

A Deep Fourier Residual Method for solving PDEs using Neural Networks

Jamie M. Taylor¹, David Pardo^{2,3,4}, and Ignacio Muga^{2,5}

¹*Department of Quantitative Methods, CUNEF University, Madrid, Spain*

²*Basque Center for Applied Mathematics (BCAM), Bilbao, Bizkaia, Spain*

³*University of the Basque Country (UPV/EHU), Leioa, Spain*

⁴*Ikerbasque (Basque Foundation for Sciences), Bilbao, Spain*

⁵*Instituto de Matemáticas, Pontificia Universidad Católica de Valparaíso, Chile.*

Abstract

When using Neural Networks as trial functions to numerically solve PDEs, a key choice to be made is the loss function to be minimised, which should ideally correspond to a norm of the error. In multiple problems, this error norm coincides with—or is equivalent to—the H^{-1} -norm of the residual; however, it is often difficult to accurately compute it. This work assumes rectangular domains and proposes the use of a Discrete Sine/Cosine Transform to accurately and efficiently compute the H^{-1} norm. The resulting Deep Fourier-based Residual (DFR) method efficiently and accurately approximate solutions to PDEs. This is particularly useful when solutions lack H^2 regularity and methods involving strong formulations of the PDE fail. We observe that the H^1 -error is highly correlated with the discretised loss during training, which permits accurate error estimation via the loss.

1 Introduction

The use of Deep Learning techniques employing Neural Networks (NNs) have been successful to solve a wide range of data-based problems across fields such as image processing, healthcare, and autonomous cars [1, 2, 20, 23, 34, 43, 52, 54]. Recently, there has been a surge of interest in the use of neural networks as function spaces that can be employed to obtain numerical solutions of Partial Differential Equations (PDEs) [5, 9, 37, 40, 47, 48]. Owing to the universal approximation theorem, and variants in Sobolev spaces [16, 24, 25, 30], it is known that a sufficiently wide or deep NN is able to approximate any given continuous function on a compact domain with arbitrary accuracy, and thus they make suitable function spaces for solving PDEs. The use of automatic differentiation (*autodiff*) [4] facilitates efficient numerical evaluation of derivatives, which allows algorithmic differentiation of the neural network itself, as well as the use of gradient-based optimisation techniques such as Stochastic Gradient Descent (SGD) [8] and Adam [31] in order to minimise appropriate loss functions over a space of neural networks.

A quantitative version of the Universal Approximation Theorem [3] demonstrates that NNs can approximate without suffering the curse of dimensionality, requiring far fewer degrees of freedom to approximate functions with high-dimensional inputs than classical piecewise-linear function spaces, making them an attractive function space for solving PDEs, in particular, in high-dimensional problems. Beyond solving single instances of PDEs, NNs have shown a capacity to learn *operators* that solve families of parametrised PDEs, allowing rapid “online” evaluation of solutions after an “offline” training of the network [13, 22, 32, 35, 36].

The flexibility of NNs to solve many classes of PDEs is owed to a general and simple framework, whereby one chooses an appropriate architecture of the NN, a loss function, whose minimiser should be an exact solution of the PDE, and an optimisation procedure to attempt to minimise the loss function. In this article, we focus on the choice of loss function when solving PDEs with NN function spaces. Previous works have considered losses based on strong [26, 44, 53] and weak [29, 28, 27] formulations of the PDE. However, the choice of a perfect loss function is generally not obvious as in practice solutions will only reach local minima, and the loss and error may have distinct or unknown convergence rates as one approaches either a local minimiser or a practically unattainable global minimiser.

Generally, a PDE operator can be described by a (possibly nonlinear) map $\mathcal{R} : X \rightarrow Y$, where X, Y are Banach spaces. The PDE then takes the form

$$\mathcal{R}(u) = 0. \quad (1)$$

For example, Poisson's equation, $-\Delta u(x) = f(x)$, on a domain Ω with homogeneous Dirichlet boundary condition and $f \in L^2(\Omega)$ may be interpreted in strong form via the map $\mathcal{R}^s : H^2(\Omega) \cap H_0^1(\Omega) \rightarrow L^2(\Omega)$ given by

$$\mathcal{R}^s(u) = \Delta u + f, \quad (2)$$

or in weak form via the map $\mathcal{R}^w : H_0^1(\Omega) \rightarrow H^{-1}(\Omega) := [H_0^1(\Omega)]^*$ given by

$$\langle \mathcal{R}^w(u), v \rangle_{H^{-1} \times H^1} = \int_{\Omega} \nabla u(x) \cdot \nabla v(x) - f(x)v(x) dx. \quad (3)$$

When employing NNs to numerically solve PDEs, a loss is often selected as the norm of the PDE residual in Y , that is,

$$\mathcal{L}(u) := \|\mathcal{R}(u)\|_Y. \quad (4)$$

One is generally confronted with two issues within this framework. The first is that norms on function spaces are generally given by integrals and thus a quadrature rule must be employed in order to numerically approximate $\mathcal{L}(u)$. In contrast to polynomial-based function spaces, an exact quadrature rule is generally unobtainable. Moreover, a poor choice of quadrature rule can lead to a form of "overfitting" and poor approximation of solutions [46]. The second issue is that in infinite dimensional spaces not all norms are equivalent, and thus the choice of norm on Y can directly affect the convergence of the error during training. Ideally, we should employ norms on X and Y which are compatible in the sense that the X -norm of the error is equivalent to the Y -norm of the residual, leading to a residual minimisation method [9, 10, 15].

Related to this second issue, progress has been made in the direction of *a priori* and *a posteriori* error estimates that allow estimation of the error via the loss [6, 7, 18, 38, 50, 51]. The works rely on coercivity-type estimates of the error in terms of the exact norms, as well as a control of quadrature and training errors.

Many PDEs can be expressed in weak form via (1) with $X = U$ is a space of trial functions, and $Y = V^*$, where V is the space of test functions. That is,

$$\langle \mathcal{R}(u), v \rangle_{V^* \times V} = 0 \quad \forall v \in V. \quad (5)$$

We commonly consider cases where \mathcal{R} represents a linear and inhomogeneous PDE, and thus may be expressed in the form

$$\langle \mathcal{R}(u), v \rangle_{V^* \times V} = b(u, v) - f(v), \quad (6)$$

where $f \in V^*$ and $b : U \times V \rightarrow \mathbb{R}$ is a bilinear form. It is clear that the PDE, in weak form, is equivalent to the statement that $\|\mathcal{R}(u^*)\| = 0$ for any norm on V^* . The most natural norm is the dual norm, induced by the norm on V , defined via

$$\|f\|_{V^*} = \sup_{v \in V \setminus \{0\}} \frac{|f(v)|}{\|v\|_V}. \quad (7)$$

The advantage of employing the dual norm on V^* is that, under certain assumptions that we will outline in more detail in Section 3.1, one can relate the dual norm of the residual to the norm of the error. Specifically, $\frac{1}{M} \|\mathcal{R}(u)\|_{V^*} \leq \|u - u^*\|_U \leq \frac{1}{\gamma} \|\mathcal{R}(u)\|_{V^*}$, where u is a candidate solution, u^* is the exact solution, and M, γ are positive, problem dependent, constants. This allows $\|\mathcal{R}(u)\|_{V^*}$ to be used as an error estimator, without needing to know the exact solution. In addition, if we can find a way to numerically approximate the dual norm, we can employ this as a loss function to be minimised over a trial function space.

We propose a Deep Fourier Residual (DFR) method to approximate the error of candidate solutions of PDEs in H^1 via an approximation of the dual norm of the residual of the PDE operator. The dual norm is then employed as a loss function to be minimised. The advantage of such a method is that the resulting norm is equivalent to the H^1 -error of the solutions for certain well-posed problems.

We consider several numerical examples, comparing the DFR approach to other losses employed to solve differential equations using NNs. Our numerical examples exhibit strong correlation between the proposed loss and H^1 -error during the training process. For sufficiently regular problems, our DFR method is qualitatively equivalent to existing methods in the literature (Section 4.1.2) [27, 44]. However, in less regular problems, our method leads to significantly more accurate solutions, both for an equation that admits a smooth solution with large gradients (Section 4.1.3), and for an elliptic equation with discontinuous parameters (Section 4.1.4). Indeed, methods based on the strong formulation of the PDE, such as PINNs [44], cannot be implemented for such applications. The DFR method is shown to be advantageous both when solutions admit $H^1 \setminus H^2$ regularity, and in regular problems where the forcing term has a large discrepancy between its L^2 and H^{-1} norm. We then consider further numerical experiments which demonstrate the DFR method's capability in a linear equation with point source (Section 4.2.1), a nonlinear ODE (Section 4.2.2), and a 2D linear problem (Section 4.2.3).

The DFR method is currently limited to rectangular domains where each face has either a Dirichlet or a Neumann Boundary condition. We rely on a Fourier-type representation of the H^{-1} norm that can be performed efficiently using the one-dimensional Discrete Cosine Transform and Discrete Sine Transform (DCT/DST), which are based on the Fast Fourier Transform (FFT), in each coordinate direction. Generally, an extension of our techniques to PDEs on arbitrary domains Ω would require access to an orthonormal basis of $H^1(\Omega)$, whose obtention may prove more costly than solving the PDE itself. Furthermore, the DST/DCT takes advantage of the FFT, which allows an inexpensive evaluation of the loss and would not be available in general domains. A possibility for the extension of the DFR method to arbitrary domains include methods analogous to embedded domain methods [19, 21, 33, 39, 41, 45, 49], which embeds domains with complex geometry into a simpler fictitious computational domain. It is also possible to borrow ideas from Goal-Oriented adaptivity (e.g., [42]) to the proposed DFR method, although this will be postponed for a future work.

The structure of the paper is as follows. In Section 2 we cover some preliminary concepts. The theoretical groundwork for the definition of the DFR method is presented in Section 3, with our proposed loss function defined in Section 3.3. Section 4.1 contains numerical examples comparing our proposed loss function with the VPINNs and collocation losses, which are roughly equivalent in regular problems, but we will demonstrate that the DFR method greatly outperforms VPINNs and PINNs when solutions are less regular. In Section 4.2 we consider further numerical experiments that demonstrate the DFR in equations with a point source, nonlinearities, and 2D results. Finally, concluding remarks are made in Section 5.

2 Preliminaries

2.1 Neural Networks

Neural networks are functions expressed as compositions of more elementary functions. In the simplest case of a fully connected feed-forward NN, an M -layer neural network is described by M layer functions, $L_i : \mathbb{R}^{N_i} \rightarrow \mathbb{R}^{N_{i+1}}$, that are of the form

$$L_i(x) = \sigma_i(A_i x + b_i), \quad (8)$$

where A_i is an $N_i \times N_{i+1}$ matrix, $b_i \in \mathbb{R}^{N_{i+1}}$, and σ_i is an *activation function* that may depend on the layer index i and acts component-wise on vectors. A fully-connected feed forward neural network is a function $\tilde{u} : \mathbb{R}^{N_1} \rightarrow \mathbb{R}^{N_{M+1}}$ defined by

$$\tilde{u}(x) = L_M \circ L_{M-1} \circ \dots \circ L_1(x). \quad (9)$$

The final activation function σ_M is taken to be the identity, $\sigma_M(x) = x$. The parameters A_i, b_i , known as the *weights* and *biases* of the network, parametrise the neural network. Optimisation over a neural network space with fixed architecture corresponds to identifying the optimal values of these trainable parameters.

In the context of NNs for PDEs, we often need to impose homogeneous Dirichlet boundary conditions on our candidate solutions. In this work, we will do this by introducing a cutoff function. That is, if we wish to consider functions $u : \Omega \rightarrow \mathbb{R}$ so that for a subset of the boundary $\Gamma_D \subset \partial\Omega$, $u|_{\Gamma_D} = u_0$, we take \tilde{u} to be of the form (9), and define

$$u(x) = \phi_1(x)\tilde{u}(x) + \phi_2(x), \quad (10)$$

where ϕ_1 is a function satisfying $\phi_1|_{\Gamma_D} = 0$ and $\phi_1 > 0$ on $\bar{\Omega} \setminus \Gamma_D$, and $\phi_2|_{\Gamma_D} = u_0$.

We include a schematic of this architecture in Figure 1

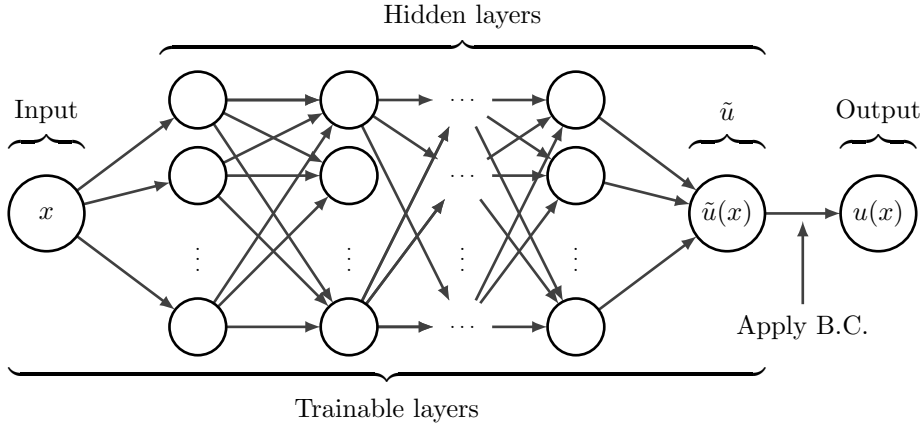


Figure 1: NN architecture

2.2 PINN and VPINN losses

Whilst there are many discrete losses employed when solving PDEs via NNs, in this section we outline two particular cases, the PINN (collocation), and the VPINN losses, which are based on strong and weak formulations of the PDE, respectively. These methods will be used for comparison in the numerical experiments of Section 4.1.

2.2.1 Collocation

We assume that the strong form of the residual can be represented in the form

$$\begin{aligned} L_u(x) &= 0 \quad (x \in \Omega), \\ G_u(x) &= 0 \quad (x \in \partial\Omega). \end{aligned} \tag{11}$$

The collocation method considers discretisations of the L^2 norms of L_u and G_u as the loss function to be minimised, according to an appropriate quadrature rule. Explicitly, we consider the loss

$$\mathcal{L}_{col}(u) := \frac{1}{K_1} \sum_{i=1}^{K_1} \omega_i |L_u(x_i)|^2 + \frac{1}{K_2} \sum_{i=1}^{K_2} \omega_i^b |G_u(x_i^b)|^2, \tag{12}$$

where $(x_i)_{i=1}^{K_1}$ and $(\omega_i)_{i=1}^{K_1}$ are quadrature points in Ω and quadrature weights, respectively, which may be taken via a Monte Carlo or a deterministic quadrature scheme. Similarly, $(x_i^b)_{i=1}^{K_2}$ and $(\omega_i^b)_{i=1}^{K_2}$ are quadrature points and weights on the boundary.

In PDEs with low regularity, the strong form of the PDE does not hold and minimisers of Eq. (12) will not accurately represent the PDE. Despite this limitation, the collocation (PINN) method is one of the most attractive methods for regular problems as it is simple to implement using *autodiff* algorithms. Furthermore, by using Monte Carlo integration techniques, integrals can be estimated in high dimension without suffering from the curse of dimensionality.

2.2.2 VPINNs

VPINNs employ a loss that utilizes the weak formulation of the PDE. They correspond to a Petrov-Galerkin method where the trial space is given by NNs. Given a set of test functions $(v_k)_{k=1}^K$, a candidate solution u and the residual $\mathcal{R}(u) \in V^*$ given in weak form, the loss is defined as

$$\mathcal{L}_{VP}(u) = \sum_{k=1}^K |\langle \mathcal{R}(u), v_k \rangle_{V^* \times V}|^2. \tag{13}$$

In [27], this method was shown to be advantageous over classical PINNs method, both in terms of accuracy and speed. A particular application within their work, relevant to this manuscript, was to consider ODEs on $[0, 1]$ with a NN architecture that consists of a single hidden layer with sine activation function, and test functions $v_k(x) = \sin(k\pi x)$. For this implementation, the authors were able to perform an exact quadrature to evaluate $\langle \mathcal{R}(u), v_k \rangle_{V^* \times V}$, which was employed in their loss function. In other implementations within their article, Legendre polynomials are considered as test functions. Whilst not directly commented within their work, in their implementation with sine test functions, the norm may be interpreted as a discretisation of the L^2 -norm of the strong form of the residual. As they consider the test functions $(v_k)_{k=1}^K$ form to be a subset of an orthonormal basis of L^2 , if there exists a strong form residual $L_u \in L^2$ such that

$$\langle \mathcal{R}(u), v \rangle_{V^* \times V} = \langle L_u, v \rangle_{L^2}$$

for all $v \in H_0^1$, we observe that

$$\sum_{k=1}^K |\langle \mathcal{R}(u), v_k \rangle_{V^* \times V}|^2 = \sum_{k=1}^K \langle L_u, v_k \rangle_{L^2}^2 \approx \|L_u\|_{L^2}^2. \tag{14}$$

In particular, for sufficiently regular problems, this implies that \mathcal{L}_{VP} and \mathcal{L}_{col} each correspond to distinct discretisations of the same loss, i.e., the L^2 -norm of the strong-form residual. The significant difference, however, is that the discretisation (13) is always well defined, even if the residual cannot be represented by an L^2 function, and we will observe the consequences of this distinction in Section 4.1.4, employing sine-based test functions, as in [27].

3 DFR Method

For exposition purposes we will only list the key results necessary for defining the problem, with the details deferred to appendices.

3.1 Dual Norms and Residual Minimisation

Let us consider a PDE of the form (5), described by a weak-form residual operator $\mathcal{R} : U \rightarrow V^*$, which is linear and inhomogeneous, so that it may be expressed as in (6). We assume V to be a Hilbert space, and take $f \in V^*$, and $b : U \times V \rightarrow \mathbb{R}$ to be a bilinear form satisfying the continuity condition

$$|b(u, v)| \leq M \|u\|_U \|v\|_V \quad (15)$$

and inf-sup stability condition

$$\inf_{u \in U \setminus \{0\}} \sup_{v \in V \setminus \{0\}} \frac{|b(u, v)|}{\|u\|_U \|v\|_V} \geq \gamma, \quad (16)$$

where $0 < \gamma < M$.

If we consider the map $B : U \rightarrow V^*$ given by $B : u \mapsto b(u, \cdot)$, then the conditions (15) and (16) ensure that B is a boundedly invertible map onto its image $B(U)$, and, in particular, if B is surjective, then there exists a unique solution to (5) for all $f \in V^*$ [14, Section 6.12]. Furthermore, if f is in the range of B , given the exact solution $u^* \in U$ to (5), and a candidate solution $u \in U$, we may estimate the error using the dual norm of the residual via the inequalities

$$\frac{1}{M} \|\mathcal{R}(u)\|_{V^*} \leq \|u - u^*\|_U \leq \frac{1}{\gamma} \|\mathcal{R}(u)\|_{V^*}. \quad (17)$$

Both inequalities are found by noting that since $\mathcal{R}(u^*) = 0$, then

$$\langle \mathcal{R}(u), v \rangle_{V^* \times V} = \langle \mathcal{R}(u), v \rangle_{V^* \times V} - \langle \mathcal{R}(u^*), v \rangle_{V^* \times V} = b(u - u^*, v)$$

for any test function $v \in V$. Correspondingly, the lower bound is a direct consequence of (15), as

$$\sup_{v \in V \setminus \{0\}} \frac{|\langle \mathcal{R}(u), v \rangle_{V^* \times V}|}{\|v\|_V} = \sup_{v \in V \setminus \{0\}} \frac{|b(u - u^*, v)|}{\|v\|_V} \leq M \|u - u^*\|_U. \quad (18)$$

Similarly, the upper bound is a direct consequence of (16), as

$$\sup_{v \in V \setminus \{0\}} \frac{|\langle \mathcal{R}(u), v \rangle_{V^* \times V}|}{\|v\|_V} = \sup_{v \in V \setminus \{0\}} \frac{|b(u - u^*, v)|}{\|v\|_V} \geq \gamma \|u - u^*\|. \quad (19)$$

This makes $\|\mathcal{R}(u)\|_{V^*}$ a natural choice of norm to be utilised as a loss function for training NNs. Generally, however, it is non-trivial to evaluate $\|\cdot\|_{V^*}$, which is defined as in (7), and, unlike classical Sobolev-type norms, generally cannot be expressed as a single integral of the function and its derivatives.

3.1.1 Nonlinear equations

Whilst our previous discussion applies only to linear equations, via a linearisation argument it is possible to obtain a local version of (17) for nonlinear PDE. We consider an abstract PDE given of the following form. Considering $\mathcal{R} : U \rightarrow V^*$ to be nonlinear, we obtain the following result.

Proposition 3.1. *Assume that there exists a (possibly non-unique) solution $u^* \in U$ of $\mathcal{R}(u^*) = 0$ such that:*

- (i) *There exists $r_0 > 0$ such that \mathcal{R} is Gateaux differentiable for all $u \in U$ with $\|u - u^*\|_U < r_0$.*

(ii) The directional derivative $\delta_w \mathcal{R}(u^*)$ is bounded below, so that there exists $\gamma > 0$ such that for all $w \in U$

$$\|\delta_w \mathcal{R}(u^*)\|_{V^*} \geq \gamma \|w\|_U.$$

(iii) There exists some $r_0 > 0$ such that the Gateaux derivative of \mathcal{R} is Lipschitz on the ball $B_{r_0}(u^*)$.

Then for every $0 < \epsilon < 1$ there exists $\delta > 0$ such that if $\|u - u^*\|_U < \delta$,

$$\frac{1 - \epsilon}{M} \|\mathcal{R}(u)\|_{V^*} \leq \|u - u^*\|_U \leq \frac{1 + \epsilon}{\gamma} \|\mathcal{R}(u)\|_{V^*}. \quad (20)$$

We defer the precise definitions of the objects in the proposition and its proof to Appendix C. If \mathcal{R} corresponds to a linear inhomogeneous PDE and is thus equal to its linearisation, δ can be taken as $+\infty$ and the estimate is global. The significance of this result is that if we have a PDE described by a sufficiently regular \mathcal{R} and a candidate solution sufficiently close to an exact solution, which need not be unique, they will satisfy estimates analogous to (17). In particular, if $\|\mathcal{R}(u)\|_{V^*}$ is used as a loss function, and the candidate solution is close enough to the exact one (which is to be expected at the end of training), we should observe strong a correlation between the loss and the H^1 -error. We will numerically illustrate this in Section 4.2.2.

3.2 Evaluation of the H^{-1} norm with the DFR method

This work takes advantage of a Parseval-type inequality to evaluate the dual norm of elements of the dual space in terms of an orthonormal basis of the original Hilbert space.

Proposition 3.2. *Let V be a real separable Hilbert space with inner product $\langle \cdot, \cdot \rangle_V : V \times V \rightarrow \mathbb{R}$, and $(\varphi_k)_{k=1}^\infty$ a countable, orthogonal basis of V , with $\|\varphi_k\|_V^2 =: \lambda_k$. Then for all $f \in V^*$,*

$$\|f\|_{V^*} := \max_{v \in V \setminus \{0\}} \frac{|f(v)|}{\|v\|_V} = \left(\sum_{k=1}^{\infty} \lambda_k^{-1} f(\varphi_k)^2 \right)^{\frac{1}{2}}. \quad (21)$$

Proof. By the Riesz Representation Theorem, there exists a unique solution $u_f \in V$ of

$$\langle u_f, v \rangle_V = f(v) \quad (22)$$

for all $v \in V$, that satisfies $\|u_f\|_V = \|f\|_{V^*}$, and thus the mapping $f \mapsto u_f$ is an isometry. As $(\varphi_k)_{k=1}^\infty$ is an orthogonal basis of V , we then have via the generalised Parseval identity that

$$\|f\|_{V^*} = \|u_f\|_V = \left(\sum_{k=1}^{\infty} \frac{\langle u_f, \varphi_k \rangle_V^2}{\|\varphi_k\|_V^2} \right)^{\frac{1}{2}} = \left(\sum_{k=1}^{\infty} \lambda_k^{-1} f(\varphi_k)^2 \right)^{\frac{1}{2}}. \quad (23)$$

□

Our approach consists of approximating the dual norm of elements $f \in V^*$ using a truncated version of this series expression. To do so, we need an appropriate orthogonal basis $(\varphi_k)_{k=1}^\infty$ of V .

In this work, we restrict ourselves to problems where the space of test functions is $V = \{u \in H^1(\Omega) : u|_{\Gamma_D} = 0\}$, where $\Gamma_D \subset \partial\Omega$ is the region of the boundary corresponding to a Dirichlet boundary condition of the PDE. For this case, we take our orthogonal basis of V to be the (weak) solutions of

$$\begin{aligned} \lambda_k \int_{\Omega} v(x) \varphi_k(x) dx &= \int_{\Omega} \nabla \varphi_k(x) \cdot \nabla v(x) dx + \varphi_k(x) v(x) dx \quad (\forall v \in V) \\ \|\varphi_k\|_{L^2} &= 1. \end{aligned} \quad (24)$$

In strong form, we may write the PDE as

$$(1 - \Delta)\varphi_k(x) = \lambda_k\varphi_k(x). \quad (25)$$

That is, φ_k are eigenvectors of $1 - \Delta$ with homogeneous Dirichlet condition on Γ_D and homogeneous Neumann condition on $\partial\Omega \setminus \Gamma_D = \Gamma_N$, with corresponding eigenvalues λ_k , and normalised to have unit norm in $L^2(\Omega)$. The key properties of φ_k , with proofs deferred to Appendix A and derived from classical spectral theory, are:

1. $(\varphi_k)_{k=1}^\infty$ forms an orthogonal basis of V , and an orthonormal basis of $L^2(\Omega)$.
2. $\|\varphi_k\|_{H^1}^2 = \lambda_k$, where λ_k is the eigenvalue of $1 - \Delta$ corresponding to φ_k .
3. $\lambda_k \geq 1$ for all k and, under a suitable reordering, λ_k is non-decreasing and unbounded.

In this case, we see that f and u_f are related via $(1 - \Delta)u_f = f$, and we may interpret (21) as stating that $\|f\|_{V^*} = \|(1 - \Delta)^{-1}f\|_{H^1}$, where our series expression allows us to evaluate the latter in a straightforward manner.

In general geometries, it is non-trivial to identify the eigenvectors and eigenvalues of $1 - \Delta$ on a domain, and thus for the sake of this work we consider only simple geometries described by n -dimensional cubes, $\Omega = (0, \pi)^n$, with Γ_D given by a union of any number of the $2n$ faces of $\partial\Omega$. In this case, the eigenvectors of $1 - \Delta$ are simply products of the eigenvectors of $1 - \frac{d}{dx_i^2}$ in each coordinate direction with the appropriate boundary conditions, and thus may be written explicitly. Moreso, as these eigenvectors are all described as sines and cosines of varying frequencies, we will be able to take advantage of the Discrete Sine/Cosine Transforms to efficiently evaluate the residual at the basis functions.

Considering the one-dimensional problem, as $\partial[0, \pi]$ consists of only two points, there are only four options for $\Gamma_D \subset \partial[0, \pi]$. Table 1 lists the four possible boundary conditions, along with corresponding eigenvalues λ_k and eigenvectors φ_k .

Γ_D	λ_k	φ_k	$\langle \cdot, \varphi_k \rangle_{L^2}$	$\langle \cdot, \varphi_k' \rangle_{L^2}$
$\{0, \pi\}$	$1 + k^2$	$\sqrt{\frac{2}{\pi}} \sin(kx)$	DST-II	DCT-II
$\{0\}$	$1 + (k - \frac{1}{2})^2$	$\sqrt{\frac{2}{\pi}} \sin((k - \frac{1}{2})x)$	DST-IV	DCT-IV
$\{\pi\}$	$1 + (k - \frac{1}{2})^2$	$\sqrt{\frac{2}{\pi}} \cos((k - \frac{1}{2})x)$	DCT-IV	DST-IV
\emptyset	$1 + (1 - k)^2$	$\sqrt{\frac{2}{\pi}} \cos((k - 1)x) - \delta_{k1}\pi^{-\frac{1}{2}}$	DCT-II	DST-II

Table 1: Basis functions and eigenvalues for $H^1(0, \pi)$ with various boundary conditions, along with the relevant transforms for evaluating integrals against the basis functions and their derivatives.

To evaluate PDE residuals acting on basis functions of the above forms, we will need to numerically evaluate integrals involving basis functions and their derivatives. As these are global basis functions, a naive calculation of the integrals of each basis function could prove prohibitively expensive. To remedy this, we consider the Discrete Sine/Cosine transforms as a means of quadrature, which reduce the number of calculations required to use an N point midpoint rule for N basis functions from $O(N^2)$ to $O(N \ln N)$. Table 1 contains in its fourth and fifth columns the quadrature scheme for evaluating integrals against the basis functions and their derivatives, respectively. We defer detailed discussion of the transforms to Appendix B. Their key use is that they are analogous to the Fast Fourier Transform, where the boundary conditions

are no longer periodic. In fact, their efficient calculation arises from their representation as special cases of the Discrete Fourier Transform under particular symmetries.

These basis functions are easily adapted to arbitrary intervals (a, b) by a rescaling argument. We may define corresponding orthonormal basis functions $\tilde{\varphi}_k \in H^1(a, b)$ by

$$\tilde{\varphi}_k(x) = \varphi_k\left(\frac{\pi(x-a)}{b-a}\right), \quad (26)$$

which are eigenvectors of $1 - \Delta$ on (a, b) with corresponding eigenvalues

$$\tilde{\lambda}_k = \frac{\pi^2}{(b-a)^2}(\lambda_k - 1) + 1. \quad (27)$$

Similarly, by considering tensor products of these 1D basis functions, we can obtain an orthonormal basis for $H^1(\Omega)$ with the appropriate boundary conditions when $\Omega = \prod_{i=1}^d (a_i, b_i)$.

3.3 Definition of the discretised loss

Our aim is to define a computable, discretised loss, \mathcal{L}_{V^*} , such that for a candidate solution u , we have that

$$\mathcal{L}_{V^*}(u) \approx \|\mathcal{R}(u)\|_{V^*}^2. \quad (28)$$

We will do this by employing a truncated series expansion of (21), and taking the sine/cosine based basis functions outlined in Table 1, with quadrature corresponding to the DCT/DST, as outlined in Appendix B. Before defining the loss in the general case, we outline its definition in a simple one-dimensional example for clearer exposition.

3.3.1 One-dimensional example

Let $f \in L^2(0, \pi)$ and $g \in \mathbb{R}$. Let us consider the ODE, in weak form, to be: find $u \in H^1(0, \pi)$ with $u(0) = 0$, such that

$$\langle \mathcal{R}(u), v \rangle_{V^* \times V} = \int_0^\pi \sigma(x) u'(x) v'(x) + f(x) v(x) dx - gv(\pi) = 0 \quad (29)$$

for all $v \in H^1(0, \pi)$ with $v(0) = 0$. Consulting Table 1, we see that our relevant basis functions for V corresponding to our homogeneous Dirichlet boundary condition at 0 are given by $\varphi_k(x) = \sqrt{\frac{2}{\pi}} \sin\left(\left(k - \frac{1}{2}\right)x\right)$ with corresponding eigenvalues $\lambda_k = 1 + \left(k - \frac{1}{2}\right)^2$. The derivatives of our basis functions are readily evaluated as $\varphi_k'(x) = \left(k - \frac{1}{2}\right) \sqrt{\frac{2}{\pi}} \cos\left(\left(k - \frac{1}{2}\right)x\right)$.

First, we choose a truncation frequency, $N > 0$. For $1 \leq k \leq N - 1$, we now aim to approximate $\langle \mathcal{R}, u, \varphi_k \rangle_{V^* \times V}$. Recalling the approximations in (57), we may then approximate the integrals appearing in the residual as

$$\begin{aligned} \int_0^\pi f(x) \varphi_k(x) dx &= \int_0^\pi f(x) \sqrt{\frac{2}{\pi}} \sin\left(\left(k - \frac{1}{2}\right)x\right) dx \\ &\approx \mathcal{S}_{N,k}^{II}(f), \\ \int_0^\pi \sigma(x) u'(x) \varphi_k'(x) dx &= \int_0^\pi \left(k - \frac{1}{2}\right) \sqrt{\frac{2}{\pi}} \cos\left(\left(k - \frac{1}{2}\right)x\right) \sigma(x) u'(x) dx \\ &\approx \left(k - \frac{1}{2}\right) \mathcal{C}_{N,k}^{II}(\sigma u'), \end{aligned} \quad (30)$$

where *autodiff* is employed to evaluate u' for a candidate solution u described by a NN, and $\mathcal{C}_{N,k}^{II}, \mathcal{S}_{N,k}^{II}$ are the type-II DCT and DST, respectively, as described in Appendix B. The boundary term may be evaluated exactly as

$$g\varphi_k(\pi) = g\sqrt{\frac{2}{\pi}} \sin\left(\left(k - \frac{1}{2}\right)\pi\right) = g\sqrt{\frac{2}{\pi}}(-1)^{k+1}. \quad (31)$$

Thus, we define the discretised transform of the residual, $\hat{\mathcal{R}}(u)(k)$ for $1 \leq k \leq N-1$ as

$$\hat{\mathcal{R}}(u)(k) := \left(k - \frac{1}{2}\right) \mathcal{C}_{N,k}^{II}(\sigma u') + \mathcal{S}_{N,k}^{II}(f) - g\sqrt{\frac{2}{\pi}}(-1)^{k+1}. \quad (32)$$

Finally, our discretised loss, denoted \mathcal{L}_{V^*} , is defined to be

$$\mathcal{L}_{V^*}(u) := \sum_{k=1}^{N-1} \frac{|\hat{\mathcal{R}}(u)(k)|^2}{1 + \left(k - \frac{1}{2}\right)^2}. \quad (33)$$

3.3.2 The general case

Let $\Omega = (0, \pi)^d$, and take $\Gamma_D \subset \partial\Omega$ to be a union of faces of the rectangular domain Ω . Take $V = \{u \in H^1(\Omega) : u|_{\Gamma_D} = 0\}$. For a candidate solution $u \in H^1(\Omega)$, consider the residual

$$\langle \mathcal{R}(u), v \rangle_{V^* \times V} = \int_{\Omega} F_u^1(x) \cdot \nabla v(x) + F_u^2(x)v(x) dx - \int_{\Gamma_N} G_u(x)v(x) dx, \quad (34)$$

where the functions F_u^1, F_u^2, G_u may be functions of x, u and derivatives of u .

The loss is thus defined according to the following process:

1. Choose a cutoff frequency $N > 0$.
2. Identify the correct basis functions $\varphi_{k_1, k_2, \dots, k_d}(x_1, x_2, \dots, x_d) = \prod_{i=1}^d \varphi_{k_i}^i(x_i)$, where $\varphi_{k_i}^i$ are the 1D basis functions described in Table 1 according to the boundary conditions on each face.
3. Identify the correct eigenvalues $\lambda_{k_1, k_2, \dots, k_d} = 1 - d + \sum_{i=1}^d \lambda_{k_i}$ for each basis function according to Table 1.
4. Express the residual operator $\mathcal{R}(u)$ in weak form, evaluate integrals across the interior and faces by performing the appropriate DCT/DST in each coordinate direction, according to the fourth and fifth columns of Table 1 for $k_1, \dots, k_d = 1, \dots, N-1$ to give $\langle \mathcal{R}(u), \varphi_{k_1, \dots, k_d} \rangle_{V^* \times V} \approx \hat{\mathcal{R}}(u)(k_1, \dots, k_d)$.
5. Evaluate the loss as

$$\mathcal{L}_{V^*}(u) := \sum_{i=1}^d \sum_{k_i=1}^{N-1} \frac{|\hat{\mathcal{R}}(u)(k_1, k_2, \dots, k_d)|^2}{\lambda_{k_1, k_2, \dots, k_d}}.$$

3.4 Potential limitations

We have two sources of error in the approximation (28). First, the error arising from quadrature, according to our mid-point rule for integration and evaluation of $\hat{\mathcal{R}}(u)(k)$. Second, errors arise from the truncation of the infinite series in (21). In contrast, the quadrature rule employed by the DCT/DST is exact when $\mathcal{R}(u)$, represented as a function in H^1 via the Riesz Representation Theorem belongs to the span of $((\varphi_{k_1 k_2, \dots, k_d})_{k_i=1}^N)_{i=1}^d$, where N is the cutoff frequency employed. That is, our discretisation error corresponds to high-frequencies of the *residual*—not to be confused with high frequencies of the solution. In principle, high-frequency components in the residual may arise as a consequence of the following:

1. The residual itself contains high-frequency modes due to the presence of terms with low-regularity.
2. The function space used has low regularity, such as NNs with a *ReLU* activation function.
3. The function space is flexible enough and the number of integration points is low enough that overfitting occurs during minimisation, which would introduce high-frequency modes, unseen by the loss.

By using a sufficiently high cutoff frequency $N > 0$, errors corresponding to (1) should be negligible. To avoid the possibility of (2), we employ smooth activation functions. With regards to (3), it is known that extreme quadrature issues can arise when training NNs to solve PDEs [46], which may be interpreted as a form of overfitting. We do not focus on this issue within this work and we use a validation and a training set to verify if overfitting occurs.

4 Numerical Experiments

4.1 Validation Results

We now consider various linear ODEs to illustrate the differences between different losses for training a neural network when solving linear PDEs. \mathcal{L}_{V^*} , \mathcal{L}_{VP} , and \mathcal{L}_{col} denote our proposed method, the VPINNs loss, and the collocation method, respectively. The obtained solutions are denoted as u_{V^*} , u_{VP} , and u_{col} , respectively.

Figure 1 describes the NN architecture of our candidate solutions. It consists of five hidden layers with tanh activation function and 25 neurons per layer. We only consider homogeneous Dirichlet boundary conditions, which are implemented according to (10) by taking $\phi_1(x) = x$ when $\Gamma_D = \{0\}$, and $\phi_1(x) = x(\pi - x)$ when $\Gamma_D = \{0, \pi\}$. For consistency between experiments, each candidate solution is initialised with the same weights and biases.

Our implementation uses *Tensorflow 2.8*. We use Adam as our optimiser with initial learning rate 10^{-2} , and an adaptive learning rate, as defined in [55], and implemented via a *Callback*. The adaptive learning rate allows the optimiser to select the “correct” learning rate according to the decay of the loss, and rejects iteration steps which lead to an increase in the loss. This choice accelerates convergence, and allows a fairer comparison between the three methods considered, as otherwise the convergence may be highly dependent on the selected learning rate.

In each case, we minimise the loss using 200 points, which in the Fourier-based losses corresponds to employing the first 200 basis functions in the truncated series expansions (21) and (13), and 200 equispaced integration points in the collocation method with a mid-point integration rule. We also measure the loss on a validation set of 274 points so that we may see if overfitting takes place, which does not factor into the updating of the NN weights, and is only used as a metric for comparison after training. In the Fourier-based losses, this also corresponds to a total of 274 frequencies used to evaluate the validation loss.

4.1.1 Losses implemented

For comparison, we implement the collocation based loss as described in Section 2.2.1, and the VPINNs loss as described in Section 2.2.2. For the latter, we consider an implementation that is highly comparable to the loss \mathcal{L}_{V^*} that we propose. Explicitly, we take

$$\mathcal{L}_{VP}(u) := \sum_{k=1}^N |\hat{\mathcal{R}}(u)(k)|^2. \quad (35)$$

where $\hat{\mathcal{R}}(u)(k)$ is defined in Section 3.3.2, and we evaluate this using DST/DCT. Thus, we have an application of VPINNs that allows a direct comparison between using an L^2 -based norm and an H^{-1} -based norm for evaluating the PDE residual in weak form. When $V = H_0^1(0, \pi)$,

this is consistent with VPINNs as considered in [27, Section 4.1], which used sine-based test functions albeit with a different quadrature rule. The only difference between (35) and \mathcal{L}_{V^*} is the weighting factor λ_k^{-1} imposed in the summation. As discussed in Section 2.2.2, this leads to the interpretation of \mathcal{L}_{VP} as a discretisation of the L^2 norm of the strong formulation of the residual. As \mathcal{L}_{col} is also a discretisation of the L^2 norm of the strong-form residual, we expect implementations that utilise \mathcal{L}_{col} and \mathcal{L}_{VP} to generally behave the same when the strong form is well-defined. The key difference, however, is that \mathcal{L}_{VP} is still well-defined when the strong form of the PDE is not equivalent to the weak form.

4.1.2 Model Problem 1 - Smooth solution

We consider the following ODE in variational form: find $u \in H_0^1(0, \pi)$ satisfying

$$\int_0^\pi u'(x)v'(x) - 4\sin(2x)v(x) dx = 0 \quad (36)$$

for all $v \in H_0^1(0, \pi)$. This has exact solution given by $u^*(x) = \sin(2x)$.

This example is selected because the weak and strong forms of the PDE are equivalent, and the solution only admits low-frequency modes. For such problems, we expect that $\mathcal{L}_{VP}(u) \approx \mathcal{L}_{col}(u)$, with the approximation being exact in the limit as the number of sampling points tends to infinity. We further expect all three implemented losses to behave similarly since only low-frequency modes play a significant role in this problem.

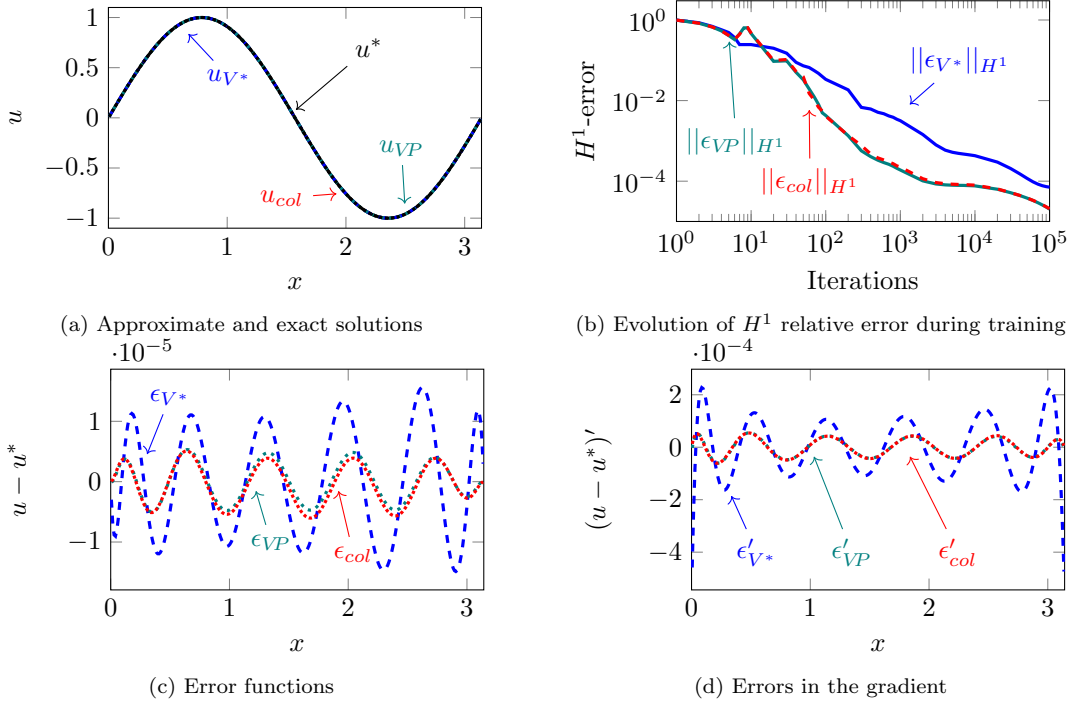


Figure 2: Model Problem 1. Obtained solutions and relative H^1 -error evolution for the three methods

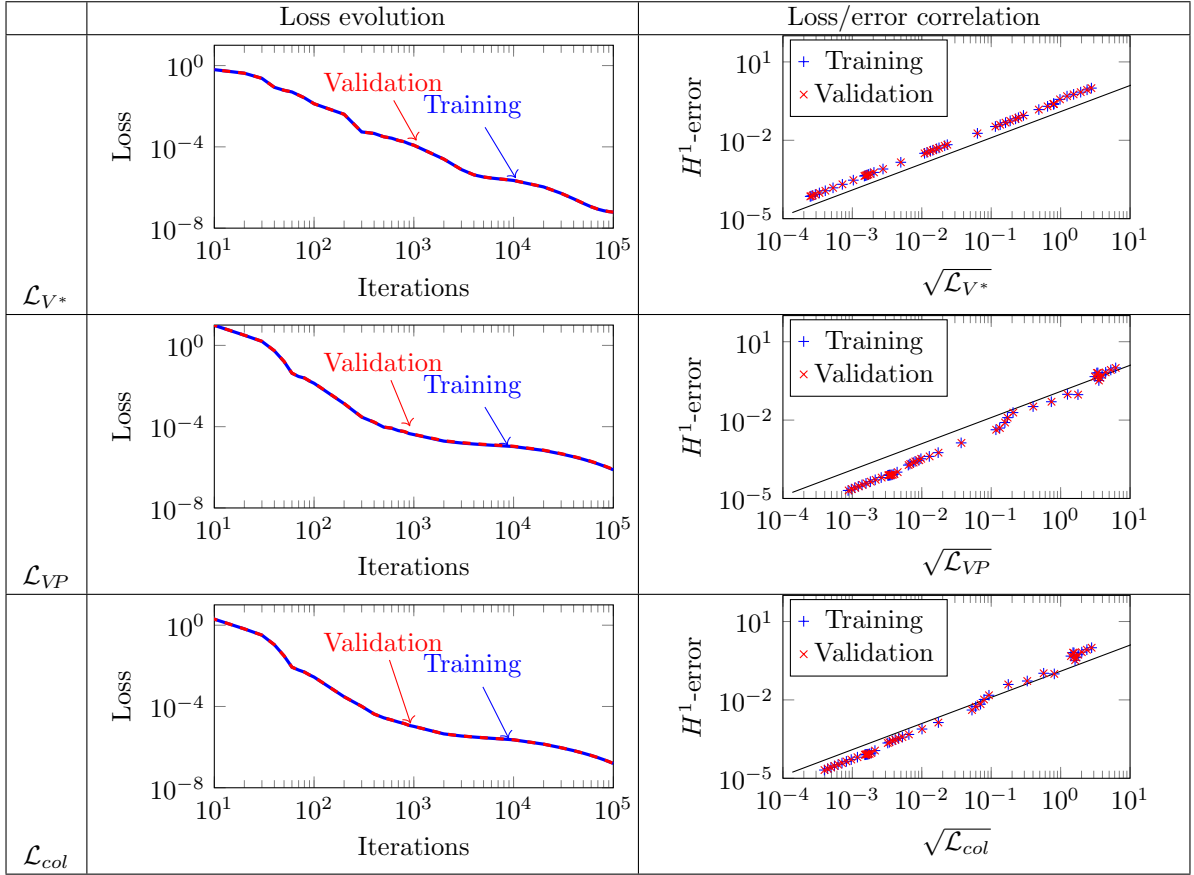


Figure 3: Model Problem 1. Evolution of the loss for the three considered losses on both the training and validation data sets, and the correlation between the loss and the relative H^1 -error during training, with a straight line corresponding to a linear relationship.

	H1 (%)	L2 (%)
u_{V^*}	7.11×10^{-3}	1.26×10^{-3}
u_{VP}	2.03×10^{-3}	4.56×10^{-4}
u_{col}	2.07×10^{-3}	5.16×10^{-4}

Table 2: Model Problem 1: Errors and losses

Figure 2 shows the obtained solutions and their errors, along with the H^1 -relative error evolution during training, where we denote errors by ϵ , so that $\epsilon_X = u^* - u_X$ for $X = V^*, VP$, and col . Table 2 presents the numerical H^1 - and L^2 -relative errors of the obtained solutions. We compare the evolution of the loss during training and the relationship between the loss and error in Figure 3.

We observe that all three approximations u_{V^*} , u_{VP} , and u_{col} converge well to the solution, and quantitatively we see via Table 2 that the relative H^1 and L^2 errors are small in each case, around $10^{-3}\%$. Generally, all metrics are comparable between the three obtained solutions, as expected by the theory, due to the presence of only low-frequency modes. In particular, we see that the implementations of \mathcal{L}_{VP} and \mathcal{L}_{col} are almost identical, which is as expected as they can both be interpreted as discretisations of the L^2 -norm of the strong-form residual. The slight differences in metrics may easily arise from the optimisation procedure, rather than the losses themselves.

In the column ‘‘Loss/error correlation’’ of Figure 3 we show the relationship between the square root of the losses and the relative H^{-1} error during training. As expected, in view of Equation (17), the square root of the discretised loss \mathcal{L}_{V^*} is an excellent approximation of the H^{-1} norm of the residual. A similar behaviour is observed with the remaining losses: \mathcal{L}_{VP} and \mathcal{L}_{col} . However, in these cases, we see slight perturbations in this linear relationship, as expected.

4.1.3 Model Problem 2 - Large gradients

Our next model problem is: find $u \in H^1(0, \pi)$ with $u(0) = 0$ that satisfies

$$\int_0^\pi u'(x)v'(x) - 2a^2 \frac{\tanh\left(a\left(x - \frac{\pi}{2}\right)\right)}{\cosh\left(a\left(x - \frac{\pi}{2}\right)\right)^2} v(x) dx + a \operatorname{sech}\left(a\left(\frac{\pi}{2}\right)\right)^2 v(\pi) = 0 \quad (37)$$

for all $v \in H^1(0, \pi)$ with $v(0) = 0$. The exact solution is given by

$$u^*(x) = \tanh\left(a\left(x - \frac{\pi}{2}\right)\right) + \tanh\left(\frac{a\pi}{2}\right) \quad (38)$$

We consider this example as this admits C^∞ solutions for all a , but for a large, a transition region with high gradients develops in the solution. In particular, the forcing term, whilst being a smooth L^2 function, has a large discrepancy between its norm in V^* and L^2 , and thus we expect to see significant differences according to the loss implemented. For our implementation, we take $a = 20$.

As we have a Neumann condition at $x = 0$, for the collocation method we need to include a further term to enforce the constraint. Our implementation for the collocation loss \mathcal{L}_{col} uses equal weights for the interior and boundary terms, i.e.

$$\mathcal{L}_{col}(u) := \left|u'(\pi) - a \operatorname{sech}\left(\frac{a\pi}{2}\right)\right|^2 + \frac{1}{N} \sum_{i=1}^N |L_u(x_i)|^2, \quad (39)$$

where L_u is the strong-form residual. Generally, one could choose to weight the two components of the loss differently, and the choice of weight is an extra parameter which may effect the convergence of the model. An advantage of the weak formulation, however, is that it does not

require such a choice. Whilst methods exist to attempt to estimate optimal weights during training within certain settings [56], we do not consider them in this work. Despite the need to choose an appropriate weight, we observe in Figure 4b that there is very little difference between the H^1 -error evolution using \mathcal{L}_{VP} and \mathcal{L}_{col} , suggesting that, in this example, the choice of weight is unimportant.

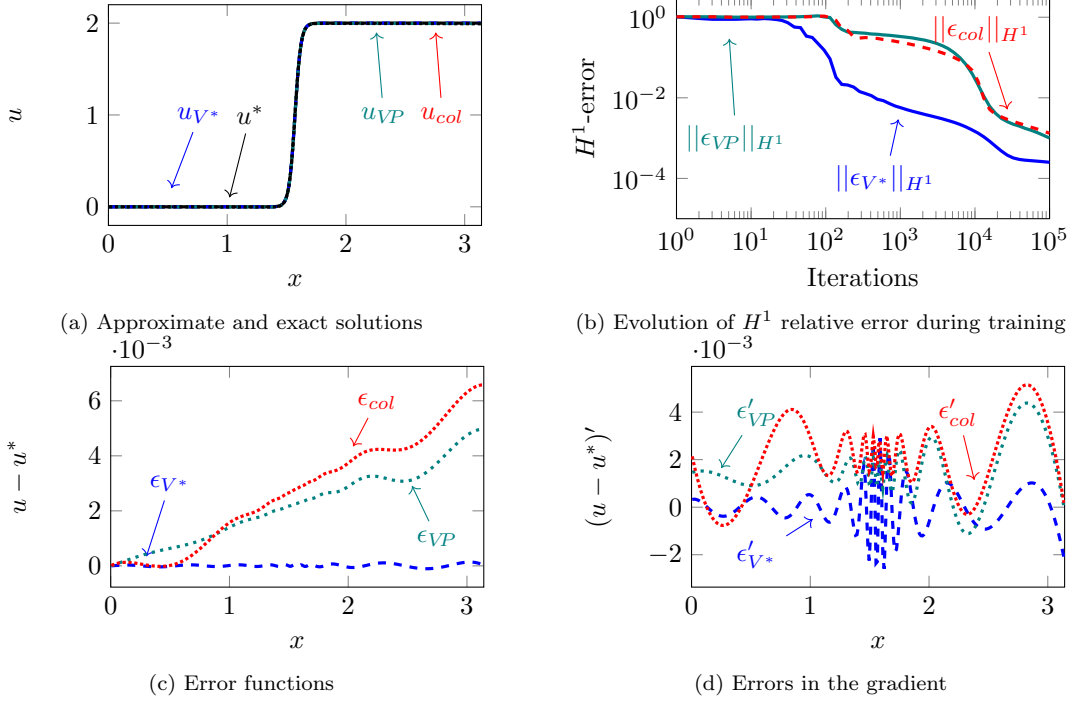


Figure 4: Model Problem 2: Obtained solutions and relative H^1 -error evolution for the three methods

	H1 (%)	L2 (%)
u_{V^*}	0.025	0.004
u_{VP}	0.100	0.184
u_{col}	0.132	0.242

Table 3: Model problem 2: Errors and losses

Figure 4 shows that all three methods converge to the exact solution, and this is seen quantitatively in Table 3. We see that u_{V^*} shows the best performance, both in terms of speed of convergence and the error of the solution at the end of training. Similar to Model Problem 1, we see that \mathcal{L}_{VP} and \mathcal{L}_{col} show similar behaviours in all regards. As before, the correlation between the H^1 -error and the square root of \mathcal{L}_{V^*} is extremely strong, showing a directly proportional relationship between them. This behaviour however is no longer seen when \mathcal{L}_{VP} and \mathcal{L}_{col} are implemented. In particular, we observe that during the initial training, \mathcal{L}_{VP} and \mathcal{L}_{col} decrease by several orders of magnitude before the error itself starts to decrease significantly. Thus, we demonstrate that even in the case of a highly regular problem with C^∞ solution, both \mathcal{L}_{col} and \mathcal{L}_{VP} can fail to be good estimators of the H^1 -error, whilst the DFR

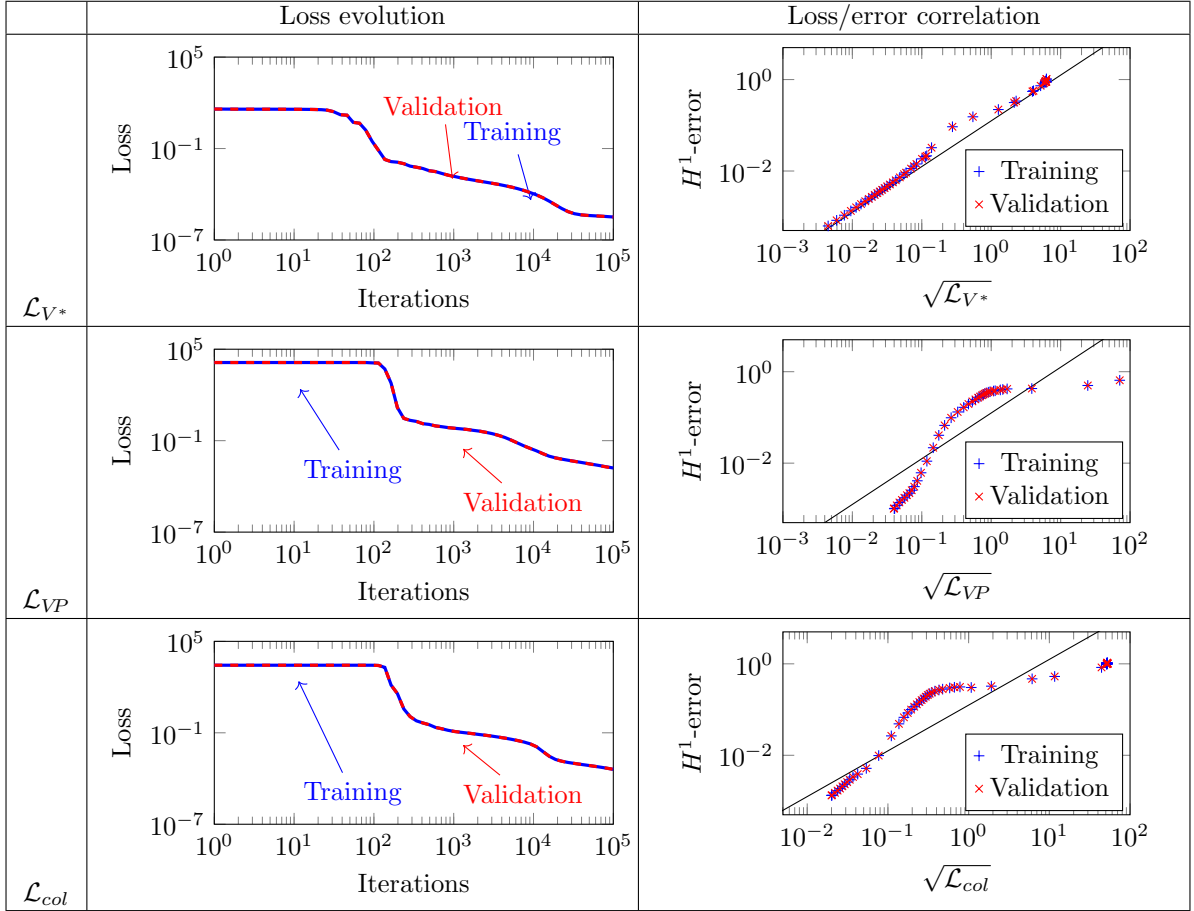


Figure 5: Model Problem 2. Evolution of the loss for the three considered losses on both the training and validation data sets, and the correlation between the loss and the relative H^1 -error during training, with a straight line corresponding to a linear relationship.

method, due to its directly proportional relationship, permits good H^1 -error estimation.

4.1.4 Model Problem 3 - Discontinuous parameters

We next consider an ODE with discontinuous parameters, whose solution is in H^1 but is not C^1 nor H^2 . We take $\Gamma_D = \{0, \pi\}$ and aim to solve

$$\int_0^\pi \sigma(x)u'(x)v'(x) - 4\sin(2x)v(x) dx = 0 \quad (40)$$

for all $v \in H_0^1(0, \pi)$, where

$$\sigma(x) = \begin{cases} 1 & x < \frac{\pi}{2}, \\ 2 & x > \frac{\pi}{2}. \end{cases} \quad (41)$$

The exact solution to this problem is given by

$$u^*(x) = \begin{cases} \sin 2x & x < \frac{\pi}{2}, \\ \frac{1}{2} \sin 2x & x > \frac{\pi}{2}. \end{cases} \quad (42)$$

In particular, u^* admits a jump discontinuity in its gradient at $x = \frac{\pi}{2}$.

Since σ is discontinuous, the strong and weak forms are not equivalent. In particular, as σ is piecewise constant, outside of its single point of discontinuity there is no difference in the (strong) PDEs between

$$(\sigma(x)u'(x))' + 4\sin 2x = 0 \quad (43)$$

and

$$u''(x) + \frac{4}{\sigma(x)} \sin 2x = 0. \quad (44)$$

There is a unique C^1 function which solves (44) on $(0, \pi) \setminus \{\frac{\pi}{2}\}$, given by

$$\tilde{u}(x) = \begin{cases} \sin 2x + \frac{1}{2}x & x < \frac{\pi}{2}, \\ \frac{1}{2} \sin 2x - \frac{1}{2}(x - \pi) & x > \frac{\pi}{2}. \end{cases} \quad (45)$$

We expect the collocation loss \mathcal{L}_{col} to fail, as it is ill-equipped to handle PDEs that lack an equivalent strong formulation. Whilst the discretised VPINN loss \mathcal{L}_{VP} is well defined at any candidate solution, due to the discontinuity in σ , for a general, smooth, trial function u , the series (13) should diverge as the number of basis functions tends to infinity, as the residual cannot generally be expressed as an L^2 function, so we expect the results to not be trustworthy.

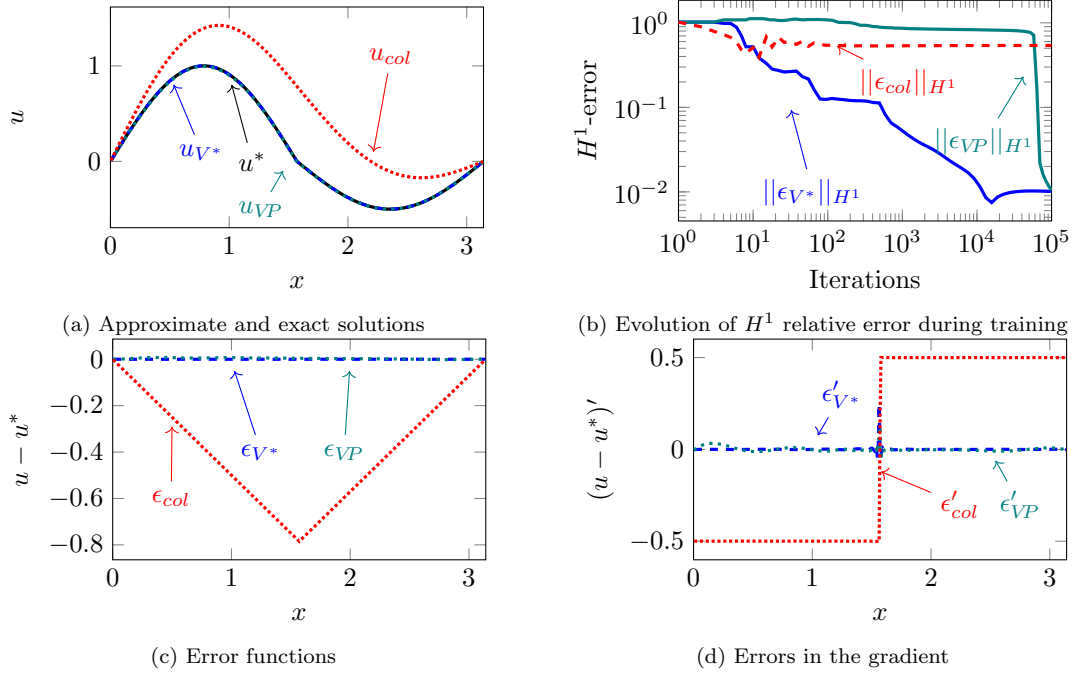


Figure 6: Model Problem 3: Obtained solutions and relative H^1 -error evolution for the three methods

	H1 (%)	L2 (%)
u_{V^*}	1.01	0.0147
u_{VP}	0.988	0.751
u_{col}	54.0	81.1

Table 4: Model Problem 3: Errors and losses

Both qualitatively in Figure 6 and quantitatively in Table 4 we see that u_{V^*} produces a good approximation of the exact solution. Figure 7 shows overfitting during the training of u_{V^*} at around 2×10^4 iterations, as shown by the divergence of the loss on the training and validation sets. At this point, the H^1 relative error stagnates and ceases to decrease significantly. Before overfitting occurs, we observe a perfect linear relationship between the square root of the loss on the training data, however the validation loss remains directly proportional until an uptick corresponding to the region where the validation loss plateaus. This is also the point at which the H^1 -error reaches its minimum, and later begins to increase.

Unsurprisingly, we see that u_{col} approximates (45), rather than u^* , as the loss implemented corresponds precisely to (44), and thus produces a very poor solution. In contrast to the previous examples, as the residual is generally not expressible as a function in L^2 , we observe a very large discrepancy between the behaviour of \mathcal{L}_{V^*} and \mathcal{L}_{VP} .

Furthermore, during the training of u_{VP} , in Figure 7 we see what would appear to be an extreme case of overfitting due to a large discrepancy between the loss evaluated on the training and validation set. However this does not translate into errors, and by comparing the H^1 -error evolution in Figure 6 with the loss evolution in Figure 7, we see that precisely at the point during training where this “overfitting” takes place, around 5×10^5 iterations, the H^1 -error

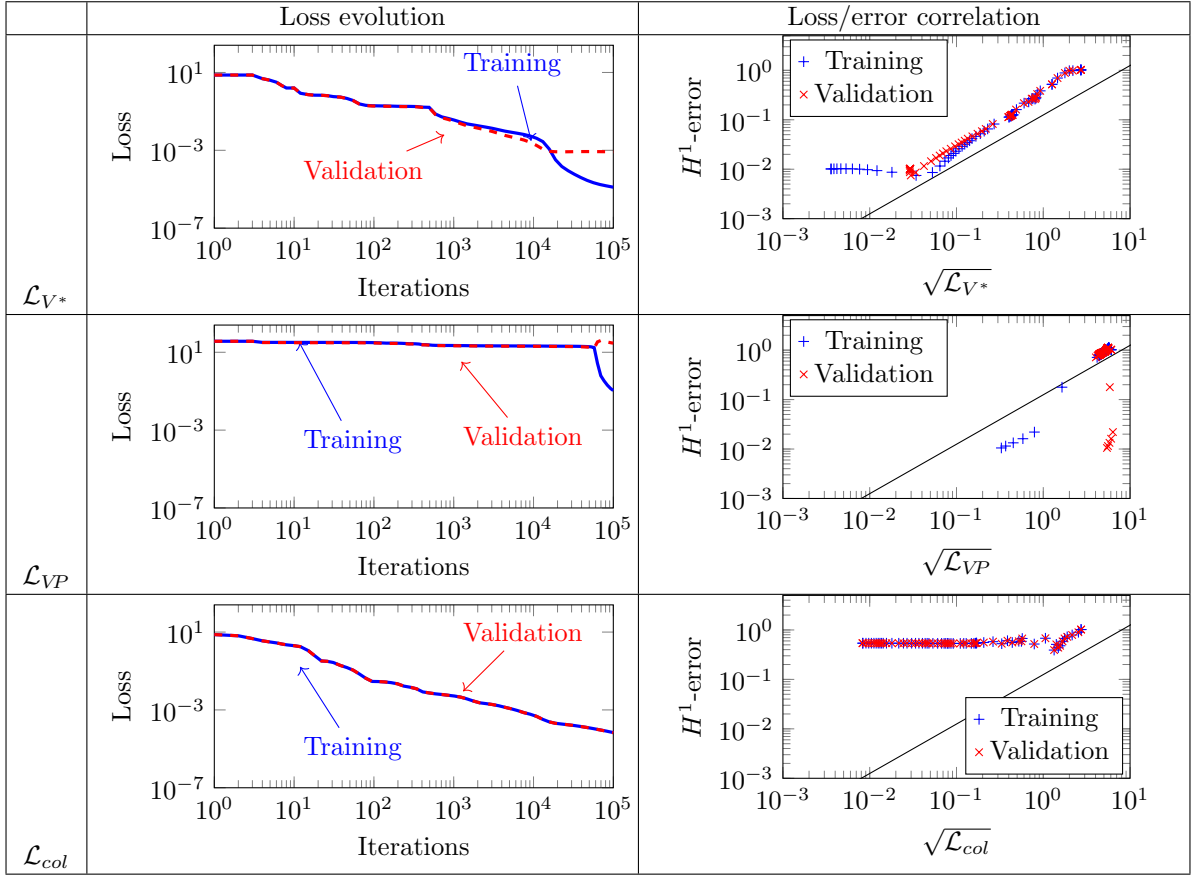


Figure 7: Model Problem 3. Evolution of the loss for the three considered losses on both the training and validation data sets, and the correlation between the loss and the relative H^1 -error during training, with a straight line corresponding to a linear relationship.

begins to drop significantly and we obtain a good approximation to u^* . This is not, however, paradoxical, as we know that the DCT/DST are exact when only low-frequency Fourier modes are present. Thus a large discrepancy between the loss evaluated on a training and validation set, which employ both distinct integration points and distinct cutoff frequencies, implies the presence of high-order Fourier modes in the residual. The smoothing effect of the PDE solution operator, however, mitigates the influence of the high-order modes in the residual on the H^1 -error. Whilst we obtain a good solution, without having the exact solution at hand, it would not be clear if this overfitting is problematic or not without resorting to some other method to attempt to quantify the error. Furthermore, if training had been stopped when this overfitting began to develop, as is traditionally done, we would obtain a solution with relative H^1 -error close to 100%. Due to this, we conclude that, despite \mathcal{L}_{VP} being a well-defined loss for PDEs in weak form, it is inappropriate to use when the weak and strong forms are non-equivalent as one cannot relate the loss on training/validation sets to errors in a clear way, just as \mathcal{L}_{col} would be inappropriate in the same situation. This shows the advantage of employing the DFR method in problems where solutions admit only H^1 regularity, making the H^{-1} -norm of the residual the appropriate loss function to be minimised.

4.2 Further Results

We have seen in Model Problem 3 that a validation set is necessary, as we can identify overfitting via a divergence in the loss evaluated on the training and validation sets. For the following examples, inspired by this, we implement an *EarlyStopping* callback to stop training when the loss evaluated on the validation set does not show improvement during 200 iterations and restores the best NN parameters according to the best obtained value of the loss evaluated on the validation set. In the following, we perform 10^5 iterations, or until the *EarlyStopping* halts training. With only these exceptions, we consider the same architectures and optimisation procedures as before.

4.2.1 Model Problem 4: Point source

We take $V = H_0^1(0, \pi)$ and aim to find $u \in V$ such that

$$\int_0^\pi u'(x)v'(x) dx - v\left(\frac{\pi}{2}\right) = 0 \quad (46)$$

for all $v \in V$. This has a unique solution given by

$$u^*(x) = \frac{\pi}{2} - \left|x - \frac{\pi}{2}\right|. \quad (47)$$

The forcing term, given by a Dirac delta function, is in V^* but not expressible as an L^2 function. In particular, it would be impossible to solve this equation using classical PINN methods.

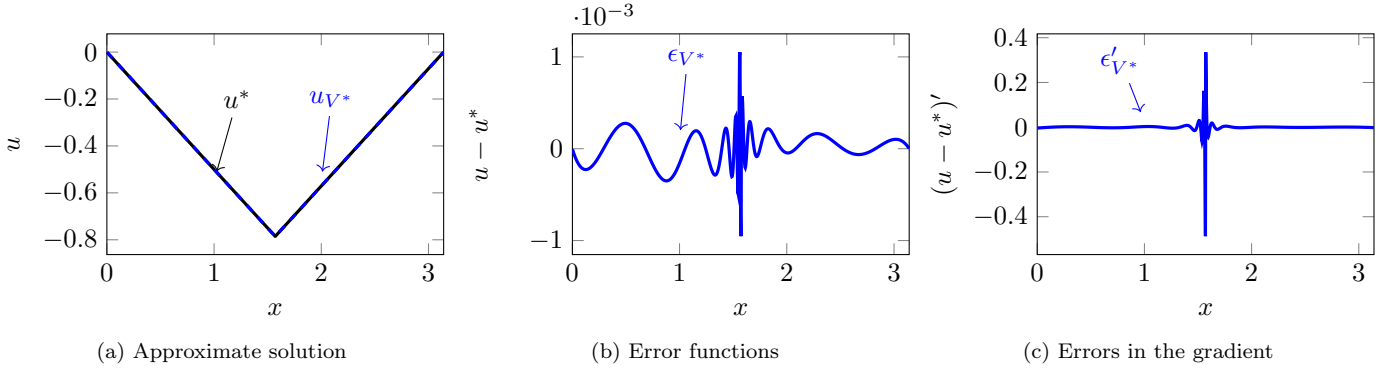


Figure 8: Model Problem 4. Obtained solution and error

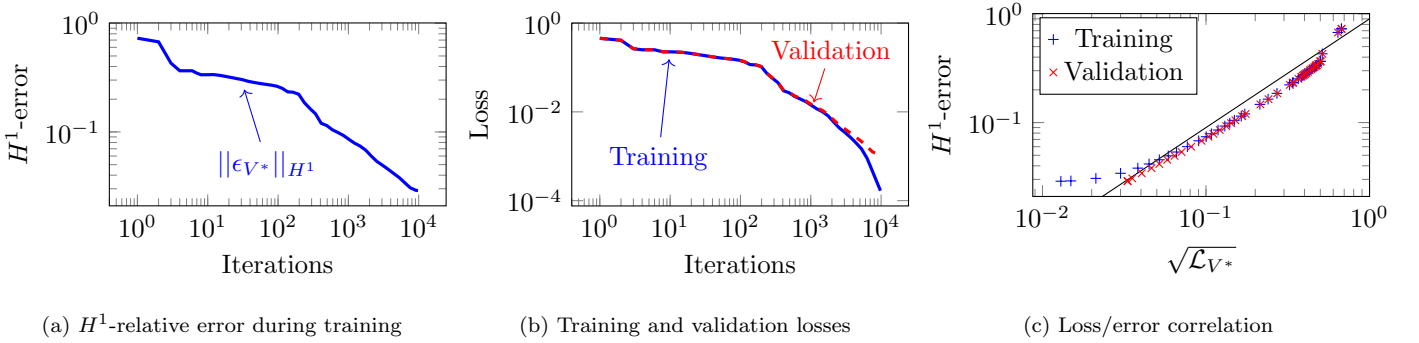


Figure 9: Model Problem 4: Loss and relative H^1 -error during training

We see qualitatively in Figure 8 that we approximate well the exact solution, with absolute pointwise errors remaining of order 10^{-3} . The exact solution is not C^1 , and its derivative admits a jump discontinuity at $x = \frac{\pi}{2}$, thus we observe a Gibbs phenomenon-like error in the gradient of our obtained solution, which is to be expected as we are approximating with smooth trial functions.

With our DFR method, we obtain a relative L^2 error of 0.039%, and relative H^1 -error of 3.60%. *EarlyStopping* halted training at 9610 iterations. Figure 9b shows that that overfitting develops towards the end of the training process. When this overfitting occurs, in Figure 9c we observe that the relative H^1 -error has a sublinear dependency on the training loss; however, the square root of the validation loss and H^1 relative error exhibit a strong linear correlation. In particular, we see that at the point where training was halted by the *EarlyStopping* callback, the H^1 -error had reached a plateau.

4.2.2 Model Problem 5: Nonlinear

We take $V = H_0^1(0, \pi)$, and aim to find $u \in V$ such that

$$\int_0^\pi \left(u'(x) + \frac{1}{2} \sin(u'(x)) \right) v'(x) + f(x)v(x) + u(x)v(x) + u(x)^3 v(x) dx = 0 \quad (48)$$

for all $v \in V$, where f is obtained via the manufactured solution

$$u^*(x) = 5x \left(x - \frac{\pi}{2} \right) \tanh(5(x - \pi)). \quad (49)$$

This problem admits a unique solution as it corresponds to the Euler-Lagrange equation of the strictly convex integral functional given by

$$\mathcal{F}(u) = \int_0^\pi \frac{1}{2} |u'(x)|^2 - \frac{1}{2} \cos(u'(x)) + f(x)u(x) + \frac{1}{2}u(x)^2 + \frac{1}{4}u(x)^4 dx. \quad (50)$$

The ODE is nonlinear, and thus the classical error estimate (17) does not directly apply. However, as commented in Section 3.1.1, for a candidate solution close to the exact solution, the equation can be interpreted as a small perturbation of a linear problem. Consequently, we expect to see a linear regime towards the end of the training.

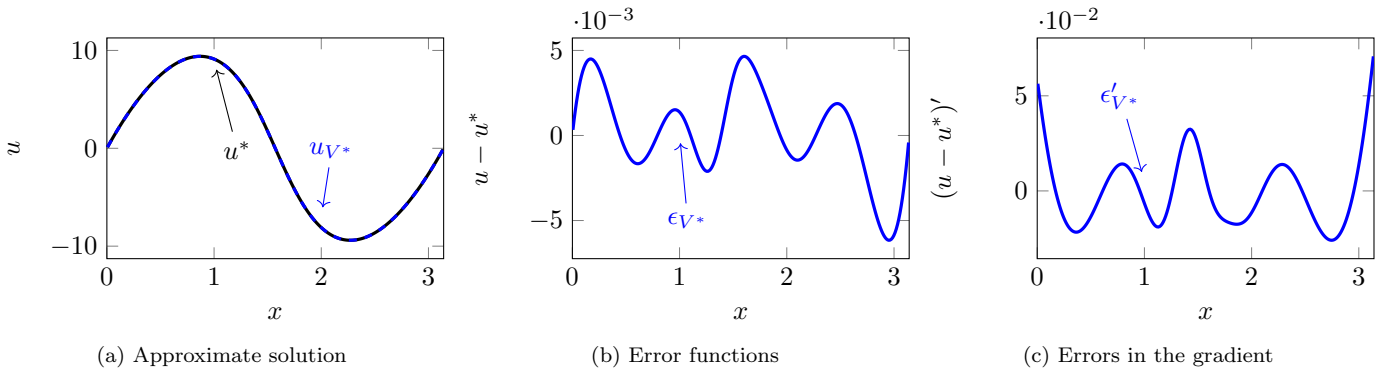


Figure 10: Model Problem 5. Obtained solution and error

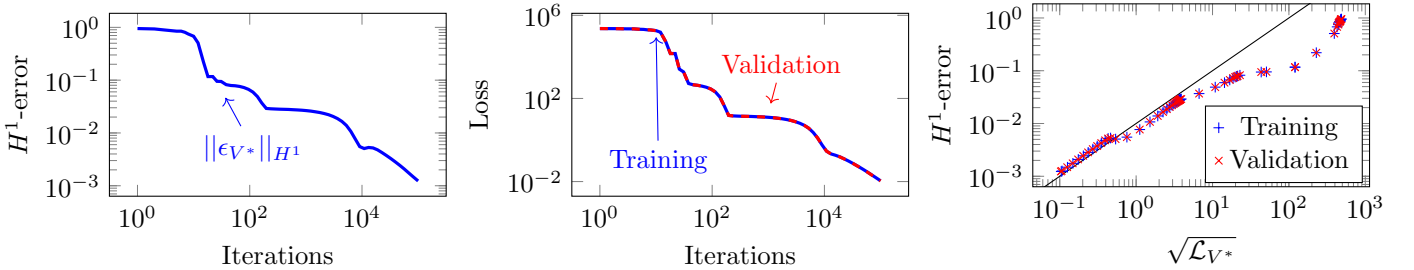


Figure 11: Model Problem 5: Loss and relative H^1 -error during training

After 10^5 iterations, we obtain a relative H^1 -error of 0.117% and relative L^2 error of 0.036%. Figure 10 shows that we have a good approximation of the exact solution, and the pointwise error is of order 10^{-3} , and pointwise error in the gradient is of order 10^{-2} . In Figure 11 we observe that in early training we have a non-linear and slightly non-monotonic relationship between the square root of the loss and H^1 -error; however, once we reach a relative error of around 10^{-2} , we recover a linear regime with proportional dependence between the two metrics in accordance with the theory.

4.2.3 Model Problem 6 - Discontinuous parameters in 2D

Let $\Omega = [0, \pi] \times [0, \pi]$. We take Γ_D to be three edges of $\partial\Omega$ corresponding to $x_1 = 0, \pi$ and $x_2 = 0$, and Γ_N the edge corresponding to $x_2 = \pi$. We aim to find the weak solution $u \in V$ to the equation

$$\int_{\Omega} \sigma(x) \nabla u(x) \cdot \nabla v(x) + f(x)v(x) dx - \int_0^{\pi} v(x_1, \pi) \pi(x_1 - \pi)x_1(1 - \pi) dx_1 = 0 \quad (51)$$

for all $v \in V$, where

$$\sigma(x) = \begin{cases} 2 & \left| x - \left(\frac{\pi}{2}, \frac{\pi}{2} \right) \right| < 1 \\ 1 & \left| x - \left(\frac{\pi}{2}, \frac{\pi}{2} \right) \right| \geq 1 \end{cases}, \quad (52)$$

$$f(x) = \Delta \left((x_1 - \pi)(x_2 - \pi)x_1x_2 \left(1 - \left| x - \left(\frac{\pi}{2}, \frac{\pi}{2} \right) \right|^2 \right) \right).$$

The exact solution is given by

$$u^*(x) = \frac{1}{\sigma(x)} (x_1 - \pi)(x_2 - \pi)x_1x_2 \left(1 - \left| x - \left(\frac{\pi}{2}, \frac{\pi}{2} \right) \right|^2 \right). \quad (53)$$

We use an NN basis of five hidden layers each containing ten neurons and tanh activation function. 200x200 points are used for integration in the training loss, and 274x274 for validation. We have an initial learning rate of 10^{-2} with Adam, and run for 10^5 iterations

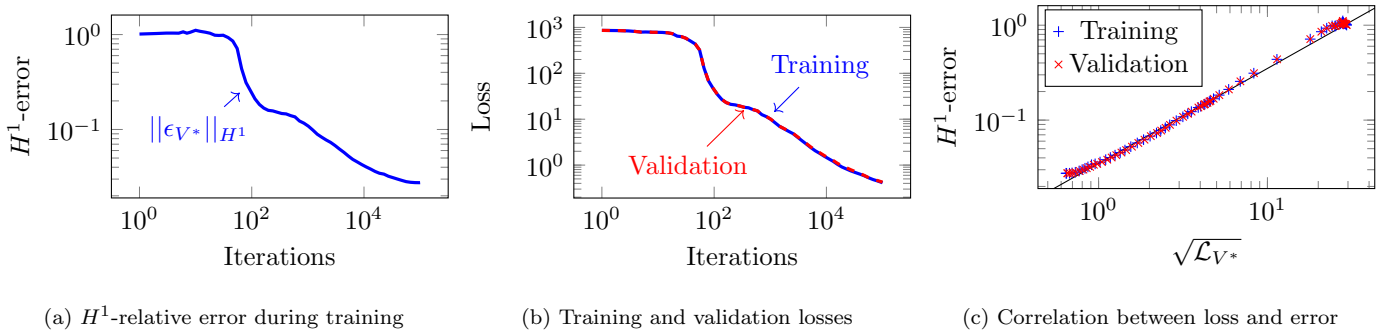


Figure 13: Model Problem 6: Loss and relative H^1 -error during training

Figure 12 shows that the method produces an accurate solution, with absolute pointwise errors remaining of the order 10^{-2} . By numerical integration we observe a relative H^1 -error of 2.7% at the end of training. We observe more significant errors in ∇u_{V^*} near the ring of discontinuity in σ and ∇u^* , which is to be expected as we are approximating discontinuous functions with smooth functions. Outside of this ring-shaped region, however, the approximation of the gradient is generally good. Figure 13a shows a monotonic decay of the H^1 -error during training, and Figure 13b shows that there is no overfitting present. Finally, Figure 13c once again shows a linear relationship between the square root of the loss and the H^1 -error.

5 Conclusions

There are a wide class of PDEs in weak form, using H^1 as their space of test functions, such that the H^1 -error of solutions can be controlled by the H^{-1} -norm of the PDE residual, as outlined in Proposition 3.1. We have developed a framework for implementing the H^{-1} norm

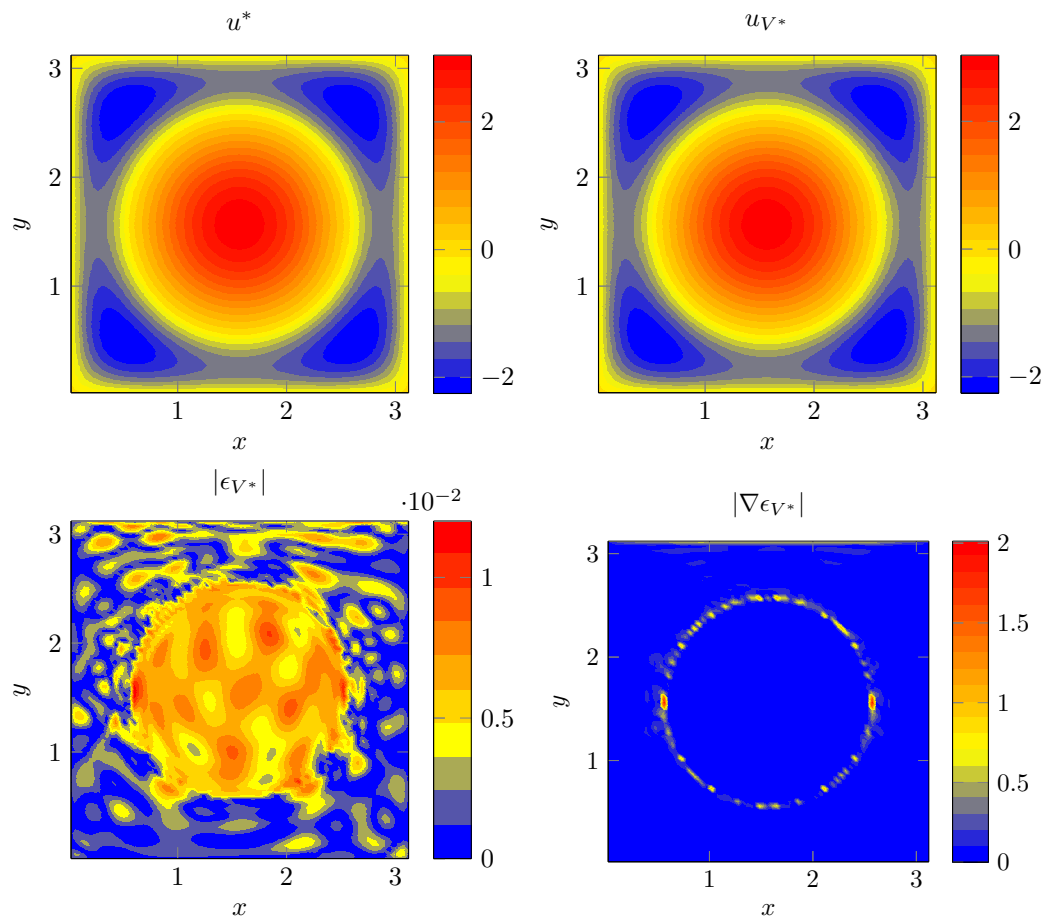


Figure 12: Model Problem 6: Obtained solution and error

as a loss function to solve PDEs using NNs, which is numerically implemented via a spectral decomposition of the residual using DCT/DST to improve efficiency. We have numerically demonstrated that in problems with sufficiently regular solutions, the method is comparable to the collocation and VPINNs methods; however, it shows a strong advantage when solutions lack H^2 regularity, in particular, when the PDE contains discontinuous material parameters or point sources. One may also use the proposed loss as a metric to assess the quality of approximate solutions, even if it is unused for optimisation.

In the absence of overfitting, we observe strong correlations between the training loss and H^1 -error of candidate solutions. Moreover, overfitting is identified in our examples when divergence between the loss evaluated on a training and validation set occurs. This provides a strong advantage over the PINN and VPINN losses, which are inappropriate to use when solutions admit low regularity and may lead to erroneous results.

The DFR has several limitations that open the possibility for future research directions. First, our method suffers from the curse of dimensionality as one must perform DCT/DST in each coordinate direction. It may be possible to overcome this issue in higher dimensions by choosing more appropriate basis functions rather than tensor products of 1D basis sets. Second, our use of DCT/DST to numerically evaluate the dual norm naturally restricts our method to rectangular domains with appropriate boundary conditions on each face/edge. In arbitrary domains, one would need to find alternative basis functions and quadrature rules to numerically approximate the dual norm, which would be dependent on the particular geometry. Finally, our method approximates the H^{-1} norm, which in certain PDEs such as the high-frequency Helmholtz equation, falls short at controlling the energy-norm error. To overcome this, one would need to find an appropriate basis to estimate the correct norm on the dual space via the series expansion (21).

6 Acknowledgements

Jamie M. Taylor is supported by the Basque Government through the BERC 2018-2021 program and by the Spanish State Research Agency through BCAM Severo Ochoa excellence accreditation SEV-2017-0718 and through project PID2020-114189RB-I00 funded by Agencia Estatal de Investigación (PID2020-114189RB-I00 / AEI / 10.13039/501100011033). David Pardo and Ignacio Muga have received funding from: the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 777778 (MATHROCKS). David Pardo has received funding from: the Spanish Ministry of Science and Innovation projects with references TED2021-132783B-I00, PID2019-108111RB-I00 (FEDER/AEI) and PDC2021-121093-I00 (AEI/Next Generation EU), the “BCAM Severo Ochoa” accreditation of excellence (SEV-2017-0718); and the Basque Government through the BERC 2022-2025 program, the three Elkartek projects 3KIA (KK-2020/00049), EXPERTIA (KK-2021/00048), and SIGZE (KK-2021/00095), and the Consolidated Research Group MATHMODE (IT1456-22) given by the Department of Education

References

- [1] AFOURAS, T., CHUNG, J. S., SENIOR, A., VINYALS, O., AND ZISSERMAN, A. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence* (2018).
- [2] ALAM, M., SAMAD, M. D., VIDYARATNE, L., GLANDON, A., AND IFTEKHARUDDIN, K. M. Survey on deep neural networks in speech and vision systems. *Neurocomputing* 417 (2020), 302–321.
- [3] BARRON, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory* 39, 3 (1993), 930–945.

- [4] BAYDIN, A. G., PEARLMUTTER, B. A., RADUL, A. A., AND SISKIND, J. M. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research* 18 (2018), 1–43.
- [5] BERG, J., AND NYSTRÖM, K. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing* 317 (2018), 28–41.
- [6] BERRONE, S., CANUTO, C., AND PINTORE, M. Variational physics informed neural networks: the role of quadratures and test functions. *arXiv preprint arXiv:2109.02035* (2021).
- [7] BERRONE, S., CANUTO, C., AND PINTORE, M. Solving pdes by variational physics-informed neural networks: an a posteriori error analysis. *arXiv preprint arXiv:2205.00786* (2022).
- [8] BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*. Springer, 2010, pp. 177–186.
- [9] BREVIS, I., MUGA, I., AND VAN DER ZEE, K. G. A machine-learning minimal-residual (ml-mres) framework for goal-oriented finite element discretizations. *Computers & Mathematics with Applications* 95 (2021), 186–199.
- [10] BREVIS, I., MUGA, I., AND VAN DER ZEE, K. G. Neural Control of Discrete Weak Formulations: Galerkin, Least-Squares and Minimal-Residual Methods with Quasi-Optimal Weights. *arXiv preprint arXiv:2206.07475* (2022).
- [11] BREZIS, H. *Functional analysis, Sobolev spaces and partial differential equations*. Springer, 2011.
- [12] BRITANAK, V., YIP, P. C., AND RAO, K. R. *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Elsevier, 2010.
- [13] CHEN, T., AND CHEN, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks* 6, 4 (1995), 911–917.
- [14] CIARLET, P. G. *Linear and nonlinear functional analysis with applications*, vol. 130. Siam, 2013.
- [15] CIER, R. J., ROJAS, S., AND CALO, V. M. Automatically adaptive, stabilized finite element method via residual minimization for heterogeneous, anisotropic advection–diffusion–reaction problems. *Computer Methods in Applied Mechanics and Engineering* 385 (2021), 114027.
- [16] CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2, 4 (1989), 303–314.
- [17] DAVIES, E. B. *Spectral theory and differential operators*. Cambridge University Press, 1996. Cambridge Studies in Advanced Mathematics, Series Number 42.
- [18] DE RYCK, T., JAGTAP, A. D., AND MISHRA, S. Error estimates for physics informed neural networks approximating the navier-stokes equations. *arXiv preprint arXiv:2203.09346* (2022).
- [19] DÜSTER, A., PARVIZIAN, J., YANG, Z., AND RANK, E. The finite cell method for three-dimensional problems of solid mechanics. *Computer methods in applied mechanics and engineering* 197, 45-48 (2008), 3768–3782.
- [20] ESTEVA, A., ROBICQUET, A., RAMSUNDAR, B., KULESHOV, V., DEPRISTO, M., CHOU, K., CUI, C., CORRADO, G., THRUN, S., AND DEAN, J. A guide to deep learning in healthcare. *Nature medicine* 25, 1 (2019), 24–29.
- [21] GLOWINSKI, R., AND KUZNETSOV, Y. Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems. *Computer Methods in Applied Mechanics and Engineering* 196, 8 (2007), 1498–1506.

- [22] GOSWAMI, S., YIN, M., YU, Y., AND KARNIADAKIS, G. E. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Computer Methods in Applied Mechanics and Engineering* 391 (2022), 114587.
- [23] GUPTA, A., ANPALAGAN, A., GUAN, L., AND KHWAJA, A. S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* 10 (2021), 100057.
- [24] HORNIK, K. Approximation capabilities of multilayer feedforward networks. *Neural networks* 4, 2 (1991), 251–257.
- [25] HORNIK, K., STINCHCOMBE, M., AND WHITE, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks* 3, 5 (1990), 551–560.
- [26] JAGTAP, A. D., KHARAZMI, E., AND KARNIADAKIS, G. E. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering* 365 (2020), 113028.
- [27] KHARAZMI, E., ZHANG, Z., AND KARNIADAKIS, G. E. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873* (2019).
- [28] KHODAYI-MEHR, R., AND ZAVLANOS, M. Varnet: Variational neural networks for the solution of partial differential equations. In *Learning for Dynamics and Control* (2020), PMLR, pp. 298–307.
- [29] KHODAYI-MEHR, R., AND ZAVLANOS, M. M. Deep learning for robotic mass transport cloaking. *IEEE Transactions on Robotics* 36, 3 (2020), 967–974.
- [30] KIDGER, P., AND LYONS, T. Universal approximation with deep narrow networks. In *Conference on learning theory* (2020), PMLR, pp. 2306–2327.
- [31] KINGMA, D. P., AND BA, J. L. Adam: A method for stochastic optimization.
- [32] LAGARIS, I. E., LIKAS, A., AND FOTIADIS, D. I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks* 9, 5 (1998), 987–1000.
- [33] LARSSON, K., KOLLMANNBERGER, S., RANK, E., AND LARSON, M. G. The finite cell method with least squares stabilized Nitsche boundary conditions. *Computer Methods in Applied Mechanics and Engineering* 393 (2022), 114792.
- [34] LITJENS, G., KOOI, T., BEJNORDI, B. E., SETIO, A. A. A., CIOMPI, F., GHAFOORIAN, M., VAN DER LAAK, J. A., VAN GINNEKEN, B., AND SÁNCHEZ, C. I. A survey on deep learning in medical image analysis. *Medical image analysis* 42 (2017), 60–88.
- [35] LU, L., JIN, P., AND KARNIADAKIS, G. E. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193* (2019).
- [36] LU, L., JIN, P., PANG, G., ZHANG, Z., AND KARNIADAKIS, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence* 3, 3 (2021), 218–229.
- [37] LU, L., MENG, X., MAO, Z., AND KARNIADAKIS, G. E. DeepXDE: A deep learning library for solving differential equations. *SIAM Review* 63, 1 (2021), 208–228.
- [38] MISHRA, S., AND MOLINARO, R. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. *IMA Journal of Numerical Analysis* (2021).
- [39] MITTAL, R., AND IACCARINO, G. Immersed boundary methods. *Annu. Rev. Fluid Mech.* 37 (2005), 239–261.

- [40] PASZYŃSKI, M., GRZESZCZUK, R., PARDO, D., AND DEMKOWICZ, L. Deep learning driven self-adaptive hp finite element method. In *International Conference on Computational Science* (2021), Springer, pp. 114–121.
- [41] PESKIN, C. S. The immersed boundary method. *Acta numerica* 11 (2002), 479–517.
- [42] PRUDHOMME, S., AND ODEN, J. T. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Computer Methods in Applied Mechanics and Engineering* 176, 1-4 (1999), 313–331.
- [43] PURUSHOTHAM, S., MENG, C., CHE, Z., AND LIU, Y. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics* 83 (2018), 112–134.
- [44] RAISSI, M., PERDIKARIS, P., AND KARNIADAKIS, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378 (2019), 686–707.
- [45] RAMIERE, I., ANGOT, P., AND BELLARD, M. A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering* 196, 4-6 (2007), 766–781.
- [46] RIVERA, J. A., TAYLOR, J. M., OMELLA, Á. J., AND PARDO, D. On quadrature rules for solving Partial Differential Equations using Neural Networks. *Computer Methods in Applied Mechanics and Engineering* 393 (2022), 114710.
- [47] RUTHOTTO, L., AND HABER, E. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision* 62, 3 (2020), 352–364.
- [48] SAMANIEGO, E., ANITESCU, C., GOSWAMI, S., NGUYEN-THANH, V. M., GUO, H., HAMDIA, K., ZHUANG, X., AND RABEZUK, T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering* 362 (2020), 112790.
- [49] SCHILLINGER, D., AND RUESS, M. The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. *Archives of Computational Methods in Engineering* 22, 3 (2015), 391–455.
- [50] SHIN, Y. On the Convergence of Physics Informed Neural Networks for Linear Second-Order Elliptic and Parabolic Type PDEs. *Communications in Computational Physics* 28, 5 (2020), 2042–2074.
- [51] SHIN, Y., ZHANG, Z., AND KARNIADAKIS, G. E. Error estimates of residual minimization using neural networks for linear PDEs. *arXiv preprint arXiv:2010.08019* (2020).
- [52] SHORTEN, C., AND KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.
- [53] SIRIGNANO, J., AND SPILIOPOULOS, K. DGM: A deep learning algorithm for solving partial differential equations. *Journal of computational physics* 375 (2018), 1339–1364.
- [54] SLUZALEC, T., GRZESZCZUK, R., ROJAS, S., DZWINEL, W., AND PASZYŃSKI, M. Quasi-optimal hp-finite element refinements towards singularities via deep neural network prediction. *arXiv preprint arXiv:2209.05844* (2022).
- [55] URIARTE, C., PARDO, D., AND OMELLA, Á. J. A Finite Element based Deep Learning solver for parametric PDEs. *Computer Methods in Applied Mechanics and Engineering* 391 (2022), 114562.
- [56] WANG, S., YU, X., AND PERDIKARIS, P. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics* 449 (2022), 110768.

A The Laplacian basis

The following results are classical, with more detailed discussion available, for example, in [11, Chapter 6] or [17]. In particular, Corollary 4.2.3 and Theorems 4.5.1 and 6.3.1. We include this discussion in a relatively self-contained framework for completeness.

Let $\Omega \subset \mathbb{R}^N$ be a bounded domain, with Γ_D, Γ_N disjoint subsets of $\partial\Omega$ such that $\partial\Omega = \overline{\Gamma_D} \cup \overline{\Gamma_N}$. Take V to be the space $V = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$. We consider an orthogonal basis for V given by the eigenvectors of the operator $1 - \Delta$ on V with boundary condition $\frac{\partial u}{\partial \nu} = 0$ on Γ_N , that is, a homogeneous Neumann condition on Γ_N and homogeneous Dirichlet condition on Γ_D .

First, we show that such a basis exists. We first define the solution operator $T : L^2(\Omega) \rightarrow L^2(\Omega)$ to be the operator taking $f \in L^2(\Omega)$ to the unique solution $u \in V$ of

$$\int_{\Omega} \nabla u \cdot \nabla v + uv \, dx = \int_{\Omega} f v \, dx. \quad (54)$$

for all $v \in V$. We remark that this is equivalent to $\langle Tf, v \rangle_{H^1} = \langle f, v \rangle_{L^2}$ for all $v \in V$. T is a symmetric and positive definite linear map: For any $f_1, f_2 \in L^2(\Omega)$, from the weak-formulation (54), as $Tf_1, Tf_2 \in V$, we have that

$$\langle Tf_1, f_2 \rangle_{L^2} = \langle Tf_1, Tf_2 \rangle_{H^1} = \langle f_1, Tf_2 \rangle_{L^2}. \quad (55)$$

Furthermore, by the classical Lax-Migram result, we have that $\|Tf\|_{H^1} \leq \|f\|_{L^2}$. In particular, we have that T is a compact symmetric operator from $L^2(\Omega)$ to itself, and thus, by classical spectral theory, this implies that T admits a decreasing countable sequence of positive eigenvalues that converges monotonically to zero. For notational convenience, we consider their inverses, so that T admits eigenvalues $\lambda_k^{-1} > 0$ where λ_k is a positive, monotonically increasing sequence with $\lambda_k^{-1} \rightarrow \infty$. The eigenvalues have corresponding eigenvectors φ_k , where $(\varphi_k)_{k=1}^{\infty}$ forms an orthonormal basis of $L^2(\Omega)$. The weak formulation (55) implies that for any $v \in V$,

$$\lambda_k^{-1} \langle \varphi_k, v \rangle_{H^1} = \langle T\varphi_k, v \rangle_{H^1} = \langle \varphi_k, v \rangle_{L^2} \quad (56)$$

In particular, as $\lambda_k \neq 0$, the L^2 -orthogonality of the sequence $(\varphi_k)_{k=1}^{\infty}$ also implies H^1 -orthogonality. By taking $v = \varphi_k$ in (56), we also see that $\|\varphi_k\|_{H^1}^2 = \lambda_k$.

We show by contradiction that φ_k also forms a *basis* of V , and not just an orthogonal set. If φ_k were not a basis, there would exist some $v \in V \setminus \{0\}$ such that $\langle \varphi_k, v \rangle_{H^1} = 0$ for all k . In light of (56), we must therefore have that $\langle \varphi_k, v \rangle_{L^2} = 0$ for all k . This contradicts that $(\varphi_k)_{k=1}^{\infty}$ is a basis for $L^2(\Omega)$.

B Estimation via Fast (Co)Sine Transforms

Discrete Sine/Cosine Transforms (DST/DCT) are efficient methods, based on the Fast Fourier Transform (FFT), to decompose a finite input vector of dimension N into N sine or cosine waves with given boundary conditions. There are numerous variations corresponding to different boundary conditions, and within this work we focus on the type-II and type-IV transforms [12, Section 4.2]. We use the notation DST-II, DST-IV to refer to the type-II and type-IV DST, respectively, and DCT-II and DCT-IV to refer to the type-II and type-IV DCT, respectively, which is employed in the summary of basis functions in Table 1.

As the DST/DCT are linear operations between two N -dimensional vector spaces, each may be represented by an $N \times N$ matrix. We represent the type-II and type-IV DST via matrices S^{II} and S^{IV} , respectively, and similarly the type-II and type-IV DCT via C^{II} and C^{IV} . Each matrix is indexed by $k, n = 0, \dots, N - 1$. For an integrable function g , we may approximate

integrals via the following four relationships:

$$\begin{aligned}
\int_0^\pi g(x) \sin(kx) dx &\approx \mathcal{S}_{N,k}^{II}(g) := \sum_{n=0}^{N-1} \frac{\pi}{\sqrt{2N}} S_{k-1,n}^{II} g\left(\frac{2n+1}{2N}\pi\right), \\
\int_0^\pi g(x) \sin\left(\left(k - \frac{1}{2}\right)x\right) dx &\approx \mathcal{S}_{N,k}^{IV}(g) := \sum_{n=0}^{N-1} \frac{\pi}{\sqrt{2N}} S_{k-1,n}^{IV} g\left(\frac{2n+1}{2N}\pi\right), \\
\int_0^\pi g(x) \cos(kx) dx &\approx \mathcal{C}_{N,k}^{II}(g) := \sum_{n=0}^{N-1} \frac{\pi}{\sqrt{2N}} C_{k-1,n}^{II} g\left(\frac{2n+1}{2N}\pi\right), \\
\int_0^\pi g(x) \cos\left(\left(k - \frac{1}{2}\right)x\right) dx &\approx \mathcal{C}_{N,k}^{IV}(g) := \sum_{n=0}^{N-1} \frac{\pi}{\sqrt{2N}} C_{k-1,n}^{IV} g\left(\frac{2n+1}{2N}\pi\right).
\end{aligned} \tag{57}$$

In each case, the approximation corresponds to a mid-point integration rule, that is,

$$\int_0^\pi f(x) dx \approx \frac{\pi}{N} \sum_{n=0}^{N-1} f\left(\frac{2n+1}{2N}\pi\right). \tag{58}$$

The matrices in the transformations are defined by:

$$\begin{aligned}
(S_N^{II})_{kn} &:= \sqrt{\frac{2}{N}} \epsilon_k \sin\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) (k+1)\right), \\
(S_N^{IV})_{kn} &:= \sqrt{\frac{2}{N}} \sin\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) \left(k + \frac{1}{2}\right)\right), \\
(C_N^{II})_{kn} &:= \sqrt{\frac{2}{N}} \epsilon'_k \cos\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) k\right), \\
(C_N^{IV})_{kn} &:= \sqrt{\frac{2}{N}} \cos\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) \left(k + \frac{1}{2}\right)\right),
\end{aligned} \tag{59}$$

for $k, n = 0, \dots, N-1$ and where

$$\begin{aligned}
\epsilon_k &= \begin{cases} 1 & k \neq N-1, \\ \frac{1}{\sqrt{2}} & k = N-1. \end{cases} \\
\epsilon'_k &= \begin{cases} 1 & k \neq 0, \\ \frac{1}{\sqrt{2}} & k = 0. \end{cases}
\end{aligned} \tag{60}$$

For both $X = II$, $X = IV$, the matrices C_N^X and S_N^X are related as follows. Let D denote the diagonal matrix with diagonal entries $D_{kk} = (-1)^k$ for $k = 0, \dots, N-1$, and J denote the matrix which reverses the order of a vector, so that $JY = (y_{N-1}, y_{N-2}, \dots, y_1, y_0)$. Then,

$$S_N^X = J C_N^X D \tag{61}$$

Under this particular normalisation, the corresponding transformation matrices are orthogonal, that is, each of them satisfies $M^T = M^{-1}$.

C Nonlinear equations

We now prove Proposition 3.1. We do so by considering a linearisation, at which point we may invoke results on the linear theory. The reader is directed to [14, Chapter 7] for definitions

and properties related to the differentiability of functions between Banach spaces, however we include below some definitions for completeness.

Given $u, w \in U$, we define the directional derivative $\delta_w \mathcal{R}(u) \in V^*$ via

$$\delta_w \mathcal{R}(u) := \lim_{\tau \rightarrow 0} \frac{\mathcal{R}(u + \tau w) - \mathcal{R}(u)}{\tau}, \quad (62)$$

when it exists. Furthermore, we state that \mathcal{R} is Gateaux differentiable at u if $\delta_w \mathcal{R}(u)$ exists for all $w \in U$, and the map $U \ni w \mapsto \delta_w \mathcal{R}(u) \in V^*$ defines a continuous linear function.

The proof of Proposition 3.1 reduces to two lemmas, corresponding to the upper and lower bounds.

Lemma C.1. *For every $\epsilon > 0$, there exists $\delta_1 > 0$ such that for all $u \in V$ with $\|u - u^*\|_U < \delta_1$,*

$$\|\mathcal{R}(u)\|_{V^*} \geq \frac{\gamma}{1 + \epsilon} \|u - u^*\|_U.$$

Proof. We consider $\|u - u^*\|_U < r_0$. For brevity, we denote $w = u - u^*$. By Taylor's theorem with explicit remainder, we can estimate

$$\begin{aligned} \|\mathcal{R}(u) - \mathcal{R}(u^*) - \delta_w \mathcal{R}(u^*)\|_{V^*} &= \left\| \int_0^1 \delta_w \mathcal{R}(u^* + t(u - u^*)) dt - \delta_w \mathcal{R}(u^*) \right\|_{V^*} \\ &\leq \int_0^1 \|\delta_w \mathcal{R}(u^* + t(u - u^*)) - \delta_w \mathcal{R}(u^*)\|_{V^*} dt \\ &\leq L \int_0^1 \|w\|_U \|u - u^*\|_U dt = L \|u - u^*\|_U^2. \end{aligned} \quad (63)$$

Thus, we may define the remainder $R : V \rightarrow V^*$ via

$$R(u) = \mathcal{R}(u) - \mathcal{R}(u^*) - \delta_w \mathcal{R}(u^*), \quad (64)$$

which satisfies $\|R(u)\|_{V^*} \leq L \|u - u^*\|_U^2$. Via the reverse triangle inequality, and noting that $\mathcal{R}(u^*) = 0$, we estimate

$$\|\mathcal{R}(u)\|_{V^*} = \|\delta_w \mathcal{R}(u^*) - R(u)\|_{V^*} \geq \left| \|\delta_w \mathcal{R}(u^*)\|_{V^*} - \|R(u)\|_{V^*} \right|. \quad (65)$$

As $\delta_w \mathcal{R}(u^*)$ is bounded below, $\|\delta_w \mathcal{R}(u^*)\|_{V^*} \geq \gamma \|u - u^*\|_U$. We observe that if

$$\|u - u^*\|_U < \frac{\gamma \epsilon}{(1 + \epsilon)L},$$

where we interpret the right-hand side as $+\infty$ if $L = 0$. Then

$$L \|u - u^*\|_U^2 \leq \frac{\gamma \epsilon}{1 + \epsilon} \|u - u^*\|_U$$

and thus

$$\|\delta_w \mathcal{R}(u^*)\|_{V^*} - \|R(u)\|_{V^*} \geq \gamma \|w\|_U - \frac{\gamma \epsilon}{(1 + \epsilon)} \|w\|_U = \frac{\gamma}{(1 + \epsilon)} \|u - u^*\|_U. \quad (66)$$

By taking $\delta_1 = \min\left(r_0, \frac{\gamma \epsilon}{(1 + \epsilon)L}\right)$ and combining equations (65) and (66), the result holds. \square

We now turn to the upper bound.

Lemma C.2. *For every $1 > \epsilon > 0$, there exists $\delta_2 > 0$ such that if $\|u - u^*\|_U < \delta_2$, then*

$$\|\mathcal{R}(u)\|_{V^*} \leq \frac{M}{1 - \epsilon} \|u - u^*\|_U. \quad (67)$$

Proof. Again, we proceed by assuming that $\|u - u^*\|_U < r_0$ and invoking Taylor's theorem. Taking the remainder R and $w = u - u^*$ as before, we observe that

$$\begin{aligned} \|\mathcal{R}(u)\|_{V^*} &= \|\delta_w \mathcal{R}(u^*) + R(u)\|_{V^*} \\ &\leq \|\delta_w \mathcal{R}(u^*)\|_{V^*} + \|R(u)\|_{V^*} \\ &\leq M\|u - u^*\|_U + L\|u - u^*\|_U^2. \end{aligned} \tag{68}$$

Thus, if $\|u - u^*\|_U < \frac{M\epsilon}{L(1-\epsilon)}$, interpreting the right-hand side of the inequality as $+\infty$ if $L = 0$, we have that $L\|u - u^*\|_U^2 \leq \frac{M\epsilon}{(1-\epsilon)}\|u - u^*\|_U$ and

$$\|\mathcal{R}(u)\|_{V^*} \leq \left(M + \frac{M\epsilon}{(1-\epsilon)} \right) \|u - u^*\|_{V^*} = \frac{M}{1-\epsilon} \|u - u^*\|_{V^*}. \tag{69}$$

By taking $\delta_2 = \min\left(r_0, \frac{M\epsilon}{(1-\epsilon)L}\right)$ we complete the proof. \square