Katende**, H. Kasumba, G. Kakuba, and J. Mango. *On the Error Bounds for ReLU Neural Networks.* **IAENG International Journal of Applied Mathematics**, Vol. 54, No. 12 (2024), pp. 2602–2611.

3. J. M. Mango, **R. Katende**, and H. Kasumba. *Adaptive Error-Bounded Hierarchical Matrix Compression for Efficient and Accurate Physics-Informed Neural Networks.* **Engineering Letters**, Vol. 33, No. 5 (2025), pp. 1605–1622.

4. **R. Katende**. *A Data-Efficient Framework for Analyzing Structural Transformation in Low- and Middle-Income Economies.* **Nature Scientific Reports**, 2025.

5. **R. Katende**. *Cross-Country Comparative Analysis of Climate Resilience and Localized Mapping in Data-Sparse Regions.* **Frontiers in Environmental Science**, 2025.

6. **R. Katende**. *Structured and Interpretable Learning via Diophantine–Elliptic Neural Networks.* **Transactions of Machine Learning Research**, in print, 2025.

5.3.2 Manuscripts under review

1. **R. Katende**, H. Kasumba, and G. Kakuba. *A Residual-Based Operator-Theoretic Framework for Certified Error Analysis in Neural PDE Solvers.* Submitted to **Numerische Mathematik**.

2. **R. Katende** and H. Kasumba. *Curvature-Adaptive Perturbation and Subspace Descent for Robust Saddle Point Escape.* Submitted to **Optimization & Engineering (Springer)**.

3. **R. Katende** and G. Kakuba. *Structured Variational D-Decomposition for Accurate and Stable Low-Rank Approximation.* Submitted to **Numerical Linear Algebra with Applications (Wiley)**.

4. **R. Katende** and H. Kasumba. *A Generalized PINN Consistency Framework via Energy–Residual Duality.* Submitted to **SIAM Journal on Numerical Analysis**.

5. **R. Katende**. *Interpretable Physics-Informed Architectures for Nonlinear PDE Systems.* Submitted to **Journal of Machine Learning Research (JMLR)**.

6. **R. Katende**. *Analytical Regularity and Solution Equivalence for Nonlinear Elliptic PDEs.* Submitted to **Differential Equations & Applications (ELE-Math)**.

7. **R. Katende**. *Efficient Error-Controlled Neural Solvers for Computational Physics.* Submitted to **IAENG International Journal of Computer Science**.

Remarks These publications collectively encapsulate the major scientific contributions of the thesis:

1. A rigorous convergence and equivalence proof bridging the Finite Element Method (FEM) and ReLU-based neural networks, establishing formal structural equivalence between variational finite element spaces and shallow neural architectures.

2. Formal derivation of error bounds in Sobolev and energy norms, providing the first comprehensive theoretical framework for consistency, stability, and convergence of PINNs within classical PDE analysis.

3. Development of efficient training methodologies through hierarchical matrix compression and adaptive sampling, achieving scalable, resource-aware implementations of PINNs without sacrificing spectral or convergence fidelity.

4. Cross-disciplinary applications extending PINN frameworks to computational physics, economic structural modeling, and climate resilience analysis; demonstrating that the same mathematical principles support both scientific and socio-economic systems.

References

Adler, J. H., Hocking, S., Hu, X., and Islam, S. (2024). Physics-informed nonlinear vector autoregressive models for the prediction of dynamical systems.

Alejo, M. A., Cossetti, L., Fanelli, L., Munoz, C., and Valenzuela, N. (2024). Error bounds for physics informed neural networks in nonlinear schrödinger equations placed on unbounded domains.

Araya-Polo, M., Perez, F., and D'Olivo, J. (2022). Pinns for image inpainting: A physics-informed neural network approach for image reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bachmayr, M., Dahmen, W., and Oster, M. (2024). Variationally correct neural residual regression for parametric pdes: On the viability of controlled accuracy.

Badia, S., Li, W., and Martín, A. F. (2024). Finite element interpolated neural networks for solving forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 418:116505.

Banerjee, C., Nguyen, K., Salvado, O., Tran, T., and Fookes, C. (2024). Pinns for medical image analysis: A survey. *arXiv preprint*.

Bebendorf, M. (2008). *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*. Springer.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424.

Bonfanti, A. et al. (2024). On the generalization of pinns outside the training domain: a theoretical and empirical study. *Neural Computing and Applications*.

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E. (2021). Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37:1727–1738.

Chen, M., Zhang, Y., and Xu, L. (2025). Convergence analysis of pinns with over-parameterization. *Preprint*.

Chen, X., Wang, W., et al. (2024). Refined generalization analysis of the deep ritz method and physics-informed neural networks. *arXiv preprint*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Daw, A., Bu, J., Wang, S., Perdikaris, P., and Karpatne, A. (2022). Mitigating propagation failures in pinns using evolutionary sampling.

De Ryck, T., Hesthaven, J. S., and Mishra, S. (2022). Error estimates for physics-informed neural networks approximating kolmogorov equations. *arXiv preprint arXiv:2201.10989*.

De Ryck, T. and Mishra, S. (2022). Generic bounds on the approximation error for physics-informed (and) operator learning. *arXiv preprint arXiv:2205.11393*.

Doľean, V., Jolivet, P., et al. (2024). Multilevel domain decomposition-based architectures for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*. In press.

Dwivedi, V., Srinivasan, B., and Krishnamurthi, G. (2024). Physics-informed contour selection for rapid image segmentation. *Scientific Reports*, 14(1):4647.

Escapil-Inchauspé, P. et al. (2023). h-analysis and data-parallel physics-informed neural networks. *Scientific Reports*, 13(21069).

Eymard, R., Gallouët, T., and Herbin, R. (2000). The finite volume method. *Handbook of Numerical Analysis*, 7:713–1020.

Ganga, S. and Uddin, Z. (2024). Exploring physics-informed neural networks: From fundamentals to applications in complex systems.

Hackbusch, W. (2015). *Hierarchical Matrices: Algorithms and Analysis*, volume 49 of *Springer Series in Computational Mathematics*. Springer.

He, J., Li, L., Xu, J., and Zheng, C. (2020). Relu deep neural networks and linear finite elements. *Journal of Computational Mathematics*, 38(3):502–527.

Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd edition.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257.

Hu, Z., Jagtap, A. D., and Karniadakis, G. E. (2022). When do extended physics-informed neural networks (xpinns) work? on the role of domain decomposition. *SIAM Journal on Scientific Computing*, 44(3):A1556–A1582.

Hughes, T. J. (2012). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY.

Jalalian, Y., Ramirez, J. F. O., Hsu, A., Hosseini, B., and Owhadi, H. (2025). Data-efficient kernel methods for learning differential equations and their solution operators: Algorithms and error analysis.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440.

Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., and Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. *arXiv preprint arXiv:2109.01050*.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000.

Lax, P. D. and Richtmyer, R. D. (1956). Survey of the stability of linear finite difference equations. *Communications on Pure and Applied Mathematics*, 9(2):267–293.

LeVeque, R. J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Li, J. and Chen, Z. (2023). Scalable pinns for high-dimensional pdes. *arXiv preprint arXiv:2307.12306*.

Li, Y., Chen, S., Shan, B., and Huang, S.-J. (2024). Causality-enhanced discreted physics-informed neural networks for predicting evolutionary equations. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4497–4505. International Joint Conferences on Artificial Intelligence Organization.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021a). Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021b). Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.

McClenny, L. and Braga-Neto, U. (2020). Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Minsky, M. and Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. MIT press.

Mishra, S. and Molinaro, R. (2022). Estimates on the generalization error of physics-informed neural networks for approximating pdes. *IMA Journal of Numerical Analysis*, 42(4):2739–2764.

Moseley, B., Markham, A., and Nissen-Meyer, T. (2023). Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*.

Mustajab, A. H., Lyu, H., Rizvi, Z., and Wuttke, F. (2024). Physics-informed neural networks for high-frequency and multi-scale problems using transfer learning. *Applied Sciences*, 14(8):3204.

Nabian, M. A., Gladstone, R. J., and Meidani, H. (2021). Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977.

Park, Y.-J. and Zobeiry, N. (2020). Convergence and error analysis of physics-informed neural networks. *Journal of Computational Physics*, 417:109545.

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

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Sau, R. C. and Yin, L. (2024). A review of neural network solvers for second-order boundary value problems.

Schmidt-Hieber, J. (2020). Nonparametric regression with deep neural networks: Theory and practice. *Inverse Problems*, 36(8):085008.

Shin, Y. L., Darbon, J., and Karniadakis, G. E. (2020). On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. *Communications in Computational Physics*, 28(5):2042–2074.

Shukla, K., Jagtap, A. D., and Karniadakis, G. E. (2021). Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447:110683.

Sirignano, J. and Spiliopoulos, K. (2018). Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364.

Strauss, W. A. (2008). *Partial Differential Equations: An Introduction*. John Wiley & Sons, Hoboken, NJ, 2nd edition.

Wang, S., Teng, Y., and Perdikaris, P. (2022a). When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768.

Wang, S., Yu, X., and Perdikaris, P. (2022b). When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:Article No. 110768.

Wang, Z., Chern, A., and Nakahira, Y. (2024). Generalizable physics-informed learning for stochastic safety-critical systems.

Wu, H., Luo, H., Ma, Y., Wang, J., and Long, M. (2024). Ropinn: Region optimized physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*. Poster presentation.

Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.

Yin, W., Chen, T., and Karniadakis, G. E. (2022). Pinn-based image inpainting with hybrid physics and perceptual losses. *arXiv preprint*, arXiv:2210.14138.

Zeidler, E. (1995). *Applied Functional Analysis: Applications to Mathematical Physics*, volume 108. Springer.

Zhang, L. and Wang, H. (2024). Physics-informed neural networks for time-dependent wave equations. *arXiv preprint arXiv:2403.19090*.

Zhang, Y. and Li, X. (2024). Pinns for singularly perturbed convection-diffusion problems. *arXiv preprint arXiv:2409.07671*.

Zhao, Z., Ding, X., and Prakash, B. A. (2024). Pinnsformer: A transformer-based framework for physics-informed neural networks. In *International Conference on Learning Representations (ICLR)*.

Zhou, H. and Xu, K. (2024). Coupled pinns for binary alloy solidification with moving boundaries. *arXiv preprint arXiv:2409.10910*.

Zienkiewicz, O. C. and Taylor, R. L. (2005). *The Finite Element Method for Solid and Structural Mechanics*. Elsevier Butterworth-Heinemann, Oxford, 6th edition.

APPENDIX A

Datasets, Algorithms, and Computational Frameworks

A.1 Computational environment and software framework

All experiments were implemented using **Python 3.10** under a Linux and Windows hybrid environment. The principal deep learning frameworks employed were **TensorFlow 2.0** and **PyTorch 1.13**, complemented by the following scientific computing and visualization libraries:

NumPy, SciPy, Matplotlib, SymPy, and Pandas.

Automatic differentiation for PDE residual computation was performed using the native back-propagation engines of PyTorch and TensorFlow. All experiments were executed on CPU-based machines with 16 GB RAM and 8-core Intel i7 processors, unless otherwise stated. The adaptive sampling and hierarchical compression routines were implemented as custom Python modules and integrated into the training loops via callable hooks.

A.2 Datasets and PDE benchmarks

The study relied exclusively on synthetic PDE datasets and analytical solutions generated for controlled experimentation. No external empirical data were used. The following benchmark problems formed the experimental corpus:

- One-dimensional viscous Burgers' equation on $(0, 1) \times (0, 1]$, with exact solution obtained from Cole–Hopf transformation.
- Heat, Laplace, and Wave equations on rectangular domains with known analytical solutions for convergence verification.

- One-dimensional Navier–Stokes equation under periodic boundary conditions for nonlinear testing.
- Schrödinger equation for oscillatory and complex-valued field analysis.
- Laplace equation for image inpainting, where image intensity channels were treated as scalar fields satisfying elliptic regularity within masked regions.

For each PDE, training data comprised uniformly or adaptively sampled collocation points. Boundary and initial condition data were synthetically generated from the analytical solutions and normalized to $[0, 1]$. For image inpainting, masked images were resized to 128×128 pixels and converted to floating-point tensors.

A.3 Algorithmic frameworks

The following algorithms summarize the full computational methodology employed in the thesis.

A.3.1 Algorithm 1: PINN–FDM residual-guided mesh refinement

Algorithm 1 Residual-Based Adaptive Mesh Refinement using PINNs

1: **Input:** Initial FDM grid $\mathcal{T}_0$, PDE operator $\mathcal{L}$, forcing term f, boundary conditions B, neural network $u_\theta(x, t)$.

2: Compute baseline FDM solution u_h on $\mathcal{T}_0$ using explicit or implicit time stepping.

3: Train u_θ to minimize

$$\mathcal{L}_{\text{PINN}}(\theta) = \|\mathcal{L}u_\theta - f\|^2_{L^2(\Omega)} + \|u_\theta - B\|^2_{\partial\Omega}.$$

4: Evaluate residual field $\mathcal{R}_\theta(x, t) = \mathcal{L}u_\theta(x, t) - f(x, t)$ on $\mathcal{T}_0$.

5: Identify high-residual zones $\mathcal{Z} = \{(x, t) : |\mathcal{R}_\theta(x, t)| > \varepsilon_{\text{th}}\}$.

6: Refine grid $\mathcal{T}_0$ locally around $\mathcal{Z}$ to obtain $\mathcal{T}_1$ using adaptive refinement ratio r.

7: Recompute FDM solution $u_{h,1}$ on $\mathcal{T}_1$ and iterate until residual field is uniformly small.

8: **Output:** Refined mesh $\mathcal{T}_k$ and improved numerical solution $u_{h,k}$.

A.3.2 Algorithm 2: FEM–PINN structural equivalence and error localization

Algorithm 2 FEM–PINN Hybrid Error Localization and Refinement

1: **Input:** FEM triangulation $\mathcal{T}_h$, PDE bilinear form $a(u, v)$, forcing term f, and neural surrogate $u_\theta(x)$.
2: Compute baseline FEM solution $u_h \in V_h$ satisfying $a(u_h, v_h) = (f, v_h)$.
3: Train u_θ to minimize the residual-based loss $\|\mathcal{L}u_\theta - f\|^2_{L^2(\Omega)}$.
4: Evaluate the element-wise residual indicator

$$\eta_K = \|\mathcal{R}_\theta\|_{L^2(K)}, \quad K \in \mathcal{T}_h.$$

5: Mark elements K with $\eta_K > \varepsilon_{\text{tol}}$ for refinement.
6: Generate refined mesh $\mathcal{T}_{h'}$ and repeat FEM solution.
7: **Output:** Adaptive FEM–PINN solution $u_{h'}$ and error field η_K.

A.3.3 Algorithm 3: residual-driven adaptive sampling for PINNs

Algorithm 3 Adaptive Collocation Sampling based on Residual Magnitude

1: **Input:** Domain Ω, PDE operator $\mathcal{L}$, data (x_i, f_i), network $u_\theta(x)$.
2: Initialize N_0 uniformly distributed collocation points $\mathcal{X}_0$.
3: **repeat**
4: Train u_θ for T epochs on current sample set $\mathcal{X}_k$.
5: Compute residuals $r_i = |\mathcal{L}u_\theta(x_i) - f_i|$ for all $x_i \in \mathcal{X}_k$.
6: Sample new points with probability proportional to r_i^α, $\alpha > 1$.
7: Update collocation set $\mathcal{X}_{k+1} = \mathcal{X}_k \cup \mathcal{X}_{\text{new}}$.
8: **until** convergence of mean residual below threshold ε_{res}.
9: **Output:** Optimized collocation distribution $\mathcal{X}_k$.

A.3.4 Algorithm 4: h-matrix compression for efficient PINN training

Algorithm 4 Hierarchical Matrix Compression of the PINN Residual Jacobian

Input: Residual vector $R(\theta) \in \mathbb{R}^{N_r}$, Jacobian $J = \partial R/\partial\theta$.
Partition J into admissible sub-blocks $\{J_{ij}\}$ by geometric clustering.
for each block J_{ij} **do**
    if block admissible **then**
        Compute low-rank approximation $J_{ij} \approx U_{ij}V_{ij}^\top$ with rank k_{ij} s.t.

$$\|J_{ij} - U_{ij}V_{ij}^\top\|_F \le \varepsilon_H.$$

    else
        Store block densely.
    end if
end for
Construct compressed Jacobian $\tilde{J}$ and approximate Hessian $\tilde{H} = \tilde{J}^\top\tilde{J}$.
Update parameters via compressed Gauss–Newton step $\tilde{H}\Delta\theta = -\tilde{J}^\top R$.
Output: Updated parameters $\theta \leftarrow \theta + \Delta\theta$, compressed structure $\tilde{J}$.

A.3.5 Algorithm 5: laplace-based PINN for image inpainting

Algorithm 5 Laplace-Governed Physics-Informed Neural Network for Image Inpainting

Input: Image domain Ω, mask $M : \Omega \to \{0, 1\}$, boundary data u_{data}, network $u_\theta(x)$.
Define PDE residual $\mathcal{R}_\theta(x) = \Delta u_\theta(x)$ and data mismatch $E_\theta(x) = u_\theta(x) - u_{\text{data}}(x)$.
Form total loss

$$\mathcal{L}(\theta) = \int_{\Omega\setminus D} \|E_\theta(x)\|^2 M(x)\,dx + \lambda \int_D \|\mathcal{R}_\theta(x)\|^2(1 - M(x))\,dx.$$

Train u_θ with Adam optimizer until $\|\nabla_\theta\mathcal{L}\|$ stagnates below tolerance $\varepsilon_{\text{train}}$.
Replace masked pixels by $u_\theta(x)$ over D to obtain inpainted image $\hat{u}$.
Output: Reconstructed image $\hat{u}$ satisfying approximate Laplace regularity.

A.4 Summary of computational components

The algorithms above constitute the methodological backbone of this thesis. Each algorithm directly corresponds to a set of experiments reported in Chapters 4.1.1–4.4. The integration of

residual-based refinement, adaptive sampling, and hierarchical compression demonstrates that the Physics-Informed Neural Network framework can be systematically implemented as a stable, consistent, and convergent numerical solver for partial differential equations, while remaining computationally efficient and extendable to practical applications such as image inpainting.

Table of notation

Domains, sets, and measures

$\Omega \subset \mathbb{R}^d$	Physical (space or space–time) domain
$D \subset \Omega$	Missing/corrupted inpainting subdomain
$\Gamma = \partial D$	Boundary of D
$X_f = \{x_j\}_{j=1}^{N_f}$	Residual/collocation sample set
$X_u = \{x_i\}_{i=1}^{N_u}$	Supervised/data sample set
N_f, N_u	Numbers of residual and data samples
$M : \Omega \to \{0, 1\}$	Binary mask (known/unknown pixels)

Unknowns, references, and networks

$u : \Omega \to \mathbb{R}^m$	True PDE solution (or image; $m = 1$ or 3)
$u^\star$	Exact/target PDE solution
u_{data}	Observed/known data on $\Omega \setminus D$ or Γ
u_θ	Neural approximation (PINN) with parameters θ
$\mathcal{F}_\theta$	Network manifold; $T_{u_\theta}\mathcal{F}_\theta$ its tangent space

Operators, forms, and residuals

$\mathcal{N}[u]$ or $L[u]$	PDE operator (strong form)
$\mathcal{R}_\theta(x)$	Strong-form residual $L[u_\theta](x) - f(x)$
$a(u, v)$	Bilinear form (weak form), e.g. $a(u, v) = \int_\Omega \nabla u^\top A \nabla v + c\, uv$
$V = H^1(\Omega)$, $V_0 = H_0^1(\Omega)$	Energy and homogeneous Sobolev spaces
$\|\cdot\|$, $\|\cdot\|_{H^s}$	L^2 and Sobolev norms

$\|\cdot\|_{V^*}$, $\|\cdot\|_{H^{-s}}$	Dual norms

Losses, weights, and empirical norms

$\mathcal{L}_u$	Data loss over X_u
$\mathcal{L}_f$	Physics/residual loss over X_f
$\mathcal{L}^{\text{PINN}}$	Composite PINN loss ($\mathcal{L}_u + \lambda\,\mathcal{L}_f$)
$\lambda > 0$	Loss weight (data–physics trade-off)
$\|g\|_{1,N}$	Empirical ℓ_1 norm: $\frac{1}{N}\sum_{\ell=1}^{N}\|g(x^{(\ell)})\|$
$\|\cdot\|_{\ell^2(X_f)}$	Sampled ℓ^2 norm over X_f

Stability, sensitivity, and generalization

S_θ	First-order sensitivity (loss-gradient surrogate)
C	Operator bound: $\|L[\delta u]\|_\infty \le C\|\delta u\|_\infty$
δ	Pointwise perturbation magnitude ($\|\delta u\|_\infty \le \delta$)
ε	Tolerance parameter in stability/generalization bounds
M_f, M_d	Uniform residual/data bounds used in concentration results

Sampling, Jacobians, and compression

$J = \partial R/\partial\theta$	Residual Jacobian
$H = J^\top J$	Gauss–Newton (approximate) Hessian/NTK Gram
$\tilde{J}, \tilde{H}$	Hierarchically compressed Jacobian/Gram
ε_H	H-matrix compression tolerance (spectral control)

Classical PDEs and models

Heat: $u_t = \kappa u_{xx}$	Parabolic model (diffusion)
Wave: $u_{tt} = c^2 u_{xx}$	Hyperbolic model (propagation)
Burgers: $u_t + uu_x = \nu u_{xx}$	Nonlinear advection–diffusion
Laplace: $\Delta u = 0$	Elliptic (harmonic) model

Cahn–Hilliard: $u_t = -\Delta(-\epsilon\Delta u + F'(u))$	Fourth-order diffusion/interface model
Perona–Malik: $u_t = \nabla \cdot (g(\lvert\nabla u\rvert)\nabla u)$	Anisotropic diffusion (edge-preserving)

Inpainting-specific

(3.1)	Inpainting Laplace model on D with $u\lvert_\Gamma = u_{\text{data}}$

List of abbreviations

PINN	Physics-Informed Neural Network
PDE	Partial Differential Equation
FDM	Finite Difference Method
FEM	Finite Element Method
AMR	Adaptive Mesh Refinement
NTK	Neural Tangent Kernel
H-matrix	Hierarchical Matrix
AD	Automatic Differentiation
MLP	Multi-Layer Perceptron
ReLU	Rectified Linear Unit
SiLU	Sigmoid Linear Unit
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno
CFL	Courant–Friedrichs–Lewy (margin/condition)
i.i.d.	Independent and Identically Distributed
BC	Boundary Condition
IC	Initial Condition
RGB	Red–Green–Blue (image channels)
FNO	Fourier Neural Operator
DeepONet	Deep Operator Network
CH	Cahn–Hilliard (equation/model)
PM	Perona–Malik (diffusion)

CPU	Central Processing Unit
GPU	Graphics Processing Unit
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation (optimizer)

A.5 Code

The Code used in this work is all available via `https://github.com/karjxenval/Numerical-Analysis-of-PINNs-for-solving-PDEs`