Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

A hybrid physics-informed neural network based multiscale solver as a partial differential equation constrained optimization problem

Michael Hintermüller[1,2], Denis Korolev [2]

submitted: November 8, 2023

[1] Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin
Germany
E-Mail: hint@mathematik.hu-berlin.de

[2] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: michael.hintermueller@wias-berlin.de
denis.korolev@wias-berlin.de

No. 3052
Berlin 2023

2020 *Mathematics Subject Classification.* 35B27, 65K10, 65J15, 65N99, 68T20, 80M40 .

Key words and phrases. Partial differential equations, learning-informed optimal control, PDE constrained optimization, physics-informed neural networks, quasi-minimization, multiscale modelling, numerical upscaling.

The authors acknowledge the support of the Leibniz Collaborative Excellence Cluster under project ML4Sim (funding reference: K377/2021). MH also acknowledges support by the DFG ExC 2046 MATH+: Berlin Mathematics Research Center under project EF1-17.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

A hybrid physics-informed neural network based multiscale solver as a partial differential equation constrained optimization problem

Michael Hintermüller, Denis Korolev

Abstract

In this work, we study physics-informed neural networks (PINNs) constrained by partial differential equations (PDEs) and their application in approximating multiscale PDEs. From a continuous perspective, our formulation corresponds to a non-standard PDE-constrained optimization problem with a PINN-type objective. From a discrete standpoint, the formulation represents a hybrid numerical solver that utilizes both neural networks and finite elements. We propose a function space framework for the problem and develop an algorithm for its numerical solution, combining an adjoint-based technique from optimal control with automatic differentiation. The multiscale solver is applied to a heat transfer problem with oscillating coefficients, where the neural network approximates a fine-scale problem, and a coarse-scale problem constrains the learning process. We show that incorporating coarse-scale information into the neural network training process through our modelling framework acts as a preconditioner for the low-frequency component of the fine-scale PDE, resulting in improved convergence properties and accuracy of the PINN method. The relevance of the hybrid solver to numerical homogenization is discussed.

1 Introduction

Solving partial differential equations (PDEs) using physics-informed neural networks (PINNs) is currently an active area of research (see, e.g. [1] for an overview and references therein). The main principle of physics-informed learning was pioneered by [2] and later reincarnated in its modern computational interpretation by Raissi *et al.* [3]. It consists of integrating physical laws, typically in the form of the residuals of underlying PDEs, into a least-squares objective and finding the approximate solution to the corresponding residual minimization problem. The approximation ansatz $u_{\boldsymbol{\theta},n}$ for the PDE solution u from a Banach space U is then sought in a neural network class $\mathfrak{N}_{\boldsymbol{\theta},n}$. The unknown network parameters $\boldsymbol{\theta} \in \mathbb{R}^n$, the so-called weights and biases, are determined by solving an associated nonlinear and non-convex optimization problem. PINN methods are typically meshless, making them potentially useful as PDE solvers on complex geometries or in high dimensions [4], [5]. The framework is quite flexible and allows data to be easily incorporated in various ways, making PINNs not only versatile, but also a competitive approach for solving inverse problems [6]–[8]. In addition, the expressiveness of neural networks is supported by universal approximation theorems [9]–[13] and transfer learning capabilities [14], [15]. Moreover, (approximate) optimization can be performed rapidly on modern computers and compute clusters, thanks to the excellent parallelization capabilities of neural networks on GPUs, advances in automatic differentiation, as well as parallel and domain decomposition techniques [16]–[18]. On the other hand, the non-convex nature of the underlying optimization and complex nonlinear dynamics within the learning process often lead to difficulties and limitations rendering the analytical and numerical handling delicate [19], [20]. Furthermore, PINNs can

DOI 10.20347/WIAS.PREPRINT.3052 Berlin 2023

be difficult to train for problems exhibiting high-frequency or multiscale behavior [21], particularly due to the spectral bias of neural networks. This bias prioritizes learning the low-frequency modes and prevents networks from effectively learning high-frequency functions [22]. However, the combination of physics-informed neural networks with numerically robust and efficient solvers may yield a way to mitigate these challenges.

Motivated by multiscale systems, in this work we enhance neural network training by incorporating a learning-informed PDE as a constraint into the PINN optimization. For the ease of exposition, we consider here a two-scale setting only, which involves a fine-scale equation (of formidable computational complexity, perhaps beyond reach) at the fine scale and a coarse-scale equation (which we consider computationally tractable). The aforementioned computational burden may stem, e.g., from fine-scale properties of composite materials (foams, textiles, etc) resulting in highly oscillatory multiscale or high-contrast coefficients (heat conductivity, permeability, etc) or domains with multiple and not necessarily periodically scattered perforations or inclusions of complex shapes. A reliable associated simulation may then require a prohibitively fine numerical resolution. In order to remedy the enormous computational complexity, we use a neural network solver for the fine-scale problem, and it informs the coarse-scale problem through a homogenization procedure of choice. In our proposed modelling framework, this gives rise to the following PDE-constrained optimization problem:

$$\begin{cases} \inf J(y, u_{\boldsymbol{\theta},n}) \quad \text{over } (y, u_{\boldsymbol{\theta},n}), \\ \text{subject to (s.t.) } \mathcal{L}\big[u_{\boldsymbol{\theta},n}\big] y = f, \end{cases} \tag{1}$$

where J stands for a (least-squares) loss functional penalizing the PDE residual of the fine-scale equation (possibly including boundary conditions). It is worth noting that in the standard PINN framework, J depends solely on $u_{\boldsymbol{\theta},n}$. Here, however, we also introduce a coupling term to make the loss also dependent on y, thus enabling the training for the aforementioned coarse-scale enrichment. Conceptually, the additional term incorporates information on the weak convergence of the fine-scale solution to the coarse-scale one into the loss functional. By $\mathcal{L}[u_{\boldsymbol{\theta},n}] : Y \to Z$ we denote a coarse-scale differential operator between Banach spaces Y and Z, which is informed by our neural network ansatz yielding $u_{\boldsymbol{\theta},n}$. Together with some given data f, it defines an equality constraint in (1). From an optimal control perspective, the ansatz $u_{\boldsymbol{\theta},n}$ serves as a control variable, while the coarse-scale solution y acts as a state variable. We note that in our applications, the lift from the fine-scale to the coarse-scale equation is based on upscaling and the related Representative Elementary Volume (REV) concept (see, e.g. [23]–[25]). However, the abstract framework is quite general, and it allows for the use of other homogenization techniques to define possible lifts, i.e., parameterizations of $\mathcal{L}\big[u_{\boldsymbol{\theta},n}\big]$ by $u_{\boldsymbol{\theta},n}$.

In the realm of learning-informed optimal control, several works, including [26], [27], have focused on approximating nonlinear constituents or source terms in the state equation using neural networks. PINNs have also been employed as solvers for underlying state and adjoint equations in various PDE-constrained optimization scenarios [28], [29]. In [30], the neural stabilization of non-stable discrete weak formulations is proposed, resulting in a non-standard PDE-constrained optimization with neural network controls. Let us point out here that all the aforementioned techniques, while structurally perhaps similar, differ from this work. Indeed, in the usual approaches the objective is typically not related to a neural network learning problem or PDE residual minimization, but rather to minimizing specific (e.g., tracking-type) cost functionals [31], [32]. This difference has significant implications in analysis and numerical implementation. Besides, to the best of the authors' knowledge, our work is the first to deal in detail with a PINN-based optimization problem constrained by a PDE. Here, problem (1) is formulated and analyzed in a function space setting, taking into account the regularity of the fine-scale and coarse-scale PDE solutions as well as the interdependent choices of activation functions and PINN losses. The concept of quasi-minimization [33] is crucial when aiming to minimize over a

non-closed set of neural networks, and we study its applicability to our problem.

Discretizing the coarse-scale equation in (1), e.g., via the finite element method, while considering a (meshless) PINN-based approach for the fine-scale problem, leads to a *hybrid physics-informed multiscale* numerical solver. In this context, the meshless approach appears particularly useful for complex geometries. Note also that in the course of the optimization process for solving the hybrid finite dimensional approximate version of (1) possibly requires to frequently solve the discretized coarse-scale PDE. The hope now is that the coarse-scale equation can be solved numerically at a significantly lower cost (compute time) than computing the PINN solution, while still well informing low-frequency components of the fine-scale solution. Our numerical experiments provide evidence that incorporating the coarse-scale solution into the learning process through a coupling term in the objective of (1) acts as a preconditioner for the low-frequency component of the fine-scale PDE, thereby accelerating the PINN training. The proposed methodology can be used with most PINN architectures (standard PINNs [3], Fourier features networks [21], [34], FBPINNs [17], etc). It, thus, has the potential to improve existing benchmarks.

PINNs find numerous applications in multiscale systems and material design (see, e.g., [6], [28], [35], [36]). Some PINN-based homogenization techniques have also been proposed [37], [38]. However, our multiscale solver relies on a different upscaling technique and aims to obtain efficient material properties via appropriate averaging methods, which are often used for general heterogeneous media. To introduce the context of applications, we first want to highlight the two averaging methods of upscaling, which for convenience, following [24] and [25], are referred to as local and global approaches. In the local approach, the domain Ω is partitioned into grid blocks, and local auxiliary problems are solved on each grid block. The respective solutions (and related fluxes, gradients, etc) are then post-processed, typically through an averaging process, and are used to compute the upscaled properties (e.g., the effective heat conductivity, permeability, elastic constants, etc.) of each grid block. In the global approach, Ω is treated as a reasonably large heterogeneous volume consisting of the union of smaller, but possibly different blocks. Then auxiliary fine-scale problems are solved on Ω, and thereafter, the upscaled property is assigned to each block based on the local averaging of solutions to these fine-scale problems. The global approach is computationally very demanding due to its necessity for solving fine-scale problems across the entire Ω, but it has the potential to offer improved scale-up accuracy by minimizing the impact of local boundary conditions used in the local approach [25]. Problem (1) can then be regarded as a special case of our global neural upscaling procedure, given a specific choice of boundary conditions and isotropic material.

The paper is organized as follows. In Section 2, we introduce an abstract framework for our learning-informed optimization problem. We recall specific details regarding physics-informed neural networks and introduce a compression operator, which embeds information on weak convergence into our loss function. We fit (1) into a function space framework and propose a numerical algorithm for its solution. Section 4 presents a numerical homogenization technique based on averaging. We then integrate our upscaling process into the learning-informed PDE-constrained optimization setting and apply it to the heat transfer problem with oscillating coefficients.

2 A hybrid multiscale approach

In this section, we define a function space and a coupling framework for two PDEs, the fine-scale and the coarse-scale problem, respectively. For the treatment of the associated PINNs, we closely follow [33]. For the bounded domain $\Omega \subset \mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$ let $L^p(\Omega)$, $H^1(\Omega)$, $H^1_0(\Omega)$,

$H^k(\Omega)$, etc. denote the standard Lebesgue and Sobolev spaces; see, e.g., [39]. We also set $\mathbb{R}_+ := \{x \in \mathbb{R} : \ x > 0\}$ and $\mathbb{R}_{\geq 0} := \{x \in \mathbb{R} : \ x \geq 0\}$.

2.1 Function spaces and PDEs

Let $(U, \|\cdot\|_U)$, $(H, \|\cdot\|_H)$ be Hilbert spaces, $(X, \|\cdot\|_X)$ be a normed vector space, $(Z, \|\cdot\|_Z)$ be a Banach space, and X a dense subspace of U. Suppose also that U is continuously embedded into H, i.e., $U \hookrightarrow H$, and $X \hookrightarrow U$. For given $f \in H$, we consider the following partial differential equation

$$\mathcal{A}^\varepsilon u = f, \quad \text{in } \Omega, \quad \mathcal{B}u = 0, \quad \text{on } \partial\Omega, \tag{2}$$

with $\mathcal{A}^\varepsilon : X \to H$ a bounded linear partial differential operator, i.e., $\mathcal{A}^\varepsilon \in L(X, H)$, depending on the fine-scale length $\varepsilon > 0$, and $\mathcal{B} \in L(X, Z)$ defining boundary conditions. These operators are understood up to extensions given in the following theorem [40, Theorem 1.7].

Theorem 2.1. *Suppose that $(X, \|\cdot\|_U)$ is a dense subset of $(U, \|\cdot\|_U)$ and $\mathcal{A}^\varepsilon \in L(X, H)$. Then there exists a unique extension $\bar{\mathcal{A}}^\varepsilon \in L(U, H)$ of $\mathcal{A}^\varepsilon$ with $\mathcal{A}^\varepsilon v = \bar{\mathcal{A}}^\varepsilon v$ for all $v \in X$ and $\|\mathcal{A}^\varepsilon\| = \|\bar{\mathcal{A}}^\varepsilon\|$, where $\|\cdot\|$ is the operator norm. The analogous assertion holds true for $\mathcal{B}$ with $\bar{\mathcal{B}} \in L(U, Z)$.*

To continue, we need the following.

Assumption 2.1. *For every $\varepsilon > 0$,* (2) *admits a unique solution $u^\varepsilon \in U$ in the following sense:*

$$\lim_{k\to\infty} \|u_k^\varepsilon - u^\varepsilon\|_U = 0, \ \ \lim_{k\to\infty} \|\mathcal{A}^\varepsilon u_k^\varepsilon - f\|_H + \|\mathcal{B}u_k^\varepsilon\|_Z = 0, \tag{3}$$

where $\{u_k^\varepsilon\} \subset X$ (in a slight misuse of notation) is an approximating sequence of u^ε

Remark 2.1. *Let $g \in Z$ and assume that there exists $u_g \in U$ such that $\bar{\mathcal{A}}^\varepsilon u_g = 0$ in Ω and $\bar{\mathcal{B}} u_g = g$ on $\partial\Omega$. If $u^\varepsilon \in U$ satisfies Assumption 3, then $v^\varepsilon := u^\varepsilon + u_g \in U$ corresponds to* (2) *with $\mathcal{B}u = g$ on $\partial\Omega$.*

Throughout this work, we assume that Assumption 2.1 holds true, whenever this is needed. Note that if $u^\varepsilon \in U$ is the solution to (2), then $u^\varepsilon \in U_0 := \{u^\varepsilon \in U : \ \mathcal{B}u^\varepsilon = 0\} \subseteq U$, and we sometimes use $\|\cdot\|_{U_0}$ instead of $\|\cdot\|_U$. The following stability estimate is standard for least-squares residual minimization, including the least-squares finite element method [41], [42] and physics-informed neural networks [33].

Assumption 2.2 (**Stability**). *There exist a stability bound $C_s \in \mathbb{R}_+$ and an upper bound $C_b \in \mathbb{R}_+$ such that*

$$C_s\|u\|_U \leq \|\mathcal{A}^\varepsilon u\|_H + \|\mathcal{B}u\|_Z \leq C_b\|u\|_U, \quad \forall u \in U, \tag{4}$$

where C_s and C_b may depend on ε.

The stability bound (4) is well-suited for PINN problems with a soft penalization of boundary conditions, including boundary residual penalty terms in the least-squares loss. Boundary conditions can also be imposed exactly, i.e. as direct (hard) constraints, for PINNS [43]. Then, one may ask for a stability bound on U_0, which does not include the boundary term $\|\mathcal{B}u\|_Z$ and is often easier to prove [44]. We demonstrate the latter approach in our example section and use (4) for our abstract formulation.

Let $(Y, \|\cdot\|_Y)$ be a reflexive Banach space with $U_0 \subseteq Y$, Y^* the topological dual space of Y and $Y \hookrightarrow H \cong H^* \hookrightarrow Y^*$ a Gelfand triple with the compact embedding $Y \hookrightarrow H$. Let $\mathcal{L}[u] \in L(Y, Y^*)$ be parameterized by $u \in U$. The corresponding bilinear form $b_{\mathcal{L}}[u] : Y \times Y \to \mathbb{R}$ is defined as usual by

$$b_{\mathcal{L}}[u](v, w) := \langle \mathcal{L}[u]v, w\rangle_{Y^*,Y}. \tag{5}$$

Assumption 2.3 (**Sequential uniformity**). *Let $u^\varepsilon \in U$ be the solution of* (2) *and $\{u_k^\varepsilon\} \subset X$ the associated approximating sequence. We assume that the forms* (5)*, with $u = u_k^\varepsilon$ for $k \geq N_\varepsilon$ and $N_\varepsilon \geq 1$, are uniformly bounded and uniformly coercive, i.e., there exist $C_b, C_c \in \mathbb{R}_+$ such that*

$$b_{\mathcal{L}}[u_k^\varepsilon](v, w) \leq C_b\|v\|_Y\|w\|_Y, \text{ and } \quad b_{\mathcal{L}}[u_k^\varepsilon](v, v) \geq C_c\|v\|_Y^2, \tag{6}$$

for all $v, w \in Y$. In addition, (6) *holds particularly for u^ε.*

For the ease of exposition, we set $N_\varepsilon = 1$ in Assumption 2.3. In our example section we show that $N_\varepsilon \geq 1$ and Assumption 2.3 holds, in general, for the truncated approximating sequence only. For given $(u^\varepsilon, f) \in U \times H$, we consider the following partial differential equation, which we refer to as the coarse-scale problem: Find $y := y(u^\varepsilon) \in Y$ such that

$$b_{\mathcal{L}}[u^\varepsilon](y, v) = \langle f, v\rangle_{Y^*,Y} \quad \forall v \in Y. \tag{7}$$

The coarse-scale problem (7) is well-posed by the Lax–Milgram lemma. Indeed, there exists a unique solution $y \in Y$ and a bound $C \in \mathbb{R}_+$ such that $\|y\|_Y \leq C\|f\|_{Y^*}$, with C independent of u^ε due to (6).

2.2 Physics-informed neural networks

For $L \in \mathbb{N}$, an L-layer feed-forward neural network (NN) is a recursively defined function $f^L : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ with

$$f^L(x) = z^L(x),\ z^l(x) = W^l\sigma(z^{l-1}(x)) + b^l,\ 2 \leq l \leq L,\ z^1(x) = W^1x + b^1,$$

where $W^l \in \mathbb{R}^{n_l \times n_{l-1}}$ is the l-th layer weight matrix, $b_l \in \mathbb{R}^{n_l}$ is the l-th layer bias vector and $\sigma : \mathbb{R} \to \mathbb{R}$ is the activation function, which is applied component-wise in case of input arguments in $\mathbb{R}^{n_l}$. The network architecture is represented by the vector $\vec{\boldsymbol{n}} = (n_0, ..., n_L)$, and the set of all possible parameters for the fixed architecture $\vec{\boldsymbol{n}}$ is defined by

$$\Theta_n(\vec{\boldsymbol{n}}) = \left\{\{(W_j, b_j)\}_{j=1}^L : W_j \in \mathbb{R}^{n_j \times n_{j-1}},\ b_j \in \mathbb{R}^{n_j}\right\} \cong \mathbb{R}^n,$$

where n denotes the total number of parameters. We refer to a network $\boldsymbol{\theta} \in \Theta_n(\vec{\boldsymbol{n}})$ and to its realization as $v_{\boldsymbol{\theta},n}(x) := f^L(x)$. For two architectures $\vec{\boldsymbol{n}}_1 = (n_0^{(1)}, ..., n_L^{(1)})$ and $\vec{\boldsymbol{n}}_2 = (n_0^{(2)}, ..., n_L^{(2)})$ with possibly different number of parameters n_1 and n_2, we write $\vec{\boldsymbol{n}}_1 \subset \vec{\boldsymbol{n}}_2$ if for any $\boldsymbol{\theta}_1 \in \Theta_{n_1}(\vec{\boldsymbol{n}}_1)$, there exists $\boldsymbol{\theta}_2 \in \Theta_{n_2}(\vec{\boldsymbol{n}}_2)$ such that $v_{\boldsymbol{\theta}_1,n_1}(x) = v_{\boldsymbol{\theta}_2,n_2}(x)$ for $x \in \mathbb{R}^{n_0}$. For $n \in \mathbb{N}$ and a given architecture $\vec{\boldsymbol{n}}$, we define the neural network class $\mathfrak{N}_{\theta,n} := \{v_{\boldsymbol{\theta},n} : \boldsymbol{\theta} \in \Theta_n(\vec{\boldsymbol{n}})\}$ and the related map

$$\mathfrak{F}_n : \mathbb{R}^n \to \mathfrak{N}_{\theta,n}, \quad \boldsymbol{\theta} \mapsto v_{\boldsymbol{\theta},n}. \tag{8}$$

Note that the regularity of $\mathfrak{F}_n$ depends on the one of the underlying activation function in $\mathfrak{N}_{\theta,n}$. From the approximation viewpoint, we are interested in neural network classes that can well approximate elements in X; compare, e.g., [9]–[13].

Assumption 2.4 (**Uniform NN approximation of elements in** X)**.** *There exists a sequence of neural network classes* $\{\mathfrak{N}_{\theta,n}\}$ *with* $\mathfrak{N}_{\theta,n} \subset X$ *and* $\mathfrak{N}_{\theta,n} \subset \mathfrak{N}_{\theta,n+1}$ *for all* $n \in \mathbb{N}$*, and* $X \subset \overline{\cup_n \mathfrak{N}_{\theta,n}}$ *in the topology of* $(X, \|\cdot\|_X)$.

The residual minimization of the fine-scale PDE over the class of neural networks leads to the following PINN optimization problem:

$$\inf \mathcal{J}(v_{\boldsymbol{\theta},n}) := \|\mathcal{A}^\varepsilon v_{\boldsymbol{\theta},n} - f\|_H^2 + \tau_1 \|\mathcal{B} v_{\boldsymbol{\theta},n}\|_Z^2 \quad \text{over } v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n} \cap U, \tag{9}$$

where $\tau_1 > 0$ is fixed. We observe that $\mathcal{J}(u) \geq 0$ for all $u \in U$, and $\mathcal{J}(u^\varepsilon) = 0$ for the solution u^ε to (2). It is well-known that Sobolev functions can be well approximated via deep ReLU neural networks; see, e.g. [10]. However, the ReLU activation function $\sigma(x) = \max(0, x)$ admits only one weak derivative, which makes it not feasible when utilizing the standard least-squares loss with $H = L^2(\Omega)$ in the presence of the differential operator $\mathcal{A}^\varepsilon$ involving higher-order (weak) derivatives. To ensure that the $L^2(\Omega)$ loss value $\mathcal{J}(u^\varepsilon)$ is well-defined requires $u^\varepsilon \in U = H^2(\Omega)$, assuming here that $\mathcal{A}^\varepsilon$ is of the second order. The hyperbolic tangent function $\tanh(x)$ can then be used to approximate functions from Sobolev spaces $H^k(\Omega)$, $k \geq 3$, in the norm of U (see [45, Theorem B.7]). Such regularity of u^ε is not guaranteed, e.g., for elliptic PDEs on domains with a Lipschitz boundary or a source term $f \in L^2(\Omega)$ [46]. The regularity requirements can be relaxed by adopting a variational loss function [47], [48]. This involves multiplying the residuals of (2) by suitably smooth test functions and integrating by parts, thereby reducing the order of $\mathcal{A}^\varepsilon$. Then, the respective weak residual is minimized over a (discrete) trial class of neural networks. In addition, a discretization of the test space is required. Typically, piecewise polynomials of low order, leading to a Petrov-Galerkin discretization, are preferred; see [49] for a stability analysis.

In our work, we focus on approximations using smooth activation functions. Instead of imposing assumptions on the high number of admissible weak derivatives of our PDE solution, we work within a dense subspace X of U, where the approximation in the sense of Assumption 2.4 is less restrictive, such as $X = C^k(\bar{\Omega})$, which motivates the triplet of spaces $\mathfrak{N}_{\theta,n} \subset X \subset U$ and our definition of solution (3).

2.3 Weak convergence-based regularization

Let $H = L^2(\Omega)$ and $H^\delta := L^2(\Omega_\delta)$, where $\Omega_\delta := \{z \in \Omega : \text{dist}(z, \partial\Omega) > \frac{\delta}{2}\} \subset \Omega$ for $\delta > 0$. The following weak convergence assumption is common in the context of homogenization.

Assumption 2.5 (**Weak convergence**)**.** *Assume that* $u^\varepsilon \rightharpoonup y$ *in* H *as* $\varepsilon \to 0$*, where* u^ε *is the solution to* (2) *and* y *is the solution to* (7)*.*

Set $Q_\delta : H \to H^\delta$ as follows: For $v \in H$, $x \in \Omega_\delta$ and $V_\delta(x) = \{z : \|z - x\|_{\mathbb{R}^d} \leq \frac{\delta}{2}\} \subset \Omega$, let

$$(Q_\delta v)(x) = \frac{1}{|V_\delta(x)|} \int_{V_\delta(x)} v(z)\, dz.$$

We define the compression operator

$$(\bar{Q}_\delta v)(x) = \begin{cases} (Q_\delta v)(x), & x \in \Omega_\delta, \\ v(x), & x \in \Omega \setminus \Omega_\delta. \end{cases} \tag{10}$$

Let us introduce the coupling term $\mathcal{R}_\delta : Y \times U \to \mathbb{R}_{\geq 0}$ as follows:

$$\mathcal{R}_\delta(y, u^\varepsilon) := \|\bar{Q}_\delta u^\varepsilon - y\|_H^2. \tag{11}$$

The purpose of (11) is to equip the optimization problem (9) with information on weak convergence of the fine-scale solution to the coarse-scale one.

Next we study (11). For this purpose we also invoke the following.

Assumption 2.6. *There exists $C \in \mathbb{R}_+$ such that $\|u^\varepsilon\|_H \leq C\|f\|_H$ for all $\varepsilon > 0$.*

Then we have the following result.

Lemma 2.1. *The operator $Q_\delta : H \to H^\delta$ has the following properties:*

1 *Suppose that Assumption 2.6 holds. Then $\bar{Q}_\delta \in L(H) := L(H, H)$.*

2 *Suppose that Assumptions 2.5 and 2.6 hold. Then, $\lim\limits_{\varepsilon \to 0} \|Q_\delta u^\varepsilon - Q_\delta y\|_{H^\delta} = 0$.*

Proof. The linearity of Q_δ is obvious. For $x \in \Omega_\delta$, using the Cauchy–Schwarz inequality and Assumption 2.6, we obtain the estimate

$$|(Q_\delta u^\varepsilon)(x)| \leq |V_\delta(x)|^{-1/2}\|u^\varepsilon\|_H \leq |V_\delta(x)|^{-1/2} C\|f\|_H. \tag{12}$$

Since $|V_\delta(x)|$ is constant for all $x \in \Omega_\delta$, the estimate is uniform. By integrating the intermediate inequality in (12) over Ω_δ, we get $\|Q_\delta u^\varepsilon\|_{H^\delta} \leq C\|u^\varepsilon\|_H$, where $C \in \mathbb{R}_+$. Therefore, $Q_\delta \in L(H, H^\delta)$ and we readily establish that $\bar{Q}_\delta \in L(H)$.

Assumption 2.5 implies that for all test functions $v \in H$ it holds that

$$\int_\Omega u^\varepsilon(x) v(x)\, dx \to \int_\Omega y(x) v(x)\, dx \quad \text{as} \quad \varepsilon \to 0. \tag{13}$$

Consider the normalized characteristic function

$$\bar{\chi}_{V_\delta(x)}(z) := \begin{cases} \frac{1}{|V_\delta(x)|}, & z \in V_\delta(x), \\ 0, & z \notin V_\delta(x), \end{cases}$$

as the test function in (13). Then one obtains the pointwise convergence of averages $(Q_\delta u^\varepsilon)(x) \to (Q_\delta y)(x)$ as $\varepsilon \to 0$ for $x \in \Omega_\delta$. The L^2 - convergence then follows from the uniform estimate (12) and Lebesque's Dominated Convergence Theorem. □

Since $Y \hookrightarrow H$, for a given $y \in Y$, almost every $x \in \Omega_\delta$ is a Lebesgue point, i.e.,

$$\lim_{\delta \to 0} \frac{1}{|V_\delta(x)|} \int_{V_\delta(x)} y(z)\, dz = y(x). \tag{14}$$

For small $\delta > 0$, we consider the following approximation of the above limit:

$$\frac{1}{|V_\delta(x)|} \int_{V_\delta(x)} y(z)\, dz \approx \lim_{\delta \to 0} \frac{1}{|V_\delta(x)|} \int_{V_\delta(x)} y(z)\, dz = y(x).$$

Additional integrability of ∇y provides a rate for such an approximation.

Lemma 2.2. *Suppose that* $\|\nabla y\|_{L^p(\Omega)} < \infty$ *with* $\Omega \subset \mathbb{R}^d$ *and* $d < p < \infty$. *Then*

$$\|y - Q_\delta y\|_{H^\delta} = \mathcal{O}(\delta^{1-d/p}),$$

where $\mathcal{O}$ *stands for the big-*$\mathcal{O}$ *Landau notation.*

Proof. First, we use a well-known trick that controls the deviation of a function from its average on convex sets (see e.g. [39, Lemma 4.28]). Let $x \in \Omega_\delta$ and define the convex ball $V_\delta(x) \subset \Omega$. For $z \in V_\delta(x)$ and for all $t \in [0,1]$, we get

$$y(z) - y(x) = \int_0^1 \frac{d}{dt} y(x + t(z-x))dt = \int_0^1 \nabla y(x + t(z-x)) \cdot (z-x)dt.$$

Integrating the above equality over $V_\delta(x)$ and performing the change of variables $\xi = x + t(z-x)$, we obtain

$$\Big| \int_{V_{\delta(x)}} y(z)dz - |V_\delta(x)|\, y(x) \Big| \le \int_0^1 \frac{dt}{t^d} \int_{V_{t\delta}(x)} |\nabla y(\xi)| |\frac{\xi - x}{t}|\, d\xi.$$

Hölder's inequality with $\frac{1}{p} + \frac{1}{q} = 1$, in conjunction with the fact that $V_{t\delta}(x) \subseteq V_\delta(x)$, yields the following estimates:

$$\begin{aligned}\int_0^1 \frac{dt}{t^{d+1}} \int_{V_{t\delta}(x)} |\nabla y(\xi)||\xi - x| d\xi &\le \|\nabla y\|_{L^p(V_\delta(x))} \int_0^1 \frac{dt}{t^{d+1}} \Big(\int_{V_{t\delta}(x)} |\xi - x|^q d\xi \Big)^{\frac{1}{q}} \\ &\le \|\nabla y\|_{L^p(V_\delta(x))} \int_0^1 \frac{dt}{t^{\frac{d}{p}}} \Big(\int_{V_\delta(x)} |x - z|^q dz \Big)^{\frac{1}{q}} \le \frac{\delta}{1 - \frac{d}{p}} \|\nabla y\|_{L^p(V_\delta(x))} |V_\delta(x)|^{\frac{1}{q}},\end{aligned}$$

where we used the bound $|x - z| \le \delta$ and boundedness of the dt-integral for $\frac{d}{p} < 1$. Dividing by $|V_\delta(x)|$, we get

$$|y(x) - Q_\delta y(x)| \le \frac{\delta}{1 - \frac{d}{p}} |V_\delta(x)|^{-\frac{1}{p}} \|\nabla y\|_{L^p(V_\delta(x))} \le C \frac{\delta}{|V_\delta(x)|^{\frac{1}{p}}} \|\nabla y\|_{L^p(\Omega)},$$

where $C < \infty$ is some δ-independent constant. Since $|V_\delta(x)| = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} \delta^d$, where Γ is Euler's gamma function, we get the estimate $|y(x) - Q_\delta y(x)| \le C\delta^{1-d/p}$, from where we readily get the desired result. □

Under a mild assumption on the regularity of $y \in Y$, we prove the following result.

Theorem 2.2 (**Upscaling consistency**). *Suppose that Assumptions 2.5, 2.6 hold,* $\delta = \mathcal{O}(\varepsilon)$ *and* $Y \subset H^2(\Omega)$ *with* $\Omega \subset \mathbb{R}^d$, $d \le 3$. *Then,* $\lim_{\varepsilon \to 0} \mathcal{R}_\delta(y, u^\varepsilon) = 0$.

Proof. The triangle inequality and Young's inequality yield

$$\|\bar{Q}_\delta u^\varepsilon - y\|_H^2 \le 2\|Q_\delta u^\varepsilon - Q_\delta y\|_{H^\delta}^2 + 2\|y - Q_\delta y\|_{H^\delta}^2 + \|u^\varepsilon - y\|_{L^2(\Omega \setminus \Omega_\delta)}^2. \tag{15}$$

Assumption 2.5 and Lemma 2.1 imply that the first term on the right-hand side of (15) vanishes as $\varepsilon \to 0$. Since $\nabla y \in H^1(\Omega)$ due to $Y \subset H^2(\Omega)$, and $d \le 3$, Sobolev embedding yields $\|\nabla y\|_{L^p(\Omega)} < \infty$ for $p \le 6$. As $\delta = \mathcal{O}(\varepsilon)$ and thus $\delta \to 0$ as $\varepsilon \to 0$, Lemma 2.2 guarantees that the second term in (15) vanishes as $\varepsilon \to 0$. Since $U_0 \subset Y$, the same Sobolev embedding and the Cauchy–Schwarz inequality imply $\|u^\varepsilon - y\|_{L^2(\Omega \setminus \Omega_\delta)}^2 \le |\Omega \setminus \Omega_\delta|^{\frac{1}{2}} \|u^\varepsilon - y\|_{L^4(\Omega)}^2 \to 0$ as $\varepsilon \to 0$. □

Suppose that $Y \subset H^1(\Omega)$, but $Y \not\subset H^2(\Omega)$. Then Lemma 2.2 is not applicable. In this case, the coupling term (11) can be modified as follows:

$$\mathcal{R}_\delta(y, u^\varepsilon) := \|\bar{Q}_\delta u^\varepsilon - \bar{Q}_\delta y\|_H^2. \tag{16}$$

Then the upscaling consistency in the sense of Theorem 2.2 is preserved.

3 Learning-informed PDE-constrained optimization

We cast our hybrid physics-informed neural network based multiscale approach into the following learning-informed PDE-constrained optimization problem:

$$\begin{cases} \inf\ J(y, v_{\boldsymbol{\theta},n}) := \mathcal{J}(v_{\boldsymbol{\theta},n}) + \tau_2 \mathcal{R}_\delta(y, v_{\boldsymbol{\theta},n}), \quad \text{over } (y, v_{\boldsymbol{\theta},n}) \in Y \times \mathfrak{N}_{\theta,n} \cap U, \\ \text{s.t. } e(y, v_{\boldsymbol{\theta},n}) = 0, \end{cases} \tag{17}$$

where $J : Y \times U \to \mathbb{R}_+$ with $\tau_1, \tau_2, \delta \in \mathbb{R}_{\geq 0}$ fixed, $e : Y \times U \to Y^*$ is given by $e : (y, u) \mapsto e(y, u) := b_{\mathcal{L}}[u](y, \cdot) - \langle f, \cdot \rangle_{Y^*,Y}$ and the coupling term (16). The *fine-to-coarse scale map* is

$$S : U \to Y, \quad u \mapsto y(u) := S(u), \tag{18}$$

with $e(y(u), u) = 0$. Indeed, Assumption 2.3 implies that $S(u) \in Y$ is well-defined for $u = u^\varepsilon$ and $u = u_k^\varepsilon$, $k \in \mathbb{N}$. We also need the following.

Assumption 3.1 (**Continuity**). *Let $u^\varepsilon \in U$ be the solution of* (2) *and $\{u_k^\varepsilon\} \subset X$ the associated approximating sequence. Then $S(u_k^\varepsilon) \rightharpoonup S(u^\varepsilon)$ in Y as $k \to \infty$.*

Eliminating y from the set of independent variables in (17) results in the reduced optimization problem

$$\inf\ \widehat{J}(v_{\boldsymbol{\theta},n}) := J(S(v_{\boldsymbol{\theta},n}), v_{\boldsymbol{\theta},n}), \quad \text{over } v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n} \cap U. \tag{19}$$

Note that Assumption 2.3 is only invoked for the approximating sequence $\{u_k^\varepsilon\}$. Thus, poor choices of $\boldsymbol{\theta}$ may lead to the loss of coercivity or boundedness of the coarse-scale forms (5) for $u = v_{\boldsymbol{\theta},n}$. In such a situation, it is not guaranteed that $S(v_{\boldsymbol{\theta},n})$ is well-defined. In order to cope with this, the continuous dependence of (5) on u is crucial. Such a continuity property, however, depends on the specific underlying problem and the parameterization of $\mathcal{L}\big[u_{\boldsymbol{\theta},n}\big]$ by $u_{\boldsymbol{\theta},n}$. It, thus, needs to be studied case by case with respect to concrete applications. Observe further that Assumption 2.3 guarantees that for each $k \geq N_\varepsilon$, there exists $r_k \in \mathbb{R}_+$ such that we get the existence of $S(v_{\boldsymbol{\theta},n})$ for $v_{\boldsymbol{\theta},n} \in B_{r_k}(u_k^\varepsilon) := \{v \in \mathfrak{N}_{\theta,n} : \|u_k^\varepsilon - v\|_U \leq r_k\}$. Assumption 2.4 and the embedding $X \hookrightarrow U$ imply that the balls $B_{r_k}(u_k^\varepsilon)$ are non-empty for $n \gg 1$. For more details, we refer to Proposition 4.5 and the accompanying discussion in our example section below. In addition, we note that guaranteeing the existence of minimizers in $\mathfrak{N}_{\theta,n} \cap U$ for (19) is not possible, in general, as $\mathfrak{N}_{\theta,n}$ may not be topologically closed in U. In order to cope with this, we resort to the notion of quasi-minimization; cf. [33], see also [30]. The latter only requires the existence of an infimum of $\widehat{J}$ over $\mathfrak{N}_{\theta,n} \cap U$.

Clearly, since $\widehat{J}(\cdot) \geq 0$, $\inf \widehat{J}$ exists over $\mathfrak{N}_{\boldsymbol{\theta},n} \cap U$ for every $n \in \mathbb{N}$. Now let $\{\gamma_n\}$ be a real sequence with $\gamma_n > 0$ for all $n \in \mathbb{N}$ and $\gamma_n \downarrow 0$ as $n \to \infty$. Then, for every $n \in \mathbb{N}$, there exists $u_{\theta,n} \in \mathfrak{N}_{\boldsymbol{\theta},n}$ such that

$$\widehat{J}(u_{\boldsymbol{\theta},n}) \leq \inf_{v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\boldsymbol{\theta},n} \cap U} \widehat{J}(v_{\boldsymbol{\theta},n}) + \gamma_n.$$

We refer to $\{u_{\boldsymbol{\theta},n}\}_{n \in \mathbb{N}}$ as a sequence of quasi-minimizers of (19). In the following result, we consider J with the coupling term (16), but extending it to (11) is straightforward.

Theorem 3.1. *Suppose that Assumptions 2.3, 2.4, 2.6, and 3.1 hold. Let* $\{u^\varepsilon_{\boldsymbol{\theta},n}\}$ *be a quasi-minimizing sequence for* $\widehat{J} : U \to \mathbb{R}_{\geq 0}$. *Then,* $\lim_{n\to\infty} \widehat{J}(u^\varepsilon_{\boldsymbol{\theta},n}) \leq \tau_2 \mathcal{R}_\delta(y, u^\varepsilon)$ *and* $\lim_{n\to\infty} \|u^\varepsilon_{\boldsymbol{\theta},n} - u^\varepsilon\|_U = 0$, *where* u^ε *is the solution to* (2) *according to Assumption 2.1, and* $y = y(u^\varepsilon)$.

Proof. Let u^ε be the solution to (2) and $\{u^\varepsilon_k\}$, $u^\varepsilon_k \in X$ for all k, be its approximating sequence (3). Assumption 2.4 and the density of X in U imply that for each k, there exists a sequence $\{v^\varepsilon_{k,n}\}_{n\in\mathbb{N}}$, with $v^\varepsilon_{k,n} \in \mathfrak{N}_{\boldsymbol{\theta},n} \cap X$, such that $v^\varepsilon_{k,n} \to u^\varepsilon_k$ in X as $n \to \infty$. Then the diagonal sequence[1] $\{v^\varepsilon_{k,k}\}_{k\in\mathbb{N}}$ satisfies $v^\varepsilon_{k,k} \to u^\varepsilon$ in U as $k \to \infty$. Here, we also used the dense embedding $X \hookrightarrow U$ and the estimate

$$\|v^\varepsilon_{k,k} - u^\varepsilon\|_U \leq C\|v^\varepsilon_{k,k} - u^\varepsilon_k\|_X + \|u^\varepsilon_k - u^\varepsilon\|_U \to 0 \quad \text{as} \quad k \to \infty,$$

with some embedding constant $C > 0$. The same embedding, the boundedness of $\mathcal{A}^\varepsilon$ and $\mathcal{B}$, respectively, and the upper bound in (4) result in the existence of $C > 0$ such that

$$\|\mathcal{A}^\varepsilon v^\varepsilon_{k,k} - f\|_H + \|\mathcal{B}v^\varepsilon_{k,k}\|_Z \leq C\|v^\varepsilon_{k,k} - u^\varepsilon_k\|_X + \|\mathcal{A}^\varepsilon u^\varepsilon_k - f\|_H + \|\mathcal{B}u^\varepsilon_k\|_Z \to 0$$

as $k \to 0$. Therefore, $\{v^\varepsilon_{k,k}\}_{k\in\mathbb{N}}$ is also an approximating sequence (3).

Since $\mathfrak{N}_{\boldsymbol{\theta},n} \subset \mathfrak{N}_{\boldsymbol{\theta},n+1}$, for all $n \in \mathbb{N}$ it holds that

$$\widehat{J}(u^\varepsilon_{\boldsymbol{\theta},n}) \leq \inf_{v\in\mathfrak{N}_{\theta,n}\cap U} \widehat{J}(v) + \gamma_n \leq \mathcal{J}(v^\varepsilon_{n,n}) + \tau_2 \mathcal{R}_\delta(S(v^\varepsilon_{n,n}), v^\varepsilon_{n,n}) + \gamma_n \tag{20}$$

with $\gamma_n > 0$ for all $n \in \mathbb{N}$ and $\gamma_n \downarrow 0$. Observe next that $\mathcal{J}(v^\varepsilon_{n,n}) \to 0$ as $n \to \infty$ by (3). Invoking the triangle inequality and $\bar{Q}_\delta \in L(H)$, we get

$$\begin{aligned}\mathcal{R}_\delta(S(v^\varepsilon_{n,n}), v^\varepsilon_{n,n}) \leq & \|\bar{Q}_\delta v^\varepsilon_{n,n} - \bar{Q}_\delta y\|^2_H + \|\bar{Q}_\delta\|^2 \|S(v^\varepsilon_{n,n}) - y\|^2_H \\ & + 2\|\bar{Q}_\delta\| \|\bar{Q}_\delta v^\varepsilon_{n,n} - \bar{Q}_\delta y\|_H \|S(v^\varepsilon_{n,n}) - y\|_H.\end{aligned} \tag{21}$$

Then, we use Assumptions 2.3, 3.1 and the compact embedding $Y \hookrightarrow H$ to obtain $\lim_{n\to\infty} \|S(v^\varepsilon_{n,n}) - y\|^2_H = 0$. In addition, $\lim_{n\to\infty} \|\bar{Q}_\delta v^\varepsilon_{n,n} - \bar{Q}_\delta y\|^2_H = \mathcal{R}_\delta(y, u^\varepsilon)$. The first limit claim then follows from (20) and (21).

The stability estimate (4), $\tau_1 > 0$ and the inequality $(a + b) \leq 2^{1/2}(a^2 + b^2)^{1/2}$ imply that

$$C_s\|u^\varepsilon_{\boldsymbol{\theta},n} - u^\varepsilon\|_U \leq \|\mathcal{A}^\varepsilon u^\varepsilon_{\boldsymbol{\theta},n} - \mathcal{A}^\varepsilon u^\varepsilon\|_H + \sqrt{\tau_1}\|\mathcal{B}u^\varepsilon_{\boldsymbol{\theta},n} - \mathcal{B}u^\varepsilon\|_Z \leq \left(2\mathcal{J}(u^\varepsilon_{\boldsymbol{\theta},n})\right)^{\frac{1}{2}}. \tag{22}$$

An argument identical to the above in (20) while replacing $\widehat{J}(u^\varepsilon_{\boldsymbol{\theta},n})$ by $\mathcal{J}(u^\varepsilon_{\boldsymbol{\theta},n})$, we get $\lim_{n\to\infty} \mathcal{J}(u^\varepsilon_{\boldsymbol{\theta},n}) = 0$. The convergence result then follows from (22). □

If the assumptions of Theorem 2.2 are satisfied in addition to those of Theorem 3.1, then it follows that $\mathcal{R}_\delta(S(u^\varepsilon), u^\varepsilon) \to 0$ as $\varepsilon \to 0$. Theorem 3.1 suggests that $\mathcal{R}_\delta$ acts as a regularizer in the training process, and one might expect from Theorem 2.2 that its efficiency increases with decreasing ε. However, as $\varepsilon \to 0$, potential stability issues may arise, since C_s and C_b, in general, may depend on ε.

[1] We denote it also by $\{v^\varepsilon_{n,n}\}_{n\in\mathbb{N}}$, depending on the importance of the index k or n in the limit.

3.1 Discrete approximation

The (fully) discrete version of (17) comes in two steps: (i) First, we consider $\boldsymbol{\theta} \to v_{\boldsymbol{\theta},n}$, reducing every NN-function to its finite set of generating parameters belonging to $\mathbb{R}^n$; (ii) then, we replace the state y by a finite dimensional approximation y_h, with h indicating the associated discretization parameter such as, e.g., the mesh width in a finite element method [50].

Let us start with item (i) and define, in a slight mis-use of earlier notation,

$$S_n := S \circ \mathfrak{F}_n : \mathbb{R}^n \to Y, \quad \boldsymbol{\theta} \mapsto y(\boldsymbol{\theta}) := S_n(\boldsymbol{\theta}), \tag{23}$$

where $\mathfrak{F}_n$ comes from (8). We invoke the following assumptions on differentiability and invertibility.

Assumption 3.2 (**Fréchet differentiability**). *The fine-to-coarse scale map* (18) *is continuously Fréchet differentiable and* $F_n \in C^\infty(\mathbb{R}^n, \mathfrak{N}_{\theta,n})$.

Assumption 3.3. *The operators* $\mathcal{L}[v_{\boldsymbol{\theta},n}] \in L(Y, Y^*)$ *and* $e_y(y, v_{\boldsymbol{\theta},n}) \in L(Y, Y^*)$ *have bounded inverses for all* $v_{\boldsymbol{\theta},n} \in B_r(u^\varepsilon) := \{v \in \mathfrak{N}_{\theta,n} : \ \|u^\varepsilon - v\|_U \le r\}$, *where* $r \in \mathbb{R}_+$ *depends on* $\mathcal{L}$, u^ε *and* $n \in \mathbb{N}$.

Assumption 3.2 and the chain rule imply that S_n is continuously Fréchet differentiable. We also note that F_n satisfies Assumption 3.2 for smooth activation functions in a neural network, i.e., $\sigma \in C^\infty(\mathbb{R})$. Accordingly, the differentiability requirement can be reduced via reducing the one on σ. For $v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\boldsymbol{\theta},n}$ and again in a slight mis-use of notation, we set $e(y, \boldsymbol{\theta}) := e(y, v_{\boldsymbol{\theta},n})$. Via the implicit function theorem for $e(y(\boldsymbol{\theta}), \boldsymbol{\theta}) = 0$ we obtain

$$e_y(y(\boldsymbol{\theta}), \boldsymbol{\theta}) y'(\boldsymbol{\theta}) + e_{\boldsymbol{\theta}}(y(\boldsymbol{\theta}), \boldsymbol{\theta}) = 0, \tag{24}$$

where we have $y'(\boldsymbol{\theta}) \in L(\mathbb{R}^n, Y)$ and, upon obvious identification, $y'(\boldsymbol{\theta})^* \in L(Y^*, \mathbb{R}^n)$. By applying the chain rule, we find the gradient of (19):

$$\nabla \widehat{J}(\boldsymbol{\theta}) = y'(\boldsymbol{\theta})^* \partial_y J(y(\boldsymbol{\theta}), \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} J(y(\boldsymbol{\theta}), \boldsymbol{\theta}), \tag{25}$$

In practical realizations of PINNs, the second summand in (25) is typically produced by automatic differentiation, and the first summand is realized via the adjoint method [32]. Concerning the latter, Assumption 3.3 and (24) yield the adjoint equation

$$e_y(y(\boldsymbol{\theta}), \boldsymbol{\theta})^* p = -\partial_y J(y(\boldsymbol{\theta}), \boldsymbol{\theta}) = 2\tau_2 \bar{Q}_\delta^* (\bar{Q}_\delta v_{\boldsymbol{\theta},n} - \bar{Q}_\delta y(\boldsymbol{\theta})), \tag{26}$$

where $p \in Y$ denotes the adjoint variable (or adjoint state, sometimes also called co-state).

Now we come to the second step of discretization. Here we use the finite element (FE) method applied to the coarse-scale equation. More specifically, let $Y_h := \operatorname{span}\{\phi_j, \ 1 \le j \le N_h\} \subset Y$, $N_h \in \mathbb{N}$, be the standard finite dimensional space of piecewise-linear and globally continuous functions over a domain $\Omega \subset \mathbb{R}^d$, possibly including boundary conditions [50]. Of course, other choices of Y_h are possible as well. The finite element approximation of (7), which involves the neural network based function $v_{\boldsymbol{\theta},n}$ as data, is then obtained by a standard Galerkin projection: Find $y_h := y_h(v_{\boldsymbol{\theta},n}) \in Y_h$ such that

$$b_{\mathcal{L}}[v_{\boldsymbol{\theta},n}](y_h, v_h) = \langle f, v_h \rangle_{Y^*,Y} \quad \forall v_h \in Y_h. \tag{27}$$

Assumption 3.3 implies that (27) admits a unique solution $y_h(v_{\boldsymbol{\theta},n}) \in Y_h$ for all $v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n}$. The adjoint equation is discretized similarly: Find $p_h \in Y_h$ such that

$$\langle e_y(y_h(\boldsymbol{\theta}), \boldsymbol{\theta})^* p_h, v_h \rangle_{Y^*,Y} = 2\tau_2 \langle \bar{Q}_\delta v_{\boldsymbol{\theta},n} - \bar{Q}_\delta y_h(\boldsymbol{\theta}), \bar{Q}_\delta v_h \rangle_H \quad \forall v_h \in Y_h. \tag{28}$$

Both FE discretized equations result in the following algebraic system:

$$\mathbb{B}_h[\boldsymbol{\theta}]\boldsymbol{y}_h = \mathbb{F}_h, \quad \mathbb{B}_h[\boldsymbol{\theta}]^\top \boldsymbol{p}_h = 2\tau_2(\mathbb{P}_h[\boldsymbol{\theta}] - \mathbb{P}_h[\boldsymbol{y}_h]), \tag{29}$$

where $\boldsymbol{y}_h \in \mathbb{R}^{N_h}$ and $\boldsymbol{p}_h \in \mathbb{R}^{N_h}$ are the coefficients of the FE functions $y_h = \sum_{i=1}^{N_h}(\boldsymbol{y}_h)_i\phi_i$ and $p_h = \sum_{i=1}^{N_h}(\boldsymbol{p}_h)_i\phi_i$. Moreover, we have $\mathbb{B}_h[\boldsymbol{\theta}] \in \mathbb{R}^{N_h\times N_h}$, $(\mathbb{B}_h[\boldsymbol{\theta}])_{ij} := b_{\mathcal{L}}[v_{\boldsymbol{\theta},n}](\phi_i,\phi_j)$, $\mathbb{F}_h \in \mathbb{R}^{N_h}$, $(\mathbb{F}_h)_j := \langle f,\phi_j\rangle_{Y^*,Y}$, $\mathbb{P}_h[\boldsymbol{\theta}] \in \mathbb{R}^{N_h}$, $(\mathbb{P}_h[\boldsymbol{\theta}])_j := \langle \bar{Q}_\delta v_{\boldsymbol{\theta},n}, \bar{Q}_\delta\phi_j\rangle_H$.

The loss function J is usually discretized by quadrature rules. Assuming $H = L^2(\Omega)$ and $Z = L^2(\partial\Omega)$, for example, a straight forward approach results in

$$\widehat{J}^{M,h}(\boldsymbol{\theta}) := \|\mathcal{A}^\varepsilon v^M_{\boldsymbol{\theta},n} - f\|^2_{H_{M_r}} + \tau_1\|\mathcal{B}v^M_{\boldsymbol{\theta},n} - g\|^2_{Z_{M_b}} + \tau_2\|\bar{Q}_\delta v^M_{\boldsymbol{\theta},n} - \bar{Q}_\delta y_h(\boldsymbol{\theta})\|^2_{H_h},$$

where, for a sufficiently regular finite dimensional approximating function v_h, we use

$$\|v_h\|^2_{H_{M_r}} := \sum_{i=1}^{M_r} w_i^r v_h^2(x_i^r), \quad \|v_h\|^2_{Z_{M_b}} := \sum_{i=1}^{M_b} w_i^b v_h^2(x_i^b), \quad \|v_h\|^2_{H_h} := \sum_{i=1}^{N_r} w_i^h v_h^2(x_i^h),$$

with $\{x_i^r, w_i^r\}_{i=1}^{M_r}$ is a set of collocation points and weights in Ω, $\{x_i^b, w_i^b\}_{i=1}^{M_b}$ is a set of collocation points and weights on $\partial\Omega$, $\{x_i^h, w_i^h\}_{i=1}^{N_h}$ are our finite element nodes and quadrature weights. Above, $M := M_r + M_b$ is the total number of collocation points.

Algorithmically and assuming that $\nabla\widehat{J}^{M,h}(\boldsymbol{\theta})$ is a sufficiently accurate approximation of $\nabla\widehat{J}(\boldsymbol{\theta})$ (i.e. securing descent properties with respect to $\widehat{J}$ at $\boldsymbol{\theta}$) we use the discrete gradient, e.g., in the Adam optimizer [51] to minimize $\widehat{J}^{M,h}(\boldsymbol{\theta})$. Next we summarize our overall computational procedure in Algorithm 1. Upon successful termination it produces neural network parameters and other outputs by setting $\hat{it} := it + 1$. In Algorithm 1, $\text{grad}_{\boldsymbol{\theta}}$ refers to automatic differentiation with respect to the NN

Algorithm 1 Hybrid physics-informed NN training

Input: Max. number of iterations it_{max}, initial NN parameters $\boldsymbol{\theta}^{(0)}$, initial state FEM coefficients $\boldsymbol{y}_h(\boldsymbol{\theta}^{(0)})$, optimizer hyperparameters.

Output: NN parameters $\hat{\boldsymbol{\theta}} := \boldsymbol{\theta}^{(\hat{it})}$, control variable $v^M_{\hat{\boldsymbol{\theta}},n} \approx u^{\varepsilon,M}_{\boldsymbol{\theta},n}$, the state variable $\boldsymbol{y}_h(\hat{\boldsymbol{\theta}}) \in \mathbb{R}^{N_h}$.

1: **while** $0 \le it \le it_{\text{max}} - 1$ **do**
2: Solve the adjoint system $\mathbb{B}_h^{\mathbf{ad}}[\boldsymbol{\theta}^{(it)}]\boldsymbol{p}_h = 2\tau_2(\mathbb{P}_h[\boldsymbol{\theta}^{(it)}] - \mathbb{P}_h[\boldsymbol{y}_h(\boldsymbol{\theta}^{(it)})])$
3: Compute $\nabla_\theta\widehat{J}^{M,h}(y_h(\boldsymbol{\theta}^{(it)}),\boldsymbol{\theta}^{(it)}) := \text{grad}_{\boldsymbol{\theta}}(\widehat{J}^{M,h}(\boldsymbol{\theta}^{(it)}))$
4: Assemble the total gradient $\nabla\widehat{J}^{M,h}(\boldsymbol{\theta}^{(it)})$
5: Update weights $\boldsymbol{\theta}^{(it+1)} \leftarrow \text{Optimizer}\big(\nabla\widehat{J}^{M,h}(\boldsymbol{\theta}^{(it)}), \text{optimizer hyperparameters}\big)$
6: Update control variable $v_{\boldsymbol{\theta}^{(it+1)},n} \leftarrow v_{\boldsymbol{\theta}^{(it)},n}$
7: Solve the state system $\mathbb{B}_h[\boldsymbol{\theta}^{(it+1)}]\boldsymbol{y}_h(\boldsymbol{\theta}^{(it+1)}) = \mathbb{F}_h$
8: $it \leftarrow it + 1$
9: **end while**

parameters. The typical optimizer of choice also depends on hyper-parameters. For the Adam optimizer, these include selecting a learning rate $lr \in \mathbb{R}_+$ or a schedule of learning rates $lr : \mathbb{N} \to \mathbb{R}_+$ with $it \mapsto lr(it)$, as well as specifying values for $\beta_1^{Ad} \in \mathbb{R}_+$ and $\beta_2^{Ad} \in \mathbb{R}_+$ for the moving average update, and setting a batch size. We initialize $\boldsymbol{\theta}^{(0)}$ using the Glorot scheme [52], but $\boldsymbol{\theta}^{(0)}$ can also be obtained by solving a neighboring, yet simpler problem [14], [15]. For a comprehensive overview of standard NN optimization techniques, we refer exemplarily to [53] and the references therein.

4 An application in heat conduction

In this section, we focus on the example of heat conduction with respect to both, the coarse and the fine scale, respectively. For this purpose let $u_{\boldsymbol{\theta},n}$ denote the *parameterization* of $\mathcal{L}\big[u_{\boldsymbol{\theta},n}\big]$. Specifically, we study our hybrid approach in view of its embedded upscaling process.

4.1 Upscaling-based parameterization

Our exemplary stationary heat transfer problem in $\bar{\Omega} = [0,1]^2$ is defined as follows:

$$-\nabla \cdot (\boldsymbol{K}^\varepsilon \nabla w^\varepsilon) = q, \quad \text{in } \Omega, \quad \text{and} \quad w^\epsilon = 0, \quad \text{on } \partial\Omega, \tag{30}$$

where w^ε is the temperature field, $q \in L^2(\Omega)$ is the source term, $\boldsymbol{K}^\varepsilon \in C^{0,1}(\bar{\Omega}, \mathbb{R}_+)$ is a Lipschitz continuous coefficient and $\varepsilon = \min(\varepsilon_1, \dots, \varepsilon_I)$, where $\varepsilon_1, \dots, \varepsilon_I \in \mathbb{R}_+$ are small parameters indicating the lengths of small scales. In addition, it holds that

$$\exists \alpha,\ \beta \in \mathbb{R}_+ :\ \alpha \leq \boldsymbol{K}^\varepsilon(x) \leq \beta, \quad \forall x \in \bar{\Omega}, \tag{31}$$

We assume further that $\|\nabla \boldsymbol{K}^\varepsilon\|_{L^\infty(\Omega)} \leq \frac{c_{\boldsymbol{K}}}{\varepsilon}$, where $c_{\boldsymbol{K}} \in \mathbb{R}_+$ is ε-independent.

Solving (30) using finite elements can be computationally demanding due to the requirement $h \ll \varepsilon$ for a mesh size $h \in \mathbb{R}_+$ and $\varepsilon \ll 1$ in order to resolve the fine scale behaviour numerically. We aim to efficiently characterize material properties of Ω and determine a computationally feasible coarse-scale (homogenized) counterpart of (30). In this vein, the concept of G-convergence is employed to formalize the notion of a homogenized equation and the related effective material; see, e.g., [54].

Definition 4.1. *A coefficient sequence $\{\boldsymbol{K}^\varepsilon(\cdot)\}$ is said to G-converge to $\boldsymbol{K}^*(\cdot)$ as $\varepsilon \to 0$, if for any $q \in H^{-1}(\Omega)$ the sequence of solutions $\{w^\varepsilon\}$ of* (30) *satisfies*

$$w^\varepsilon \rightharpoonup w^0 \quad \textit{in} \quad H_0^1(\Omega), \quad \boldsymbol{K}^\varepsilon \nabla w^\varepsilon \rightharpoonup \boldsymbol{K}^* \nabla w^0 \quad \textit{in} \quad L^2(\Omega), \tag{32}$$

where w^0 is the solution to the homogenized equation

$$-\nabla \cdot (\boldsymbol{K}^* \nabla w^0) = q, \quad \textit{in } \Omega, \quad \textit{and} \quad w^0 = 0, \quad \textit{on } \partial\Omega.$$

Various techniques exist to find the G-limit. Their respective applicability, however, depends on the specific problem and properties of the underlying medium. If such a G-limit exists, then it does not depend on q and on the boundary data on $\partial\Omega$. We also note that the existence of $\boldsymbol{K}^*$ for general heterogeneous media still remains an open problem. Often, the representative volume element (RVE) technique can be applied to find an approximation of $\boldsymbol{K}^*$. This approach is widely utilized in engineering applications; see, e.g., [23], [24], [55], [56]. Here we state its equivalent characterization via weak convergence of gradients and fluxes in (32): For any measurable set $V \subseteq \Omega$ of positive measure $|V| > 0$ and $\langle \cdot \rangle_V = \frac{1}{|V|} \int_V \cdot \, dx$ (understood component-wise), one has

$$\lim_{\varepsilon \to 0} \langle \nabla w^\varepsilon \rangle_V = \langle \nabla w^0 \rangle_V, \quad \lim_{\varepsilon \to 0} \langle \boldsymbol{K}^\varepsilon \nabla w^\varepsilon \rangle_V = \langle \boldsymbol{K}^* \nabla w^0 \rangle_V. \tag{33}$$

The existence of the above limits for general heterogeneous media is difficult to verify. Thus, here we only assume that these limits exist; see also Assumption 2.5. Further, we introduce the following approximations:

$$\langle \nabla w^\varepsilon \rangle_V \approx \lim_{\varepsilon \to 0} \langle \nabla w^\varepsilon \rangle_V = \langle \nabla w^0 \rangle_V,\ \langle \boldsymbol{K}^\varepsilon \nabla w^\varepsilon \rangle_V \approx \lim_{\epsilon \to 0} \langle \boldsymbol{K}^\varepsilon \nabla w^\varepsilon \rangle_V \approx \widetilde{\boldsymbol{K}} \langle \nabla w^\varepsilon \rangle_V,$$

where $\widetilde{\boldsymbol{K}}$ is defined as follows.

Definition 4.2 (**Upscaled coefficient**). *The upscaled coefficient* $\widetilde{\boldsymbol{K}}$ *satisfies*

$$\langle \boldsymbol{K}^\varepsilon \nabla w^\varepsilon \rangle_V = \widetilde{\boldsymbol{K}}(x) \langle \nabla w^\varepsilon \rangle_V, \quad \text{for all } x \in V, \tag{34}$$

where $\widetilde{\boldsymbol{K}}$ *is a* 2×2 *tensor on* V *approximating* $\boldsymbol{K}^*$ *on* V.

Remark 4.1. *The upscaling process of this section admits a straighforward generalization for the matrix coefficient* $\boldsymbol{K}^\varepsilon$*; cf. [25]. In our setting,* $\boldsymbol{K}^\varepsilon$ *and* $\boldsymbol{K}^*$ *are scalar-valued coefficients, but it is beneficial to consider* $\widetilde{\boldsymbol{K}}$ *as a tensor for anisotropic materials; see Example 4.1.*

Note that $\widetilde{\boldsymbol{K}}$ is constant on V as a consequence of its definition. Further we observe that in our concrete setting, two solutions of the fine-scale problem are required to obtain $\widetilde{\boldsymbol{K}}$ from (34). In contrast to $\boldsymbol{K}^*$, boundary conditions may then affect $\widetilde{\boldsymbol{K}}$. Of course, it is very desirable to identify a set of boundary conditions such that the dependence of $\widetilde{\boldsymbol{K}}$ on them is weak. In our application context, we choose the linear temperature drop boundary conditions $w_i^\varepsilon = x_i$ on $\partial\Omega$, $i = 1, 2$, but periodic boundary conditions or temperature drop no-flow conditions can be applied as well. Their related analyses are similar; compare [25], [55], [57]. The calculation of the upscaled thermal conductivity coefficient then leads to the following fine-scale problems: For $i \in \{1, 2\}$, find $w_i^\varepsilon : \Omega \to \mathbb{R}$ with

$$-\nabla \cdot (\boldsymbol{K}^\varepsilon \nabla w_i^\varepsilon) = q, \quad \text{in } \Omega, \quad \text{and} \quad w_i^\varepsilon = x_i, \quad \text{on } \partial\Omega. \tag{35}$$

We note that $q = 0$ is typically chosen to prevent any influence of the source term on $\widetilde{\boldsymbol{K}}$. However, our multiscale solver is designed for computations of fine-scale solutions with $q \neq 0$, and setting $q = 0$ defines a special case for our NN-based homogenization scheme below. Therefore, a more general form of (35) is considered, but the influence of q on $\widetilde{\boldsymbol{K}}$ is assumed to be weak. The upscaling process is described in Algorithm 2; cf. [25, Section 4].

Algorithm 2 Upscaling algorithm

Input: $\Omega = \bigcup_{j=1}^N V_j$ with $V_i \cap V_j = \emptyset$, $i \neq j$, the solutions $\boldsymbol{w}^\varepsilon = \{w_1^\varepsilon, w_2^\varepsilon\}$ of (35).

Output: The upcaled coefficient $\widetilde{\boldsymbol{K}}$.

1: **for** $j \in \{1, \dots, N\}$ **do**

2: Compute $\widetilde{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{j}} := \langle \boldsymbol{K}^\varepsilon \nabla w_i^\varepsilon \rangle_{V_j} \in \mathbb{R}^2$, $\widetilde{\boldsymbol{T}}_{\boldsymbol{i}}^{\boldsymbol{j}} := \langle \nabla w_i^\varepsilon \rangle_{V_j} \in \mathbb{R}^2$ for $i \in \{1, 2\}$.

3: Insert $\widetilde{\boldsymbol{F}}_{\boldsymbol{i}}^{\boldsymbol{j}}$ and $\widetilde{\boldsymbol{T}}_{\boldsymbol{i}}^{\boldsymbol{j}}$ into (34) and find $\widetilde{\boldsymbol{K}}^j \in \mathbb{R}^{2\times 2}$ by solving the respective matrix equation:

$$\begin{pmatrix} \widetilde{\boldsymbol{K}}_{11}^j & \widetilde{\boldsymbol{K}}_{12}^j \\ \widetilde{\boldsymbol{K}}_{21}^j & \widetilde{\boldsymbol{K}}_{22}^j \end{pmatrix} \begin{pmatrix} (\widetilde{\boldsymbol{T}}_{\boldsymbol{1}}^{\boldsymbol{j}})_1 & (\widetilde{\boldsymbol{T}}_{\boldsymbol{2}}^{\boldsymbol{j}})_1 \\ (\widetilde{\boldsymbol{T}}_{\boldsymbol{1}}^{\boldsymbol{j}})_2 & (\widetilde{\boldsymbol{T}}_{\boldsymbol{2}}^{\boldsymbol{j}})_2 \end{pmatrix} = \begin{pmatrix} (\widetilde{\boldsymbol{F}}_{\boldsymbol{1}}^{\boldsymbol{j}})_1 & (\widetilde{\boldsymbol{F}}_{\boldsymbol{2}}^{\boldsymbol{j}})_1 \\ (\widetilde{\boldsymbol{F}}_{\boldsymbol{1}}^{\boldsymbol{j}})_2 & (\widetilde{\boldsymbol{F}}_{\boldsymbol{2}}^{\boldsymbol{j}})_2 \end{pmatrix}. \tag{36}$$

4: **end for**

5: Get $\widetilde{\boldsymbol{K}}$ with $\widetilde{\boldsymbol{K}}(x) = \widetilde{\boldsymbol{K}}^j$ for $x \in V_j$

Clearly, we need to ensure that the matrix of averaged gradients in (36) is invertible. This requires the partition $\{V_j\}$ to be sufficiently heterogeneous with each V_j of "reasonable" size to prevent linear dependence between $\widetilde{\boldsymbol{T}}_{\boldsymbol{1}}^{\boldsymbol{j}}$ and $\widetilde{\boldsymbol{T}}_{\boldsymbol{2}}^{\boldsymbol{j}}$. Algorithm 2 is based on the insertion of averaged quantities into problem-dependent constitutive relations. In our case, we apply Fourier's law of heat conduction; cf. [56], [57]. In view of other applications, let us note that Darcy's law is used for porous media flows [25], [58], [59] as the relation between fluid velocity and pressure, and Hooke's law is applied [60] in elasticity theory to relate stress and strain fields.

4.2 Analysis of the model equations

Next we show that the assumptions of Section 2 are satisfied by our fine- and coarse-scale model problems. For this purpose, let $X := C^k(\bar{\Omega}) \cap C_0(\bar{\Omega})$ for some $k \geq 2$, and $U = U_0 := H^2(\Omega) \cap H_0^1(\Omega)$ be equipped with the norms

$$\|v\|_X = \sum_{|\mathfrak{a}|\leq 2} \sup_{x\in\Omega} |\partial_x^{\mathfrak{a}} v(x)|, \quad \|v\|_U = \big(\sum_{|\mathfrak{a}|\leq 2} \|\partial_x^{\mathfrak{a}} v\|^2_{L^2(\Omega)}\big)^{\frac{1}{2}}, \tag{37}$$

respectively. Recall that X is dense in U and $\|u\|_U \leq |\Omega|^{1/2}\|u\|_X$ for $v \in X$, i.e, $X \hookrightarrow U$. Neural network maps $v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n}$ in Section 2 do not vanish on $\partial\Omega$ and $X \not\subset \mathfrak{N}_{\theta,n}$ for the chosen X; $\mathfrak{N}_{\theta,n}$ is later modified to guarantee that Assumption 2.4 holds. Let $H := L^2(\Omega)$, $Z := L^2(\partial\Omega)$, $Y := H_0^1(\Omega)$, $Y^* = H^{-1}(\Omega)$ with $Y \hookrightarrow H \hookrightarrow Y^*$, where Y is compactly embedded into H by the Rellich–Kondrachov Theorem. For $\boldsymbol{u} = (u_1, u_2)$ with $u_1, u_2 \in U$, we set $\mathcal{U} = U \times U$ with $\|\boldsymbol{u}\|_{\mathcal{U}} := (\|u_1\|_U^2 + \|u_2\|_U^2)^{\frac{1}{2}}$ and $\mathcal{X} := X \times X$ with $\|\boldsymbol{u}\|_{\mathcal{X}} := \|u_1\|_X + \|u_2\|_X$. For the discussion below, the sub-index $i \in \{1, 2\}$ is fixed for the fine-scale solution u_i^ε and the coarse-scale solution y_i.

It is convenient to study problem (35) as a problem with homogeneous Dirichlet boundary conditions. For this purpose, let $G_i := x_i$ be the extension of our boundary conditions to Ω, and set $w_i^\varepsilon := u_i^\varepsilon + G_i$, where u_i^ε satisfies

$$-\nabla \cdot (\boldsymbol{K}^\varepsilon \nabla u_i^\varepsilon) = f_i^\varepsilon, \quad \text{in } \Omega, \quad \text{and} \quad u_i^\varepsilon = 0, \quad \text{on } \partial\Omega, \tag{38}$$

with $f_i^\varepsilon := q + \partial_{x_i} \boldsymbol{K}^\varepsilon \in L^2(\Omega) = H$. By the Lax–Milgram lemma the fine-scale problem (38) has a unique solution $u_i^\varepsilon \in Y$. The following proposition shows that Assumption 2.1, Assumption 2.2 and Assumption 2.6 are satisfied, respectively.

Proposition 4.1. *Let $\mathcal{A}^\varepsilon : \tilde{U} \to H$ with $v \mapsto \mathcal{A}^\varepsilon v = -\nabla \cdot (\boldsymbol{K}^\varepsilon \nabla v)$, and $q \in H$. Then*

$$C_s(\varepsilon)\|u\|_U \leq \|\mathcal{A}^\varepsilon u\|_H \leq C_b(\varepsilon)\|u\|_U, \quad \forall u \in \tilde{U}, \tag{39}$$

where $C_s(\varepsilon) > 0$, $C_b(\varepsilon) > 0$ with $C_s(\varepsilon) \to 0$, $C_b(\varepsilon) \to \infty$ as $\varepsilon \to 0$, and $\|u_i^\varepsilon\|_H \leq C(1 + \|q\|_H)$, with $C \in \mathbb{R}_+$ ε-independent.

Proof. We multiply (38) by u_i^ε and integrate by parts to get

$$\alpha \int_\Omega |\nabla u_i^\varepsilon|^2 \, dx \leq \int_\Omega \boldsymbol{K}^\varepsilon \nabla u_i^\varepsilon \cdot \nabla u_i^\varepsilon \, dx = -\int_\Omega \boldsymbol{K}^\varepsilon \boldsymbol{e}_i \cdot \nabla u_i^\varepsilon + q u_i^\varepsilon \, dx.$$

Using the Cauchy-Schwarz and Poincaré inequalities, we get

$$\|\nabla u_i^\varepsilon\|_H \leq \frac{\beta + c_p\|q\|_H}{\alpha}, \tag{40}$$

where the Poincaré constant $c_p \in \mathbb{R}_+$ depends only on Ω, and β is according to (31). Note that problem (38) can be written as follows:

$$-\Delta u_i^\varepsilon = g_i^\varepsilon := \frac{f_i^\varepsilon + \nabla \boldsymbol{K}^\varepsilon \cdot \nabla u_i^\varepsilon}{\boldsymbol{K}^\varepsilon}, \quad \text{in } \Omega, \quad \text{and} \quad u_i^\varepsilon = 0, \quad \text{on } \partial\Omega.$$

Invoking a standard $H^2(\Omega)$ regularity result for convex domains [46], we get the existence of $\widehat{C} = \widehat{C}(\Omega) \in \mathbb{R}_+$ with

$$\|u_i^\varepsilon\|_U \leq \widehat{C}\|g_i^\varepsilon\|_H \leq \frac{\widehat{C}}{\alpha}\big(\|q\|_H + (\|\nabla u_i^\varepsilon\|_H + 1)\|\nabla \boldsymbol{K}^\varepsilon\|_{L^\infty(\Omega)}\big). \tag{41}$$

Using (40) and $\|\nabla \boldsymbol{K}^\varepsilon\|_{L^\infty(\Omega)} \leq \frac{c_{\boldsymbol{K}}}{\varepsilon}$, we obtain the estimate

$$\|u_i^\varepsilon\|_U \leq \frac{\widehat{C}}{\alpha}\big((1+\frac{\beta}{\alpha})\frac{c_{\boldsymbol{K}}}{\varepsilon} + (1+\frac{c_{\boldsymbol{K}}}{\varepsilon}\frac{c_p}{\alpha})\|q\|_H\big) \leq (1+\frac{C}{\varepsilon})\|q\|_H, \tag{42}$$

where $C \in \mathbb{R}_+$ is ε-independent. The bound (42) implies that $\mathcal{A}^\varepsilon$ has a bounded inverse $(\mathcal{A}^\varepsilon)^{-1} : H \to U$, i.e., $\|(\mathcal{A}^\varepsilon)^{-1}\| \leq (1+\frac{C}{\varepsilon}) < \infty$ for $\varepsilon > 0$, and the lower bound in (39) readily follows. Using the Young and the Cauchy–Schwarz inequalities, for $u \in U$, we get

$$\|\mathcal{A}^\varepsilon u\|_H^2 \leq 2\big(\|\nabla \boldsymbol{K}^\varepsilon \cdot \nabla u\|_H^2 + \|\boldsymbol{K}^\varepsilon \Delta u\|_H^2\big) \leq \left(\frac{C}{\varepsilon^2} + \beta^2\right)\|u\|_U^2,$$

where $C \in \mathbb{R}_+$ is ε-independent. This proves the upper bound in (39).

Applying the Poincaré inequality to (40), we get $\|u_i^\varepsilon\|_H \leq c_p\frac{\beta}{\alpha} + \frac{c_p^2}{\alpha}\|q\|_H \leq C(1+\|q\|_H)$, where $C \in \mathbb{R}_+$ is chosen sufficiently large and independently of ε. □

The next result, taken from [12, Theorem 3.3], shows that NNs with smooth activation functions are universal approximators of $C^k(\bar{\Omega})$.

Proposition 4.2. *Suppose that $\sigma \in C^\infty(\mathbb{R})$, $\sigma^{(s)}(0) \neq 0$ for $s = 0, 1, ...,$ and $\bar{\Omega} := [0,1]^d$. If $v \in C^k(\bar{\Omega})$, then there exists an architecture $\vec{\boldsymbol{n}}$ with one hidden layer and $\bar{v}_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n}$ such that*

$$\sup_{x\in\Omega} |\partial_x^{\mathfrak{a}} v(x) - \partial_x^{\mathfrak{a}} \bar{v}_{\boldsymbol{\theta},n}(x)| = \mathcal{O}\left(\frac{1}{n^{(k-|\mathfrak{a}|)/d}}\omega\big(\partial_x^{\mathfrak{b}} v, \frac{1}{n^{1/2}}\big)\right)$$

holds for all multi-indices $\mathfrak{a}, \mathfrak{b}$ with $|\mathfrak{a}| \leq k, |\mathfrak{b}| = k$, where, for $\delta_c > 0$, $\omega(v, \delta_c) = \sup\{|v(x) - v(y)| : |x-y| \leq \delta_c, x, y \in \bar{\Omega}\}$ is the modulus of continuity of v.

In order to attain homogeneous Dirichlet boundary conditions, the neural network class $\mathfrak{N}_{\theta,n}$ in Proposition 4.2 requires modification: for $\bar{v}_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n}$, we set $v_{\boldsymbol{\theta},n} := l(x)\bar{v}_{\boldsymbol{\theta},n}$, where $l \in C^\infty(\bar{\Omega})$ with $l(x) = 0$ for $x \in \partial\Omega$ and $l(x) > 0$ for $x \in \Omega$.

Remark 4.2. *Signed distance functions $l \in C^\infty(\bar{\Omega})$ are readily constructed for simple boundaries [43], [61] or can be approximated for complicated geometries using splines [62] or neural networks [63]. If $l(x)$ is (partially) unknown on $\Gamma \subseteq \partial\Omega$, using the penalization term $\|\mathcal{B}u\|_{L^2(\Gamma)}^2$ in the loss* (9) *is necessary, making implementation easily feasible, but typically increasing optimization difficulty [64].*

With a slight misuse of notation, the modified class is still denoted by $\mathfrak{N}_{\theta,n}$ with $v_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n}$. Then we have $\mathfrak{N}_{\theta,n} \subset X$ verifying Assumption 2.4. We note that the swish function $\sigma(x) = x\,\mathrm{sigmoid}(bx)$ with $b \in \mathbb{R}_+$ being a hyperparameter, satisfies the assumptions of Proposition 4.2. However, the widely-used $\tanh(x)$ does not satisfy $\sigma^{(2)}(0) \neq 0$ rendering Proposition 4.2 non-applicable. The following result [9, Theorem 5.1] is then useful.

Proposition 4.3. *Suppose that $\sigma(x) = \tanh(x)$ and $\bar{\Omega} := [0,1]^d$. If $v \in C^k(\bar{\Omega})$, then there exists an architecture $\vec{\boldsymbol{n}}$ with two hidden layers and $\bar{v}_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n}$ such that $\|v - \bar{v}_{\boldsymbol{\theta},n}\|_{W^{m,\infty}(\bar{\Omega})} = \mathcal{O}(n^{-(k-m)})$ holds[2], where $\|w\|_{W^{m,\infty}(\bar{\Omega})} = \max\limits_{|\mathfrak{a}|\leq m} \sup\limits_{x\in\Omega} |\partial_x^{\mathfrak{a}} w(x)|$.*

[2]We state the asymptotic convergence rate, but explicit constants are estimated in [9].

From the prerequisites of Proposition 4.3 we infer $X := C^k(\bar{\Omega}) \cap C_0(\bar{\Omega})$ with $k > 2$ as $m = 2$ in our example. Next, $\mathfrak{N}_{\theta,n}$ is modified as above using the transformation $l(x)$. Clearly, $\mathfrak{N}_{\theta,n} \subset X$ and $\|v\|_X \leq C\|v\|_{W^{2,\infty}(\bar{\Omega})}$ for $v \in X$ and some $C \in \mathbb{R}_+$. The latter implies $X \subset \overline{\cup_n \mathfrak{N}_{\theta,n}}$, thereby verifying Assumption 2.4 for the hyperbolic tangent activation function.

Next, we derive a simplified formula for the upscaled coefficient; cf. [25].

Proposition 4.4 (**Upscaling formula**). *Let $V = \Omega = (0,1)^2$ and $\boldsymbol{w}^\varepsilon = \{w_1^\varepsilon, w_2^\varepsilon\}$, $\boldsymbol{u}^\varepsilon = \{u_1^\varepsilon, u_2^\varepsilon\}$ be the solutions to* (35) *and* (38)*, respectively. Then the upscaled coefficient $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon] \in \mathbb{R}^{2\times 2}$ is given by*

$$\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]\boldsymbol{e}_i = \int_\Omega \boldsymbol{K}^\varepsilon \nabla w_i^\varepsilon \, dx = \int_\Omega \boldsymbol{K}^\varepsilon (\nabla u_i^\varepsilon + \boldsymbol{e}_i) \, dx, \tag{43}$$

where $\boldsymbol{e}_i \in \mathbb{R}^2$ denotes the i-th unit vector. In addition,

$$\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ij} = \langle \nabla w_i^\varepsilon \cdot \boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon \rangle_\Omega - \int_\Omega q u_i^\varepsilon \, dx. \tag{44}$$

Proof. We apply Algorithm 2 to $\widetilde{\boldsymbol{F}}_{\boldsymbol{i}} := \langle \boldsymbol{K}^\varepsilon \nabla w_i^\varepsilon \rangle_\Omega$ and $\widetilde{\boldsymbol{T}}_{\boldsymbol{i}} := \langle \nabla w_i^\varepsilon \rangle_\Omega$. Since $w_i^\varepsilon = u_i^\varepsilon + G_i$ with $\nabla G_i = \boldsymbol{e}_i$, the divergence theorem yields

$$\langle \nabla w_i^\varepsilon \rangle_\Omega = \int_{\partial\Omega} u_i^\varepsilon \, \boldsymbol{\eta} \, ds + \int_\Omega \boldsymbol{e}_i \, dx = \boldsymbol{e}_i, \tag{45}$$

where $\boldsymbol{\eta}(x)$ is the outward unit normal at $x \in \partial\Omega$. Inserting $\widetilde{\boldsymbol{F}}_{\boldsymbol{i}}$ and $\widetilde{\boldsymbol{T}}_{\boldsymbol{i}}$ into (36), we get (43). From (43), we deduce

$$\boldsymbol{e}_i \cdot \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]\boldsymbol{e}_j = \langle \nabla w_i^\varepsilon \cdot \boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon \rangle_\Omega - \langle \nabla u_i^\varepsilon \cdot \boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon \rangle_\Omega. \tag{46}$$

Integrating the last term in (46) by parts gives (44), since

$$\int_\Omega \nabla u_i^\varepsilon \cdot \boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon dx = \int_{\partial\Omega} u_i^\varepsilon (\boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon \cdot \boldsymbol{\eta}) \, ds - \int_\Omega \nabla \cdot (\boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon) u_i^\varepsilon \, dx, \tag{47}$$

where $u_i^\varepsilon|_{\partial\Omega} = 0$ and $-\nabla \cdot (\boldsymbol{K}^\varepsilon \nabla w_j^\varepsilon) = q$ in Ω. □

The weak form of our model coarse-scale problem reads: Find $y_i \in Y$ such that

$$b_{\mathcal{L}}[\boldsymbol{u}^\varepsilon](y_i, v) = \langle \widetilde{f}_i, v \rangle_{Y^*,Y} \quad \forall v \in Y, \tag{48}$$

where $\widetilde{f}_i := q + \nabla \cdot (\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon] \nabla G_i)$, $\widetilde{\boldsymbol{K}}[\boldsymbol{u}] \in \mathbb{R}^{2\times 2}$ is computed by using (43) and

$$b_{\mathcal{L}}[\boldsymbol{u}^\varepsilon](y_i, v) := \int_\Omega \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon] \nabla y_i \cdot \nabla v \, dx, \; \langle \widetilde{f}_i, v \rangle_{Y^*,Y} = \int_\Omega qv \, dx - \int_\Omega \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon] \boldsymbol{e}_i \cdot \nabla v \, dx.$$

We note that compared to Section 2, $\widetilde{f}_i \neq f_i$. Since $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]$ is a constant matrix, for $i \in \{1,2\}$ it holds that

$$\int_\Omega \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon] \boldsymbol{e}_i \cdot \nabla v \, dx = \sum_{j=1}^{2} \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ji} \int_\Omega \frac{\partial v}{\partial x_j} \, dx = 0, \tag{49}$$

where $\int_\Omega \frac{\partial v}{\partial x_j} \, dx = 0$ for $j \in \{1,2\}$ and $v \in Y$. Therefore, $y_1 = y_2$ and the sub-index i is omitted in (48).

Example 4.1. *Let* $\boldsymbol{K}_1^\varepsilon(x) = 3 + \sin(\frac{2\pi x_1}{\varepsilon}) + \sin(\frac{2\pi x_2}{\varepsilon})$ *(*$\varepsilon = 1/32$*),* $\boldsymbol{K}_2^\varepsilon(x) = 3 + \sin(\frac{2\pi x_1}{\varepsilon_1}) + 1.5\cos(\frac{2\pi x_2}{\varepsilon_2})$ *(*$\varepsilon_1 = 1/16, \varepsilon_2 = 1/32$*) and* $\boldsymbol{K}_3^\varepsilon(x) = 1/[2 + 1.8\sin(\frac{2\pi(2x_1 - x_2)}{\varepsilon})]$ *(*$\varepsilon = 1/16$*). For* $q = \exp(-\|x - 0.5\|_2^2)$*, we discretize* (38) *using finite elements with a mesh size* $h \ll \varepsilon$ *and use* (43) *to calculate the upscaled coefficients*

$$\widetilde{\boldsymbol{K}}_1 = \begin{pmatrix} 2.85 & 0.00 \\ 0.00 & 2.85 \end{pmatrix}, \ \widetilde{\boldsymbol{K}}_2 = \begin{pmatrix} 2.83 & 0.00 \\ 0.00 & 2.59 \end{pmatrix}, \ \widetilde{\boldsymbol{K}}_3 = \begin{pmatrix} 0.65 & 0.25 \\ 0.25 & 1.02 \end{pmatrix}. \tag{50}$$

The influence of q *on* $\widetilde{\boldsymbol{K}}_1, \widetilde{\boldsymbol{K}}_2, \widetilde{\boldsymbol{K}}_3$ *is rather weak due to the rotational symmetry and the moderate amplitude of* q*. In Fig. 1, we depict the FEM approximation of the respective fine-scale solution* u_1^ε*,* u_2^ε *used to calculate each of the coefficients* (50)*, as well as the FEM approximation of* $y(u_1^\varepsilon, u_2^\varepsilon)$*. Note that the coarse-scale solution is similar to its fine-scale counterparts* u_1^ε *and* u_2^ε *for all three examples, but it does not contain the fine-scale oscillations of the latter. Since* (49) *holds true, only one approximation of* (48) *is available.*

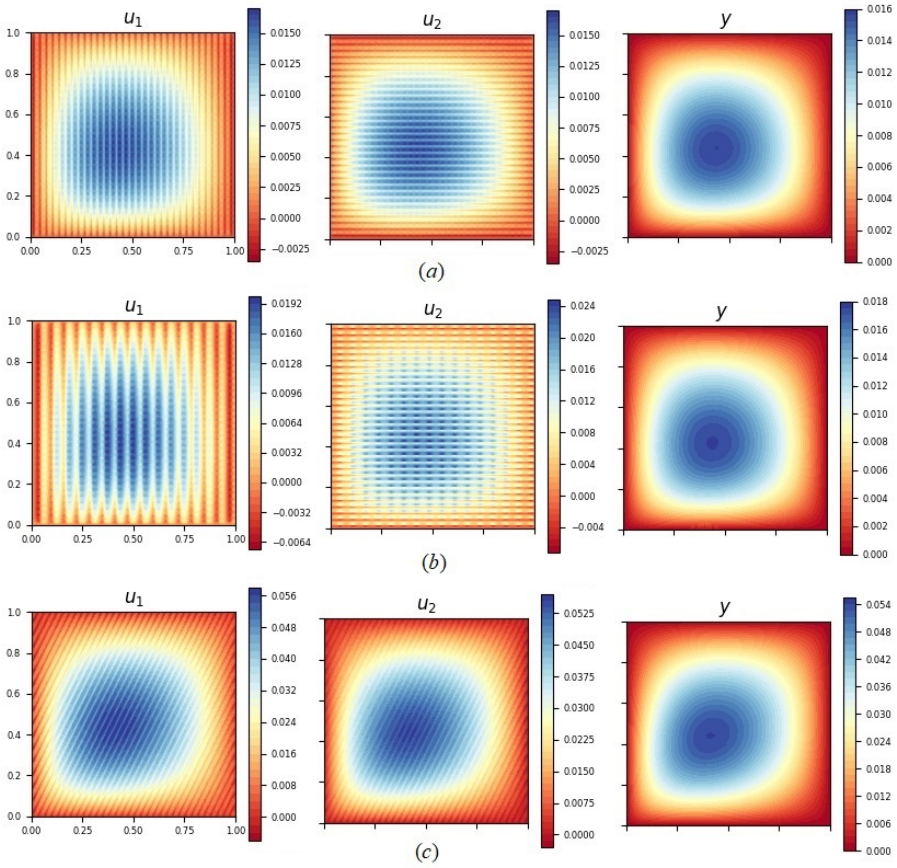

Figure 1: Two respective FEM approximations of problem (38) for the coefficients in (50) (left, middle), and one respective approximation of (48) (right): **(a)** isotropic media $\boldsymbol{K}_1^\varepsilon(x)$, **(b)** anisotropic media with principal axis anisotropy $\boldsymbol{K}_2^\varepsilon(x)$, **(c)** fully anisotropic media $\boldsymbol{K}_3^\varepsilon(x)$.

Lemma 4.1. *The mapping* $\boldsymbol{u} \in \mathcal{U} \mapsto \widetilde{\boldsymbol{K}}[\boldsymbol{u}] \in \mathbb{R}^{2\times 2}$ *defined by* (43) *is Lipschitz continuous, i.e., for* $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathcal{U}$ *we have*

$$\|\widetilde{\boldsymbol{K}}[\boldsymbol{u}_1] - \widetilde{\boldsymbol{K}}[\boldsymbol{u}_2]\| \leq \beta \|\boldsymbol{u}_1 - \boldsymbol{u}_2\|_{\mathcal{U}}, \tag{51}$$

with $\beta > 0$ *from* (31)*, and the Frobenius norm* $\|\widetilde{\boldsymbol{K}}[\boldsymbol{u}]\| := \big(\sum_{i,j=1}^2 |\widetilde{\boldsymbol{K}}[\boldsymbol{u}]_{ij}|^2\big)^{\frac{1}{2}}$ *of* $\widetilde{\boldsymbol{K}}[\boldsymbol{u}]$*.*

Proof. From (43) note that $\widetilde{\boldsymbol{K}}[\boldsymbol{u}]_{ij} = \int_\Omega \boldsymbol{K}^\varepsilon(\partial_{x_i} u_j + \delta_{ij})\,dx$, where δ_{ij} denotes the Kronecker delta. The Cauchy–Schwarz inequality yields

$$|\widetilde{\boldsymbol{K}}[\boldsymbol{u}_1]_{ij} - \widetilde{\boldsymbol{K}}[\boldsymbol{u}_2]_{ij}| \le \beta \|\partial_{x_i}(\boldsymbol{u}_1)_j - \partial_{x_i}(\boldsymbol{u}_2)_j\|_H. \tag{52}$$

After some straightforward computations, we get

$$\sum_{i,j=1}^{2} |\widetilde{\boldsymbol{K}}[\boldsymbol{u}_1]_{ij} - \widetilde{\boldsymbol{K}}[\boldsymbol{u}_2]_{ij}|^2 \le \beta^2 \big(\|\nabla(\boldsymbol{u}_1)_1 - \nabla(\boldsymbol{u}_2)_1\|_H^2 + \|\nabla(\boldsymbol{u}_1)_2 - \nabla(\boldsymbol{u}_2)_2\|_H^2\big).$$

The result then follows from the above estimate. □

Next, we establish the uniformity conditions of Assumption 2.3. Assuming momentarily that the off-diagonal values in (44) are negligible compared to the main diagonal values, and that the influence of q is rather weak, we show that the coefficients in (44) are uniformly bounded for $q = 0$. Moreover, these bounds hold even for $q \neq 0$, potentially with adjusted constants. This results then extends to the truncated approximating sequence due to the Lipschitz dependence (52) of our upscaled coefficient on $\boldsymbol{u} \in \mathcal{U}$.

Proposition 4.5. *Suppose that* $\|q\|_H \le \frac{\alpha(\nu + \|\nabla u^\varepsilon\|_H^2)}{\|u^\varepsilon\|_H}$ *for* $i \in \{1, 2\}$*, some* $0 < \nu < 1$*, and* $\alpha \in \mathbb{R}_+$ *from* (31)*. Let* $\{\boldsymbol{u}_k^\varepsilon\} \subset \mathcal{X}$ *be the approximating sequence (component-wise) of* $\boldsymbol{u}^\varepsilon \in \mathcal{U}$ *in the sense of* (3) *and* $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]$ *be given by* (44) *with* $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ij} = 0$ *for* $i \neq j$*. Then, there exists* $N_\epsilon \in \mathbb{N}$ *such that* $\tilde{C}_\alpha \le \widetilde{\boldsymbol{K}}[\boldsymbol{u}_k^\varepsilon]_{ii} \le \tilde{C}_\beta$ *for all* $k \ge N_\varepsilon$*, and* $\tilde{C}_\alpha, \tilde{C}_\beta \in \mathbb{R}_+$ *are independent of* $\boldsymbol{u} = \boldsymbol{u}_k^\varepsilon$*,* $\boldsymbol{u} = \boldsymbol{u}^\varepsilon$*.*

Proof. Set $\boldsymbol{v}_1^\varepsilon = (u_1^\varepsilon, 0)$ and $\boldsymbol{v}_2^\varepsilon = (0, u_2^\varepsilon)$. Then $\nabla \cdot \boldsymbol{v}_i^\varepsilon = \frac{\partial u_i^\varepsilon}{\partial x_i}$ and the divergence theorem yield

$$\int_\Omega \frac{\partial u_i^\varepsilon}{\partial x_i}\,dx = \int_\Omega \nabla \cdot \boldsymbol{v}_i^\varepsilon\,dx = \int_{\partial\Omega} \boldsymbol{v}_i^\varepsilon \cdot \boldsymbol{\eta}\,dx = 0, \tag{53}$$

since $u_i^\varepsilon|_{\partial\Omega} = 0$. From (44) and (53) we obtain

$$\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ii} = \langle \nabla w_i^\varepsilon \cdot \boldsymbol{K}^\varepsilon \nabla w_i^\varepsilon \rangle_\Omega - \int_\Omega q u_i^\varepsilon\,dx \ge \alpha \int_\Omega |\nabla w_i^\varepsilon|^2 dx - \int_\Omega q u_i^\varepsilon\,dx$$
$$= \alpha \int_\Omega (|\nabla u_i^\varepsilon|^2 + 2\frac{\partial u_i^\varepsilon}{\partial x_i} + 1)dx - \int_\Omega q u_i^\varepsilon\,dx \ge \alpha + \alpha\|\nabla u_i^\varepsilon\|_H^2 - \int_\Omega q u_i^\varepsilon\,dx.$$

Applying the Cauchy–Schwarz inequality and our (amplitude) assumption on q, we get

$$\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ii} \ge \alpha + \alpha\|\nabla u_i^\varepsilon\|_H^2 - \|q\|_H \|u_i^\varepsilon\|_H \ge (1 - \nu)\alpha =: C_\alpha.$$

Integration by parts, using the equation satisfied by u_i^ε, the Cauchy–Schwarz inequality, and (40) result in the estimate

$$|\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ii}| \le \beta + \beta\|\nabla u_i^\varepsilon\|_H \le \beta + \beta(\frac{\beta + c_p\|q\|_H}{\alpha}) =: C_\beta.$$

Let N_ε be the smallest integer such that $\forall k \ge N_\varepsilon$ we have $\|\boldsymbol{u}^\varepsilon - \boldsymbol{u}_k^\varepsilon\|_{\mathcal{U}} \le \frac{\epsilon_k}{\beta}$ with $0 < \epsilon_k \le \nu(1-\nu)\alpha$ and $\epsilon_k \to 0$ as $k \to \infty$. The continuity (52) for $k \ge N_\varepsilon$ implies

$$0 < (1-\nu)^2\alpha \le \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ii} - \epsilon_k \le \widetilde{\boldsymbol{K}}[\boldsymbol{u}_k^\varepsilon]_{ii} \le \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]_{ii} + \epsilon_k \le C_\beta + \nu(1-\nu)\alpha. \tag{54}$$

We set $\tilde{C}_\alpha := (1-\nu)^2\alpha$ and $\tilde{C}_\beta := C_\beta + \nu(1-\nu)\alpha$ to complete the proof. □

The Lax–Milgram lemma and Proposition 4.5 imply that for $\boldsymbol{u}^\varepsilon \in \mathcal{U}$ and the truncated sequence $\{\boldsymbol{u}^\varepsilon_{k'}\} \subset \mathcal{X}$ with $k' = k + N_\varepsilon - 1$, there exists, for every k', a unique solution $y(\boldsymbol{u}^\varepsilon_{k'}) \in Y$ of (48) with $\|y(\boldsymbol{u}^\varepsilon_{k'})\|_Y \leq C\|q\|_H$, where $C \in \mathbb{R}_+$ is independent of the fine-scale data. Let n in $\mathfrak{N}_{\theta,n} \times \mathfrak{N}_{\theta,n}$ be chosen sufficiently large such that for each $k \geq N_\varepsilon$, there exists $0 < r_k \leq \nu\alpha$, $0 < \nu < 1$, with $\|\boldsymbol{v}_{\boldsymbol{\theta},n} - \boldsymbol{u}^\varepsilon_k\|_{\mathcal{U}} \leq \nu\alpha$ for some $\boldsymbol{v}_{\boldsymbol{\theta},n} \in \mathfrak{N}_{\theta,n} \times \mathfrak{N}_{\theta,n}$. Invoking an argument identical to (54), we get a unique solution $y(\boldsymbol{v}_{\boldsymbol{\theta},n}) \in Y$. Then, Proposition 4.5 suggests that a significant deviation of $\boldsymbol{v}_{\boldsymbol{\theta},n}$ from $\boldsymbol{u}^\varepsilon$ in Algorithm 1 may lead to non-existing coarse-scale solutions due to the loss of coercivity of $b_{\mathcal{L}}[\boldsymbol{v}_{\boldsymbol{\theta},n}](\cdot,\cdot)$. The positive-definiteness of $\widetilde{\boldsymbol{K}}[\boldsymbol{v}_{\boldsymbol{\theta},n}]$ can also be checked once $\boldsymbol{v}_{\boldsymbol{\theta},n}$ is computed. In fact, we can then directly evaluate $\widetilde{\boldsymbol{K}}[\boldsymbol{v}_{\boldsymbol{\theta},n}]$ and check its positive-definiteness. This ensures the well-posedness of (48) and guarantees that Assumption 3.3 is satisfied. In case the latter does not hold true and $y(\boldsymbol{v}_{\boldsymbol{\theta},n})$ is not well-defined, then we update $\boldsymbol{\theta}$ using the standard PINN approach without coarse-scale constraints.

Proposition 4.5 motivates the following rather general data assumption.

Assumption 4.1. *For $q \in H$, $\boldsymbol{K}^\varepsilon \in C^{0,1}(\bar{\Omega})$ and the truncated approximating sequence $\{\boldsymbol{u}^\varepsilon_{k'}\}$ with $k' = k + N_\varepsilon - 1$, there exists $N_\epsilon \in \mathbb{N}$ such that $y(\boldsymbol{u}^\varepsilon_{k'}) \in Y$ and $\|y(\boldsymbol{u}^\varepsilon_{k'})\|_Y \leq C\|q\|_H$ for each $k \in \mathbb{N}$, where $C \in \mathbb{R}_+$ is independent of $\boldsymbol{u}^\varepsilon_{k'}$ and $\boldsymbol{u}^\varepsilon$.*

Note that Assumption 4.1 does not characterize q and $\boldsymbol{K}^\varepsilon$. For a precise characterization, it is necessary to establish tight bounds on $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]$ which are independent of fine scale data. Here, Proposition 4.5 only serves the purpose of illustrating the underlying principle. If $y(\boldsymbol{v}_{\boldsymbol{\theta},n})$ is well-defined, we note that the constant coefficient in (48) implies that $y(\boldsymbol{v}_{\boldsymbol{\theta},n}) \in U$ and $\|\nabla y(\boldsymbol{v}_{\boldsymbol{\theta},n})\|_{L^p(\Omega)} < \infty$ for all $p > 1$, which is relevant to Theorem 2.2 and concludes our discussion on regularity issues.

We verify Assumption 3.1 about the continuity of our fine-to-coarse scale map.

Proposition 4.6. *Suppose that Assumption 4.1 holds true. Then $S(\boldsymbol{u}^\varepsilon_{k'}) \rightharpoonup S(\boldsymbol{u}^\varepsilon)$ in Y as $k \to \infty$, where $S : \mathcal{U} \to Y$ is the fine-to-coarse scale map.*

Proof. Note that $\|y(\boldsymbol{u}^\varepsilon_{k'})\|_Y \leq C\|q\|_H$ for $\{\boldsymbol{u}^\varepsilon_{k'}\}$, $C \in \mathbb{R}_+$ and $k' \in \mathbb{N}$ as stated in Assumption 4.1. The reflexivity of Y and the Banach—Alaoglu theorem imply that $\{y(\boldsymbol{u}^\varepsilon_{k'})\}$ admits a weakly convergent subsequence (with its elements still denoted by $y(\boldsymbol{u}^\varepsilon_{k'})$). Let $\hat{y} \in Y$ denote that weak limit. Rearranging terms in (48), we obtain

$$b_{\mathcal{L}}[\boldsymbol{u}^\varepsilon](y(\boldsymbol{u}^\varepsilon), v) + \int_\Omega \big(\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon_{k'}] - \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]\big)\nabla y(\boldsymbol{u}^\varepsilon) \cdot \nabla v \, dx = \int_\Omega qv \, dx, \quad \forall v \in Y.$$

Note that $b_{\mathcal{L}}[\boldsymbol{u}^\varepsilon](y(\boldsymbol{u}^\varepsilon_{k'}), v) \to b_{\mathcal{L}}[\boldsymbol{u}^\varepsilon](\hat{y}, v)$ as $k \to \infty$ for all $v \in Y$, since $\mathcal{L}[\boldsymbol{u}^\varepsilon] \in L(Y, Y^*)$ and hence it is weakly continuous. The continuity (51) and the Cauchy–Schwartz inequality yield

$$\int_\Omega \big(\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon_{k'}] - \widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]\big)\nabla y(\boldsymbol{u}^\varepsilon_{k'}) \cdot \nabla v \, dx \leq C\|\boldsymbol{u}^\varepsilon_{k'} - \boldsymbol{u}^\varepsilon\|_{\mathcal{U}}\|q\|_H\|v\|_Y \to 0 \text{ as } k \to \infty,$$

since $C \in \mathbb{R}_+$ does not depend on $\boldsymbol{u}^\varepsilon_{k'}$ and $\boldsymbol{u}^\varepsilon$. This shows that $\hat{y} = y(\boldsymbol{u}^\varepsilon)$. □

The continuity of $S' : \mathcal{U} \to L(\mathcal{U}, Y)$ can be established through standard techniques, which we briefly outline here by examining the sensitivity equation. The sensitivity $z := S'(\boldsymbol{u})\boldsymbol{h} \in Y$ of $S(\cdot)$ at $\boldsymbol{u} \in \mathcal{U}$ in the direction $\boldsymbol{h} \in \mathcal{U}$ is given as the solution of the linearized state equation

$$\langle e_y(y(\boldsymbol{u}), \boldsymbol{u})z, v\rangle_{Y^*,Y} = -\langle e_u(y(\boldsymbol{u}), \boldsymbol{u})\boldsymbol{h}, v\rangle_{Y^*,Y} \quad \forall v \in Y, \tag{55}$$

where the partial derivatives $e_y(y,\boldsymbol{u}) : Y \to Y^*$ and $e_u(y,\boldsymbol{u}) : \mathcal{U} \to Y^*$ read

$$\langle e_y(y,\boldsymbol{u})w, v\rangle_{Y^*,Y} = \int_\Omega \widetilde{\boldsymbol{K}}[\boldsymbol{u}]\nabla w \cdot \nabla v \, dx, \ \langle e_u(y,\boldsymbol{u})\boldsymbol{h}, v\rangle_{Y^*,Y} = \int_\Omega \widetilde{\boldsymbol{K}}_u[\boldsymbol{h}]\nabla y \cdot \nabla v \, dx,$$

respectively, and the coefficient $\widetilde{\boldsymbol{K}}_u[\boldsymbol{h}] \in \mathbb{R}^{2\times 2}$ is given by $\widetilde{\boldsymbol{K}}_u[\boldsymbol{h}]_{ij} = \int_\Omega \boldsymbol{K}^\varepsilon \partial_{x_i}\boldsymbol{h}_j \, dx$. Similar to Lemma 4.1, we get $\|\widetilde{\boldsymbol{K}}_u[\boldsymbol{h}]\| \le \beta\|\boldsymbol{h}\|_{\mathcal{U}}$ for $\boldsymbol{h} \in \mathcal{U}$. For $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathcal{U}$, let $z_1 = S'(\boldsymbol{u}_1)\boldsymbol{h}$ and $z_2 = S'(\boldsymbol{u}_2)\boldsymbol{h}$. Invoking the well-posedness of (55) and (51) while estimating the right-hand side of (55) provide us with the estimate

$$\|\big(S'(\boldsymbol{u}_1) - S'(\boldsymbol{u}_2)\big)\boldsymbol{h}\|_Y \le C_{S'}\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|_{\mathcal{U}}\|\boldsymbol{h}\|_{\mathcal{U}},$$

where $C_{S'} \in \mathbb{R}_+$ generally depends on $\boldsymbol{u}$. Therefore, $S(\boldsymbol{u})$ is continuously Fréchet differentiable and Assumption 3.2 holds true. In the above argument, the invertibility of $e_y(y,\boldsymbol{u})$ in (55) is required. In our case, $e_y(y,\boldsymbol{u})$ coincides with $e(y,\boldsymbol{u})$, where the invertibility of the latter depends on the coercivity of $b_{\mathcal{L}}[\boldsymbol{u}](\cdot,\cdot)$ and is only guaranteed for $\boldsymbol{u} = \boldsymbol{u}^\varepsilon$ and $\boldsymbol{u} = \boldsymbol{u}_k^\varepsilon$ with $k \ge N_\varepsilon$; see Assumption 2.3, Proposition 4.5 and the related discussion.

4.3 Neural network-based upscaling

We present several applications of our hybrid multiscale solver to the upscaling for the heat transfer problem (35). Firstly, we adapt Algorithm 2 to our hybrid approach. We set $q = 0$ in (38) to prevent its influence on $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]$. Therefore, $f_i^\varepsilon = \partial_{x_i}\boldsymbol{K}^\varepsilon \in L^\infty(\Omega)$ with $i \in \{1,2\}$. Given $\Omega = \cup_{j=1}^N V_j$, we compute the upscaled coefficient on each grid block V_j by employing (36). Then, the right-hand side $\tilde{f}_i$ of the coarse-scale equation (48) has a non-vanishing contribution (49) and $y_1 \ne y_2$. For $\boldsymbol{y} = (y_1, y_2)$, we define the space $\mathcal{Y} := Y \times Y$, equipped with the standard product norm. The loss $J : \mathcal{Y} \times \mathcal{U} \to \mathbb{R}_{\ge 0}$ and the PDE-constraints $e_i : Y \times \mathcal{U} \to Y^*$ are defined as follows:

$$\begin{aligned} J(\boldsymbol{y},\boldsymbol{u}) &:= \sum_{i=1}^{2} \big(\|\mathcal{A}^\varepsilon u_i - f_i^\varepsilon\|_H^2 + \tau_1\|\mathcal{B}u_i\|_Z^2 + \tau_2\mathcal{R}_\delta(u_i, y_i)\big), \\ e_i(y_i,\boldsymbol{u}) &:= b_{\mathcal{L}}[\boldsymbol{u}](y_i,\cdot) - \tilde{f}_i, \quad 1 \le i \le 2. \end{aligned}$$

The learning-informed optimal control problem for numerical homogenization reads:

$$\begin{cases} \inf J(\boldsymbol{y}, \boldsymbol{v}_{\boldsymbol{\theta},n}) \text{ over } (\boldsymbol{y}, \boldsymbol{v}_{\boldsymbol{\theta},n}) \in \mathcal{Y} \times (\mathfrak{N}_{\theta,n} \times \mathfrak{N}_{\theta,n}) \cap \mathcal{U}, \\ \text{s.t. } e_i(y_i, \boldsymbol{v}_{\boldsymbol{\theta},n}) = 0, \quad 1 \le i \le 2. \end{cases} \tag{56}$$

The analysis of the fine-scale constituents of (56) and the related compression operator (10) remains largely unchanged compared to the previous sections. However, $\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon]$ is now a piecewise-constant tensor. Therefore, $\nabla y_i \notin H^1(\Omega)$ and Lemma 2.2 is not applicable. The coupling term (16) must be used to preserve the upscaling consistency as stated in Theorem 2.2. The optimization algorithm for (56) needs modification: it now requires solving two discrete state and adjoint equations, respectively, for the gradient update.

Suppose that $V = \Omega$. The coupling term is given by (11) and $\widetilde{\boldsymbol{K}}$ is computed via (43). For $q = 0$, the coarse-scale problem admits only the trivial solution $y \equiv 0$, and (56) degenerates to the standard PINN problem due to the absence of non-trivial constraints. For $q \ne 0$ we distinguish between isotropic and anisotropic materials. We note that for isotropic materials and q with a weak influence

on $\widetilde{\boldsymbol{K}}$ (moderate amplitude, rotational symmetry, etc), $(\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon])_{11} = (\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon])_{22}$ and $(\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon])_{12} = (\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon])_{21} = 0$; see also Example 4.1. Therefore, only one fine-scale problem must be solved to determine $(\widetilde{\boldsymbol{K}}[\boldsymbol{u}^\varepsilon])_{11}$ and (56) corresponds to the abstract problem (17). For anisotropic coefficients, we need to consider two fine-scale problems, leading to our sub-index notation adjustments, when compared to the abstract setting of Section 2.

4.4 Implementation issues

We provide the implementation details of Algorithm 1 for the 1-D problem

$$-\partial_x(\boldsymbol{K}^\varepsilon \partial_x u^\varepsilon) = f, \quad \text{in } \Omega := (0,1), \quad \text{and} \quad u^\varepsilon(0) = 0, \;\; u^\varepsilon(1) = 0, \tag{57}$$

where $\boldsymbol{K}^\varepsilon(x) = 1/(1.2 + \sin(\frac{2\pi x}{\varepsilon}))$, $f := q + \partial_x \boldsymbol{K}^\varepsilon$ and $q = -3(2x-1)$. The bilinear and linear forms for the coarse-scale problem are defined as in (48) with $\widetilde{\boldsymbol{K}}[u] = \int_0^1 \boldsymbol{K}^\varepsilon(\partial_x u + 1)\, dx$. The derivative $e_{\boldsymbol{\theta}}(y, \boldsymbol{\theta}) \in L(\mathbb{R}^n, Y^*)$ is given by

$$\langle e_{\boldsymbol{\theta}}(y,\boldsymbol{\theta})\boldsymbol{s}, v\rangle_{Y^*,Y} = \int_0^1 \widetilde{\boldsymbol{K}}_{\boldsymbol{\theta}}[\boldsymbol{s}]\partial_x y \cdot \partial_x v \, dx, \quad \widetilde{\boldsymbol{K}}_{\boldsymbol{\theta}}[\boldsymbol{s}] = \int_0^1 \boldsymbol{K}^\varepsilon \langle \nabla_{\boldsymbol{\theta}}(\partial_x v_{\boldsymbol{\theta},n}), \boldsymbol{s}\rangle_{\mathbb{R}^n} \, dx$$

for $\boldsymbol{s} \in \mathbb{R}^n$. Our implementation is based on the JAX framework [65] as it appears well-suited for obtaining derivatives such as $\nabla_{\boldsymbol{\theta}}(\partial_x v_{\boldsymbol{\theta},n})$. Given the k-th unit vector $\boldsymbol{e}_k \in \mathbb{R}^n$, one obtains

$$\langle y'(\boldsymbol{\theta})^* \partial_y J(y(\boldsymbol{\theta}), \boldsymbol{\theta}), \boldsymbol{e}_k\rangle_{\mathbb{R}^n} = \langle e_{\boldsymbol{\theta}}(y(\boldsymbol{\theta}), \boldsymbol{\theta})\boldsymbol{e}_k, p\rangle_{Y^*,Y}. \tag{58}$$

The formula (58) is useful for the assembly of the first summand in (25). We get $\partial_y J(y(\boldsymbol{\theta}), \boldsymbol{\theta}) = -2\tau_2(\bar{Q}_\delta v_{\boldsymbol{\theta},n} - y(\boldsymbol{\theta}))$ for the chosen coupling term (11), and the right-hand side of the adjoint system in (29) is given by $2\tau_2(\mathbb{P}_h[\boldsymbol{\theta}] - \mathbb{M}_h \boldsymbol{y}_h)$, $\mathbb{M}_h \in \mathbb{R}^{N_h \times N_h}$, $(\mathbb{M}_h)_{ij} := \langle \phi_i, \phi_j\rangle_H$.

The uniform quadrature rule is applied to discretize the PINN loss on the set of collocation points $\{x_i^r\}_{i=1}^M$ with $\frac{1}{M} \ll \varepsilon$. Piecewise-linear finite elements on the uniform partition $\{x_i^h\}_{i=1}^{N_h}$ with $\dim Y_h = N_h$ are chosen for the discretization of the coarse-scale equation. The algebraic systems are described in Section 2, but we note that $\mathbb{B}_h[\boldsymbol{\theta}] = \widetilde{\boldsymbol{K}}[v_{\boldsymbol{\theta},n}]\mathbb{A}_h$ with $(\mathbb{A}_h)_{ij} = \langle \partial_x \phi_i, \partial_x \phi_j\rangle_H$. The discrete counterpart of (58) is given by

$$\langle e_{\boldsymbol{\theta}}(y_h(\boldsymbol{\theta}), \boldsymbol{\theta})\boldsymbol{e}_k, p_h\rangle_{Y^*,Y} = \boldsymbol{y}_h^T \mathbb{E}_h[\boldsymbol{\theta}_k]\, \boldsymbol{p}_h, \quad 1 \le k \le n,$$

where $\mathbb{E}_h[\boldsymbol{\theta}_k] \in \mathbb{R}^{N_h \times N_h}$, $(\mathbb{E}_h[\boldsymbol{\theta}_k])_{ij} := \langle e_{\boldsymbol{\theta}}(\phi_i, \boldsymbol{\theta})\boldsymbol{e}_k, \phi_j\rangle_{Y^*,Y}$. We get $(\mathbb{E}_h[\boldsymbol{\theta}_k])_{ij} = \tilde{\boldsymbol{k}}_M[\boldsymbol{\theta}](\mathbb{A}_h)_{ij}$, where $\tilde{\boldsymbol{k}}_M[\boldsymbol{\theta}] \in \mathbb{R}^n$ represents approximations of $\widetilde{\boldsymbol{K}}_{\boldsymbol{\theta}}[\boldsymbol{e}_k]$ using the uniform quadrature rule

$$(\tilde{\boldsymbol{k}}_M[\boldsymbol{\theta}])_k = \frac{1}{M}\sum_{i=1}^M \boldsymbol{K}^\varepsilon(x_i^r)\frac{\partial^2 v_{\boldsymbol{\theta},n}^M(x_i^r)}{\partial \boldsymbol{\theta}_k \, \partial x}, \quad 1 \le k \le n.$$

Let $\omega_{av}(\delta) := \lfloor N_h \delta \rfloor$ with $\delta = \varepsilon$ and $\lfloor \cdot \rfloor$ returns the greatest integer less than or equal its argument, and consider the discrete sets of finite element mesh points $V_\delta^D(x_i^h) := \{x_j^h : \; |j-i| \le \omega_{av}(\delta)\}$ centered around x_i^h. We use the approximation

$$Q_\delta v_{\boldsymbol{\theta},n}^M(x_i^h) \approx \frac{1}{|V_\delta^D(x_i^h)|} \sum_{x_j^h \in V_\delta^D(x_i^h)} v_{\boldsymbol{\theta},n}^M(x_j^h) =: Q_\delta^D v_{\boldsymbol{\theta},n}^M(x_i^h), \tag{59}$$

where $|V_\delta^D(x_i^h)|$ denotes the number of mesh points in $V_\delta^D(x_i^h)$. The discretization (59) computes the moving average of $v_{\boldsymbol{\theta},n}^M$ on our FEM mesh with nodes $\{x_i^h\}_{i=1}^{N_h}$ with ω_{av} denoting the window size. Then, the discrete compression operator is given by

$$\bar{Q}_\delta^D v_{\boldsymbol{\theta},n}^M(x) = \begin{cases} Q_\delta^D v_{\boldsymbol{\theta},n}^M(x),\ x \in [\frac{\delta}{2}, 1-\frac{\delta}{2}], \\ v_{\boldsymbol{\theta},n}^M(x),\ x \in [0, \frac{\delta}{2}) \cup (1-\frac{\delta}{2}, 1]. \end{cases} \tag{60}$$

The oscillating coefficient and the spectral bias make it difficult to approximate (57) using a standard feed-forward NN. Rather we apply a multiscale Fourier feature NN [34], which uses Fourier feature mappings $\mathcal{F}^{(k)} : \mathbb{R} \to \mathbb{R}^{2m}$:

$$\mathcal{F}^{(k)}(x) = \left(\cos(2\pi \boldsymbol{B}^{(k)} x),\ \sin(2\pi \boldsymbol{B}^{(k)} x)\right), \quad 1 \le k \le K,$$

where each entry of $\boldsymbol{B}^{(k)} \in \mathbb{R}^{m \times d}$ is sampled from a Gaussian distribution $\mathcal{N}(0, \varrho_k^2)$ with $\varrho_k > 0$ a specified hyperparameter. These features are used as inputs for the hidden layers, which are defined for $1 \le k \le K$ and $2 \le l \le L-1$ as follows:

$$z_1^{(k)} = \sigma(W_1 \mathcal{F}^{(k)}(x) + b_1),\ z_l^{(k)} = \sigma(W_l z_{l-1}^{(k)} + b_l).$$

Next, we concatenate the above outputs within the linear layer as follows:

$$\bar{v}_{\boldsymbol{\theta},n}^M = W_L\left[z_L^{(1)}, ..., z_L^{(K)}\right] + b_L,$$

where W_L and b_L are the weights and biases of the output layer. The boundary conditions are imposed exactly by using $v_{\boldsymbol{\theta},n}^M = l(x)\bar{v}_{\boldsymbol{\theta},n}^M$ with $l(x) = x(1-x)$.

4.5 Numerical results

The numerical simulations are conducted for three values of ε, namely $1/16$, $1/48$, and $1/64$, respectively. The multi-scale Fourier feature network is used as the main architecture of choice, with the two Fourier features initialized by $\varrho_1 = 1$ and $\varrho_2 = 1/\varepsilon$. Table 1 provides hyperparameters, the window size $\omega_{av}(\delta)$, collocation points M, and "true" (i.e. numerically approximated) values of the upscaled coefficient $\widetilde{\boldsymbol{K}}[u_h^\varepsilon]$ obtained from finite element solutions $u_{h_u}^\varepsilon$. These solutions are computed on the uniform mesh $\{x_i^{h_u}\}_{i=1}^{N_h^u}$ with $N_h^u = 1000$ points. This mesh is also used as the validation set to track the relative discrete L^2 error during training. The parameter τ_2 is chosen without heavy fine-tuning to approximately balance $\mathcal{J}$ and $\mathcal{R}_\delta$ and to achieve good results in terms of the relative error. In the neural networks, we use the hyperbolic tangent activation function for $\varepsilon = 1/16$ and $\varepsilon = 1/48$, and the swish function with $b = 1$ for $\varepsilon = 1/64$, which outperforms $\tanh(x)$ for this value of ε in our experiments. The collocation points $\{x_i^r\}_{i=1}^M$ are equidistantly sampled on $[0, 1]$ with $1/M \ll \varepsilon$. The exponential learning rate decay schedule is used for all experiments with the Fourier feature networks, i.e., the learning rate is initialized as $5e-4$, with a decay-rate of 0.75 every 1000 training iterations. The full batch is used to train the neural networks with the Adam algorithm [51], and the respective parameters are chosen as $\beta_1^{Ad} = 0.9$ and $\beta_2^{Ad} = 0.999$. The weights and biases are initialized using the Glorot scheme [52]. The coarse-scale finite element discretization uses $N_h = 50$ degrees of freedom for $\varepsilon = 1/16$, $\varepsilon = 1/48$ and $N_h = 64$ for $\varepsilon = 1/64$ with $h = \frac{1}{N_h}$. In addition, we compute the "true" state y_h using finite elements and the same coarse mesh as in the hybrid approach, as well as the "true" values of $\widetilde{\boldsymbol{K}}[u_{h_u}^\varepsilon]$. The standard PINN is compared to the hybrid approach employing the identical neural network architecture used in the hybrid solver.

Table 1: Configuration of the simulation experiment

ε	$\omega_{av}(\delta)$	Depth $\times$ [Width]	M	τ_2	$\widetilde{\boldsymbol{K}}[u^{\varepsilon}_{h_u}]$
1/16	3	2 $\times$ [100]	280	10	0.834
1/48	1	2 $\times$ [100]	840	1000	0.842
1/64	1	2 $\times$ [150]	1000	1200	0.848

Table 2 summarizes our experimental results, including the relative (discrete) L^2 error and the required (approximate) number of iterations for each method. For $\varepsilon = 1/16$, the hybrid approach shows comparable convergence and accuracy to the Fourier feature PINN (Ms-PINN). For $\varepsilon = 1/16$, we also used a standard feed-forward NN (vanilla, V-PINN) with two hidden layers (100 neurons each) in our hybrid approach. We ran 14500 iterations of Algorithm 1 with $\tau_2 = 5 \cdot 10^5$, followed by an additional 500 iterations with $\tau_2 = 5 \cdot 10^2$, as depicted in Fig 2. For such a large value of τ_2, the optimization mostly focuses on minimizing the coupling term. On the other hand, we can see that the PINN term also plays a role, as the expected gap between $\bar{Q}_\delta u^{\varepsilon}_{\boldsymbol{\theta},n}$ and y_h is preserved (see Theorem 2.2 and Theorem 3.1). Good accuracy is achieved using our hybrid approach; however, despite multiple attempts, we were unable to attain reasonable accuracy with the standard MLP

For $\varepsilon = 1/48$, the hybrid method requires about 30% fewer iterations compared to the PINN method, while still achieving similar accuracy; see Fig 3. We obtained good accuracy for $\varepsilon = 1/64$ with the hybrid solver, while Ms-PINN still has a higher relative error despite a larger number of iterations; see Fig 4. Comparing with Fig 2, we note that the gap between $\bar{Q}_\delta u^{\varepsilon}_{\boldsymbol{\theta},n}$ and y_h decreases with ε, as Theorem 2.2 predicts. We refrain from reporting the computational time as the optimization algorithm requires further development. We observe the stabilization of $\widetilde{\boldsymbol{K}}[u^{\varepsilon}_{\boldsymbol{\theta},n}]$ around $\widetilde{\boldsymbol{K}}[u^{\varepsilon}_{h_u}] = 0.834$ after around 7500 iterations; see Fig 2(b). This suggests that once the coarse-scale component is approximated, it might be meaningful to deactivate the constraints and use $y_h(\boldsymbol{\theta})$ solely as a regularizer.

Table 2: Convergence and accuracy results

ε	Method	# iterations	Rel. L^2 error
1/16	Ms-PINN	30000	0.034
	Hybrid (Ms-PINN)	30000	0.036
	Hybrid (V-PINN)	15000	0.062
1/48	Ms-PINN	73000	0.101
	Hybrid (Ms-PINN)	53000	0.105
1/64	Ms-PINN	120000	0.124
	Hybrid (Ms-PINN)	83500	0.101

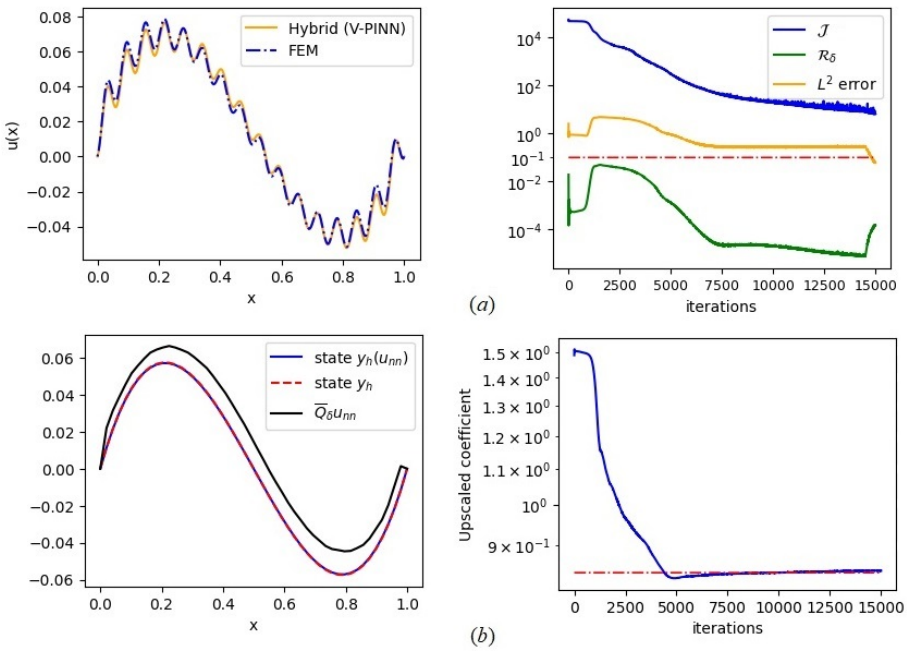

Figure 2: Results for $\varepsilon = 1/16$ (no Fourier features): **(a)** Hybrid fine-scale solution and FEM solution (left), residual loss $\mathcal{J}$, coupling loss $\mathcal{R}_\delta$, relative L^2 error vs iterations (right). **(b)** Predicted state $y_h(u_{nn})$, true state y_h, compressed NN control $\bar{Q}_\delta u_{nn}$ (left). $\widetilde{\boldsymbol{K}}[u^\varepsilon_{nn}]$ vs iterations. The red dashed line is $\widetilde{\boldsymbol{K}}[u^\varepsilon_{h_u}] = 0.834$ (right).

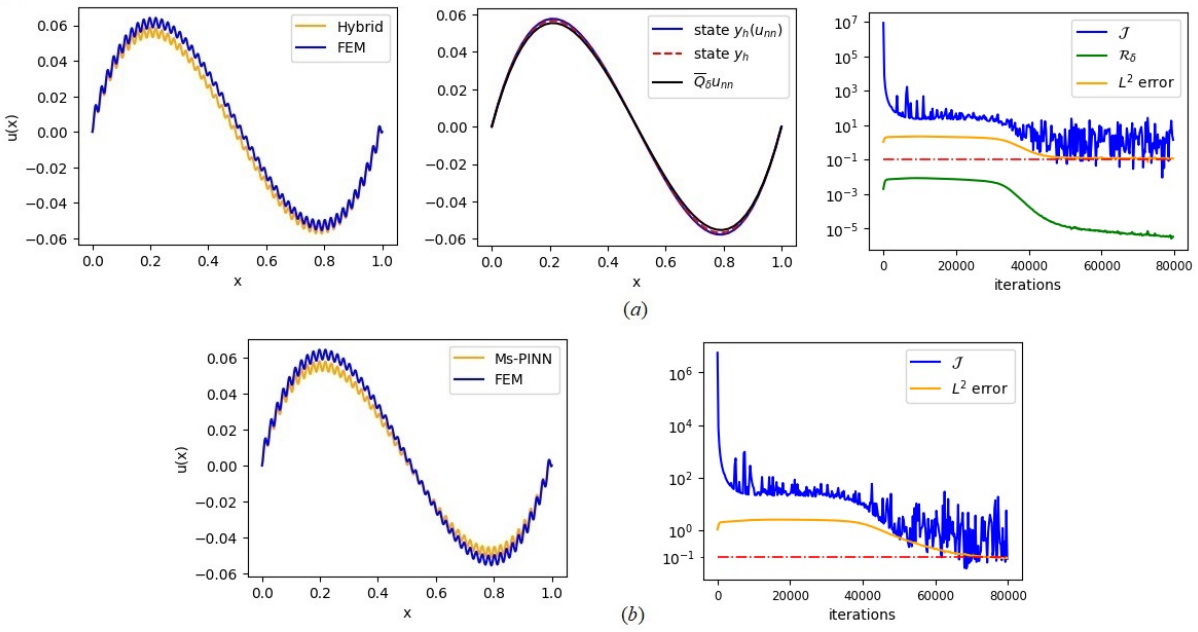

Figure 3: Results for $\varepsilon = 1/48$: **(a)** Hybrid and FEM fine-scale solution (left). Predicted state $y_h(u_{nn})$, true state y_h, compressed neural control $\bar{Q}_\delta u_{nn}$ (center). Residual loss $\mathcal{J}$, coupling loss $\mathcal{R}_\delta$, relative L^2 error vs iterations (right). **(b)** PINN and FEM fine-scale solution (left), residual loss $\mathcal{J}$ and relative L^2 error vs iterations (right).

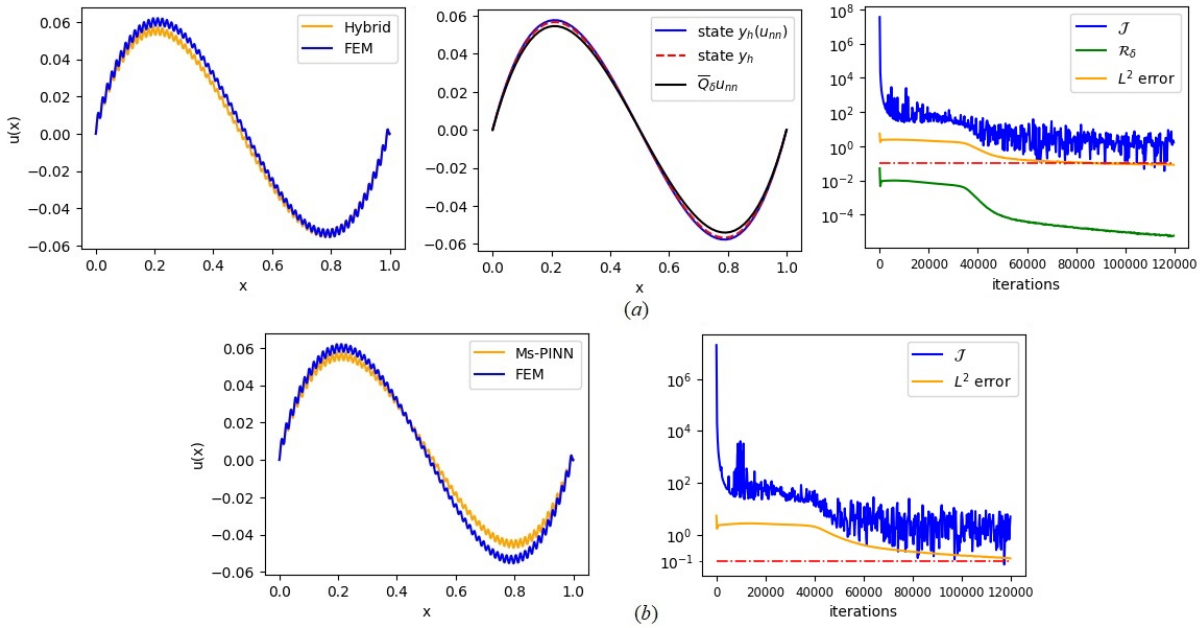

Figure 4: Results for $\varepsilon = 1/64$: **(a)** Hybrid and FEM fine-scale solution (left). Predicted state $y_h(u_{nn})$, true state y_h, compressed neural control $\bar{Q}_\delta u_{nn}$ (center). Residual loss $\mathcal{J}$, coupling loss $\mathcal{R}_\delta$, relative L^2 error vs iterations (right). **(b)** PINN and FEM fine-scale solution (left), residual loss $\mathcal{J}$ and relative L^2 error vs iterations (right).

5 Conclusion

This paper focuses on the structural properties of a learning-informed PDE-constrained optimization problem with a PINN based objective giving rise to a hybrid multiscale solver. Our approach integrates conventional multiscale methodologies and deep learning techniques, and the proposed PDE-constrained optimization setting seems particularly well-suited for this purpose. In this regard, we have introduced an abstract two-scale coupling approach and a neural network based upscaling technique. We show that incorporating coarse-scale information into the optimization has the potential to improve a fine-scale neural network approximation. In this context, selecting a suitable neural network architecture and developing an efficient optimization algorithm, aimed at enhancing accuracy while minimizing computational time, are both essential. This task requires taking into account recent advancements in PINNs and the field of PDE-constrained optimization. Our future research aims to develop hybrid coupling techniques along with efficient optimization algorithms to address more general multiscale problems, where homogenization theory results are generally unavailable, and other multiscale approximation techniques are generally inefficient.

From a technical perspective, it is worth noting that we use the standard $L^2(\Omega)$ PINN loss for the fine-scale problem and neural networks with smooth activation functions. Our example of (stationary) heat conduction demonstrates that despite its simple implementation, the natural appearance of ε-dependence in the stability constant can slow down convergence of the PINN approximation in the $H^2(\Omega)$ norm for small values of ε. Alternatively, one could develop numerical schemes using weaker residual norms (see e.g. [66]) in the objective or variational PINNs, with the intention of relaxing regularity requirements on u^ε and improving convergence rates. While such formulations are also suitable for PDEs with non-smooth data, they may also introduce non-smoothness into the re-

lated PDE-constrained optimization, e.g., due to a non-smooth activation functions. Such a setting challenges both, the derivation of suitable stationarity conditions for the PDE-constrained multiscale approach as well as its numerical solution. A comprehensive investigation of such a setting remains part of our future work in this area.

References

[1] 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", *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.

[2] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations", *IEEE transactions on neural networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[3] 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", *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[4] Z. Hu, K. Shukla, G. E. Karniadakis, and K. Kawaguchi, "Tackling the curse of dimensionality with physics-informed neural networks", *arXiv preprint arXiv:2307.12306*, 2023.

[5] Y. Zang, G. Bao, X. Ye, and H. Zhou, "Weak adversarial networks for high-dimensional partial differential equations", *Journal of Computational Physics*, vol. 411, p. 109 409, 2020.

[6] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials", *Optics express*, vol. 28, no. 8, pp. 11 618–11 633, 2020.

[7] A. D. Jagtap, Z. Mao, N. Adams, and G. E. Karniadakis, "Physics-informed neural networks for inverse problems in supersonic flows", *Journal of Computational Physics*, vol. 466, p. 111 402, 2022.

[8] S. Mishra and R. Molinaro, "Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for pdes", *IMA Journal of Numerical Analysis*, vol. 42, no. 2, pp. 981–1022, 2022.

[9] T. De Ryck, S. Lanthaler, and S. Mishra, "On the approximation of functions by tanh neural networks", *Neural Networks*, vol. 143, pp. 732–750, 2021.

[10] I. Gühring, G. Kutyniok, and P. Petersen, "Error bounds for approximations with deep relu neural networks in w s, p norms", *Analysis and Applications*, vol. 18, no. 05, pp. 803–859, 2020.

[11] H. N. Mhaskar and N. Hahm, "Neural networks for functional approximation and system identification", *Neural Computation*, vol. 9, no. 1, pp. 143–159, 1997.

[12] T. Xie and F. Cao, "The errors of simultaneous approximation of multivariate functions by neural networks", *Computers & Mathematics with Applications*, vol. 61, no. 10, pp. 3146–3152, 2011.

[13] D. Yarotsky, "Error bounds for approximations with deep relu networks", *Neural Networks*, vol. 94, pp. 103–114, 2017.

[14] S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk, "Transfer learning enhanced physics informed neural network for phase-field modeling of fracture", *Theoretical and Applied Fracture Mechanics*, vol. 106, p. 102 447, 2020.

[15] C. Xu, B. T. Cao, Y. Yuan, and G. Meschke, “Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios”, *Computer Methods in Applied Mechanics and Engineering*, vol. 405, p. 115 852, 2023.

[16] 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, pp. 2002–2041.

[17] B. Moseley, A. Markham, and T. Nissen-Meyer, “Finite basis physics-informed neural networks (fbpinns): A scalable domain decomposition approach for solving differential equations”, *Advances in Computational Mathematics*, vol. 49, no. 4, p. 62, 2023.

[18] K. Shukla, A. D. Jagtap, and G. E. Karniadakis, “Parallel physics-informed neural networks via domain decomposition”, *Journal of Computational Physics*, vol. 447, p. 110 683, 2021.

[19] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient flow pathologies in physics-informed neural networks”, *SIAM Journal on Scientific Computing*, vol. 43, no. 5, A3055–A3081, 2021.

[20] S. Wang, X. Yu, and P. Perdikaris, “When and why pinns fail to train: A neural tangent kernel perspective”, *Journal of Computational Physics*, vol. 449, p. 110 768, 2022.

[21] S. Wang, H. Wang, and P. Perdikaris, “On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks”, *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113 938, 2021.

[22] N. Rahaman, A. Baratin, D. Arpit, *et al.*, “On the spectral bias of neural networks”, in *International Conference on Machine Learning*, PMLR, 2019, pp. 5301–5310.

[23] R. M. Christensen, *Mechanics of composite materials*. Courier Corporation, 2012.

[24] C. Farmer, “Upscaling: A review”, *International journal for numerical methods in fluids*, vol. 40, no. 1-2, pp. 63–78, 2002.

[25] X.-H. Wu, Y. Efendiev, and T. Y. Hou, “Analysis of upscaling absolute permeability”, *Discrete & Continuous Dynamical Systems-B*, vol. 2, no. 2, p. 185, 2002.

[26] G. Dong, M. Hintermüller, and K. Papafitsoros, “Optimization with learning-informed differential equation constraints and its applications”, *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 28, p. 3, 2022.

[27] J. Sirignano, J. MacArt, and K. Spiliopoulos, “Pde-constrained models with neural network terms: Optimization and global convergence”, *Journal of Computational Physics*, vol. 481, p. 112 016, 2023.

[28] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, “Physics-informed neural networks with hard constraints for inverse design”, *SIAM Journal on Scientific Computing*, vol. 43, no. 6, B1105–B1132, 2021.

[29] S. Mowlavi and S. Nabi, “Optimal control of pdes using physics-informed neural networks”, *Journal of Computational Physics*, vol. 473, p. 111 731, 2023.

[30] I. Brevis, I. Muga, and K. G. van der Zee, “Neural control of discrete weak formulations: Galerkin, least squares & minimal-residual methods with quasi-optimal weights”, *Computer Methods in Applied Mechanics and Engineering*, vol. 402, p. 115 716, 2022.

[31] E. Casas, “Optimal control in coefficients of elliptic equations with state constraints”, *Applied Mathematics and Optimization*, vol. 26, no. 1, pp. 21–37, 1992.

[32] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*. Springer Science & Business Media, 2008, vol. 23.

[33] Y. Shin, Z. Zhang, and G. E. Karniadakis, "Error estimates of residual minimization using neural networks for linear pdes", *arXiv preprint arXiv:2010.08019*, 2020.

[34] M. Tancik, P. Srinivasan, B. Mildenhall, *et al.*, "Fourier features let networks learn high frequency functions in low dimensional domains", *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[35] Y. Diao, J. Yang, Y. Zhang, D. Zhang, and Y. Du, "Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology", *Computer Methods in Applied Mechanics and Engineering*, vol. 413, p. 116 120, 2023.

[36] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh, "Analyses of internal structures and defects in materials using physics-informed neural networks", *Science advances*, vol. 8, no. 7, eabk0644, 2022.

[37] W. T. Leung, G. Lin, and Z. Zhang, "Nh-pinn: Neural homogenization-based physics-informed neural network for multiscale problems", *Journal of Computational Physics*, vol. 470, p. 111 539, 2022.

[38] J. S. R. Park and X. Zhu, "Physics-informed neural networks for learning the homogenized coefficients of multiscale elliptic equations", *Journal of Computational Physics*, vol. 467, p. 111 420, 2022.

[39] R. A. Adams and J. J. Fournier, *Sobolev spaces*. Elsevier, 2003.

[40] M. Reed, B. Simon, *et al.*, *I: Functional analysis*. Gulf Professional Publishing, 1980, vol. 1.

[41] P. Bochev and M. Gunzburger, "Least-squares methods for hyperbolic problems", in *Handbook of Numerical Analysis*, vol. 17, Elsevier, 2016, pp. 289–317.

[42] P. B. Bochev and M. D. Gunzburger, *Least-squares finite element methods*. Springer Science & Business Media, 2009, vol. 166.

[43] N. Sukumar and A. Srivastava, "Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks", *Computer Methods in Applied Mechanics and Engineering*, vol. 389, p. 114 333, 2022.

[44] S. Mishra and R. Molinaro, "Estimates on the generalization error of physics-informed neural networks for approximating PDEs", *IMA Journal of Numerical Analysis*, vol. 43, no. 1, pp. 1–43, Jan. 2022.

[45] T. De Ryck, A. D. Jagtap, and S. Mishra, "Error estimates for physics informed neural networks approximating the navier-stokes equations", *IMA Journal of Numerical Analysis*, 2023.

[46] P. Grisvard, *Elliptic problems in nonsmooth domains*. SIAM, 2011.

[47] E. Kharazmi, Z. Zhang, and G. E. Karniadakis, "Hp-vpinns: Variational physics-informed neural networks with domain decomposition", *Computer Methods in Applied Mechanics and Engineering*, vol. 374, p. 113 547, 2021.

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

[49] S. Berrone, C. Canuto, and M. Pintore, "Variational physics informed neural networks: The role of quadratures and test functions", *Journal of Scientific Computing*, vol. 92, no. 3, p. 100, 2022.

[50] S. C. Brenner, *The mathematical theory of finite element methods*. Springer, 2008.

[51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.

[52] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[54] V. V. Jikov, S. M. Kozlov, and O. A. Oleinik, *Homogenization of differential operators and integral functionals*. Springer Science & Business Media, 2012.

[55] L. J. Durlofsky, “Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media”, *Water resources research*, vol. 27, no. 5, pp. 699–708, 1991.

[56] R. Ewing, O. Iliev, R. Lazarov, I. Rybak, and J. Willems, “A simplified method for upscaling composite materials with high contrast of the conductivity”, *SIAM journal on scientific computing*, vol. 31, no. 4, pp. 2568–2586, 2009.

[57] Z. K. Low, N. Blal, N. Naouar, and D. Baillis, “Influence of boundary conditions on computation of the effective thermal conductivity of foams”, *International Journal of Heat and Mass Transfer*, vol. 155, p. 119 781, 2020.

[58] M. Griebel and M. Klitz, “Homogenization and numerical simulation of flow in geometries with textile microstructures”, *Multiscale Modeling & Simulation*, vol. 8, no. 4, pp. 1439–1460, 2010.

[59] O. Iliev, Z. Lakdawala, and V. Starikovicius, “On a numerical subgrid upscaling algorithm for stokes–brinkman equations”, *Computers & Mathematics with Applications*, vol. 65, no. 3, pp. 435–448, 2013.

[60] F. Chalon, M. Mainguy, P. Longuemare, and P. Lemonnier, “Upscaling of elastic properties for large scale geomechanical simulations”, *International journal for numerical and analytical methods in geomechanics*, vol. 28, no. 11, pp. 1105–1119, 2004.

[61] P. L. Lagari, L. H. Tsoukalas, S. Safarkhani, and I. E. Lagaris, “Systematic construction of neural forms for solving partial differential equations inside rectangular domains, subject to initial, boundary and interface conditions”, *International Journal on Artificial Intelligence Tools*, vol. 29, no. 05, p. 2 050 009, 2020.

[62] H. Sheng and C. Yang, “Pfnn: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries”, *Journal of Computational Physics*, vol. 428, p. 110 085, 2021.

[63] J. Berg and K. Nyström, “A unified deep artificial neural network approach to partial differential equations in complex geometries”, *Neurocomputing*, vol. 317, pp. 28–41, 2018.

[64] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, “An expert’s guide to training physics-informed neural networks”, *arXiv preprint arXiv:2308.08468*, 2023.

[65] J. Bradbury, R. Frostig, P. Hawkins, *et al.*, “Jax: Composable transformations of python+ numpy programs (v0. 2.5)”, *Software available from https://github. com/google/jax*, 2018.

[66] J. M. Taylor, D. Pardo, and I. Muga, “A deep fourier residual method for solving pdes using neural networks”, *Computer Methods in Applied Mechanics and Engineering*, vol. 405, p. 115 850, 2023.