



**HAL**  
open science

# Enriching continuous Lagrange finite element approximation spaces using neural networks

Hélène Barucq, Michel Duprez, Florian Faucher, Emmanuel Franck, Frédérique Lecourtier, Vanessa Lleras, Victor Michel-Dansac, Nicolas Victorion

## ► To cite this version:

Hélène Barucq, Michel Duprez, Florian Faucher, Emmanuel Franck, Frédérique Lecourtier, et al.. Enriching continuous Lagrange finite element approximation spaces using neural networks. 2025. <hal-04935072v4>

**HAL Id: hal-04935072**

**<https://hal.science/hal-04935072v4>**

Preprint submitted on 11 Feb 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# Enriching continuous Lagrange finite element approximation spaces using neural networks

Hélène Barucq<sup>1</sup>, Michel Duprez<sup>2</sup>, Florian Faucher<sup>1</sup>, Emmanuel Franck<sup>3</sup>, Frédérique Lecourtier<sup>2</sup>,  
Vanessa Lleras<sup>4</sup>, Victor Michel-Dansac<sup>3</sup>, and Nicolas Victorion<sup>1</sup>

<sup>1</sup>Project-Team Makutu, Inria, University of Pau and Pays de l'Adour, TotalEnergies, CNRS UMR  
5142, Pau, France

<sup>2</sup>Université de Strasbourg, CNRS, Inria, ICube, F-67000, Strasbourg, France

<sup>3</sup>Université de Strasbourg, CNRS, Inria, IRMA, F-67000, Strasbourg, France

<sup>4</sup>IMAG, University of Montpellier, CNRS UMR 5149, Montpellier, France

February 11, 2026

## Abstract

In this work, we present a study combining two approaches in the context of solving PDEs: the continuous finite element method (FEM) and more recent techniques based on neural networks. In recent years, physics-informed neural networks (PINNs) have become particularly interesting for rapidly solving PDEs, especially in high dimensions. However, their lack of accuracy can be a significant drawback in this context, hence the interest in combining them with FEM, for which error estimates are already known. The complete pipeline proposed here consists in modifying the classical FEM approximation spaces by taking information from a prior, chosen as the prediction of a neural network. On the one hand, this combination improves and certifies the prediction of neural networks, to obtain a fast and accurate solution. On the other hand, error estimates are proven, showing that such strategies outperform classical ones by a factor that depends only on the quality of the prior. We validate our approach with numerical results performed on parametric problems with 1D, 2D and 3D geometries. These experiments demonstrate that to achieve a given accuracy, a coarser mesh can be used with our enriched FEM compared to the standard FEM, leading to reduced computational time, particularly for parametric problems.

## Contents

<b>1</b>	<b>Continuous finite element method</b>	<b>5</b>
<b>2</b>	<b>Enriching the finite element method with additive priors</b>	<b>6</b>
2.1	Construction of the modified problem	7
2.2	Convergence analysis	7
<b>3</b>	<b>Enriching the finite element method with multiplicative priors</b>	<b>8</b>
3.1	Construction of the modified problem	9
3.2	Convergence analysis	9
<b>4</b>	<b>Comparison of the two enriched methods</b>	<b>11</b>
<b>5</b>	<b>Prior construction using parametric PINNs</b>	<b>14</b>
5.1	Physics-Informed Neural Networks for parametric PDEs	14
5.2	Improving PINN training and prediction	16
5.2.1	Exact imposition of boundary conditions	16
5.2.2	Sobolev training for PINNs	16
5.2.3	Overcoming the spectral bias	16

<b>6</b>	<b>Implementation details</b>	<b>17</b>
6.1	Using PINN prediction effectively	17
6.2	Imposing boundary conditions	18
6.2.1	Additive approach	18
6.2.2	Multiplicative approach	18
<b>7</b>	<b>Numerical results</b>	<b>19</b>
7.1	Setup of the numerical experiments	19
7.1.1	Error estimates	20
7.1.2	Gains achieved with the enriched bases	20
7.2	1D Poisson problem	21
7.2.1	Construction of the two priors	21
7.2.2	Error estimates — with the PINN prior	22
7.2.3	Derivatives — with both priors	24
7.2.4	Gains achieved with the additive and multiplicative approaches – with both priors	25
7.2.5	Influence of the PINN initialization	26
7.3	1D general elliptic system and convection-dominated regime	27
7.3.1	Error estimates	27
7.3.2	Comparison of different approaches	28
7.3.3	Gains achieved with the additive and the multiplicative approaches	30
7.4	1D non-smooth transmission problem	30
7.4.1	Construction of the different priors	31
7.4.2	Error estimates — with the three non-parametric priors	32
7.4.3	Derivatives — with the three non-parametric priors	33
7.4.4	Gains achieved with the additive approach – with the parametric enriched PINN prior	34
7.5	2D Poisson problem in a square domain	34
7.5.1	Low-frequency case	34
7.5.2	Low-frequency case — Sobolev training	39
7.5.3	Low-frequency case — Boundary loss training	40
7.5.4	High-frequency case	41
7.6	2D anisotropic elliptic problem on a square	44
7.6.1	Error estimates	45
7.6.2	Comparison of different approaches	45
7.6.3	Gains achieved with the additive approach	45
7.7	2D Poisson problem on an annulus, with mixed boundary conditions	46
7.7.1	Error estimates	47
7.7.2	Comparison of different approaches	47
7.7.3	Gains achieved with the additive approach	48
7.8	3D Poisson problem in a cube domain	49
7.8.1	Error estimates	50
7.8.2	Computation times of the different methods	50
7.8.3	Gains achieved with the additive approach	52
7.8.4	Influence of prior quality on the additive approach	52
<b>8</b>	<b>Conclusion and future work</b>	<b>52</b>
<b>9</b>	<b>Acknowledgements</b>	<b>53</b>
<b>A</b>	<b>Notations and definitions</b>	<b>57</b>

## Introduction

The finite element method (FEM, e.g., [Cia02, EG04, BS08]) is widely used for computing accurate solutions of complex PDEs for which there is no analytic solution. It begins with a mesh as a subdivision of the computational domain into elements, typically simplexes or quadrilaterals/hexahedra. Then the numerical solution is constructed from degrees of freedom (dofs) that are coefficients of the finite element discretization defined by basis functions related to the elements.

Standard FEM performs poorly when the solution has strong local variations (like sharp gradients near cracks, re-entrant corners, material interfaces, etc). Then refining the mesh and/or increasing the order of approximation can help, but the method becomes very (sometimes too) expensive.

This has motivated intensive research on new finite element methods to combine accuracy and lower computational burden. For instance, the Generalized Finite Element Method (GFEM, [SCB01, FB10]) enriches the classical finite element approximation space with special functions that capture local solution features (singularities, oscillations, boundary layers, etc). Basically, GFEM belongs to the family of Partition of Unity Methods [BM97]. The Extended Finite Element Method (XFEM) is a special case of GFEM, developed mainly for fracture mechanics. It was popularized by Belytschko and co-workers [SMMB00], who suggested to enrich the finite element space with discontinuous functions and/or singularity functions, without re-meshing. Therefore, both GFEM and XFEM rely on the enrichment of the finite element spaces to treat local structures in order to increase the accuracy of the solution, [BGV09, BBO04]. The approach we propose has the same idea, but a different conception as it uses a (global) prior of the solution (on the entire domain), and the FEM is used to correct it; this prior information further allows the use of a low-order FEM. Furthermore, our application focuses on a parametric problem, with a prior learnt from a set of parameter values. In [CLC<sup>+</sup>16], the authors deal with parametric problems by enriching finite element spaces additively with functions determined by Proper Generalized Decomposition (PGD). However, PGD requires a separability assumption on the parametric dependence of the solution, which is not necessary in our approach. More generally, XFEM/GFEM is sensitive to the choice of functions for enrichment, which must be chosen problem-dependent for best efficiency, while NN-based priors are more adapted to parametric problems. Another way to improve the accuracy of FEM solutions is through post-processing superconvergence (e.g., [Wah95] and the references therein). However, their effectiveness is often problem-specific and depends on particular properties of the PDE, mesh, and element type. Moreover, they are not particularly suited for parametric problems.

One approach to lowering the cost further consists of reducing the size of the discrete system. This can be done by decreasing the number of degrees of freedom as in the Trefftz method (e.g., [HMP12, MP17, IGMS22]) or the hybridizable discontinuous Galerkin (HDG) method (e.g., [CDG08, HPS17, PFB24]), along with hybrid high-order (HHO) approximations [ES24], which solve a global system associated only with the degrees of freedom on the skeleton of the mesh. It is then worth noting that these approaches are much easier to implement in the stationary case. In the same vein, we can mention reduced-order methods, which focus more on the number of basis functions used to construct the approximate solution. The idea is to construct a low-dimensional subspace from high-fidelity simulations and then solve the PDE projected onto this subspace. This approach is very interesting when the parametric family or the dominant structure of the solutions is known. Examples include the snapshot method or POD (proper orthogonal decomposition, see [BHL93, Rav00]) and the reduced basis method [PR07, PRV<sup>+</sup>01]. Avoiding mesh refinement is a significant advantage, making it possible to control often exorbitant computational costs without any special preprocessing. This justifies other approaches that do not use meshes. Meshless methods have been investigated in the last decades, and isogeometric analysis has been employed, see e.g. [BBDVC<sup>+</sup>06, HCB05, FBCD22].

In recent years, learning-based alternatives have emerged, such as Physics-Informed Neural Networks (PINNs, e.g., [RPK19]) or the Deep Ritz method [EY18]. Similar methods were already present in the 1990s, see for instance [LK90, MF94, LLF98], but the advances in deep learning and computational power have significantly boosted their applicability in recent years. The idea is to approximate the solution of the PDE under consideration using a neural network. If this solution lives in some function space  $H$ , this amounts to projecting it onto a finite-dimensional subset of  $H$  (e.g., a submanifold) defined by the parametrization inherent to the neural network. We stress that, conversely, FEM projects the solution onto a finite-dimensional linear subspace of  $H$ . The neural network is then trained by minimizing a loss function, taking the underlying physics into account. Unlike neural networks trained with more conventional data-driven loss functions, these methods share similarities with traditional solvers: they require the same inputs, namely the PDE, physical parameters, boundary, and initial conditions. In addition, the training phase requires approximating the PDE solution in a discrete set of points of the space domain. Thus, these approaches have some advantages, notably their relative dimension-insensitivity and their mesh-free quality. Indeed, PINNs do not use a mesh, but rather require sampling points in the domain, as it is done in meshless methods (see for instance [FBCD22]). But on complex geometries, sampling can be challenging; however, it is very often easier than meshing. Further refinements have been proposed to improve PINNs. For instance, we refer to the review paper [CDCG<sup>+</sup>22] for some analysis on PINNs, to [JKK20] for a conservative version, to [KZK21] for PINNs using variational formulas, or to [DRMM24] for PINNs using weak formulations to improve discontinuity handling. Since they do not require data (in the form of reference solutions), they are particularly well-suited to high-dimensional problems on complex domains. However, at present, these learning-based techniques are not competitive with classical finite element methods (see [GKLS24]), mainly because network-based methods lack precision and convergence guarantees, see [SKPP24] for a comparison between PINNs and FEM. While FEM has a better error/computation time ratio for a single resolution, PINNs are more advantageous for

parametric systems where a multitude of resolutions is needed.

This paper aims to propose a new method that combines learning-based and finite element methods applied on coarse meshes. The general idea is to assume that some so-called “prior” information is available about the solution of the PDE. It can be thought of as a function that approximates the solution, or a family of functions that approximate the solution for different parameters. This prior then modifies the finite element approximation space, which is finally used to compute the solution of the PDE. More precisely, we use a PINN to compute either one offline solution in the non-parametric case, or a parametric family of offline solutions in the parametric case. This is followed by calculating an online solution using coarse finite elements, with the PINN solution used to modify the finite element approximation space; in the parametric case, this is done for a single parameter. The result is a method capable of rapidly predicting a PDE solution while guaranteeing convergence properties, thanks to the FEM framework. Finite element resolution improves the prediction while remaining cheap as it is performed on a coarse mesh, benefiting from the network prediction. This paper proposes two ways to enrich the FEM. In both cases, the finite element error will be exhibited as a function of the network error with respect to the true solution. These corrections will be called “additive” and “multiplicative” depending on how the prediction is incorporated in the FEM spaces. The error estimates, resulting from these two enriched approaches, are mainly meant to show that enriching the space lets us get the same convergence orders as the standard approach. In addition, this study helps guide choices about prior learning by showing that the higher the approximation orders, the more high-order derivatives are involved. The manuscript does not aim to study the gain constants obtained in these error estimates, which depend on the (hard to quantify) prediction error associated with neural networks.

Work has already been undertaken to combine numerical methods and the use of a prior, be it obtained through a neural network prediction or by some other means. Starting with the enhancement of FEM-related methods, in the FEM itself, the approximation space can also be enriched to ensure stability, see for instance the introduction of bubble functions in mixed problems (see e.g. [EG04]). In  $\varphi$ -FEM, developed in [DL20] (see also [DLL22, CDL<sup>+</sup>23, DLLV23, DLL23] for other contexts), the FEM prior is a level-set function used to localize the domain boundary. In [CLC<sup>+</sup>16], the authors deal with parametric problems by additively enriching finite element spaces with functions determined by Proper Generalized Decomposition (PGD). However, PGD requires a separability assumption on the parametric dependence of the solution. Combining FEM and neural networks, in [FST<sup>+</sup>24], the authors solve a time-dependent PDE by splitting the problem into a spatial resolution (handled by a neural network) and a temporal one (handled by FEM). Other works have also explored related ideas, such as [BLM24] where neural networks are interpolated onto FEM spaces, [MJLR24] where a coarse FEM is augmented with fine information from a neural network, or [WLZ25] where GFEM is enriched with neural networks. It is also possible to include the FEM shape functions in PINNs, as proposed in [vDSG25, XLBJ25]. Concerning other numerical methods, in [BMP<sup>+</sup>19, AFH<sup>+</sup>25], the authors initialize Newton’s algorithm, used when solving nonlinear equations, with an initial guess derived from the prediction of a neural network. Such a prediction can also be used as a prior for discontinuous Galerkin methods (see [FMDN24]). In the finite difference context, the authors of [XPYZ25] propose to replace automatic differentiation with finite difference discretization when possible.

In this paper, we consider general parametric linear elliptic differential equations defined on a domain  $\Omega \subset \mathbb{R}^d$  with  $d$  space dimensions, with a smooth boundary  $\partial\Omega$  (for instance a Lipschitz polytope). Consider a parameter space  $\mathcal{M} = \{\boldsymbol{\mu} = (\mu_1, \dots, \mu_p) \in \mathbb{R}^p\}$ . The typical problem of interest is, for one or several  $\boldsymbol{\mu} \in \mathcal{M}$ , to find  $u : \Omega \rightarrow \mathbb{R}$  such that

$$\mathcal{L}(u; \boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}, \boldsymbol{\mu}), \quad (0.1)$$

with  $\boldsymbol{x} = (x_1, \dots, x_d) \in \Omega$  the space variable, and where  $\mathcal{L}$  is the parametric differential operator defined by

$$\mathcal{L}(\cdot; \boldsymbol{x}, \boldsymbol{\mu}) : u \mapsto R(\boldsymbol{x}, \boldsymbol{\mu})u + C(\boldsymbol{x}, \boldsymbol{\mu}) \cdot \nabla u - \frac{1}{\text{Pe}} \nabla \cdot (D(\boldsymbol{x}, \boldsymbol{\mu}) \nabla u), \quad (0.2)$$

with  $f(\boldsymbol{x}, \boldsymbol{\mu}) \in L^2(\Omega)$  the source term,  $R(\boldsymbol{x}, \boldsymbol{\mu}) \in L^\infty(\Omega)$  the reaction coefficient,  $C(\boldsymbol{x}, \boldsymbol{\mu}) \in W^{1,\infty}(\Omega)^d$  the convection coefficient,  $D(\boldsymbol{x}, \boldsymbol{\mu}) \in (W^{1,\infty}(\Omega))^{d \times d}$  the (symmetric and positive definite) diffusion matrix, and  $\text{Pe} \in \mathbb{R}_+^*$  the Péclet number representing the ratio between convection and diffusion. We suppose that  $\mathcal{L}$  is coercive. The differential operator is considered with Dirichlet, Neumann or Robin boundary conditions, which can also depend on  $\boldsymbol{\mu}$ .

Figure 1 presents the pipeline of our enriched method, described above.

The manuscript is organized as follows: in Section 1, we recall the classical finite element method applied to our problem, introducing the notations needed in the following sections. In Sections 2 and 3, we present the two proposed enrichments and prove error estimates. Both approaches rely on modifying the functions of the FEM approximation space, using information from prior knowledge of the solution. This prior is first introduced in an additive way, and then in a multiplicative way. Both approaches are compared in Section 4. Section 5 is devoted to the construction of the prior, justifying the use of PINNs and recalling methods for improving their efficiency. In Section 6, we give details on the implementation. Numerical simulations conclude this manuscript in Section 7 and show that the proposed methods can significantly reduce the computational cost of solving parametric problems. A short conclusion, as well as plans for future work, are given in Section 8. Appendix A reviews the notations used throughout the manuscript.

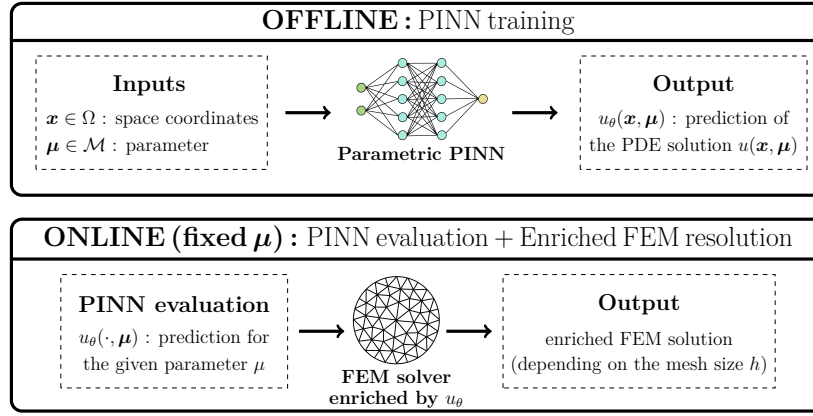


Figure 1: Pipeline of the enriched method. Top: offline phase (PINN training). Bottom: online phase (PINN evaluation + Enriched FEM resolution).

## 1 Continuous finite element method

The goal of this section is to recall the classical FEM, and to introduce the notation that will be used throughout the paper. Recall that, in the online step, the goal is to perform a coarse finite element resolution of the PDE for a single parameter  $\boldsymbol{\mu}$ . Therefore, in this section and the next two concerning error estimates, we work a fixed  $\boldsymbol{\mu}$ , and we omit the dependence on  $\boldsymbol{\mu}$  for conciseness. To solve the problem (0.1) under consideration for a fixed parameter  $\boldsymbol{\mu}$  with homogeneous Dirichlet boundary conditions using the continuous FEM, we rewrite it as the following variational problem:

$$\text{Find } u \in V^0 \text{ such that, } \forall v \in V^0, a(u, v) = l(v), \quad (1.1)$$

where  $V^0 = H_0^1(\Omega)$ , and where the bilinear form  $a$  is given by

$$a(u, v) = \frac{1}{\text{Pe}} \int_{\Omega} D(\mathbf{x}) \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x} + \int_{\Omega} R(\mathbf{x}) u(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} + \int_{\Omega} v(\mathbf{x}) C(\mathbf{x}) \cdot \nabla u(\mathbf{x}) \, d\mathbf{x},$$

while the linear form  $l$  reads

$$l(v) = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x}.$$

**Remark 1.** Note that since  $a$  is continuous on  $V^0 \times V^0$  and coercive and  $l$  is continuous on  $V^0$ , the existence and uniqueness of the solution  $u$  are ensured by the Lax-Milgram theorem.

Let  $\mathcal{T}_h$  be a mesh of the domain  $\Omega$  composed of simplexes, where  $h$  denotes the characteristic size of the mesh, i.e. the biggest diameter of the simplexes. We suppose that  $\mathcal{T}_h$  satisfies the Ciarlet condition (see e.g. [EG04]) and that its boundary is exactly  $\partial\Omega$ . Consider  $V_h^0 \subset V_h \subset V = H^1(\Omega)$  the two continuous Lagrange finite elements spaces of degree  $k \geq 1$  defined by

$$V_h = \{v_h \in C^0(\Omega), \forall K \in \mathcal{T}_h, v_h|_K \in \mathbb{P}_k\}, \quad (1.2)$$

and

$$V_h^0 = \{v_h \in C^0(\Omega), \forall K \in \mathcal{T}_h, v_h|_K \in \mathbb{P}_k, v_h|_{\partial\Omega} = 0\},$$

with  $\mathbb{P}_k$  the space of polynomials with real coefficients of degree at most  $k$ . The solution to (1.1) will be approximated by the solution  $u_h$  to

$$\text{Find } u_h \in V_h^0 \text{ such that, } \forall v_h \in V_h^0, a(u_h, v_h) = l(v_h). \quad (1.3)$$

Let us now introduce some results used in the remainder of the paper. We first define the Lagrange interpolation operator by

$$\mathcal{I}_h : C^0(\Omega) \ni v \mapsto \sum_{i=1}^{N_{\text{dofs}}} v(\mathbf{x}^{(i)}) \psi_i \in V_h, \quad (1.4)$$

with  $(\mathbf{x}^{(i)})_{i \in \{1, \dots, N_{\text{dofs}}\}}$  the  $N_{\text{dofs}}$  degrees of freedom (dofs) associated to the mesh, and  $(\psi_i)_{i \in \{1, \dots, N_{\text{dofs}}\}}$  the associated Lagrange shape functions of degree  $k$ . These Lagrange shape functions are the unique continuous piecewise polynomials of total degree at most  $k$  satisfying the interpolation property

$$\psi_i(\mathbf{x}^{(j)}) = \delta_{ij}, \quad \forall i, j \in \{1, \dots, N_{\text{dofs}}\},$$

where  $\delta_{ij}$  is the Kronecker symbol.

**Remark 2.** In the whole manuscript, for a Sobolev space  $H$ , the notation  $|\cdot|_H$  and  $\|\cdot\|_H$  will represent respectively the semi-norm and the norm in  $H$ . Namely, for  $v \in H^1(\Omega)$ , we set

$$|v|_{H^1(\Omega)}^2 = \int_{\Omega} |\nabla v(\mathbf{x})|^2 d\mathbf{x} \quad \text{and} \quad \|v\|_{H^1(\Omega)}^2 = \int_{\Omega} |v(\mathbf{x})|^2 d\mathbf{x} + \int_{\Omega} |\nabla v(\mathbf{x})|^2 d\mathbf{x}.$$

The following result gives a bound of the interpolation error:

**Theorem 3** (see e.g. [EG04]). *There exists  $C_q > 0$  such that for all  $v \in H^{q+1}(\Omega)$  and  $1 \leq q \leq k$ ,*

$$\|v - \mathcal{I}_h v\|_{H^1} \leq C_q h^q |v|_{H^{q+1}}.$$

The next estimate is associated to the elliptic regularity:

**Theorem 4** (see e.g. [Eva22, Theorem 4, p. 317]). *Suppose that the boundary  $\partial\Omega$  is  $\mathcal{C}^2$ ,  $R \in L^\infty(\Omega)$ ,  $C \in W^{1,\infty}(\Omega)^d$  and  $D \in (W^{1,\infty}(\Omega))^{d \times d}$  in the definition of  $\mathcal{L}$  given in (0.2). There exists  $C_e > 0$  such that for all  $\xi \in L^2(\Omega)$  and all associated weak solution  $w \in H_0^1(\Omega)$  to*

$$\mathcal{L}^* w = \xi \tag{1.5}$$

with homogeneous Dirichlet boundary condition, we have  $w \in H^2(\Omega)$  and

$$\|w\|_{H^2} \leq C_e \|\xi\|_{L^2}.$$

Here  $\mathcal{L}^*$  represents the adjoint of the operator  $\mathcal{L}$ .

These estimates, combined with Céa's Lemma, which uses the continuity and coercivity of  $a$ , give the following error estimate:

**Theorem 5** (see e.g. [EG04]). *Let  $1 \leq q \leq k$ . Considering  $u \in H^{q+1}(\Omega)$  and  $u_h \in V_h^0$  the solutions to (1.1) and (1.3), one has*

$$|u - u_h|_{H^1} \leq C_q \frac{\gamma}{\alpha} h^q |u|_{H^{q+1}}$$

and

$$\|u - u_h\|_{L^2} \leq C_e C_1 C_q \frac{\gamma^2}{\alpha} h^{q+1} |u|_{H^{q+1}},$$

where  $\gamma$  and  $\alpha$  are respectively the constants of continuity and coercivity of  $a$ .

For the sake of simplicity, we consider an elliptic boundary value problem with homogeneous Dirichlet conditions. Obviously, we can use more general boundary conditions as Robin-like conditions depending on the parameters. This will be investigated in the numerical experiments to show that the proposed methodology applies to a larger class of boundary value problems, see Section 7.7.

## 2 Enriching the finite element method with additive priors

In this section, we assume that a prior knowledge of the solution to (0.1) is available. In what follows, we call this information a ‘‘prior’’. This prior is denoted by  $\mathbf{x} \mapsto u_\theta(\mathbf{x})$  with parameters  $\theta$ , and we assume that it can be constructed with the desired regularity  $u_\theta \in H^{q+1}(\Omega) \cap H_0^1(\Omega)$  for  $1 \leq q \leq k$ , where  $k$  is the polynomial degree of the enriched FEM. In this section and the following two, the prior is assumed to be a general function. However, from Section 5 onwards, the prior will be the prediction of a parametric PINN. In Section 2.1, we first show how to use this prior to enriching classical finite element spaces. Then, in Section 2.2, we prove a convergence estimate for the resulting method.

## 2.1 Construction of the modified problem

In the general setting of FEM, we follow the Bubnov–Galerkin method [EG04], where the basis functions and the numerical solutions are in the same space (see (1.3), where both  $u_h$  and  $v_h$  are in  $V_h^0$ ). As we intend to enrich the classical approximation space, we exploit the idea formalized as the Petrov–Galerkin method (e.g., [Red19, BS08, Dem23]), where the test and trial functions belong to different spaces. This approach is often used for convection-dominated problems, see [AS97]. We propose to enrich the trial space using the prior  $u_\theta$  by defining

$$V_h^+ = \{u_h^+ = u_\theta + p_h^+, \quad p_h^+ \in V_h^0\}, \quad (2.1)$$

and we use the space  $V_h^0$  for the test functions. Since we have assumed that  $u_\theta \in H^{q+1}(\Omega) \cap H_0^1(\Omega)$ ,  $V_h^+$  is also a subset of  $V^0$ , like  $V_h^0$ . Plugging this new trial space into the approximate problem (1.3), we obtain the formulation

$$\text{Find } u_h^+ \in V_h^+ \text{ such that, } \forall v_h \in V_h^0, \quad a(u_h^+, v_h) = l(v_h), \quad (2.2)$$

which leads to the following approximation problem:

$$\text{Find } p_h^+ \in V_h^0 \text{ such that, } \forall v_h \in V_h^0, \quad a(p_h^+, v_h) = l(v_h) - a(u_\theta, v_h). \quad (2.3)$$

Therefore, we obtain a classical Galerkin approximation with a modified source term.

## 2.2 Convergence analysis

The objective is to prove that the FEM solution to problem (2.3) converges, with an error depending on the quality of the prior. Due to this dependence on the prior (neural network in the numerical section), we are not conducting an in-depth study of this gain. We are simply aiming to show that adding the prior does not degrade the convergence orders and that it is the high-order derivatives that control the gain in comparison with classical finite element method, which helps to guide choices regarding the prior.

**Theorem 6.** *Let  $u \in H^{q+1}(\Omega)$  be the solution to problem (1.1) and  $u_\theta \in H^{q+1}(\Omega) \cap H_0^1(\Omega)$  be a prior on  $u$ . We consider  $u_h^+ \in V_h^+$  as the solution to the discrete problem (2.2) with  $V_h^+$  the modified trial space defined in (2.1). The following estimates hold. For all  $1 \leq q \leq k$ ,*

$$|u - u_h^+|_{H^1} \leq C_q \frac{\gamma}{\alpha} C_{\text{gain}}^+ h^q |u|_{H^{q+1}} \quad (2.4)$$

and, supposing that dual problem (1.5) admits strong solutions (i.e. for all  $\xi \in L^2(\Omega)$  there exists an associated solution  $w \in H_0^1(\Omega) \cap H^2(\Omega)$  to (1.5)),

$$\|u - u_h^+\|_{L^2} \leq C_e C_1 C_q \frac{\gamma^2}{\alpha} C_{\text{gain}}^+ h^{q+1} |u|_{H^{q+1}}, \quad (2.5)$$

with  $C_e, C_1, C_q, \gamma, \alpha$  defined in Section 1 and

$$C_{\text{gain}}^+ = \frac{|u - u_\theta|_{H^{q+1}}}{|u|_{H^{q+1}}}. \quad (2.6)$$

**Remark 7.** *The constant  $C_{\text{gain}}^+$  represents the potential gain compared to the error of the classical FEM presented in Theorem 5. Note that this constant is the same in  $L^2$  norm and  $H^1$  semi-norm.*

*Proof of Theorem 6.  $H^1$ -error:* To prove (2.4), we adapt the proof of C ea’s lemma to the additive prior case. Considering the trial space defined in (2.1), the numerical solution  $u_h^+$  is given by

$$u_h^+ = u_\theta + p_h^+,$$

with  $p_h^+ \in V_h^0 \subset V$  solution to (2.3). We have

$$a(u - u_h^+, u - u_h^+) = a(u - u_h^+, (u - u_\theta) - v_h) + a(u - u_h^+, v_h - p_h^+), \quad \forall v_h \in V_h^0.$$

Let us first treat the second term on the right-hand side. By Galerkin orthogonality (difference of the continuous problem (1.1) and discrete problem (2.2)), we obtain

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

Denoting by  $\alpha$  and  $\gamma$  the coercivity and continuity constants of the bilinear form  $a$ , we have

$$\begin{aligned}\alpha|u - u_h^+|_{H^1}^2 &\leq a(u - u_h^+, u - u_h^+) = a(u - u_h^+, (u - u_\theta) - v_h), \quad \forall v_h \in V_h^0, \\ &\leq \gamma|u - u_h^+|_{H^1}|(u - u_\theta) - v_h|_{H^1}, \quad \forall v_h \in V_h^0,\end{aligned}$$

which immediately leads to

$$|u - u_h^+|_{H^1} \leq \frac{\gamma}{\alpha}|(u - u_\theta) - v_h|_{H^1}, \quad \forall v_h \in V_h^0.$$

We apply it to  $v_h = \mathcal{I}_h(u - u_\theta) \in V_h^0$  with  $\mathcal{I}_h$  the Lagrange interpolation operator (1.4) in  $V_h$ , it holds using interpolation estimate given in Theorem 3,

$$|u - u_h^+|_{H^1} \leq C_q \frac{\gamma}{\alpha} h^q |u - u_\theta|_{H^{q+1}},$$

with  $C_q$  defined in Section 1.

The above expression can be rewritten as

$$|u - u_h^+|_{H^1} \leq C_q \frac{\gamma}{\alpha} C_{\text{gain}}^+ h^q |u|_{H^{q+1}}, \quad (2.7)$$

with

$$C_{\text{gain}}^+ = \frac{|u - u_\theta|_{H^{q+1}}}{|u|_{H^{q+1}}},$$

which completes the first part of the proof.

**$L^2$ -error:** We will follow the Aubin–Nitsche technique. Consider  $w \in H_0^1(\Omega) \cap H^2(\Omega)$  solution to

$$\mathcal{L}^* w = u - u_h^+,$$

with homogeneous Dirichlet boundary condition. Thanks to Theorem 4, one has

$$\|w\|_{H^2} \leq C_e \|u - u_h^+\|_{L^2}. \quad (2.8)$$

Using the Galerkin orthogonality and the continuity of the bilinear form  $a$ ,

$$\|u - u_h^+\|_{L^2}^2 = a(u - u_h^+, w - I_h w) \leq \gamma|u - u_h^+|_{H^1}|w - I_h w|_{H^1}.$$

Thanks to Theorem 3 and (2.8),

$$|w - I_h w|_{H^1} \leq C_e C_1 h \|u - u_h^+\|_{L^2},$$

which leads to the conclusion by using (2.7).  $\square$

**Remark 8.** The gain constant  $C_{\text{gain}}^+$  defined in (2.6) shows that the closer the prior is to the solution, the smaller is the error constant associated with the FEM while keeping the same order of accuracy. Therefore, as soon as  $C_{\text{gain}}^+ < 1$ , the FEM with additive prior will be more accurate than the classical one. While this gives us a particularly flexible constraint, our objective is to balance this gain by relaxing the contribution  $h^q$ , using a coarser grid and low-order polynomial, to reduce the computational cost of the FEM while maintaining accuracy. Nonetheless, the gain is related to the  $L^2$  error associated with the derivatives of  $(q+1)^{\text{th}}$  order (with  $1 \leq q \leq k$ ) of the prior. This shows that the prior must accurately approximate the derivatives of the solution in addition to the solution itself. This highlights that we need to build our prior by ensuring a good approximation of the derivatives of the solution. It also shows that the higher the order of the finite elements  $k$  is, the better our prior should approximate the higher-order derivatives. Therefore, it is more appropriate to use only low-order FEM so that  $k$  remains small.

### 3 Enriching the finite element method with multiplicative priors

This section employs the same assumptions as in Section 2, namely that we have a sufficiently smooth prior  $u_\theta$  on the solution  $u$  of the PDE (0.1). However, this prior will now be multiplied to elements of  $V_h$  rather than added to them. We construct the underlying modified problem in Section 3.1. Then, similarly to the additive approach of Section 2, error estimates are obtained in Section 3.2. The objectives of this section on error estimates are the same as in the previous section.

### 3.1 Construction of the modified problem

To construct the modified problem in this case, we must ensure that the prior  $u_\theta$  never vanishes. To that end, we lift the initial problem (0.1) by a constant  $M \in \mathbb{R}_+$ , chosen large enough to ensure that  $u_M = u + M > 0$ , to get

$$\begin{cases} \mathcal{L}(u_M) = f, & \text{in } \Omega, \\ u_M = M, & \text{on } \partial\Omega. \end{cases} \quad (3.1)$$

We then introduce the associated variational problem, defined by

$$\text{Find } u_M = u + M, \text{ with } u \in V^0 \text{ such that, } \forall v \in V^0, a(u_M, v) = l(v). \quad (3.2)$$

Therefore, solving (3.2), we recover the solution  $u$  of the initial problem (0.1) by setting

$$u = u_M - M.$$

The prior

$$u_{\theta, M} = u_\theta + M > 0$$

is associated with problem (3.2).

Let us now introduce a new modified finite element space, defined by

$$V_h^\times = \left\{ u_{h, M}^\times = u_{\theta, M} p_h^\times, \quad p_h^\times \in 1 + V_h^0 \right\}, \quad (3.3)$$

with, for all  $\mathbf{x} \in \Omega$ ,  $u_{\theta, M}(\mathbf{x}) \neq 0$ . From (3.2), this leads to the following approximate formulation:

$$\text{Find } p_h^\times \in 1 + V_h^0 \text{ such that, } \forall v_h \in V_h^0, a(u_{\theta, M} p_h^\times, u_{\theta, M} v_h) = l(u_{\theta, M} v_h). \quad (3.4)$$

Therefore, solving (3.4), we recover the solution  $u_h^\times \in V_h^\times - M$  of the original problem (0.1) by setting

$$u_h^\times = u_{h, M}^\times - M.$$

Based on the  $N_{\text{dofs}}$  dofs  $(\mathbf{x}^{(i)})_{i \in \{1, \dots, N_{\text{dofs}}\}}$  of the mesh, we consider the interpolation operator on  $V_h^\times$  given by

$$\tilde{\mathcal{I}}_h : C^0(\Omega) \ni v \mapsto \sum_{i=1}^{N_{\text{dofs}}} \frac{v(\mathbf{x}^{(i)})}{u_{\theta, M}(\mathbf{x}^{(i)})} \tilde{\psi}_i \in V_h^\times,$$

where the shape functions  $\tilde{\psi}_i$  associated to  $V_h^\times$  are defined by

$$\tilde{\psi}_i = u_{\theta, M} \psi_i,$$

with  $\psi_i$  the classical shape functions presented in Section 1. Note that the new interpolation operator  $\tilde{\mathcal{I}}_h$  is related to the classical Lagrange interpolation operator defined in (1.4) as follows

$$\forall v \in C^0(\Omega), \quad \tilde{\mathcal{I}}_h(v) = u_{\theta, M} \mathcal{I}_h \left( \frac{v}{u_{\theta, M}} \right). \quad (3.5)$$

### 3.2 Convergence analysis

In this section, we finally prove that the modified FEM (3.4) converges to the solution to (3.1), and that it satisfies the same type of estimate as the classical one. Equipped with the lifting trick from Section 3.1, we can state the following convergence theorem.

**Theorem 9.** *Let  $u_M \in H^{q+1}(\Omega)$  be the solution of the enhanced problem (3.2) and  $u_{\theta, M} \in M + H^{q+1}(\Omega) \cap H_0^1(\Omega)$  be a prior on  $u_M$ . We consider  $u_{h, M}^\times \in V_h^\times$  the solution to the finite element problem (3.4) with  $V_h^\times$  the modified trial space defined in (3.3), considering  $\mathbb{P}_k$  polynomials. We define  $u = u_M - M$  and  $u_h^\times = u_{h, M}^\times - M$ . Then, for all  $1 \leq q \leq k$*

$$|u - u_h^\times|_{H^1} \leq C_q \frac{\gamma}{\alpha} C_{\text{gain}, H^1}^{\times, M} h^q |u|_{H^{q+1}} \quad (3.6)$$

and, supposing that dual problem (1.5) admits strong solutions (i.e. for all  $\xi \in L^2(\Omega)$ , there exists an associated solution  $w \in H_0^1(\Omega) \cap H^2(\Omega)$  to (1.5)),

$$\|u - u_h^\times\|_{L^2} \leq C_e C_1 C_q \frac{\gamma^2}{\alpha} C_{\text{gain}, L^2}^{\times, M} h^{q+1} |u|_{H^{q+1}}, \quad (3.7)$$

with  $C_e, C_1, C_q, \gamma, \alpha$  defined in Section 1, and where

$$C_{\text{gain}, H^1}^{\times, M} = \left| \frac{u_M}{u_{\theta, M}} \right|_{H^{q+1}} \frac{\|u_{\theta, M}\|_{W^{1, \infty}}}{|u|_{H^{q+1}}}, \quad (3.8)$$

and

$$C_{\text{gain}, L^2}^{\times, M} = C_{\theta, M} \left| \frac{u_M}{u_{\theta, M}} \right|_{H^{q+1}} \frac{\|u_{\theta, M}\|_{W^{1, \infty}}^2}{|u|_{H^{q+1}}}, \quad (3.9)$$

with

$$C_{\theta, M} = \|u_{\theta, M}^{-1}\|_{L^\infty} + 2|u_{\theta, M}^{-1}|_{W^{1, \infty}} + |u_{\theta, M}^{-1}|_{W^{2, \infty}}. \quad (3.10)$$

**Remark 10.** The constants  $C_{\text{gain}, H^1}^{\times, M}$  and  $C_{\text{gain}, L^2}^{\times, M}$  represent the potential gains in both  $H^1$  semi-norm and  $L^2$  norm when using the multiplicative approach, compared to the error of the classical FEM presented in Theorem 5 with  $\mathbb{P}_k$  polynomials.

*Proof of Theorem 9.  $H^1$ -error:* Considering the trial space defined in (3.3), the numerical solution  $u_{h, M}^\times$  is given by

$$u_{h, M}^\times = u_{\theta, M} p_h^\times,$$

with  $p_h^\times \in 1 + V_h^0 \subset 1 + V^0$  solution to (3.4). By coercivity of  $a$ ,

$$\alpha |u_M - u_{h, M}^\times|_{H^1}^2 \leq a(u_M - u_{h, M}^\times, u_M - u_{h, M}^\times).$$

Thanks to (3.2) and (3.4), we have the following Galerkin orthogonality: for all  $v_h \in V_h^0$ ,

$$a(u_M - u_{h, M}^\times, u_{\theta, M} v_h) = 0. \quad (3.11)$$

Now, choosing

$$v_h = p_h^\times - \mathcal{I}_h \left( \frac{u_M}{u_{\theta, M}} \right),$$

we note that  $v_h$  is well-defined since the lifting trick ensures that  $u_{\theta, M} > 0$ , and that it belongs to  $V_h^0$ . With this choice, we deduce by the definition (3.5) of  $\tilde{\mathcal{I}}_h$  that

$$a(u_M - u_{h, M}^\times, u_M - u_{h, M}^\times) = a \left( u_M - u_{h, M}^\times, u_M - u_{\theta, M} \mathcal{I}_h \left( \frac{u_M}{u_{\theta, M}} \right) \right) = a(u_M - u_{h, M}^\times, u_M - \tilde{\mathcal{I}}_h(u_M)).$$

By continuity of  $a$ ,

$$|u_M - u_{h, M}^\times|_{H^1} \leq \frac{\gamma}{\alpha} |u_M - \tilde{\mathcal{I}}_h(u_M)|_{H^1}. \quad (3.12)$$

Again, using the definition (3.5) of  $\tilde{\mathcal{I}}_h$ ,

$$|u_M - \tilde{\mathcal{I}}_h(u_M)|_{H^1} \leq \|u_{\theta, M}\|_{W^{1, \infty}} \left\| \frac{u_M}{u_{\theta, M}} - \mathcal{I}_h \left( \frac{u_M}{u_{\theta, M}} \right) \right\|_{H^1}.$$

Finally, applying interpolation estimate given in Theorem 3, it holds

$$|u_M - \tilde{\mathcal{I}}_h(u_M)|_{H^1} \leq C_q \|u_{\theta, M}\|_{W^{1, \infty}} h^q \left| \frac{u_M}{u_{\theta, M}} \right|_{H^{q+1}}, \quad (3.13)$$

with  $C_q$  defined in Section 1. Combining the last inequality with (3.12), we obtain

$$|u - u_h^\times|_{H^1} = |u_M - u_{h, M}^\times|_{H^1} \leq C_q \frac{\gamma}{\alpha} C_{\text{gain}, H^1}^{\times, M} h^q |u|_{H^{q+1}}, \quad (3.14)$$

with  $C_{\text{gain}, H^1}^{\times, M}$  given in (3.8), which conclude the first part of the proof.

**$L^2$ -error:** Again, we follow the Aubin-Nitsche strategy here. Consider  $w - M \in H_0^1(\Omega) \cap H^2(\Omega)$  solution to

$$\mathcal{L}^* w = u - u_h^\times = u_M - p_h^\times u_{\theta, M},$$

with  $w = M$  on  $\partial\Omega$ . Thanks to Theorem 4, one has

$$\|w\|_{H^2} \leq C_e \|u - u_h^\times\|_{L^2}.$$

Then, using the Galerkin orthogonality (3.11) for  $v_h = \mathcal{I}_h \left( \frac{u_M}{u_{\theta, M}} \right)$ ,

$$\|u - u_h^\times\|_{L^2}^2 = \|u_M - p_h^\times u_{\theta, M}\|_{L^2}^2 = a(u_M - p_h^\times u_{\theta, M}, w) = a(u_M - p_h^\times u_{\theta, M}, w - \tilde{I}_h(w)).$$

Hence, by continuity of  $a$ ,

$$\|u - u_h^\times\|_{L^2}^2 \leq \gamma |u_M - p_h^\times u_{\theta, M}|_{H^1} |w - \tilde{I}_h(w)|_{H^1}.$$

Using (3.14) and (3.13) for  $q = 1$  to the term in the right-hand side,

$$\|u - u_h^\times\|_{L^2}^2 \leq C_1 C_q \frac{\gamma^2}{\alpha} \|u_{\theta, M}\|_{W^{1, \infty}} \left| \frac{w}{u_{\theta, M}} \right|_{H^2} C_{\text{gain}, H^1}^{\times, M} h^{q+1} |u|_{H^{q+1}}.$$

Moreover,

$$\left| \frac{w}{u_{\theta, M}} \right|_{H^2} \leq C_{\theta, M} \|w\|_{H^2},$$

with  $C_{\theta, M}$  given in (3.10). Thanks to the elliptic regularity, we obtain

$$\|u - u_h^\times\|_{L^2} \leq C_e C_1 C_q \frac{\gamma^2}{\alpha} C_{\text{gain}, L^2}^{\times, M} h^{q+1} |u|_{H^{q+1}},$$

with  $C_{\text{gain}, L^2}^{\times, M}$  defined in (3.9). □

**Remark 11.** We note that the gain constants  $C_{\text{gain}, H^1}^{\times, M}$  and  $C_{\text{gain}, L^2}^{\times, M}$  are similar to the constant  $C_{\text{gain}}^+$  introduced in Section 2, in that, it depends on high-order derivatives of the prior. Hence, a high-quality prior will necessarily involve a good approximation of the derivatives of the exact solution, and Remark 8 also applies in the present context. The major difference with the additive approach lies in the choice of the lifting constant  $M$ . To better understand this dependency in  $M$ , the following section provides a study of the behavior of our two gain constants when  $M$  goes to infinity. Moreover, the actual choice of  $M$  will be numerically investigated in Section 7.

## 4 Comparison of the two enriched methods

This section aims to compare the two proposed methods, namely the additive approach presented in Section 2 and the multiplicative one proposed in Section 3. Recall that the constant  $M$  is chosen in the multiplicative approach so that  $u_M > 0$ . Let  $u$  be the solution of problem (1.1) and  $u_\theta \in H^{q+1}(\Omega) \cap H_0^1(\Omega)$  be a prior on  $u$ , with  $1 \leq q \leq k$  ( $k$  the polynomial degree of the finite element method). For clarity, we first recall the additive and multiplicative enrichments, and their main error estimates.

**Additive approach.** We consider  $u_h^+ \in V_h^+$  as the solution to the finite element method associated to problem (2.3) with  $V_h^+$  the modified trial space defined in (2.1), considering  $\mathbb{P}_k$  polynomials. Using Theorem 6, we have for  $1 \leq q \leq k$ , the following additive theoretical gain constant

$$C_{\text{gain}}^+ = \frac{|u - u_\theta|_{H^{q+1}}}{|u|_{H^{q+1}}}. \quad (4.1)$$

**Multiplicative approach.** Let  $u_M = u + M$  be the solution of the enhanced problem (3.2) and  $u_{\theta,M} = u_\theta + M$  be a prior on  $u_M$ . We consider  $u_{h,M}^\times \in V_h^\times$  the solution to the finite element problem (3.4) with  $V_h^\times$  the modified trial space defined in (3.3), considering  $\mathbb{P}_k$  polynomials. Using Theorem 9, we have for  $1 \leq q \leq k$ , the following multiplicative theoretical gain constants

$$C_{\text{gain},H^1}^{\times,M} = \left| \frac{u_M}{u_{\theta,M}} \right|_{H^{q+1}} \frac{\|u_{\theta,M}\|_{W^{1,\infty}}}{|u|_{H^{q+1}}}, \quad (4.2)$$

and

$$C_{\text{gain},L^2}^{\times,M} = C_{\theta,M} \left| \frac{u_M}{u_{\theta,M}} \right|_{H^{q+1}} \frac{\|u_{\theta,M}\|_{W^{1,\infty}}^2}{|u|_{H^{q+1}}}, \quad (4.3)$$

with  $C_{\theta,M}$  given in (3.10).

**Comparison of the two approaches.** The following result proves that the upper bound in (3.6) and in (3.7) converges respectively to the one in (2.4) and in (2.5) when  $M$  goes to infinity. In other words, the multiplicative gain constants defined in (4.2) and (4.3) converge to the additive gain constant defined in (4.1) when  $M$  goes to infinity.

**Theorem 12.** *We have*

$$C_{\text{gain},H^1}^{\times,M} \xrightarrow{M \rightarrow \infty} C_{\text{gain}}^+, \quad (4.4)$$

and

$$C_{\text{gain},L^2}^{\times,M} \xrightarrow{M \rightarrow \infty} C_{\text{gain}}^+. \quad (4.5)$$

*Proof. Convergence in  $H^1$ :* According to the expressions (4.1) and (4.2) of the gain constants, the objective of the proof is to show that

$$\|u_{\theta,M}\|_{W^{1,\infty}} \left| \frac{u_M}{u_{\theta,M}} \right|_{H^{q+1}} \xrightarrow{M \rightarrow \infty} |u - u_\theta|_{H^{q+1}}.$$

Denoting by

$$E_\theta = u - u_\theta,$$

the error made by the prior  $u_\theta$  when approximating the solution  $u$ , we have

$$|u - u_\theta|_{H^{q+1}} = |E_\theta|_{H^{q+1}}.$$

On the one hand, we have,

$$\|u_{\theta,M}\|_{W^{1,\infty}} = \|u_\theta + M\|_{W^{1,\infty}} = M \left\| 1 + \frac{u_\theta}{M} \right\|_{W^{1,\infty}}.$$

On the other hand, we have,

$$\left| \frac{u_M}{u_{\theta,M}} \right|_{H^{q+1}} = \left| \frac{u + M}{u_\theta + M} \right|_{H^{q+1}} = \left| \frac{u - u_\theta + u_\theta + M}{u_\theta + M} \right|_{H^{q+1}} = \left| 1 + \frac{u - u_\theta}{u_\theta + M} \right|_{H^{q+1}} = \frac{1}{M} \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} \right|_{H^{q+1}}.$$

Multiplying these expressions, we obtain

$$\|u_{\theta,M}\|_{W^{1,\infty}} \left| \frac{u_M}{u_{\theta,M}} \right|_{H^{q+1}} = \underbrace{\left\| 1 + \frac{u_\theta}{M} \right\|_{W^{1,\infty}}}_{\text{(I)}} \underbrace{\left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} \right|_{H^{q+1}}}_{\text{(II)}}. \quad (4.6)$$

We now estimate term by term the right-hand side of the above equality (4.6), looking at their limits when  $M$  goes to infinity.

**Term (I):** By decomposing the first term, we obtain

$$\text{(I)} = \left\| 1 + \frac{u_\theta}{M} \right\|_{W^{1,\infty}} = \left\| 1 + \frac{u_\theta}{M} \right\|_{L^\infty} + \frac{1}{M} \|\nabla u_\theta\|_{L^\infty} \xrightarrow{M \rightarrow \infty} 1,$$

since

$$1 - \frac{\|u_\theta\|_{L^\infty}}{M} \leq \left\| 1 + \frac{u_\theta}{M} \right\|_{L^\infty} \leq 1 + \frac{\|u_\theta\|_{L^\infty}}{M}. \quad (4.7)$$

**Term (II):** Let us prove that

$$(II) = \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} \right|_{H^{q+1}} \xrightarrow{M \rightarrow \infty} |E_\theta|_{H^{q+1}}.$$

By the triangular inequality,

$$\left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} \right|_{H^{q+1}}^2 \leq \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} - E_\theta \right|_{H^{q+1}}^2 + |E_\theta|_{H^{q+1}}^2$$

and

$$|E_\theta|_{H^{q+1}}^2 \leq \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} - E_\theta \right|_{H^{q+1}}^2 + \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} \right|_{H^{q+1}}^2.$$

Hence, by definition of the semi-norm  $|\cdot|_{H^{q+1}}$ ,

$$\begin{aligned} \left| \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} \right|_{H^{q+1}}^2 - |E_\theta|_{H^{q+1}}^2 \right| &\leq \left| \frac{E_\theta}{1 + \frac{u_\theta}{M}} - E_\theta \right|_{H^{q+1}}^2 \\ &= \left\| \nabla^{q+1} \left( \frac{E_\theta}{1 + \frac{u_\theta}{M}} - E_\theta \right) \right\|_{L^2}^2 \leq |\Omega| \left\| \nabla^{q+1} \left( \frac{E_\theta}{1 + \frac{u_\theta}{M}} - E_\theta \right) \right\|_{L^\infty}^2. \end{aligned}$$

Then, using the general Leibniz rule, we have,

$$\begin{aligned} \left\| \nabla^{q+1} \left( \frac{E_\theta}{1 + \frac{u_\theta}{M}} - E_\theta \right) \right\|_{L^\infty} &\leq \left\| \frac{\nabla^{q+1} E_\theta}{1 + \frac{u_\theta}{M}} - \nabla^{q+1} E_\theta \right\|_{L^\infty} + \sum_{s=1}^{q+1} \binom{q+1}{s} \|\nabla^{q+1-s} E_\theta\|_{L^\infty} \left\| \nabla^s \left( \frac{1}{1 + \frac{u_\theta}{M}} \right) \right\|_{L^\infty} \\ &\leq \underbrace{\left\| \frac{\nabla^{q+1} E_\theta}{1 + \frac{u_\theta}{M}} - \nabla^{q+1} E_\theta \right\|_{L^\infty}}_{(1)} + \|E_\theta\|_{W^{q+1,\infty}} \sum_{s=1}^{q+1} \binom{q+1}{s} \underbrace{\left\| \nabla^s \left( \frac{1}{1 + \frac{u_\theta}{M}} \right) \right\|_{L^\infty}}_{(2)}. \end{aligned}$$

We now estimate terms (1) and (2) in the right-hand side of the above inequality.

**Term (1):** Taking  $M > \|u_\theta\|_{L^\infty}$ , we obtain

$$(1) \leq \left\| \frac{1}{1 + \frac{u_\theta}{M}} - 1 \right\|_{L^\infty} \|\nabla^{q+1} E_\theta\|_{L^\infty} \leq \frac{1}{M} \frac{\|u_\theta\|_{L^\infty}}{1 - \frac{\|u_\theta\|_{L^\infty}}{M}} \|\nabla^{q+1} E_\theta\|_{L^\infty} \xrightarrow{M \rightarrow \infty} 0.$$

**Term (2):** Let us define  $g = 1 + \frac{u_\theta}{M}$ . The Faà di Bruno formula (see [Joh02]) gives the following expression for the  $s$ -th derivative of the inverse of  $g$ :

$$\nabla^s \left( \frac{1}{g} \right) = \frac{s!}{g^{s+1}} \sum \frac{(-1)^\alpha \alpha!}{\prod_{i=1}^s (i!)^{m_i} m_i!} g^{s-\alpha} \prod_{i=1}^s (\nabla^i g)^{m_i},$$

where the sum covers all  $s$ -tuples  $(m_1, \dots, m_s)$  satisfying the constraint  $s = \sum_{i=1}^s i m_i$  and  $\alpha = \sum_{i=1}^s m_i$ .

Thus, using (4.7), there exists a constant  $C > 0$  such that, for any  $M > \|u_\theta\|_{L^\infty}$  and  $1 \leq s \leq q+1$ , the following estimate holds

$$\begin{aligned} (2) &\leq C \frac{1}{\min_{x \in \Omega} \left| 1 + \frac{u_\theta(x)}{M} \right|^{s+1}} \sum \left\| 1 + \frac{u_\theta}{M} \right\|_{L^\infty}^{s-\alpha} \prod_{i=1}^s \left\| \nabla^i \left( 1 + \frac{u_\theta}{M} \right) \right\|_{L^\infty}^{m_i} \\ &\leq C \frac{1}{\left( 1 - \frac{\|u_\theta\|_{L^\infty}}{M} \right)^{s+1}} \sum \left( 1 + \frac{\|u_\theta\|_{L^\infty}}{M} \right)^{s-\alpha} \frac{1}{M^\alpha} \prod_{i=1}^s \|\nabla^i u_\theta\|_{L^\infty}^{m_i} \xrightarrow{M \rightarrow \infty} 0, \end{aligned}$$

which leads to (4.4).

**Convergence in  $L^2$ :** Using the convergence of the gain for the  $H^1$  semi-norm, we only need to prove that

$$C_{\theta,M} \|u_{\theta,M}\|_{W^{1,\infty}} \xrightarrow{M \rightarrow \infty} 1,$$

with  $C_{\theta,M}$  given by

$$C_{\theta,M} = \|u_{\theta,M}^{-1}\|_{L^\infty} + 2|u_{\theta,M}^{-1}|_{W^{1,\infty}} + |u_{\theta,M}^{-1}|_{W^{2,\infty}}.$$

Since

$$\frac{1}{M} \|u_{\theta,M}\|_{W^{1,\infty}} \xrightarrow{M \rightarrow \infty} 1,$$

we are only required to prove that

$$MC_{\theta,M} \xrightarrow{M \rightarrow \infty} 1.$$

Considering  $M > \|u_\theta\|_{L^\infty}$ , we have

$$M \|u_{\theta,M}^{-1}\|_{L^\infty} = \left\| \frac{1}{1 + \frac{u_\theta}{M}} \right\|_{L^\infty} \leq \frac{1}{1 - \frac{\|u_\theta\|_{L^\infty}}{M}} \xrightarrow{M \rightarrow \infty} 1.$$

Moreover,

$$2M |u_{\theta,M}^{-1}|_{W^{1,\infty}} = 2M \left\| \frac{\nabla u_\theta}{(u_\theta + M)^2} \right\|_{L^\infty} \xrightarrow{M \rightarrow \infty} 0.$$

Similarly,

$$M |u_{\theta,M}^{-1}|_{W^{2,\infty}} \xrightarrow{M \rightarrow \infty} 0,$$

which leads to the conclusion (4.5). □

## 5 Prior construction using parametric PINNs

We have introduced new finite element approximation spaces in Sections 2 and 3 depending on the construction of priors. Physics-Informed Neural Networks (PINNs) are a good choice to build such priors. Indeed, since PINNs minimize the PDE residual, they inherently give a good approximation of the derivative of the solution, in addition to the solution itself (see e.g. [RPK19]). This section is therefore dedicated to introducing PINNs in Section 5.1, and then to show how to improve them in Section 5.2.

### 5.1 Physics-Informed Neural Networks for parametric PDEs

Physics-Informed Neural Networks, or PINNs, were introduced by [RPK19] for solving a PDE with neural networks. The main idea is to recast a PDE as an optimization problem. We illustrate the method on our problem (0.1), which we now extend to non-homogeneous Dirichlet boundary conditions. Moreover, recall that the problem of interest is a parametric PDE. Unlike classical PINNs, which are trained for specific physical parameters, parametric PINNs seek to learn a generalized solution covering a range of parameters. They incorporate these parameters as additional inputs to the network, allowing greater flexibility in solving problems where physical conditions or properties vary. Moreover, since they are based on neural networks, PINNs are ideally suited to solving such higher-dimensional problems. This advantage is compounded when one uses Monte-Carlo integration to estimate the loss functions, as its convergence rate is also insensitive to the dimension.

Considering  $p$  parameters  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_p) \in \mathcal{M} \subset \mathbb{R}^p$ , with some parameter space  $\mathcal{M}$ , the parametric PDE (0.1) reads, with non-homogeneous Dirichlet boundary conditions:

$$\begin{cases} \mathcal{L}(u(\mathbf{x}, \boldsymbol{\mu}); \mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{x}, \boldsymbol{\mu}), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}, \boldsymbol{\mu}) = g(\mathbf{x}, \boldsymbol{\mu}), & \mathbf{x} \in \partial\Omega, \end{cases} \quad (5.1)$$

with  $g$  the trace of a  $H^2$  function on  $\partial\Omega$ . Note that the solution of the equation depends on the parameters  $\boldsymbol{\mu}$ , as do the operator  $\mathcal{L}$  and the boundary conditions. We then denote  $u_\theta(\cdot, \boldsymbol{\mu})$  the approximate PINN prediction for given parameters  $\boldsymbol{\mu}$ .

The first idea of PINNs comes from the observation that, by construction, neural networks with smooth activation functions are nothing but smooth functions of their weights and inputs. Therefore, neural networks form natural candidates for approximating solutions to PDEs, especially with the advent of automatic differentiation tools. In our case, a PINN is a neural network that takes  $d + p$  inputs, where  $d$  is the dimension of the space variable  $\mathbf{x} \in \Omega$  and

$p$  is the number of parameters  $\boldsymbol{\mu} \in \mathcal{M}$ . We denote by  $u_\theta(\mathbf{x}, \boldsymbol{\mu})$  the output, where  $\theta$  are the learnable weights of the network. Classically, this neural network is a coordinate-based neural network, such as a multi-layer perceptron (MLP). It depends on several user-defined hyperparameters, to be specified in the numerical experiments.

Once this network is defined, solving the PDE can be rewritten as a minimization problem on  $\theta$ , namely finding the optimal weights  $\theta^*$  that satisfy the following minimization problem:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} (\omega_r J_r(\theta) + \omega_b J_b(\theta) + \omega_{\text{data}} J_{\text{data}}(\theta)), \quad (5.2)$$

with  $\omega_r$ ,  $\omega_b$  and  $\omega_{\text{data}}$  some weights to balance the different terms of the loss function. In (5.2), the loss function has three terms: the residual loss function

$$J_r(\theta) = \int_{\mathcal{M}} \int_{\Omega} |\mathcal{L}(u_\theta(\mathbf{x}, \boldsymbol{\mu}); \mathbf{x}, \boldsymbol{\mu}) - f(\mathbf{x}, \boldsymbol{\mu})|^2 \mathrm{d}\mathbf{x} \mathrm{d}\boldsymbol{\mu}, \quad (5.3)$$

the boundary loss function

$$J_b(\theta) = \int_{\mathcal{M}} \int_{\partial\Omega} |u_\theta(\mathbf{x}, \boldsymbol{\mu}) - g(\mathbf{x}, \boldsymbol{\mu})|^2 \mathrm{d}\mathbf{x} \mathrm{d}\boldsymbol{\mu}, \quad (5.4)$$

and the data loss function

$$J_{\text{data}}(\theta) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} |u_\theta(\mathbf{x}_{\text{data}}^{(i)}, \boldsymbol{\mu}_{\text{data}}^{(i)}) - u_{\text{data}}^{(i)}|^2, \quad (5.5)$$

where  $(\mathbf{x}_{\text{data}}^{(i)}, \boldsymbol{\mu}_{\text{data}}^{(i)}, u_{\text{data}}^{(i)})_{i=1, \dots, N_{\text{data}}}$  are  $N_{\text{data}}$  known data points, with  $u_{\text{data}}^{(i)}$  a reference solution at points  $\mathbf{x}_{\text{data}}^{(i)}$  and for given parameters  $\boldsymbol{\mu}_{\text{data}}^{(i)}$ . These reference solutions can be the exact solutions of the parametric PDE, defined in this case by  $u_{\text{data}}^{(i)} = u(\mathbf{x}_{\text{data}}^{(i)}; \boldsymbol{\mu}_{\text{data}}^{(i)})$ . They can also be an approximation produced by a numerical method, such as finite elements on a fine mesh.

**Remark 13.** In Section 7, the focus is on PINNs trained only with a residual loss function (with boundary conditions imposed exactly as presented in Section 5.2.1). We will only consider the BC loss function in Section 7.5.3. Furthermore, we will not use the data loss function in PINNs except in Section 7.2 where we will seek to compare a true physics-informed network to a merely data-driven one.

Solving the minimization problem (5.2) requires computing the gradient of the loss function with respect to  $\theta$ , which involves calculating the integrals in (5.3) and (5.4). The most natural idea is to estimate them with a Monte-Carlo method, see e.g. [Caf98]. One could also use Gauss-type quadrature rules to evaluate integrals, as is done in Variational Physics-Informed Neural Networks [KZK21], but the limitation is the impossibility of selecting an adequate quadrature order due to the unknown properties of the neural network approximation. For that purpose, we define so-called ‘‘collocation points’’ on  $\Omega \times \mathcal{M}$  and its boundary  $\partial\Omega \times \mathcal{M}$ , denoted respectively by  $(\mathbf{x}_{\text{col}}^{(i)}, \boldsymbol{\mu}_{\text{col}}^{(i)})_{i=1, \dots, N_{\text{col}}}$  and  $(\mathbf{x}_{\text{bc}}^{(i)}, \boldsymbol{\mu}_{\text{bc}}^{(i)})_{i=1, \dots, N_{\text{bc}}}$ . Then, we approximate the residuals and boundary losses by

$$J_r(\theta) \simeq \frac{1}{N_{\text{col}}} \sum_{i=1}^{N_{\text{col}}} |\mathcal{L}(u_\theta(\mathbf{x}_{\text{col}}^{(i)}, \boldsymbol{\mu}_{\text{col}}^{(i)}); \mathbf{x}_{\text{col}}^{(i)}, \boldsymbol{\mu}_{\text{col}}^{(i)}) - f(\mathbf{x}_{\text{col}}^{(i)}, \boldsymbol{\mu}_{\text{col}}^{(i)})|^2 \quad (5.6)$$

and

$$J_b(\theta) \simeq \frac{1}{N_{\text{bc}}} \sum_{i=1}^{N_{\text{bc}}} |u_\theta(\mathbf{x}_{\text{bc}}^{(i)}, \boldsymbol{\mu}_{\text{bc}}^{(i)}) - g(\mathbf{x}_{\text{bc}}^{(i)}, \boldsymbol{\mu}_{\text{bc}}^{(i)})|^2, \quad (5.7)$$

where  $N_{\text{col}}$  and  $N_{\text{bc}}$  are heuristically determined and should be large enough to ensure that the Monte-Carlo integration is accurate enough. The precise values of these parameters will be given in the numerical experiments. In the case of complex geometries, one solution for obtaining a sample of these collocation points in the  $\Omega$  domain is to use a level-set function, denoted  $\varphi$ . This function, which vanishes on the boundary of  $\Omega$ , can be obtained differently. The authors of [SS22] propose different approaches to obtain a level-set function analytically in the case of polygonal or curved geometries. Learning-based approaches have also been proposed, notably in e.g. [PFS<sup>+</sup>19, SMB<sup>+</sup>20].

Because of the minimization problem (5.2), the PINN  $u_\theta$  does not exactly satisfy the boundary conditions. Moreover, loss functions compete, which may require fine-tuning the coefficients between  $J_r$  and  $J_b$ . In addition, classical PINNs do not include information on higher-order derivatives. As highlighted in Remarks 8 and 11, for our purposes, a good prior should yield a good approximation of the derivatives of the solution. For these reasons, the following section recalls several improvements of classical PINNs in the literature.

## 5.2 Improving PINN training and prediction

This section focuses on several ways of improving PINNs: exactly imposing the boundary conditions in [Section 5.2.1](#), adding a higher-order derivative term in the loss function in [Section 5.2.2](#), and countering the spectral bias in [Section 5.2.3](#). Although these approaches are presented separately, they can easily be combined with one another.

### 5.2.1 Exact imposition of boundary conditions

To avoid the issues of classical PINNs discussed in [Section 5.1](#), the authors of [[LLF98](#), [FMDN24](#)] propose a method to enforce inhomogeneous Dirichlet boundary conditions exactly. To that end, they search the approximation  $u_\theta$  of solution to [\(5.1\)](#) with the form: for all  $\mathbf{x} \in \Omega$  and  $\boldsymbol{\mu} \in \mathcal{M}$

$$u_\theta(\mathbf{x}, \boldsymbol{\mu}) = \varphi(\mathbf{x})w_\theta(\mathbf{x}, \boldsymbol{\mu}) + g(\mathbf{x}, \boldsymbol{\mu}),$$

where  $\varphi$  and  $w_\theta$  are, respectively, the level-set function and a neural network as defined in [Section 5.1](#). Thus  $u_\theta$  will automatically satisfy the boundary conditions, since  $u_\theta(\mathbf{x}, \boldsymbol{\mu}) = g(\mathbf{x}, \boldsymbol{\mu})$  for all  $\mathbf{x} \in \partial\Omega$  and  $\boldsymbol{\mu} \in \mathcal{M}$ . Note that this level-set function can be used in a few different ways, firstly to generate a sample of points in  $\Omega$ , as shown in [Section 5.1](#), and secondly to impose boundary conditions. However, to use it directly in the formulation of the prior, it will require a certain regularity. For example the signed distance function is not a good candidate. Similar methods to impose boundary conditions exist for Robin and Neumann conditions; see [[SS22](#)].

In this case, only the residual and data loss functions are minimized, and the minimization problem [\(5.2\)](#) becomes

$$\theta^* = \underset{\theta}{\operatorname{argmin}} (\omega_r J_r(\theta) + \omega_{\text{data}} J_{\text{data}}(\theta)),$$

with  $\omega_r$  and  $\omega_{\text{data}}$  some weights to balance the terms of the loss function.

### 5.2.2 Sobolev training for PINNs

As presented in [Section 5.1](#), PINNs approximate the PDE solution by directly incorporating the equations into their training. Despite their effectiveness, these models can sometimes struggle to learn correctly, especially when the solution or its derivatives are complicated. The authors of [[SJHH21](#)] have proposed an approach called Sobolev training to try and overcome these difficulties, in the case of solutions with high regularity. Note that, in this case, the source term needs to be differentiable to ensure sufficient regularity and to compute the additional term. This method simply imposes constraints not only on the solutions themselves, but also on their derivatives. In the context of solving the problem [\(5.1\)](#) under consideration, Sobolev training is applied by adding a cost term  $J_{\text{sob}}$  to the initial minimization problem [\(5.2\)](#):

$$\theta^* = \underset{\theta}{\operatorname{argmin}} (\omega_r J_r(\theta) + \omega_{\text{sob}} J_{\text{sob}}(\theta) + \omega_b J_b(\theta) + \omega_{\text{data}} J_{\text{data}}(\theta)), \quad (5.8)$$

with  $J_r$ ,  $J_b$  and  $J_{\text{data}}$  defined as in [\(5.3\)](#), [\(5.4\)](#) and [\(5.5\)](#) respectively and  $\omega_r$ ,  $\omega_{\text{sob}}$ ,  $\omega_b$  and  $\omega_{\text{data}}$  the weights to balance the different terms of the loss function. The Sobolev loss function  $J_{\text{sob}}$  in [\(5.8\)](#) is defined by

$$J_{\text{sob}}(\theta) = \int_{\mathcal{M}} \int_{\Omega} |\nabla_{\mathbf{x}} (\mathcal{L}(u_\theta(\mathbf{x}, \boldsymbol{\mu}); \mathbf{x}, \boldsymbol{\mu}) - f(\mathbf{x}, \boldsymbol{\mu}))|^2 \mathrm{d}\mathbf{x} \mathrm{d}\boldsymbol{\mu}, \quad (5.9)$$

where the integral is estimated by the Monte-Carlo method, similarly to the other loss functions.

**Remark 14.** *Please note that this training has an additional cost, due to the calculation of higher-order derivatives. For example, considering exactly the network from the first 2D test case [Section 7.5.1](#) over 1000 epochs with Adam, Sobolev training takes two times longer than standard training (79 seconds versus 38 seconds).*

### 5.2.3 Overcoming the spectral bias

Multiple ways of overcoming the spectral bias of MLPs are available. For instance, in [[Tea20](#)], the authors introduce Fourier features to improve the network, while the authors of [[DHMM24](#)] rely on a domain decomposition-based approach.

In this work, when dealing with high-frequency solutions (i.e., solutions with more than three wavelengths propagating), we use the Fourier features from [[Tea20](#)]. It relies on modifying the input of the neural network. Indeed, the prior  $u_\theta$  is now defined, for all  $\mathbf{x} \in \Omega$  and  $\boldsymbol{\mu} \in \mathcal{M}$ , as

$$u_\theta(\mathbf{x}, \boldsymbol{\mu}) = w_\theta(\mathbf{x}, \boldsymbol{\mu}; \sin(\pi a_1 \mathbf{x}), \cos(\pi b_1 \mathbf{x}), \dots, \sin(\pi a_{n_f} \mathbf{x}), \cos(\pi b_{n_f} \mathbf{x})),$$

where  $(a_i)_i \in \mathbb{R}^{n_f}$  and  $(b_i)_i \in \mathbb{R}^{n_f}$  are additional trainable parameters. This makes it possible to learn higher-frequency solutions, by also learning the frequency itself. This MLP with Fourier features (MLP w/ FF) needs the same parameters as the classical MLP (defined in [Remark 15](#)), but also the number  $n_f$  of Fourier features.

## 6 Implementation details

Before moving on to numerical experiments, we discuss some practical details regarding the implementation of the methods introduced in Sections 2 and 3. In Section 6.1, we first look at how to effectively plug the PINN prediction in the FEM solver. Then, in Section 6.2, we discuss the imposition of boundary conditions in the two proposed methods.

The tools used to implement the methods and obtain the numerical results in Section 7 are, on the one hand, PyTorch [Pea19] and ScimBa<sup>1</sup> for the prior construction, in particular the implementation of the PINN, and, on the other hand, FEniCS [Aea15] (version 2019.1.0) or FEniCSx [Bea23, SDRW22, SBRW22, ALØ<sup>+</sup>14] (version 0.8) for the implementation of the finite element methods. For mesh generation, use either FEniCS/FEniCSx or mshr mesh generators. Note that, we define the characteristic mesh size  $h$  as the length of the longest edge, for the space dimension  $d \in \{1, 2, 3\}$ .

Moreover, in Section 7, we do not specify the training times of the networks for each test case. To give the reader a rough idea, PINN training takes less than ten minutes on a NVIDIA RTX 2000 (8GB VRAM). For cases using the LBFGS optimizer, training takes a little longer, but remains under an hour. Regarding the EF part, the CPU considered depends on the numerical cost required for each test case; we ran the solvers using either Intel I7-13800H or AMD EPYC 7713.

**Remark 15.** *To construct the PINN, we require some hyperparameters for the MLP and the training phase. In particular, the MLP activation function is denoted by  $\sigma$ , and “layers” represents a sequence of integers describing the number of neurons associated with each layer of the MLP. For training, we will denote by “lr” the learning rate and  $n_{\text{epochs}}$  the number of epochs considered, as well as “decay” the multiplicative factor of the learning rate decay considered every 20 epochs thanks to PyTorch’s StepLR scheduler. Unless otherwise specified, the batch size will correspond to the number of collocation points chosen. The Adam optimizer [KB15] will be used to train the network, but in some cases, we will switch to the LBFGS optimizer [SY06] at the  $n_{\text{switch}}$ -th epoch.*

### 6.1 Using PINN prediction effectively

To be effective, our methods will in practice depend on the quality of the approximation of the prior’s derivatives, computed from automatic differentiation, and its precise integration on the domain.

**Automatic differentiation.** It is important to use the automatic differentiation offered by neural networks, enabling exact (in the sense of machine precision) derivative computation without having to manipulate complex symbolic expressions. In particular, in the context of PINNs, automatic differentiation will play a fundamental role in integrating the PDE under consideration. This automatic differentiation will enable the two improved finite element methods to use the exact derivatives of the prior  $u_\theta$  and thus avoid introducing an additional error in the computation of the derivative.

**Numerical integration.** Our approaches require the numerical integration, in the weak problem, of several functions with a closed-form expression (most notably the prior; also the right-hand side of the PDE for instance). This integration has to be done with sufficient precision for our methods to be effective. Thus, in e.g. the additive approach, a quadrature rule with a higher degree than the traditional FEM has to be applied to discretize the term  $l(v_h) - a(u_\theta, v_h)$  in (2.3). This point, and the required degree of the quadrature rule, will be studied in more detail in the first 2D test case considered in Section 7.5.1.

**Remark 16.** *In practice, the source term in the additive approach will be computed in the strong way. For instance, in the case of the Laplacian equation with  $u_\theta = 0$  on  $\partial\Omega$ , the term*

$$l(v_h) - a(u_\theta, v_h) = \int_{\Omega} f(\mathbf{x})v_h(\mathbf{x}) \, d\mathbf{x} - \int_{\Omega} \nabla u_\theta(\mathbf{x}) \cdot \nabla v_h(\mathbf{x}) \, d\mathbf{x}$$

will be replaced by

$$\int_{\Omega} (f(\mathbf{x}) + \Delta u_\theta(\mathbf{x}))v_h(\mathbf{x}) \, d\mathbf{x}.$$

If  $u_\theta$  is not equal to zero on  $\partial\Omega$ , one needs to include a boundary term.

<sup>1</sup><https://gitlab.inria.fr/scimba/scimba>

## 6.2 Imposing boundary conditions

In this section, we focus on the crucial question of imposing boundary conditions. We first look at this problem in the context of the additive approach presented in [Section 2](#) and then in the context of the multiplicative approach presented in [Section 3](#).

For simplicity, this section focuses on (non-homogeneous) Dirichlet conditions. For our enriched FEM, just like in classical FEM, these boundary conditions are imposed by manually eliminating essential dofs, see [\[EG04\]](#), and more precisely by modifying the matrix and the right-hand side of the linear system. This approach is not needed for Neumann and Robin conditions.

### 6.2.1 Additive approach

In this first approach, if our Dirichlet problem satisfies  $u = g$  on  $\partial\Omega$ , then  $p_h^+$  has to satisfy

$$p_h^+ = g - u_\theta \quad \text{on } \partial\Omega,$$

with  $u_\theta$  the PINN prior. This non-homogeneous boundary condition becomes homogeneous as soon as  $u_\theta$  is exact at the boundary, or, in other words, as soon as the boundary conditions are imposed exactly in the PINN, as presented in [Section 5.2.1](#).

**Remark 17.** *However, in the case of curved geometries (e.g. disks) where the meshes do not coincide with the boundary of the geometry, problems occur when  $k > 2$ , and especially on coarse meshes. In the numerical results, to avoid this problem and check the error estimates, we assume that  $g = u$  on  $\partial\Omega_h$  where  $\Omega_h$  is the domain covered by the mesh. Furthermore, we need to be careful because even if, in PINN, the conditions are imposed exactly (as shown in [Section 5.2.1](#)), the prediction  $u_\theta$  will not be exact on the approximate boundary  $\partial\Omega_h$ . We made this choice here to simplify the problem, but in practice, solutions exist to improve the quality of the results.*

### 6.2.2 Multiplicative approach

In this second method, the boundary conditions are a bit more complex to handle. In the [Section 3](#), we have denoted by

$$u_{h,M}^\times = u_{\theta,M} p_h^\times$$

the solution obtained by the multiplicative approach to the modified problem [\(3.1\)](#) with the prior  $u_{\theta,M} = u_\theta + M$ . Therefore, we can recover the solution  $u_h^\times$  of the original problem [\(0.1\)](#) by setting  $u_h^\times = u_{h,M}^\times - M$ .

**Standard PINN.** In the case where our prior  $u_\theta$  is the prediction of a standard PINN, as presented in [Section 5.1](#), the boundary conditions are not imposed exactly. Thus, if our problem satisfies  $u = g$  on  $\partial\Omega$ , then  $p_h^\times$  has to satisfy

$$p_h^\times = \frac{g + M}{u_{\theta,M}} \quad \text{on } \partial\Omega.$$

**PINN with exact BC.** We now tackle the case where we exactly impose the boundary conditions in the PINN, as presented in [Section 5.2.1](#). Then, supposing that  $M > 0$ , we have  $u_{\theta,M} = g + M$  on  $\partial\Omega$  and therefore  $p_h^\times$  has to satisfy

$$p_h^\times = 1 \quad \text{on } \partial\Omega.$$

However, there is a specific case when this condition is not necessarily true. Indeed, if the boundary conditions are homogeneous, then  $g = 0$  and  $u_{\theta,M} = M$  on  $\partial\Omega$ . Considering that  $u_\theta > 0$  in  $\Omega$ ,  $M = 0$  is a possible choice. In this case,  $u_{\theta,M} = u_{\theta,0} = 0$  on  $\partial\Omega$ , and  $u_{h,M}^\times = u_{h,0}^\times = u_h^\times$  automatically satisfies the boundary conditions. Hence, imposing a boundary condition on  $p_h^\times$  becomes unnecessary.

**Remark 18.** *In the numerical results of [Section 7.3](#), one of these specific cases is considered in 1D. We will see that leaving  $p_h^\times$  free will give better results here than imposing  $p_h^\times = 1$  on  $\partial\Omega$ . Indeed, this approach leaves more freedom to capture the correct derivatives at the boundary.*

## 7 Numerical results

This section is dedicated to validating the proposed method on several numerical experiments, which are mostly instances of the problem (0.1) with spatial dimension  $d \in \{1, 2, 3\}$ , and with increasing complexity. The idea is to compare, in different ways, the additive and multiplicative approaches presented in Section 2 and Section 3 to the standard finite element method presented in Section 1. The multiplicative approach will only be considered in dimension  $d = 1$ , showing that only in rare cases, and with a good choice of the lifting constant  $M$ , does it provide better results than the additive one. We will also show that the approach proposed in Section 5 for choosing the prior is more efficient than more classical ones, still in the 1D case.

In Section 7.1, we present the two tests that will be performed for each test case. We are interested in three 1D test cases ( $d = 1$ ): the Poisson problem with homogeneous Dirichlet conditions in Section 7.2, a general elliptic system in a convection-dominated regime in Section 7.3 and a non-smooth transmission problem in Section 7.4. We then consider three 2D cases ( $d = 2$ ). We start with a Poisson problem with homogeneous Dirichlet conditions on a square domain in Section 7.5. We then continue with a more complicated elliptic problem, still with Dirichlet conditions and on a square domain, but with parameter-dependent anisotropy, in Section 7.6. In Section 7.7, we return to the 2D Poisson problem, but this time considering mixed boundary conditions on a ring-shaped domain. Finally, we consider a 3D Poisson problem with homogeneous Dirichlet conditions on a cube domain in Section 7.8. To reproduce these results, an open-source code is available on GitHub<sup>2</sup>.

More specifically, in each test case, we will consider the additive approach and use a prior constructed by a PINN (where we mostly impose boundary conditions using a level-set function). We will perform an analysis of error estimates and an evaluation of the gains of the enriched approach over the standard approach on a set of parameters (in  $L^2$  norm). In the first two 1D test cases (Sections 7.2 and 7.3), we will also choose to consider the multiplicative approach. A study in the  $H^1$  semi-norm will be performed in the first 1D and 2D test case (Sections 7.5.1 and 7.8). Additionally, this first 1D test case includes a comparative study between theoretical and numerical constants. In the third 1D test case, we will focus on a problem where the gradient is discontinuous. Subsequently, we will test the approach with other networks, notably: a data-driven network (Section 7.2) that will justify the use of PINNs, networks with weakly imposed conditions (Sections 7.4 and 7.5.3), and an improved PINN where we used Sobolev training (Section 7.5.2). Two studies on the numerical costs of the additive approach will be carried out in Sections 7.5.1 and 7.8.2. We will also perform a brief analysis of the influence of PINN quality, depending on the initialization of the network weights, the number of collocation points considered, and the number of epochs (Sections 7.2.5 and 7.8.4). In some test cases, we will also look at the visualization of the solutions obtained by the different FE approaches.

The first two 1D test cases show that the multiplicative approach only gives better results than the additive approach in certain very specific cases. The first test case also validates the theoretical results, in particular the importance of the derivatives of the prior by validating the choice of PINNs. The third 1D test case extends the theoretical framework to a non-smooth problem, showing that, once again, high-order derivatives have the greatest impact on the results. The first 2D test case (Poisson in a square) introduces training with Sobolev loss and discusses the numerical costs of FE approaches (in terms of number of degrees of freedom). It also discusses the imposition of weak or strong boundary conditions in the network and compares the gains between the “high frequency” and “low frequency” cases. The second 2D test case highlights the flexibility of the method for more complex problems, particularly with parameter-dependent anisotropy. The third 2D test case introduces the approach in a slightly more complex geometric setting (with an annulus) as well as Robin conditions. Finally, the 3D test case validates the approach in a larger dimension and analyzes the associated numerical costs. This time in terms of computation time, considering the complete parametric framework. It also introduces the influence of PINN quality on the final results. More generally, all test cases show that, with the right choice of priors, the additive approach yields significant gains over the standard approach. In particular, the average results obtained over a set of parameters show that for elements  $\mathbb{P}_1$ , this enriched approach provides the same accuracy as the standard approach with meshes 5 (2D anisotropic problem) to 28 (low-frequency Poisson with Sobolev training) times coarser, which leads to a gain in computation time (from a purely online perspective). The 3D test case takes a closer look at execution times, particularly in the parametric context in which the enriched approach is developed. More specifically, gains in terms of mesh size make it possible to achieve a speed-up from 19 parameter sets to reach a relative accuracy of  $10^{-3}$  (in the  $L^2$  norm).

### 7.1 Setup of the numerical experiments

For each of the proposed test cases, we consider a parametric problem, on which we train a PINN as presented in Section 5.1, to resolve it on a set of parameters, denoted by  $\mathcal{M}$ . Let  $p = \dim(\mathcal{M})$  be the number of parameters, and

---

<sup>2</sup><https://github.com/flecourtier/EnrichedFEMUsingPINNs>

consider a set  $\mathcal{S}$  of  $n_p$  parameter instances:

$$\mathcal{S} = \left\{ \boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(n_p)} \right\},$$

with, for  $j = 1, \dots, n_p$ ,

$$\boldsymbol{\mu}^{(j)} = \left( \mu_1^{(j)}, \dots, \mu_p^{(j)} \right) \in \mathcal{M}.$$

In the following, we denote by  $u^{(j)}$  a reference solution to problem (5.1) for a given parameter  $\boldsymbol{\mu}^{(j)}$  and by  $u_h^{(j)}$  the solution obtained by the standard finite element method (1.3), where  $V_h$  is the  $\mathbb{P}_k$  Lagrange space defined in (1.2) and  $h$  is the characteristic mesh size. We also denote by  $u_\theta^{(j)}$  the solution obtained by the parametric PINN, and by  $u_{h,+}^{(j)}$  the solution obtained by the additive approach (2.3) with  $V_h^+$  the  $\mathbb{P}_k$  Lagrange space defined in (2.1). In some test cases, we also consider the solution  $u_{h,M}^{(j)}$  of the multiplicative approach (3.4) with  $V_h^\times$  the  $\mathbb{P}_k$  Lagrange space defined in (3.3), depending on the lifting constant  $M$ .

**Remark 19.** *In the following, to estimate the error, we consider the reference solution to be either an analytical solution or a solution obtained with a very fine mesh and a high polynomial degree. More precisely, we need the characteristic size  $h_{\text{ref}}$  associated with the reference mesh to be much smaller than the size associated with the current mesh  $h$ , i.e.  $h_{\text{ref}} \ll h$ , and we will consider  $k_{\text{ref}} = 3$  the polynomial degree associated with the reference solution.*

In each test case, we investigate two aspects; the first in Section 7.1.1 involves verifying the error estimates and the second in Section 7.1.2 is the evaluation of the gains achieved by the proposed methods compared with the standard one.

### 7.1.1 Error estimates

Consider a small set  $\mathcal{S}$  with  $n_p = 2$  parameter instances. Given a fixed parameter  $\boldsymbol{\mu}^{(j)}$ ,  $j = 1, 2$ , we start by testing the error estimates obtained in Theorem 6 for the additive approach. In the case  $d = 1$ , we will also be interested in the error estimates in Theorem 9 for the multiplicative approach. By varying the mesh size  $h$ , we then estimate the errors obtained with the two methods. To evaluate these errors, we compare the approximations to the reference solution  $u^{(j)}$  (see Remark 19). We then define by

$$e_h^{(j)} = \frac{\|u^{(j)} - u_h^{(j)}\|_{L^2}}{\|u^{(j)}\|_{L^2}} \quad \text{and} \quad e_\theta^{(j)} = \frac{\|u^{(j)} - u_\theta^{(j)}\|_{L^2}}{\|u^{(j)}\|_{L^2}}, \quad (7.1)$$

the  $L^2$  relative error obtained for the standard FEM and the PINN respectively. We further define,

$$e_{h,+}^{(j)} = \frac{\|u^{(j)} - u_{h,+}^{(j)}\|_{L^2}}{\|u^{(j)}\|_{L^2}} \quad \text{and} \quad e_{h,M}^{(j)} = \frac{\|u^{(j)} - u_{h,M}^{(j)}\|_{L^2}}{\|u^{(j)}\|_{L^2}}, \quad (7.2)$$

the  $L^2$  relative errors obtained for the additive and multiplicative approach (depending on the lifting constant  $M$ ), respectively. Relative errors in the semi-norm  $H^1$  can be defined in the same way.

### 7.1.2 Gains achieved with the enriched bases

As we have trained the network to be parameter-dependent to predict a solution for a set of parameters, we are interested in the average gains we obtain with our enriched approaches compared to the PINN and the standard FEM. More precisely, for a fixed mesh size  $h$  and a fixed polynomial degree  $k$ , for a set  $\mathcal{S}$  of  $n_p$  parameter instances, the numerical gains obtained by the additive approach on PINN and standard FEM are respectively defined for  $j = 1, \dots, n_p$  by:

$$G_{+,\theta}^{(j)} = \frac{e_\theta^{(j)}}{e_{h,+}^{(j)}} \quad \text{and} \quad G_+^{(j)} = \frac{e_h^{(j)}}{e_{h,+}^{(j)}}, \quad (7.3)$$

with  $e_\theta^{(j)}$ ,  $e_h^{(j)}$  and  $e_{h,+}^{(j)}$  respectively the  $L^2$  relative errors obtained with the PINN, the standard FEM and the additive approach, defined in Section 7.1.1. Similarly, the theoretical gains obtained by the multiplicative approach (depending on the lifting constant  $M$ ) on PINN and standard FEM are respectively defined for  $j = 1, \dots, n_p$  by:

$$G_{M,\theta}^{(j)} = \frac{e_\theta^{(j)}}{e_{h,M}^{(j)}} \quad \text{and} \quad G_M^{(j)} = \frac{e_h^{(j)}}{e_{h,M}^{(j)}}, \quad (7.4)$$

with  $e_{h,M}^{(j)}$  the  $L^2$  relative error obtained with the multiplicative approach (depending on the lifting constant  $M$ ), defined in [Section 7.1.1](#). Therefore, we will be interested in the minimum, maximum, mean and standard deviation obtained on the following samples:

$$G_{+, \theta} = \{G_{+, \theta}^{(1)}, \dots, G_{+, \theta}^{(n_p)}\} \quad \text{and} \quad G_+ = \{G_+^{(1)}, \dots, G_+^{(n_p)}\}, \quad (7.5)$$

which respectively represent the gains obtained with our additive approach over PINN and over standard FEM on the set  $\mathcal{S}$  of parameter instances. In the same way, we define  $G_{M, \theta}$  and  $G_M$ , which respectively represent the gains obtained with our multiplicative approach over PINN and over standard FEM on  $\mathcal{S}$  by:

$$G_{M, \theta} = \{G_{M, \theta}^{(1)}, \dots, G_{M, \theta}^{(n_p)}\} \quad \text{and} \quad G_M = \{G_M^{(1)}, \dots, G_M^{(n_p)}\}. \quad (7.6)$$

## 7.2 1D Poisson problem

In this section, we will consider the problem [\(0.1\)](#) in its most simplified Poisson form, with homogeneous Dirichlet boundary conditions. In the 1D case ( $d = 1$ ), we have,

$$\begin{cases} -\partial_{xx} u = f, & \text{in } \Omega \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \end{cases} \quad (7.7)$$

with  $\Omega = (0, 1)$ ,  $\partial\Omega$  its boundary and  $\mathcal{M} \subset \mathbb{R}^p$  the parameter space (with  $p$  the number of parameters). We prescribe a family of exact solutions depending on the parameter vector  $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3) \in \mathcal{M} = [0, 1]^3$  ( $p = 3$  parameters) defined by

$$u(x, \boldsymbol{\mu}) = \mu_1 \sin(2\pi x) + \mu_2 \sin(4\pi x) + \mu_3 \sin(6\pi x).$$

Note that the associated right-hand side  $f$  in [\(7.7\)](#) also depends on  $\boldsymbol{\mu}$ . This problem is thus four-dimensional: one dimension in space and three dimensions for the parameters  $\boldsymbol{\mu}$ .

For this first test case, we construct two priors, as detailed in [Section 7.2.1](#). The first one, denoted by  $u_\theta$ , is built from a PINN as presented in [Section 5](#). The second one, denoted by  $u_\theta^{\text{data}}$  is constructed only from data (obtained from the analytical solution). The aim is to show that using physics-informed training to construct the prior leads to better results than data-driven training. In this test case, we also compare the additive and multiplicative approaches (presented for several values of  $M$ ), but only with  $k = 1$  polynomial order to remain concise.

We first present the error estimates obtained with the PINN prior in [Section 7.2.2](#). First, we check the orders of convergence of the two enriched approaches. Then, we verify the results expected in [Section 4](#) by comparing the theoretical gain constants of the additive and multiplicative approaches. Afterwards, in [Section 7.2.3](#), we compare, for a given parameter, the derivatives obtained with the two priors and analyze the associated gains. Then, we evaluate the gains obtained with the two priors in [Section 7.2.4](#) on a sample of parameters. Finally, in [Section 7.2.5](#), we analyze the influence of the weight initialization of the PINN prior on the gains obtained with our enriched approaches.

**Remark 20.** In the following, the characteristic mesh size is  $h = \frac{1}{N-1}$ , where  $N$  represents the number of considered nodes.

### 7.2.1 Construction of the two priors

The hyperparameters used to construct the two priors are presented in [Table 1](#). We discuss below the specific differences in training both priors.

Network - MLP		Training				Loss weights							
<i>layers</i>	20, 80, 80, 80, 20, 10	<i>lr</i>	9e-2	<i>n<sub>epochs</sub></i>	10 000	PINN prior $u_\theta$				Data prior $u_\theta^{\text{data}}$			
<i><math>\sigma</math></i>	sine	<i>decay</i>	0.99			$\omega_r$	1	$\omega_{\text{data}}$	0	$\omega_r$	0	$\omega_{\text{data}}$	1
		<i><math>N_{\text{col/data}}</math></i>	5 000			$\omega_b$	0	$\omega_{\text{sob}}$	0	$\omega_b$	0	$\omega_{\text{sob}}$	0

Table 1: Network, training parameters ([Remark 15](#)) and loss weights for  $u_\theta$  and  $u_\theta^{\text{data}}$  in the 1D Poisson problem. Considering  $N_{\text{col}}$  collocation points for the PINN prior and  $N_{\text{data}}$  data for the data prior.

**Physics-informed training.** For the first prior, we will consider a parametric PINN, depending on the problem parameters  $\boldsymbol{\mu}$ , where we exactly impose the Dirichlet boundary conditions as presented in [Section 5.2.1](#) and without using data in training. Thus, we construct  $u_\theta$  as in [\(5.2.1\)](#) with the level-set function  $\varphi$  defined by

$$\varphi(x) = x(x - 1),$$

which vanishes exactly on  $\partial\Omega$  and  $g = 0$ . Since we impose the boundary conditions by using the level-set function, we will only consider the residual loss  $J_r$  approached by a Monte-Carlo method as defined in [\(5.6\)](#) with  $N_{\text{col}} = 5000$  collocation points uniformly chosen on  $\Omega \times \mathcal{M}$ . The hyperparameters are given in [Table 1](#); we use the Adam optimizer [\[KB15\]](#).

**Data-driven training.** For the second prior considered, noted  $u_\theta^{\text{data}}$ , a network is trained only on the data (constructed from the analytical solution). Therefore, we will only consider the data loss  $J_{\text{data}}$  defined in [\(5.5\)](#), considering  $N_{\text{data}} = 5000$  points. As for the physics-informed training, we still consider the hyperparameters defined in [Table 1](#) and construct the prior  $u_\theta^{\text{data}}$  in the same way using the level-set function.

## 7.2.2 Error estimates — with the PINN prior

In this section, we look at the theoretical results of the additive and multiplicative approaches, considering the PINN prior  $u_\theta$ . First, we check the orders of convergence of [Theorems 6](#) and [9](#) (in the  $L^2$  norm), associated with both methods. Next, we numerically verify [Theorem 12](#), showing that the multiplicative correction converges, for sufficiently large  $M$ , towards the additive correction (in both the  $L^2$  norm and  $H^1$  semi-norm).

**Convergence rate.** We test the error estimates of [Theorems 6](#) and [9](#) for the following two sets of parameters:

$$\boldsymbol{\mu}^{(1)} = (0.3, 0.2, 0.1) \quad \text{and} \quad \boldsymbol{\mu}^{(2)} = (0.8, 0.5, 0.8),$$

with the PINN prior  $u_\theta$ . For  $j \in \{1, 2\}$ , the aim is to compare, by varying the mesh size  $h$ , the  $L^2$  relative errors  $e_h^{(j)}$  obtained with the standard FEM, defined in [\(7.1\)](#),  $e_{h,+}^{(j)}$  obtained with the additive approach, defined in [\(7.2\)](#) and  $e_{h,M}^{(j)}$  obtained with the multiplicative approach (taking  $M = 3$  and  $M = 100$ ), defined in [\(7.2\)](#). The results are presented in [Figure 2](#) for polynomial orders  $k = 1$  and  $k = 2$ , with  $h$  depending on the number of nodes  $N \in \{16, 32, 64, 128, 256\}$  as presented in [Remark 20](#). We also present in [Figure 3](#) the results obtained for the semi-norm  $H^1$ .

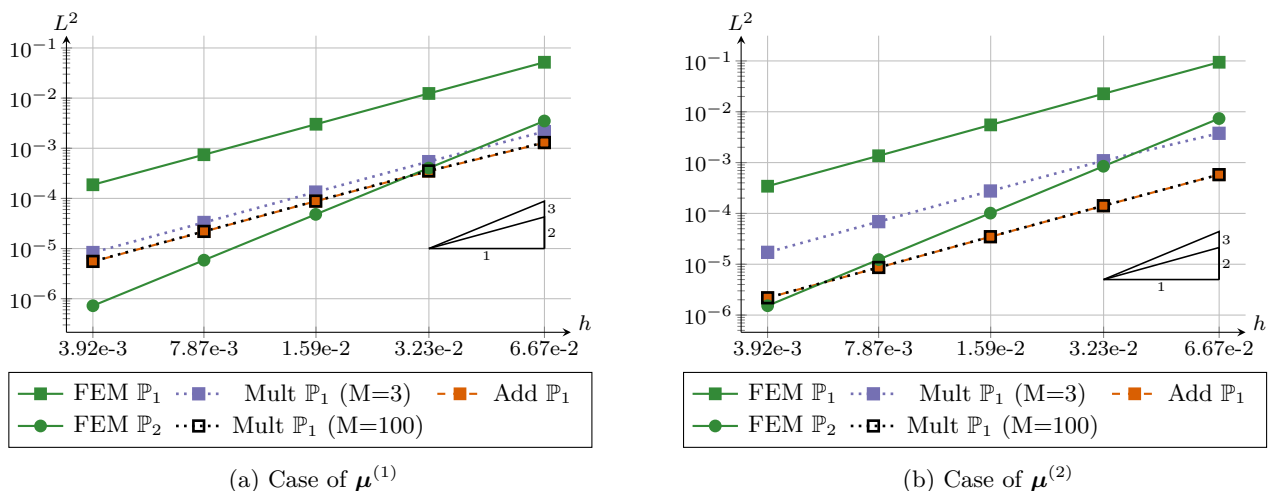


Figure 2: Considering the *1D Poisson problem* and the PINN prior  $u_\theta$ . Left – Considering  $\boldsymbol{\mu}^{(1)}$ .  $L^2$  relative error on  $h$  obtained with standard FEM  $e_h^{(1)}$  (solid lines) with  $k = 1$  and  $k = 2$ , the additive approach  $e_{h,+}^{(1)}$  (dashed lines) with  $k = 1$  and the multiplicative approach  $e_{h,M}^{(1)}$  (dotted lines) with  $k = 1$  ( $M = 3$  and  $M = 100$ ). Right – Same for  $\boldsymbol{\mu}^{(2)}$ .

The results of [Figures 2](#) and [3](#) show that all the enriched finite elements increase the accuracy of the method and that they also converge at the same rate as the classical approach (i.e., for polynomial approximation of order  $k = 1$ , the convergence order in the  $L^2$  norm is 2 and in the semi-norm  $H^1$  is 1). Furthermore, the theoretical analysis (which showed that the multiplicative correction has the same error as the additive one when  $M \rightarrow \infty$ ) is confirmed for both

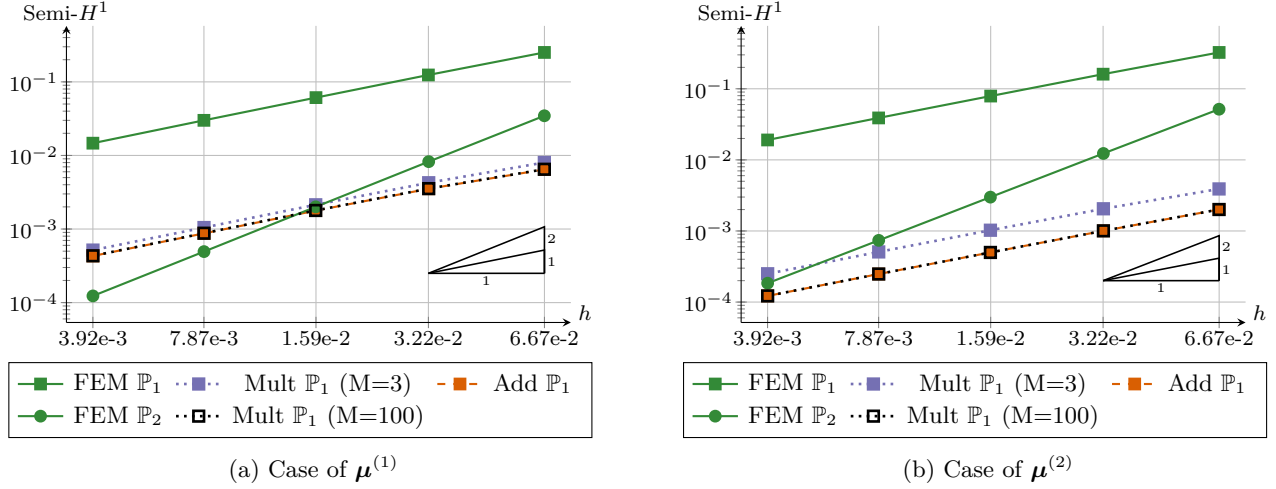


Figure 3: Considering the *1D Poisson problem* and the PINN prior  $u_\theta$ . Left – Considering  $\mu^{(1)}$ . Semi- $H^1$  relative error on  $h$  obtained with standard FEM (solid lines) with  $k = 1$  and  $k = 2$ , the additive approach (dashed lines) with  $k = 1$  and the multiplicative approach (dotted lines) with  $k = 1$  ( $M = 3$  and  $M = 100$ ). Right – Same for  $\mu^{(2)}$ .

sets of parameters. In addition, [Figures 2](#) and [3](#) also shows that this multiplicative enrichment can be less efficient for small  $M$  when the  $(k + 1)^{\text{th}}$  derivative of the solution is large. Indeed, for the first parameter considered in [Figures 2a](#) and [3a](#), for which the second derivative takes lower values, we observe that the multiplicative approach with small  $M$  is closer to the additive one than for the second set of parameters considered in [Figures 2b](#) and [3b](#), for which the derivatives are larger. Moreover, it seems that we gain almost one order of interpolation with the additive approach: the additive method with polynomial order  $k = 1$  gives relative errors close to the original FEM with  $k = 2$ , although the rate of convergence is different.

**Gain constants.** We consider the first parameter  $\mu^{(1)}$  and the PINN prior  $u_\theta$ . We now evaluate the gain constants  $C_{\text{gain}}^+$ ,  $C_{\text{gain},H^1}^{\times,M}$  and  $C_{\text{gain},L^2}^{\times,M}$  (for different values of  $M$ ), which are respectively defined in [Theorems 6](#) and [9](#) for the additive and multiplicative approaches. The idea is to check the convergence of the two multiplicative gain constants towards the additive one, as proven in [Theorem 12](#). The results are presented in [Figure 4](#) in  $L^2$  norm and  $H^1$  semi-norm, considering  $M > \min |u_\theta|$ .

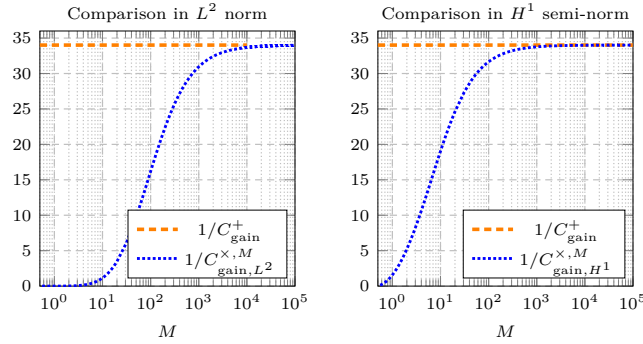


Figure 4: Considering the *1D Poisson problem* with  $\mu^{(1)}$ ,  $k = 1$  and the PINN prior  $u_\theta$ . Left – Convergence of [Theorem 12](#) with the  $L^2$  error. Right – Convergence of [Theorem 12](#) with the semi- $H^1$  error.

[Figure 4](#) shows that the multiplicative gain constant converges to the additive gain constant when  $M$  increases, as expected in the theoretical results of [Theorem 12](#). Furthermore, we can compare the expected theoretical gains with the numerical gains obtained. For the additive approach, it would appear that the expected theoretical gain (approximately 34.01) corresponds well with the numerical results of [Tables 2](#) and [3](#) (left subtable). As for the multiplicative approach, it appears that these constants align well when  $M$  becomes large; however, for smaller  $M$ , the theoretical error estimates seem suboptimal. For example, for  $M = 3$ , we find a theoretical gain of approximately 8.22 in semi-norm  $H^1$  compared

to the 30 obtained in practice in Table 3 (left subtable). Moreover, in Figure 4, we see that the theoretical gain (in  $L^2$  norm and  $H^1$  semi-norm) converges to 0 when  $M$  tends to  $\min |u_\theta|$ , whereas in practice the numerical gains obtained have never been less than 1. More precisely, for this test case in the  $H^1$  semi-norm, we converge to 2.45 numerically when  $M$  is small.

### 7.2.3 Derivatives — with both priors

To better explain the results of Section 7.2.2, we compare the solution, the first- and second-order derivatives between the exact solution and the prediction of both priors, for selected parameter  $\mu^{(1)}$ . Figures 5 and 6 respectively present this comparison for the PINN prior  $u_\theta$  and the data prior  $u_\theta^{\text{data}}$ . We also compare the relative errors and gains obtained with these two priors for  $N \in \{16, 32\}$  in Tables 2 and 3, respectively in  $L^2$  norm and  $H^1$  semi-norm. More precisely for the  $L^2$  norm, we evaluate the additive error  $e_{h,+}^{(1)}$  and the additive gain on FEM  $G_+^{(1)}$ , respectively defined in (7.2) and (7.3), for both PINN and data priors. We also evaluate the multiplicative error  $e_{h,M}^{(1)}$  and the multiplicative gain on FEM  $G_M^{(1)}$  defined in (7.2) and (7.4), for both priors, with  $M = 3$  and  $M = 100$ .

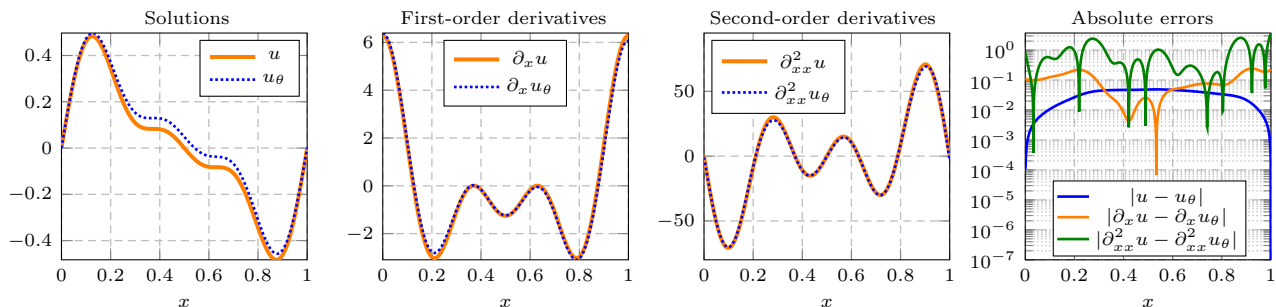


Figure 5: Considering the 1D Poisson problem with  $\mu^{(1)}$  and the PINN prior  $u_\theta$ , comparison between analytical solution and network prediction. From left to right: solution; first derivative; second derivative; errors.

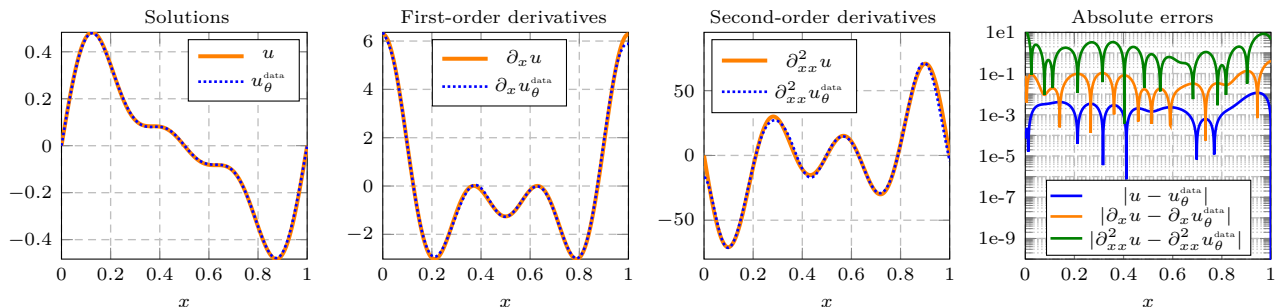


Figure 6: Considering the 1D Poisson problem with  $\mu^{(1)}$  and the data prior  $u_\theta^{\text{data}}$ , comparison between analytical solution and network prediction. From left to right: solution; first derivative; second derivative; errors.

The results reported in Figures 5 and 6 and Tables 2 and 3 show that, even if the approach chosen to build the prior (physics-informed or data-driven training) gives a good approximation of the solution, the important point lies in the derivatives and mainly in the second-order derivatives, which are clearly better learned by PINNs. Indeed, while the enriched FEM solution is more accurate using the PINN prior (Tables 2 and 3), we see from Figures 5 and 6 that the raw PINN approximates the solution  $u$  less accurately than the raw data-driven the solution, but the PINN better approximates the derivatives. As the error of the enriched FEM is mainly due to the  $(k + 1)^{\text{th}}$  derivatives of the network being close to the  $(k + 1)^{\text{th}}$  derivatives of the solution, this explains why the enriched FEM with data prior does not perform as well as with a PINN prior. Therefore, PINNs have two advantages: they do not require training data and give better results. Their main shortcoming is that training takes longer; data could be used in addition to physics-informed training to speed up the process, but we do not expect it to improve the accuracy of the prior derivatives. However, we mention that if data are available for first- and second-order derivatives, it could also be used to improve a purely data-driven prior.

		method	N	PINN prior $u_\theta$		Data prior $u_\theta^{\text{data}}$	
FEM				error	gain	error	gain
N	error	Add	16	$1.29 \cdot 10^{-3}$	40.34	$3.51 \cdot 10^{-3}$	14.78
			32	$3.49 \cdot 10^{-4}$	35.41	$8.8 \cdot 10^{-4}$	14.06
16	$5.18 \cdot 10^{-2}$	Mult (M=3)	16	$2.15 \cdot 10^{-3}$	24.13	$3.38 \cdot 10^{-3}$	15.36
	$1.24 \cdot 10^{-2}$		32	$5.41 \cdot 10^{-4}$	22.86	$8.53 \cdot 10^{-4}$	14.51
32		Mult (M=100)	16	$1.3 \cdot 10^{-3}$	39.84	$3.5 \cdot 10^{-3}$	14.83
			32	$3.53 \cdot 10^{-4}$	35.08	$8.78 \cdot 10^{-4}$	14.09

Table 2: Considering the *1D Poisson problem* with  $\mu^{(1)}$ ,  $k = 1$  and  $N \in \{16, 32\}$ . Left –  $L^2$  relative error obtained with FEM. Right – Considering the PINN prior  $u_\theta$  and the data prior  $u_\theta^{\text{data}}$ ,  $L^2$  relative errors and gains with respect to FEM, obtained with our methods. Our methods : additive approach, multiplicative approach with  $M = 3$  and  $M = 100$ .

		method	N	PINN prior $u_\theta$		Data prior $u_\theta^{\text{data}}$	
FEM				error	gain	error	gain
N	error	Add	16	$6.45 \cdot 10^{-3}$	38.94	$1.74 \cdot 10^{-2}$	14.45
			32	$3.53 \cdot 10^{-3}$	35.21	$8.91 \cdot 10^{-3}$	13.93
16	$2.51 \cdot 10^{-1}$	Mult (M=3)	16	$8.02 \cdot 10^{-3}$	31.31	$1.64 \cdot 10^{-2}$	15.32
	$1.24 \cdot 10^{-1}$		32	$4.26 \cdot 10^{-3}$	29.16	$8.51 \cdot 10^{-3}$	14.59
32		Mult (M=100)	16	$6.49 \cdot 10^{-3}$	38.7	$1.74 \cdot 10^{-2}$	14.48
			32	$3.54 \cdot 10^{-3}$	35.04	$8.9 \cdot 10^{-3}$	13.95

Table 3: Considering the *1D Poisson problem* with  $\mu^{(1)}$ ,  $k = 1$  and  $N \in \{16, 32\}$ . Left – Semi- $H^1$  relative error obtained with FEM. Right – Considering the PINN prior  $u_\theta$  and the data prior  $u_\theta^{\text{data}}$ , Semi- $H^1$  relative errors and gains with respect to FEM, obtained with our methods. Our methods : additive approach, multiplicative approach with  $M = 3$  and  $M = 100$ .

#### 7.2.4 Gains achieved with the additive and multiplicative approaches – with both priors

Considering a set  $\mathcal{S}$  of  $n_p = 100$  parameter instances, we now evaluate the gains  $G_{+, \theta}$  and  $G_+$  defined in (7.5) with the PINN prior  $u_\theta$  and with the data prior  $u_\theta^{\text{data}}$ . We also compute  $G_{M, \theta}$  and  $G_M$ , defined in (7.6), similarly for both priors. For fixed polynomial order  $k = 1$  and  $N \in \{20, 40\}$ , the results with the physics-informed prior  $u_\theta$  and the data-driven prior  $u_\theta^{\text{data}}$  are respectively presented in Tables 4 and 5.

method	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
Add	20	10.47	251.6	92.19	50.96	26.49	271.92	140.74	55.16
	40	42.12	969.51	362.88	195.53	23.4	258.37	134.11	51.86
Mult (M=3)	20	5.19	106.29	25.34	20.57	3.93	163.81	42.72	34.93
	40	17.25	443.21	99.66	85.19	3.16	149.83	40.22	33.75
Mult (M=100)	20	10.48	253.57	92.59	51.9	26.29	270.87	140.82	55
	40	42.17	979.91	364.38	199.36	23.28	257.71	134.15	51.62

Table 4: Considering the *1D Poisson problem* and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  error of our methods with respect to PINN by taking  $k = 1$ . Right – Gains in  $L^2$  error of our methods with respect to FEM by taking  $k = 1$ . Our methods : additive approach, multiplicative approach with  $M = 3$  and  $M = 100$ .

The previous results indicate that the average gain provided by the enriched FE with the PINN prior is significant, particularly when using the additive approach. These findings also confirm the behavior of the multiplicative prior

method	N	Gains in $L^2$ rel error of our method w.r.t. Data Network				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
Add	20	1.34	10.95	3.33	2.09	6.91	60.85	26.12	12.45
	40	4.35	34.37	11.19	6.97	7.13	39.34	20.55	7.94
Mult (M=3)	20	0.85	10.4	2.56	1.98	3.54	53.99	19.66	11.11
	40	2.64	35.3	8.68	6.98	2.18	39.06	15.66	8.25
Mult (M=100)	20	1.35	10.96	3.33	2.09	6.9	60.73	26.11	12.43
	40	4.35	34.41	11.19	6.97	7.13	39.38	20.55	7.93

Table 5: Considering the *1D Poisson problem* and the data prior  $u_\theta^{\text{data}}$ . Left – Gains in  $L^2$  error of our methods with respect to data Network by taking  $k = 1$ . Right – Gains in  $L^2$  error of our methods with respect to FEM by taking  $k = 1$ . Our methods : additive approach, multiplicative approach with  $M = 3$  and  $M = 100$ .

method for varying values of  $M$ . In contrast, when applied with data-driven instead of physics-informed training, the same method does not yield similarly favourable results. Consequently, in the experiments we perform below, we only employ the PINN prior.

Furthermore, it is important to note that we cannot expect enriched methods to have generalization or extrapolation properties for parameters  $\mu \notin \mathcal{M}$ . Indeed, these potential extrapolation properties would depend on those of the PINN. Since the PINN is a very smooth function, we might expect it to give reasonable results near the boundaries of  $\mathcal{M}$ . Intuitively, we might also expect that the further the parameters are from the training domain, the worse the results will be. However, there is no way to control this potential extrapolation property. For this test case, considering the additive approach with the PINN prior,  $N = 20$  and a sample of 100 parameters in  $[1, 1.1]^3$  (close to  $\mathcal{M}$ ), we find a mean gain of 9.18, compared to the 140.74 in Table 4 for parameters in  $\mathcal{M}$ : the gain has been lowered by a factor of 15. If we move away from the training area with parameters in  $[1.1, 1.5]^3$ , we once again halve this factor with a mean gain of 4.48, confirming the previous intuition.

### 7.2.5 Influence of the PINN initialization

In this section, we focus on the influence of network weight initialization on the quality of our results. To do this, we consider a set of 30 PINN initializations with the hyperparameters defined in Table 1. Thus, for each network, we are interested in the same way as in Section 7.2.4, in the gain of our enriched approaches on a set of  $n_p = 50$  parameter instances. For each prior, we evaluate the minimum, maximum, mean, and standard deviation on this sample. We then regroup these 30 values in Table 6, in the form  $mean \pm std$ , where  $mean$  and  $std$  represent the mean and standard deviation, respectively, on the 30 networks considered.

method	N	Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std
Add	20	$23.04 \pm 5.66$	$306.85 \pm 50.27$	$148.56 \pm 24.05$	$61.57 \pm 10.46$
Add	40	$21.65 \pm 5.19$	$278.92 \pm 44.55$	$141.72 \pm 21.92$	$56.69 \pm 8.62$
Mult (M=3)	20	$6.10 \pm 2.12$	$166.78 \pm 52.32$	$44.54 \pm 10.76$	$31.25 \pm 8.81$
Mult (M=3)	40	$4.76 \pm 1.69$	$154.64 \pm 49.58$	$41.51 \pm 10.18$	$29.74 \pm 8.41$
Mult (M=100)	20	$23.03 \pm 5.65$	$305.61 \pm 50.89$	$148.07 \pm 23.85$	$61.28 \pm 10.31$
Mult (M=100)	40	$21.64 \pm 5.17$	$277.30 \pm 44.34$	$141.29 \pm 21.77$	$56.45 \pm 8.51$

Table 6: Consider the *1D Poisson problem* and a set of 30 PINN priors.  $L^2$  error gains for  $k = 1$  of our methods compared to FEM :  $mean \pm std$  on the 30 networks considered. Our methods : additive approach, multiplicative approach with  $M = 3$  and  $M = 100$ .

In Table 6, we can see that the initialization of the PINN weights has an influence on the gains obtained with the different enriched approaches. More specifically, for the additive approach ( $N = 20$ ), for example, we can see that the minimum gain obtained on the sample averages 23 across the 30 networks considered, with a standard deviation of 6.

It would therefore seem that there is always a gain in using the enriched approach, even if initialization influences its magnitude. On the other hand, even though the maximum is on average 307, the standard deviation is close to 50, which is not negligible. However, the results on the mean column seem close to the results obtained in Table 4. The results obtained with the multiplicative approach also appear to be consistent with those obtained previously.

### 7.3 1D general elliptic system and convection-dominated regime

In this experiment, we consider the problem (0.1) in a more complex form, still in a 1D ( $d = 1$ ) configuration:

$$\begin{cases} \partial_x u - \frac{1}{\text{Pe}} \partial_{xx} u = r, & \text{in } \Omega \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \end{cases}$$

with  $\Omega = (0, 1)$  and  $\partial\Omega$  its boundary,  $r$  the reaction constant term, and  $\text{Pe}$  the Péclet number, describing the ratio between the convection and the diffusion terms. For all  $x \in \Omega$ , the analytical solution reads

$$u(x, \boldsymbol{\mu}) = r \left( x - \frac{e^{\text{Pe} x} - 1}{e^{\text{Pe}} - 1} \right), \quad (7.8)$$

with  $p = 2$  parameters  $\boldsymbol{\mu} = (r, \text{Pe}) \in \mathcal{M} = [1, 2] \times [10, 100]$ .

**Remark 21.** *In the large Péclet regime, i.e., for convection-dominated flows, the classical finite element method may generate oscillations when no specific treatment is applied, see e.g. [JKN18].*

In this test case, we construct only one prior, denoted  $u_\theta$ , built from a PINN as presented in Section 5. We also compare the additive and multiplicative approaches by considering polynomial order  $k = 1$ , and since the solution is positive in  $\Omega$ , we consider  $M = 0$  for the multiplicative approach. We start by evaluating the error in Section 7.3.1, then we compare the derivatives of the PINN prior and compare the different approaches in Section 7.3.2. Finally, we evaluate the gains obtained in Section 7.3.3 on a sample of parameters. As we are dealing with a specific case, we will compare two methods for imposing boundary conditions, as presented in Section 6.2.2: the strong and the weak approaches.

**Remark 22.** *As in Section 7.2, the characteristic mesh size is  $h = \frac{1}{N-1}$ , where  $N$  is the number of nodes considered.*

We consider a parametric PINN, depending on the problem parameters  $\boldsymbol{\mu}$ , where we exactly impose the Dirichlet boundary conditions as presented in Section 5.2.1. We define the prior  $u_\theta$  and the level-set  $\varphi$  as in Section 7.2.

Network - MLP		Training		Loss weights					
<i>layers</i>	40, 40, 40, 40, 40	<i>lr</i>	1e-3	<i>n<sub>epochs</sub></i>	20 000	<i><math>\omega_r</math></i>	1	<i><math>\omega_{\text{data}}</math></i>	0
<i><math>\sigma</math></i>	tanh	<i>decay</i>	0.99			<i><math>\omega_b</math></i>	0	<i><math>\omega_{\text{sob}}</math></i>	0
		<i><math>N_{\text{col}}</math></i>	5 000						

Table 7: Network, training parameters (Remark 15) and loss weights for  $u_\theta$  in the 1D Elliptic case.

Therefore, we will only consider the residual loss  $J_r$  approached by a Monte-Carlo method as defined in (5.6) with  $N_{\text{col}} = 5\,000$  collocation points (uniformly chosen on  $\Omega \times \mathcal{M}$ ) and we seek to solve the minimisation problem (5.2). The hyperparameters are given in Table 7; we use the Adam optimizer [KB15].

#### 7.3.1 Error estimates

We start by testing the error estimates (Theorems 6 and 9) for the following two sets of parameters:

$$\boldsymbol{\mu}^{(1)} = (1.2, 40) \quad \text{and} \quad \boldsymbol{\mu}^{(2)} = (1.5, 90),$$

by considering the PINN prior  $u_\theta$ . For  $j \in \{1, 2\}$ , the aim is to compare for different mesh sizes  $h$ , the  $L^2$  relative errors  $e_h^{(j)}$  obtained with the standard FEM, defined in (7.1),  $e_{h,+}^{(j)}$  obtained with the additive approach and  $e_{h,M}^{(j)}$  obtained with the multiplicative approach (taking  $M = 0$ ), defined in (7.2). We will consider the two implementations of the boundary conditions for the multiplicative approach: the strong and the weak BC, as presented in Section 6.2. The results are presented in Figure 7 by varying the mesh size  $h$ , considering  $N \in \{16, 32, 64, 128, 256\}$  as presented in Remark 22.

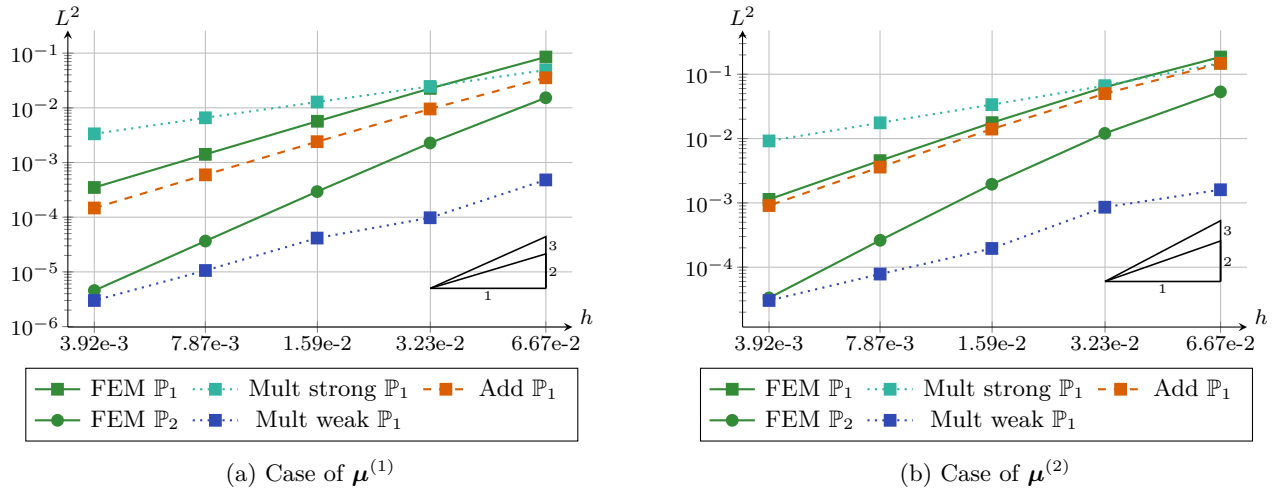


Figure 7: Considering the *1D Elliptic case* and the PINN prior  $u_\theta$ . Left – Considering  $\mu^{(1)}$ .  $L^2$  error on  $h$  obtained with standard FEM  $e_h^{(1)}$  (solid lines) with  $k = 1$  and  $k = 2$ , the additive approach  $e_{h,+}^{(1)}$  (dashed lines) with  $k = 1$  and the multiplicative approach  $e_{h,M}^{(1)}$  (dotted lines) with  $k = 1$ , considering strong and weak BC. Right – Same for  $\mu^{(2)}$ .

In Figure 7, we see that the enriched approaches seem to give better results than standard FEM except for the multiplicative approach with strong imposition of boundary conditions. Moreover, this approach, which imposes  $p_h^\times = 1$  on  $\partial\Omega$ , does not follow the expected convergence order (i.e., for  $k = 1$ , the expected convergence order in the  $L^2$  norm is 2, compared to 1 obtained numerically). The additive approach seems much less effective here than in the previous experiment of Section 7.2, whereas the multiplicative approach with weak BC seems to significantly improve the results obtained with standard FEM. Nevertheless, these two approaches appear to respect the expected convergence order in the  $L^2$  norm, namely 2 for polynomials of order  $k = 1$ . A comparative study of the different methods is given in Section 7.3.2. In addition, we see that the standard FEM with polynomial order  $k = 2$  is clearly less accurate than the multiplicative approach using weak BC applied with polynomial order  $k = 1$ .

### 7.3.2 Comparison of different approaches

We now focus on the second parameter  $\mu^{(2)}$ . We first look at the PINN prediction for this parameter and its derivatives in Figure 8. As in the previous section, we consider the following approaches: standard FEM, the additive approach and the multiplicative approach (with  $M = 0$ ) with strong or weak imposition of boundary conditions. We compare the different methods in Table 8, where we can see the different errors obtained with the considered methods for  $k = 1$  and  $N \in \{16, 32\}$  as well as the gains obtained in comparison with standard FEM. Next, we take a closer look at the solutions obtained with the different approaches in Figure 9; for each method, we compare the solution obtained ( $u_h$  for standard FEM,  $u_h^+$  for the additive approach and  $u_h^\times$  for the multiplicative approach, with strong or weak BC imposition) with the analytical solution  $u$ . For the enriched methods, using the PINN prior, we will also compare the proposed correction; namely, for the additive approach, we will compare  $p_h^+$  with  $u - u_\theta$  and for the multiplicative one  $p_h^\times$  with  $u/u_\theta$  (with  $u_\theta > 0$  in  $\Omega$ ).

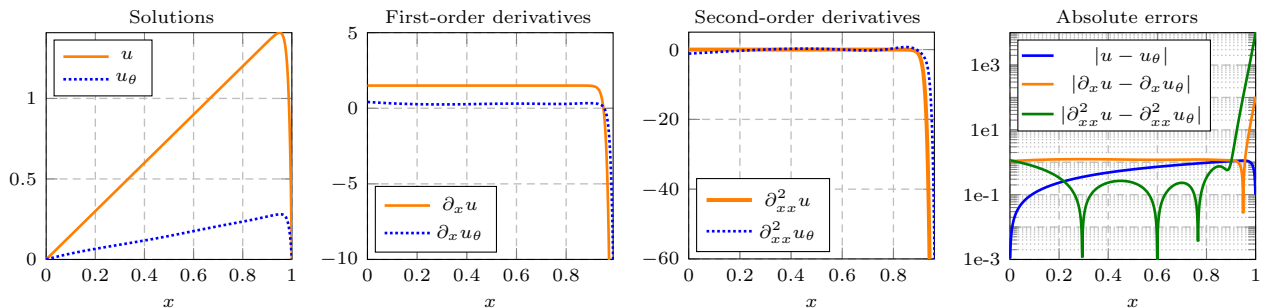
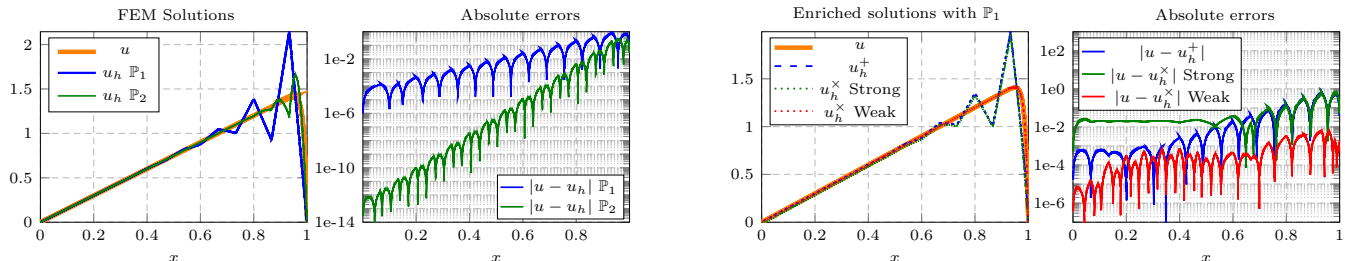


Figure 8: Considering the *1D Elliptic case* with  $\mu^{(2)}$  and the PINN prior  $u_\theta$ , comparison between analytical solution and network prediction. From left to right: solution; first derivative; second derivative; errors.

In Figure 8, we can see that PINN has difficulties to capture the solution and that the prediction it provides is far from the analytical solution. As for its derivatives, they seem to be relatively inaccurate compared to the analytical. Indeed, since the PINN is a smooth function, it has trouble approximating functions with very sharp gradients such as the one of (7.8).

		PINN prior $u_\theta$			
FEM		method	N	error	gain
N	error	Add	16	$1.47 \cdot 10^{-1}$	1.26
			32	$5 \cdot 10^{-2}$	1.26
16	$1.85 \cdot 10^{-1}$	Mult (strong)	16	$1.49 \cdot 10^{-1}$	1.24
	$6.31 \cdot 10^{-2}$		32	$6.65 \cdot 10^{-2}$	0.95
32		Mult (weak)	16	$1.6 \cdot 10^{-3}$	115.82
			32	$8.57 \cdot 10^{-4}$	73.7

Table 8: Considering the *1D Elliptic case* with  $\mu^{(2)}$ ,  $k = 1$  and  $N \in \{16, 32\}$ . Left –  $L^2$  relative error obtained with FEM. Right – Considering the PINN prior  $u_\theta$ ,  $L^2$  relative errors and gains with respect to FEM, obtained with our methods. Our methods : additive approach, multiplicative approach by taking  $M = 0$  (strong and weak BC).



(a) FEM solutions with polynomial order approximation  $k = 1$  and  $k = 2$ , and absolute errors.

(b) Enriched solutions with polynomial order approximation  $k = 1$ , and absolute errors.

Figure 9: Considering the *1D Elliptic case* with  $\mu^{(2)}$ ,  $N = 16$  and the PINN prior  $u_\theta$ . Comparison of the solution obtained with the different methods with the analytical solution. For each enriched method, comparison of the correction term with the analytical one. Different methods : standard FEM, additive approach, multiplicative approach by taking  $M = 0$  (strong and weak BC).

In Figure 9 and Table 8, we present a comparison of the different approaches proposed. In Figure 9a, we first notice the oscillations anticipated in Remark 21 for standard FEM at both polynomial orders  $k = 1$  and  $k = 2$ . This behavior is also seen in the additive enrichment (blue dashed line in Figure 9b), which does not seem to give better results than standard FEM due to the derivatives presented in Figure 8, and for the multiplicative approach with strong boundary conditions (green dotted line in Figure 9b). However, weakly imposing the BC gives the appropriate results (red dotted line in Figure 9b). It seems that for this specific test case, the prediction obtained (close to a multiplicative constant of the solution) favors the multiplicative approach over the additive one.

More specifically, since the accuracy of enriched approaches is completely dependent on the prior, it is not clear in advance which approach (additive or multiplicative) will yield the best results. However, this strong dependence means that we can intuitively choose the enriched approach based on our knowledge of PINNs. For example, as explained above, since PINNs are regular functions, they have difficulty approximating functions with very sharp gradients, as in this test case. Thus, knowing the nature of the test case in question, we can get an idea of the quality of the prior and therefore estimate which approach should be more compatible with it. To go further, if we want a more precise idea of the approach to focus on, we could, after training, calculate the theoretical gain constants using a coarse FEM as a reference solution, for example, which in a parametric approach could be seen as an offline cost. However, there is no theoretical guarantee that the constants will exhibit the same behavior for different parameters, particularly depending on the type of parameter considered.

### 7.3.3 Gains achieved with the additive and the multiplicative approaches

Considering a set  $\mathcal{S}$  of  $n_p = 50$  parameter instances, we will evaluate the gains  $G_{+, \theta}$  and  $G_+$  defined in (7.5) with the PINN prior  $u_\theta$ . We will evaluate  $G_{M, \theta}$  and  $G_M$ , defined in (7.6), in the same way with  $M = 0$  for the multiplicative approach, by considering the two implementations of the boundary conditions; strong and weak BC. The results are presented in Table 9 for fixed  $k = 1$  and  $N \in \{20, 40\}$  fixed.

method	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
Add	20	7.01	93.15	17.97	16.59	1.2	52.18	3.35	6.57
	40	21.19	363.77	68.7	69.83	1.2	46.32	3.29	6.17
Mult (strong)	20	5.29	19.77	9.35	2.51	0.33	13.95	1.58	1.76
	40	10.48	39.89	18.9	4.48	0.16	6.74	0.87	0.84
Mult (weak)	20	69.49	6 247.43	1 479.77	1 094.37	65.32	359.24	186.72	77.12
	40	262.57	24 907.73	4 423.73	3 674.41	54.45	252.57	148.83	48.35

Table 9: Considering the *1D Elliptic case*,  $k = 1$  and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  relative error of our methods with respect to PINN. Right – Gains in  $L^2$  relative error of our methods with respect to FEM. Our methods : additive approach, multiplicative approach with  $M = 0$  (strong and weak BC).

Table 9 confirms the above results. The multiplicative approach with weak BC seems to give the best results on our set  $\mathcal{S}$  of parameter instances. The additive and multiplicative approaches with strong BC imposition do not appear to be very effective on this test case. In particular, even though the additive approach improves the standard FEM error by a factor of 3, we have seen in Section 7.3.2 that the solutions obtained do not correspond to the expected solution, whereas the multiplicative approach with low BCs does. In the following, we will only consider the additive approach, as it seems to be the most efficient one, except in special cases such as the one under consideration in this Section 7.3. Indeed, the following test cases will not contain boundary layers or strong gradients.

## 7.4 1D non-smooth transmission problem

Let  $\Omega = (0, \pi)$  and  $\partial\Omega$  its boundary. In this test case, the physical properties of the medium are discontinuous with respect to space, and we define the interface point  $I = \frac{\pi}{2}$  at the middle of the domain. We consider the following one-dimensional non-smooth transmission problem:

$$\left\{ \begin{array}{ll} -\sigma \partial_{xx} u = f, & \text{in } \Omega \setminus \{I\} \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \\ \lim_{x \rightarrow I^+} \sigma(x, \boldsymbol{\mu}) \partial_x u(x, \boldsymbol{\mu}) = \lim_{x \rightarrow I^-} \sigma(x, \boldsymbol{\mu}) \partial_x u(x, \boldsymbol{\mu}), & \boldsymbol{\mu} \in \mathcal{M}, \\ \lim_{x \rightarrow I^+} u(x, \boldsymbol{\mu}) = \lim_{x \rightarrow I^-} u(x, \boldsymbol{\mu}), & \boldsymbol{\mu} \in \mathcal{M}, \end{array} \right. \quad (7.9)$$

where the diffusion coefficient  $\sigma$  is defined by

$$\sigma(x, \boldsymbol{\mu}) = \begin{cases} \sigma_1, & \text{if } x \in [0, I], \\ \sigma_2, & \text{if } x \in (I, \pi], \end{cases}$$

with  $\boldsymbol{\mu} = (\sigma_1, \sigma_2) \in \mathcal{M} = [2.5, 3.5] \times [0.5, 1.5]$  (we have  $p = 2$  parameters). We prescribe a family of exact solutions (depending on the parameter vector) defined by

$$u(x, \boldsymbol{\mu}) = \begin{cases} \frac{1}{\sigma_1} \sin(2x), & \text{if } x \in (0, I), \\ \frac{1}{\sigma_2} \sin(2x), & \text{if } x \in (I, \pi), \end{cases}$$

and we deduce the associated right-hand side  $f(x, \boldsymbol{\mu}) = 4 \sin(2x)$ .

At first, we will resonate in a non-parametric framework, focusing on the parameter  $\boldsymbol{\mu}^{(1)} = (3, 1)$ . Therefore, in [Section 7.4.2](#), we first present the error estimates obtained on this parameter, considering three different priors (whose construction is detailed in [Section 7.4.1](#)). The first one (denoted by  $u_\theta$ ) is a PINN with a generic MLP architecture. The second one (denoted by  $u_\theta^{\text{data}}$ ) is constructed from data on the solution and its derivatives (obtained from the analytical solution). For the third one (denoted by  $u_\theta^{\text{sing}}$ ), we use a PINN with the architecture proposed in [\[TPMM25\]](#), specifically designed to better capture the behavior of the solution around the interface by enriching the MLP with a singular function. The aim is to show that for this type of problem, using an architecture adapted to the problem leads to better results than a generic one. More specifically, we will show that a naive network is not suited to the enriched approach we propose. However, by enriching the network in a similar way that classical finite element methods are enriched to capture known discontinuities, we obtain results similar to the other test cases considered, for low values of the polynomial degree  $k$ . To do this, we will focus in [Section 7.4.3](#) on the predictions and derivatives of the different non-parametric priors and analyze the results of the enriched approach with the three priors. Finally, in [Section 7.4.4](#), we will return to our parametric framework by constructing the enriched prior  $u_\theta^{\text{sing}}$  in a parametric way (considering  $\boldsymbol{\mu} \in \mathcal{M}$ ) and evaluate the gains obtained with it on a sample of parameters.

**Remark 23.** *As in the previous sections, the characteristic mesh size is  $h = \frac{1}{N-1}$ , where  $N$  is the number of nodes considered. For the finite element methods, we will choose to consider that the interface  $I$  lies on a mesh node (by taking odd values of  $N$ ) to avoid integration issues.*

### 7.4.1 Construction of the different priors

Here, we will discuss the differences in training the three non-parametric networks (used in [Sections 7.4.2](#) and [7.4.3](#)) for the specific parameter  $\boldsymbol{\mu}^{(1)}$ . We will also add the construction of the parametric network  $u_\theta^{\text{sing}}$  (considered in [Section 7.4.4](#)). For PINNs, we will not strongly impose the boundary conditions (i.e. with a levelset function), but rather include them in the loss function.

**Non-parametric physics-informed training with classical architecture.** The first non-parametric prior  $u_\theta$  is a PINN with a generic MLP architecture. Since the problem is not defined on  $I$ , we can simply consider that the integral of the residual loss is divided between the two subdomains (to the left and right of  $I$ ). By applying an independent Monte Carlo method, we obtain the same approximate residual loss  $J_r$  defined in [\(5.6\)](#) (considering  $N_{\text{col}} = 3000$  collocation points uniformly chosen on  $\Omega \setminus \{I\}$ ). Since the boundary conditions are not strongly imposed in this network, we also include the boundary loss  $J_b$  defined in [\(5.7\)](#). It should be noted that, by definition of the network in question, the continuity condition is automatically validated, but the flow condition cannot be satisfied. In practice, the considered prior will aim to be accurate on the residual of transmission problem, while the solution itself will be very poorly represented. The hyperparameters used are given in [Table 10](#).

Network - MLP		Training				Loss weights			
		$lr$	1e-3	$n_{\text{epochs}}$	2 000	$u_\theta$	$u_\theta^{\text{data}}$	$u_\theta^{\text{sing}}$	
$layers$	20, 20, 20	$decay$	0.99			$\omega_r$	1	$\omega_r$	1
$\sigma$	tanh	$N_{\text{col}/\text{data}}$	3 000			$\omega_b$	20	$\omega_b$	20
								$\omega_{\text{int}}$	1

Table 10: Network, training parameters ([Remark 15](#)) and loss weights for non-parametric priors  $u_\theta$ ,  $u_\theta^{\text{data}}$  and  $u_\theta^{\text{sing}}$  in the *1D transmission problem*. Considering  $N_{\text{col}}$  collocation points for PINNs and  $N_{\text{data}}$  data for the data prior. Weights not specified in the table are defined to be zero.

**Non-parametric data-driven training with classical architecture.** For the second non-parametric prior  $u_\theta^{\text{data}}$ , a network is trained only on the data (constructed from the analytical solution), considering the same architecture as above. In [Section 7.2](#), we have already seen that a network based on data is not sufficient to correctly capture the derivatives of the solution. Hence, based on [\[TPMM25\]](#), we define the following modified data loss function, including information on the derivatives:

$$J_{\text{data}}(\theta) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} |u_\theta^{\text{data}}(\mathbf{x}_{\text{data}}^{(i)}, \boldsymbol{\mu}^{(1)}) - u(\mathbf{x}_{\text{data}}^{(i)}, \boldsymbol{\mu}^{(1)})|^2 + |\partial_x u_\theta^{\text{data}}(\mathbf{x}_{\text{data}}^{(i)}, \boldsymbol{\mu}^{(1)}) - \partial_x u(\mathbf{x}_{\text{data}}^{(i)}, \boldsymbol{\mu}^{(1)})|^2,$$

with  $N_{\text{data}} = 3000$  points uniformly chosen on  $\Omega \setminus \{I\}$ . The hyperparameters used are also given in [Table 10](#).

**Non-parametric physics-informed training with enriched architecture.** The third non-parametric prior  $u_\theta^{\text{sing}}$  is a PINN with an enriched architecture specifically designed to capture the behavior of the solution around the interface. This architecture was proposed in [\[TPMM25\]](#) under the name ReCoNNs (Regularity-Conforming Neural Networks). More precisely, we define the non-parametric network as follows:

$$u_\theta^{\text{sing}}(x, \boldsymbol{\mu}) = w_\theta^0(x, \boldsymbol{\mu}) + w_\theta^1(x, \boldsymbol{\mu}) \frac{|x - I|}{2}, \quad (7.10)$$

with  $w_\theta = (w_\theta^0, w_\theta^1)$  a classic MLP and  $\boldsymbol{\mu} = \boldsymbol{\mu}^{(1)}$ . This architecture allows us to capture the discontinuity of the first derivative of the solution at the interface  $I$ . The network is trained in a physics-informed way, minimizing the same residual loss function  $J_r$  as for the first prior and the same boundary loss function  $J_b$ . The continuity condition is automatically satisfied by the construction of the network. For the flow condition, we then add the interface loss  $J_{\text{int}}$  defined by:

$$J_{\text{int}}(\theta) = \left| \lim_{x \rightarrow I^+} \sigma(x, \boldsymbol{\mu}^{(1)}) \partial_x u_\theta^{\text{sing}}(x, \boldsymbol{\mu}^{(1)}) - \lim_{x \rightarrow I^-} \sigma(x, \boldsymbol{\mu}^{(1)}) \partial_x u_\theta^{\text{sing}}(x, \boldsymbol{\mu}^{(1)}) \right|^2,$$

where the limits of network derivatives at the interface are calculated analytically, as detailed in [\[TPMM25\]](#). The minimization problem then becomes:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left( \omega_r J_r(\theta) + \omega_b J_b(\theta) + \omega_{\text{int}} J_{\text{int}}(\theta) \right),$$

with  $\omega_r$ ,  $\omega_b$  and  $\omega_{\text{int}}$  some weights to balance the different terms of the loss function. The hyperparameters used are still given in [Table 10](#).

**Parametric physics-informed training with enriched architecture.** For [Section 7.4.4](#), we will consider the same enriched architecture as above, but in a parametric way. Thus, we define the parametric network as defined in [\(7.10\)](#), considering  $\boldsymbol{\mu} \in \mathcal{M}$  and the modified interface loss function:

$$J_{\text{int}}(\theta) = \frac{1}{N_{\text{col}}} \sum_{i=1}^{N_{\text{col}}} \left| \lim_{x \rightarrow I^+} \sigma(x, \boldsymbol{\mu}_{\text{col}}^{(i)}) \partial_x u_\theta^{\text{sing}}(x, \boldsymbol{\mu}_{\text{col}}^{(i)}) - \lim_{x \rightarrow I^-} \sigma(x, \boldsymbol{\mu}_{\text{col}}^{(i)}) \partial_x u_\theta^{\text{sing}}(x, \boldsymbol{\mu}_{\text{col}}^{(i)}) \right|^2.$$

The hyperparameters used for the network and training are the same as for the non-parametric PINN prior with enriched architecture (defined in [Table 10](#)), except for the number of epochs set to  $n_{\text{epochs}} = 10000$  and the initial learning rate set to  $lr = 1.8e-2$ .

#### 7.4.2 Error estimates — with the three non-parametric priors

We perform the same test as in the previous sections, considering only the additive approach with the three non-parametric priors presented in [Section 7.4.1](#). The results in  $L^2$  norm are presented for  $\boldsymbol{\mu}^{(1)}$  in [Figure 10](#) for fixed  $k \in \{1, 2, 3\}$  and by varying the number of nodes  $N \in \{21, 41, 61, 81, 101\}$  ([Remark 23](#)).

The results of the enriched approach are obtained by calculating the source term weakly, meaning only using the first derivatives of the prior. Also, like in the other test cases, the derivatives are calculated analytically using `PyTorch`. However, in the case of the particular architecture used in  $u_\theta^{\text{sing}}$ , the derivatives produced are discontinuous, which must be taken into account during integration. More precisely, using an odd number of nodes  $N$  (as stated in [Remark 23](#)), two cells of the mesh intersect at the interface  $I$ . Using a Galerkin discontinuous space to represent the first derivative of the prior, we will define the value at  $I$  of the left (resp. right) cell by its value at  $I - \epsilon$  (resp.  $I + \epsilon$ ) with  $\epsilon = 10^{-6}$ . This way, we ensure that the correct value of the derivative is used on each side of the interface during integration.

The results presented in [Figure 10](#) show that the additive approach with the PINN prior and the enriched architecture  $u_\theta^{\text{sing}}$  provides the best results. In particular, we observe that for polynomial order  $k = 1$ , the errors obtained with PINNs are much lower than those obtained with the data network, and that the gain compared to standard FEM is significant. Furthermore, we also observe that the convergence rates are similar for all three priors and match those of the classical FEM. This confirms that using a network architecture adapted to the problem yields better results in our enriched finite element method. On the other hand, we can see that the data-driven network seems to deteriorate the results of the standard approach, particularly for  $k \neq 1$ , which is a behavior we did not observe elsewhere. As this test case does

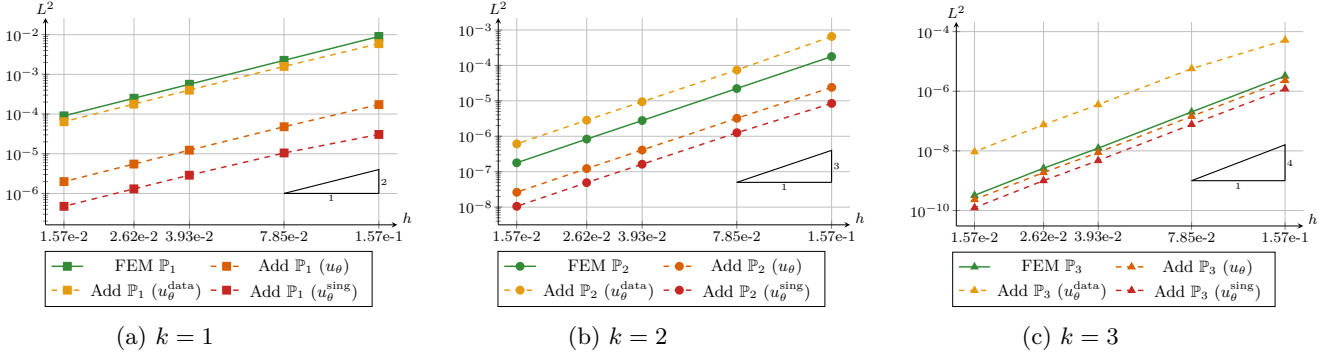


Figure 10: Considering the *1D transmission problem* with  $\mu^{(1)}$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid line) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines), with  $k = 1$ , by considering the PINN prior with classic architecture  $u_\theta$ , the data-driven prior  $u_\theta^{\text{data}}$  and the PINN with enriched architecture  $u_\theta^{\text{sing}}$  (all in a non-parametric way). Middle – Same with  $k = 2$ . Right – Same with  $k = 3$ .

not enter into the theoretical framework of Section 2, we cannot assert the causes of this; further study is warranted. However, from what we have seen so far, we have observed that the more we increase the polynomial order, the more the order of the derivatives that control the error increases. If the reasoning is similar here, a network trained solely by data could produce such a result. To better understand these results, we now analyze the predictions and derivatives of the three priors in Section 7.4.3.

### 7.4.3 Derivatives — with the three non-parametric priors

To better explain the results of Section 7.4.2 for selected parameter  $\mu^{(1)}$ , we compare the solution, the first- and second-order derivatives between the exact solution and the prediction of the three priors constructed in Section 7.4.1. For analytical second derivatives, they are calculated independently on the left and right intervals. The results are presented in Figure 11.

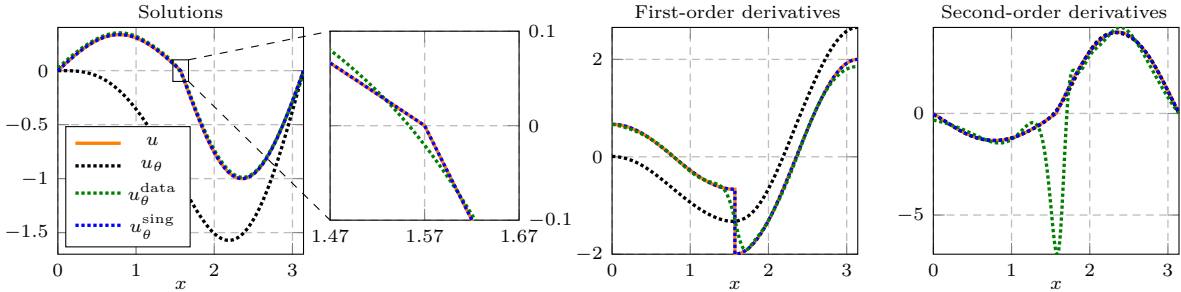


Figure 11: Considering the *1D transmission problem* with  $\mu^{(1)}$  and the three different non-parametric priors, the PINN prior with classic architecture  $u_\theta$  (black), the data-driven prior  $u_\theta^{\text{data}}$  (green) and the PINN with enriched architecture  $u_\theta^{\text{sing}}$  (blue). Comparison between analytical solution and network prediction. From left to right: solution; zoom on solution at interface; first derivative; second derivative.

In Figure 11, we see that the prior that gives the best approximation of the solution is the PINN with the singular architecture, followed by the data-driven network, which seems to smooth the solution at the interface somewhat. The standard PINN, on the other hand, has great difficulty capturing the solution. In terms of first derivatives, we see that the enriched PINN is also very successful, as is the case for the solution. The data-driven network seems to give a smooth approximation of the first derivative, while the standard PINN still seems quite far off. However, the data-driven network has the most difficulty approximating second derivatives, followed by the standard PINN, while the singular network approximates them very well. In view of the results of Section 7.4.2, it would therefore seem that it is the higher-order derivatives that have the greatest impact on the results, even if the prior itself is not optimal. This conclusion also holds true in other test cases where our theoretical framework is applicable.

#### 7.4.4 Gains achieved with the additive approach – with the parametric enriched PINN prior

Considering a set  $\mathcal{S}$  of  $n_p = 50$  parameter instances and the parametric enriched PINN prior  $u_\theta^{\text{sing}}$  defined in Section 7.4.1, we now evaluate the gains  $G_{+,\theta}$  and  $G_+$  defined in (7.5). The results are presented in Table 11 for  $k \in \{1, 2, 3\}$  and  $N \in \{21, 41\}$ .

k	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	21	37.43	185.85	101.87	40.6	32.41	270.99	139.17	41.77
	41	141.02	629.45	383.64	133.39	33.64	231.89	132.2	35.7
2	21	211.26	3 555.64	1 033.67	643.81	16.72	34.59	25.08	4.7
	41	1 057.29	25 483.75	5 719.05	4 470.3	13.52	19.16	16.56	1.54
3	21	770.08	28 142.92	4 569.45	4 825.81	1.48	2.68	1.79	0.23
	41	11 038.01	354 298.53	58 440.14	60 029.7	1.2	2.02	1.45	0.22

Table 11: Considering the 1D transmission problem,  $k \in \{1, 2, 3\}$  and the parametric PINN prior with enriched architecture  $u_\theta^{\text{sing}}$ . Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

The results in Table 11 confirm the results obtained for  $\mu^{(1)}$  in Section 7.4.2. It would appear that the additive approach with the enriched PINN prior  $u_\theta^{\text{sing}}$  provides significant gains, particularly for  $k = 1$ . For  $k = 3$ , the minimum obtained is greater than 1, so we see that, unlike when using the data-driven prior, the results of the standard method do not appear to be degraded. On the other hand, as the degree of the polynomial increases, the average gains appear to decrease, and quite considerably so.

## 7.5 2D Poisson problem in a square domain

We now consider the problem of Section 7.2 but in two dimensions ( $d = 2$ ), with,

$$\begin{cases} -\Delta u = f, & \text{in } \Omega \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \end{cases} \quad (7.11)$$

with  $\Delta$  the Laplace operator on the domain  $\Omega = (-0.5\pi, 0.5\pi)^2$  with boundary  $\partial\Omega$ , and  $\mathcal{M} \subset \mathbb{R}^p$  the parameter space (with  $p$  the number of parameters). We define the right-hand side  $f$  such that the solution is given by

$$u(\mathbf{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(x - \mu_1)^2 + (y - \mu_2)^2}{2}\right) \sin(\kappa x) \sin(\kappa y), \quad (7.12)$$

with  $\mathbf{x} = (x, y) \in \Omega$  and some parameters  $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{M} = [-0.5, 0.5]^p$ , hence with  $p = 2$  parameters. With an abuse of language as well, we refer to the quantity  $\kappa$  in (7.12) as the frequency of the solution, in the sense that it characterizes the number of oscillations in the solution.

We start with a “low frequency” case in Section 7.5.1, taking  $\kappa = 2$  and considering a PINN where we impose the Dirichlet boundary conditions as presented in Section 5.2.1., i.e. using a level-set function. To further improve the prior quality, we introduce an augmented loss function in Section 7.5.2 by using the Sobolev training presented in Section 5.2.2. Afterwards, we test another loss in Section 7.5.3 that includes the Dirichlet condition, or in other words, that does not use a level-set function. Finally, we consider a “higher frequency” case in Section 7.5.4, with  $\kappa = 8$ .

**Remark 24.** In the following, the characteristic mesh size  $h = \frac{\pi\sqrt{2}}{N-1}$  is defined as a function of  $N$ , considering a cartesian mesh of  $N^2$  nodes for our squared 2D domain of length  $\pi$ .

### 7.5.1 Low-frequency case

We consider a “low-frequency” problem, taking  $\kappa = 2$ . In this section, we consider the additive approach, as presented in Section 2, by considering the PINN prior  $u_\theta$ . We start by testing the error estimates in (in  $L^2$  norm and  $H^1$  semi-norm)

with polynomial order  $k \in \{1, 2, 3\}$ , then we compare the different approaches. We evaluate the gains obtained on a sample of parameters. Then, we compare the numerical costs of the different methods. Finally we discuss the importance of integrating analytical functions, as presented in [Section 6.1](#).

Since the problem under consideration is parametric we deploy a parametric PINN, which depends on both the space variable  $\mathbf{x} = (x, y) \in \Omega$  and the parameters  $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{M}$ . Moreover, we strongly impose the Dirichlet boundary conditions, as presented in [Section 5.1](#). To do this, we use the prior  $u_\theta$  defined in (5.2.1), where we choose the level-set function  $\varphi$  defined by

$$\varphi(\mathbf{x}) = (x + 0.5\pi)(x - 0.5\pi)(y + 0.5\pi)(y - 0.5\pi).$$

Thus, we will only consider the residual loss  $J_r$  approached by a Monte-Carlo method as defined in (5.6) with  $N_{\text{col}} = 6000$  collocation points uniformly chosen on  $\Omega \times \mathcal{M}$ . The parametric network is defined as an MLP with the hyperparameters defined in [Table 12](#); we use the Adam optimizer and then switch to the LBFGS optimizer after the  $n_{\text{switch}}$ -th epoch.

Network - MLP		Training - with LBFGS				Loss weights			
<i>layers</i>	40, 60, 60, 60, 40	<i>lr</i>	1.7e-2	<i>n_epochs</i>	5 000	$\omega_r$	1	$\omega_{\text{data}}$	0
$\sigma$	sine	<i>decay</i>	0.99	<i>n_switch</i>	1 000	$\omega_b$	0	$\omega_{\text{sob}}$	0
		$N_{\text{col}}$	6 000						

Table 12: Network, training parameters ([Remark 15](#)) and loss weights for  $u_\theta$  in the 2D low-frequency case.

**Error estimates.** We start by testing the error estimates of [Theorem 6](#) for the following two sets of parameters, randomly selected in  $\mathcal{M}$ :

$$\boldsymbol{\mu}^{(1)} = (0.05, 0.22) \quad \text{and} \quad \boldsymbol{\mu}^{(2)} = (0.1, 0.04),$$

by considering the PINN prior  $u_\theta$ . We perform the same tests as in the previous sections, but this time considering only the additive approach. The results are presented in [Figures 12](#) and [13](#) (respectively in norm  $L^2$  and semi-norm  $H^1$ ) for fixed  $k \in \{1, 2, 3\}$  with  $h$  depending on  $N \in \{16, 32, 64, 128, 256\}$ , as presented in [Remark 24](#).

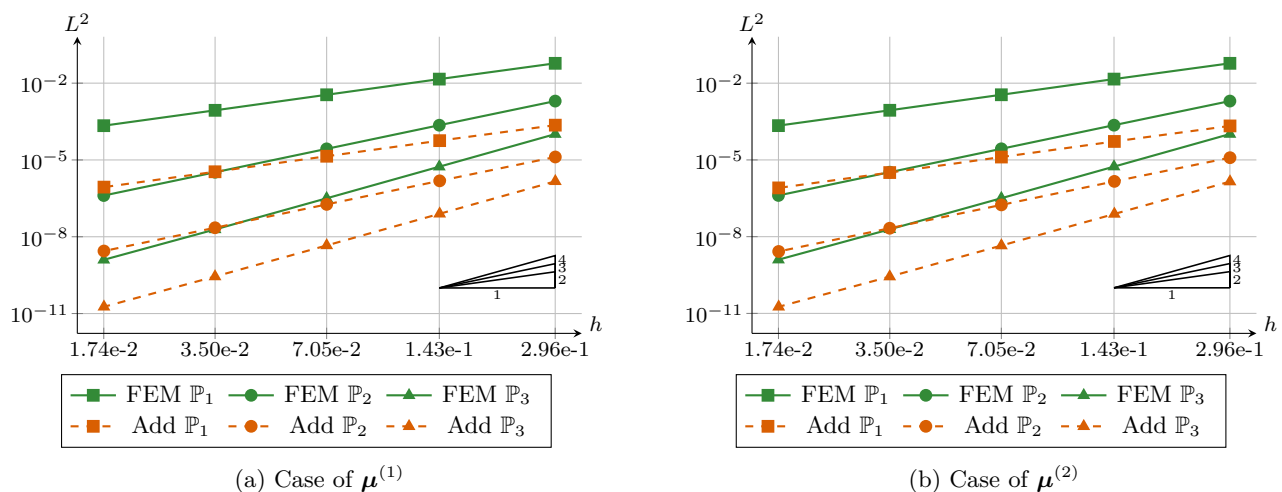


Figure 12: Considering the 2D low-frequency case and the PINN prior  $u_\theta$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid lines) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines) for  $\boldsymbol{\mu}^{(1)}$ , with  $k \in \{1, 2, 3\}$ . Right – Same for  $\boldsymbol{\mu}^{(2)}$ .

In [Figures 12](#) and [13](#), we observe the expected behavior. Indeed, the error decreases with the correct order of accuracy as the mesh size  $h$  decreases (i.e., with a slope of  $k + 1$  in the  $L^2$  norm and  $k$  in the  $H^1$  semi-norm). This observation is valid for both the classical and enriched FEM. Moreover, we observe that the error constant of the additive approach is significantly lower than that of the classical FEM. In the following paragraph, we will compare these different approaches in more detail. As noted in the 1D case, we can see that the additive enriched approach for  $k = 1$  (resp.  $k = 2$ ) seems to give relative errors close to the original FEM for  $k = 2$  (resp.  $k = 3$ ), although the rate of convergence is different.

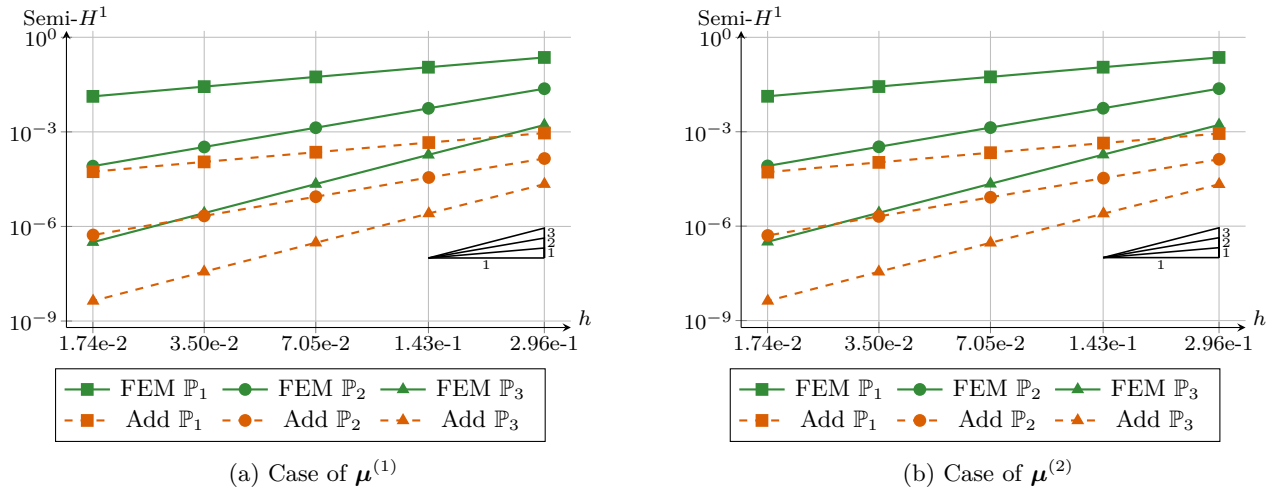


Figure 13: Considering the  $2D$  low-frequency case and the PINN prior  $u_\theta$ . Left – Semi- $H^1$  relative error on  $h$ , obtained with the standard FEM (solid lines) and the additive approach (dashed lines) for  $\mu^{(1)}$ , with  $k \in \{1, 2, 3\}$ . Right – Same for  $\mu^{(2)}$ .

**Comparison of different approaches.** We now focus on the first parameter  $\mu^{(1)}$ . We compare the standard FEM with the additive approach, first in the Table 13 where we can see the different errors obtained with the different methods for  $k = 1$  fixed and  $N \in \{16, 32\}$  as well as the gains obtained in comparison with standard FEM. Next, we take a closer look at the solution obtained with the different approaches in Figure 14; for each method, we compare the solution obtained ( $u_h$  for standard FEM and  $u_h^+$  for the additive approach) with the analytical solution  $u$ . For the enriched method, using the PINN prior  $u_\theta$ , we will also compare the proposed correction; namely, for the additive approach, we will compare  $p_h^+$  with  $u - u_\theta$ .

FEM		PINN prior $u_\theta$			
N	error	method	N	error	gain
16	$5.95 \cdot 10^{-2}$	Add	16	$2.29 \cdot 10^{-4}$	260.38
32	$1.44 \cdot 10^{-2}$		32	$5.67 \cdot 10^{-5}$	253.66

Table 13: Considering the  $2D$  low-frequency case with  $\mu^{(1)}$ ,  $k = 1$  and  $N \in \{16, 32\}$ . Left –  $L^2$  relative error obtained with FEM. Right – Considering the PINN prior  $u_\theta$ ,  $L^2$  relative errors and gains with respect to FEM, obtained with the additive approach.

In Table 13, we observe that the additive approach significantly improves the error of the standard FEM, with gains of around 260 for  $N = 16$  and  $k = 1$ , which is equivalent to refining the mesh by a factor of 16 for  $\mathbb{P}_1$  elements. Indeed, in this case, our enriched approach gives much better results than standard FEM. In Figure 14, we observe that the solution obtained with the additive approach is very close to the analytical solution, with a correction term that is also very close to the analytical one. This shows the effectiveness of the additive approach in this case.

**Gains achieved with the additive approach.** Considering a set  $\mathcal{S}$  of  $n_p = 50$  parameter instances, we now evaluate the gains  $G_{+, \theta}$  and  $G_+$  defined in (7.5). The results are presented in Table 14 for  $k \in \{1, 2, 3\}$  and  $N \in \{20, 40\}$ .

In Table 14, we observe (left subtable) that our method significantly improves the error of the PINN, especially for large values of  $k$ , where the enrichment is performed in a richer approximation space. Moreover, we also observe (right subtable) significant gains with respect to classical FEM. For instance, as expected from the results of the previous paragraph, the mean gains for  $k = 1$  are around 270, which corresponds to refining the mesh approximately 16 times for  $\mathbb{P}_1$  elements. This means that our  $\mathbb{P}_1$  enhanced bases capture the solution as accurately as classical  $\mathbb{P}_1$  bases with a mesh sixteen times finer. For  $k = 2$  and  $k = 3$ , the mean gains are around 134 and 61, respectively, which corresponds to refining the mesh approximately 5 times for  $\mathbb{P}_2$  elements and 2.8 times for  $\mathbb{P}_3$  elements. A natural follow-up question consists in assessing the impact of the PINN quality on our results. This will be the subject of the Section 7.5.2.

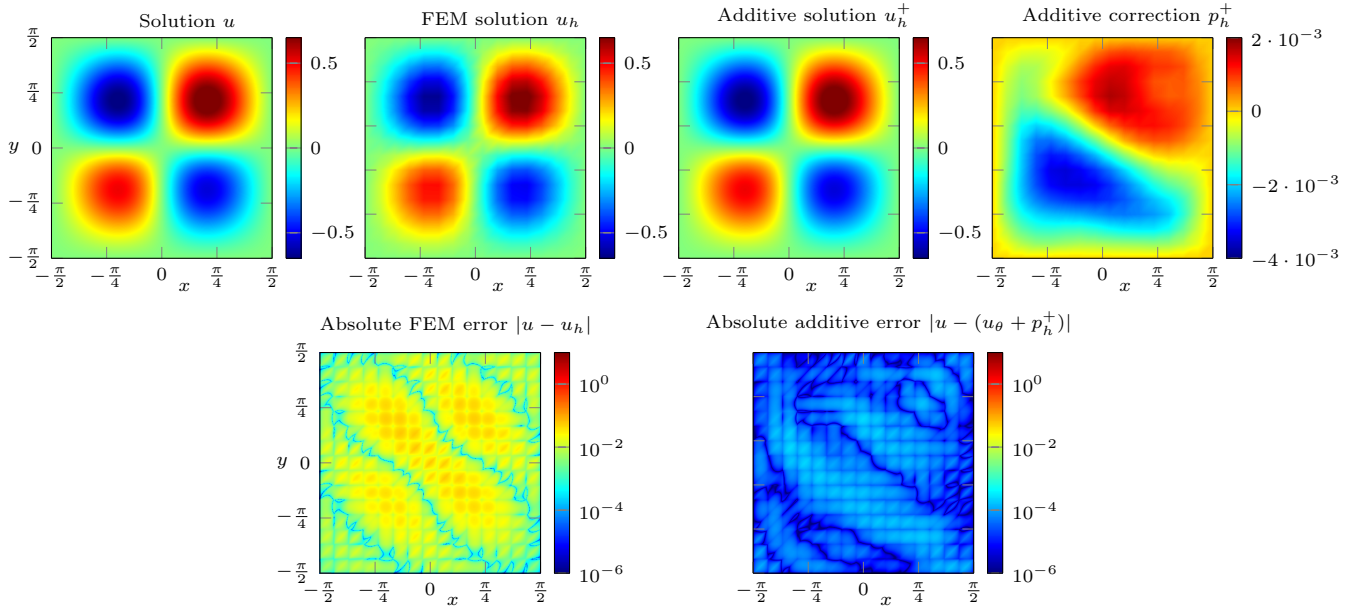


Figure 14: Considering the  $2D$  low-frequency case with  $\mu^{(1)}$ ,  $k = 1$ ,  $N = 16$  and the PINN prior  $u_\theta$ . Comparison of the solution obtained with the standard FEM and the additive approach with the analytical solution. For the additive method, comparison of the correction term with the analytical one.

k	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	20	15.69	48.33	33.62	5.56	134.32	377.36	269.39	43.66
	40	61.4	195.53	135.25	23.18	131.2	362.12	262.13	41.68
2	20	244.05	993.21	653.07	153.16	67.02	164.65	134.85	21.29
	40	2055.16	8340.94	5501.88	1286.49	66.45	159.48	131.86	20.34
3	20	2780.52	11705.79	7542.95	1766.24	39.52	72.65	61.55	7.02
	40	50649.24	211156.91	136714.04	31882.69	39.83	72.62	61.67	6.87

Table 14: Considering the  $2D$  low-frequency case,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

**Costs of the different methods.** To more accurately assess the benefits of using the enriched methods, we look in this section at the costs of the different methods proposed, considering the parameter  $\mu^{(1)}$ . Thus, we will consider that the cost of using the PINN prior  $u_\theta$ , corresponds to the total number of weights of the network considered. In this case, it is given as an MLP with the hyperparameters defined in Table 12, for a total of  $N_{\text{weights}} = 12461$  weights.

For the different finite element methods, we endeavor to determine, for a fixed polynomial degree  $k$ , the characteristic mesh size  $h$  (depending on  $N$  as described in Remark 24) required to reach a given fixed error  $e$ . In Table 15, we study, for  $k \in \{1, 2, 3\}$ , considering standard FEM and the additive approach, the  $N$  required to achieve the same error  $e$ . More precisely, the characteristic mesh size required by standard FEM so that  $e_h^{(1)} \approx e$  and the one required by the additive approach so that  $e_{h,+}^{(1)} \approx e$ . Depending on the polynomial degree  $k$ , we can also determine the number of degrees of freedom  $N_{\text{dofs}}$  associated with each case. These results are obtained by interpolating the convergence curves of Figure 12 for the different methods for a given  $e$ .

In Table 15, we see that the additive approach proposed in Section 2 requires a much coarser mesh than standard FEM to achieve the same  $e$  error. This is due to the error estimations of Theorem 6 which show that the error of the enhanced FEM is significantly lower than that of the classical FEM (depending on the quality of the prior). This is also reflected in the number of degrees of freedom required to achieve the same error  $e$ .

However, the enriched approaches proposed require using the prior PINN  $u_\theta$ , which also includes its inference cost.

<b>k</b>	<b>e</b>	<b>N</b>		<b>N<sub>dofs</sub></b>	
		<b>FEM</b>	<b>Add</b>	<b>FEM</b>	<b>Add</b>
1	$1 \cdot 10^{-3}$	119	8	14 161	64
	$1 \cdot 10^{-4}$	379	24	143 641	576
2	$1 \cdot 10^{-4}$	42	8	6 889	225
	$1 \cdot 10^{-5}$	89	17	31 329	1 089
3	$1 \cdot 10^{-5}$	28	10	6 724	784
	$1 \cdot 10^{-6}$	48	18	20 164	2 704

Table 15: Considering the *2D low-frequency case* with  $\boldsymbol{\mu}^{(1)}$ ,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta$ . Left – Characteristic  $N$  (associated to the characteristic mesh size  $h$ ) required to reach a fixed error  $e$  for standard FEM and the additive approach. Right – Number of degrees of freedom  $N_{\text{dofs}}$  associated with each case.

For this reason, it is also interesting to study these same costs on a set of parameters, say of size  $n_p = 100$ . Since we are in the context of parametric PINN, we can estimate that the computational cost of solving (7.11) on this sample of  $n_p$  parameters corresponds, for the additive approach, to  $n_p$  times its number of dofs plus the cost of using PINN (i.e. its total number of weights), thus  $n_p \times N_{\text{dofs}} + N_{\text{weights}}$  (with  $N_{\text{dofs}}$  the number of dofs associated to the additive approach). For standard FEM, this cost is equivalent to  $n_p$  times its estimated number of dofs  $n_p \times N_{\text{dofs}}$  (with  $N_{\text{dofs}}$  the number of dofs associated with standard FEM). We will then compare these costs for a set of  $n_p = 100$  parameters, considering the same error  $e$  to be achieved for both methods. The results are presented in Table 16. Note that these results (right subtable) are, in fact, only an estimate of the real cost of solving  $n_p$  problems. In practice, the number of degrees of freedom  $N_{\text{dofs}}$  associated with each method depends on the parameter itself. The error to be achieved  $e$  will require more or less fine meshes for each parameter.

<b>k</b>	<b>e</b>	$n_p = 1$		$n_p = 100$	
		<b>FEM</b>	<b>Add</b>	<b>FEM</b>	<b>Add</b>
1	$1 \cdot 10^{-3}$	14 161	12 525	1 416 100	18 861
	$1 \cdot 10^{-4}$	143 641	13 037	14 364 100	70 061
2	$1 \cdot 10^{-4}$	6 889	12 686	688 900	34 961
	$1 \cdot 10^{-5}$	31 329	13 550	3 132 900	121 361
3	$1 \cdot 10^{-5}$	6 724	13 245	672 400	90 861
	$1 \cdot 10^{-6}$	20 164	15 165	2 016 400	282 861

Table 16: Considering the *2D low-frequency case*,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta$ . Left – Total costs of standard FEM and the additive approach to reach an error  $e$  for a set of  $n_p = 1$  parameter. Right – Same for a set of  $n_p = 100$  parameters.

In Table 16, for  $n_p = 1$  (left subtable), we can see that the cost of the additive method is generally lower than that of standard FEM, even though they are of the same range. However, it is important to note that these are not entirely comparable: in fact, a large part of the cost of the additive method lies in PINN prediction, which will be more or less well estimated depending on a number of hyper-parameters (number of epochs, learning rate, etc.). If we then take  $n_p = 100$  (right subtable), we can see that the cost of standard FEM becomes radically higher than with the additive approach. This is why the improved approach is particularly interesting for solving the (7.11) problem on a set of parameters.

**Integration of analytical functions.** This section aims to discuss one of the important points, presented in Section 6.1, that enables PINN to be used effectively and can make our enriched methods more or less effective. Indeed, according to (2.3), we have to integrate  $f + \Delta u_\theta$  multiplied by the test function. To perform this integration, we first interpolate this term on a polynomial space and then integrate it exactly. The degree of this polynomial approximation is an important parameter to make our technique effective.

The goal here is simply to show that for enriched approaches to be effective, particularly the additive method, this

polynomial approximation must be of a sufficiently high degree. Consider the parameter  $\mu^{(1)}$ , a polynomial degree  $k = 3$  and a number of nodes  $N^2$  with  $N = 128$ . In Figure 15, we display the  $L^2$  relative error of the additive approach with respect to the degree of polynomial approximation of  $f + \Delta u_\theta$ .

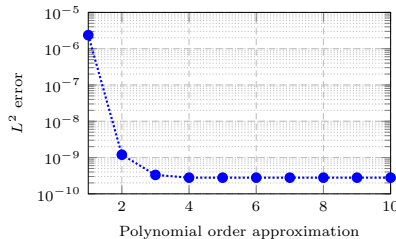


Figure 15: Considering the 2D low-frequency case with  $\mu^{(1)}$ ,  $k = 3$ ,  $N = 128$  and the PINN prior  $u_\theta$ . Considering the additive approach,  $L^2$  error  $e_{h,+}^{(1)}$  with respect to the degree of polynomial approximation of  $f + \Delta u_\theta$ .

In Figure 15, we observe that the error decreases as the degree of polynomial approximation increases. This shows the importance of properly interpolating analytical functions in the context of enriched methods.

### 7.5.2 Low-frequency case — Sobolev training

This section focuses on the same problem as in Section 7.5.1. The aim here is to show that the network quality has a non-negligible impact on the results obtained with our method and that if the network is better, our results will be, too. To this end, we defined a new prior  $u_\theta^{\text{sob}}$  by using the Sobolev training presented in Section 5.2.2, where the derivatives of the solution should be better approximated than by a standard training and compared it with the PINN prior  $u_\theta$  defined in Section 7.5.1. We start by testing the error estimation with  $k \in \{1, 2, 3\}$  polynomial order and evaluate the gains obtained on the same sample of parameters as in Section 7.5.1.

We deploy here a parametric PINN, denoted by  $u_\theta^{\text{sob}}$ , where we consider the residual loss  $J_r$  and the Sobolev loss  $J_{\text{sob}}$ , respectively defined in (5.3) and (5.9), whose integrals are both approximated by a Monte-Carlo method with  $N_{\text{col}} = 6000$  collocation points. We then solve the minimisation problem defined in (5.8) considering that the Dirichlet boundary conditions are strongly imposed as in Section 7.5.1. The hyperparameters are defined in Table 17; we use the Adam optimizer.

**Remark 25.** Adding the Sobolev loss can make training more difficult, so we only consider 3000 epochs but a batch size of 2000. This means that for each epoch, the weights will be updated 3 times (because  $N_{\text{col}} = 6000$ ).

Network - MLP		Training - with LBFGS				Loss weights			
<i>layers</i>	40, 60, 60, 60, 40	<i>lr</i>	1.7e-2	<i>n<sub>epochs</sub></i>	3 000	<i>ω<sub>r</sub></i>	1	<i>ω<sub>data</sub></i>	0
<i>σ</i>	sine	<i>decay</i>	0.99	batch size	2 000	<i>ω<sub>b</sub></i>	0	<i>ω<sub>sob</sub></i>	0.1
		<i>N<sub>col</sub></i>	6 000						

Table 17: Network, training parameters (Remark 15) and loss weights for  $u_\theta^{\text{sob}}$  in the 2D low-frequency case.

**Error estimates.** For simplicity, we consider the first parameter  $\mu^{(1)} = (0.05, 0.22)$  presented in Section 7.5.1. We perform the same test as in the previous section, considering the two priors,  $u_\theta$  and  $u_\theta^{\text{sob}}$ , in the enriched approach. The results are presented in Figure 16 for fixed  $k \in \{1, 2, 3\}$  (in  $L^2$  norm).

We observe that Sobolev training improves the results obtained with the  $L^2$  training, for  $k \in \{1, 2, 3\}$ . This shows the impact of the quality of the network prediction on our method. To further investigate this, we evaluate the gains obtained with the Sobolev training on the same sample of parameters as in Section 7.5.1.

**Gains achieved with the additive approach.** Considering the same set  $\mathcal{S}$  of  $n_p = 50$  parameter instances as in Section 7.5.1, we now evaluate the gains  $G_{+,\theta}$  and  $G_+$  defined in (7.5) considering the PINN prior  $u_\theta^{\text{sob}}$  using Sobolev training. The results are presented in Table 18 for  $k \in \{1, 2, 3\}$  and  $N \in \{20, 40\}$ .

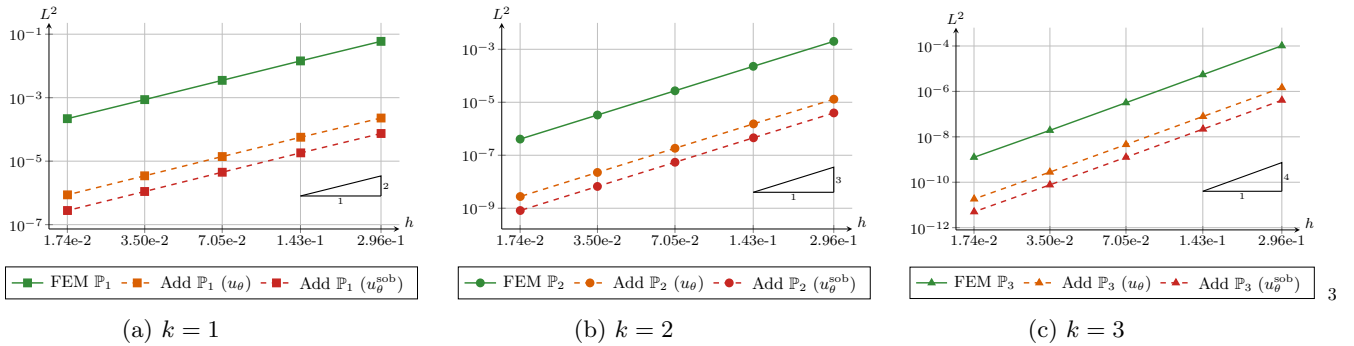


Figure 16: Considering the  $2D$  low-frequency case with  $\mu^{(1)}$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid line) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines), with  $k = 1$ , by considering the PINN prior with standard training  $u_\theta$  and Sobolev training  $u_\theta^{\text{sob}}$ . Middle – Same with  $k = 2$ . Right – Same with  $k = 3$ .

k	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	20	16.14	80.38	41.1	15.04	188.69	1 085.32	768.22	200.13
	40	63.93	324.11	166.45	61.46	187.82	1 051.55	751.01	190.72
2	20	263.88	1 751.13	896.03	394.57	147.08	536.32	414.91	78.1
	40	2 235.97	14 849.11	7 600.79	3 349.18	146.35	528.57	408.6	76.94
3	20	3 040.5	22 972.38	10 662.78	5 092.35	91.84	244.09	192.96	32.23
	40	55 299.93	416 438.72	194 157.02	92 537.38	93.16	246.78	194.31	32.51

Table 18: Considering the  $2D$  low-frequency case,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta^{\text{sob}}$  (Sobolev training). Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

The gains reported in Table 18 show that, compared to  $L^2$  training, Sobolev training increases the mean gains by a factor of about 3. This corresponds to almost half an additional mesh refinement for the  $\mathbb{P}_1$  elements. We also note that this Sobolev training is particularly interesting for higher polynomial degrees, with standard  $L^2$  training having lower gains than for  $k = 1$ .

### 7.5.3 Low-frequency case — Boundary loss training

In this section, we focus on the same problem as in Section 7.5.1 and Section 7.5.2. We now turn to a standard PINN, denoted by  $u_\theta^{\text{bc}}$ , where we impose the boundary conditions in the loss function (no longer with the level-set function). The aim here is to show that our enriched methods also work with priors that do not have exact boundary conditions. To this end, we start by testing the error estimation and evaluate the gains obtained on the same sample of parameters as in Section 7.5.1.

We deploy a new parametric PINN, denoted  $u_\theta^{\text{bc}}$ , where we consider the residual loss function  $J_r$  and the boundary loss function  $J_b$ , respectively defined in (5.6) and (5.7). These integrals are both approximated by a Monte-Carlo method considering  $N_{\text{col}} = 6\,000$  collocation points and  $N_{\text{bc}} = 2\,000$  boundary collocation points. We insist that the collocation points are resampled at each epoch during the training process. Here, we no longer strongly impose the Dirichlet boundary conditions. The hyperparameters are defined in Table 19; we use the Adam optimizer and then switch to the LBFGS optimizer after the  $n_{\text{switch}}$ -th epoch. The training loss curves are presented in Figure 17 for both priors  $u_\theta$  and  $u_\theta^{\text{bc}}$ .

**Error estimates.** We consider the first parameter  $\mu^{(1)} = (0.05, 0.22)$  presented in Section 7.5.1 and perform the same test as before, considering the two priors  $u_\theta$  and  $u_\theta^{\text{bc}}$ . The results are presented in Figure 18 for fixed  $k \in \{1, 2, 3\}$  (in  $L^2$  norm).

Network - MLP		Training - with LBFGS				Loss weights			
<i>layers</i>	40, 60, 60, 60, 40	<i>lr</i>	1.7e-2	<i>n_epochs</i>	5 000	$\omega_r$	1	$\omega_{\text{data}}$	0
$\sigma$	sine	<i>decay</i>	0.99	<i>n_switch</i>	1 000	$\omega_b$	30	$\omega_{\text{sob}}$	0
		$N_{\text{col}}$	6 000	$N_{\text{bc}}$	2 000				

Table 19: Network, training parameters (Remark 15) and loss weights for  $u_\theta^{\text{bc}}$  in the 2D low-frequency case.

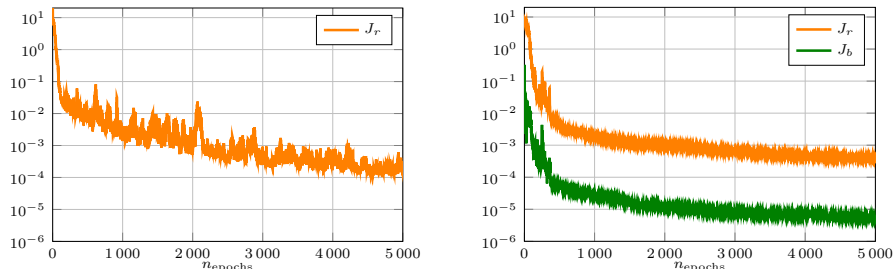


Figure 17: Training loss curves obtained in the 2D low-frequency case. Left – Considering the PINN prior  $u_\theta$  with strong boundary conditions (Section 7.5.1). Right – Considering the PINN prior  $u_\theta^{\text{bc}}$  with boundary loss training (Section 7.5.3).

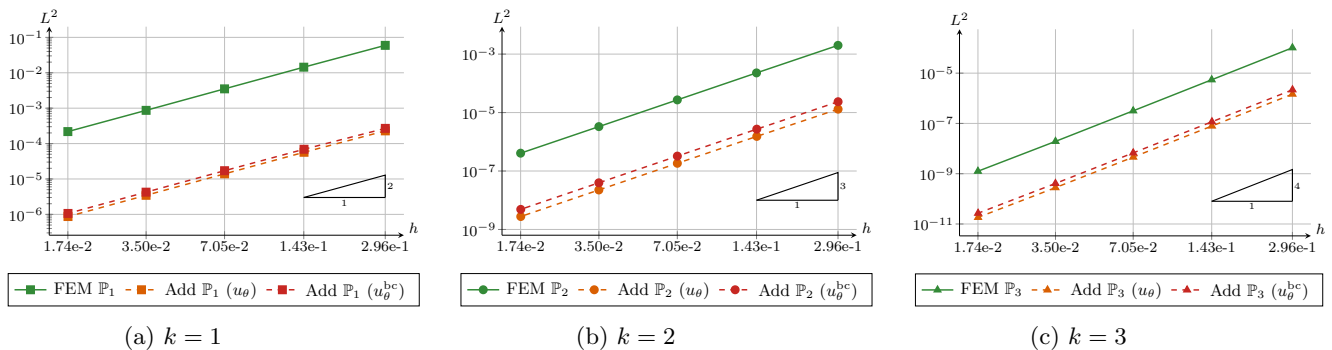


Figure 18: Considering the 2D low-frequency case with  $\mu^{(1)}$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid line) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines), with  $k = 1$ , by considering the PINN prior with standard training  $u_\theta$  and the BC loss training  $u_\theta^{\text{bc}}$ . Middle – Same with  $k = 2$ . Right – Same with  $k = 3$ .

We can see in Figure 18 that the additive approach also works when the prior is not exact on the boundary, as here with  $u_\theta^{\text{bc}}$ . In particular, for  $k \in \{1, 2, 3\}$  and the parameter  $\mu^{(1)}$ , our enriched approach using the prior  $u_\theta^{\text{bc}}$  seems to give very similar results to those obtained with  $u_\theta$  even if the approach with level-set is, for this case, slightly better.

**Gains achieved with the additive approach.** Considering the same set  $\mathcal{S}$  of  $n_p = 50$  parameter instances as in Section 7.5.1, we now evaluate the gains  $G_{+, \theta}$  and  $G_+$  defined in (7.5) considering the PINN prior  $u_\theta^{\text{bc}}$  using BC loss training. The results are presented in Table 20 for  $k \in \{1, 2, 3\}$  and  $N \in \{20, 40\}$ .

The gains reported in Table 20 show that for this test case, the use of the prior  $u_\theta$  (using the level-set) in our enriched approach, seems to give better gains than those of Table 14, considering the current prior  $u_\theta^{\text{bc}}$ . This may be due to the addition of the  $\omega_{\text{bc}}$  hyperparameter for balancing losses in training, which may make training less efficient. However, the results obtained with the current prior  $u_\theta^{\text{bc}}$  are still very good, and the gains are still significant compared to the standard FEM.

#### 7.5.4 High-frequency case

To increase in complexity, we investigate a higher-frequency problem by taking  $\kappa = 8$  in (7.12). In this section, we start by testing the error estimates. Then, we compare the different methods and evaluate the gains obtained on a sample of parameters.

k	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	20	21.37	54.17	34.72	6.45	81.13	246.75	168.01	45.92
	40	85.88	220.61	139.04	26.75	80.46	236.88	162.51	43.23
2	20	356.19	1 248.67	579.58	179.56	48.74	91.6	70.04	12.06
	40	3 046.74	10 658.21	4 959.71	1 534.67	48.42	90.91	69.6	11.98
3	20	5 100.32	16 798.35	8 398.62	2 524.2	27.54	52.09	40.41	6.78
	40	92 448.59	308 340.35	152 537.79	46 156.7	27.93	52.21	40.55	6.75

Table 20: Considering the  $2D$  low-frequency case,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta^{\text{bc}}$  (BC loss training). Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

This time, we use the Fourier features from [Tea20] as presented in Section 5.2.3 to construct the PINN prior  $u_\theta$ . The hyperparameters are defined in Table 21; we use the Adam optimizer and then switch to the LBFGS optimizer after the  $n_{\text{switch}}$ -th epoch. We consider  $N_{\text{col}} = 6000$  collocation points, uniformly chosen on  $\Omega$ . We impose the Dirichlet boundary conditions as in Section 7.5.1 using the level-set function and the same residual loss.

Network - MLP w/ FF		Training - with LBFGS				Loss weights			
layers	40, 60, 60, 60, 40	lr	1.7e-2	n_epochs	20 000	$\omega_r$	1	$\omega_{\text{data}}$	0
$\sigma$	sine	decay	0.99	n_switch	1 000	$\omega_b$	0	$\omega_{\text{sob}}$	0
$n_f$	40	$N_{\text{col}}$	6 000						

Table 21: Network, training parameters (Remark 15) and loss weights for  $u_\theta$  in the  $2D$  high-frequency case.

**Error estimates.** We perform the same test as in Section 7.5.1, with a standard training (Sobolev training is not considered). The results are displayed in Figure 19, where we observe the expected behavior. Indeed, all schemes have the correct order of accuracy, and the enhanced FEM has a significantly lower error constant than the classical FEM.

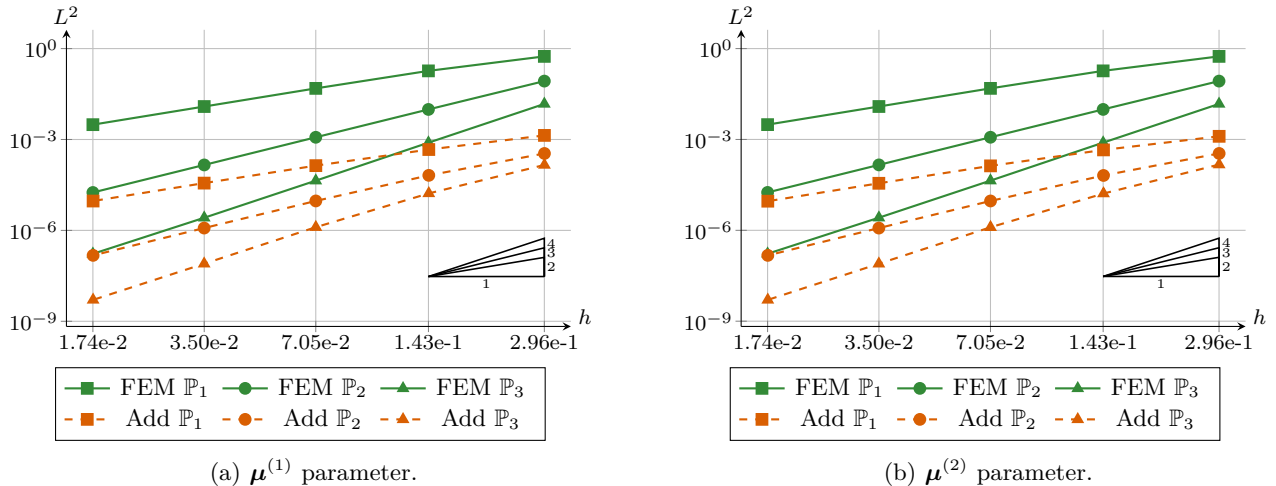


Figure 19: Considering the  $2D$  high-frequency case and the PINN prior  $u_\theta$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid lines) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines) for  $\mu^{(1)}$ , with  $k \in \{1, 2, 3\}$ . Right – Same for  $\mu^{(2)}$ .

**Comparison of different approaches.** Considering the first parameter  $\mu^{(1)}$ , we perform the same comparison as in Section 7.5.1 for the high-frequency case. The results are presented in Table 22 and Figure 20.

FEM		PINN prior $u_\theta$			
N	error	method	N	error	gain
16	$5.53 \cdot 10^{-1}$	Add	16	$1.36 \cdot 10^{-3}$	405.23
32	$1.84 \cdot 10^{-1}$		32	$4.66 \cdot 10^{-4}$	394.18

Table 22: Considering the 2D high-frequency case with  $\mu^{(1)}$ ,  $k = 1$  and  $N \in \{16, 32\}$ . Left –  $L^2$  relative error obtained with FEM. Right – Considering the PINN prior  $u_\theta$ ,  $L^2$  relative errors and gains with respect to FEM, obtained with the additive approach.

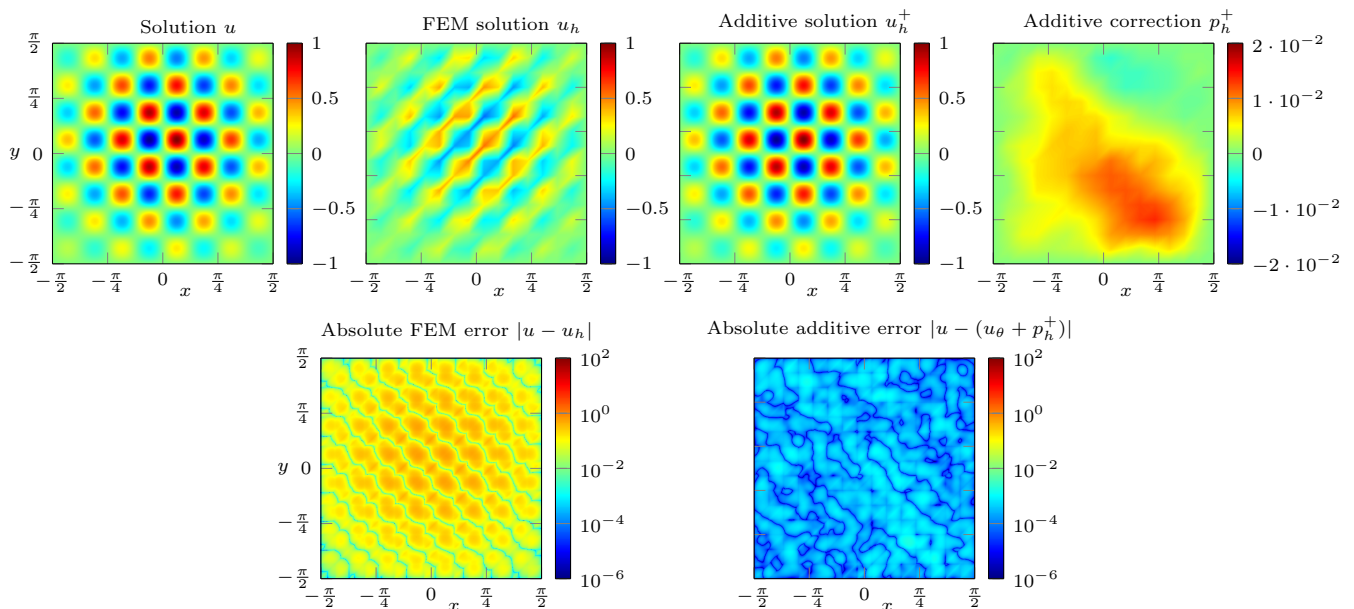


Figure 20: Considering the 2D high-frequency case with  $\mu^{(1)}$ ,  $k = 1$ ,  $N = 16$  and the PINN prior  $u_\theta$ . Comparison of the solution obtained with the standard FEM and the additive approach with the analytical solution. For the additive method, comparison of the correction term with the analytical one.

We can see here that the gains obtained in Table 22 are much better than for the “low frequency” case presented in Section 7.5.1. This is, in fact, due to FEM’s difficulty in approximating the solution for high frequencies, especially on coarse meshes. In fact, for the same choice of parameters, the FEM error on this high-frequency problem is 10 times worse than on the low-frequency one, which explains why our gains are so much greater. This also makes the use of the proposed enriched methods particularly interesting. Moreover, we note that while FEM provides a reasonable approximation of the mean of the solution (as evidenced by the second figure on the top row of Figure 20), it is unable to correctly resolve the small-scale oscillating behavior of the solution. The additive correction restores this ability, and the new solution (third figure on the top row of Figure 20) is much better able to capture the oscillations.

**Gains achieved with the additive approach.** We now evaluate the gains  $G_{+, \theta}$  and  $G_+$ , defined in (7.5), using the same set  $\mathcal{S}$  of  $n_p = 50$  parameter instances. The results are reported in Table 23 for  $k \in \{1, 2, 3\}$  and  $N \in \{20, 40\}$ .

The same results can be observed in Table 23 as for the  $\mu^{(1)}$  parameter. These could be improved by considering a Sobolev training, as in the “low-frequency” case presented in Section 7.5.2.

k	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	20	9.16	36.18	19.8	6.63	112.19	450.8	349.84	82.9
	40	26.1	111.31	58.78	19.77	106.01	388.91	308.44	71.79
2	20	35.89	164.85	87.44	29.21	65.44	210.59	159.41	38.96
	40	204.25	1088.02	516.55	179.21	52.5	139.6	109.72	22.04
3	20	100.37	533.16	252.86	88.38	32.91	81.05	63	12.73
	40	971.7	6165.56	2592.38	978.71	19.8	41.1	33.53	4.98

Table 23: Considering the 2D high-frequency case,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

## 7.6 2D anisotropic elliptic problem on a square

In this section, we will consider the (0.1) problem in a more complex form than in Section 7.5, by considering the following elliptic problem with homogeneous Dirichlet boundary conditions, in the 2D case ( $d = 2$ ),

$$\begin{cases} -\operatorname{div}(D\nabla u) = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

with  $\Omega = (0, 1)^2$ ,  $\partial\Omega$  its boundary and  $\mathcal{M} \subset \mathbb{R}^p$  the parameter space (with  $p$  the number of parameters). Considering  $\mathbf{x} = (x, y) \in \Omega$ , we define  $p = 4$  parameters  $\boldsymbol{\mu} = (\mu_1, \mu_2, \epsilon, \sigma) \in \mathcal{M} = [0.4, 0.6] \times [0.4, 0.6] \times [0.01, 1] \times [0.1, 0.8]$ . We define the (symmetric and positive definite) diffusion matrix  $D$  by

$$D(\mathbf{x}, \boldsymbol{\mu}) = \begin{pmatrix} \epsilon x^2 + y^2 & (\epsilon - 1)xy \\ (\epsilon - 1)xy & x^2 + \epsilon y^2 \end{pmatrix}$$

and the right-hand side  $f$  by

$$f(\mathbf{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(x - \mu_1)^2 + (y - \mu_2)^2}{0.025\sigma^2}\right).$$

Note that the matrix  $D$  has eigenvalues  $x^2 + y^2$  and  $\epsilon(x^2 + y^2)$ , leading to a diffusion process whose anisotropy increases as  $\epsilon$  decreases.

In this section, we consider the following three sets of parameters:

$$\boldsymbol{\mu}^{(1)} = (0.51, 0.54, 0.52, 0.55), \quad \boldsymbol{\mu}^{(2)} = (0.48, 0.53, 0.41, 0.89) \quad \text{and} \quad \boldsymbol{\mu}^{(3)} = (0.46, 0.52, 0.05, 0.12).$$

In Section 7.6.1, we start by testing the error estimates of Theorem 6. Then, we compare the two different approaches in Section 7.6.2. Finally, we evaluate the gains achieved with the additive approach as in Section 7.6.3.

**Remark 26.** In the following, the characteristic mesh size  $h = \frac{\sqrt{2}}{N-1}$  is defined as a function of  $N$ , considering a cartesian mesh of  $N^2$  nodes.

We consider a parametric PINN where we exactly impose the Dirichlet boundary conditions as presented in Section 5.2.1. Thus, we construct  $u_\theta$  as in (5.2.1) with the level-set function  $\varphi$  defined by

$$\varphi(\mathbf{x}) = x(x - 1)y(y - 1).$$

Since we impose the boundary conditions by using the level-set function, we will only consider the residual loss  $J_r$  approached by a Monte-Carlo method as defined in (5.6) with  $N_{\text{col}} = 8000$  collocation points uniformly chosen on  $\Omega \times \mathcal{M}$ . The hyperparameters are given in Table 24; we use the Adam optimizer [KB15].

**Remark 27.** Here, we do not know the analytical solution associated with the problem under consideration. So, in order to analyze the results obtained, we will define  $u$  as a reference solution  $u_{\text{ref}}$  obtained from a FEM solver on an over-refined mesh of characteristic mesh size  $h_{\text{ref}}$  and with  $k_{\text{ref}}$  polynomial order. In this section, we set  $N_{\text{ref}} = 1000$  (and the associated characteristic mesh size  $h_{\text{ref}}$ , as defined in Remark 26) and  $k_{\text{ref}} = 3$ . The linear systems resulting from the finite element discretization is solved using the Conjugate Gradient iterative method (tolerance :  $10^{-13}$ , maximum number of iterations : 10 000) preconditioned by the algebraic multigrid (AMG).

Network - MLP		Training		Loss weights					
<i>layers</i>	40, 60, 60, 60, 40	<i>lr</i>	1.6e-2	<i>n_epochs</i>	15 000	$\omega_r$	1	$\omega_{\text{data}}$	0
$\sigma$	tanh	<i>decay</i>	0.99			$\omega_b$	0	$\omega_{\text{sob}}$	0
		$N_{\text{col}}$	8 000						

Table 24: Network, training parameters (Remark 15) and loss weights for  $u_\theta$  in the 2D Elliptic case.

### 7.6.1 Error estimates

We first test the error estimates (Theorem 6) for the following two sets of parameters, randomly chosen from  $\mathcal{M}$ :  $\mu^{(1)}$  and  $\mu^{(2)}$ . We perform the same tests as in the Section 7.5, considering only the additive approach. The results are presented in Figure 21 for a fixed  $k \in \{1, 2, 3\}$  with  $N \in \{16, 32, 64, 128, 256\}$ , as presented in Remark 26.

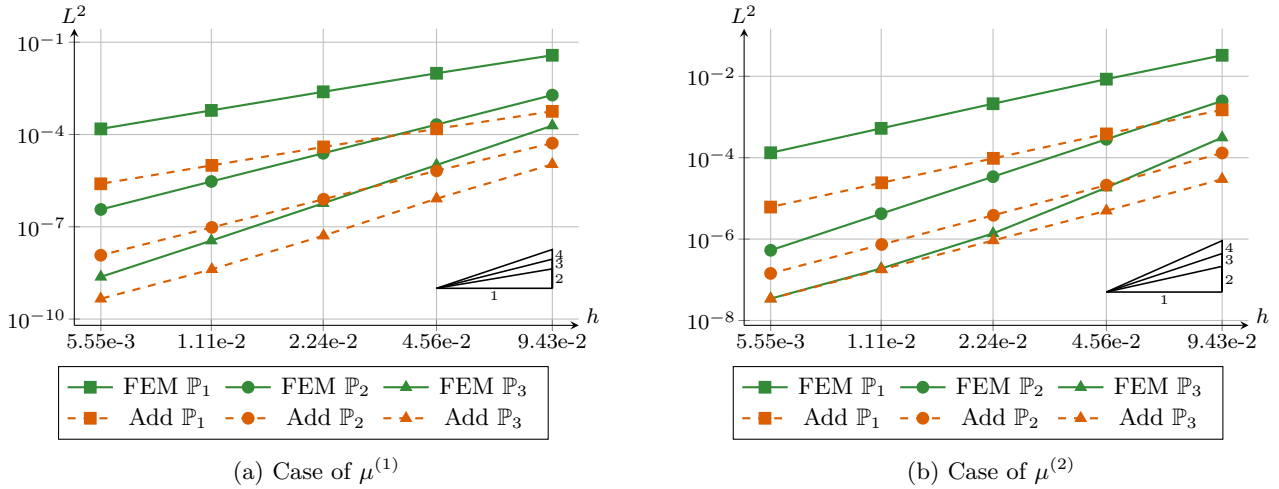


Figure 21: Considering the 2D elliptic case and the PINN prior  $u_\theta$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid lines) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines) for  $\mu^{(1)}$ , with  $k \in \{1, 2, 3\}$ . Right – Same for  $\mu^{(2)}$ .

As in the other test cases, the two approaches tested appear to respect the correct slopes of Theorem 5 and Theorem 6. The additive approach seems to be more efficient than the standard FEM for polynomial orders  $k \in \{1, 2, 3\}$  and for the two sets of parameters considered.

### 7.6.2 Comparison of different approaches

We perform the same comparison as in Section 7.5.1 for this elliptic case. We focus on the third parameter  $\mu^{(3)}$  by taking a closer look at the solution obtained with the different approaches in Figure 22 considering  $N = 16$  and  $k = 2$ .

We observe that the enriched FEM provides a more accurate solution compared to the standard FEM one. The results indicate that the additive approach is particularly effective in capturing the solution’s finer details. This demonstrates its potential in solving anisotropic problems with higher accuracy than standard methods.

### 7.6.3 Gains achieved with the additive approach

Considering a set  $\mathcal{S}$  of  $n_p = 50$  parameter instances, we will evaluate the gains  $G_{+,\theta}$  and  $G_+$  defined in (7.5). The results are presented in Table 25 for  $k \in \{1, 2, 3\}$  fixed and  $N \in \{20, 40\}$  fixed.

As in the previous test cases, the additive approach seems to be more efficient than the standard FEM for the three polynomial orders  $k \in \{1, 2, 3\}$  and for the two mesh sizes  $N \in \{20, 40\}$ . However, the gains obtained are less significant than in the previous test cases. This is due to the fact that the problem under consideration is more complex, and thus, the prior  $u_\theta$  is less accurate. Namely, PINNs have trouble converging on such highly anisotropic problems, see e.g. [ZAK22].

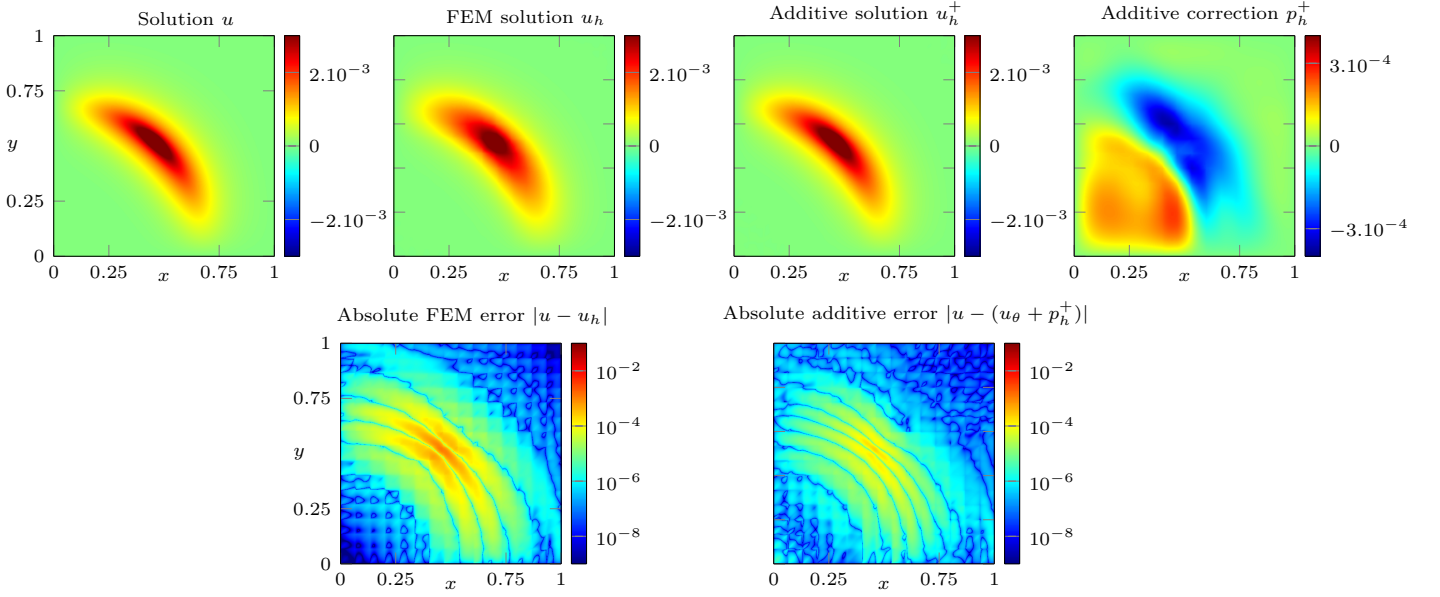


Figure 22: Considering the  $2D$  elliptic case with  $\mu^{(3)}$ ,  $k = 2$ ,  $N = 16$  and the PINN prior  $u_\theta$ . Comparison of the solution obtained with the standard FEM and the additive approach with the analytical solution. For the additive method, comparison of the correction term with the analytical one.

k	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	20	2.26	50.1	17.35	11.21	7.12	82.57	35.67	17.55
	40	4.84	185.73	62.52	43.15	6.8	77.55	33.91	16.36
2	20	15.03	623.17	213.18	143.68	3.54	35.88	18.32	7.95
	40	118.35	4942.28	1560	1096.15	2.37	31.82	14.83	7.49
3	20	151.67	3295.53	1111.11	750.19	1.33	26.51	8.32	5.02
	40	1939.42	62146.51	14375.23	12088.42	1.03	19.8	5.77	3.86

Table 25: Considering the  $2D$  elliptic case,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

## 7.7 2D Poisson problem on an annulus, with mixed boundary conditions

This section concerns the problem (0.1), considering the Poisson problem with mixed (Dirichlet and Robin) boundary conditions defined in two space dimensions ( $d = 2$ ) by

$$\begin{cases} -\Delta u = f, & \text{in } \Omega \times \mathcal{M}, \\ u = g, & \text{on } \Gamma_E \times \mathcal{M}, \\ \frac{\partial u}{\partial n} + u = g_R, & \text{on } \Gamma_I \times \mathcal{M}, \end{cases}$$

with  $\mathcal{M} \subset \mathbb{R}^p$  the parameter space (with  $p$  the number of parameters). We consider  $\Omega$  to be an annulus centered at the origin, defined by the unit circle with a circular hole of radius 0.25. Or, in other words,  $\Omega$  is defined by

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid 0.25 < \sqrt{x^2 + y^2} < 1 \right\}.$$

We then define  $\partial\Omega = \Gamma_I \cup \Gamma_E$  the boundary of  $\Omega$ , with  $\Gamma_I$  the inner boundary (the hole) and  $\Gamma_E$  the outer boundary (the unit circle). We consider the analytical solution defined for all  $\mathbf{x} = (x, y) \in \Omega$  by

$$u(\mathbf{x}, \boldsymbol{\mu}) = 1 - \frac{\ln(\mu_1 \sqrt{x^2 + y^2})}{\ln(4)},$$

with some parameters  $\boldsymbol{\mu} = \mu_1 \in [2.4, 2.6]$  ( $p = 1$  parameter), and the associated right-hand side  $f = 0$ . The Dirichlet condition  $g$  on  $\Gamma_E$  and the Robin condition  $g_R$  on  $\Gamma_I$  are defined by

$$g(\mathbf{x}, \boldsymbol{\mu}) = 1 - \frac{\ln(\mu_1)}{\ln(4)} \quad \text{and} \quad g_R(\mathbf{x}, \boldsymbol{\mu}) = 2 + \frac{4 - \ln(\mu_1)}{\ln(4)}.$$

These boundary conditions are thus parameter-dependent, contrary to the previous cases, which makes the problem more complex. To avoid geometric errors, we apply [Remark 17](#), by considering that  $g = u$  on  $\Gamma_{E,h}$  and  $g_R = \frac{\partial u}{\partial n}$  on  $\Gamma_{I,h}$ , with  $\Gamma_{E,h}$  and  $\Gamma_{I,h}$  the respective outer and inner boundaries of  $\Omega_h$ , the domain covered by the mesh. Note also that  $u_\theta$  is not exact on these approximate boundaries.

In this section, we consider the additive approach, as presented in [Section 2](#), by considering the PINN prior  $u_\theta$ . We start by testing the error estimates in [Section 7.7.1](#). Then, we compare the different approaches in [Section 7.7.2](#) and evaluate the gains obtained in [Section 7.7.3](#) on a sample of parameters.

Since the problem under consideration is parametric, we deploy a parametric PINN, which depends on both the space variable  $\mathbf{x} = (x, y) \in \Omega$  and the parameters  $\boldsymbol{\mu} = \mu_1 \in \mathcal{M}$ . To improve the derivatives' quality, we consider the Sobolev training presented in [Section 5.2.2](#). Moreover, we strongly impose the Dirichlet boundary conditions, as explained in [Section 5.2.1](#), by using the formulation proposed in [\[SS22\]](#). To do this, we define the prior

$$u_\theta = \frac{\varphi_E}{\varphi_E + \varphi_I^2} [w_\theta + \varphi_I(w_\theta - \nabla \varphi_I \cdot \nabla w_\theta - g_R)] + \frac{\varphi_I^2}{\varphi_E + \varphi_I^2} g + \varphi_E \varphi_I^2 w_\theta, \quad (7.13)$$

where  $w_\theta$  is the neural network under consideration and  $\varphi_I$  and  $\varphi_E$  are respectively the signed distance functions to  $\Gamma_I$  and  $\Gamma_E$  defined by

$$\varphi_I(\mathbf{x}) = \sqrt{x^2 + y^2} - 0.25, \quad \varphi_E(\mathbf{x}) = 1 - \sqrt{x^2 + y^2},$$

which cancels out exactly on  $\Gamma_I$  and  $\Gamma_E$ . Note that the level-sets considered are signed distance functions in this specific test case, which is not the case in the other test cases. In this test case, this is necessary because of the formulation proposed in [\(7.13\)](#) by [\[SS22\]](#).

In this case, we consider an MLP with 5 layers and a tanh activation function with the hyperparameters defined in [Table 26](#); we use the Adam optimizer [\[KB15\]](#). We consider the residual and Sobolev losses in the same way as in [Section 7.5.2](#) with  $N_{\text{col}} = 6000$  collocation points.

Network - MLP		Training - with LBFGS				Loss weights			
<i>layers</i>	40, 40, 40, 40, 40	<i>lr</i>	1e-2	<i>n<sub>epochs</sub></i>	4000	<i>ω<sub>r</sub></i>	1	<i>ω<sub>data</sub></i>	0
<i>σ</i>	tanh	<i>decay</i>	0.99	<i>n<sub>switch</sub></i>	3000	<i>ω<sub>b</sub></i>	0	<i>ω<sub>sob</sub></i>	0.1
		<i>N<sub>col</sub></i>	6000						

Table 26: Network, training parameters ([Remark 15](#)) and loss weights for  $u_\theta$  in the *2D Laplacian case on an Annulus*.

### 7.7.1 Error estimates

We start by testing the error estimation of [Theorem 6](#) for the following two sets of parameters, uniformly selected from  $\mathcal{M}$ :

$$\boldsymbol{\mu}^{(1)} = (2.51) \quad \text{and} \quad \boldsymbol{\mu}^{(2)} = (2.54),$$

by considering the PINN prior  $u_\theta$ . We perform the same tests as in the previous sections, considering only the additive approach. The results are presented in [Figure 23](#) for fixed  $k \in \{1, 2, 3\}$ .

As expected, we see in [Figure 23](#) that the error estimates are confirmed by the numerical results obtained with the standard FEM and the additive approach. The error decreases with the correct order of convergence for these two methods. Furthermore, the enriched approach provides a better accuracy than the standard FEM, as expected.

### 7.7.2 Comparison of different approaches

We perform the same comparison as in [Section 7.5.1](#) for this Laplacian case on an annulus. We focus on the first parameter  $\boldsymbol{\mu}^{(1)}$  by taking a closer look at the solution obtained with the different approaches in [Figure 24](#) considering  $h \simeq 1.67 \cdot 10^{-1}$  and  $k = 1$ .

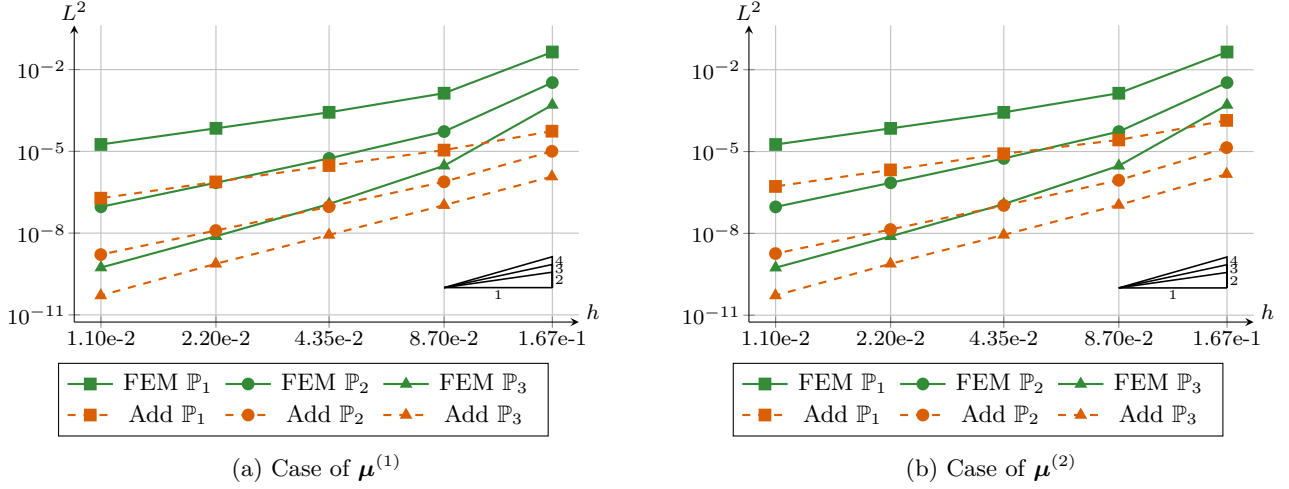


Figure 23: Considering the *2D Laplacian case on an Annulus* and the PINN prior  $u_\theta$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid lines) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines) for  $\mu^{(1)}$ , with  $k \in \{1, 2, 3\}$ . Right – Same for  $\mu^{(2)}$ .

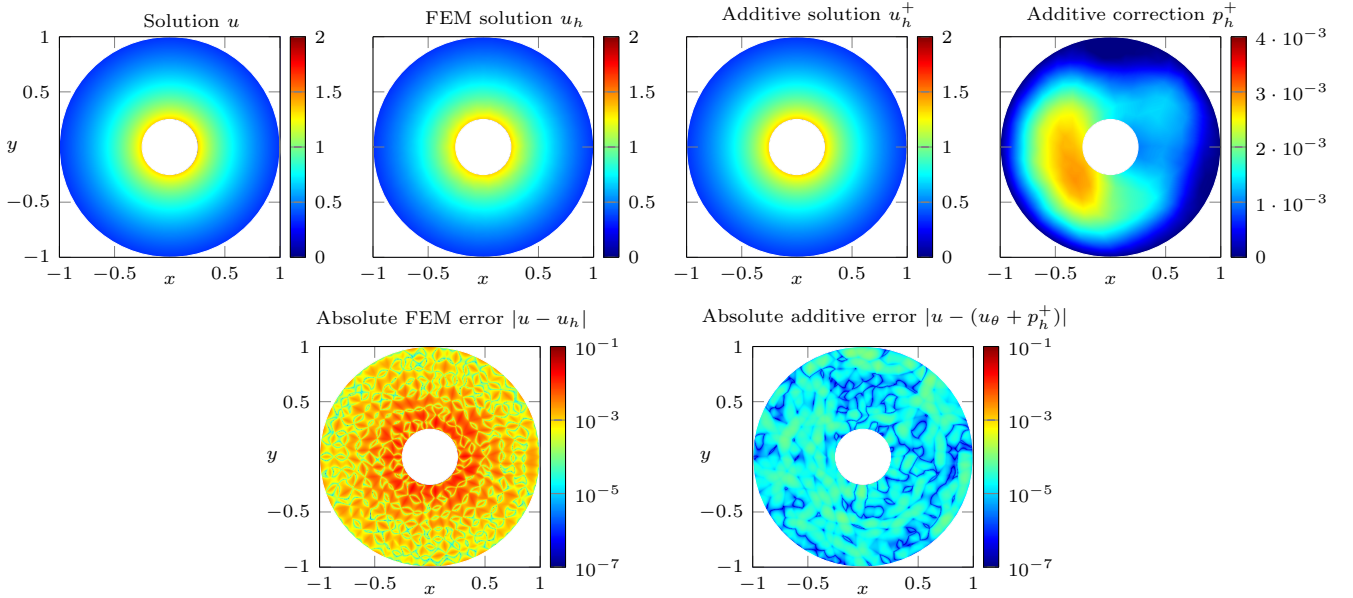


Figure 24: Considering the *2D Laplacian case on an Annulus* with  $\mu^{(1)}$ ,  $k = 1$ ,  $h \simeq 1.67 \cdot 10^{-1}$  and the PINN prior  $u_\theta$ . Comparison of the solution obtained with the standard FEM and the additive approach with the analytical solution. For the additive method, comparison of the correction term with the analytical one.

Once again, we observe that the enriched approach provides a significant improvement in accuracy compared to the standard FEM. This demonstrates the effectiveness of incorporating neural network priors in the case of mixed boundary conditions on more complex geometries than squares (here, on an annulus).

### 7.7.3 Gains achieved with the additive approach

Considering a set  $\mathcal{S}$  of  $n_p = 50$  parameter instances, we now evaluate the gains  $G_{+,\theta}$  and  $G_+$  defined in (7.5). The results are presented in Table 27 for  $k \in \{1, 2, 3\}$  and  $h \in \{1.33 \cdot 10^{-1}, 6.90 \cdot 10^{-2}\}$ .

As in previous sections, the PINN-enriched approach seems to give better results than standard FEM. For  $k = 1$  we see a gain of 50 on average for this test case, which is equivalent to refining the mesh by a factor of 7 for  $\mathbb{P}_1$  elements.

k	h	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
1	$1.33 \cdot 10^{-1}$	51.44	154.48	124.76	31.45	15.12	137.72	55.5	38.24
	$6.9 \cdot 10^{-2}$	186.44	569.39	460.53	118.34	14.35	124.91	51.72	34.68
2	$1.33 \cdot 10^{-1}$	551.64	5 688.06	3 401.69	1 674.34	31	77.46	58.41	15.46
	$6.9 \cdot 10^{-2}$	3 148.05	39 267.58	21 619.73	11 644.3	28.45	58.98	47.17	10.1
3	$1.33 \cdot 10^{-1}$	2 005.65	51 857	19 839.47	14 735.78	18.72	21.49	20.6	0.82
	$6.9 \cdot 10^{-2}$	29 270.47	681 417.4	281 187.12	197 765.09	17.92	19.85	19.33	0.57

Table 27: Considering the  $2D$  Laplacian case on an Annulus,  $k \in \{1, 2, 3\}$  and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

## 7.8 3D Poisson problem in a cube domain

We now consider the problem (7.11) of Section 7.5.1 but in three dimensions ( $d = 3$ ), in the space domain  $\Omega = (-0.5\pi, 0.5\pi)^3$  and the parameter domain  $\mathcal{M} = [-0.5, 0.5]^3$ . We define the right-hand side  $f$  such that the solution is given for all  $\mathbf{x} = (x, y, z) \in \Omega$  and  $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3) \in \mathcal{M}$  by

$$u(\mathbf{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(x - \mu_1)^2 + (y - \mu_2)^2 + (z - \mu_3)^2}{2}\right) \sin(2x) \sin(2y) \sin(2z).$$

In this section, we study the additive approach with  $k = 1$ . In Section 7.8.1, we start by testing the error estimates. Then, in Section 7.8.2, we conduct a study of the computation times of the different methods. In Section 7.8.3, we evaluate the gains achieved with the enriched approach. Finally, we discuss the importance of the prior quality in Section 7.8.4.

**Remark 28.** In this section, the cube  $\Omega$  of side length  $\pi$  is discretized using a Cartesian mesh with  $N^3$  nodes. Consequently, the characteristic mesh size is defined as a function of  $N$  by  $h = \frac{\pi\sqrt{3}}{N-1}$ .

We deploy the same type of parametric PINN as in the previous sections, where we strongly impose the Dirichlet boundary conditions. This PINN depends on both the spatial variable  $\mathbf{x}$  and the parameters  $\boldsymbol{\mu}$ . We use the prior  $u_\theta$  defined in (5.2.1), where we choose the level-set function  $\varphi$  defined by

$$\varphi(\mathbf{x}) = (x + 0.5\pi)(x - 0.5\pi)(y + 0.5\pi)(y - 0.5\pi)(z + 0.5\pi)(z - 0.5\pi).$$

The residual loss function (5.6) is approximated with  $N_{\text{col}} = 40\,000$  collocation points uniformly chosen in  $\Omega \times \mathcal{M}$ . The hyperparameters are defined in Table 28; we use the Adam optimizer and then switch to the LBFGS optimizer after the  $n_{\text{switch}}$ -th epoch.

Network - MLP		Training - with LBFGS				Loss weights			
layers	40, 60, 60, 60, 40	lr	1.7e-2	$n_{\text{epochs}}$	5 000	$\omega_r$	1	$\omega_{\text{data}}$	0
$\sigma$	tanh	decay		$n_{\text{switch}}$	2 000	$\omega_b$	0	$\omega_{\text{sob}}$	0
		$N_{\text{col}}$	40 000						

Table 28: Network, training parameters (Remark 15) and loss weights for  $u_\theta$  in the  $3D$  Poisson problem.

**Remark 29.** The results are obtained using the iterative conjugate gradient method (tolerance:  $10^{-8}$ , maximum number of iterations: 1 000) preconditioned by preconditioned by HYPRE’s BoomerAMG (a classical algebraic multigrid method) with a strong connection threshold of 0.7 and 4 levels of non-local aggregation using 2 paths (recommended options for the Poisson problem in FEniCS<sup>3</sup>). For both FE approaches, the right-hand side is interpolated on a  $\mathbb{P}^2$  space. This means

<sup>3</sup><https://github.com/FEniCS/performance-test>

that for the additive approach, the derivatives of the network are evaluated analytically (using `torch.autograd`) on the degrees of freedom associated to the space  $\mathbb{P}^2$ . To give an order of magnitude, the evaluation of the second derivatives of the prior, on the mesh with  $N = 100$  (i.e., 7880599 evaluation points for a  $\mathbb{P}^2$  space) takes 1.01 seconds on an NVIDIA H100 GPU.

### 7.8.1 Error estimates

We start by testing the error estimates of [Theorem 6](#) for the following two sets of parameters, randomly selected in  $\mathcal{M}$ :

$$\boldsymbol{\mu}^{(1)} = (0.05, 0.22, 0.1) \quad \text{and} \quad \boldsymbol{\mu}^{(2)} = (0.04, -0.08, 0.15),$$

by considering the PINN prior  $u_\theta$  and by performing the same tests as in the previous sections. The results are presented in [Figure 25](#) with  $h$  depending on  $N \in \{20, 40, 60, 80, 100\}$ , as presented in [Remark 28](#). We make the same observations as in the previous sections, both in terms of convergence rate and gains.

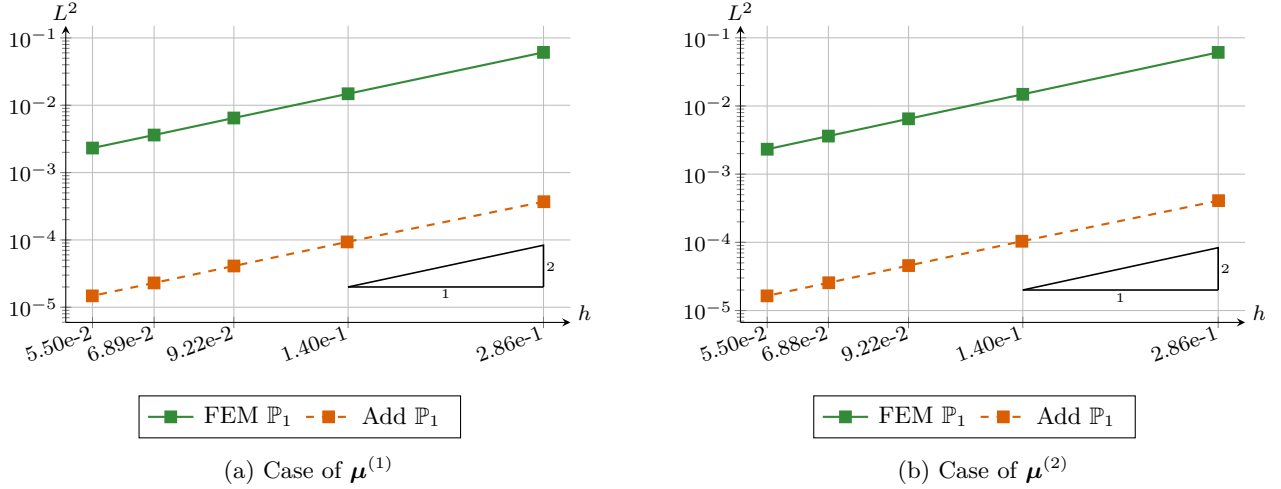


Figure 25: Considering the 3D Poisson problem and the PINN prior  $u_\theta$ . Left –  $L^2$  relative error on  $h$ , obtained with the standard FEM  $e_h^{(1)}$  (solid lines) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines) for  $\boldsymbol{\mu}^{(1)}$ , with  $k = 1$ . Right – Same for  $\boldsymbol{\mu}^{(2)}$ .

### 7.8.2 Computation times of the different methods

Just like in [Section 7.5.1](#), we are now interested in the numerical costs of the different approaches, but this time in terms of computation time. First, we perform a study in a non-parametric framework, considering only the parameter  $\boldsymbol{\mu}^{(1)}$ . Thus, in [Figure 26](#), we examine the execution time of the two approaches (training time not included for the additive approach), comparing it to the relative  $L^2$  error. In [Table 29](#), we perform a study similar to the one done previously. More precisely, for a given error  $e$ , we look at the numerical cost of the two approaches that allow us to achieve it. Therefore, for a given  $e$ , we report the characteristic mesh size  $h$ , the number of degrees of freedom  $N_{\text{dofs}}$ , and the execution time. In this purely online context on the parameter  $\boldsymbol{\mu}^{(1)}$ , the times indicated will include all the steps necessary to solve the EF problem, i.e., mesh construction, system assembly, solving the linear system, and, for the additive approach, the cost of evaluating the derivatives of the prior.

However, the offline cost of the additive approach cannot be neglected, as it includes the training time of the prior (in this case: 707.84 seconds). This is why, in [Figure 27](#), we are interested in the numerical costs of the two approaches in a parametric framework (since PINN is trained parametrically). To do this, we seek to determine, for a fixed error  $e$ , the number of parameter instances at which the additive approach is better than the standard approach, including the PINN training time. More precisely, the offline cost of the standard approach consists simply of constructing the mesh (of the size required to achieve  $e$ ), while the additive approach includes the construction of the (coarser) mesh as well as the training time. For the online phase, both approaches include all the FE steps (assembly and resolution of the system as well as the evaluation of derivatives for the enriched method). It should be noted that the online costs are much higher for the standard approach, due to a finer mesh size.

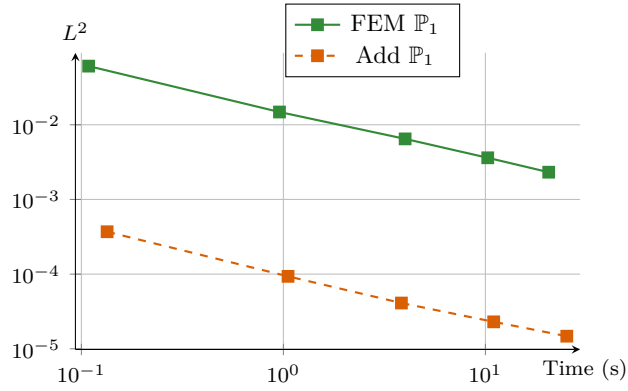


Figure 26: Considering the  $3D$  Poisson problem with  $\mu^{(1)}$ ,  $k = 1$  and the PINN prior  $u_\theta$ .  $L^2$  relative error obtained with the standard FEM  $e_h^{(1)}$  (solid lines) and the additive approach  $e_{h,+}^{(1)}$  (dashed lines) as a function of the online computation time (in seconds), including mesh construction.

e	N		N <sub>dofs</sub>		Computation time	
	FEM	Add	FEM	Add	FEM	Add
$1 \cdot 10^{-3}$	152	12	$3.51 \cdot 10^6$	$1.73 \cdot 10^3$	$(7.65 \cdot 10^1)$	$5.2 \cdot 10^{-2}$
$1 \cdot 10^{-4}$	484	39	$1.13 \cdot 10^8$	$5.93 \cdot 10^4$	$(2.8 \cdot 10^3)$	$9.26 \cdot 10^{-1}$
$1 \cdot 10^{-5}$	1539	122	$3.65 \cdot 10^9$	$1.82 \cdot 10^6$	$(1.02 \cdot 10^5)$	$(5.26 \cdot 10^1)$

Table 29: Considering the  $3D$  Poisson problem with  $\mu^{(1)}$ ,  $k = 1$  and the PINN prior  $u_\theta$ . Left – Characteristic  $N$  (associated to the characteristic mesh size  $h$ ) required to reach a fixed error  $e$  for standard FEM and the additive approach. Middle – Number of degrees of freedom  $N_{\text{dofs}}$  associated with each case. Right – Execution time (in seconds) of the online phase associated with each case. The results in brackets are extrapolations provided to give an order of magnitude.

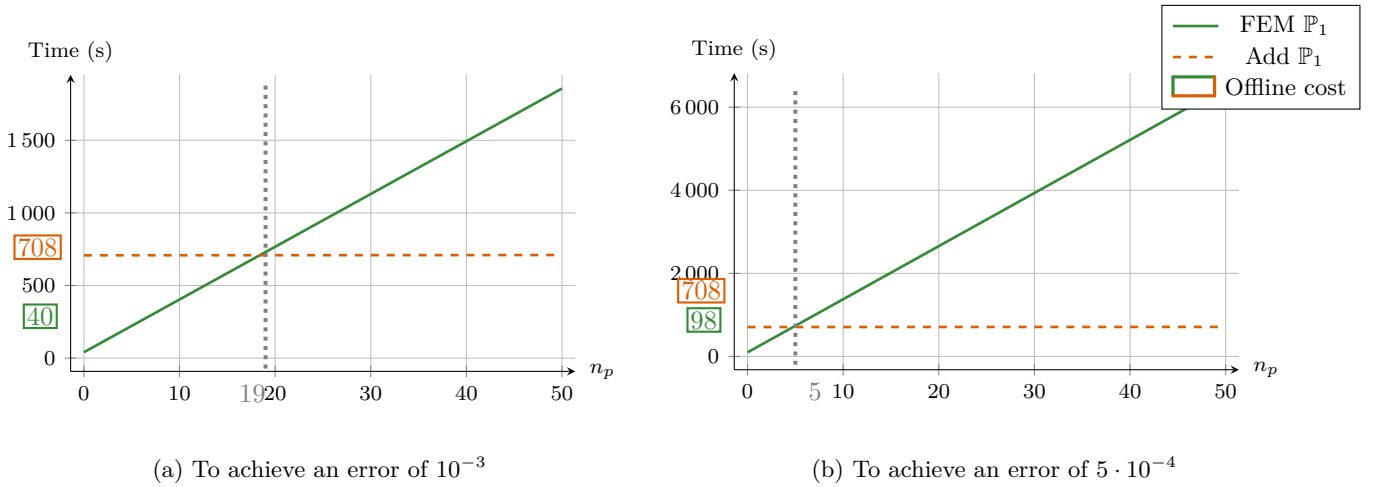


Figure 27: Computation time (in seconds) of the total process (including the offline phase) as a function of the number of parameter instances  $n_p$  to be solved. The vertical dashed line represents the number of parameter instances from which the additive approach becomes more efficient than the standard FEM.

In Figure 26, we observe that for a given error, the additive approach is significantly faster than the standard FEM. This is confirmed in Table 29, where we can see that a coarser mesh can be used to achieve a given accuracy with our enriched FEM compared to the standard FEM, leading to reduced computational time. For example, to reach an error of  $10^{-3}$ , we need  $N = 152$  (execution time: 76.5 seconds) with the standard FEM, whereas with the additive approach,

we only need  $N = 12$  (execution time: 0.052 seconds). This represents a speed-up of approximately 1471 times.

However, the cost of training the network cannot be ignored. In Figure 27, we see that when we include a parametric context (including offline cost), the additive approach becomes more advantageous than the standard approach when using  $n_p = 19$  parameter sets for an error of  $10^{-3}$  (left figure) and  $n_p = 5$  parameter sets for an error of  $5 \cdot 10^{-4}$  (right figure). And it is with this framework that approaches enriched by PINNs become interesting.

### 7.8.3 Gains achieved with the additive approach

Considering a set  $\mathcal{S}$  of  $n_p = 50$  parameter instances, we now evaluate the gains  $G_{+, \theta}$  and  $G_+$  defined in (7.5). The results are presented in Table 30 for  $k = 1$  and  $N \in \{20, 40\}$ , where we observe gains relatively close to the 2D test case (Section 7.5.1).

method	N	Gains in $L^2$ rel error of our method w.r.t. PINN				Gains in $L^2$ rel error of our method w.r.t. FEM			
		min	max	mean	std	min	max	mean	std
Add	20	8.57	20.19	14.11	2.69	75.34	166.98	137.34	24.07
	40	33.67	82.21	56.47	11.15	75.57	161.7	133.31	22.14

Table 30: Considering the 3D Poisson problem,  $k = 1$  and the PINN prior  $u_\theta$ . Left – Gains in  $L^2$  relative error of the additive method with respect to PINN. Right – Gains in  $L^2$  relative error of our approach with respect to FEM.

### 7.8.4 Influence of prior quality on the additive approach

Lastly, we focus on the impact of PINN quality on the results obtained with the additive approach. In Table 31, we look at the average gain obtained with different priors on the same sample of  $n_p = 50$  parameters as in Section 7.8.3. More specifically, we train 9 PINNs (including the PINN from the previous sections) by varying only the number of epochs  $n_{\text{epochs}}$  and the number of collocation points  $N_{\text{col}}$ .

$N_{\text{col}}$	$n_{\text{epochs}}$		
	2 500	5 000	10 000
10 000	57.27	76.38	77.74
20 000	78.33	87.79	88.26
40 000	110.52	137.34	138.59

Table 31: Considering the 3D Poisson problem,  $k = 1$ . Mean gain in  $L^2$  relative error of the additive method with respect to FEM, by varying the quality of the PINN prior  $u_\theta$  (by changing  $n_{\text{epochs}}$  and  $N_{\text{col}}$ ).

In Table 31, we can see that the number of collocation points considered seems to have a significant impact on the gains obtained. On the other hand, the number of epochs appears to be less significant. The previous results, therefore, seem to have been obtained by choosing the training among the nine that gives the best gains, while having a relatively correct number of epochs.

## 8 Conclusion and future work

In this work, we explored a new approach combining FEM and predictions from neural networks. The neural network prediction is used to enhance the FEM prediction, by correcting the FEM approximation space. Two strategies were investigated: an additive correction and a multiplicative one. For both approaches, we have proved a priori error estimates for both the  $H^1$  semi-norm and the  $L^2$  norm. We have also highlighted a link between these two techniques. Moreover, the constant appearing in these inequalities is compared with the case of classical FEM. Numerical simulations on parametric problems in one, two and three dimensions confirm our theoretical analyses. The various numerical test cases have shown that PINNs are good candidates for our enriched methods due to their ability to approximate the derivatives of the solution, which is necessary for the quality of our a priori error estimates. The non-smooth one-dimensional transmission test case, which does not fit into the theoretical framework of the two enriched approaches,

also showed a similar behavior when using an appropriate prior. In addition, the ability of PINNs to approximate the solution of the parametric PDE over a set of parameters also showed that the proposed approaches are much more interesting in terms of numerical cost than the standard method. Solutions to improve the quality of the prior and, thus, the quality of the results have also been highlighted, with Sobolev training in particular. We have also observed that the additive approach offers greater robustness and a more straightforward implementation than the multiplicative one.

The present work opens up several perspectives. For instance, the additive and multiplicative can be easily adapted to non-linear equations. Moreover, the prediction could also be used to build an optimal mesh before the FEM resolution, for instance, via a posteriori error estimates. Furthermore, more complex geometries can be considered through the use of level-sets. In this paper, we restrict our attention to regular shapes. For geometries with re-entrant corners, the corresponding singularities can be incorporated into the neural network, following the approach of [TPMM25], while a posteriori error estimators may be employed to drive mesh adaptation in the FEM correction. This direction will be explored in future work. In this work, the prior is provided by the prediction of a PINN, which offers the advantage of accurately approximating derivatives. Alternatively, a reduced-order model could serve as a prior, provided its derivatives are sufficiently precise, although this option may become computationally demanding. Finally, the use of neural operators (e.g. Fourier Neural Operators, see [LKA<sup>+</sup>21]) instead of neural networks could be an interesting alternative to explore, allowing for better generalization without being restricted to parameterized functions.

## 9 Acknowledgements

As part of the “France 2030” initiative, this work has benefited from a national grant managed by the French National Research Agency (Agence Nationale de la Recherche) attributed to the Exa-MA project of the NumPEX PEPR program, under the reference ANR-22-EXNU-0002. F. F. acknowledges funding by the European Union with ERC Project INCORWAVE – grant 101116288. This work was also supported by the Agence Nationale de la Recherche, Project PhiFEM, under grant ANR-22-CE46-0003-01.

## References

- [Aea15] M. Alnæs and J. Blechta et al. The FEniCS Project Version 1.5. *Archive of Numerical Software*, 3(100), 2015.
- [AFH<sup>+</sup>25] J. Aghili, E. Franck, R. Hild, V. Michel-Dansac, and V. Vigon. Accelerating the convergence of Newton’s method for nonlinear elliptic PDEs using Fourier neural operators. *Commun. Nonlinear Sci.*, 140(2):108434, 2025.
- [ALØ<sup>+</sup>14] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans. Math. Softw.*, 40(2):1–37, 2014.
- [AS97] R. C. Almeida and R. S. Silva. A stable Petrov-Galerkin method for convection-dominated problems. *Comput. Method. Appl. M.*, 140(3–4):291–304, 1997.
- [BBDVC<sup>+</sup>06] Y. Bazilevs, L. Beirão Da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for  $h$ -refined meshes. *Math. Models Methods Appl. Sci.*, 16(07):1031–1090, 2006.
- [BBO04] Ivo Babuška, Uday Banerjee, and John E Osborn. Generalized finite element methods—main ideas, results and perspective. *International Journal of Computational Methods*, 1(01):67–103, 2004.
- [Bea23] I. A. Baratta and J. P. Dean et al. DOLFINx: The next generation FEniCS problem solving environment, 2023.
- [BGV09] Ted Belytschko, Robert Gracie, and Giulio Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17(4):043001, 2009.
- [BHL93] G. Berkooz, P. Holmes, and J. L. Lumley. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annu. Rev. Fluid Mech.*, 25(1):539–575, 1993.

- [BLM24] S. Badia, W. Li, and A. F. Martín. Finite element interpolated neural networks for solving forward and inverse problems. *Comput. Method. Appl. M.*, 418:116505, 2024.
- [BM97] I. Babuška and J. M. Melenk. The partition of unity method. *Int. J. Numer. Meth. Eng.*, 40(4):727–758, 1997.
- [BMP<sup>+</sup>19] J.-N. Brunet, A. Mendizabal, A. Petit, N. Golse, É. Vibert, and S. Cotin. Physics-based deep neural network for augmented reality during liver surgery. In *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019*, pages 137–145. Springer International Publishing, 2019.
- [BS08] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer New York, 2008.
- [Caf98] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numer.*, 7:1–49, 1998.
- [CDCG<sup>+</sup>22] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What’s Next. *J. Sci. Comput.*, 92(3), 2022.
- [CDG08] B. Cockburn, B. Dong, and J. Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Math. Comput.*, 77(264):1887–1916, 2008.
- [CDL<sup>+</sup>23] S. Cotin, M. Duprez, V. Lleras, A. Lozinski, and K. Vuillemot.  $\phi$ -FEM: An Efficient Simulation Tool Using Simple Meshes for Problems in Structure Mechanics and Heat Transfer. In *Partition of Unity Methods*, pages 191–216. Wiley Online Library, 2023.
- [Cia02] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.
- [CLC<sup>+</sup>16] D. Canales, A. Leygue, D. Chinesta, D. González, E. Cueto, E. Feulvarch, J.-M. Bergheau, and A. Huerta. Vademecum-based GFEM (V-GFEM): optimal enrichment for transient problems. *Int. J. Numer. Meth. Eng.*, 108(9):971–989, 2016.
- [Dem23] L. F. Demkowicz. *Mathematical Theory of Finite Elements*. Society for Industrial and Applied Mathematics, 2023.
- [DHMM24] V. Dolean, A. Heinlein, S. Mishra, and B. Moseley. Multilevel domain decomposition-based architectures for physics-informed neural networks. *Comput. Method. Appl. M.*, 429:117116, 2024.
- [DL20] M. Duprez and A. Lozinski.  $\phi$ -FEM: A Finite Element Method on Domains Defined by Level-Sets. *SIAM J. Numer. Anal.*, 58(2):1008–1028, 2020.
- [DLL22] M. Duprez, V. Lleras, and A. Lozinski. A new  $\phi$ -FEM approach for problems with natural boundary conditions. *Numer. Meth. Part. D. E.*, 39(1):281–303, 2022.
- [DLL23] M. Duprez, V. Lleras, and A. Lozinski.  $\phi$ -FEM: an optimally convergent and easily implementable immersed boundary method for particulate flows and Stokes equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 57(3):1111–1142, 2023.
- [DLLV23] M. Duprez, V. Lleras, A. Lozinski, and K. Vuillemot.  $\phi$ -FEM for the heat equation: optimal convergence on unfitted meshes in space. *C. R. Math.*, 361(G11):1699–1710, 2023.
- [DRMM24] T. De Ryck, S. Mishra, and R. Molinaro. wPINNs: Weak Physics Informed Neural Networks for Approximating Entropy Solutions of Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.*, 62(2):811–841, 2024.
- [EG04] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Springer New York, 2004.
- [ES24] A. Ern and M. Steins. Convergence Analysis for the Wave Equation Discretized with Hybrid Methods in Space (HHO, HDG and WG) and the Leapfrog Scheme in Time. *J. Sci. Comput.*, 101(1), 2024.
- [Eva22] L. C. Evans. *Partial differential equations*. Number 19 in Graduate studies in mathematics. American Mathematical Society, Providence, Rhode Island, second edition, 2022.

- [EY18] W. E and B. Yu. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Commun. Math. Stat.*, 6(1):1–12, 2018.
- [FB10] T.-P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *Int. J. Numer. Meth. Eng.*, 84(3):253–304, 2010.
- [FBCD22] S. Frambati, H. Barucq, H. Calandra, and J. Diaz. Practical unstructured splines: Algorithms, multi-patch spline spaces, and some applications to numerical analysis. *J. Comput. Phys.*, 471:111625, 2022.
- [FMDN24] E. Franck, V. Michel-Dansac, and L. Navoret. Approximately well-balanced Discontinuous Galerkin methods using bases enriched with Physics-Informed Neural Networks. *J. Comput. Phys.*, 512:113144, 2024.
- [FST+24] X. Feng, H. Shangguan, T. Tang, X. Wan, and T. Zhou. A hybrid FEM-PINN method for time-dependent partial differential equations, 2024. arXiv:2409.02810 [math].
- [GKLS24] T. G. Grossmann, U. J. Komorowska, J. Latz, and C.-B. Schönlieb. Can physics-informed neural networks beat the finite element method? *IMA J. Appl. Math.*, 89(1):143–174, 2024.
- [HCB05] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Method. Appl. M.*, 194(39-41):4135–4195, 2005.
- [HMP12] R. Hiptmair, A. Moiola, and I. Perugia. Error analysis of Trefftz-discontinuous Galerkin methods for the time-harmonic Maxwell equations. *Math. Comput.*, 82(281):247–268, 2012.
- [HPS17] A. Hungria, D. Prada, and F.-J. Sayas. HDG methods for elastodynamics. *Comput. Math. Appl.*, 74(11):2671–2690, 2017.
- [IGMS22] L.-M. Imbert-Gérard, A. Moiola, and P. Stocker. A space-time quasi-Trefftz DG method for the wave equation with piecewise-smooth coefficients. *Math. Comp.*, 92(341):1211–1249, 2022.
- [JKK20] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput. Methods Appl. Mech. Engrg.*, 365:113028, 2020.
- [JKN18] V. John, P. Knobloch, and J. Novo. Finite elements for scalar convection-dominated equations and incompressible flow problems: a never ending story? *Comput. Vis. Sci.*, 19(5–6):47–63, 2018.
- [Joh02] Warren Johnson. The curious history of faa di bruno’s formula. *American Mathematical Monthly*, 109:217–234, 03 2002.
- [KB15] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [KZK21] E. Kharazmi, Z. Zhang, and G. E. M. Karniadakis. *hp*-VPINNs: Variational physics-informed neural networks with domain decomposition. *Comput. Methods Appl. Mech. Engrg.*, 374:113547, 2021.
- [LK90] H. Lee and I. S. Kang. Neural algorithm for solving differential equations. *J. Comput. Phys.*, 91(1):110–131, 1990.
- [LKA+21] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021.
- [LLF98] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.*, 9(5):987–1000, 1998.
- [MF94] A. J. Meade and A. A. Fernandez. The numerical solution of linear ordinary differential equations by feedforward neural networks. *Math. Comput. Model.*, 19(12):1–25, 1994.
- [MJLR24] N. Margenberg, R. Jendersie, C. Lessig, and T. Richter. DNN-MG: A hybrid neural network/finite element method with applications to 3D simulations of the Navier-Stokes equations. *Comput. Method. Appl. M.*, 420:116692, 2024.

- [MP17] A. Moiola and I. Perugia. A space–time Trefftz discontinuous Galerkin method for the acoustic wave equation in first-order formulation. *Numer. Math.*, 138(2):389–435, 2017.
- [Pea19] A. Paszke and S. Gross et al. *PyTorch: an imperative style, high-performance deep learning library*, pages 8026–8037. Curran Associates Inc., Red Hook, NY, USA, 2019.
- [PFB24] H. Pham, F. Faucher, and H. Barucq. Numerical investigation of stabilization in the Hybridizable Discontinuous Galerkin method for linear anisotropic elastic equation. *Comput. Method. Appl. M.*, 428:117080, 2024.
- [PFS<sup>+</sup>19] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174. IEEE, 2019.
- [PR07] A. T. Patera and E. M. Rønquist. Reduced basis approximation and *a posteriori* error estimation for a Boltzmann model. *Comput. Method. Appl. M.*, 196(29–30):2925–2942, 2007.
- [PRV<sup>+</sup>01] C. Prud’homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods. *J. Fluids Eng.*, 124(1):70–80, 2001.
- [Rav00] S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *Int. J. Numer. Meth. Fl.*, 34(5):425–448, 2000.
- [Red19] J. N. Reddy. *Introduction to the Finite Element Method, Fourth Edition*. McGraw-Hill Education, New York, N.Y., 4th edition. edition, 2019.
- [RPK19] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.
- [SBRW22] M. W. Scroggs, I. A. Baratta, C. N. Richardson, and G. N. Wells. Basix: a runtime finite element basis evaluation library. *J. Open Source Softw.*, 7(73):3982, 2022.
- [SCB01] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Comput. Method. Appl. M.*, 190(32-33):4081–4193, 2001.
- [SDRW22] M. W. Scroggs, J. S. Dokken, C. N. Richardson, and G. N. Wells. Construction of Arbitrary Order Finite Element Degree-of-Freedom Maps on Polygonal and Polyhedral Cell Meshes. *ACM Trans. Math. Softw.*, 48(2):1–23, 2022.
- [SJHH21] H. Son, J. W. Jang, W. J. Han, and H. J. Hwang. Sobolev training for physics informed neural networks, 2021.
- [SKPP24] M. Sikora, P. Krukowski, A. Paszyńska, and M. Paszyński. Comparison of Physics Informed Neural Networks and Finite Element Method Solvers for advection-dominated diffusion problems. *J. Comput. Sci.*, 81:102340, 2024.
- [SMB<sup>+</sup>20] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020.
- [SMMB00] N. Sukumar, N. Moës, B. Moran, and T. Belytschko. Extended finite element method for three-dimensional crack modelling. *Int. J. Numer. Meth. Eng.*, 48(11):1549–1570, 2000.
- [SS22] N. Sukumar and A. Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Comput. Method. Appl. M.*, 389:114333, 2022.
- [SY06] W. Sun and Y.-X. Yuan. Quasi-Newton Methods. In *Numerical Optimization*, New York, NY, 2006. Springer.

- [Tea20] M. Tancik and P. Srinivasan et al. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.
- [TPMM25] J. M. Taylor, D. Pardo, and J. Muñoz-Matute. Regularity-conforming neural networks (ReCoNNs) for solving partial differential equations, 2025.
- [vDSG25] K. Škardová, A. Daby-Seesaram, and M. Genet. Finite element neural network interpolation: Part I—interpretable and adaptive discretization for solving PDEs. *Comput. Mech.*, 2025.
- [Wah95] L. B. Wahlbin. *Superconvergence in Galerkin Finite Element Methods*. Springer Berlin Heidelberg, 1995.
- [WLZ25] D. Wang, H. Li, and Q. Zhang. General enrichments of stable GFEM for interface problems: Theory and extreme learning machine construction. *Appl. Numer. Math.*, 214:143–159, 2025.
- [XLBJ25] W. Xiong, X. Long, S. P. A. Bordas, and C. Jiang. The deep finite element method: A deep learning framework integrating the physics-informed neural networks with the finite element method. *Comput. Method. Appl. M.*, 436:117681, 2025.
- [XPYZ25] Z. Xiang, W. Peng, W. Yao, and W. Zhou. Hybrid Finite-Difference Physics-Informed Neural Networks Partial Differential Equation Solver for Complex Geometries. *J. Thermophys. Heat Tr.*, pages 1–18, 2025.
- [ZAK22] W. Zhang and M. Al Kobaisi. On the Monotonicity and Positivity of Physics-Informed Neural Networks for Highly Anisotropic Diffusion Equations. *Energies*, 15(18):6823, 2022.

## A Notations and definitions

The aim of this section is to introduce the notations used throughout the paper. We present the notations related to the parametric PDE (Table 32), to the neural network (Table 33), and to the finite element methods (Table 34).

Notation	Definition
$\Omega$	Spatial domain
$d$	Spatial dimension
$\mathbf{x} = (x_1, \dots, x_d)$	Spatial coordinates
$\mathcal{M}$	Parameter space
$p$	Number of parameters
$\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)$	Parameter vector
$M$	Lifting constant
$u$	Solution of the problem
$u_M$	Solution of the lifted problem (by $M$ )
$f$	Right-hand side of the problem
$\mathcal{L}$	Parametric differential operator of the problem
$R$	Reaction coefficient
$C$	Convection coefficient
$D$	Diffusion matrix
Pe	Péclet number

Table 32: Notations introduced for the parametric PDE.

Notation	Description
$u_\theta$	Neural network prediction of $u$
$u_{\theta,M}$	Neural network prediction of $u_M$
$\varphi$	Level-set function used to impose BCs and generate sampling in PINNs
$\theta$	Trainable parameters of the neural network
$\theta^*$	Optimal parameters of $\theta$
$J_r$	Residual loss
$J_b$	Boundary loss
$J_{\text{data}}$	Data loss
$J_{\text{sob}}$	Sobolev loss

Table 33: Notations considered for the neural network.

	Notation	Description
Standard FEM	$V_h^0$	Finite element approximation space
	$u_h$	Finite element approximation of $u$
	$h$	Characteristic size of the mesh
	$\mathcal{I}_h$	Lagrange interpolation operator
	$k$	Polynomial degree of the finite element approximation
Additive enrichment	$V_h^+$	Finite element approximation space enriched with additive prior
	$u_h^+$	Finite element approximation of $u$ in $V_h^+$
	$p_h^+$	Finite element approximation of $u - u_\theta$ in $V_h^0$
	$C_{\text{gain}}^+$	Additive gain constant
Multiplicative enrichment	$V_h^\times$	Finite element approximation space enriched with multiplicative prior
	$u_h^\times$	Finite element approximation of $u$ in $V_h^\times - M$
	$p_h^\times$	Finite element approximation of $u_M/u_{\theta,M}$ in $1 + V_h^0$
	$C_{\text{gain},H^1}^\times$	Multiplicative gain constant in $H^1$ semi-norm
	$C_{\text{gain},L^2}^\times$	Multiplicative gain constant in $L^2$ norm
	$\tilde{\mathcal{I}}_h$	Modified Lagrange interpolation operator

Table 34: Notations used in the various finite element methods.