



Numerical Studies of the Convergence of PINNs for Elliptic Boundary Value Problems

Master's Thesis submitted for the degree of Master of Science (M.Sc.) at the Department of
Mathematics and Computer Science of Freie Universität Berlin

Student: Meichen Lu

Supervisor: Univ.-Prof. Dr. Volker John

Second Reviewer: PD Dr. Alfonso Caiazzo

Berlin, September 14, 2026

Declaration of Authorship

I hereby declare that I have written this thesis independently and that I have used no sources or aids other than those cited. All passages that have been taken from other works, either directly or indirectly, have been clearly marked and properly referenced.

I further declare that this thesis has not been submitted, either in identical or similar form, to any other university or examination authority, and that it has not been published elsewhere.

During the preparation of this thesis, I used AI-assisted tools, such as ChatGPT, solely to improve the readability, language quality, and formatting of the text. AI-assisted tools were also used to support the extension and improvement of some contents in Chapters 2 and 3. After using these tools, I carefully reviewed and edited the content. I take full responsibility for all mathematical content, analysis, arguments, and conclusions presented in this thesis.

Berlin, September 14, 2026

Date



Signature

Abstract

This thesis investigates physics-informed neural networks for steady elliptic boundary value problems, with particular emphasis on Poisson and convection–diffusion–reaction equations. The analytical part reviews the variational setting for elliptic problems, including weak formulations, well-posedness, finite element comparison principles, and maximum principles. These results provide the mathematical reference framework for assessing residual-based neural network approximations.

The machine learning part introduces neural networks as parametric approximation spaces and explains the construction of physics-informed neural networks through residual minimization. Particular attention is paid to the role of automatic differentiation, collocation sampling, optimization error, and the imposition of boundary conditions. Since elliptic boundary value problems are strongly influenced by boundary data and maximum principles, the thesis also discusses hard-constrained boundary representations and their connection with the qualitative behaviour of the computed solutions.

The numerical part studies the convergence behaviour of PINNs under uniform collocation refinement and compares the influence of activation functions and hard-constrained boundary indicators. Seven numerical experiments are considered. They start from a Poisson problem and then move to smooth and layer-type convection–diffusion–reaction problems. Additional comparisons investigate the effect of replacing the hyperbolic tangent activation by GELU and of replacing the polynomial boundary indicator by an exponential indicator. The experiments show that PINNs can approximate smooth elliptic and convection–diffusion–reaction solutions accurately in selected cases. However, the final errors generally do not decrease monotonically with respect to the number of collocation points, and the behaviour becomes more sensitive for convection–diffusion–reaction problems with layers. The results indicate that residual loss, error norms, pointwise bounds, and oscillation diagnostics should be considered together when evaluating the reliability of PINN approximations.

Contents

Declaration of Authorship	i
Abstract	ii
List of Notations	iv
1 Introduction	1
2 Steady Elliptic Boundary Value Problems	2
2.1 Preliminaries: domains, Sobolev spaces, and variational tools	2
2.2 The Poisson Problem	3
2.3 A Convection–Diffusion–Reaction Problem	6
2.4 Maximum Principles for Elliptic Problems	8
3 Introduction to Machine Learning and Physics-Informed Neural Networks	11
3.1 Fundamentals of Machine Learning	11
3.1.1 Learning problems and parametric approximation	12
3.1.2 Supervised regression and empirical risk minimization	12
3.1.3 Risk functionals and approximation properties	13
3.1.4 Gradient-based optimization	14
3.1.5 Automatic differentiation	15
3.2 Neural Networks as Parametric Function Spaces	16
3.2.1 Neural networks as parameterized mappings	17
3.2.2 Regularity properties	18
3.2.3 Approximation properties	18
3.3 Physics-Informed Neural Networks (PINNs)	19
3.3.1 Learning paradigm underlying physics-informed neural networks	20
3.3.2 Neural network representation of the PDE solution	21
3.3.3 Automatic differentiation and residual evaluation	22
3.3.4 Residual minimization and loss functional formulation	22
3.3.5 Gradient-based training algorithm	24
3.4 Practical considerations and relation to elliptic problems	25
3.4.1 Sampling as stochastic quadrature	25
3.4.2 Optimization landscape and non-convexity	26
3.4.3 Regularity and approximation capability	27
3.4.4 Convection-dominated regimes	28
3.4.5 Compatibility with maximum principles	29
4 Numerical Experiments	31
4.1 Experimental Design	31
4.2 Exact and Reference Solutions	32
4.3 Poisson Problem: Standard Polynomial Indicator	33
4.4 CDR Problem with Smooth Solution: Standard Polynomial Indicator	35
4.5 Layer CDR Problem: Standard Indicator	36
4.6 Poisson Problem: GELU Activation	38
4.7 Poisson Problem: Exponential Indicator	39
4.8 CDR Problem with Smooth Solution: GELU Activation	41
4.9 Layer CDR Problem: GELU Activation	42
4.10 Discussion of the Numerical Results	44
5 Conclusion	45

List of Notations

Notation	Description	Section
Ω	Computational domain	2
$\partial\Omega$	Boundary of the domain	2
$H^1(\Omega)$	Sobolev space of square-integrable functions with square-integrable weak derivatives	2.1
$H_0^1(\Omega)$	Sobolev space with zero trace on the boundary	2.1
$\ \cdot\ _{L^2(\Omega)}$	L^2 norm on the domain	2.1
$\ \cdot\ _{H^1(\Omega)}$	H^1 norm on the domain	2.1
$\ \cdot\ _{L^\infty(\Omega)}$	Maximum norm used for pointwise error evaluation	4
κ	Scalar diffusion coefficient	2.2
A	Diffusion tensor or stiffness matrix, depending on the context	2.3
b, β	Convection field	2.4
c, σ_r	Reaction coefficient	2.4, 4
ρ	Activation function of the neural network	3
f	Source term or right-hand side	2
g	Dirichlet boundary data	2
$\mathcal{L}$	Differential operator	2
h	Mesh width or grid spacing	3, 4
V_h	Finite element trial space	3
u	Exact or weak solution of the boundary value problem	2
u_θ	Neural network approximation of the PDE solution	3, 4
θ	Trainable parameters of the neural network	3
N_Ω	Number of interior collocation points	3, 4
$N_{\partial\Omega}$	Number of boundary collocation points	3, 4
Pe	Péclet number	2.4

List of Figures

1	Exact solution used in the Poisson experiments.	32
2	Exact solution used in the CDR experiments with smooth solution.	33
3	Exact solution used in the outflow-layer CDR experiments.	33
4	Experiment 1: final error norms for the Poisson problem.	34
5	Experiment 1: observed convergence orders for the Poisson problem.	35
6	Experiment 2: final error norms for the CDR problem with smooth solution. . .	36
7	Experiment 2: observed convergence orders for the CDR problem with smooth solution.	36
8	Experiment 3: final error norms for the layer CDR problem.	37
9	Experiment 3: observed convergence orders for the layer CDR problem.	38
10	Experiment 4: final error norms for the Poisson problem with GELU activation. .	39
11	Experiment 4: observed convergence orders for the Poisson problem with GELU activation.	39
12	Experiment 5: final error norms for the Poisson problem with the exponential indicator.	40
13	Experiment 5: observed convergence orders for the Poisson problem with the exponential indicator.	41
14	Experiment 6: final error norms for the CDR problem with smooth solution and GELU activation.	42
15	Experiment 6: observed convergence orders for the CDR problem with smooth solution and GELU activation.	42
16	Experiment 7: final error norms for the layer CDR problem with GELU activation. .	43
17	Experiment 7: observed convergence orders for the layer CDR problem with GELU activation.	44

List of Tables

1	Experiment 1: Poisson problem with tanh activation and polynomial hard-constraint indicator.	34
2	Experiment 2: CDR problem with smooth solution, tanh activation, and polynomial indicator.	35
3	Experiment 3: layer CDR problem with the standard layer indicator.	37
4	Experiment 4: Poisson problem with GELU activation and polynomial indicator.	38
5	Experiment 5: Poisson problem with tanh activation and exponential boundary indicator.	40
6	Experiment 6: CDR problem with smooth solution, GELU activation, and polynomial indicator.	41
7	Experiment 7: layer CDR problem with GELU activation and layer-type indicator.	43

1 Introduction

Partial differential equations (PDEs) arise naturally in the mathematical modeling of numerous physical, biological, and engineering processes, including heat conduction, diffusion phenomena, electrostatics, and transport-reaction dynamics. In many practical situations, these models lead to elliptic boundary value problems posed on bounded domains, for which analytical solutions are rarely available. As a consequence, the development and analysis of reliable numerical methods for elliptic PDEs has long been a central topic in applied analysis and scientific computing.

Among classical numerical approaches, the finite element method (FEM) occupies a prominent position due to its solid theoretical foundation and flexibility with respect to complex geometries. The FEM is rooted in a variational formulation of the governing equations and provides systematic tools for establishing stability and convergence, together with quantitative a priori error estimates under appropriate assumptions on the data and the domain.

In recent years, advances in machine learning have stimulated interest in alternative strategies to approximate solutions of PDEs. In particular, physics-informed neural networks (PINNs) have been proposed as a mesh-free approach in which the solution of a differential equation is approximated by a neural network whose parameters are determined by minimizing a loss function incorporating the residual of the PDE and the associated boundary conditions. As stated by Raissi et al. “the physical laws that govern the data generation process are encoded in the neural network through the loss function” [1].

The PINNs framework was further characterized as “a data-efficient and mesh-free method for solving forward and inverse problems involving nonlinear partial differential equations” [2]. A broader perspective is provided in [3], where physics-informed machine learning is presented as a bridge between data-driven modeling and classical scientific computing. Despite their conceptual appeal, PINNs can have optimization difficulties, sensitivity to sampling strategies, and challenges in solving sharp gradients or boundary layers. These issues motivate a careful mathematical study of the PINN approximations against well-established analytical baselines.

The aim of this thesis is to investigate steady elliptic boundary value problems from a mathematical point of view and to prepare a rigorous baseline for later comparison between FEM and PINN-based approximations. The present work focuses on the Poisson equation and a more general convection–diffusion–reaction equation, which together capture diffusion-dominated and transport-influenced regimes frequently encountered in applications.

2 Steady Elliptic Boundary Value Problems

Elliptic boundary value problems are among the most fundamental classes of PDEs in both theory and applications. They typically describe equilibrium or steady-state configurations: diffusion-driven processes (e.g. stationary heat conduction, electrostatic potential), as well as steady transport with lower-order effects such as convection and reaction. In the PDE literature, second-order elliptic equations constitute a prototypical setting for weak solutions, variational methods, and regularity theory; see, for example, [4, Ch. 6, pp. 303–351] and [5, Chs. 3–6, pp. 31–112]. From the perspective of numerical analysis, elliptic problems also provide the canonical framework in which Galerkin methods, and in particular the finite element method, can be rigorously analyzed with stability and convergence guarantees [6, Ch. 2, pp. 58–65].

The discussion proceeds by first establishing a functional-analytic setting used for all model classes, including Sobolev spaces, weak boundary conditions, and basic variational tools. The Poisson problem is then treated as a baseline elliptic model, and the passage from strong to weak formulations, well-posedness, regularity, and finite element error analysis is presented in a unified chain. Finally, the analysis is extended to a convection–diffusion–reaction model, which introduces additional qualitative considerations and provides a more demanding testbed for later approximation methods [7, Ch. 1, pp. 1–34].

2.1 Preliminaries: domains, Sobolev spaces, and variational tools

Throughout this section, let $\Omega \subset \mathbb{R}^d$ ($d \geq 1$) be a bounded domain with boundary $\partial\Omega$. Since elliptic boundary value problems are formulated together with boundary conditions, both the geometry of the domain and the functional setting in which solutions are sought play a crucial role. The assumptions and analytical tools introduced below will be used repeatedly in the subsequent analysis.

Definition 2.1 (Lipschitz domain). A bounded open set $\Omega \subset \mathbb{R}^d$ is called a *Lipschitz domain* if for every $x_0 \in \partial\Omega$ there exists a neighborhood U of x_0 and a bijective change of coordinates such that, in the new coordinates, $\partial\Omega \cap U$ coincides with the graph of a Lipschitz continuous function.

This assumption is mild and covers most domains encountered in applications. Its importance lies in the fact that it guarantees several fundamental analytical results. In particular, for Lipschitz domains, the trace operator from $H^1(\Omega)$ to $L^2(\partial\Omega)$ is well-defined and continuous, and Green’s identities extend to functions with weak derivatives. These properties allow boundary conditions and integration-by-parts formulas to be interpreted in a weak sense, which is essential for variational formulations of elliptic problems; see [4, Ch. 5, pp. 250–276] and [5, Ch. 7, pp. 150–170].

Standard Sobolev spaces $L^2(\Omega)$ and $H^1(\Omega)$ are employed. For problems with homogeneous Dirichlet boundary conditions, the natural energy space is

$$H_0^1(\Omega) := \overline{C_c^\infty(\Omega)}^{\|\cdot\|_{H^1(\Omega)}}.$$

Functions in $H_0^1(\Omega)$ vanish on $\partial\Omega$ in the sense of traces. From a physical perspective, this space enforces boundary conditions directly at the level of admissible functions. From a mathematical perspective, it provides a Hilbert space structure in which variational formulations are well-posed.

A key analytical tool in this setting is Poincaré’s inequality, which controls the L^2 norm of a function by the L^2 norm of its gradient.

Theorem 2.2 (Poincaré inequality). *There exists a constant $C_P > 0$ such that*

$$\|v\|_{L^2(\Omega)} \leq C_P \|\nabla v\|_{L^2(\Omega)}, \quad \forall v \in H_0^1(\Omega).$$

A proof of this inequality and its variants for Sobolev spaces can be found in [4, Ch. 5, pp. 275–276]. As a consequence, the seminorm $\|\nabla v\|_{L^2(\Omega)}$ defines a norm on $H_0^1(\Omega)$ that is equivalent to the full H^1 norm. This equivalence is of fundamental importance in elliptic theory: it implies that estimates expressed purely in terms of gradients are sufficient to control the solution in the natural energy norm. In particular, coercivity estimates for bilinear forms translate directly into stability and uniqueness of weak solutions.

The concept of a weak solution arises naturally when classical solutions are not available. For second-order elliptic equations, classical solutions require second derivatives, which may fail to exist even for smooth data or complex geometries. Weak solutions relax this requirement by transferring derivatives onto test functions via integration by parts, thereby lowering regularity demands on the solution itself. This idea is central in modern PDE theory and underlies both analytical and numerical treatments of elliptic problems.

Weak formulations of elliptic boundary value problems can be written in an abstract variational form. Let V be a Hilbert space and consider the problem of finding $u \in V$ such that

$$a(u, v) = \ell(v), \quad \forall v \in V,$$

where $a(\cdot, \cdot)$ is a bilinear form on $V \times V$ and ℓ is a continuous linear functional on V . In this framework, the central analytical question is whether the problem admits a unique solution that depends continuously on the data.

The answer is provided by the Lax–Milgram theorem, which constitutes the cornerstone of the variational theory of linear elliptic problems.

Theorem 2.3 (Lax–Milgram). *Let V be a Hilbert space. Assume that the bilinear form $a : V \times V \rightarrow \mathbb{R}$ is continuous and coercive, i.e.,*

$$|a(u, v)| \leq C_a \|u\|_V \|v\|_V, \quad a(v, v) \geq \alpha \|v\|_V^2 \quad (\alpha > 0),$$

and that $\ell \in V'$ is continuous. Then there exists a unique solution $u \in V$ satisfying

$$a(u, v) = \ell(v), \quad \forall v \in V,$$

and the solution obeys the stability estimate

$$\|u\|_V \leq \alpha^{-1} \|\ell\|_{V'}.$$

A proof of the Lax–Milgram theorem in the Hilbert space setting is given in [4, Sec. 5.8, pp. 299–303]. The significance of this result in the context of elliptic PDEs is twofold. First, it provides existence and uniqueness of weak solutions under minimal regularity assumptions on the data. Second, it identifies the structural properties—continuity and coercivity of the bilinear form—that must be verified for a given differential operator. These properties are verified explicitly for each model problem considered below.

Finally, the same variational structure forms the basis of many numerical discretizations. In particular, conforming Galerkin methods approximate the infinite-dimensional space V by finite-dimensional subspaces $V_h \subset V$. The analysis of such methods relies on comparing the continuous variational problem with its discrete counterpart, using tools such as Galerkin orthogonality and Céa’s lemma; see [6, Ch. 2, pp. 58–65].

2.2 The Poisson Problem

The Poisson problem represents the prototypical example of a second-order elliptic boundary value problem and occupies a central position in the theory of partial differential equations. Many structural properties of general elliptic operators—such as coercivity, variational formulations, and regularity of solutions—can already be observed in this simplest setting. For this

reason, the Poisson equation is frequently adopted as a reference model in analytical and numerical studies of elliptic problems; see, for example, [4, Ch. 6, pp. 303–351] and [6, Ch. 1, pp. 1–38].

In the present context, the Poisson problem provides a natural and mathematically transparent starting point for the study of steady elliptic equations. Its relatively simple structure allows the essential concepts of weak solutions, well-posedness, and finite element approximation to be introduced rigorously, while at the same time establishing tools and ideas that will be reused for more general elliptic models.

Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain and let $f \in L^2(\Omega)$. Consider the boundary value problem

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega. \end{cases} \quad (1)$$

If the solution u is sufficiently smooth, equation (1) is understood pointwise and the boundary condition is satisfied in the classical sense. For smooth right-hand sides on domains with sufficiently smooth boundary, classical elliptic regularity yields smooth solutions. Loss of smoothness may occur when the boundary, coefficients, boundary data, or right-hand side fail to satisfy the required regularity assumptions. This limitation motivates the introduction of a weaker notion of solution that is compatible with lower regularity assumptions.

The passage from the classical formulation to a weak formulation is based on integration by parts. Formally assuming u to be smooth and multiplying (1) by a test function $v \in C_c^\infty(\Omega)$ yields

$$\int_{\Omega} (-\Delta u) v \, dx = \int_{\Omega} \nabla u \cdot \nabla v \, dx.$$

Since test functions vanish on the boundary, the boundary term disappears. Importantly, the right-hand side involves only first derivatives of u , suggesting that the problem can be reformulated in a function space where first derivatives are square-integrable. This observation leads naturally to the Sobolev space $H_0^1(\Omega)$.

Definition 2.4 (Weak solution of the Poisson problem). A function $u \in H_0^1(\Omega)$ is called a weak solution of (1) if

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx, \quad \forall v \in H_0^1(\Omega). \quad (2)$$

The choice of the space $H_0^1(\Omega)$ ensures that the homogeneous Dirichlet boundary condition is enforced in the sense of traces. In particular, while the boundary condition $u = 0$ may not hold pointwise, it is satisfied in a weak sense that is compatible with the variational formulation. This interpretation of boundary conditions is standard in elliptic theory and is essential for both analytical well-posedness and numerical approximation.

The weak formulation (2) can be written in abstract variational form. Define the bilinear form

$$a(u, v) := \int_{\Omega} \nabla u \cdot \nabla v \, dx,$$

and the linear functional

$$\ell(v) := \int_{\Omega} f v \, dx,$$

for $u, v \in H_0^1(\Omega)$. The problem consists in finding $u \in H_0^1(\Omega)$ such that $a(u, v) = \ell(v)$ for all $v \in H_0^1(\Omega)$.

It remains to verify that this variational problem is well posed. The bilinear form $a(\cdot, \cdot)$ is continuous on $H_0^1(\Omega)$ by the Cauchy–Schwarz inequality, and it is coercive due to Poincaré’s inequality. Moreover, the functional ℓ is continuous whenever $f \in L^2(\Omega)$. These properties allow the application of the Lax–Milgram theorem.

Theorem 2.5 (Existence, uniqueness, and stability). *Let $f \in L^2(\Omega)$. Then there exists a unique weak solution $u \in H_0^1(\Omega)$ of (2). Moreover, the solution satisfies the stability estimate*

$$\|u\|_{H_0^1(\Omega)} \leq C\|f\|_{L^2(\Omega)},$$

where $C > 0$ depends only on the domain Ω .

This is a direct application of the Lax–Milgram theorem to the bilinear form $a(\cdot, \cdot)$ on $H_0^1(\Omega)$; see [4, Sec. 5.8, pp. 299–303].

The stability estimate in Theorem 2.5 plays a crucial role in numerical analysis. It shows that small perturbations in the data f lead to proportionally small changes in the solution, a property that is indispensable for reliable numerical approximation.

While the existence of a weak solution in $H_0^1(\Omega)$ is sufficient for well-posedness, additional smoothness of the solution is often required in order to derive quantitative error estimates for numerical methods. Elliptic regularity theory provides such results. Under suitable assumptions on the domain Ω , one has

$$\|u\|_{H^2(\Omega)} \leq C\|f\|_{L^2(\Omega)},$$

which implies that the weak solution possesses square-integrable second derivatives; see [5, Ch. 6, pp. 107–112]. This higher regularity is not guaranteed in general and depends not only on the smoothness of the domain and the right-hand side, but also on the type and regularity of the boundary conditions. In the homogeneous Dirichlet case one obtains the standard variational formulation in $H_0^1(\Omega)$. For non-homogeneous Dirichlet data, the boundary values must first be incorporated by a lifting, or enforced directly in the approximation space.

Finite element approximation of the Poisson problem is now considered. Let $\{\mathcal{T}_h\}$ be a shape-regular family of triangulations of Ω , and let $V_h \subset H_0^1(\Omega)$ be a conforming finite element space. The Galerkin approximation is defined by seeking $u_h \in V_h$ such that

$$a(u_h, v_h) = \ell(v_h), \quad \forall v_h \in V_h. \quad (3)$$

The discrete problem (3) inherits the variational structure of the continuous problem, which is the key to its stability. In particular, the error between the exact solution and its finite element approximation satisfies an orthogonality relation with respect to the bilinear form $a(\cdot, \cdot)$.

Beyond existence, uniqueness, and stability, elliptic equations possess important qualitative properties that provide additional insight into the behavior of their solutions. One of the most fundamental such properties is the maximum principle, which asserts that solutions of elliptic equations cannot attain their extrema in the interior of the domain unless they are constant. This principle plays a central role in the qualitative theory of elliptic PDEs and has far-reaching consequences for both analysis and numerical approximation.

For the Poisson problem with homogeneous Dirichlet boundary conditions, the maximum principle provides a bound on the solution purely in terms of the data. Informally, it expresses the idea that the interior behavior of the solution is controlled by the forcing term and the boundary conditions, ruling out the appearance of spurious oscillations.

Theorem 2.6 (Maximum principle for the Poisson problem). *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain and let $f \in C(\Omega)$. Assume that $u \in C^2(\Omega) \cap C(\overline{\Omega})$ satisfies*

$$-\Delta u = f \quad \text{in } \Omega.$$

and assume that $u = 0$ on $\partial\Omega$. If $f \leq 0$ in Ω , then

$$\max_{x \in \overline{\Omega}} u(x) \leq 0.$$

Similarly, if $f \geq 0$ in Ω , then

$$\min_{x \in \overline{\Omega}} u(x) \geq 0.$$

Remark 2.7. The regularity assumption $u \in C^2(\Omega)$ is imposed for simplicity of presentation. More general versions of the weak maximum principle hold for weak solutions under suitable assumptions on the domain, the data, and the sign of the right-hand side; see [5, Ch. 3, pp. 31–35]. Since the strong maximum principle is not developed here, only this weak form is used in the sequel.

The maximum principle provides a qualitative complement to the stability estimate derived earlier. While stability bounds control the solution in an integral (energy) norm, the maximum principle yields control in the $L^\infty(\Omega)$ norm and ensures that the solution does not exhibit unphysical overshoots or undershoots. In particular, it implies uniqueness of solutions without explicit reliance on variational arguments and provides a useful a priori bound in terms of the source term.

From a numerical perspective, the maximum principle serves as an important benchmark for assessing the reliability of approximation schemes. A numerical method that violates the maximum principle may produce non-physical oscillations even when it converges in an energy norm. This observation is especially relevant in the context of finite element and finite difference methods and will reappear in the discussion of more general elliptic and convection–diffusion–reaction problems.

Lemma 2.8 (Galerkin orthogonality). *Let u be the weak solution of (2) and let u_h be the finite element solution of (3). Then*

$$a(u - u_h, v_h) = 0, \quad \forall v_h \in V_h.$$

This orthogonality property implies that the finite element error is minimal in the energy norm over the discrete space. More precisely, Céa’s lemma yields a quasi-optimality result.

Theorem 2.9 (Céa’s lemma).

$$\|u - u_h\|_{H_0^1(\Omega)} \leq \inf_{v_h \in V_h} \|u - v_h\|_{H_0^1(\Omega)}.$$

For the symmetric Poisson problem this form follows from Galerkin orthogonality and the identity between the bilinear form and the energy inner product. The more general quasi-optimality statement, including continuity and coercivity constants, is proved in [6, Ch. 2, pp. 58–65]. To obtain explicit convergence rates, the best-approximation term is estimated using interpolation theory. If the exact solution satisfies $u \in H^2(\Omega)$ and V_h consists of continuous piecewise linear functions, standard interpolation estimates imply

$$\|u - u_h\|_{H_0^1(\Omega)} \leq Ch \|u\|_{H^2(\Omega)},$$

where h denotes the mesh size. This estimate shows that the convergence rate of the finite element method is directly determined by the regularity of the exact solution, thereby emphasizing the close interplay between elliptic regularity theory and numerical approximation; see [6, Ch. 4, pp. 115–124].

In summary, the Poisson problem provides a complete and representative example in which the fundamental analytical and numerical aspects of elliptic boundary value problems can be examined in a unified framework. The insights obtained here will serve as a reference point for the study of more general elliptic models, including convection–diffusion–reaction equations, where additional terms modify but do not destroy the underlying variational structure.

2.3 A Convection–Diffusion–Reaction Problem

A steady convection–diffusion–reaction (CDR) problem extends the Poisson equation by incorporating first-order transport effects and zeroth-order reaction terms. Such equations arise

naturally in models of mass transport, heat transfer with advection, and simplified chemical reaction processes. From a mathematical perspective, the CDR problem represents a natural generalization of the Poisson problem: while uniform ellipticity is retained through the diffusion operator, the presence of convection and reaction terms introduces non-symmetry and additional analytical challenges; see [4, Ch. 6, pp. 303–351] and [7, Ch. 1, pp. 1–34].

Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain. Consider the boundary value problem

$$\begin{cases} -\nabla \cdot (\kappa \nabla u) + \beta \cdot \nabla u + cu = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega. \end{cases} \quad (4)$$

Here κ denotes the diffusion coefficient, β is a prescribed convection field, and c is a reaction coefficient.

Throughout the analysis, the coefficients are assumed to satisfy the following standard conditions.

Assumption 2.10 (Coefficients). Assume that $\kappa \in L^\infty(\Omega)$ with $\kappa(x) \geq \kappa_0 > 0$ almost everywhere in Ω , that $\beta \in (L^\infty(\Omega))^d$, and that $c \in L^\infty(\Omega)$ with $c(x) \geq 0$ almost everywhere in Ω .

Under Assumption 2.10, the diffusion operator remains uniformly elliptic, ensuring that the problem retains a fundamentally elliptic character. In contrast to the Poisson problem, however, the convection term $\beta \cdot \nabla u$ renders the differential operator non-symmetric, which has important consequences for both the analytical theory and the design of numerical approximation schemes.

As in the Poisson case, a weak formulation is introduced in order to relax regularity requirements and to incorporate the boundary condition in a variational framework. Multiplying (4) by a test function $v \in H_0^1(\Omega)$ and integrating over Ω , the diffusion term is treated by integration by parts, while the convection and reaction terms are retained in their natural weak form. This leads to the following definition.

Definition 2.11 (Weak solution of the CDR problem). A function $u \in H_0^1(\Omega)$ is called a weak solution of (4) if

$$\int_{\Omega} \kappa \nabla u \cdot \nabla v \, dx + \int_{\Omega} (\beta \cdot \nabla u) v \, dx + \int_{\Omega} cuv \, dx = \int_{\Omega} fv \, dx, \quad \forall v \in H_0^1(\Omega). \quad (5)$$

The weak formulation (5) can be written in abstract variational form on the real Hilbert space $H_0^1(\Omega)$ by introducing the bilinear form $a_{\text{cdr}}(\cdot, \cdot)$ and the linear functional $\ell(\cdot)$. Continuity of both follows from the boundedness of the coefficients and standard inequalities. In contrast to the Poisson problem, coercivity of the bilinear form is not automatic due to the presence of the convection term.

A commonly used sufficient condition ensuring coercivity relates the reaction term to the divergence of the convection field.

Assumption 2.12 (Weak coercivity condition). Assume that

$$c(x) - \frac{1}{2} \nabla \cdot \beta(x) \geq 0 \quad \text{for almost every } x \in \Omega.$$

Under Assumption 2.12, the bilinear form associated with (5) satisfies a coercivity estimate in the $H_0^1(\Omega)$ norm.

Theorem 2.13 (Existence, uniqueness, and stability). Assume that Assumptions 2.10 and 2.12 hold. Then there exists a unique weak solution $u \in H_0^1(\Omega)$ of (5). Moreover, the solution depends continuously on the data in the sense that

$$\|u\|_{H_0^1(\Omega)} \leq C \|f\|_{L^2(\Omega)},$$

where $C > 0$ depends only on the coefficients and the domain Ω .

The assertion follows from the Lax–Milgram theorem applied to the variational problem on $H_0^1(\Omega)$; see [4, Sec. 5.8, pp. 299–303]. Continuity of the bilinear form is a consequence of the boundedness of κ , β , and c . Coercivity follows by setting $v = u$ in (5) and integrating the convection term by parts:

$$\int_{\Omega} (\beta \cdot \nabla u) u \, dx = -\frac{1}{2} \int_{\Omega} (\nabla \cdot \beta) u^2 \, dx,$$

where the boundary term vanishes due to the homogeneous Dirichlet condition. Assumption 2.12 then yields the required coercivity estimate.

In addition to well-posedness, convection–diffusion–reaction equations possess important qualitative properties. In particular, under suitable assumptions on the coefficients and the data, solutions satisfy a maximum principle analogous to that of the Poisson problem.

Theorem 2.14 (Maximum principle for the CDR problem). *Assume that Assumptions 2.10 and 2.12 hold, and let $u \in H_0^1(\Omega) \cap C^2(\Omega)$ be a classical solution of (4). If $f \leq 0$ almost everywhere in Ω , then*

$$\max_{x \in \Omega} u(x) \leq 0.$$

An analogous statement holds when $f \geq 0$.

Remark 2.15. The regularity assumption $u \in C^2(\Omega)$ is imposed for clarity. Extensions of the weak maximum principle to weak solutions can be found in classical elliptic theory; see [5, Ch. 3, pp. 31–35].

The qualitative behavior of solutions may change notably depending on the relative strength of diffusion and convection. When the diffusion coefficient κ is small compared to the magnitude of the convection field β , the problem enters a convection-dominated regime. This regime can be characterized by a large Péclet number,

$$\text{Pe} := \frac{\|\beta\|_{L^\infty(\Omega)} L}{\kappa},$$

where L denotes a characteristic length scale of the domain.

In convection-dominated regimes, the elliptic regularization provided by the diffusion operator becomes weak. As a consequence, solutions may exhibit sharp boundary or interior layers, and classical elliptic regularity estimates deteriorate uniformly with respect to the diffusion coefficient. The solution can still be continuous on $\bar{\Omega}$ or belong to $H^2(\Omega)$ under suitable assumptions, but derivatives and regularity constants may scale with inverse powers of κ as $\kappa \rightarrow 0$; see [7, Ch. 1, pp. 1–34]. These phenomena indicate that, while variational well-posedness remains valid, energy-based estimates alone provide limited information on the pointwise behavior of solutions.

The convection–diffusion–reaction problem therefore extends the Poisson model by introducing transport and reaction effects that alter the qualitative structure of the solution without destroying ellipticity under suitable assumptions. The resulting variational framework parallels that of the Poisson problem but requires additional conditions to ensure coercivity and stability. At the same time, the presence of weak diffusion highlights the need for qualitative tools capable of controlling solutions beyond integral norms. This observation motivates a separate discussion of maximum principles, which provide pointwise bounds and complement the variational analysis developed above.

2.4 Maximum Principles for Elliptic Problems

The limitations of purely energy-based estimates underscore the importance of qualitative tools that provide pointwise control of solutions. Maximum principles address this need by

imposing bounds that depend on the structure of the elliptic operator, the boundary conditions, and sign conditions on the data. These principles complement variational well-posedness results by excluding non-physical interior extrema and by ensuring sign preservation under appropriate hypotheses.

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. Consider a second-order linear operator of divergence form

$$\mathcal{L}u := - \sum_{i,j=1}^d \partial_i(a_{ij}(x) \partial_j u) + \sum_{i=1}^d b_i(x) \partial_i u + c(x) u, \quad (6)$$

where $A(x) = (a_{ij}(x))_{i,j=1}^d$ denotes the diffusion matrix, $b(x) = (b_i(x))_{i=1}^d$ is a convection field, and $c(x)$ is a reaction coefficient. This notation is consistent with the scalar CDR notation $A = \kappa I$ and $b = \beta$. The validity of maximum principles relies on structural properties of the operator $\mathcal{L}$, which are summarized below.

Assumption 2.16 (Ellipticity and boundedness of coefficients). Assume that $a_{ij} \in L^\infty(\Omega)$ and that there exists a constant $\lambda > 0$ such that

$$\xi^\top A(x) \xi \geq \lambda |\xi|^2 \quad \text{for all } \xi \in \mathbb{R}^d \text{ and for almost every } x \in \Omega.$$

Assume further that $b \in (L^\infty(\Omega))^d$ and $c \in L^\infty(\Omega)$.

Uniform ellipticity ensures that the second-order part of the operator provides elliptic regularization, while boundedness of the coefficients prevents the lower-order terms from overwhelming the principal part.

Assumption 2.17 (Sufficient sign condition). Assume that

$$c(x) - \frac{1}{2} \nabla \cdot b(x) \geq 0 \quad \text{for almost every } x \in \Omega.$$

Assumption 2.17 is a convenient sufficient condition under which pointwise bounds can be established. While not necessary in full generality, it aligns naturally with coercivity requirements encountered in the variational analysis of convection–diffusion–reaction problems.

Under Assumptions 2.16 and 2.17, maximum principles can be formulated for sufficiently regular solutions. Consider the Dirichlet boundary value problem

$$\begin{cases} \mathcal{L}u = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \quad (7)$$

If $u \in C^2(\Omega) \cap C(\bar{\Omega})$ satisfies $\mathcal{L}u \leq 0$ in Ω in the classical sense, then the weak maximum principle asserts that the maximum of u over $\bar{\Omega}$ is attained on $\partial\Omega$; see [5, Ch. 3, pp. 31–35]. A function w is called a subsolution if $\mathcal{L}w \leq f$ in the appropriate pointwise or weak sense and if it satisfies the corresponding boundary inequality. In particular, if $g \leq 0$ on $\partial\Omega$ and the sign assumptions of the weak maximum principle hold, then $u \leq 0$ throughout Ω in the homogeneous source case. The homogeneous problem used earlier is recovered by taking $g = 0$. These results exclude non-trivial interior extrema and provide qualitative information that is not accessible from energy estimates alone.

The general operator framework (6) clarifies the connection with the model problems discussed earlier. The Poisson operator corresponds to the special case $A(x) = I$, $b(x) \equiv 0$, and $c(x) \equiv 0$, for which Assumption 2.17 is satisfied trivially and the classical maximum principle reduces to the familiar extremum control for harmonic and subharmonic functions; see [4, Sec. 6.4, pp. 327–334] and [5, Ch. 3, pp. 31–35]. For the convection–diffusion–reaction operator studied in Section 2.3, the identification

$$A(x) = \kappa(x)I, \quad b(x) = \beta(x), \quad c(x) = c(x)$$

shows that uniform ellipticity follows from the positivity of κ , while Assumption 2.17 coincides with the sign/coercivity condition previously imposed. Consequently, the maximum principle for the convection–diffusion–reaction problem is consistent with the same structural hypotheses that guarantee well-posedness of the weak formulation.

Maximum principles are particularly relevant in convection-dominated regimes, where diffusion is weak relative to convection and solutions may develop sharp boundary or interior layers [7, Ch. 1, pp. 1–34]. In such regimes, elliptic regularization is reduced and energy estimates, although still valid, provide only integral control. When the continuous problem satisfies a maximum principle, pointwise bounds furnish an additional qualitative benchmark: violations of these bounds by an approximation indicate the presence of non-physical oscillations, even when convergence in energy norms is observed.

In summary, maximum principles provide pointwise bounds and sign preservation properties for elliptic boundary value problems under suitable structural assumptions. Within the present framework, they unify qualitative aspects of the Poisson and convection–diffusion–reaction models and supply a continuous benchmark that complements the variational analysis developed in Sections 2.2–2.3.

The preceding analysis establishes a rigorous baseline for steady elliptic boundary value problems. In particular, the variational framework identifies the natural energy space, provides well-posedness through coercivity and continuity of the associated bilinear forms, and clarifies the role of regularity assumptions in quantitative estimates. In addition, maximum principles supply qualitative, pointwise constraints that complement energy bounds and exclude non-physical interior extrema under suitable structural hypotheses.

These analytical properties are central when assessing approximation strategies for elliptic PDEs. In practical settings, the underlying problems may involve high-dimensional inputs, complex geometries, parameter dependence, or convection-dominated regimes in which solutions develop sharp layers and classical regularity deteriorates [7, Ch. 1, pp. 1–34]. Such features increase the difficulty of constructing accurate approximations and motivate alternative computational approaches that do not rely exclusively on traditional mesh-based discretization paradigms.

Physics-informed neural networks provide one such approach by representing the unknown solution through a parameterized neural network and by enforcing the governing equation and boundary conditions through an optimization objective. The next chapter introduces the necessary elements of machine learning and optimization and formulates the PINN methodology for the steady elliptic models considered in this chapter, thereby preparing the ground for a systematic analysis of learning-based approximations against the established analytical benchmarks.

3 Introduction to Machine Learning and Physics-Informed Neural Networks

The preceding section developed a variational and functional-analytic framework for steady elliptic boundary value problems. Weak formulations, well-posedness results, regularity estimates, and qualitative principles such as maximum bounds were established for representative model equations. These results characterize the structure of the continuous problems and provide a mathematical reference against which numerical approximations can be assessed.

Classical finite element methods construct approximation spaces by means of local basis functions defined on a spatial mesh. Finite difference methods, by contrast, approximate derivatives at mesh nodes through difference quotients. An alternative approach is obtained by replacing mesh-based discrete representations with global parametric families of functions and determining the approximation by solving an optimization problem in parameter space. In this setting, an unknown solution is represented by a function u_θ depending on finitely many parameters $\theta \in \mathbb{R}^p$, and the parameters are selected so as to minimize a loss functional that measures the violation of prescribed constraints. From a mathematical perspective, this procedure may be interpreted as a nonlinear least-squares approximation posed over a finite-dimensional trial space.

Neural networks constitute a particularly flexible class of parametric function families and have been extensively studied in the context of statistical learning and optimization; see [8, Ch. 6, pp. 164–223]. When the loss functional is constructed from the residuals of differential equations and boundary conditions rather than from labeled observational data, the resulting methods are referred to as physics-informed neural networks (PINNs). This methodology was introduced systematically in [1, 2] and further developed within the broader framework of physics-informed machine learning [3]. Within the present work, PINNs are interpreted as residual minimization schemes formulated over nonlinear parametric spaces and are analyzed in direct comparison with the elliptic models introduced previously.

The subsequent subsections provide the mathematical components required for this formulation, including optimization principles, neural network function spaces, and the construction of residual-based loss functionals and training algorithms.

3.1 Fundamentals of Machine Learning

Many problems in modern scientific computing can be formulated as the reconstruction of an unknown function from partial information by restricting the search to a finite-dimensional parametric family and selecting an element that minimizes a prescribed discrepancy functional. Such problems arise naturally in regression, statistical learning, and nonlinear approximation theory, where one seeks an unknown mapping by solving an optimization problem over a class of admissible candidates; see [8, Ch. 5, pp. 105–161] and [9, Ch. 1, pp. 1–66] for general background.

From a mathematical perspective, this structure is analogous to classical approximation methods. Given a parameter set $\Theta \subset \mathbb{R}^p$ and a family of functions

$$\mathcal{F}_\Theta := \{u_\theta : \theta \in \Theta\},$$

one defines a loss functional $\mathcal{L} : \mathcal{F}_\Theta \rightarrow \mathbb{R}$ that measures the discrepancy between a candidate function and the available information. The approximation problem is then formulated as

$$\min_{\theta \in \Theta} \mathcal{L}(u_\theta).$$

This viewpoint reduces function reconstruction to a finite-dimensional nonlinear optimization problem.

The same formulation is directly applicable to differential equations. Indeed, the numerical solution of a boundary value problem may be interpreted as the approximation of an

unknown function subject to constraints imposed by a differential operator and boundary conditions. Replacing classical mesh-based trial spaces by parametric families therefore leads to an optimization-based discretization strategy that parallels the variational formulations introduced in Section 2. Within this framework, the key ingredients consist of the choice of the parametric function space, the construction of an appropriate loss functional, and the design of efficient gradient-based optimization methods.

The following subsections introduce these components in a systematic manner and establish the optimization principles that underlie the formulation of physics-informed neural networks.

3.1.1 Learning problems and parametric approximation

Let $X \subset \mathbb{R}^d$ be a domain and let $y : X \rightarrow \mathbb{R}$ denote an unknown target function. The fundamental objective of a learning problem is to construct an approximation of y from partial information, which may consist of pointwise observations, statistical measurements, or structural constraints. From a mathematical viewpoint, this task can be interpreted as a nonlinear approximation problem in which the search for y is restricted to a prescribed class of admissible functions.

A standard strategy consists in selecting a finite-dimensional parametric family of candidate functions and determining the parameters by solving an optimization problem; see [8, Ch. 5, pp. 105–161] and [9, Ch. 1, pp. 1–66]. This approach leads naturally to the following notion.

Definition 3.1 (Hypothesis class). Let $\Theta \subset \mathbb{R}^p$ be a parameter set. A family

$$\mathcal{H} := \{\Phi_\theta : X \rightarrow \mathbb{R} \mid \theta \in \Theta\}$$

is called a hypothesis class. Each element Φ_θ is referred to as a parametric model.

The introduction of $\mathcal{H}$ reduces the reconstruction of the unknown function to the selection of an element of a finite-dimensional nonlinear manifold. Given a discrepancy functional $\mathcal{L} : \mathcal{H} \rightarrow \mathbb{R}$ that measures the error of a candidate with respect to the available information, the approximation problem takes the form

$$\min_{\theta \in \Theta} \mathcal{L}(\Phi_\theta).$$

Consequently, learning may be formulated as a deterministic optimization problem over the parameter space Θ .

This perspective parallels classical numerical approximation methods. In particular, the hypothesis class $\mathcal{H}$ plays the role of a trial space, analogous to the finite-dimensional spaces employed in Galerkin or finite element discretizations, while the loss functional $\mathcal{L}$ determines the criterion according to which candidates are selected. The choice of the parametric family therefore controls the expressive power of the approximation, and the optimization procedure determines how effectively this expressive capacity is exploited.

Neural networks provide a flexible and expressive realization of such parametric families and will serve as the concrete hypothesis classes in the constructions that follow.

3.1.2 Supervised regression and empirical risk minimization

Assume that information about the unknown target function $y : X \rightarrow \mathbb{R}$ is available in the form of pointwise samples

$$\{(x_i, y_i)\}_{i=1}^N \subset X \times \mathbb{R}, \quad y_i \approx y(x_i).$$

The task of supervised regression is to determine a function within the hypothesis class $\mathcal{H}$ that approximates these observations in a prescribed sense.

Let $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+$ be a nonnegative loss function measuring the discrepancy between a prediction and the corresponding observation. At the continuous level, one may introduce a risk functional that quantifies the average prediction error over the domain. Formally, for a probability measure μ on X , the population risk of a model Φ_θ is given by

$$\mathcal{J}(\theta) = \int_X \ell(\Phi_\theta(x), y(x)) \, d\mu(x).$$

Minimization of $\mathcal{J}$ defines an ideal approximation problem. Since the distribution μ and the exact function y are typically unknown, this functional cannot be evaluated directly in practice.

The standard approach replaces the integral by a discrete approximation based on the available samples, leading to empirical risk minimization (ERM); see [8, Ch. 5, pp. 105–161] and [9, Ch. 1, pp. 1–66].

Definition 3.2 (Empirical risk). Given samples $\{(x_i, y_i)\}_{i=1}^N$, the empirical risk of a model Φ_θ is defined by

$$\mathcal{J}_N(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(\Phi_\theta(x_i), y_i).$$

Any minimizer

$$\theta^* \in \arg \min_{\theta \in \Theta} \mathcal{J}_N(\theta)$$

is called an empirical risk minimizer.

The functional $\mathcal{J}_N$ can be interpreted as a Monte Carlo approximation of the population risk, and the learning problem is thereby reduced to a finite-dimensional optimization problem in the parameter space Θ .

For real-valued outputs, the quadratic loss

$$\ell(a, b) = (a - b)^2$$

is of particular importance. In this case, the empirical risk takes the form of a discrete least-squares functional.

Definition 3.3 (Mean squared error). The mean squared error (MSE) associated with Φ_θ is defined by

$$\text{MSE}(\theta) = \frac{1}{N} \sum_{i=1}^N (\Phi_\theta(x_i) - y_i)^2.$$

Minimization of the MSE coincides with the solution of a least-squares problem and endows the parameter space with a natural Hilbert space structure induced by the discrete ℓ^2 norm of the residual vector. This least-squares formulation provides the prototype for the loss constructions employed later: in physics-informed neural networks, the data residuals $(\Phi_\theta(x_i) - y_i)$ are replaced by residuals of differential operators and boundary conditions, while the overall optimization structure remains unchanged.

3.1.3 Risk functionals and approximation properties

The empirical risk introduced in the preceding subsection measures approximation quality only at the sampled points. For the assessment of global approximation properties, it is necessary to compare this discrete quantity with the corresponding continuous error functional.

Recall that, at the continuous level, the quality of a model Φ_θ may be quantified by a population risk of the form

$$\mathcal{J}(\theta) = \int_X \ell(\Phi_\theta(x), y(x)) \, d\mu(x),$$

where μ is a measure on X . The empirical risk $\mathcal{J}_N(\theta)$ can then be interpreted as a discrete quadrature or Monte Carlo approximation of this integral. Consequently, the training process replaces the minimization of a continuous functional by the minimization of a sampled surrogate.

This distinction gives rise to two different notions of error. The empirical risk describes the discrepancy on the discrete data set, whereas the population risk characterizes the approximation error over the entire domain. Control of the former does not automatically imply control of the latter, since the two functionals coincide only in the limit of dense or sufficiently representative sampling.

From the viewpoint of approximation theory, the behavior of these risks depends on the expressive properties of the hypothesis class $\mathcal{H}$. If $\mathcal{H}$ is too small, the target function cannot be approximated with sufficient accuracy, and both risks remain large. If $\mathcal{H}$ is excessively rich, minimization of the discrete functional may lead to highly oscillatory or unstable approximants that interpolate the sampled data while exhibiting poor global behavior. This trade-off between approximation capability and stability is closely related to classical regularization principles and to the conditioning of nonlinear least-squares problems; see [9, Ch. 3, pp. 137–182] and [10, Ch. 10, pp. 245–269].

These considerations have direct analogues in the solution of differential equations by residual minimization. When residuals are enforced only at finitely many collocation points, the discrete objective controls the equation locally but does not by itself guarantee uniform or global accuracy. The quality of the resulting approximation therefore depends simultaneously on the richness of the parametric function class and on the distribution of sampling points. This parallel between supervised regression and residual-based PDE approximation provides the conceptual link that motivates the formulation of physics-informed neural networks developed in the subsequent section.

3.1.4 Gradient-based optimization

The empirical risk minimization problems introduced above reduce learning to the minimization of a finite-dimensional functional

$$\min_{\theta \in \Theta} \mathcal{J}_N(\theta), \quad \Theta \subset \mathbb{R}^p,$$

where p denotes the number of trainable parameters. In modern neural network models, p may be very large, often ranging from thousands to millions. Consequently, classical direct minimization techniques are generally infeasible, and iterative first-order optimization methods are employed instead; see [10, Ch. 3, pp. 48–82] and [11].

Throughout this work, the loss functional $\mathcal{J}_N$ is assumed to be differentiable with respect to the parameters. Under this assumption, descent directions may be constructed from the gradient $\nabla_{\theta} \mathcal{J}_N(\theta)$, which yields the following standard scheme.

Definition 3.4 (Gradient descent). Given an initial parameter $\theta^{(0)} \in \Theta$, the gradient descent iteration is defined by

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k \nabla_{\theta} \mathcal{J}_N(\theta^{(k)}),$$

where $\{\eta_k\}_{k \geq 0}$ is a sequence of positive step sizes.

This iteration may be viewed as an explicit time discretization of the gradient flow

$$\dot{\theta}(t) = -\nabla_{\theta} \mathcal{J}_N(\theta(t)),$$

which corresponds to the steepest descent dynamics in the Euclidean metric. Under suitable regularity assumptions, such flows decrease the objective monotonically and converge to stationary points of $\mathcal{J}_N$.

In practical learning problems, however, the evaluation of the full gradient may be computationally expensive when the loss involves a large number of samples. To reduce this cost, stochastic or mini-batch approximations are commonly used, leading to stochastic gradient descent (SGD) and related algorithms. These methods replace the full gradient by an unbiased estimator computed from a subset of samples, thereby lowering the per-iteration complexity while preserving convergence properties in expectation; see [11]. Adaptive variants such as Adam modify the step size using local estimates of first and second moments of the gradient and have proved particularly effective in large-scale settings [12].

A further structural aspect arises from the composite nature of neural network models. The mapping $\theta \mapsto \Phi_\theta(x)$ is defined by repeated compositions of affine transformations and nonlinear activation functions. Gradients of the loss with respect to θ are therefore obtained efficiently by reverse-mode automatic differentiation, commonly referred to as backpropagation [13, 14]. The computational cost of evaluating $\nabla_\theta \mathcal{J}_N$ is of the same order as that of a single forward evaluation of the network, which makes first-order methods feasible even for high-dimensional parameter spaces.

From a mathematical perspective, the training of parametric models may thus be interpreted as the solution of a high-dimensional, generally non-convex nonlinear least-squares problem by gradient-type methods. The effectiveness of the resulting approximation depends not only on the expressive properties of the hypothesis class but also on the behavior of the underlying optimization dynamics. These considerations remain valid when residuals of differential equations replace observational data, and they form the algorithmic foundation for the physics-informed neural networks introduced subsequently.

3.1.5 Automatic differentiation

Parametric models such as neural networks are constructed as compositions of affine mappings and nonlinear activation functions. Consequently, both the model output and the associated loss functionals are expressed as compositions of elementary differentiable operations. The efficient and accurate computation of derivatives of such composite mappings is therefore a fundamental requirement, not only for gradient-based optimization but also for the evaluation of differential operators that arise in scientific computing.

Let $F : \mathbb{R}^p \rightarrow \mathbb{R}$ denote a differentiable function defined through a sequence of elementary operations,

$$F = f_m \circ f_{m-1} \circ \cdots \circ f_1,$$

where each f_i represents an elementary mapping such as an affine transformation or a scalar nonlinearity. By repeated application of the chain rule, derivatives of F can be expressed as products of local Jacobians. Automatic differentiation (AD) provides a systematic computational framework for carrying out this procedure exactly at the level of machine arithmetic; see [14].

Definition 3.5 (Automatic differentiation). Automatic differentiation is a computational methodology for evaluating derivatives of composite functions by propagating derivative information through the computational graph representing the function, using repeated applications of the chain rule.

The essential feature of automatic differentiation is that it does not approximate derivatives by difference quotients. Unlike finite-difference schemes, which introduce truncation and discretization errors depending on a chosen step size, AD computes derivatives analytically by differentiating each elementary operation. As a result, the computed derivatives coincide with the exact derivatives of the represented function up to machine precision.

Two principal modes of automatic differentiation are commonly distinguished. Forward-mode differentiation propagates directional derivatives alongside function evaluations and is well

suited for problems with few input variables. Reverse-mode differentiation, often referred to as backpropagation, accumulates gradients with respect to many parameters at a computational cost comparable to a single evaluation of the function. For neural networks with a large number of parameters and a scalar loss, reverse-mode differentiation is therefore particularly efficient; see [13, 14].

From the standpoint of optimization, automatic differentiation enables the practical computation of gradients $\nabla_{\theta} \mathcal{J}_N(\theta)$ required by first-order methods discussed previously. From the standpoint of differential equations, it allows the evaluation of spatial derivatives of the parametric approximation $u_{\theta}(x)$ of arbitrary order by differentiating with respect to the input variable x . Hence, quantities such as $\nabla u_{\theta}(x)$ or $\Delta u_{\theta}(x)$ can be obtained directly without introducing auxiliary discretizations or numerical differentiation schemes.

The availability of exact derivatives for both parameters and inputs is a structural property that distinguishes neural network models from many classical approximation methods. In particular, it permits differential operators to be incorporated directly into optimization-based formulations of boundary value problems, a feature that will be essential for the construction of physics-informed neural networks in the subsequent section.

3.2 Neural Networks as Parametric Function Spaces

The approximation framework employed for elliptic boundary value problems is fundamentally based on the selection of admissible trial spaces in which approximate solutions are sought. In classical numerical analysis, these spaces are typically chosen as finite-dimensional subspaces of Sobolev spaces, constructed for instance from piecewise polynomial basis functions or other prescribed global expansions. The underlying principle is to replace the infinite-dimensional solution space by a parametrized family of functions whose elements are computationally accessible and whose complexity can be controlled a priori; see [6, Ch. 1, pp. 1–38].

From a functional-analytic perspective, this procedure may be viewed abstractly as follows. Let V denote the continuous solution space (for example $H_0^1(\Omega)$). One selects a finite-dimensional set of functions

$$V_h \subset V,$$

and seeks an approximation within V_h by solving an associated optimization or variational problem. Thus, the construction of the approximation method reduces to the specification of a suitable function class that serves as the search space.

Learning-based approaches adopt the same structural viewpoint but replace linear subspaces generated by fixed basis functions with nonlinear families of parameterized mappings. Instead of expressing an approximation as a linear combination of predetermined basis elements, one considers compositions of affine transformations and nonlinear activation functions depending on finitely many parameters. These parameterized constructions define sets of functions that can naturally be interpreted as nonlinear approximation spaces.

Neural networks provide a particularly flexible realization of such parametric function families. Given a parameter set $\Theta \subset \mathbb{R}^p$, a neural network architecture determines a mapping

$$\Theta \ni \theta \mapsto u_{\theta} : X \rightarrow \mathbb{R},$$

thereby inducing a hypothesis space

$$\mathcal{F}_{\Theta} := \{u_{\theta} \mid \theta \in \Theta\}.$$

This set plays the same conceptual role as a classical trial space: it specifies the admissible candidates among which an approximate solution is sought. The essential difference is that $\mathcal{F}_{\Theta}$ is generally nonlinear and highly expressive, rather than a linear subspace.

The purpose of the present subsection is to formalize neural networks within this functional framework and to justify, from an analytical standpoint, their use as parametric approximation

spaces. To this end, we introduce their algebraic structure, regularity properties, and approximation capabilities, following standard treatments in the machine learning and approximation theory literature; see [8, Ch. 6, pp. 164–223], [15], and [16].

3.2.1 Neural networks as parameterized mappings

Let $d \geq 1$ and let $\Omega \subset \mathbb{R}^d$ be a bounded domain. In the present context, a neural network is regarded purely as a parametric family of functions. Each choice of parameters determines a mapping $\Omega \rightarrow \mathbb{R}$, and the collection of all such mappings constitutes the admissible search space for approximation.

We begin by introducing the standard feedforward architecture.

Definition 3.6 (Feedforward neural network). Let $L \geq 1$ and let $n_0 = d, n_1, \dots, n_L = 1$ be positive integers describing the layer widths. For each layer $\ell = 1, \dots, L$, let

$$W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}, \quad b_\ell \in \mathbb{R}^{n_\ell},$$

and let $\rho : \mathbb{R} \rightarrow \mathbb{R}$ be a prescribed activation function applied componentwise.

For parameters

$$\theta = \{W_\ell, b_\ell\}_{\ell=1}^L,$$

the associated feedforward neural network is the mapping

$$\Phi_\theta : \Omega \rightarrow \mathbb{R}, \quad \Phi_\theta(x) = T_L \circ \rho \circ T_{L-1} \circ \dots \circ \rho \circ T_1(x), \quad T_\ell(z) = W_\ell z + b_\ell.$$

The parameters θ consist of finitely many weights and biases, so that they may be identified with a vector in $\mathbb{R}^p$ for some $p \in \mathbb{N}$. Consequently, the architecture induces a mapping

$$\Theta \subset \mathbb{R}^p \longrightarrow \{\text{functions on } \Omega\}, \quad \theta \mapsto \Phi_\theta.$$

Definition 3.7 (Neural network hypothesis class). For a fixed architecture and activation function, the associated hypothesis class is defined by

$$\mathcal{H}_{\text{NN}} = \{\Phi_\theta : \Omega \rightarrow \mathbb{R} \mid \theta \in \Theta\}.$$

The set $\mathcal{H}_{\text{NN}}$ is therefore a finite-dimensional parametric family of functions. In contrast to polynomial or finite element spaces, it is not a linear subspace but rather a nonlinear manifold embedded in an ambient function space such as $C(\overline{\Omega})$ or $H^1(\Omega)$. From an approximation-theoretic viewpoint, it plays the same conceptual role as a classical trial space, namely that of restricting the search for an unknown function to a computationally manageable class.

The regularity of the functions in $\mathcal{H}_{\text{NN}}$ is inherited from the activation function.

Proposition 3.8 (Regularity and smooth parameter dependence). *Assume that $\rho \in C^k(\mathbb{R})$ for some $k \geq 1$. Then for every $\theta \in \Theta$, the mapping Φ_θ belongs to $C^k(\Omega)$. Moreover, for each fixed $x \in \Omega$, the map*

$$\Theta \ni \theta \mapsto \Phi_\theta(x)$$

is of class C^k .

The claim follows from the fact that affine maps are smooth and compositions of C^k functions remain C^k ; see also the standard discussion of multilayer network mappings in [8, Ch. 6, pp. 164–223].

The preceding result ensures that, provided smooth activation functions are employed, neural network outputs possess classical derivatives of the orders required for the evaluation of differential operators. This regularity property is essential in residual-based formulations, where spatial derivatives of the approximation enter explicitly into the loss functional.

3.2.2 Regularity properties

In applications to partial differential equations, the admissible function class must possess sufficient smoothness so that the differential operators appearing in the model are well defined. In particular, second-order elliptic operators involve first- and second-order spatial derivatives, and therefore require the approximation functions to admit derivatives in an appropriate classical or weak sense.

As established in the previous subsection, if the activation function ρ is of class C^k , then every realization Φ_θ belongs to $C^k(\Omega)$. This classical regularity immediately guarantees the existence of pointwise derivatives of order up to k . However, for the variational formulations considered earlier, it is often more natural to interpret regularity in terms of Sobolev spaces.

Proposition 3.9 (Sobolev regularity). *Let $k \geq 1$ and assume that $\rho \in C^k(\mathbb{R})$. Assume in addition that the derivatives of the network realization up to order k extend continuously to $\bar{\Omega}$. Then for every $\theta \in \Theta$ one has*

$$\Phi_\theta \in H^m(\Omega) \quad \text{for all } m \leq k.$$

In particular, if $\rho \in C^2(\mathbb{R})$, then $\Phi_\theta \in H^2(\Omega)$ and the quantities $\nabla \Phi_\theta$ and $\Delta \Phi_\theta$ are well defined in $L^2(\Omega)$.

Indeed, under the stated extension assumption, all classical derivatives up to order k are continuous on the compact set $\bar{\Omega}$ and are therefore bounded. Since Ω is bounded, these derivatives are square-integrable. The assumption cannot be replaced by mere membership in $C^k(\Omega)$ on an open domain: for instance, $x \mapsto 1/x$ is smooth on $(0, 1)$ but does not belong to $L^2(0, 1)$.

This observation ensures that neural network realizations are admissible in the same functional settings employed for the elliptic boundary value problems discussed previously. In particular, for smooth activations the network outputs belong to $H_0^1(\Omega)$ after incorporation of boundary conditions and therefore constitute legitimate trial functions in variational formulations. Moreover, the existence of second-order derivatives permits the direct evaluation of differential operators such as the Laplacian, which is required in residual-based constructions.

3.2.3 Approximation properties

In addition to regularity, a function class intended for numerical approximation must possess sufficient expressive power to approximate a broad range of target functions. Within approximation theory, this requirement is typically formulated in terms of density or convergence properties with respect to appropriate norms. For neural networks, such properties are provided by a family of classical results known collectively as universal approximation theorems; see [15, 16, 17].

The first results of this type concern approximation in spaces of continuous functions.

Theorem 3.10 (Universal approximation in $C(\Omega)$). *Let $\Omega \subset \mathbb{R}^d$ be compact and let $\rho : \mathbb{R} \rightarrow \mathbb{R}$ be continuous and non-polynomial. Then the class of single-hidden-layer neural networks is dense in $C(\Omega)$ with respect to the uniform norm. That is, for every $f \in C(\Omega)$ and every $\varepsilon > 0$, there exists $\Phi_\theta \in \mathcal{H}_{\text{NN}}$ such that*

$$\|f - \Phi_\theta\|_{L^\infty(\Omega)} < \varepsilon.$$

This theorem shows that neural network hypothesis classes are sufficiently rich to approximate arbitrary continuous functions on bounded domains. A proof and several variants are given in [17, pp. 143–195]; see also the original density results [15, 16]. However, for the analysis of elliptic boundary value problems, stronger norms are typically required. In particular, variational formulations are posed in Sobolev spaces such as $H^1(\Omega)$ or $H^2(\Omega)$, and therefore approximation properties must be understood in these spaces as well.

Such extensions follow from the density of smooth functions in Sobolev spaces together with the density result above.

Proposition 3.11 (Approximation in Sobolev spaces). *Assume that $\rho \in C^k(\mathbb{R})$ for some $k \geq 1$ and that Ω is bounded with Lipschitz boundary. Then, for every $m \leq k$, the union of neural network classes obtained by allowing sufficiently many neurons and parameters is dense in $H^m(\Omega)$. In particular, for any $u \in H^m(\Omega)$ and $\varepsilon > 0$ there exists a network realization Φ_θ from such a sufficiently large architecture such that*

$$\|u - \Phi_\theta\|_{H^m(\Omega)} < \varepsilon.$$

Sobolev approximation requires approximation of the function and its derivatives, and therefore does not follow from the $C(\overline{\Omega})$ universal approximation theorem alone. Under suitable smooth activation functions and sufficiently rich architectures, density results in Sobolev norms are available in the approximation theory literature; see [17, pp. 143–195].

Beyond qualitative density, quantitative error estimates have also been established. For instance, deep ReLU networks admit explicit convergence rates for the approximation of Sobolev-regular functions, with approximation errors depending polynomially on the number of parameters; see [18, pp. 103–114]. Such results indicate that increasing depth can notably enhance the efficiency of representation.

The regularity and approximation results established above justify interpreting the neural network hypothesis class $\mathcal{H}_{\text{NN}}$ as a nonlinear trial space for function approximation. Analogously to classical finite element spaces, the search for an unknown function is restricted to a finite-dimensional family parameterized by $\theta \in \Theta$, and the parameters are determined by minimizing an appropriate functional. The principal distinction is that $\mathcal{H}_{\text{NN}}$ forms a nonlinear manifold rather than a linear subspace. Nevertheless, the essential variational structure remains identical: approximation is achieved through optimization within a prescribed function class. This functional viewpoint provides the mathematical foundation for the residual-based formulations introduced in the next subsection.

3.3 Physics-Informed Neural Networks (PINNs)

The preceding subsections established a mathematical framework in which neural networks are interpreted as finite-dimensional parametric families of functions and their training is formulated as the minimization of empirical risk functionals by gradient-based optimization. Within this setting, the approximation of an unknown function reduces to the selection of parameters that minimize a prescribed loss over a nonlinear trial space. This viewpoint provides a direct connection between function approximation and optimization and permits differential equation problems to be reformulated as minimization problems over admissible function classes rather than as the construction of discrete algebraic systems.

Physics-informed neural networks constitute a class of methods that adopt this variational-optimization perspective for the numerical treatment of differential equations. In the standard approach considered first below, the unknown solution of a boundary value problem is represented by a neural network, and the governing equations together with the associated boundary conditions are enforced through residual terms in the loss functional. Other PINN variants use augmented loss functionals, weighted residuals, or hard-constrained boundary conditions in which Dirichlet data are built directly into the ansatz. The resulting objective therefore quantifies violations of analytical constraints induced by the differential operator rather than discrepancies with labeled observations. This methodology was introduced systematically in [1, 2] and further developed in the broader context of physics-informed machine learning [3].

Accordingly, the construction of PINNs may be interpreted as a residual-based discretization posed over a nonlinear parametric function space, in which the continuous boundary value problem is replaced by an optimization problem in parameter space. The following subsections provide a precise formulation of this procedure, including the representation of the solution by neural networks, the evaluation of differential operators, the definition of residual-based loss functionals, and the associated gradient-based training algorithm.

3.3.1 Learning paradigm underlying physics-informed neural networks

The formulation of physics-informed neural networks can be interpreted within the general framework of statistical learning theory. In this framework, an unknown function is approximated by selecting an element from a parameterized hypothesis class through the minimization of a suitable loss functional; see, for example, [9, Ch. 1, pp. 1–66] and [8, Ch. 5, pp. 105–161]. Let $\Theta \subset \mathbb{R}^p$ denote a parameter space and let

$$\mathcal{H} := \{u_\theta : \Omega \rightarrow \mathbb{R} \mid \theta \in \Theta\}$$

be the family of functions induced by a fixed neural network architecture. The learning problem consists in determining parameters θ such that u_θ provides an accurate approximation of a desired target function.

In classical supervised learning, one is given labeled samples

$$\{(x_i, y_i)\}_{i=1}^N \subset \Omega \times \mathbb{R},$$

where y_i represent observations of the unknown function at inputs x_i . Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+$, the parameters are obtained by solving the empirical risk minimization problem

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \ell(u_\theta(x_i), y_i). \quad (8)$$

This formulation replaces an inaccessible population error by a finite average over samples and leads naturally to gradient-based optimization procedures; see [13, 12, 11].

In many problems arising in scientific computing, however, such labeled observations are unavailable. Instead, the behavior of the system is characterized by known physical principles expressed through differential equations and boundary conditions. The unknown solution is therefore specified implicitly by the satisfaction of these constraints rather than by prescribed data values. This observation motivates replacing the data misfit in (8) by a residual-based criterion derived directly from the governing model.

Consider a boundary value problem of the form

$$\mathcal{L}u = f \quad \text{in } \Omega, \quad \mathcal{B}u = g \quad \text{on } \partial\Omega,$$

where $\mathcal{L}$ is a differential operator and $\mathcal{B}$ represents boundary conditions. For an approximation $u_\theta \in \mathcal{H}$, the quantities

$$\mathcal{L}u_\theta - f, \quad \mathcal{B}u_\theta - g$$

serve as natural measures of model inconsistency. At the continuous level, this leads to the least-squares functional

$$\mathcal{J}(\theta) = \|\mathcal{L}u_\theta - f\|_{L^2(\Omega)}^2 + \|\mathcal{B}u_\theta - g\|_{L^2(\partial\Omega)}^2. \quad (9)$$

Since the integrals in (9) are typically not available in closed form, they are approximated by empirical averages over collocation points. Let $\{x_i\}_{i=1}^{N_r} \subset \Omega$ and $\{x_j^\partial\}_{j=1}^{N_b} \subset \partial\Omega$. The discrete objective functional is then given by

$$\mathcal{J}_h(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} |\mathcal{L}u_\theta(x_i) - f(x_i)|^2 + \frac{1}{N_b} \sum_{j=1}^{N_b} |\mathcal{B}u_\theta(x_j^\partial) - g(x_j^\partial)|^2. \quad (10)$$

This objective has the same structural form as the empirical risk (8) and leads to a finite-dimensional optimization problem in the parameter space Θ .

This motivates the following definition.

Definition 3.12 (Standard physics-informed neural network). A standard physics-informed neural network is a parameterized function $u_\theta \in \mathcal{H}$ whose parameters are obtained by minimizing a residual-based empirical loss of the form (10), where the loss functional is derived from the governing differential operator and the associated boundary conditions.

The adjective “standard” is used because other approaches, including hard-constrained boundary conditions and modified or augmented loss functionals, are also referred to as PINNs.

The resulting optimization problem is typically solved by first-order gradient methods. Gradients of $\mathcal{J}_h$ with respect to θ are computed efficiently by backpropagation combined with automatic differentiation, which enables exact evaluation of derivatives of the network output and of the differential operator; see [13, 14]. Consequently, the determination of the approximation reduces to optimization in the parameter space Θ under a residual-based objective functional.

3.3.2 Neural network representation of the PDE solution

Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain and let

$$u : \bar{\Omega} \rightarrow \mathbb{R}$$

denote the exact solution of the boundary value problem under consideration. In the physics-informed formulation, this function is approximated by a parameterized neural network

$$u_\theta : \bar{\Omega} \rightarrow \mathbb{R}, \quad \theta \in \Theta \subset \mathbb{R}^p,$$

where the spatial coordinate x serves as the input and $u_\theta(x)$ represents the corresponding approximate solution value.

The architecture induces the hypothesis class

$$\mathcal{F}_\Theta := \{u_\theta : \theta \in \Theta\},$$

which forms a nonlinear finite-dimensional manifold embedded in function spaces such as $L^2(\Omega)$ or $H^1(\Omega)$. The approximation of the PDE solution is therefore reduced to selecting an element of $\mathcal{F}_\Theta$ through optimization in the parameter space Θ . In this sense, $\mathcal{F}_\Theta$ plays the role of a nonlinear trial space analogous to the finite-dimensional spaces employed in classical Galerkin and finite element methods.

Throughout this work, u_θ is chosen as a fully connected feedforward neural network of depth L of the form

$$u_\theta(x) = W_L \rho(W_{L-1} \rho(\cdots \rho(W_1 x + b_1) \cdots) + b_{L-1}) + b_L,$$

where ρ denotes a nonlinear activation function and θ collects all weights and biases; see [8, Ch. 6, pp. 164–223] and [9, Ch. 5, pp. 225–290]. This construction yields a parametric family of functions obtained by compositions of affine mappings and nonlinear transformations.

In principle, different layers may use different activation functions. In some architectures, a special activation function in the final layer is used to enforce qualitative properties such as bounds or boundary conditions; see, for example, the bound-preserving constructions in [19].

The suitability of neural networks as approximation spaces is supported by classical results from approximation theory. Universal approximation theorems assert that such networks are dense in spaces of continuous functions on compact domains [15, 16]. Moreover, quantitative analyses establish convergence rates in Sobolev norms and demonstrate advantages of deep architectures for approximating smooth functions; see [17, 18]. Consequently, the class $\mathcal{F}_\Theta$ is sufficiently expressive to approximate a broad range of admissible solutions.

Since elliptic differential operators involve spatial derivatives, regularity of the representation is essential. If the activation function ρ is sufficiently smooth, then $u_\theta \in C^k(\Omega)$ for all required orders k , ensuring that derivatives appearing in the operator can be evaluated pointwise. This regularity permits the direct computation of differential residuals used in the loss functional.

The mapping $x \mapsto u_\theta(x)$ is defined globally on Ω and does not rely on a spatial mesh or local basis functions. Accordingly, the neural network defines a mesh-free global trial space on which the residual minimization procedure is carried out.

3.3.3 Automatic differentiation and residual evaluation

Residual-based formulations for differential equations require the evaluation of differential operators acting on the parametric approximation u_θ . Since elliptic operators involve first- and second-order spatial derivatives, it is necessary to ensure both the existence of these derivatives for the neural network representation and their consistent numerical evaluation.

Let $\mathcal{L}$ denote the differential operator associated with the boundary value problem and let f be the corresponding source term.

Definition 3.13 (Residual function). For $\theta \in \Theta$, the residual associated with the neural network approximation u_θ is defined by

$$r_\theta(x) := \mathcal{L}u_\theta(x) - f(x), \quad x \in \Omega.$$

Whenever the derivatives required by $\mathcal{L}$ exist, the residual is well defined pointwise and can be interpreted as an element of $L^2(\Omega)$ or of a suitable Sobolev space.

For interior collocation points this pointwise interpretation is justified whenever the network realization is sufficiently differentiable in Ω . If derivatives are used in boundary residuals, they are interpreted pointwise on $\partial\Omega$ only when the corresponding derivatives extend continuously to the boundary.

The computation of r_θ therefore reduces to evaluating spatial derivatives of u_θ . For instance, the Poisson operator involves second-order derivatives through the Laplacian, while convection–diffusion–reaction operators additionally contain first-order terms.

The required interior regularity of u_θ follows directly from the smoothness of the activation function. In particular, by the regularity result established in Section 3.2, if the activation ρ belongs to $C^k(\mathbb{R})$ for some $k \geq 1$, then $u_\theta \in C^k(\Omega)$ for every $\theta \in \Theta$. Consequently, all derivatives appearing in $\mathcal{L}u_\theta$ exist in the classical sense.

Although these derivatives can in principle be obtained by repeated symbolic differentiation, practical implementations rely on automatic differentiation, as introduced above. Unlike finite-difference schemes, automatic differentiation does not introduce a mesh width or truncation error for derivative evaluation. The application of $\mathcal{L}$ to u_θ is therefore performed directly on the continuous parametric representation, while reverse-mode differentiation supplies the parameter gradients needed during training; see [14, 13].

The residual function r_θ obtained in this manner forms the basic component of the loss functional used for training.

3.3.4 Residual minimization and loss functional formulation

Given a parametric representation u_θ of the solution and a consistent procedure for evaluating derivatives, the construction of a physics-informed neural network is formulated through the definition of a residual-based objective functional. The approximation problem is thereby interpreted as a least-squares minimization posed over the nonlinear trial space introduced previously.

Let $\mathcal{L}$ denote the differential operator associated with the boundary value problem

$$\mathcal{L}u = f \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega.$$

For any sufficiently regular function v defined on $\overline{\Omega}$, the quantities

$$\mathcal{L}v - f \quad \text{and} \quad v|_{\partial\Omega} - g$$

represent the interior and boundary residuals, respectively. A global measure of consistency is obtained by combining these residuals in L^2 norms.

Definition 3.14 (Continuous residual functional). Let $\mathcal{F}_\Theta$ be the neural network trial space defined in Section 3.3.2. The continuous residual functional $\mathcal{J} : \mathcal{F}_\Theta \rightarrow \mathbb{R}$ is defined by

$$\mathcal{J}(v) := \|\mathcal{L}v - f\|_{L^2(\Omega)}^2 + \|v - g\|_{L^2(\partial\Omega)}^2. \quad (11)$$

The functional (11) corresponds to a least-squares formulation of the boundary value problem. If u denotes the exact solution, then $\mathcal{J}(u) = 0$. More generally, minimizers of $\mathcal{J}$ satisfy the governing equation and the boundary condition in an L^2 sense. For homogeneous Dirichlet data, this formula reduces to the case $g = 0$. In hard-constrained PINN formulations, the boundary condition is instead enforced by the ansatz and the boundary residual term can be omitted or modified.

Because the neural network representation yields a finite-dimensional parametric family

$$\mathcal{F}_\Theta = \{u_\theta : \theta \in \Theta \subset \mathbb{R}^p\},$$

the minimization of $\mathcal{J}$ reduces to an optimization problem over the parameter space Θ .

Definition 3.15 (Continuous PINN training problem). Determine $\theta^* \in \Theta$ such that

$$\theta^* \in \arg \min_{\theta \in \Theta} \mathcal{J}(u_\theta).$$

The integrals appearing in (11) are generally not available in closed form and are therefore approximated by numerical quadrature based on collocation points. When the points are sampled randomly, this procedure yields Monte Carlo estimators of the L^2 norms and leads to an empirical risk functional; see [9, Ch. 1, pp. 1–66] and [8, Ch. 5, pp. 105–161].

Let

$$\{x_i^\Omega\}_{i=1}^{N_\Omega} \subset \Omega, \quad \{x_j^{\partial\Omega}\}_{j=1}^{N_{\partial\Omega}} \subset \partial\Omega,$$

be interior and boundary samples, and define the pointwise residuals

$$r_\theta(x) := \mathcal{L}u_\theta(x) - f(x), \quad b_\theta(x) := u_\theta(x) - g(x).$$

Definition 3.16 (Empirical PINN loss). The empirical loss functional $\mathcal{J}_N : \Theta \rightarrow \mathbb{R}$ is defined by

$$\mathcal{J}_N(\theta) = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} |r_\theta(x_i^\Omega)|^2 + \frac{1}{N_{\partial\Omega}} \sum_{j=1}^{N_{\partial\Omega}} |b_\theta(x_j^{\partial\Omega})|^2. \quad (12)$$

The training problem is consequently formulated as

$$\min_{\theta \in \Theta} \mathcal{J}_N(\theta), \quad (13)$$

which constitutes an empirical risk minimization problem in the sense of Section 3.1. Gradients of $\mathcal{J}_N$ with respect to θ are computed by automatic differentiation and evaluated efficiently by reverse-mode differentiation (backpropagation); see [13, 14]. The parameters are updated by first-order stochastic optimization methods such as stochastic gradient descent or adaptive variants [12, 11].

The residual-based empirical functional (12) coincides with the formulation of physics-informed neural networks introduced in [1, 2] and subsequently surveyed in [3]. It may be interpreted as a mesh-free least-squares discretization of the boundary value problem carried out in parameter space.

3.3.5 Gradient-based training algorithm

The empirical loss functional $\mathcal{J}_N : \Theta \rightarrow \mathbb{R}$ defined in Definition 3.16 induces a finite-dimensional nonlinear optimization problem in the parameter space $\Theta \subset \mathbb{R}^p$. The construction of a physics-informed neural network is therefore formulated as the numerical minimization of $\mathcal{J}_N(\theta)$ with respect to θ ; see [10, Ch. 3, pp. 48–82] and [11].

For any parameter vector θ , evaluation of $\mathcal{J}_N(\theta)$ consists of three operations: the forward evaluation of the network u_θ at the collocation points, the computation of the required spatial derivatives by automatic differentiation as described in Section 3.3.3, and the assembly of the interior and boundary residuals entering the loss. Since each operation depends smoothly on θ , the gradient $\nabla_\theta \mathcal{J}_N(\theta)$ is computed efficiently by reverse-mode automatic differentiation (back-propagation); see [13, 14]. The computational complexity of evaluating the gradient is of the same order as that of a single forward evaluation of the network.

The training problem

$$\min_{\theta \in \Theta} \mathcal{J}_N(\theta)$$

is generally non-convex and high dimensional. First-order iterative methods are therefore employed due to their favorable computational scaling with respect to the number of parameters. Given a sequence of stepsizes $\{\eta_k\}_{k \geq 0}$, the basic gradient descent iteration is defined by

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k \nabla_\theta \mathcal{J}_N(\theta^{(k)}), \quad k \geq 0. \quad (14)$$

Stochastic or mini-batch variants of this update, as well as adaptive first-order schemes such as Adam [12], are frequently employed in practice.

Definition 3.17 (Training iteration). Let $\mathcal{J}_N$ denote the empirical loss functional. For $\eta > 0$, define the training map

$$T : \Theta \rightarrow \Theta, \quad T(\theta) := \theta - \eta \nabla_\theta \mathcal{J}_N(\theta).$$

A sequence $\{\theta^{(k)}\}_{k \geq 0}$ generated by

$$\theta^{(k+1)} = T(\theta^{(k)})$$

is called a gradient-based training sequence.

The complete training procedure is summarized as follows.

Algorithm 3.1 (Gradient-based PINN training)

1. Select interior collocation points $\{x_i^\Omega\}_{i=1}^{N_\Omega} \subset \Omega$ and boundary points $\{x_j^{\partial\Omega}\}_{j=1}^{N_{\partial\Omega}} \subset \partial\Omega$.
2. Initialize parameters $\theta^{(0)} \in \Theta$.
3. For $k = 0, 1, 2, \dots$ until convergence:
 - (a) Evaluate $u_{\theta^{(k)}}$ at all collocation points;
 - (b) compute spatial derivatives using automatic differentiation;
 - (c) assemble residuals and evaluate $\mathcal{J}_N(\theta^{(k)})$;
 - (d) compute $\nabla_\theta \mathcal{J}_N(\theta^{(k)})$ by backpropagation;
 - (e) update $\theta^{(k+1)}$ according to (14) or an adaptive first-order rule.
4. Return θ^* and the approximation u_{θ^*} .

The above iteration constitutes a residual-minimization procedure carried out in the non-linear trial space $\mathcal{F}_\Theta$. The differential operator is enforced at collocation points in the physical domain, while the approximation is obtained through optimization in parameter space. This formulation coincides with the physics-informed neural network methodology introduced in [1, 2] and surveyed in [3].

3.4 Practical considerations and relation to elliptic problems

The preceding subsections provided a mathematical and algorithmic formulation of physics-informed neural networks. In particular, the solution of an elliptic boundary value problem was expressed as the minimization of a residual-based loss functional over a nonlinear parametric trial space. Although this formulation is well defined at the variational level, the quality of the resulting approximation depends on the interaction between the differential operator, the sampling strategy used to approximate integral functionals, and the properties of the underlying optimization problem. It is therefore necessary to examine how the PINN construction behaves when applied to the elliptic models introduced in Section 2.

The following discussion analyzes several qualitative and numerical aspects that arise directly from the standard formulation, including the interpretation of sampling as quadrature, the effect of stochastic approximation on the residual, and the relation between these features and classical discretization principles.

3.4.1 Sampling as stochastic quadrature

The empirical loss (12) is obtained by evaluating the residual of the differential operator at finitely many collocation points. This procedure admits a precise interpretation in terms of numerical integration. More specifically, the discrete loss can be regarded as a stochastic quadrature approximation of an L^2 -type integral functional.

Recall that the continuous residual functional is given by

$$\mathcal{J}(v) = \int_{\Omega} |\mathcal{L}v(x) - f(x)|^2 dx + \int_{\partial\Omega} |v(x)|^2 ds, \quad v \in \mathcal{F}_{\Theta}. \quad (15)$$

Minimization of $\mathcal{J}$ corresponds to a least-squares formulation of the boundary value problem in $L^2(\Omega)$.

Let $\{x_i^{\Omega}\}_{i=1}^{N_{\Omega}} \subset \Omega$ and $\{x_j^{\partial\Omega}\}_{j=1}^{N_{\partial\Omega}} \subset \partial\Omega$ be collocation points. The discrete loss functional takes the form

$$\mathcal{J}_N(\theta) = \frac{1}{N_{\Omega}} \sum_{i=1}^{N_{\Omega}} |\mathcal{L}u_{\theta}(x_i^{\Omega}) - f(x_i^{\Omega})|^2 + \frac{1}{N_{\partial\Omega}} \sum_{j=1}^{N_{\partial\Omega}} |u_{\theta}(x_j^{\partial\Omega})|^2.$$

Assume that the interior samples are drawn independently according to the uniform distribution on Ω . Let X be a random variable uniformly distributed on Ω . Then

$$\int_{\Omega} |\mathcal{L}v(x) - f(x)|^2 dx = |\Omega| \mathbb{E}[|\mathcal{L}v(X) - f(X)|^2].$$

Proposition 3.18 (Unbiasedness of the empirical residual). *Let $X_1, \dots, X_{N_{\Omega}}$ be independent copies of X . Define*

$$\widehat{\mathcal{J}}_{N_{\Omega}}(v) = \frac{|\Omega|}{N_{\Omega}} \sum_{i=1}^{N_{\Omega}} |\mathcal{L}v(X_i) - f(X_i)|^2.$$

Then

$$\mathbb{E}[\widehat{\mathcal{J}}_{N_{\Omega}}(v)] = \int_{\Omega} |\mathcal{L}v - f|^2 dx.$$

Hence the empirical loss provides an unbiased estimator of the continuous residual functional.

Proposition 3.19 (Monte Carlo variance estimate). *If $|\mathcal{L}v - f|^2 \in L^2(\Omega)$, then*

$$\text{Var}(\widehat{\mathcal{J}}_{N_{\Omega}}(v)) = \mathcal{O}(N_{\Omega}^{-1}),$$

and the root-mean-square quadrature error decays at rate

$$\mathcal{O}(N_{\Omega}^{-1/2}).$$

The convergence rate is independent of the spatial dimension, which explains the use of Monte Carlo sampling in high-dimensional settings.

This interpretation clarifies the relation between PINNs and classical discretization techniques. In finite element or finite difference methods, the residual is controlled through deterministic quadrature rules on a mesh and the approximation error is governed by the mesh size h ; see [6, Ch. 4, pp. 115–124]. In contrast, the PINN loss depends on statistical sampling of the domain. Consequently, the accuracy of the residual approximation is determined by the number and distribution of samples rather than by geometric resolution.

If very large values of the residual are strongly localized, for example in boundary or interior layers, uniform sampling may provide insufficient coverage of these regions. In such cases, the empirical loss may underestimate the true L^2 residual, and minimizers of $\mathcal{J}_N$ may not approximate minimizers of $\mathcal{J}$ accurately. This effect is particularly pronounced for convection-dominated elliptic problems, where sharp layers occupy only a small portion of the domain. This observation should not be interpreted as a general recommendation to concentrate collocation points in the layers. Numerical studies for convection-dominated convection–diffusion problems show that layer-adapted point distributions do not necessarily improve PINN accuracy; for some boundary-layer tests, equispaced or uniformly random points performed better [20, pp. 335–344].

3.4.2 Optimization landscape and non-convexity

In addition to the sampling error introduced by stochastic quadrature, the training of a physics-informed neural network involves an additional source of inaccuracy arising from non-linear optimization in parameter space. Although the continuous residual functional is quadratic with respect to the function variable, the empirical loss $\mathcal{J}_N(\theta)$ is generally non-convex with respect to the network parameters. Consequently, the quality of the computed approximation depends not only on the approximation properties of the hypothesis space but also on the behavior of the optimization algorithm.

Consider the continuous least-squares functional

$$\mathcal{J}(v) = \|\mathcal{L}v - f\|_{L^2(\Omega)}^2 + \|v - g\|_{L^2(\partial\Omega)}^2.$$

When this functional is minimized over a linear finite-dimensional space $V_h \subset H_0^1(\Omega)$, for example a finite element space, the discrete problem reduces to the minimization of a quadratic form in the expansion coefficients. Equivalently, one obtains a linear system

$$Au_h = b,$$

where the stiffness matrix A is symmetric positive definite under standard assumptions. In the symmetric positive-definite case, the corresponding energy is a strictly convex quadratic functional and possesses a unique global minimizer; see [6, Ch. 2, pp. 58–65]. For nonsymmetric convection–diffusion problems one instead relies on well-posedness of the associated linear system rather than on a symmetric quadratic minimization principle. In this setting, the numerical error is determined solely by approximation and quadrature effects.

The situation differs when the minimization is performed over the nonlinear manifold

$$\mathcal{F}_\Theta = \{u_\theta : \theta \in \Theta\}.$$

Proposition 3.20 (Non-convexity of the parameterized loss). *Let u_θ be represented by a multilayer neural network with nonlinear activation functions. Then the mapping*

$$\theta \mapsto \mathcal{J}(u_\theta)$$

is generally non-convex. Consequently, the empirical loss $\mathcal{J}_N(\theta)$ may admit multiple local minima and saddle points.

This non-convexity is caused by the repeated composition of affine maps with nonlinear activations. The functional $\mathcal{J}(u_\theta)$ is therefore not a quadratic form in θ , even when $\mathcal{J}$ is quadratic in the function variable.

The training problem

$$\min_{\theta \in \Theta} \mathcal{J}_N(\theta)$$

is therefore a high-dimensional non-convex optimization problem. First-order methods such as stochastic gradient descent or adaptive variants generate sequences $\{\theta^{(k)}\}$ satisfying approximate descent conditions, but convergence to a global minimizer cannot be guaranteed in general. Instead, one typically obtains convergence to stationary points characterized by

$$\nabla_\theta \mathcal{J}_N(\theta^*) = 0;$$

see [10, Ch. 3, pp. 48–82], [11], and [8, Ch. 8, pp. 267–324]. In practice, optimization algorithms may stagnate at a positive loss value before even this ideal stationarity condition is accurately satisfied.

Remark 3.21 (Error decomposition). The overall approximation error of the PINN method may be interpreted as the combination of three contributions: the approximation error induced by the restriction to the hypothesis class $\mathcal{F}_\Theta$, the sampling error arising from the stochastic quadrature of the residual functional, and the optimization error resulting from incomplete minimization of $\mathcal{J}_N$. Only the first two components appear in classical discretization methods, whereas the third is intrinsic to parameterized learning approaches.

The conditioning of the optimization problem is further influenced by the structure of the differential operator. Residuals containing higher-order derivatives may lead to gradients with widely varying magnitudes, which can deteriorate the stability and convergence speed of gradient-based updates. Such sensitivity is particularly pronounced for stiff or convection-dominated problems, where small perturbations of the parameters may produce large variations in higher-order derivatives.

The PINN methodology thus replaces the solution of a convex linear system by the minimization of a nonlinear objective over parameter space. This formulation provides flexibility and mesh independence, but introduces additional computational complexity associated with non-convex optimization.

3.4.3 Regularity and approximation capability

A further source of error intrinsic to the PINN methodology arises from the restriction of the approximation to the parametric function class

$$\mathcal{F}_\Theta = \{u_\theta : \theta \in \Theta\}.$$

Even if the continuous residual functional is evaluated exactly and the global minimizer of the loss is attained, the computed solution belongs to $\mathcal{F}_\Theta$ only. The attainable accuracy is therefore determined by the best-approximation properties of this hypothesis space.

The approximation properties must be considered together with the regularity theory of elliptic boundary value problems. For diffusion-dominated equations on sufficiently smooth domains, elliptic regularity implies that weak solutions possess additional Sobolev smoothness. For instance, for the Poisson equation with $f \in L^2(\Omega)$ and smooth boundary, one has $u \in H^2(\Omega)$ together with the estimate

$$\|u\|_{H^2(\Omega)} \leq C \|f\|_{L^2(\Omega)};$$

see [4, Ch. 6, pp. 303–351] and [5, Ch. 6, pp. 107–112]. Hence, the exact solution is globally smooth.

The general approximation results from Section 3.2 show that neural networks can be rich trial spaces, but these results do not by themselves determine the accuracy of a trained PINN. What matters for elliptic problems is the interaction between the regularity of the exact solution, the architecture, and the norm in which the residual is minimized.

For smooth diffusion-dominated problems, increasing the depth and width of the network can reduce the best-approximation error, analogously to increasing the polynomial degree or reducing the mesh size in finite element methods; see [6, Ch. 4, pp. 115–124]. For problems with reduced regularity, however, the best-approximation error may dominate. Convection–diffusion–reaction equations may exhibit boundary or interior layers whose characteristic width scales with the diffusion coefficient. Such localized structures require high resolution in small subregions of the domain and are therefore difficult to capture with global parametric representations.

The attainable accuracy of physics-informed neural networks is consequently governed jointly by the regularity of the exact solution, the expressive capacity of the chosen architecture, and the success of the training procedure. This is the point at which the analytical results from Section 2 become relevant for the practical performance of PINNs.

3.4.4 Convection-dominated regimes

The performance of residual-based neural network approximations depends on the interaction between sampling, approximation properties of the hypothesis space, and the conditioning of the associated optimization problem. These effects become particularly significant for convection–diffusion–reaction equations in parameter regimes where convection dominates diffusion.

Consider the boundary value problem

$$-\nabla \cdot (\kappa \nabla u) + \beta \cdot \nabla u + cu = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega,$$

with diffusion coefficient $\kappa > 0$ and convection field β .

Definition 3.22 (Péclet number). Let L denote a characteristic length scale of the domain. The dimensionless quantity

$$\text{Pe} := \frac{\|\beta\|_{L^\infty(\Omega)} L}{\kappa}$$

is called the Péclet number. The regime $\text{Pe} \gg 1$ is referred to as convection-dominated; see [7, Ch. 1, pp. 1–34].

Uniform ellipticity is preserved for $\kappa > 0$, but the strength of the elliptic regularization decreases as $\kappa \rightarrow 0$. Classical singular perturbation theory shows that solutions may develop boundary or interior layers whose characteristic thickness is often proportional to the diffusion parameter,

$$\text{layer width} \sim \mathcal{O}(\kappa),$$

or, depending on the balance between reaction, convection, and diffusion, of order $\mathcal{O}(\sqrt{\kappa})$. The latter case is especially important in applications and can be more challenging numerically. Away from these narrow regions, solutions may remain smooth, but their derivatives can scale with inverse powers of κ ; see [7, Ch. 1, pp. 1–34]. Hence, the solution exhibits steep gradients concentrated on sets of small measure.

Remark 3.23 (Influence on sampling). When the empirical loss is interpreted as a Monte Carlo quadrature of an L^2 residual, the accuracy of the estimator depends on the distribution of collocation points. If localized layers occupy only a small portion of the domain, uniformly sampled points may fail to resolve these regions unless the number of samples is sufficiently large. The empirical residual may therefore underestimate the true residual contribution within the layers. However, this does not imply that concentrating collocation points in layers always improves

PINN approximations. Numerical experiments indicate that uniform point distributions together with global parameter representations can be more robust for some convection-dominated problems [20, pp. 335–344].

Remark 3.24 (Influence on approximation). The presence of sharp gradients increases the intrinsic complexity of the target function. Although neural networks admit universal approximation properties, functions with rapid local variation typically require a larger number of parameters to achieve a prescribed accuracy. This behavior is analogous to the need for local mesh refinement or layer-adapted meshes in classical finite element methods for singularly perturbed problems; see [7, Ch. 1, pp. 1–34].

Remark 3.25 (Influence on optimization). The differential operator involves second derivatives scaled by the small parameter κ . Consequently, small perturbations of the network parameters may induce large variations of the residual inside the layers, which can lead to ill-conditioning of the loss landscape and slow convergence of gradient-based optimization methods.

These effects originate from structural properties of the underlying differential equation rather than from a particular implementation of the learning algorithm. Convection-dominated regimes therefore constitute intrinsically more demanding test cases for residual-based neural network approximations than purely diffusion-driven elliptic problems. Detailed numerical studies of this convergence behavior are the topic of the computational part of this thesis.

3.4.5 Compatibility with maximum principles

In addition to stability estimates expressed in integral norms, second-order elliptic boundary value problems possess qualitative properties that constrain the pointwise behavior of solutions. Among the most important of these properties are maximum principles, which provide bounds that depend on the boundary data and the sign of the source term; see [5, Ch. 3, pp. 31–35] and [4, Sec. 6.4, pp. 327–334].

Theorem 3.26 (Maximum principle, abstract form). *Let $\mathcal{L}$ be a uniformly elliptic operator satisfying appropriate sign conditions and let u solve*

$$\mathcal{L}u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega.$$

If $f \leq 0$ almost everywhere in Ω , then

$$\sup_{x \in \Omega} u(x) \leq 0,$$

and an analogous statement holds for $f \geq 0$.

Such results exclude interior extrema that contradict the boundary data and, in particular, imply preservation of positivity or negativity. They provide pointwise control that cannot be deduced solely from coercivity or L^2 -based stability estimates.

The residual-based formulation underlying physics-informed neural networks is defined through an objective functional of the form

$$\|\mathcal{L}u_\theta - f\|_{L^2(\Omega)}^2 + \|u_\theta\|_{L^2(\partial\Omega)}^2,$$

which measures violations of the governing equation only in an averaged least-squares sense. Control of the residual in $L^2(\Omega)$ does not imply control of the supremum norm. Functions may possess arbitrarily small L^2 norm while exhibiting large localized deviations on sets of small measure.

Remark 3.27 (Norm mismatch). Maximum principles concern pointwise or L^∞ behavior, whereas the PINN loss functional controls only integral quantities. Consequently, minimization of the residual functional does not, in general, guarantee preservation of bounds such as positivity, monotonicity, or absence of spurious extrema.

Remark 3.28 (Relation to least-squares discretizations). The absence of automatic preservation of qualitative properties is not specific to neural networks but is characteristic of least-squares and residual minimization methods. Classical finite element discretizations may likewise violate discrete maximum principles unless additional geometric or monotonicity conditions are imposed; see the recent comprehensive treatment of monotone discretizations and discrete maximum principles in [21, Chs. 2–3, pp. 7–102].

Remark 3.29 (Diagnostic role). Since the continuous problem satisfies maximum principles, violations of these bounds by an approximation may indicate insufficient sampling, inadequate representational capacity, or failure of the optimization procedure. Pointwise constraints therefore provide complementary diagnostic information beyond integral residual measures.

Maximum principles thus constitute qualitative benchmarks for assessing the reliability of residual-based neural network approximations of elliptic problems.

A related research direction is to modify the PINN architecture or loss so that physically admissible bounds are preserved. For steady-state convection–diffusion–reaction problems, Matthaiou, John, and Zainelabdeen propose several bound-preserving PINN and *hp*-VPINN strategies, including loss penalization, cut-off postprocessing, and output-layer constructions that enforce bounds directly [19, pp. 167–183].

From an analytical standpoint, this observation reflects the fact that residual minimization controls the error primarily in integral norms, whereas maximum principles concern pointwise behavior. Accordingly, small residuals in $L^2(\Omega)$ do not in general imply preservation of qualitative bounds such as positivity or monotonicity. The absence of an inherent maximum principle is therefore a structural feature of least-squares-type formulations rather than a consequence of a particular network architecture.

Consequently, when physics-informed neural networks are applied to elliptic boundary value problems, the reliability of the approximation must be assessed not only through residual norms but also through qualitative criteria consistent with the underlying analytical theory. In this sense, the maximum principle provides an additional benchmark that complements the variational and optimization-based framework developed above and clarifies the limitations and expected behavior of residual-based neural network approximations for the elliptic models considered here.

4 Numerical Experiments

This section reports the numerical experiments carried out for the Poisson and convection–diffusion–reaction problems. The purpose is not only to record final errors, but also to compare the influence of the activation function and of the hard-constrained boundary indicator used in the PINN ansatz. Unless stated otherwise, the architecture is fixed, the training budget is 10000 epochs, and the number of interior collocation points is varied through uniform grids

$$M = 10, 20, 40, 50, 60, 80, 100, \quad N = M^2, \quad h = 1/M.$$

The error values reported below are the final recorded values after training. The L^2 error, H^1 seminorm error, and L^∞ error are evaluated against the known exact solution or reference solution on the evaluation grid. The figures are the original error plots generated by the experiment scripts, inserted without changing their aspect ratios.

For all experiments, the same basic fully connected feedforward architecture is used unless explicitly stated otherwise. The network has eight hidden layers with 20 neurons per hidden layer, and it is trained with the Adam optimizer, learning rate 10^{-3} , batch size 64, and 512 boundary points. The homogeneous Dirichlet boundary conditions are imposed by a hard-constrained ansatz. The numerical assessment is based mainly on the error norms, and the observed solution range, undershoots, and overshoots are used as additional diagnostics for pointwise behaviour.

4.1 Experimental Design

The experiments are arranged so that the numerical behaviour of the PINN can be studied first in a simple setting and then under gradually more demanding conditions. The starting point is the Poisson equation, which serves as a diffusion-dominated reference problem with a smooth exact solution. This baseline is useful because the PDE contains no convection or reaction term, and the effect of the collocation grid can therefore be observed without additional transport-dominated features. The same training configuration is then applied to a smooth convection–diffusion–reaction problem. This second model keeps the solution smooth, but introduces first-order convection and zeroth-order reaction terms in the residual. The third problem adds the main difficulty of the later experiments: an outflow-layer solution with sharp local variation near the boundary.

These first three tests form the reference sequence and establish how the same fixed PINN architecture behaves as the PDE changes from Poisson, to the CDR problem with smooth solution, and finally to the layer CDR problem. The later experiments are introduced as controlled modifications of this sequence. In each case, the PDE, collocation grids, training budget, and most network settings are kept unchanged, while one modelling choice is altered. This structure makes it possible to distinguish the effects of the activation function and the boundary indicator from the effect of changing the differential equation.

Two follow-up tests are made for the Poisson problem. In the first one, the polynomial hard-constraint indicator $x(1-x)y(1-y)$ is retained, but the activation function is changed from tanh to GELU. This comparison with the Poisson baseline isolates the influence of the activation function. In the second one, the activation function remains tanh, but the boundary indicator is replaced by the exponential function

$$(1 - e^{-50x})(1 - e^{-50y})(1 - e^{-50(1-x)})(1 - e^{-50(1-y)}).$$

This comparison is motivated by the question whether a different hard-constrained ansatz can improve the enforcement of the boundary behaviour or the optimization process.

The activation-function comparison is then repeated for the two CDR models. The CDR experiment with smooth solution and GELU is compared with the corresponding tanh experiment, while the layer CDR experiment with GELU is compared with the corresponding layer

CDR experiment using tanh. These comparisons make it possible to see whether the effect observed for the Poisson equation persists when convection, reaction, and layer behaviour are present. Overall, the experimental design separates three issues: the influence of collocation-grid refinement, the role of the activation function, and the effect of the indicator used to impose hard-constrained Dirichlet boundary conditions.

4.2 Exact and Reference Solutions

All reported error norms are computed by comparing the trained PINN approximation with an exact or reference solution. For the Poisson experiments, the exact solution is

$$u(x, y) = \sin(2\pi x) \sin(2\pi y).$$

For the CDR experiments with smooth solution, the exact solution is

$$u(x, y) = 16 \sin(2\pi x) y^2 (1 - y)^2.$$

For the outflow-layer CDR experiments, the exact solution is

$$u(x, y) = xy^2 - y^2 \exp\left(\frac{2(x-1)}{\varepsilon}\right) - x \exp\left(\frac{3(y-1)}{\varepsilon}\right) + \exp\left(\frac{2(x-1)}{\varepsilon}\right) \exp\left(\frac{3(y-1)}{\varepsilon}\right), \quad \varepsilon = 10^{-8}.$$

The right-hand side in each problem is obtained by applying the corresponding differential operator to the chosen exact solution.

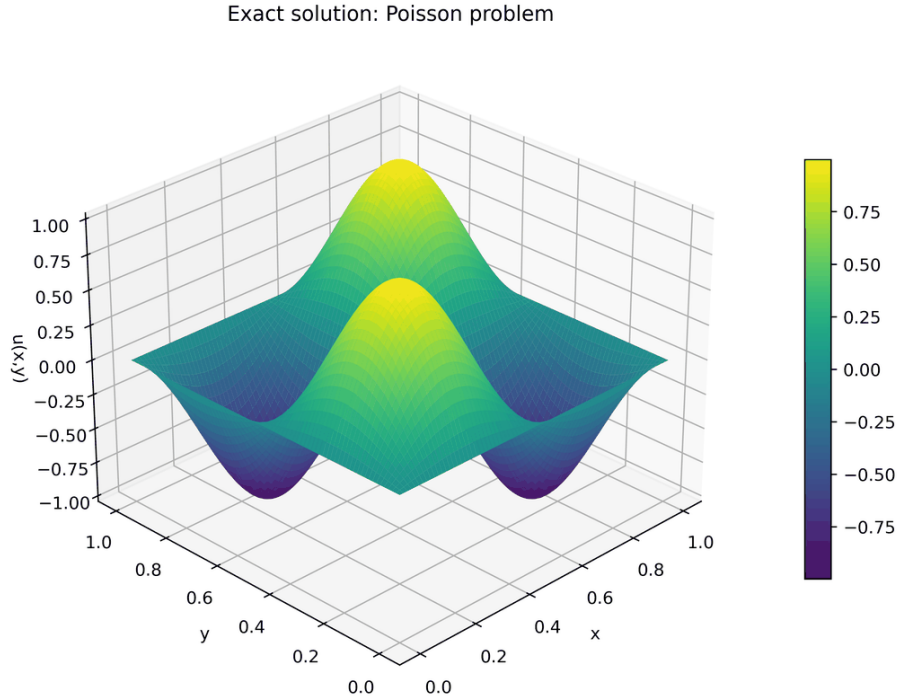


Figure 1: Exact solution used in the Poisson experiments.

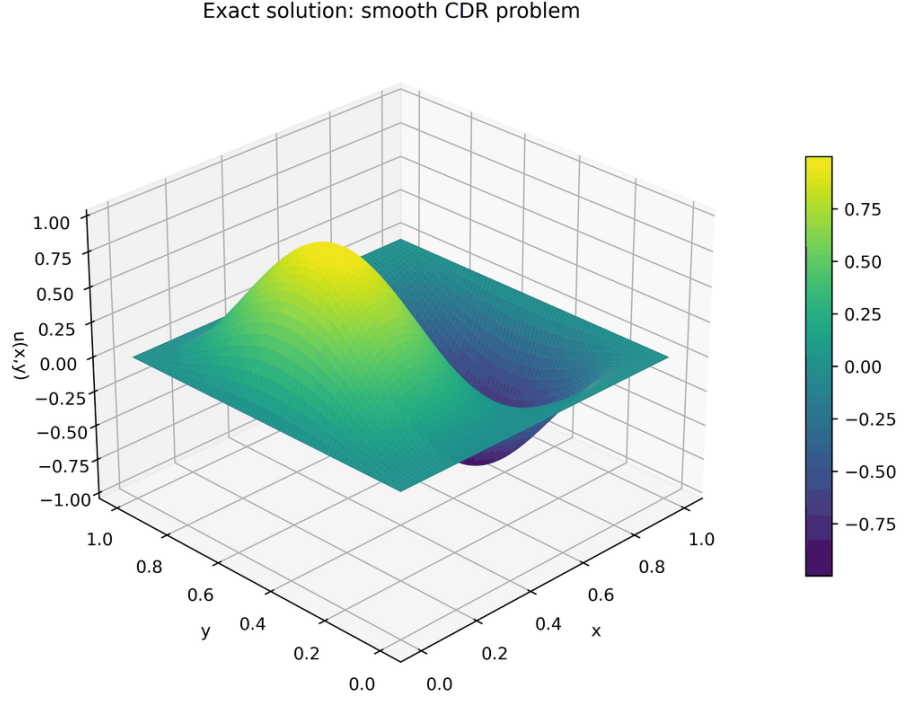


Figure 2: Exact solution used in the CDR experiments with smooth solution.

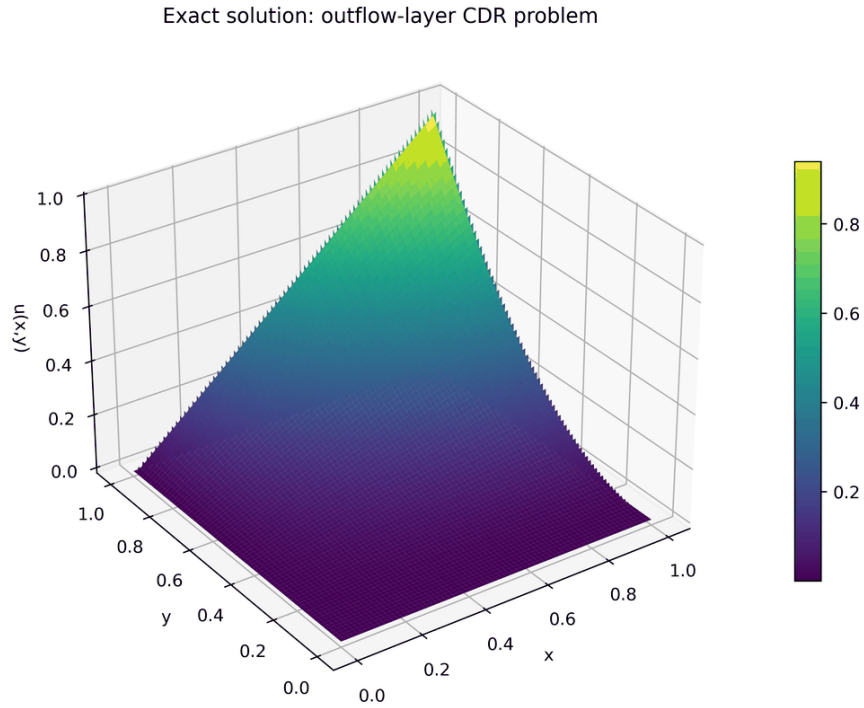


Figure 3: Exact solution used in the outflow-layer CDR experiments.

4.3 Poisson Problem: Standard Polynomial Indicator

The first experiment is the baseline Poisson test. The hard-constrained ansatz uses the polynomial indicator $x(1-x)y(1-y)$, so that the Dirichlet boundary condition is satisfied

exactly by construction. The errors are measured against the exact solution shown in Figure 1. Table 1 shows that the error decreases notably from coarse to finer collocation grids, although the improvement is not perfectly monotone.

Table 1: Experiment 1: Poisson problem with tanh activation and polynomial hard-constraint indicator.

M	N	loss	L^2	H^1 semi	L^∞	range of u_N
10	100	7.455e+02	7.229e-03	1.112e-01	1.604e-02	[-1.0117, 1.0060]
20	400	8.685e+03	8.735e-03	8.791e-02	2.087e-02	[-1.0056, 1.0019]
40	1600	5.440e+02	1.449e-03	1.233e-02	3.151e-03	[-0.9997, 1.0013]
50	2500	1.104e+03	2.982e-03	2.976e-02	7.153e-03	[-1.0019, 1.0005]
60	3600	2.164e+02	3.185e-03	2.442e-02	9.663e-03	[-0.9994, 1.0053]
80	6400	5.054e+02	8.974e-04	9.697e-03	2.911e-03	[-0.9994, 1.0003]
100	10000	1.519e+02	1.447e-03	1.108e-02	3.218e-03	[-1.0002, 1.0029]

The numerical solution remains close to the expected range $[-1, 1]$ for all tested grids. The small deviations from this interval indicate mild pointwise oscillations, but no severe boundary violation is observed with the polynomial hard constraint.

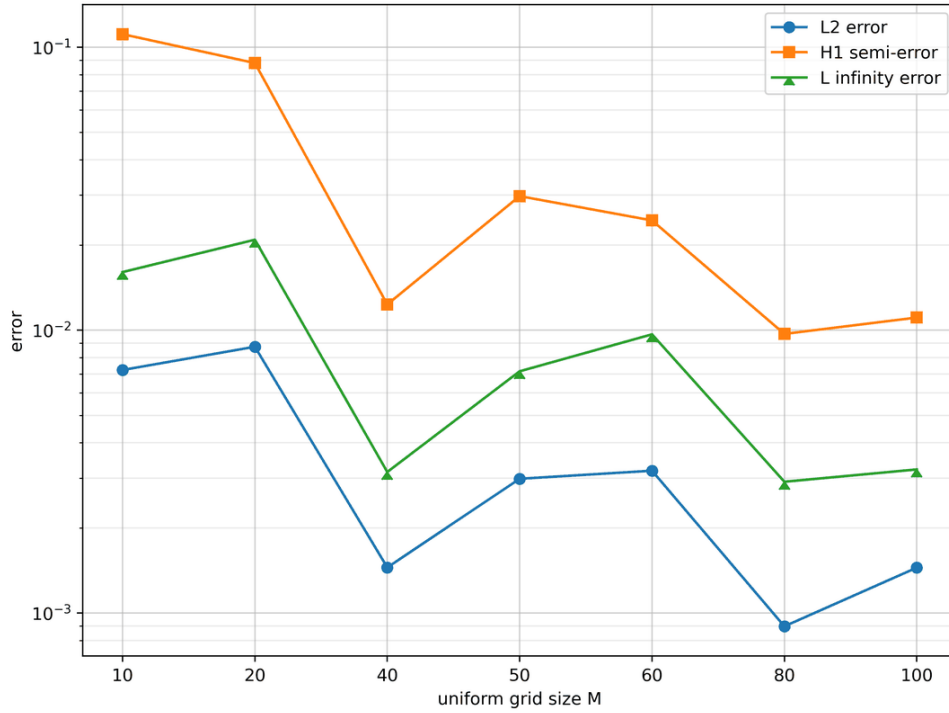


Figure 4: Experiment 1: final error norms for the Poisson problem.

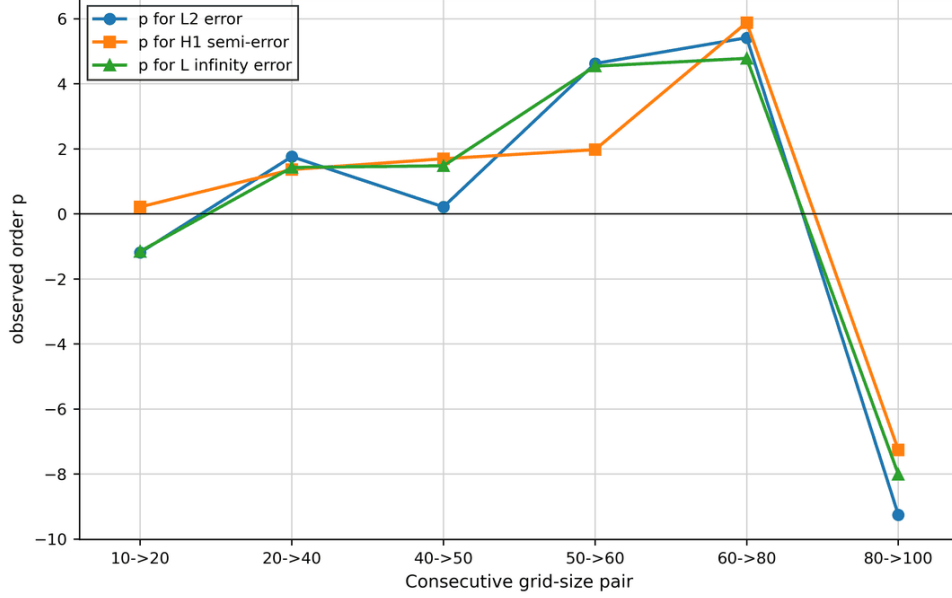


Figure 5: Experiment 1: observed convergence orders for the Poisson problem.

4.4 CDR Problem with Smooth Solution: Standard Polynomial Indicator

The second experiment applies the same basic network architecture and polynomial indicator to the convection–diffusion–reaction problem with the smooth solution and $\varepsilon = 10^{-8}$. The errors are measured against the exact solution in Figure 2. Compared with the corresponding Poisson problem, the residual contains convection and reaction terms, and the final error is more sensitive to the training process. The smallest errors are observed for $M = 80$, but the error does not decrease monotonically as M increases.

Table 2: Experiment 2: CDR problem with smooth solution, tanh activation, and polynomial indicator.

M	N	loss	L^2	H^1 semi	L^∞	range of u_N
10	100	2.485e+00	1.610e-02	2.897e-01	4.826e-02	[-1.0180, 1.0223]
20	400	3.691e-01	8.651e-04	2.797e-02	2.906e-03	[-1.0012, 1.0009]
40	1600	5.985e-01	9.612e-04	1.373e-02	3.034e-03	[-1.0004, 0.9982]
50	2500	2.652e+00	2.178e-03	3.080e-02	6.345e-03	[-0.9975, 0.9969]
60	3600	4.179e-01	1.484e-02	8.669e-02	4.026e-02	[-1.0003, 1.0256]
80	6400	8.556e-01	2.142e-04	8.078e-03	1.000e-03	[-1.0002, 0.9998]
100	10000	5.493e-01	2.268e-03	2.399e-02	6.541e-03	[-1.0000, 0.9938]

The solution range is again close to the expected interval $[-1, 1]$, although the case $M = 60$ shows a visible overshoot. This supports the observation that the CDR residual is more sensitive to training than the Poisson residual, even when the exact solution is smooth.

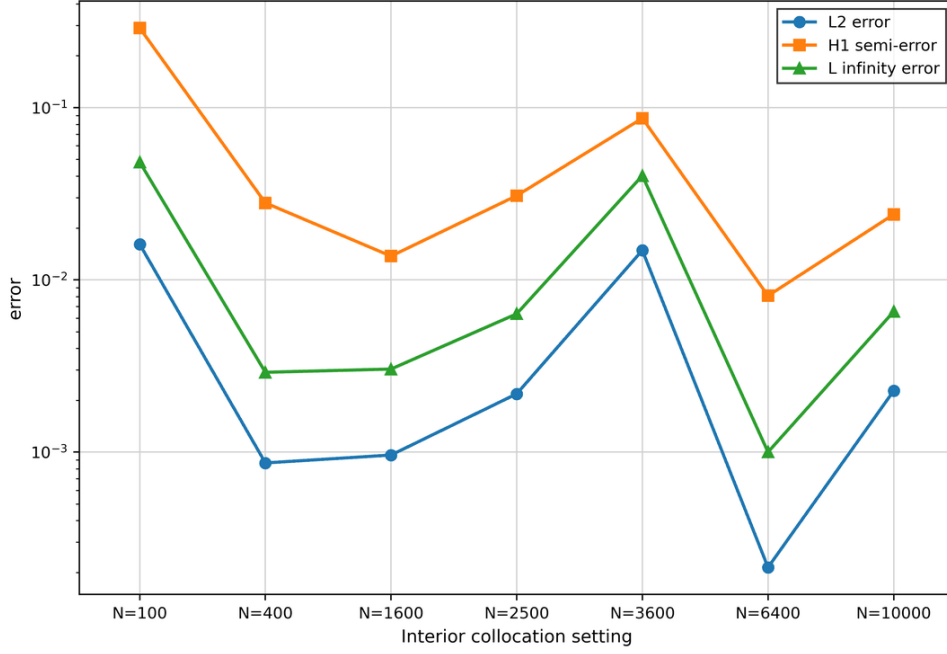


Figure 6: Experiment 2: final error norms for the CDR problem with smooth solution.

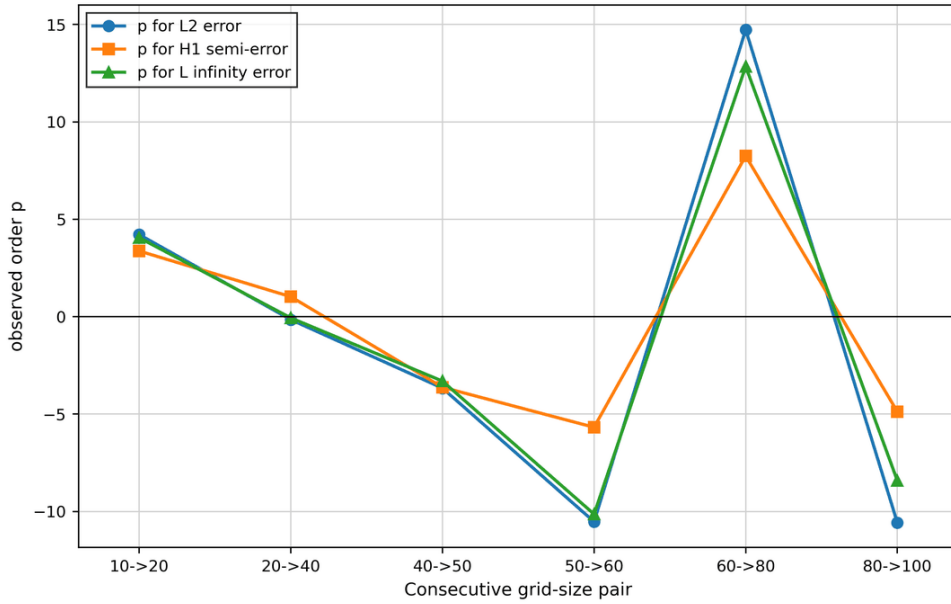


Figure 7: Experiment 2: observed convergence orders for the CDR problem with smooth solution.

4.5 Layer CDR Problem: Standard Indicator

The third experiment considers a convection–diffusion–reaction benchmark with outflow boundary layers. The errors are measured against the exact solution in Figure 3. This problem is more demanding because the exact solution has sharp variations near the outflow boundary. The final errors show that the PINN can capture the layer reasonably well for some grids, especially $M = 60$ and $M = 80$, but the non-monotone behavior remains visible.

Table 3: Experiment 3: layer CDR problem with the standard layer indicator.

M	N	loss	L^2	H^1 semi	L^∞	undershoot	overshoot
10	100	5.902e+01	1.354e-02	1.717e-01	3.373e-02	2.813e-02	1.036e-03
20	400	2.020e-01	3.491e-03	7.150e-02	1.757e-02	3.094e-03	0.000e+00
40	1600	1.074e+00	4.754e-03	2.758e-02	1.236e-02	1.758e-03	4.950e-03
50	2500	3.710e-01	5.424e-03	3.917e-02	6.601e-03	4.871e-03	2.081e-03
60	3600	2.825e-01	1.167e-04	2.953e-03	8.124e-04	1.986e-04	0.000e+00
80	6400	5.625e-01	3.128e-04	1.716e-03	1.608e-03	3.230e-05	3.047e-04
100	10000	4.089e-01	1.923e-03	7.603e-03	4.753e-03	8.117e-04	0.000e+00

For this layer problem, undershoots and overshoots are particularly relevant because the exact solution is expected to remain in the interval $[0,1]$. The values in Table 3 show that the hard-constrained ansatz keeps these violations small for the finer grids, but coarse and intermediate grids can still produce localized pointwise deviations.

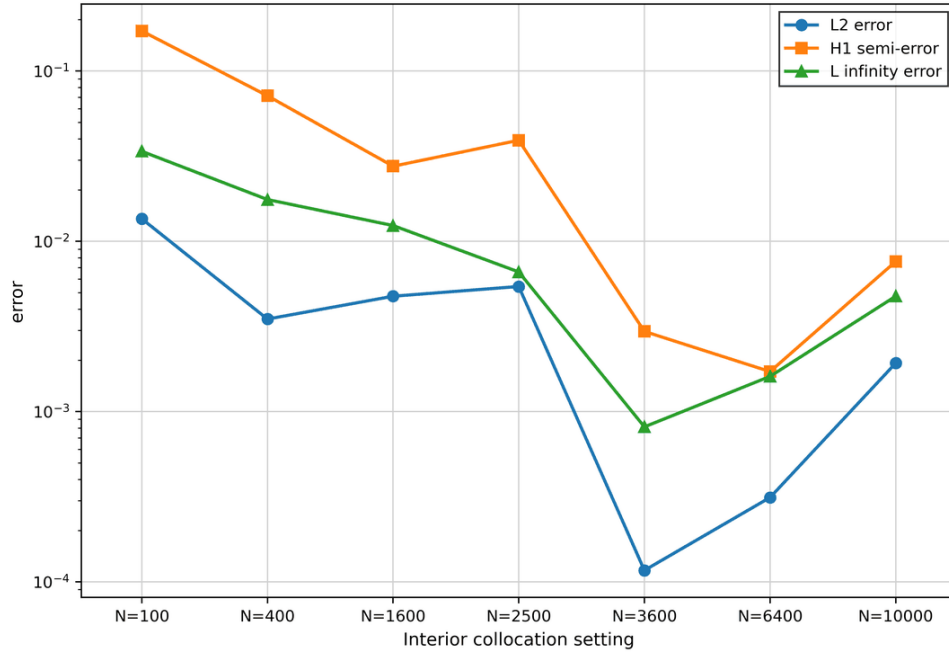


Figure 8: Experiment 3: final error norms for the layer CDR problem.

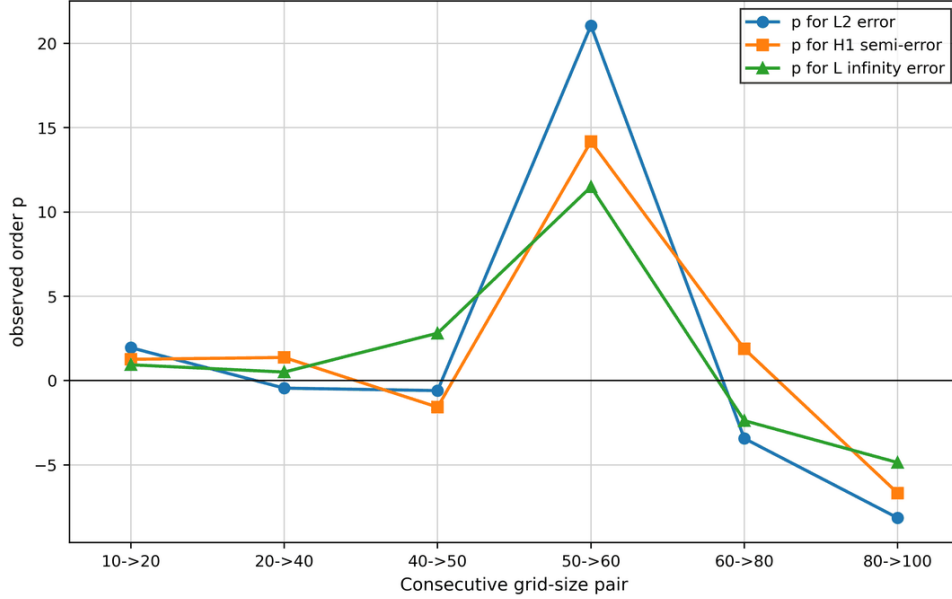


Figure 9: Experiment 3: observed convergence orders for the layer CDR problem.

4.6 Poisson Problem: GELU Activation

Experiment 4 is the first controlled comparison with the Poisson baseline. It repeats Experiment 1 with the same PDE, exact solution, collocation grids, network size, optimizer settings, and polynomial indicator, but replaces the hidden-layer activation function by GELU. Therefore, differences between Tables 1 and 4 mainly reflect the effect of the activation function. The results show that GELU can improve the best achieved accuracy, most clearly for $M = 100$, where all three error norms are smaller than in the corresponding tanh run. At the same time, the residual losses remain irregular across the grid sequence. Thus, the activation function influences the approximation quality, but it does not by itself restore monotone convergence with respect to the number of collocation points.

Table 4: Experiment 4: Poisson problem with GELU activation and polynomial indicator.

M	N	loss	L^2	H^1 semi	L^∞	range of u_N
10	100	4.887e+01	2.830e-03	3.366e-02	1.127e-02	[-1.0016, 1.0032]
20	400	8.177e+02	7.987e-03	5.910e-02	2.170e-02	[-1.0145, 1.0001]
40	1600	4.792e+01	3.171e-03	2.237e-02	9.806e-03	[-1.0089, 1.0008]
50	2500	2.280e+02	4.883e-03	4.049e-02	1.124e-02	[-0.9956, 0.9939]
60	3600	2.015e+02	7.020e-04	5.436e-03	1.739e-03	[-1.0014, 1.0004]
80	6400	6.635e+00	8.563e-04	5.973e-03	2.656e-03	[-1.0004, 0.9996]
100	10000	6.239e+01	3.409e-04	2.574e-03	7.793e-04	[-1.0003, 1.0004]

The computed range remains close to $[-1, 1]$ throughout the grid sequence. Compared with Experiment 1, the GELU activation does not introduce stronger spurious oscillations and improves the pointwise accuracy for the finest grid.

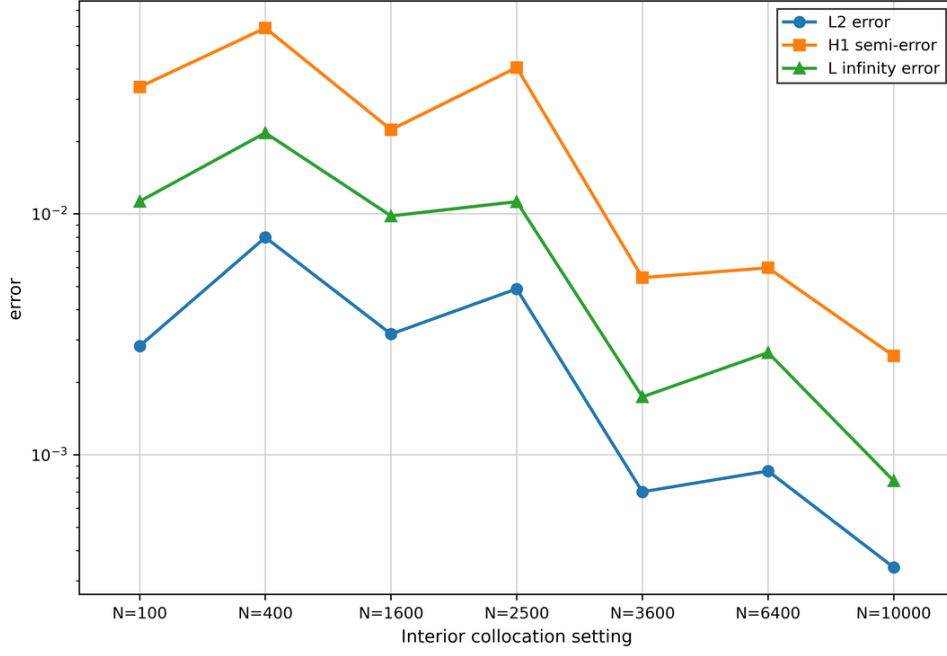


Figure 10: Experiment 4: final error norms for the Poisson problem with GELU activation.

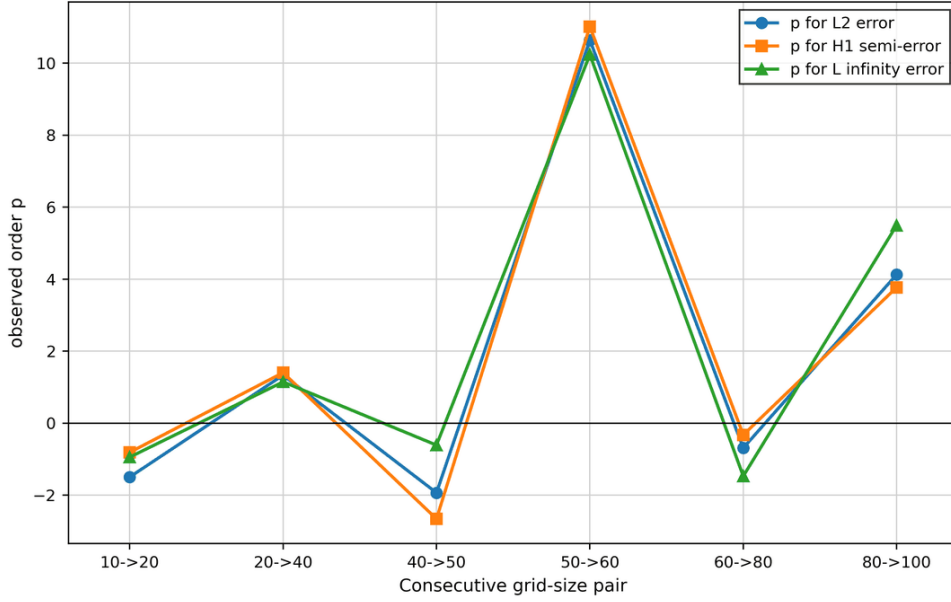


Figure 11: Experiment 4: observed convergence orders for the Poisson problem with GELU activation.

4.7 Poisson Problem: Exponential Indicator

Experiment 5 is the second controlled comparison with the Poisson baseline. In contrast to Experiment 4, the activation function remains tanh, and only the hard-constrained boundary indicator is changed. The exact Poisson solution is again the one shown in Figure 1. This comparison is important because the indicator is not only a boundary-enforcing factor; it also changes the shape and scaling of the trainable ansatz inside the domain. The results indicate that the exponential indicator with parameter 50 is less robust than the polynomial indicator

for this Poisson test. In particular, the coarse-grid errors and residual losses are much larger than in Experiment 1, and the improvement on finer grids does not compensate for this loss of stability.

Table 5: Experiment 5: Poisson problem with tanh activation and exponential boundary indicator.

M	N	loss	L^2	H^1 semi	L^∞	range of u_N
10	100	1.348e+04	2.596e-01	4.285e+00	8.286e-01	[-1.3917, 1.1643]
20	400	7.972e+04	4.569e-02	8.482e-01	1.113e-01	[-1.0284, 0.9902]
40	1600	1.598e+04	1.247e-02	1.794e-01	2.988e-02	[-1.0056, 1.0183]
50	2500	1.227e+05	1.840e-02	1.088e-01	3.848e-02	[-0.9848, 1.0020]
60	3600	1.489e+03	8.652e-03	6.443e-02	1.793e-02	[-0.9946, 1.0091]
80	6400	3.352e+04	5.120e-03	6.388e-02	1.436e-02	[-1.0019, 0.9983]
100	10000	1.290e+04	2.337e-03	2.609e-02	6.291e-03	[-0.9964, 1.0048]

The solution range contains much stronger undershoots and overshoots on the coarse grids than in Experiment 1. This suggests that the exponential indicator changes the optimization problem in a way that can amplify pointwise oscillations, even though the boundary values are still imposed by a hard constraint.

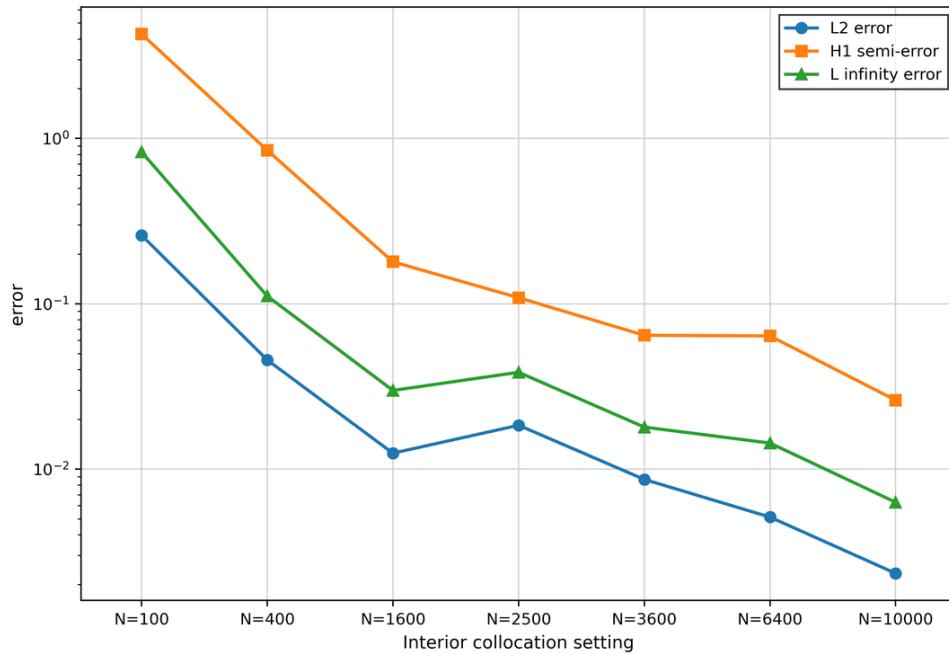


Figure 12: Experiment 5: final error norms for the Poisson problem with the exponential indicator.

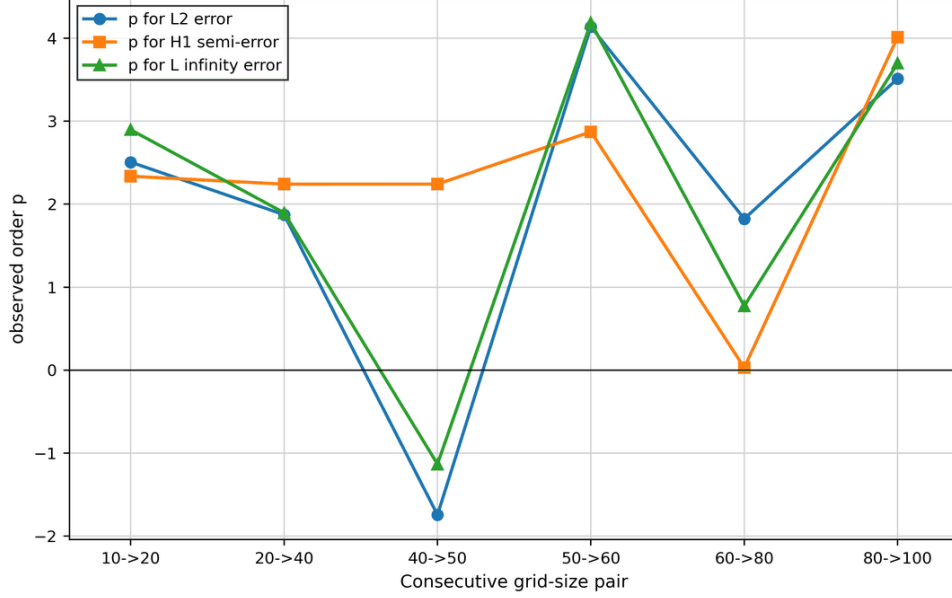


Figure 13: Experiment 5: observed convergence orders for the Poisson problem with the exponential indicator.

4.8 CDR Problem with Smooth Solution: GELU Activation

Experiment 6 transfers the activation-function comparison from the Poisson problem to the CDR problem with smooth solution. It is therefore compared directly with Experiment 2: the PDE, exact solution, polynomial indicator, and collocation grids are unchanged, while the activation function is changed from tanh to GELU. This comparison tests whether the advantage of GELU in the Poisson problem persists once convection and reaction terms enter the residual. The results improve on several finer grids, and the smallest error is obtained at $M = 50$. However, the error still does not decrease monotonically as M increases. The comparison therefore suggests that GELU can improve some CDR approximations, but the convection–reaction residual remains sensitive to the training process.

Table 6: Experiment 6: CDR problem with smooth solution, GELU activation, and polynomial indicator.

M	N	loss	L^2	H^1 semi	L^∞	range of u_N
10	100	5.640e+00	8.157e-03	1.732e-01	3.148e-02	[-1.0032, 0.9902]
20	400	1.487e+00	5.438e-03	4.614e-02	1.556e-02	[-0.9863, 0.9923]
40	1600	8.100e-02	5.905e-04	7.788e-03	1.831e-03	[-0.9996, 0.9983]
50	2500	3.027e-01	3.618e-04	6.057e-03	9.802e-04	[-1.0003, 1.0006]
60	3600	1.968e-01	2.887e-03	2.239e-02	6.372e-03	[-1.0013, 0.9944]
80	6400	1.741e-01	2.288e-03	1.783e-02	5.147e-03	[-0.9987, 1.0047]
100	10000	2.464e-01	1.184e-03	9.041e-03	3.021e-03	[-0.9983, 1.0029]

The range of the numerical solution is mostly close to $[-1, 1]$, with only moderate deviations. Compared with Experiment 2, the GELU activation reduces some of the larger pointwise errors, but small oscillations remain visible in selected grid cases.

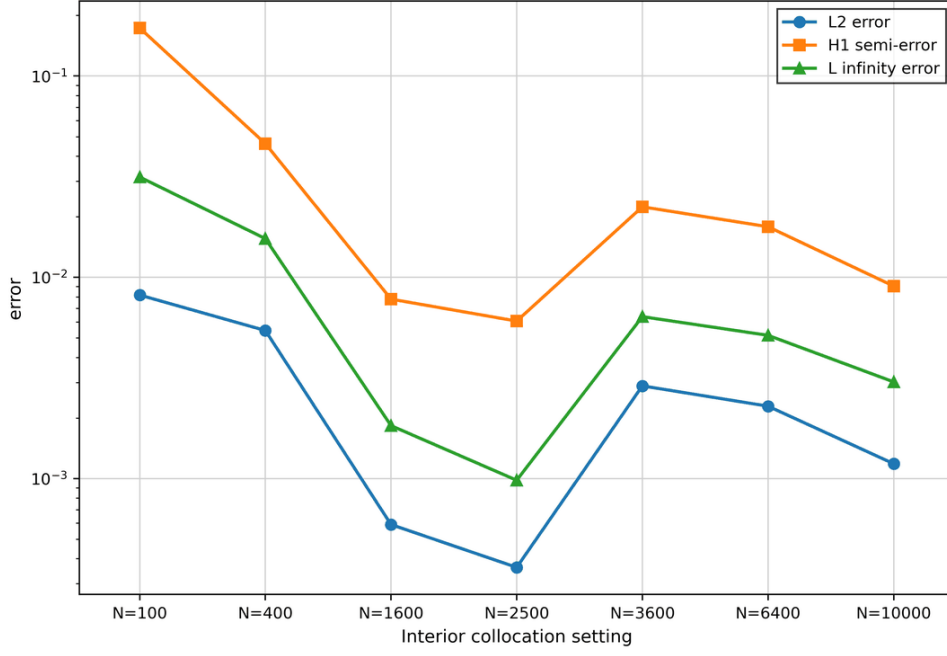


Figure 14: Experiment 6: final error norms for the CDR problem with smooth solution and GELU activation.

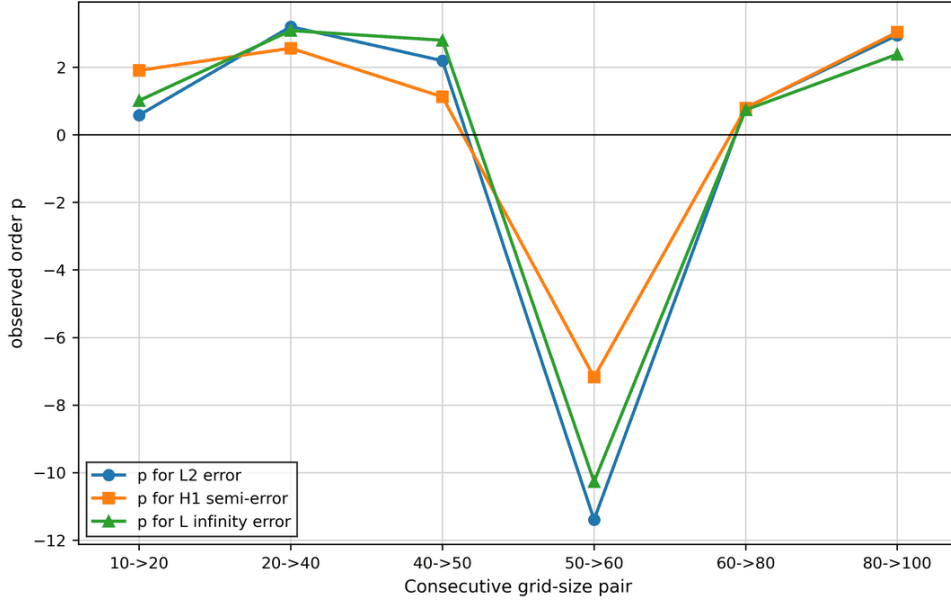


Figure 15: Experiment 6: observed convergence orders for the CDR problem with smooth solution and GELU activation.

4.9 Layer CDR Problem: GELU Activation

Experiment 7 is the analogue of Experiment 6 for the layer CDR problem. It is compared directly with Experiment 3: the PDE, exact solution, layer-type hard-constraint indicator, and collocation grids are unchanged, while the activation function is changed from tanh to GELU. This is the most demanding activation comparison, because the target solution contains localized outflow layers. The GELU variant improves some fine-grid results, in particular the $M = 100$

case, and the undershoot/overshoot values remain small on the finer grids. Nevertheless, the errors are again non-monotone. This indicates that the activation function can affect the ability to represent the layer, but accurate layer resolution still depends on the joint effect of sampling, indicator choice, architecture, and optimization.

Table 7: Experiment 7: layer CDR problem with GELU activation and layer-type indicator.

M	N	loss	L^2	H^1 semi	L^∞	undershoot	overshoot
10	100	9.093e+00	1.816e-02	2.043e-01	4.844e-02	4.564e-02	0.000e+00
20	400	6.780e-02	3.245e-03	6.764e-02	1.878e-02	2.291e-03	0.000e+00
40	1600	5.631e-02	1.276e-04	2.269e-03	3.738e-04	0.000e+00	0.000e+00
50	2500	4.238e-01	2.372e-03	2.005e-02	3.342e-03	2.335e-03	1.608e-03
60	3600	8.064e-02	7.037e-05	1.071e-03	5.171e-04	1.832e-06	0.000e+00
80	6400	2.991e-01	8.670e-04	3.819e-03	2.306e-03	0.000e+00	1.413e-03
100	10000	2.007e-01	1.709e-04	9.022e-04	9.452e-04	0.000e+00	0.000e+00

The undershoot and overshoot values are small for the finer grids and are reduced compared with several corresponding cases in Experiment 3. Thus, GELU improves the bound behaviour in some layer tests, although it does not remove all non-monotone effects in the computed solutions.

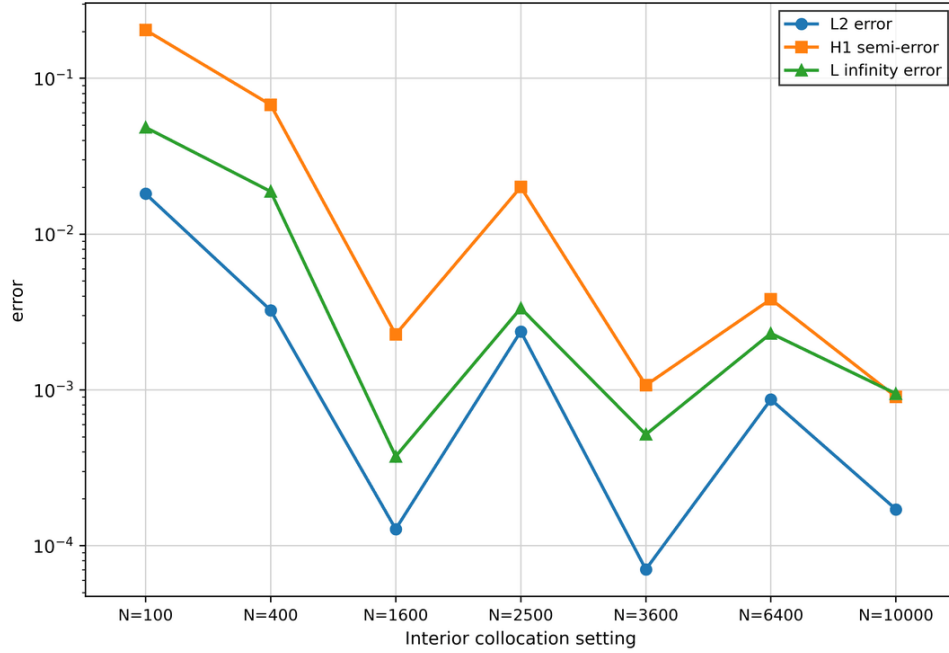


Figure 16: Experiment 7: final error norms for the layer CDR problem with GELU activation.

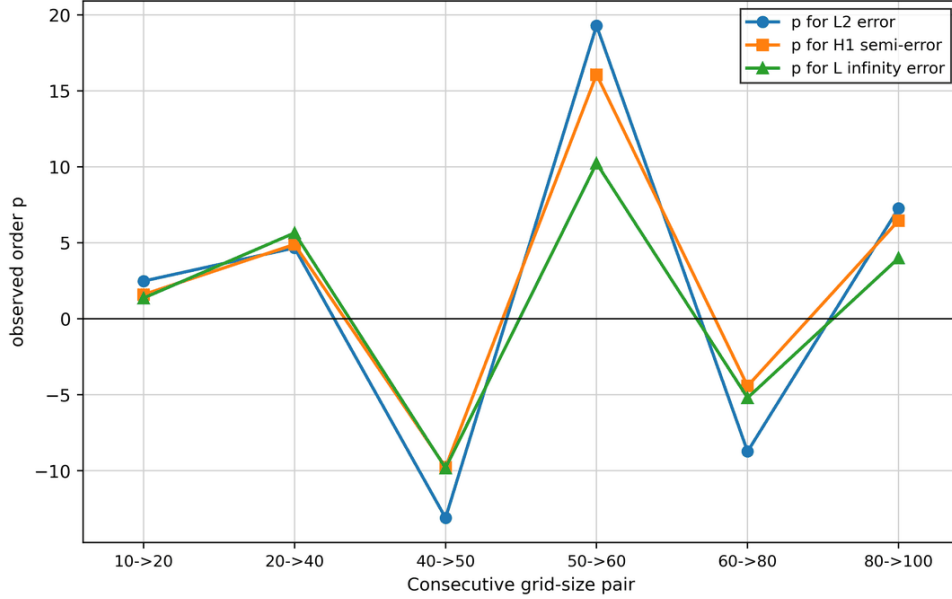


Figure 17: Experiment 7: observed convergence orders for the layer CDR problem with GELU activation.

4.10 Discussion of the Numerical Results

The numerical experiments show that increasing the number of collocation points alone does not guarantee a monotone decrease of the solution error. This behavior appears already in the Poisson problem and becomes more pronounced for the CDR examples. The reason is that the final error is affected by several mechanisms at the same time: the approximation capacity of the fixed architecture, the quadrature quality of the collocation points, the conditioning of the residual loss, and the finite optimization budget.

For the Poisson problem, the polynomial indicator $x(1-x)y(1-y)$ provides a stable way to impose homogeneous Dirichlet boundary conditions. Replacing $\tanh$ by GELU can improve the best achieved error, especially at $M = 100$, but the loss values are still irregular. In contrast, the exponential indicator with parameter 50 leads to much larger residual losses and poorer coarse-grid behavior. This suggests that the choice of indicator is part of the numerical method: it changes the scaling and gradient structure of the optimization problem, even though all variants satisfy the boundary condition by construction.

For the CDR problem with smooth solution, both the standard and GELU variants can reach small errors for selected grids. However, the convection and reaction terms make the training more sensitive than in the Poisson case. The residual loss and the actual solution error are not perfectly correlated, because different terms in the differential operator contribute with different scales. The GELU activation improves several fine-grid cases, but it still does not produce a strictly monotone convergence curve.

For the layer CDR problem, the main difficulty is the localized structure of the solution. The experiments show that the PINN can approximate the layer accurately in some cases, and the hard-constrained boundary treatment keeps boundary violations small. Nevertheless, uniform collocation points do not always resolve the layer equally well for every M . The comparison between the standard and GELU variants indicates that the activation function can help, but it does not remove the need to assess undershoot, overshoot, and pointwise error in addition to integral norms.

5 Conclusion

The analytical and numerical results presented in this thesis show that physics-informed neural networks provide a flexible framework for approximating steady elliptic boundary value problems, but that their behaviour cannot be understood from residual minimization alone. The variational theory of elliptic equations gives a stable reference point: coercivity, weak formulations, finite element error estimates, and maximum principles identify the analytical properties that a reliable approximation should respect. In the PINN setting, however, these properties are mediated by a nonlinear parametrization, empirical collocation, and gradient-based optimization. As a consequence, small residual values and large numbers of collocation points do not automatically imply uniformly accurate or qualitatively reliable approximations.

This conclusion is reflected in the numerical experiments. The Poisson problem can be approximated accurately with a hard-constrained polynomial indicator, but even this simple case does not exhibit completely monotone error decay under grid refinement. The smooth convection–diffusion–reaction problem remains accessible to the same architecture, yet the added convection and reaction terms make the relation between residual loss and actual error less regular. The layer example is more demanding still: sharp localized structures can be captured in selected cases, but the quality of the approximation depends sensitively on how the collocation points, activation function, boundary indicator, and finite training budget interact.

The experiments also indicate that implementation choices used to impose analytical constraints are part of the numerical method. Hard-constrained boundary conditions are effective for enforcing Dirichlet data, but the choice of indicator changes the trainable function class and the conditioning of the optimization problem. Similarly, replacing $\tanh$ by GELU can improve some fine-grid results, but it does not remove the fundamental non-monotonicity caused by approximation, sampling, and optimization errors. A reliable assessment of PINNs for elliptic problems should therefore combine residual information with error norms, pointwise bounds, undershoot and overshoot diagnostics, and comparisons across several collocation regimes. These observations suggest that future improvements should focus not only on larger networks or more collocation points, but also on problem-adapted boundary treatments, sampling strategies, and loss formulations that better reflect the qualitative structure of the underlying PDE.

References

- [1] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear pdes. *Journal of Computational Physics*, 378:686–707, 2019.
- [2] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: Deep learning of nonlinear pdes. *SIAM Journal on Scientific Computing*, 41:A260–A280, 2019.
- [3] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021.
- [4] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2 edition, 2010.
- [5] D. Gilbarg and N. S. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Springer, 2001.
- [6] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2008.
- [7] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations*. Springer, 2008.
- [8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
- [9] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [10] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.
- [11] L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60:223–311, 2018.
- [12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [14] A. G. Baydin, B. Pearlmutter, A. Radul, and J. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43, 2018.
- [15] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.
- [16] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.
- [17] A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.
- [18] D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.
- [19] M. Matthaiou, V. John, and M. Zainelabdeen. Bound-preserving physics-informed neural networks for steady-state convection-diffusion-reaction problems. *Computers & Mathematics with Applications*, 199:167–183, 2025.

- [20] D. Frerichs-Mihov, V. John, and M. Zainelabdeen. On collocation points for physics-informed neural networks applied to convection-dominated convection-diffusion problems. In *Numerical Mathematics and Advanced Applications ENUMATH 2023*, pages 335–344. Springer, 2025.
- [21] G. R. Barrenechea, V. John, and P. Knobloch. *Monotone Discretizations for Elliptic Second Order Partial Differential Equations*, volume 61 of *Springer Series in Computational Mathematics*. Springer, Cham, 2025.