Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 11, n. 1, 2025.

Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024

**Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**

# Interplay of Physics-Informed Neural Networks and Multiscale Numerical Methods

Antônio T. A. Gomes[1], Larissa M. da Silva[2], Frédéric Valentin[3]
LNCC, Petrópolis, RJ

**Abstract**. Physics-Informed Neural Networks (PINNs) are machine learning tools that approximate the solution of general Partial Differential Equations (PDEs) by incorporating them as terms in the loss/cost function of a Neural Network (NN). However, they can present some difficulties when the PDEs present multiscale features. To mitigate these issues, we propose an approach to embed PINNs within the framework of the Multiscale Hybrid-Mixed (MHM) method. In the MHM method, multiscale basis functions are obtained on a coarse mesh by solving completely independent local problems. Here, we propose an approach to estimate these multiscale basis functions through PINN models. Thus, the model is adjusted to generate basis functions, adapting to the structure and characteristics provided by the local domain. Through numerical validations, we show the model's capability to approximate multiscale basis functions for the Poisson problem.

**Palavras-chave**. Physics-Informed Neural Networks , Multiscale Methods, Surrogate methods

## 1 Introduction

In many problems, the analytical solution for a PDE is not available, prompting the need for numerical methods to approximate this solution, such as Finite Elements (FEM), Finite Differences (FDM), or Finite Volumes (FVM). When the solution of those problems presents a high variability in small spatial regions or short periods [1], we say that it has a multiscale behavior. In such cases, obtaining numerical solutions through standard numerical methods becomes challenging.

Multiscale numerical methods have emerged as an attractive option for dealing with problems that present multiscale behavior. The Multiscale Hybrid-Mixed (MHM for short) method, introduced in [1], is one of these options. The MHM method arises as a result of a hybrid formulation, at the continuous level, posed on a *coarse* partition $\mathscr{P}$ of the domain. Then, a decomposition of the exact solution is obtained in terms of a global problem defined on the skeleton of $\mathscr{P}$ and a set of independent local problems defined within each of the subdomains of $\mathscr{P}$. These local problems' solutions—which are typically obtained using classical numerical methods applied over a mesh defined within each subdomain of $\mathscr{P}$—are known as multiscale basis functions. Of particular interest is the fact that the multiscale basis functions can be computed locally through entirely independent problems. Although the solution of said local problems can be computed in parallel, there is still a need to solve a linear system for each subdomain.

In recent years, deep learning approaches have emerged as a promising methodology for the numerical solution of PDEs. Among these approaches are the so-called Physics-Informed Neural Networks (PINNs) [9]. The central idea behind PINNs is to minimize a functional representing the residual of the PDE and its initial and boundary conditions. This approach has shown significant success across various types of PDEs. Nevertheless, the fact that PINNs do not outperform

[1]atagomes@lncc.br
[2]lamiguez@lncc.br
[3]valentin@lncc.br

classical numerical methods in the general case [4]—in terms of both computational efficiency and accuracy—have led to specializations in different aspects [6, 7], including their integration with decomposition strategies [5]. In particular, the Finite Basis PINN (FBPINN) approach proposed in [8] tries to tackle the challenge of approximating highly oscillatory solutions. FBPINNs partition the domain into overlapping subdomains, each of which is associated to a NN trained locally, and the numerical approximation of the solution is given as a summation over the inference provided by each of the NNs. FBPINNs are nonetheless dependent on the choice of a window function that confines each NN to its local subdomain. The selection of such a function is especially challenging when dealing with high-dimensional problems or irregular and complex boundary conditions, which limits the direct applicability of the FBPINN approach.

The present work aims to explore a new methodology that integrates established properties of multiscale numerical methods, such as MHM, into PINNs. Thus, PINNs benefit from the domain decomposition techniques presented in the MHM method to improve the efficient approximation of PDEs. The premise is that partitioning the domain into non-overlapping, independent subdomains mitigates oscillations by confining them in regions of the domain without resorting to a window function; the price to pay is the computation of a global problem—which is nevertheless much cheaper than in a classical numerical method—followed by a post-processing that is done in an embarrassingly parallel way in each of the non-overlapping subdomains. The proposed methodology inherits not only some of the mathematical properties from the MHM method in terms of convergence but also the potential of PINNs to efficiently compute the multiscale basis functions from the MHM method by enabling a larger number of collocation points and taking advantage of parallelism. The numerical results show the effective approximation of these basis functions for a special case of the Poisson problem.

## 2 PINN-Based MHM

We start by considering the Poisson problem. Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, be an open bounded, polytopal domain with a Lipschitz continuous boundary $\partial\Omega$ whose unit outward normal will be denoted by $\boldsymbol{n}$. We consider the second-order elliptic problem: Find $u \in H^1(\Omega)$, such that

$$\begin{cases} -\nabla \cdot A\nabla u = f & \text{in } \Omega\,, \\ u = 0 & \text{on } \partial\Omega\,, \end{cases} \tag{1}$$

where $f \in L^2(\Omega)$. Here $A \in L^\infty(\Omega)^{d\times d}$ is a symmetric positive definite matrix uniformly bounded.

The MHM method is a byproduct of a hybrid formulation that starts at the continuous level and is posed on a coarse partition. It consists of decomposing the exact solution into local and global contributions. When discretized, such a characterization decouples local and global problems: the global formulation turns out to be responsible for the degrees of freedom over the skeleton of the coarse partition, and the local problems refer to a series of smaller calculations in different, non-overlapping parts of the domain. By solving these problems, we obtain multiscale basis functions that capture the behavior of the solution at a finer level of detail. We introduce $\mathscr{P}$, a collection of open, bounded, disjoint polytopes $K$, such that $\overline{\Omega} = \cup_{K\in\mathscr{P}} K$. Then, the exact solution $u$ is characterized with respect to the solutions of both global and local problems.

To present the MHM method we need to introduce two partitions that do not coincide but are not independent. Considering $\mathcal{E}$ as the set of faces within $\partial\mathscr{P}$ (also known as the skeleton of $\mathscr{P}$), we start by discretizing the set of faces $E \in \mathcal{E}$. For this, we introduce $\mathcal{E}_H$, a partition of $\mathcal{E}$, for which each $E \in \mathcal{E}$ may be split into faces $F$ of diameter $H_F \leq H := \max_{F\in\mathcal{E}_H} H_F$.

In addition, for each $K \in \mathscr{P}$, we introduce a shape regular family of simplicial triangulations $\{\mathcal{T}_h^K\}_{h>0}$ made up of simplices $S \in \mathcal{T}_h^K$ of diameter $h_S \leq h := \max_{K\in\mathscr{P}} \max_{S\in\mathcal{T}_h^K} h_S$.

DOI: 10.5540/03.2025.011.01.0413  

We define finite element spaces associated to $\mathcal{T}_h^K$ given as follows

$$V_h := \prod_{K\in\mathscr{P}} V_h(K) \text{ where } V_h(K) := \left\{v_h \in C^0(K) : v_h|_S \in \mathbb{P}_k(S),\, \forall\, S \in \mathcal{T}_h^K\right\}, \tag{2}$$

$$\tilde{V}_h := \prod_{K\in\mathscr{P}} \tilde{V}_h(K) \text{ where } \tilde{V}_h(K) := V_h(K) \cap L_0^2(K) \quad \text{and} \quad V_0 := \mathbb{P}_0(\mathcal{T}_h), \tag{3}$$

where $\mathbb{P}_0(\mathcal{T}_h)$ denotes the space of constant functions on $K \in \mathcal{T}_h$ and $L_0^2(K)$ is the space function $L^2(K)$ with zero mean value functions. We also introduce

$$\Lambda := \{\boldsymbol{\tau}\cdot\boldsymbol{n}^K|_{\partial K} \,:\, \boldsymbol{\tau} \in H(\mathrm{div},\Omega) \text{ for all } K \in \mathscr{P}\},$$

and from it we define a finite element space associated to $\mathcal{E}_H$ given as follows

$$\Lambda_H := \{\mu_H \in \Lambda : \mu_H|_F \in \mathbb{P}_\ell(F),\, \forall\, F \in \mathcal{E}_H\} \quad \text{with } k \geq \ell + d\,,\, \ell \geq 0, \tag{4}$$

where $\mathbb{P}_m(D)$ stands for the space of polynomials of a total degree less or equal to $m \geq 0$ on a measurable set $D \in \mathbb{R}^d$.

Using the spaces (2)-(4), the discrete mappings $T_h : \Lambda \to \tilde{V}_h$ and $\hat{T}_h : L^2(\Omega) \to \tilde{V}_h$ read:

- for all $\mu \in \Lambda$, $T_h\mu \in \tilde{V}_h$ is the unique solution of

$$\int_K A\nabla T_h\mu \cdot \nabla v_h = \langle \mu, v_h\rangle_{\partial K} \quad \text{for all } v_h \in \tilde{V}_h(K) \quad \text{and} \quad K \in \mathscr{P}\,; \tag{5}$$

- for all $q \in L^2(\Omega)$, $\hat{T}_h q \in \tilde{V}_h$ is the unique solution of

$$\int_K A\nabla \hat{T}_h q \cdot \nabla v_h = \int_K q\, v_h \quad \text{for all } v_h \in \tilde{V}_h(K) \quad \text{and} \quad K \in \mathscr{P}\,. \tag{6}$$

where the duality pairing $\langle\cdot,\cdot\rangle_{\partial K}$ denotes the interaction between a function and its dual space over the space skeleton $\partial K$. Using the mappings (5)-(6), the standard *two-level* MHM method consists of finding $(\lambda_H, u_0^h) \in \Lambda_H \times V_0$ such that

$$\begin{cases} \langle \mu_H, T_h\lambda_H\rangle_{\partial\mathscr{P}} + \langle \mu_H, u_0^h\rangle_{\partial\mathscr{P}} = -\langle \mu_H, \hat{T}_h f\rangle_{\partial\mathscr{P}} & \text{for all } \mu_H \in \Lambda_H\,, \\ \langle \lambda_H, v_0\rangle_{\partial\mathscr{P}} = -(f, v_0)_{\mathscr{P}} & \text{for all } v_0 \in V_0\,. \end{cases} \tag{7}$$

So, the approximate solution is given by

$$u_{Hh} = u_0^h + T_h\lambda_H + \hat{T}_h f\,. \tag{8}$$

$T_h$ and $\hat{T}_h$ in (5)-(6) are discrete versions of the mappings $T$ and $\hat{T}$ defined by, respectively,

$$\begin{cases} -\nabla\cdot(A\nabla T\mu) = c_K \text{ in } K, \\ A\nabla T\mu\cdot\mathbf{n} = \mu \text{ on } \partial K, \end{cases} \tag{9}$$

where $c_K := \frac{1}{|K|}\int_{\partial K}\mu$. and

$$\begin{cases} -\nabla\cdot(A\nabla \hat{T}q) = q - \bar{q} \text{ in } K, \\ A\nabla \hat{T}q\cdot\mathbf{n} = 0 \text{ on } \partial K, \end{cases} \tag{10}$$

where $\bar{q} = \frac{1}{|K|}\int_K q$. The standard *one-level* MHM method corresponds to solving (7) with $T_h$ and $\hat{T}_h$ replaced by $T$ and $\hat{T}$.

DOI: 10.5540/03.2025.011.01.0413  

Note that there is a similarity between the local problems (9)-(10) and the original problem (1). In MHM, the differential operator associated with the original problem is typically embedded within the local problems, not in the global problem.

The PINN-based MHM is a general approach built from the local problem formulation of the one-level MHM to solve bigger and multiscale problems based on PDEs. Our main goal in this paper is to validate this methodology for the Poisson problem presented in (1), which is achieved using a combination of domain decomposition and prediction of multiscale basis functions from the local problems (9)-(10).

Two neural networks are placed within each subdomain $K \in \mathscr{P}$ so that the first one learns the solution of $T\lambda$ and the other learns $\hat{T}f$, based on the strong formulations (9)-(10). Note that, although we have two neural networks per subdomain, by dividing the domain into many subdomains, one large optimization problem modelled as a single, "vanilla" PINN—typically demanding a large number of collocation points—is transformed into many smaller subdomain optimization problems. In the following, we provide an overview of the PINN-based MHM methodology, which is summarized in Figure 1.

1. Given a subdomain $K \in \mathscr{P}$, we generate a set of collocation points $\mathbf{x}_{PDE}^K$ and $\mathbf{x}_{BC}^K$ sampled over the interior of the subdomain $K$ and on the boundary $\partial K$, respectively. Note that, as in MHM, the PINN-based MHM can use any type of subdivision, whether conforming or non-conforming.

2. In the PINN-based MHM methodology, the solution to the problem defined by (10) and (9) is approximated by the following optimization process: Find $\theta^* = \arg\min_{\theta} \mathcal{L}(\theta)$, with

$$\mathcal{L}(\theta) := c_{PDE}\mathcal{L}_{PDE}(\theta) + c_B\mathcal{L}_B(\theta)\,,$$

where $c_{\mathrm{B}}$ and $c_{\mathrm{PDE}}$ are positive weights, $\mathcal{L}_{PDE}$ and $\mathcal{L}_B$ depend on their respective operators. This loss function has the same form as used when training vanilla PINNs, with the difference that it will not be computed for the entire domain.

For the loss function of the problem presented in (9), we have $\mathcal{L}_{PDE}$ and $\mathcal{L}_B$, respectively,

$$\mathcal{L}_{PDE}(\theta) := \frac{1}{N_{\mathrm{PDE}}} \sum_{i=1}^{N_{\mathrm{PDE}}} \left(\nabla \cdot \left(A\nabla \left(\mathrm{NN}^{\mu}\left(\mathbf{x}_{PDE}^K;\theta\right) + c_K\right)\right)\right)^2 ,$$

$$\mathcal{L}_B(\theta) := \frac{1}{N_{\mathrm{B}}} \sum_{i=1}^{N_{\mathrm{B}}} \left(\mathbf{n} \cdot \nabla \left(\mathrm{NN}^{\mu}\left(\mathbf{x}_{BD}^K;\theta\right)\right) - \mu\right)^2 ,$$

where $\mathrm{NN}^{\mu}(\cdot;\theta)$ is a function defined from a neural network (parameterized by $\theta$) in each $K$ and $\mu \in \Lambda_H$. For the problem in (10),

$$\mathcal{L}_{PDE}(\theta) := \frac{1}{N_{\mathrm{PDE}}} \sum_{i=1}^{N_{\mathrm{PDE}}} \left(\nabla \cdot \left(A\nabla \left(\mathrm{NN}^{q}\left(\mathbf{x}_{PDE}^K;\theta\right)\right)\right) + q - \bar{q}\right)^2 ,$$

$$\mathcal{L}_B(\theta) = \frac{1}{N_{\mathrm{B}}} \sum_{i=1}^{N_{\mathrm{B}}} \left(\mathbf{n} \cdot \nabla \left(\mathrm{NN}^{q}\left(\mathbf{x}_{BD}^K;\theta\right)\right)\right)^2 ,$$

where $\mathrm{NN}^{q}(\cdot;\theta)$ is a function also defined from a neural network in each $K$ and $q \in L^2(\Omega)$. We impose the (discrete) zero mean value constraint on $\mathrm{NN}^{\mu}(\cdot;\theta)$ and $\mathrm{NN}^{q}(\cdot;\theta)$ by calculating their averages over the collocation points, and removing such values of $\mathrm{NN}^{\mu}(\cdot;\theta)$ and $\mathrm{NN}^{q}(\cdot;\theta)$ at the end of the process.

DOI: 10.5540/03.2025.011.01.0413  

3. Given that $T_N\mu := \mathrm{NN}^{\mu}(\cdot;\theta)$ and $\hat{T}_N q := \mathrm{NN}^{q}(\cdot;\theta)$ approach $T\mu$ and $\hat{T}q$ with precision, the PINN-based MHM method corresponds to obtaining $u_0^N \in V_0$ and $\lambda_H \in \Lambda_H$ solving (7) with $T_N\lambda_H$ and $\hat{T}_N f$, and approximating the exact solution of (1) by

$$u \sim u_0^N + T_N\lambda_H + \hat{T}_N f. \tag{11}$$

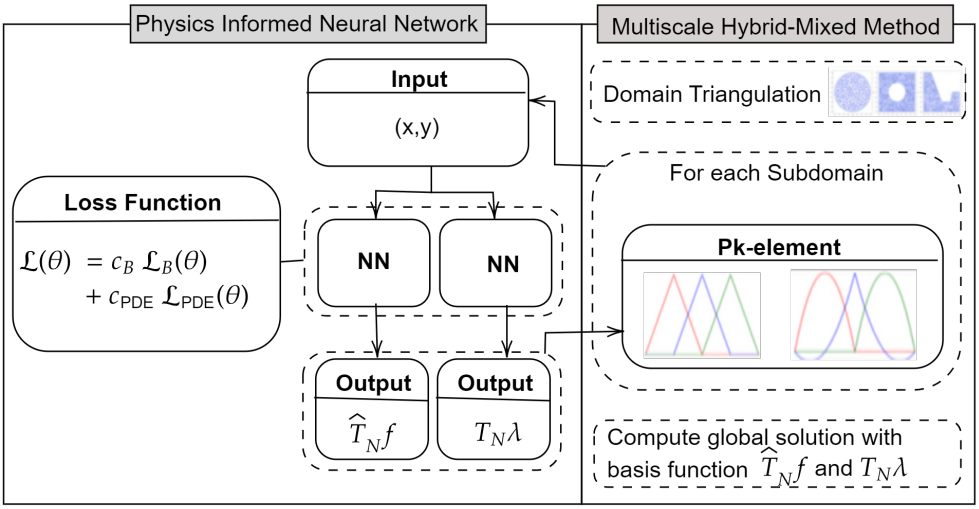



Figure 1: PINN-based MHM workflow. Source: Authors.

Another interesting aspect in the construction of the partition, in the particular case where $A$ is homogeneous throughout the domain, is that each subdomain of the mesh is the image of a reference cell through some smooth diffeomorphism referred to as a geometric mapping (c.f. [3]). As we are dealing with a simplicial mesh, the geometric transformation can be more straightforwardly computed by an affine transformation. Therefore, in this particular case $T_N\lambda$ is trained only for the reference element and not for all $K$.

## 3 Numerical Validation

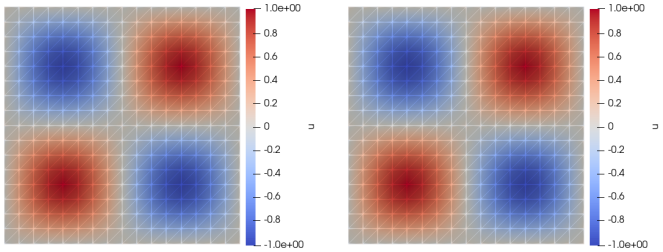
For the model problem (1), we consider $f = 8\pi^2 sin(2\pi x)sin(2\pi y)$ and the analytical solution

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

For all tests, we design a network with 3 hidden layers and in each layer, 50 neurons. The size of the training data set for the PDE loss is 200 in each subdomain for $T\lambda$ and 400 for $\hat{T}f$. For $T\lambda$, where $\lambda$ represents a basis for space $\Lambda_H$ in practice, we used a dataset of 40 points for each edge of the subdomain, which gives us 120 points for the considered triangular mesh. Additionally, the weight values assigned to different loss terms are $c_{PDE} = 10e-6$ and $c_B = 10$ for all $K \in \mathscr{P}$. Regarding $\hat{T}f$, we used a dataset of 80 points for each edge of the subdomain, i.e., 240 points. The weight values assigned were $c_{PDE} = 10e-4$ and $c_{BD} = 1$. For both cases we use $\sigma = \sin(\cdot)$ as the nonlinear activation function for all NNs used in this study and trained for the same number of optimization epochs (5000 epochs). The optimal values for the weights and biases of the proposed PINNs are attained using the Adam optimizer with a learning rate of $1e-4$.

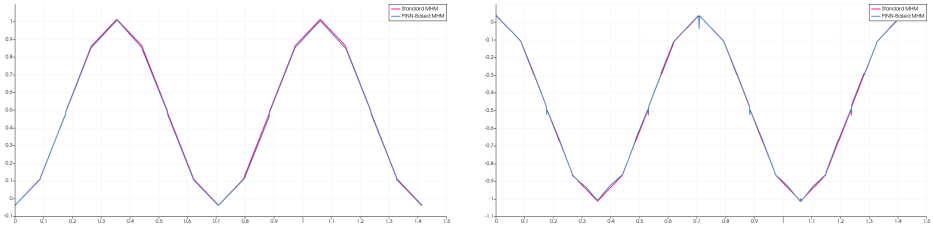
DOI: 10.5540/03.2025.011.01.0413

In this first case, we use global meshes based on triangles. We illustrate in Figure 2 a triangular global mesh with its overlaid solution. In Figures 2(a) and 2(b), we compare the solution provided by the MHM Standard method, where the basis functions are computed locally, and a solution for the PINN-based MHM where the basis functions are predicted via PINNs, respectively. The diagonal plots in Figures 2(c) and 2(d) further illustrate that the PINN-based MHM provides a good approximation for the basis functions, consequently yielding a similar solution.




(a) Final Solution MHM-Standard

(b) Final Solution PINN-Based MHM

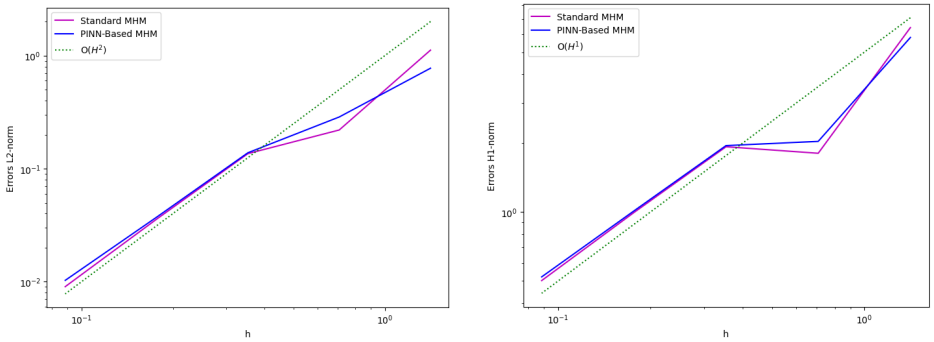


(c) Main Diagonal Solution

(d) Second Diagonal Solution

Figure 2: Comparison of Standard and PINN-Based MHM (512 elements). Source: Authors

Regarding convergence validation, we let the mesh diameter $h$ tend to 0 and compared the convergence rates of the standard MHM and PINN-based MHM methods. The results with $\ell = 0$ are illustrated in Figure 3, showing the convergence rate $\mathcal{O}(H^2)$ and $\mathcal{O}(H^1)$ in $L^2(\Omega)$ norm and $H^1(\mathcal{T}_h)$ broken-norm, respectively. It is notable that all errors for the PINN-based MHM method go to zero, aligning with the theoretical results found in [2] for the standard MHM method.




(a) $\|u - u_h\|_{L^2(\Omega)}$

(b) $\|u - u_h\|_{H_1(\mathcal{T}_h)}$

Figure 3: The mesh-based convergence on simplicial elements for $\ell = 0$. Source: Authors.

# 4 Final considerations

We adapted the techniques proposed in the PINN methodology to predict multiscale basis functions within the MHM framework. We employed this new methodology to simulate the Poisson problem. Numerical tests confirm that the PINN-based MHM methodology offers a promising approach for approximating the multiscale basis functions. It was observed that the error tends to zero as we refine our global mesh, leveraging the properties of the MHM. As future works, we intend to study the mathematical properties of the method, extend our validation to highly oscillatory problems and explore the performance of the parallel version of the PINN-based MHM, as well as subdomain refinement, to further enhance its accuracy and efficiency. Furthermore, we saw that the PINN-based MHM methodology can also benefit from how the mesh is constructed. However, this is a specific case where the data must be identical within the element, as well as the geometry. Conversely, the function $\hat{T}f$ itself depends on the function $f$ varying in the domain. In this case, we employ parameterized PINNs along with parallelization techniques for training.

# Acknowledgments

The authors were partially supported by FAPERJ E-26/201.182.2021, Project EOLIS (MATH-AMSUD 21- MATH-04). The first author was supported by CAPES. The third author was supported by CNPq/Brazil No. 309173/2020-5 and No. 401526/2022-4, and Inria/France.

# References

[1] R. Araya, C. Harder, D. Paredes, and F. Valentin. "Multiscale Hybrid-Mixed Method". In: **SIAM J. Numer. Anal.** 51.6 (2013), pp. 3505–3531.

[2] G. R. Barrenechea, F. Jaillet, D. Paredes, and F. Valentin. "The multiscale hybrid mixed method in general polygonal meshes". In: **Numerische Mathematik** 145.1 (2020), pp. 197–237.

[3] A. Ern and J. Guermond. **Finite elements I: Approximation and interpolation**. Vol. 72. Springer Nature, 2021.

[4] T G Grossmann, U J Komorowska, J Latz, and C-B Schönlieb. "Can physics-informed neural networks beat the finite element method?" In: **arXiv preprint arXiv:2302.04107** (2023).

[5] A. D. Jagtap and G. E. Karniadakis. "Extended Physics-informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition based Deep Learning Framework for Nonlinear Partial Differential Equations." In: **AAAI Spring Symposium: MLPS**. 2021.

[6] 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". In: **Computer Methods in Applied Mechanics and Engineering** 365 (2020), p. 113028.

[7] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. "Variational physics-informed neural networks for solving partial differential equations". In: **arXiv preprint arXiv:1912.00873** (2019).

[8] B. Moseley, A. Markham, and T. Nissen-Meyer. "Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations". In: **Advances in Computational Mathematics** 49.4 (2023), p. 62.

[9] 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". In: **Journal of Computational physics** 378 (2019), pp. 686–707.