Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

DeepBND: A machine learning approach to enhance multiscale solid mechanics

Felipe Rocha [a,b,*], Simone Deparis [a], Pablo Antolin [a], Annalisa Buffa [a]

[a] *École polytechnique fédérale de Lausanne - SB MATH MNS MA, Station 8, CH-1015 Lausanne, Switzerland*
[b] *Nantes Université, Ecole Centrale de Nantes, CNRS, GeM (UMR 6183), 1 Rue de la Noë, 44300, Nantes, France*

ARTICLE INFO

Article history:
Received 20 October 2021
Received in revised form 26 January 2023
Accepted 1 February 2023
Available online 14 February 2023

Keywords:
Computational homogenisation
Deep Neural Networks
Reduced basis method
Boundary conditions

ABSTRACT

Effective properties of materials with random heterogeneous structures are typically determined by homogenising the mechanical quantity of interest in a window of observation. The entire problem setting encompasses the solution of a local PDE and some averaging formula for the quantity of interest in such domain. There are relatively standard methods in the literature to completely determine the formulation except for two choices: i) the local domain itself and the ii) boundary conditions. Hence, the modelling errors are governed by the quality of these two choices. The choice i) relates to the degree of representativeness of a microscale sample, i.e., it is essentially a statistical characteristic. Naturally, its reliability is higher as the size of the observation window becomes larger and/or the number of samples increases. On the other hand, excepting few special cases there is no automatic guideline to handle ii). Although it is known that the overall effect of boundary condition becomes less important with the size of the microscale domain, the computational cost to simulate such large problem several times might be prohibitive even for relatively small accuracy requirements. Here we introduce a machine learning procedure to select most suitable boundary conditions for multiscale problems, particularly those arising in solid mechanics. We propose the combination Reduced-Order Models and Deep Neural Networks in an offline phase, whilst the online phase consists in the very same homogenisation procedure plus one (cheap) evaluation of the trained model for boundary conditions. Hence, the method allows an implementation with minimal changes in existing codes and the use of relatively small domains without losing accuracy, which reduces the computational cost by several orders of magnitude. A few test cases accounting for random circular and elliptical inclusions are reported aiming at proving the potentials of the DeepBND method.

1. Introduction

Multiscale methods can be regarded as theories that link the macroscopic behaviour of continua to phenomena occurring at smaller spatial scales, a situation found in a large range of applications such as porous media or composites, to name a few. Early developments in solid mechanics date back at least from the mid-twentieth century, with the estimation of

[*] Corresponding author.
E-mail addresses: felipe.figueredo-rocha@ec-nantes.fr, felipe.f.rocha@gmail.com (F. Rocha).

https://doi.org/10.1016/j.jcp.2023.111996

macroscopic properties of heterogeneous materials, using the self-consistent approach [16,20,23], and methods based on the asymptotic analysis of partial differential equations with periodic coefficients [6,38]. Despite methodological differences, common across the range of different approaches is the fact that macroscopic continuum quantities, so-called homogenised, are invariably linked to their microscale counterpart fields upon a solution of an auxiliary local problem defined at microscale domain followed by some kind of averaging process.

As for of the local problem, also known as microscale or corrector problem, besides very specific situations, the choice of the boundary conditions (BCs) cannot be exactly determined. Depending on the BCs assumed, homogenised quantities can be under or overestimated, yielding to the so-called Reuss' and Voigt's estimates, the lower and upper bounds, respectively [37]. In problems involving strain localisation in solids, such choice can be even more critical, affecting the directions of nucleation bands [35]. The most commonly known approach to mitigate the sensibility with the BCs choice is by artificially assuming periodic BCs (even for non-periodic cells) and/or by increasing the microscopic domain size [45]. However, such an approach also entails larger computational costs in the context of numerical approximations. Recently, some improvement attempts have been proposed in this regard, as for example by adding specific parabolic and source terms to the original correct problem [2], and by a generalisation of periodic BCs to stochastic media [28].

In situations of limited physical guidance, as such case of choosing the most suitable BC, data-driven and machine learning (ML) approaches are known to be useful in providing insights. Among several ML techniques, Deep Neural Networks (DNN) is known to be a universal approximator for several activation functions [15,27], also established in the context of parametrised PDEs [21]. More specifically, we are interested in the coupling of ML techniques with a more compact representation of functions provided by Reduced-Order Modelling (ROM) techniques, such as the Reduced-Basis (RB-ROM) approach [30,18] that has been recently explored in combination with deep neural networks [17,32,39]. Indeed, we have designed a ML strategy able to predict a ROM representation of accurate boundary conditions. In our approach, RB-ROM is used to obtain an optimised representation of the boundary conditions, thus, substantially reducing the size of the network to be trained. Moreover, the use of RB-ROM automatically induces an optimal choice for the cost function for the NN training and also provides the guarantee of physically meaningful BCs predictions. The main contribution of this paper is the definition of such methodology which is then also tested on a meaningful set of test cases. In particular, to exemplify our technique, we consider microstructures with circular and ellipsoidal random inclusions, but several other examples could be easily constructed.

The manuscript is organised as follows: in Section 2 we introduce necessary concepts for the sake of completeness purposes, in particular the basic background on multiscale solid mechanics and reduced-basis for parametrised PDEs are provided in Section 2.1 and 2.2, respectively; Section 3 deals with the formulation of the proposed method, where special attention is devoted to cast the problem of Multiscale Solid Mechanics into the Parametrised PDE format in Section 3.1 that, among other steps, presents the key idea of the method of finding the reduced-order description of boundary (Section 3.1.3). The section ends with an useful overview of the method is Section 3.2. In Section 4 we propose a novel physically inspired DNN architecture tailored to exploit particular physical symmetries in the reduced-basis generation (Section 4.2). Finally, Section 5 and Section 6 are dedicated to the numerical implementation of the proposed methodology, concerning dataset construction, DNN model training, and relevant numerical examples in coupled micro-macro simulations, respectively. These last sections are primarily dedicated to the simulation case of microstructures with circular inclusions for the sake of simplicity. In Section 6.3 additional numerical examples using for elliptical inclusions at microscale as well as randomly positioned circular inclusions are reported to exemplify the generality of the methodology.

2. Preliminaries

As already commented, one of the main goals of this section is to set the general context of the computational homogenisation for random media, here specialised for multiscale solid mechanics (Section 2.1), which is instrumental to the presentation of our method (Section 3). Moreover, the second goal is to review the reduced-basis setting for parametrised PDEs (Section 2.2), which can be skipped by an expert reader.

2.1. Multiscale solid mechanics

In this section, we review the basic equations of computational homogenisation in multiscale solid mechanics. The literature in this realm is vast and dates back to the '70s with the postulation of the cornerstone Hill-Mandel Macro-homogeneity Principle [20,23]. More recently, emerging from the applied mathematical community, the so-called Heterogeneous Multiscale Method (HMM) [3] has brought significant new insights into the field. Although these two methods have independent points of departure at the first sight, it has been shown that the resulting models are coincident [9,11]. Hence our method can be applied to both methodologies. Our presentation follows closely [8,7], that provides a natural mathematical framework for the formulation of our method. Additional details that are not necessary to follow the manuscript are included in Appendix A, for the sake of completeness.

Before digging into the formulation itself, let us consider the following conventions. As usual, the unit vectors $\{\mathbf{e}_i\}_{i=1}^d$ refer to the canonical basis for $\mathbb{R}^d$. We also use the font convention $\mathbf{a} \in \mathbb{R}^d$, $\mathbf{A} \in \mathbb{R}^{d,d}$, $\mathbb{A} \in \mathbb{R}^{d,d,d,d}$ to discern between vectors, second- and fourth-order tensors, respectively. Regular type face fonts are always used to denote components (or

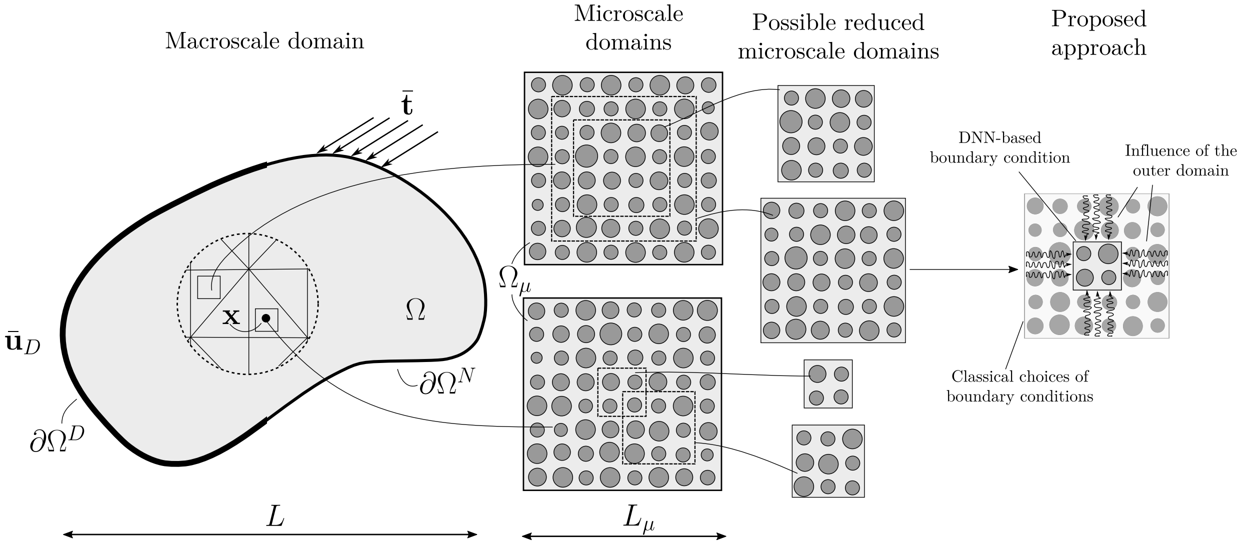


Fig. 1. Homogenisation in random media and motivation of the proposed approach.

scalars) in cartesian coordinates, e.g. $\mathbf{a} = a_i\mathbf{e}_i$, $\mathbf{A} = A_{ij}\mathbf{e}_i \otimes \mathbf{e}_j$ and so-forth (Einstein's summation convention applied). Space $\mathbb{R}^{\mathrm{d,d}}_{sym}$ denotes symmetric real-valued second-order tensors. Finally, we also consider the following operations:

$$\mathbf{a}\cdot\mathbf{b} := a_i b_i, \tag{1a}$$

$$\mathbf{A}\cdot\mathbf{B} := A_{ij}B_{ij}, \tag{1b}$$

$$\mathbb{A}\mathbf{B} := A_{ijkl}B_{kl}\mathbf{e}_i\otimes\mathbf{e}_j, \tag{1c}$$

$$\mathbf{a}\otimes^s\mathbf{b} := \frac{1}{2}(\mathbf{a}\otimes\mathbf{b}+\mathbf{b}\otimes\mathbf{a}). \tag{1d}$$

In the following, we present the multiscale solid mechanics computational homogenisation framework, starting with the macroscale followed by the microscale model.

2.1.1. Setting of the homogenisation problem

Let us consider $\Omega \subset \mathbb{R}^{\mathrm{d}}$, $\mathrm{d} = 2$ or 3, the domain in which the physical problem of interest is defined as shown in the left-hand side of Fig. 1. Thereafter, we refer to it as the *macroscale domain*, whose characteristic size is L, as opposed the to *microscale domains*, displayed on the remaining columns of Fig. 1, whose size L_μ is in the same other of the magnitude of the length-scale of material properties heterogeneities. For the sake of simplicity, the example depicted is a composite material with two phases, the matrix (in light grey) and circular inclusions (in dark grey), although more general structures for the heterogeneity are also allowed.

It is important to remark that the length of heterogeneities can be arbitrary small, posing computational difficulties if one is interested in solving such problem up to that scale. Notwithstanding, it is well established in the literature that in the scale separation limit, i.e. $\frac{L_\mu}{L} \ll 1$, the original problem with (fast-)varying parameters can be replaced by the so-called homogenised problem [6]. Indeed, the homogenised problem have the very same mathematical nature as the original at hand, but with effective, so-called homogenised, coefficients. The very process of finding such coefficients is called *homogenisation*, which encompasses the solution of an auxiliary problem, known as *corrector problem* or simply microscale problem, solved in the microscale domain, and a posterior averaging scheme using its solution, both established in Section 2.1.2 for our problem.

In the specific setting of this work, we consider as macroscale (or homogenised) problem a standard solid continuum mechanics model in the infinitesimal strain regime. We are interested in solving a quasi-static mechanical equilibrium problem, in which the displacement vector field $\mathbf{u} : \Omega \to \mathbb{R}^d$ is obtained as the solution to the equilibrium problem once suitable BCs and constitutive equations are provided. Associated with the field $\mathbf{u}$, the symmetric gradient tensor field $\boldsymbol{\varepsilon} := \nabla^s\mathbf{u} : \Omega \to \mathbb{R}^{\mathrm{d,d}}_{sym}$ is the strain measure of choice. We adopt the linear elasticity constitutive model, hence the Cauchy stress tensor $\boldsymbol{\sigma} \in \mathbb{R}^{d,d}_{sym}$ is given through the Hookean law $\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon}$, where $\mathbb{C} \in L^\infty(\Omega)^{d,d,d,d}$ is a fourth-order positive-definite elastic tensor, with symmetries $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}$. In the context of this work, we aim at obtaining $\mathbb{C}$ via the homogenisation procedure. Using the notation of the left part of Fig. 1, the macroscale problem is read as below[1]

$$\begin{cases} \operatorname{div}\boldsymbol{\sigma} = \mathbf{0} & \text{in } \Omega \\ \boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon} & \text{in } \Omega \\ \mathbf{u} = \bar{\mathbf{u}}_D & \text{on } \partial\Omega^D \\ \boldsymbol{\sigma}\mathbf{n} = \bar{\mathbf{t}} & \text{on } \partial\Omega^N \end{cases}, \tag{2}$$

where $\partial\Omega^D$ and $\partial\Omega^N$ are the Dirichlet and Neumann boundaries, respectively, forming partition of $\partial\Omega$, with the associated prescribed displacement $\bar{\mathbf{u}}_D$ and traction $\bar{\mathbf{t}}$. It is worth noticing that the computational implementation via FEM of the above problem follows standard procedures, apart from the evaluation of the $\boldsymbol{\sigma}$ (or $\mathbb{C}$) at the integration point level. For each integration point, there is an associated microstructure which should be coupled kinematically and energetically with the macroscale quantities at that point. This defines the corrector problem, which however lacks boundary condition information that needs to be chosen properly. This choice lies behind the very idea of the proposed method, to be explored in Section 3.

2.1.2. *Micro-macro coupling*

For each macroscale (integration) point $\mathbf{x} \in \Omega$ we associate a microscale domain (MD)[2] $\Omega_\mu \subset \mathbb{R}^d$. It is considered that $L_\mu/L \ll 1$, being L and L_μ the characteristic lengths related to the macroscale and microscale, respectively. The characteristic size L_μ is chosen such that Ω_μ should have meaningful numbers and type of material heterogeneities but also keeping the compromise in terms of computational cost. Thereafter, for the sake of simplicity, we use the notation $\mathbf{u} = \mathbf{u}(\mathbf{x})$, $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\mathbf{x})$ and $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{x})$, to represent point-valued vectors or tensors at a point $\mathbf{x} \in \Omega$, rather than vector or tensor fields in Ω. On the other hand, objects associated with Ω_μ are denoted by the subindex μ.

We also consider the notation $\langle\cdot\rangle_{\Omega_\mu} = \frac{1}{|\Omega_\mu|}\int_{\Omega_\mu}(\cdot)\,\mathrm{d}\Omega_\mu$, $\langle\cdot\rangle_{\partial\Omega_\mu} = \frac{1}{|\Omega_\mu|}\int_{\partial\Omega_\mu}(\cdot)\,\mathrm{d}\partial\Omega_\mu$ to denote microscale domain and boundary averages respectively. Also, we consider a local coordinate system at Ω_μ, with points denoted $\mathbf{y} \in \Omega_\mu$ and such that $\int_{\Omega_\mu}\mathbf{y}\mathrm{d}\Omega_\mu = \mathbf{0}$ (centred at the centroid). Accordingly, the symmetric gradient operator in the microscale coordinates becomes $\boldsymbol{\varepsilon}_\mu(\cdot) := \nabla^s_{\mathbf{y}}(\cdot)$.

The basic idea of the multiscale mechanics approach is to be able to describe the microscopic displacement $\mathbf{u}_\mu : \Omega_\mu \to \mathbb{R}^d$ which is related to the macroscale kinematics but not entirely determined by it. We assume that $\mathbf{u}$ and $\boldsymbol{\varepsilon}$ dictate affine deformations at the microscale while the so-called microscale displacement fluctuations field $\tilde{\mathbf{u}}_\mu : \Omega_\mu \to \mathbb{R}^d$ (supposedly $\tilde{\mathbf{u}}_\mu \in [H^1(\Omega_\mu)]^d$) is responsible for higher-order terms. This yields to the decomposition

$$\mathbf{u}_\mu(\mathbf{y}) = \mathbf{u} + \boldsymbol{\varepsilon}\mathbf{y} + \tilde{\mathbf{u}}_\mu(\mathbf{y}), \qquad \forall \mathbf{y} \in \Omega_\mu. \tag{3}$$

Following some natural physical constraints (see Appendix A for the interested reader), fluctuations $\tilde{\mathbf{u}}_\mu$ should live in

$$V_\mu^M := \left\{\boldsymbol{\eta} \in [H^1(\Omega_\mu)]^d; \langle\boldsymbol{\eta}\rangle_{\Omega_\mu} = \mathbf{0}; \left\langle\boldsymbol{\eta} \otimes^s \mathbf{n}\right\rangle_{\partial\Omega_\mu} = \mathbf{0}\right\}, \tag{4}$$

the so-called Minimally Constrained Space for fluctuations, also known as uniform traction model. Variations of this model are possible considering subspaces of V_μ^M. Remarkably, popular models (subspaces) in the literature are

$$V_\mu^T = \{\mathbf{0}\} \qquad \text{(Taylor Model)}, \tag{5a}$$

$$V_\mu^L = \{\boldsymbol{\eta} \in V_\mu^M; \boldsymbol{\eta}|_{\partial\Omega_\mu} = \mathbf{0}\} \qquad \text{(Linear Boundary Model)}, \tag{5b}$$

$$V_\mu^P = \{\boldsymbol{\eta} \in V_\mu^M; \boldsymbol{\eta}(\mathbf{y}^+) = \boldsymbol{\eta}(\mathbf{y}^-)\ \forall \mathbf{y}^\pm \in \partial\Omega_\mu^\pm\} \qquad \text{(Periodic Model)}, \tag{5c}$$

where the notation $(\cdot)^\pm$ denotes opposite boundary/points, e.g., left(-)/right(+) or bottom(-)/top(+) in a unit square Ω_μ. Fig. 2 depicts the kinematical effect of the different classical models, where V_μ^T constrain the admissible displacements to affine transformations, whilst in V_μ^L just the boundary deforms affinely and models V_μ^M and V_μ^P allow non-affine displacements on $\partial\Omega_\mu$. Excepting the space V_μ^T, the other three classical spaces will be useful for different reasons in the proposed method. It is worth noticing that $V_\mu^L \subset V_\mu^P \subset V_\mu^M$, where V_μ^L and V_μ^M are known to deliver upper and lower bounds in terms of homogenised stress respectively [45]. In turn, in the absence of physical guidance, V_μ^P is the standard choice even if the microstructure is not periodic since it delivers a more balanced response. In this work, we seek an affine subspace $V_\mu \subset V_\mu^M$ that outperforms V_μ^P, by exploiting *a priori* knowledge provided by RB-ROM and ML techniques. For convenience, thereafter we denote simply as $V_\mu \subset V_\mu^M$, the choice of one of these subspaces. The notation $V_\mu(\Omega_\mu)$ may be used to avoid ambiguity in the domain of definition.

[1] For the sake of simplicity, but without loss of generality, we are disregarding body forces (see [8]).

[2] We decided to use the acronym MD, instead of Representative Volume Element (RVE), to avoid discussions of statistical nature that go beyond the scope of this work. Other possible nomenclature, preferred by some authors to avoid this confusion, is SVE (Statistical Volume Element) [25].

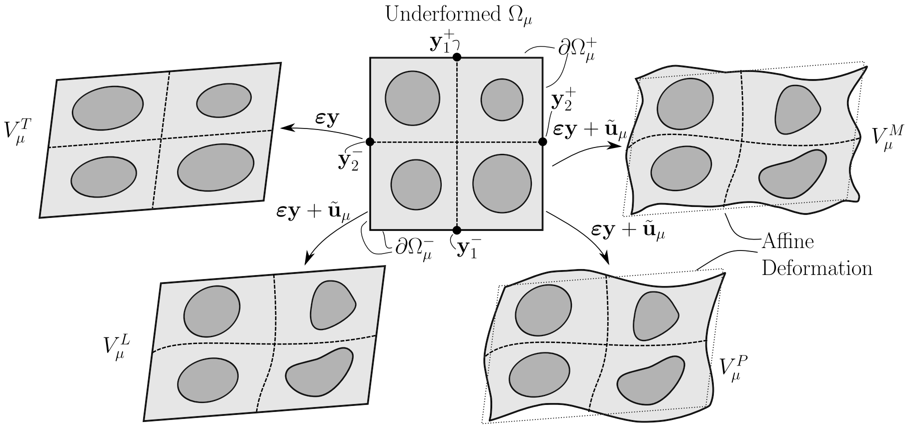


Fig. 2. Comparison of classical multiscale models of kinematical constraints.

It is also helpful for the subsequent sections to introduce an auxiliary space V_μ^{ZA} (from zero-average) as

$$V_\mu^{ZA} := \left\{ \boldsymbol{\eta} \in [H^1(\Omega_\mu)]^d; \, \langle \boldsymbol{\eta} \rangle_{\Omega_\mu} = \mathbf{0} \right\}, \tag{6}$$

which differs from V_μ^M only by relaxing the restriction on the boundary. From the definition $V_\mu^M \subset V_\mu^{ZA}$.

In terms of material constitutive behaviour at microscale, we also consider a Hookean material characterised by the fourth-order micro-elasticity tensor $\mathbb{C}_\mu \in L^\infty(\Omega_\mu)^{d,d,d,d}$ positive-definite and with the symmetries $(C_\mu)_{ijkl} = (C_\mu)_{jikl} = (C_\mu)_{ijlk} = (C_\mu)_{klij}$, which defines the microscopic stress state as $\boldsymbol{\sigma}_\mu = \mathbb{C}_\mu \boldsymbol{\varepsilon}_\mu(\mathbf{u}_\mu)$.

As result of balance of virtual powers in macro- and microscale (see details in Appendix A), the corrector (or microscale equilibrium) problem and the homogenisation read as follows.

Problem 1 *(Microscale).* Given $\boldsymbol{\varepsilon} \in \mathbb{R}^{d,d}_{sym}$, solve:

C(I) Corrector (or microscale equilibrium): find $\tilde{\mathbf{u}}_\mu \in V_\mu$ such that

$$\left\langle \mathbb{C}_\mu \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{u}}_\mu) \cdot \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{v}}_\mu) \right\rangle_{\Omega_\mu} = -\left\langle \mathbb{C}_\mu \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{v}}_\mu) \right\rangle_{\Omega_\mu} \qquad \forall \tilde{\mathbf{v}}_\mu \in V_\mu. \tag{7}$$

C(II) Stress homogenisation: let $\tilde{\mathbf{u}}_\mu \in V_\mu$ be the solution of (7), then

$$\boldsymbol{\sigma} = \left\langle \mathbb{C}_\mu(\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{u}}_\mu)) \right\rangle_{\Omega_\mu}. \tag{8}$$

Instead of computing the stress homogenisation through (8), it is convenient to obtain the homogenised elasticity tensor $\mathbb{C}$ and then evaluate $\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon}$. This can be easily performed by solving Problem 1 replacing $\boldsymbol{\varepsilon}$ by the unitary strains $\mathbf{E}_{kl} = \mathbf{e}_k \otimes^s \mathbf{e}_l$, with $k, l = 1, \ldots, d$, and then averaging as follows

$$\mathbb{C} = \left\langle \mathbb{C}_\mu(\mathbf{E}_{kl} + \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{u}}_\mu^{\mathbf{E}_{kl}})) \right\rangle_{\Omega_\mu} \otimes \mathbf{E}_{kl}, \tag{9}$$

where and $\tilde{\mathbf{u}}_\mu^{\mathbf{E}_{kl}}$ denotes the solution of (7) for a given $\boldsymbol{\varepsilon} = \mathbf{E}_{kl}$.

Remark 1. As already commented, the theory developed so far is consistent for any choice of $V_\mu \subset V_\mu^M$. Moreover, variationally, we are also allowed to choose different subsets of V_μ^M to work as trial set or test spaces. This property is explored in Section 3.1 to reduce the MD size of problem.

Remark 2. As already commented, the choice of BCs is not readily defined in the case of random media. The only case in which this choice is straightforward if for periodic cells, i.e. periodic BCs. Particularly, the material is periodic if we can identify a microscale domain which is repeated throughout (unique up to a translation). If microstructures are not periodic, the corrector problem, although valid and useful, is inexact for several reasons as discussed next.

Remark 3. It is worth noticing that Problem 1 does not depend on the macroscale displacement $\mathbf{u}$ itself, but rather on its gradient, already taken into account with $\boldsymbol{\varepsilon}$. As consequence, in terms of macroscale kinematical dependence, $\boldsymbol{\sigma}$ is invariant to $\mathbf{u}$ and only depends on $\boldsymbol{\varepsilon}$. In other words, the zero-average constraint in (4) has merely the role of eliminating rigid-body movements and can be replaced by any other equivalent strategy. We use this property in Section 3.1. The dependence with $\mathbf{u}$ is important in dynamics problems and for non-uniform body forces at the macroscale [8].

Remark 4. Instead of adopting displacement fluctuations $\tilde{\mathbf{u}}_\mu$ as primal variables, some authors prefer adopting the full microscale displacement $\mathbf{u}_\mu$ [37], which is an equivalent framework. Nevertheless, the formulation in terms of $\tilde{\mathbf{u}}_\mu$ is convenient since the kinematical constraints become homogeneous.

2.2. Reduced basis method for parametrised PDEs

In this section, we aim at providing fundamental concepts of the RB-ROM technique for parametrised PDEs. The presentation focuses on a generic problem to set a suitable notation before linking it to the specific PDE concerned, which is the focus of Section 3.1. For a deeper understanding of the method, we refer to [18,30].

Consider a vector space $\mathcal{W}$, where the solutions of a parametrised PDE live (in our case a multiscale solid mechanics boundary value problem described in Section 2.1), equipped with the inner product $(\cdot,\cdot)_{\mathcal{W}}$ with the induced norm $\|\cdot\|_{\mathcal{W}}$. Let $\mathcal{P} \subset \mathbb{R}^{N_p}$ be a manifold of admissible parameters and let $\mathcal{L}: \mathcal{P} \to \mathcal{W}$ be the mapping from parameters onto the solutions of the parametrised PDE at hand. Let $\mathbb{P} = \{\mathbf{p}^{(i)}; i = 1, \dots, N_s \in \mathcal{P}\}$ be a set containing N_s samples of $\mathcal{P}$, so-called *sampling* set, and the associated images upon $\mathcal{L}$, so-called *snapshots*, collected in the set $\mathbb{S} := \mathcal{L}(\mathbb{P}) = \{\mathbf{w}(\mathbf{p}^{(i)}) = \mathcal{L}(\mathbf{p}^{(i)}) \in \mathcal{W}; \mathbf{p}^{(i)} \in \mathbb{P}, i = 1, \dots, N_s\}$.

Note that $\operatorname{span}\mathbb{S}$ is an approximation of $\mathcal{L}(\mathcal{P})$, which becomes better accordingly to how well $\mathbb{P}$ samples $\mathcal{P}$ and depends upon the complexity of the mapping $\mathcal{L}$. We are interested in finding the orthonormal basis $\mathcal{B}_{N_{rb}} := \{\boldsymbol{\xi}_i \in \operatorname{span}\mathbb{S}\}_{i=1}^{N_{rb}}$, with $N_{rb} << N_s$, so-called Reduced Basis (RB), which is optimal in the sense that among all possible basis choices, $\mathcal{B}_{N_{rb}}$ is chosen such that $\frac{1}{N_s}\sum_{i=1}^{N_s}\|\mathbf{w}^{(i)} - \Pi_{N_{rb}}\mathbf{w}^{(i)}\|_{\mathcal{W}}^2$ is minimised, where $\Pi_{N_{rb}}(\cdot) = \sum_{i=1}^{N_{rb}}(\boldsymbol{\xi}_i, \cdot)\boldsymbol{\xi}_i$ is the orthogonal projection of a function on the space spanned by the RB.

This task can be performed by the Proper Orthogonal Decomposition (POD) procedure detailed in Algorithm 1, yielding the POD-error

$$\mathcal{E}_{POD}^2(N_{rb}) = \frac{1}{N_s}\sum_{i=1}^{N_s}\|\mathbf{w}^{(i)} - \Pi_{N_{rb}}\mathbf{w}^{(i)}\|_{\mathcal{W}}^2 = \sum_{j=N_{rb}+1}^{N_s}\lambda_j, \tag{10}$$

which is related the ordered eigenvalues λ_j, for $j = 1, \cdots, N_s$, of the correlation matrix defined in (11). The size N_{rb} is selected to attain a given tolerance $\texttt{tol}_{\texttt{POD}} \in (0,1)$ in terms of relative POD-errors.

Given $\mathbb{S}$, $\texttt{tol}_{\texttt{POD}} \in (0,1)$, do:

1. Correlation matrix:
$$\mathbf{C} = \frac{1}{N_s}\sum_{i,j=1}^{N_s}(\mathbf{w}(\mathbf{p}^{(i)}), \mathbf{w}(\mathbf{p}^{(j)}))_{\mathcal{W}}\mathbf{e}_i \otimes \mathbf{e}_j \tag{11}$$
2. Eigenvalues analysis:
$$\mathbf{C}\mathbf{v}^i = \lambda_i\mathbf{v}^i \quad \forall i = 1, 2, \dots, N_s, \tag{12}$$
where the order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_s}$ is assumed.
3. Basis determination: The i-th basis vector is constructed as
$$\boldsymbol{\xi}^i = \frac{1}{\sqrt{\lambda_i N_s}}\sum_{j=1}^{N_s}\mathbf{v}_j^i\mathbf{w}(\mathbf{p}^{(j)}) \quad \forall i = 1, 2, \dots, N_s. \tag{13}$$
4. RB construction: choose the first $N_{rb} << N_s$ such that $1 - \frac{\sum_{i=1}^{N_{rb}}\lambda_i}{\sum_{i=1}^{N_s}\lambda_i} < \texttt{tol}_{\texttt{POD}}$ and define
$$\mathcal{B}_{N_{rb}} = \left\{\boldsymbol{\xi}^i\right\}_{i=1}^{N_{rb}}. \tag{14}$$

Algorithm 1: Proper Orthogonal Decomposition procedure to obtain the Reduced Basis with a given error tolerance.

In the remaining of the manuscript, we target an RB representation of the traces of solutions on an internal boundary Γ (properly specified later). Therefore we always set $\mathcal{W} = [L^2(\Gamma)]^d$ with the $[L^2(\Gamma)]^d$-inner product as our ambient Hilbert space.

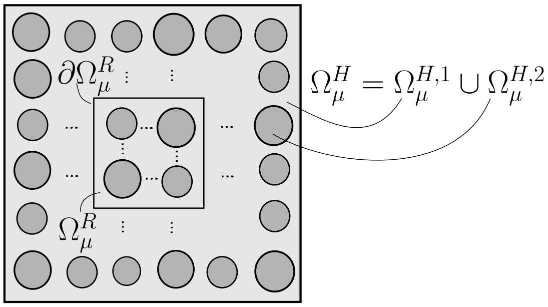


Fig. 3. Larger and smaller microscale domains and boundary where the enhanced boundary condition is imposed.

Remark 5. Note that POD procedure format in Algorithm 1 is independent of discretisation. As result, the RB can be obtained even using snapshots with heterogeneous discretisation. Just for computational purposes, e.g., speed-up calculations, we use coincident meshes on $\partial\Omega_\mu^R$ (see Section 5).

3. Formulation of the method

Before formulating the method itself, it is worth remembering that in order to have a computable microscopic cell problem, we need to truncate a problem posed essentially in an infinite domain, intrinsically connected to the choices below:

1. Boundary condition: As already commented, there is no inherently correct boundary condition to the corrector problem apart from periodic geometries. Notwithstanding, the error arising from the artificial imposition of BCs fades according to the MD size [45,22].
2. Microscale domain size: regarding the window of observation as a realisation of an underlying random variable that dictates the microstructure, the larger is such window, the better the heterogeneities are sampled. As result, as the microscale domain gets larger, the homogenised coefficients tend to the infinite cell values [26,22].

Concerning the microscale domain size decision, we have to leverage two competing characteristics, namely the computational cost and the accuracy in the two senses above. The remaining option to increase the accuracy without interfering the computational cost is improving the BCs choice. Therefore, the main contribution of this work is to bring errors due to BCs choices in small MDs down to the same levels of larger domains.

The very central rationale behind the method is the reduction of the computational domain in the local problem, as depicted on the rightmost column of Fig. 1. First, according to the final accuracy desired and the computational effort we can afford, two given sizes of microscale domains should be chosen (see Fig. 3). The larger cell is denoted Ω_μ^H and is used for the computation of reference solutions, so-called High-Fidelity (HF) solutions, by using a chosen classical BC. On the other hand, Ω_μ^R is the smaller MD (reduced) where the corrector problem is solved with an enriched, non-classical, BC. Instrumental for formulation of the admissible space that encodes such novel BC is the parametrised PDE setting, introduced in the following section. Such a structure is an essential step towards the use RB-ROM framework, which justification is also provided along this section.

3.1. Reformulation of the multiscale problem in terms of parametrised PDE

In this section, we aim at providing a parametrised PDE structure to the corrector problem for solid mechanics. It is worth mentioning that the conversion of the general problem into its parametrised format encompasses necessarily a simplification. The practical effect is a new problem with a finite number of parameters that spans a controlled scenario of randomness, representative of the problem at hand. However, the general idea of the method remains true independently of the particular parametrisation chosen.

We are interested in approximating the fluctuation $\tilde{\mathbf{u}}_\mu^H$, solution of Problem 1 in an enlarged domain Ω_μ^H, which depends on some parameters $\mathbf{p} \in \mathcal{P} \subset \mathbb{R}^{N_p}$, with N_p the number of parameters. Noticing that the triple $(\boldsymbol{\varepsilon}, \mathbb{C}_\mu, V_\mu)$ fully defines the latter problem, the role of $\mathbf{p}$ is to parametrise each of these objects, as respectively detailed in the following subsections. Importantly, we reserve the word *parameter*, and notation $\mathbf{p}$, to designate non-fixed properties, i.e., those that should be sampled. For other numbers helping to define the problem, but that are fixed, we omit their explicit dependence. Importantly, the space V_μ stands for different spaces depending if the problem is formulated on Ω_μ^H or Ω_μ^R. For Ω_μ^H, due to lack of previous information, we fix $V_\mu(\Omega_\mu^H) = V_\mu^P(\Omega_\mu^H)$, leading to Problem 1 rewritten as follows.

Problem 2 *(High-fidelity parametrised corrector PDE).* Given $\mathbf{p} \in \mathcal{P}$, find $\tilde{\mathbf{u}}_\mu^H \in V_\mu^P(\Omega_\mu^H)$ such that

$$a^H(\mathbf{p}; \tilde{\mathbf{u}}_\mu^H, \mathbf{v}) = b^H(\mathbf{p}; \mathbf{v}) \quad \forall \mathbf{v} \in V_\mu^P(\Omega_\mu^H), \tag{15}$$

where

$$a^H(\mathbf{p}; \tilde{\mathbf{u}}_\mu^H, \mathbf{v}) = (\mathbb{C}_\mu(\mathbf{p})\nabla^s \tilde{\mathbf{u}}_\mu^H, \nabla^s \mathbf{v})_{L^2(\Omega_\mu^H)}, \tag{16a}$$

$$b^H(\mathbf{p}; \mathbf{v}) = -(\mathbb{C}_\mu(\mathbf{p})\boldsymbol{\varepsilon}(\mathbf{p}), \nabla^s \mathbf{v})_{L^2(\Omega_\mu^H)}. \tag{16b}$$

To reduce the problem into Ω_μ^R, we choose $\mathbf{v} \in V_\mu^P(\Omega_\mu^H)$ such that $\mathbf{v}|_{\Omega_\mu^R} \in V_\mu^L(\Omega_\mu^R)$ and $\mathbf{v}|_{\Omega_\mu^H \setminus \Omega_\mu^R} = \mathbf{0}$, inducing the following trial set for the fluctuations

$$V_\mu(\Omega_\mu^R) = V_\mu(\mathbf{p})(\Omega_\mu^R) := \left\{ \boldsymbol{\eta} \in [H^1(\Omega_\mu^R)]^d; \langle \boldsymbol{\eta} \rangle_{\Omega_\mu^R} = \left\langle \tilde{\mathbf{u}}_\mu^H(\mathbf{p}) \right\rangle_{\Omega_\mu^R}; T_{\partial\Omega_\mu^R} \boldsymbol{\eta} = T_{\partial\Omega_\mu^R} \tilde{\mathbf{u}}_\mu^H(\mathbf{p}) \right\}, \tag{17}$$

where the dependence on $\mathbf{p}$ is made explicitly and $T_{\partial\Omega_\mu^R} : [H^1(\Omega_\mu^H)]^d \to [L^2(\partial\Omega_\mu^R)]^d$ is a trace operator. Using the previously defined spaces, the constrained version of Problem 2 on Ω_μ^R read as:

Problem 3 *(Reduced parametrised corrector PDE).* Given $\mathbf{p} \in \mathcal{P}$, find $\tilde{\mathbf{u}}_\mu^R \in V_\mu(\Omega_\mu^R)(\mathbf{p})$ such that

$$a^R(\mathbf{p}; \tilde{\mathbf{u}}_\mu^R, \mathbf{v}) = b^R(\mathbf{p}; \mathbf{v}) \quad \forall \mathbf{v} \in V_\mu^L(\Omega_\mu^R), \tag{18}$$

where

$$a^R(\mathbf{p}; \tilde{\mathbf{u}}_\mu^R, \mathbf{v}) = (\mathbb{C}_\mu(\mathbf{p})\nabla^s \tilde{\mathbf{u}}_\mu^R, \nabla^s \mathbf{v})_{L^2(\Omega_\mu^R)}, \tag{19a}$$

$$b^R(\mathbf{p}; \mathbf{v}) = -(\mathbb{C}_\mu(\mathbf{p})\boldsymbol{\varepsilon}(\mathbf{p}), \nabla^s \mathbf{v})_{L^2(\Omega_\mu^R)}. \tag{19b}$$

Importantly, we should notice that $\tilde{\mathbf{u}}_\mu^H|_{\Omega_\mu^R} = \tilde{\mathbf{u}}_\mu^R$ holds by construction and, as already commented in Remark 3, the homogenised stress is invariant upon translations. Therefore, it is convenient to define

$$V_\mu^*(\Omega_\mu^R) = V_\mu^*(\mathbf{p})(\Omega_\mu^R) := \left\{ \boldsymbol{\eta} \in V_\mu^{ZA}(\Omega_\mu^R); T_{\partial\Omega_\mu^R} \boldsymbol{\eta} = \mathbf{w}(\mathbf{p}) \right\}, \tag{20}$$

where $\mathbf{w}(\mathbf{p}) \in L^2(\partial\Omega_\mu^R)$ denotes

$$\mathbf{w}(\mathbf{p}) = T_{\partial\Omega_\mu^R} \tilde{\mathbf{u}}_\mu^H(\mathbf{p}) - \left\langle \tilde{\mathbf{u}}_\mu^H(\mathbf{p}) \right\rangle_{\Omega_\mu^R}. \tag{21}$$

Notice that Problem 3 can be rephrased replacing $V_\mu(\mathbf{p})(\Omega_\mu^R)$ by $V_\mu^*(\mathbf{p})(\Omega_\mu^R)$. As commented, the new solution is mechanically equivalent (leads to the same homogenised stress). As a result, the only information needed coming from the solution of Problem 2 is its trace to the internal boundary $\partial\Omega_\mu^R$ and the necessary translation to obtain a zero-averaged fluctuation field in Ω_μ^R. For convenience, we rearranged this operation in the goal solution $\mathbf{w}$, as above.

The key idea of proposed method is the following: Instead of solving the larger Problem 2 to obtain $\mathbf{w}$, we aim at obtaining an approximation $\mathbf{w}^{\mathcal{N}}$ by a method that combines RB-ROM and DNN. This approximation yields to the definition of the novel set $V_\mu^{\mathcal{N}}(\Omega_\mu^R)(\mathbf{p})$ (or just $V_\mu^{\mathcal{N}}$). Our final goal is to show that $V_\mu^{\mathcal{N}}$ outperforms the classical choices of BCs.

Finally, before proceeding with the parametrisation of $\mathbb{C}_\mu$ and $\boldsymbol{\varepsilon}$ in Sections 3.1.1 and 3.1.2, respectively, it is useful to assume the split $\mathbf{p} = (\mathbf{p}_c, \mathbf{p}_{\boldsymbol{\varepsilon}})$, where $\mathbb{C}_\mu(\mathbf{p}) = \mathbb{C}_\mu(\mathbf{p}_c)$ and $\boldsymbol{\varepsilon}(\mathbf{p}) = \boldsymbol{\varepsilon}(\mathbf{p}_{\boldsymbol{\varepsilon}})$. The explicitly definition of $V_\mu^{\mathcal{N}}(\Omega_\mu^R)(\mathbf{p})$ is postponed to Section 3.1.3

3.1.1. Parametrisation of $\mathbb{C}_\mu$

In our context, parametrising the constitutive law is intrinsically linked with the parametrisation of microstructural geometrical features of material constituents. In this work, since the number of parameters have a direct impact on the quality of sampling, if we want to keep the number of samples limited, we aim at keeping the geometry and number of materials simple.

We admit a partition of two materials on Ω_μ^H where those properties are constant by parts. Let be $\Omega_\mu^{H,1}$, $\Omega_\mu^{H,2}$ be such that $\overline{\Omega}_\mu^{H,1} \cup \overline{\Omega}_\mu^{H,2} = \overline{\Omega}_\mu^H$ and $\Omega_\mu^{H,1} \cap \Omega_\mu^{H,2} = \emptyset$, i.e., a partition of Ω_μ^H (see Fig. 3). Each of these materials are linear isotropic and thus modelled by the Hookean law characterised by the two spatially varying Lamé coefficients λ_μ and G_μ, or conversely the pair Young modulus and Poisson ratio. It reads

$$\mathbb{C}_\mu(\mathbf{y}) = \lambda_\mu(\mathbf{y})\mathbf{I} \otimes \mathbf{I} + 2G_\mu(\mathbf{y})\mathbb{I}^s \quad , \forall \mathbf{y} \in \Omega_\mu^H. \tag{22}$$

Let us assume λ_μ and G_μ are linked with $\Omega_\mu^{H,1}$ and $\Omega_\mu^{H,2}$ via a constant $\gamma > 0$. It is useful to define $\chi_\gamma : \Omega_\mu^H \to \mathbb{R}$ as follows

$$\chi_\gamma(\mathbf{y}) = \begin{cases} \gamma, & \text{if } \mathbf{y} \in \Omega_\mu^{H,2} \\ 1, & \text{otherwise} \end{cases}, \tag{23}$$

so $\lambda_\mu(\mathbf{y}) := \chi_\gamma(\mathbf{y})\lambda_\mu^1$, $G_\mu(\mathbf{y}) = \chi_\gamma(\mathbf{y})G_\mu^1$, i.e., $\mathbb{C}_\mu(\mathbf{y}) = \chi_\gamma(\mathbf{y})\mathbb{C}_\mu^1$.

To exemplify the parametrisation, for the sake of simplicity, let us consider an elastic matrix containing elastic circular inclusions in the two-dimensional setting. Let $\Omega_\mu^H = (0,1)^2$ and $\Omega_\mu^{H,2}$ composed by union of disjoint balls, distributed on a lattice. Let N_b the number of balls, $\mathbf{y}_i^c$ and r_i, $i = 1, 2, \ldots, N_b$ the centre positions and radii, respectively. Hence $\Omega_\mu^{H,2} = \bigcup_{i=1}^{N_b} B(\mathbf{y}_i^c, r_i)$, where $B(\mathbf{y}_0, r) := \{\mathbf{y} \in \mathbb{R}^d;\ \|\mathbf{y} - \mathbf{y}_0\| < r\}$ denotes the ball centred in $\mathbf{y}_0$ with radius r. Furthermore, we assume a regular grid of $\sqrt{N_b} \times \sqrt{N_b}$ balls, for N_b a perfect square number, with positions as follows

$$\mathbf{y}^c_{(i-1)\sqrt{N_b}+j+1} = \frac{2}{\sqrt{N_b}}(2j-1, 2i-1), \quad i, j = 1, 2, \ldots, \sqrt{N_b}. \tag{24}$$

Regarding other properties used in the model, for the sake of simplicity, let us take λ_μ^1, G_μ^1, and γ fixed. Notice that each element of $(\lambda_\mu^1, G_\mu^1, \gamma, \mathbf{y}_1^c, \ldots, \mathbf{y}_1^{N_b})$ has fixed value and is not considered as a parameter of the problem. In turn, the *de facto* parameters ruling $\mathbb{C}_\mu$ are $\mathbf{p}_c = (r_1, r_2, \ldots, r_{N_b})$.

3.1.2. Parametrisation of $\boldsymbol{\varepsilon}$

Differently as in the last section, the parametrisation of macroscale strain is straightforward and we do not need to assume any simplification, besides the symmetry of $\boldsymbol{\varepsilon}$. Focusing on $d = 2$, we introduce $\mathbf{p}_{\boldsymbol{\varepsilon}} = (p_{\boldsymbol{\varepsilon},1}, p_{\boldsymbol{\varepsilon},2}, p_{\boldsymbol{\varepsilon},3})$ and the mapping $\mathbf{p}_{\boldsymbol{\varepsilon}} \mapsto \boldsymbol{\varepsilon}(\mathbf{p}_{\boldsymbol{\varepsilon}}) = \begin{bmatrix} p_{\boldsymbol{\varepsilon},1} & \frac{1}{2}p_{\boldsymbol{\varepsilon},3} \\ \frac{1}{2}p_{\boldsymbol{\varepsilon},3} & p_{\boldsymbol{\varepsilon},2} \end{bmatrix}$. This choice of parametrisation is not unique, but it turns out that in this specific choice each component of $\mathbf{p}_{\boldsymbol{\varepsilon}}$ is physically meaningful: i) $p_{\boldsymbol{\varepsilon},1,2}$ is the axial strain along the $1,2$ directions and ii) $p_{\boldsymbol{\varepsilon},3}$ is the shear strain. Indeed, $\mathbf{p}_{\boldsymbol{\varepsilon}}$ corresponds to the so-called Voigt's notation of $\boldsymbol{\varepsilon}$, here represented by the mapping $\text{Voigt} : \mathbb{R}^{2,2}_{\text{sym}} \to \mathbb{R}^3$, such that $\boldsymbol{\varepsilon} \mapsto \mathbf{p}_{\boldsymbol{\varepsilon}} = \text{Voigt}(\boldsymbol{\varepsilon}) = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12} + \varepsilon_{21})$. Finally, the extension for $d = 3$ is immediate, but will not be used in this manuscript.

Taking advantage of the problem structure in the right-hand side of (15), we can exploit its linearity with respect to $\mathbf{p}_{\boldsymbol{\varepsilon}}$ by writing

$$b^H(\mathbf{p}; \mathbf{v}) = \sum_{i=1}^{3} p_{\boldsymbol{\varepsilon},i} b^H((\mathbf{p}_c, \mathbf{e}_i); \mathbf{v}). \tag{25}$$

Accordingly, it is easy to see that linearity is also valid for $\mathbf{w}$ as follows

$$\mathbf{w}(\mathbf{p}) = \sum_{i=1}^{3} p_{\boldsymbol{\varepsilon},i} \mathbf{w}^{(i)}(\mathbf{p}_c), \tag{26}$$

where $\mathbf{w}^{(i)}(\mathbf{p}_c) := T_{\partial\Omega_\mu} \tilde{\mathbf{u}}_\mu^H((\mathbf{p}_c, \mathbf{e}_i))$, for $i = 1, 2, 3$. We exploit this decomposition in Section 4.2 to reduce the number of RB constructions.

3.1.3. Parametrisation of $V_\mu^{\mathcal{N}}$

By defining a learning-based set $V_\mu^{\mathcal{N}}$, the goal is to characterise the set where fluctuations live as accurate as possible (not necessarily as large as possible) based on the prior knowledge decoded in the RB-ROM and DNN model. Indeed, we must recall that $V_\mu^{\mathcal{N}}$ aims to approximate $V_\mu(\Omega_\mu^R)$ in Problem 3. To this end, let us assume we dispose of an N_{rb}-dimensional RB $\mathcal{B}_{N_{rb}} = \{\boldsymbol{\xi}_i;\ \boldsymbol{\xi}_i \in L^2(\partial\Omega_\mu^R), i = 1, \cdots, N_{rb}\}$ and the mapping $\mathcal{N} : \mathbb{R}^{N_p} \to \mathbb{R}^{N_{rb}}$. We postulate

$$V_\mu^{\mathcal{N}}(\mathbf{p}) = \left\{ \boldsymbol{\eta} \in V_\mu^{ZA};\ T_{\partial\Omega_\mu^R} \boldsymbol{\eta} = \sum_{i=1}^{N_{rb}} [\mathcal{N}(\mathbf{p})]_i \boldsymbol{\xi}_i \right\} = V_\mu^L + \tilde{\mathbf{u}}_\mu^{\mathcal{N}}(\mathbf{p}), \tag{27}$$

where $\tilde{\mathbf{u}}_\mu^{\mathcal{N}}(\mathbf{p}) \in V_\mu^{ZA}$ is a continuous extension of $\sum_{i=1}^{N_{rb}} [\mathcal{N}(\mathbf{p})]_i \boldsymbol{\xi}_i \in L^2(\partial\Omega_\mu^R)$ to Ω_μ^R, whose construction is detailed next.

Solving Problem 3 with solutions in $V_\mu(\mathbf{p})$ involves finding $\tilde{\mathbf{u}}_\mu^0(\mathbf{p}) \in V_\mu^L$ and the explicit form of $\tilde{\mathbf{u}}_\mu^{\mathcal{N}}(\mathbf{p}) \in V_\mu^{\mathcal{N}}$, such that $\tilde{\mathbf{u}}_\mu^0(\mathbf{p}) + \tilde{\mathbf{u}}_\mu^{\mathcal{N}}(\mathbf{p}) \in V_\mu(\mathbf{p})$. Hence, Problem 3 encompasses the solution of two sub-problems: i) find $\tilde{\mathbf{u}}_\mu^0(\mathbf{p}) \in V_\mu^{\mathcal{N}}$ such that

$$a^R(\mathbf{p}; \tilde{\mathbf{u}}_\mu^0, \mathbf{v}) = b^R(\mathbf{p}; \mathbf{v}) \quad \forall \mathbf{v} \in V_\mu^L, \tag{28}$$

and ii) find $\tilde{\mathbf{u}}_\mu^{\mathcal{N}}(\mathbf{p}) \in V_\mu^L$ such that

$$a^R(\mathbf{p}; \tilde{\mathbf{u}}_\mu^{\mathcal{N}}(\mathbf{p}), \mathbf{v}) = 0 \quad \forall \mathbf{v} \in V_\mu^L. \tag{29}$$

This decoupling is possible thanks to the linearity of the problem.

Remark 6. It is worth mentioning that in general $V_\mu^{\mathcal{N}}$ is not a subset of V_μ^M. In Appendix B we prove this does not impose any limitation to the method and that, in fact, we can equivalently rewrite the original problem using a set $(V_\mu^{\mathcal{N}})^* \subset (V_\mu^M)$ just by incorporating a new term on the right-hand side of the variational formulation in Problem 3.

Remark 7. In practice, it is more convenient to solve Problem 3 (using $V_\mu^{\mathcal{N}}$) at once rather (28) and (29) separately. However, this split motivates and justifies the structure $V_\mu^{\mathcal{N}}$ postulated in (27).

3.2. Overview of the method

Before digging into methodological details of (Deep) Neural Networks (Section 4), it is useful to start a short overview of the method we propose. For easiness of notation, thereafter we denote our method as DeepBND, which stands for Deep Boundary, as reference of a deep learning methodology to enrich multiscale problems with more accurate BCs. It is worth highlighting that the basic requirements behind the development of DeepBND and the strategies adopted to address each of them are the following:

1. Computational efficiency: coupled multiscale simulations are computationally demanding by their very nature. Since the local-problem solving is embarrassingly parallel, this step is the most critical regarding the optimisation of the method. Therefore, such speed-up is mandatory and our main strategy is to reduce the microscale domain.
2. Accuracy: To keep the same accuracy of more computationally demanding larger microscale domains, our strategy focuses on the choice of a more precise BCs for smaller microscale domains.
3. Exploitation of *a priori* knowledge: To be able to enrich the corrector problem with more suitable BCs, we exploit previous knowledge from pre-computed accurate solutions, also called high-fidelity (HF) solutions, which are necessarily more computationally demanding. This is performed by fully exploiting the potential of Reduced Order Modelling using Reduced Basis Method (RB-ROM) in combination with (Deep) Neural Networks (DNN). The use of RB-ROM allows for a more compact representation of the BC, reducing the size of the needed DNN, by automatically ranking the basis in a meaningful order of importance, which will be also exploited to suitably calibrate the training process of the DNN in next section.
4. Easiness of implementation: The very mathematical structure of the local problem is unchanged, only the BCs are touched. Our method leads to non-homogeneous Dirichlet BCs, not leading to any additional difficulties in terms of implementation.

As usual in RB-ROM strategies, DeepBND can be structured into *offline* and *online* phases. Having chosen some meaningful microstructural parametrisation for the HF problem at hand, the main goal of the *offline phase* is to find a mapping that translates parameters from the high-fidelity problem into an accurate BC of the problem solved in the *online phase*. It consists in the following pipeline: i) the dataset acquisition by simulating those models for some representative sampling of the parameter space, ii) the RB construction using the previous dataset, iii) extraction of relevant features of the dataset through the orthogonal projections onto this RB and, iv) the training of a DNN model aiming to predict these projections. Mathematically, this procedure results in a) $\mathcal{B}_{N_{rb}} = \{\boldsymbol{\xi}_i \in [H^{1/2}(\partial\Omega_\mu)]^d\}_{i=1}^{N_{rb}}$, a RB with N_{rb} functions and b) the mapping $\mathcal{N} : \mathbb{R}^{N_p} \to \mathbb{R}^{N_{rb}}$, where N_p is the number of parameters of the high-fidelity model. As for the *online* phase, we can predict the Dirichlet-like BC encoding these two previous outputs into $V_\mu^{\mathcal{N}}$ for a given parameter of the HF problem $\mathbf{p} \in \mathbb{R}^{N_p}$. Finally, we proceed by solving the reduced corrector problem followed by some homogenisation post-processing. The general outline of the method is depicted in Algorithm 2.

Offline phase:

1. Dataset generation
2. Reduced Basis Construction
3. Features extractions (Input-Output)
4. DNN training

Online phase:

1. Given a microstructure and a micro-strain: Prediction of BC using the DNN
2. Solving the microscale problem (Problem 3) using the BC of step 1.
3. Homogenisation of the obtained solution

Algorithm 2: General workflow of the DeepBND method detailing Offline and Online phases.

It is worth mentioning that the RB-ROM and DNN concentrate most of the computational efforts in the *offline phase* rather than in the *online phase*. To generate the dataset, some parametrisation of the microstructure should be chosen, together with the range of these parameters, their probability distributions, and sample strategy. This is a problem-dependent choice

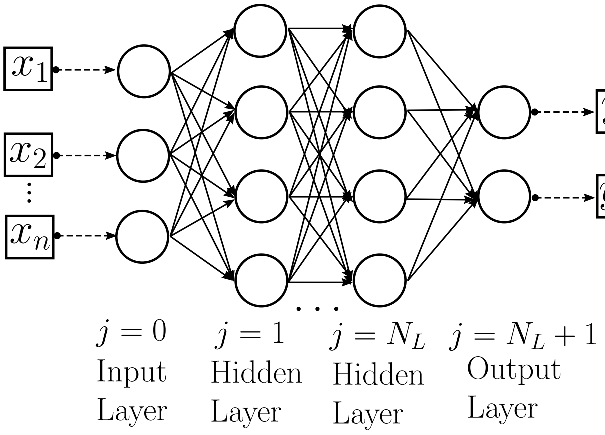


Fig. 4. Multilayer Perceptron Neural Network with N_L hidden layers.

and we discuss in Section 5 the examples treated in the manuscript. In turn, the final cost added in comparison to a standard coupled multiscale simulation is the evaluation of a DNN model to determine the BC that feeds the corrector problem solver. The DNN evaluation step (prediction) is many orders of magnitude less computationally demanding than the corrector problem numerical solution, by using FEM for example.

4. Designing the neural network

In this section, we introduce the DNN model to predict the BCs. As mentioned before, we aim at characterising the nonlinear mapping from the parameter space into RB coefficients $\mathcal{N} : \mathbb{R}^{N_p} \to \mathbb{R}^{N_{rb}}$. In terms of DNN architecture, our model is based on two instances of the standard fully connected Multilayer Perceptron (MLP) model, which is revisited in Section 4.1, combined seamlessly to exploit: i) the splitting between macro-strain and microscale constitutive parameters and ii) the problem symmetries. We also explore such principles for dataset and RB generation (Section 4.2), which, in fact, naturally leads to a specific format for combining MLP submodels, as explained in Section 4.3. Regarding the loss function, we also adopt a non-standard setting motivated from the RB-ROM (see Section 4.4).

4.1. Basic notation for Deep Neural Networks

Here we review some basic concepts and notation for supervised learning using DNN. For the interested reader, we refer to [19,14] for a deeper presentation.

Let $D = \{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}) \in \mathbb{R}^m \times \mathbb{R}^n; i = 1, \dots, N_s\}$ be a *dataset* with N_s samples (or snapshots). We aim at finding the mapping $\mathcal{N}(\boldsymbol{\theta}; \cdot) : \mathbb{R}^m \to \mathbb{R}^n$ such that it minimises the regularised empirical risk

$$\min_{\boldsymbol{\theta}} \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{L}(\mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\alpha}^{(i)}), \boldsymbol{\beta}^{(i)}) + \lambda \mathcal{R}(\boldsymbol{\theta}), \tag{30}$$

where $\mathcal{L} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$ is the so-called cost or loss function, and $\mathcal{R} : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^+$ is a regularising function controlled by an intensity factor $\lambda \in \mathbb{R}^+$. The vector $\boldsymbol{\theta}$ should be understood as a collection of parameters that defines the mapping $\mathcal{N}(\boldsymbol{\theta}; \cdot)$, which is taken accordingly with the usual definition of the MLP architecture for DNN [14]. The size of $\boldsymbol{\theta}$, and thus the complexity of the DNN, depends on the pre-fixed number of hidden layers N_L and the quantity of neurons in each layer (see Fig. 4 to help with the interpretation).

Moreover, concerning the practical implementation of the training, i.e., the optimisation procedure to solve (30), we use the ADAM version of the stochastic gradient with mini-batches, and, as regularisation techniques, we make use of dropout [42], l^2 regularisation, and early stop based on the validation error [14]. The exact numerical values used in our examples are specified in the appropriate section.

4.2. Dataset and RB construction strategies

As seen in Section 3.1.2, the trace $\mathbf{w}$ of the fluctuations on $\partial\Omega_\mu^R$, is the linear combination of auxiliary solutions obtained to each load direction ($\mathbf{w}^{(I)}$), for $I = 1, 2, 3$. Obtaining them involves fewer parameters, since the loads are fixed in unitary directions, which lead to a reduced parameter space to sample. In other words, exploiting this property in the dataset generation means that for a fixed number of high-fidelity snapshots, we can sample better a smaller parameter set $\mathcal{P}_r \subset \mathbb{R}^{N_b}$

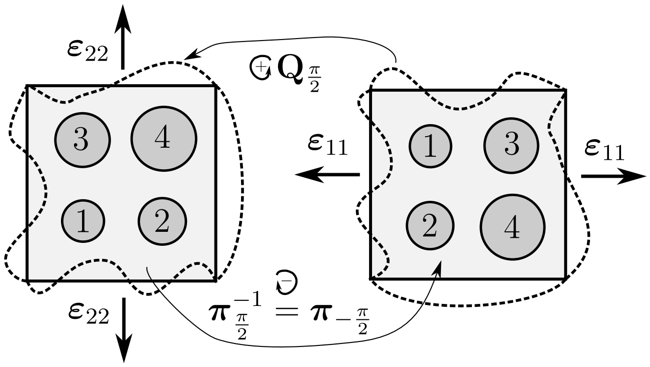

ε_{22}
$\oplus \mathbf{Q}_{\frac{\pi}{2}}$
3
4
1
2
ε_{11}
1
3
2
4
ε_{11}
$\ominus$
ε_{22}
$\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1} = \boldsymbol{\pi}_{-\frac{\pi}{2}}$

Fig. 5. Symmetry in load-geometry relation. Here $\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}((1,2,3,4)) = \boldsymbol{\pi}_{-\frac{\pi}{2}}((1,2,3,4)) = (2,4,1,3)$.

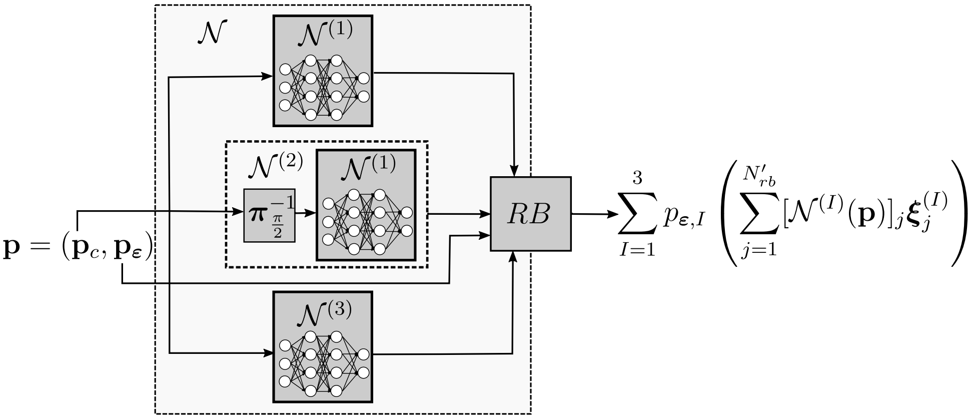

$\mathcal{N}$
$\mathcal{N}^{(1)}$
$\mathcal{N}^{(2)}$
$\mathcal{N}^{(1)}$
$\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}$
RB
$\mathbf{p} = (\mathbf{p}_c, \mathbf{p}_{\boldsymbol{\varepsilon}})$
$\mathcal{N}^{(3)}$
$\sum_{I=1}^{3} p_{\boldsymbol{\varepsilon},I} \left(\sum_{j=1}^{N'_{rb}} [\mathcal{N}^{(I)}(\mathbf{p})]_j \boldsymbol{\xi}_j^{(I)} \right)$

Fig. 6. DeepBND-NN architecture: exploiting the physical structure of the problem using a combination of MLP submodels and RB functions.

instead of $\mathcal{P} \subset \mathbb{R}^{N_p}$, where $N_p = N_b + 3$ (in 2D). In this regard, it is also important to maximise the covering of $\mathcal{P}_r$ by using a smart sampling strategy, for example one of the Quasi-Monte Carlo Family such as the Latin Hypercube Sampling (LHS) used in this work.

Furthermore, in terms of mechanical load, solutions $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ as in (26) are equivalent in the sense one can be obtained by rotation of another, as depicted in Fig. 5. Hence, choosing N'_{rb} basis functions to satisfy some accuracy criteria, we obtain $\mathcal{B}^{(I)}_{N'_{rb}} = \left\{ \boldsymbol{\xi}_i^{(I)} \right\}_{i=1}^{N'_{rb}}$ associated to $\mathbf{w}^{(I)}$ for $I = 1, 3$, by applying the POD procedure of Algorithm 1. The missing basis $\mathcal{B}^{(2)}_{N'_{rb}}$ can be obtained by $\mathcal{B}^{(2)}_{N'_{rb}} = \mathbf{Q}_{\frac{\pi}{2}} \mathcal{B}^{(1)}_{N'_{rb}}$, where the anti-clockwise rotation of $\pi/2$ rad is applied for each function in the basis by means of the orthonormal matrix $\mathbf{Q}_{\frac{\pi}{2}}$. On the other hand, for a given $\mathbf{p}_c \in \mathbb{R}^{N_c}$, the solution $\mathbf{w}^{(2)}(\mathbf{p}_c)$ transforms according to

$$\mathbf{w}^{(2)}(\mathbf{p}_c) = \mathbf{w}((\mathbf{p}_c, \mathbf{e}_2)) = \mathbf{Q}_{\frac{\pi}{2}} \mathbf{w}((\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}(\mathbf{p}_c), \mathbf{e}_1)) = \mathbf{Q}_{\frac{\pi}{2}} \mathbf{w}^{(1)}(\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}(\mathbf{p}_c)), \tag{31}$$

where $\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}$ denotes a permutation on the parameter vector entries to reflect a clockwise rotation of $\pi/2$ rad on the microstructure pattern (see Fig. 5). The actual format for the permutation is problem- and implementation-dependent. In the next section, we retake these ideas to construct the DNN model that predicts the BCs.

4.3. DeepBND-NN model: definition and combination of the axial and shear DNN submodels

As already commented, before defining the DNN model, we need to obtain a DNN submodel for the axial unitary macro-strain (without loss of generality, chosen here as the horizontal one) and another DNN submodel for shear unitary macro-strain. Following the previous conventions, we denote them as $\mathcal{N}^{(1)}, \mathcal{N}^{(3)} : \mathbb{R}^{N_b} \to \mathbb{R}^{N'_{rb}}$, for the axially horizontal and shear loading, respectively. In turn, for the axially vertical loading, we invoke (31) to define $\mathcal{N}^{(2)}(\mathbf{p}) = \mathcal{N}^{(1)}\left(\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}(\mathbf{p})\right)$.

Finally, the BC prediction for some $\mathbf{p} \in \mathbb{R}^{N_p}$ reads as

$$\tilde{\mathbf{u}}_{\mu}^{\mathcal{N}}(\mathbf{p})|_{\partial\Omega_{\mu}^{R}} = \sum_{I=1}^{3} p_{\boldsymbol{\varepsilon},I} \left(\sum_{j=1}^{N'_{rb}} [\mathcal{N}^{(I)}(\mathbf{p_c})]_j \boldsymbol{\xi}_j^{(I)} \right) \in L^2(\partial\Omega_{\mu}^{R}). \tag{32}$$

It is important to mention that by taking into account the physical structure of the problem into the DNN architecture (reviewed in Fig. 6), we are able to: i) minimise the size of datasets needed since the size of parameter space (input) is smaller for each submodel; ii) use fewer number of neurons and layers (smaller complexity of the DNN architecture) since the prediction of directional load-wise solutions are also less complex; iii) profit from the full advantage of the RB representation due to their specialisation according to the load; iv) use fewer RB modes for the same accuracy; v) naturally take into account symmetries concerning horizontal and vertical loads and vi) training of 2 instead of 3 submodels. All points mentioned here are intrinsically related and overall render better accuracy and smaller computational burden in the dataset generation and training.

4.4. DNN-POD error analysis and loss function choice

In this section, we precisely define the total error committed in the BC prediction, which accounts for the intrinsic truncation error when a RB is chosen (POD error) and also the error due to the DNN prediction. The latter naturally yields to the natural choice for the loss-function $\mathcal{L}$ used in (30). All notation here should be understood for one specific load, axial or shear, with indices omitted for convenience.

First, let us consider some $\mathcal{B}_{N_{rb}}$ and $\mathbb{S}$ given. For some $i = 1, \dots, N_s$ and target HF solution $\mathbf{w}^{(i)} \in \mathbb{S} \subset L^2(\partial\Omega_{\mu}^{R})$, let us denote its projection upon $\operatorname{span}\mathcal{B}_{N_{rb}}$ as $\Pi_{N_{rb}}\mathbf{w}^{(i)} := \sum_{j=1}^{N_{rb}} \beta_j^{(i)} \boldsymbol{\xi}_j$, where $\beta_j^{(i)}$ represents its projection along $\boldsymbol{\xi}_j$. Analogously, we denote the DNN prediction of $\mathbf{w}^{(i)}$ as $\hat{\mathbf{w}}^{(i)} := \sum_{i=1}^{N_{rb}} \hat{\beta}_j^{(i)} \boldsymbol{\xi}_j \in \operatorname{span}\mathcal{B}_{N_{rb}}$, where $\hat{\beta}_j^{(i)}$ follows the same interpretation of $\beta_j^{(i)}$.

A natural choice to measure the expected error is to consider the empirical averaged error over all elements of $\mathbb{S}$ namely

$$\mathcal{E}_T^2(N_{rb}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \|\hat{\mathbf{w}}^{(i)} - \mathbf{w}^{(i)}\|_{L^2(\partial\Omega_{\mu}^{R})}^2, \tag{33}$$

where the sub-index $(\cdot)_T$ stands for total error, as opposed to the contributions to the POD and DNN errors defined next. To this aim, using the fact that $\mathbf{w}^{(i)} - \Pi_{N_{rb}}(\mathbf{w}^{(i)}) \in (\operatorname{span}\mathcal{B}_{N_{rb}})^{\perp}$ and that $\mathcal{B}_{N_{rb}}$ is orthonormal, it yields

$$\begin{aligned} \|\hat{\mathbf{w}}^{(i)} - \mathbf{w}^{(i)}\|_{L^2(\partial\Omega_{\mu}^{R})}^2 &= \|\hat{\mathbf{w}}^{(i)} - \mathbf{w}^{(i)} + \Pi_{N_{rb}}(\mathbf{w}^{(i)}) - \Pi_{N_{rb}}(\mathbf{w}^{(i)})\|_{L^2(\partial\Omega_{\mu}^{R})}^2 \\ &= \|\hat{\mathbf{w}}^{(i)} - \Pi_{N_{rb}}(\mathbf{w}^{(i)})\|_{L^2(\partial\Omega_{\mu}^{R})}^2 + \|\mathbf{w}^{(i)} - \Pi_{N_{rb}}(\mathbf{w}^{(i)})\|_{L^2(\partial\Omega_{\mu}^{R})}^2 \\ &= \|\hat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^{(i)}\|_2^2 + \|\mathbf{w}^{(i)} - \Pi_{N_{rb}}(\mathbf{w}^{(i)})\|_{L^2(\partial\Omega_{\mu}^{R})}^2, \quad \text{for}\, i = 1, \dots, N_s. \end{aligned} \tag{34}$$

The boldface notation for $\hat{\boldsymbol{\beta}}^{(i)}$ and $\boldsymbol{\beta}^{(i)}$ simply mean the component-wise collection for their respective projections for a snapshot i. Note that the second term of the above expression when averaged over the N_s snapshots corresponds to the POD mean squared error ($\mathcal{E}_{POD}^2$) already defined in (10). On the other hand, the first term accounts for the DNN error, in which the mean squared version is defined below

$$\mathcal{E}_{DNN}^2(N_{rb}) := \frac{1}{N_s} \sum_{i=1}^{N_s} \|\hat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^{(i)}\|_2^2. \tag{35}$$

The expression above provides a physically-based formula for the loss function to be considered in (30). More precisely,

$$\mathcal{L}(\hat{\boldsymbol{\beta}}^{(i)}, \boldsymbol{\beta}^{(i)}) := \frac{1}{N_{rb}} \sum_{j=1}^{N_{rb}} \omega_j (\bar{\hat{\beta}}_j^{(i)} - \bar{\beta}_j^{(i)})^2, \tag{36}$$

where we have used the min-max scaling $\overline{(\cdot)} = \frac{(\cdot) - \beta_j^{min}}{\beta_j^{max} - \beta_j^{min}}$ (both for $\hat{\beta}_j^{(i)}$ and $\beta_j^{(i)}$), where: $\beta_j^{min} = \min_i \beta_j^{(i)}$, $\beta_j^{max} = \max_i \beta_j^{(i)}$, $\omega_j = (\beta_j^{max} - \beta_j^{max})^2$, for $j = 1, \dots, N_{rb}$.

Finally, we can rewrite (33) as

$$\mathcal{E}_T^2(N_{rb}) = \mathcal{E}_{DNN}^2(N_{rb}) + \mathcal{E}_{POD}^2(N_{rb}), \tag{37}$$

where we should notice that while $\mathcal{E}_{POD}^2$ has a monotonous decreasing (w.r.t. N_{rb}) estimation given by (10), the $\mathcal{E}_{DNN}^2$ depends on the DNN architecture, initialisation, optimisation algorithm, etc. Therefore, to assess an optimal N_{rb} for a given DNN architecture we should study the trade-off between these two errors. This is explored in Section 5.

Table 1
From left to right: the two Lamé coefficients (λ^1_μ and G^1_μ) for the matrix and the multiplicative constant factor (γ) in the inclusion applied for both previous coefficients. The values for λ^1_μ and G^1_μ corresponds to $E = 1.0$ (units disregarded), the Young modulus, and $\nu = 0.3$, the Poisson ratio, using standard transformation expressions for Hookean isotropic media.

λ^1_μ	G^1_μ	γ
0.576923	0.384615	10

It is worth highlighting that the RB-based strategy adopted also induces a specific choice of weighted mean-squared error, where weights are given by eigenvalues found in the POD procedure, which are ranked by magnitude. This choice also leads to a natural total estimator in the $L^2(\partial\Omega^R_\mu)$-norm, which is the intrinsic choice to measure the BC discrepancies, that links the loss function reached in the training and the error committed by POD truncation itself. Importantly, each submodel can be trained separately and be put together only for prediction purposes, which is interesting from the computational point of view.

5. DNN model construction: datasets, RB, and training

In this section, we specify the choice of numerical parameters adopted to assess our method. We choose to constrain the more general setting of Sections 3.1 and 4. As for the size High-fidelity and Reduced domains with respect to the number of inclusions, we take Ω^H_μ containing 36 circular inclusions arranged in a 6 × 6 grid, with and Ω^R_μ restricted to the central 2 × 2 grid. We refer to the Remarks 9 and 8 to comments about this particular choice. The fixed coefficients used in all simulations are presented in Table 1, while the random sampling of the inclusion sizes, which are the random parameters of the problem, are shown next.

Concerning the inclusions, they are placed in a 6 × 6 grid (see (24)), so $N_b = 36$, with radii computed as follows

$$r^{(i)}_j = \exp(a + \theta^{(i)}_j b) \in [r_{min}, r_{max}], \qquad j = 1, \dots, N_b,\ i = 1, \dots, N_s, \tag{38}$$

where $a = \log(r_{max} r_{min})$, $b = \log(\frac{r_{max}}{r_{min}})$, $r_{max} = 0.4\frac{L^H_\mu}{\sqrt{N_b}} = \frac{L^H_\mu}{15}$ and $r_{min} = 0.1\frac{L^H_\mu}{\sqrt{N_b}} = \frac{L^H_\mu}{60}$, and $\theta^{(i)}_j \in [-1, 1]$ is a random variable detailed next. The strategy (38) is used by [28] in a similar scenario and has the advantages of being empirically inspired and also a simple manner to constrain the value range for the radii. Regarding the sampling, we use the LHS with an underlying uniform distribution to sample $\theta^{(i)}_j$ for $j = 1, \dots, N_b$, $i = 1, \dots, N_s$. Notice that the LHS requires *a priori* knowledge of N_s and N_b, since it aims at achieving an optimal coverage of the high-dimensional parameter while keeping the number of samples small. Indeed, each realisation is not independent from each other. We use the default LHS implementation provided `scikit-opt` library [4], with three different seeds for random generation and sample sizes according to Table 2, yielding the samples $\mathbb{P}^{(1)}$, $\mathbb{P}^{(2)}$, and $\mathbb{P}^{(3)}$. Examples of some realisations of microstructures are found in Fig. 7.

As already discussed, the DNN model construction encompasses the RB extraction and the training of a DNN for a given dataset, denoted here as *training/RB* dataset, with size $N^{train}_s = 51200$. Moreover, from the point of view of DNN training, we also need to simulate auxiliary datasets for *validation* and *testing* reasons, in which the size is assumed to be respectively 20% ($N^{val}_s = 10240$) and 10% ($N^{tes}_s = 5120$) of the *training/RB* dataset size. Importantly, each dataset is considered for the axial and shear loads separately, as summarised in Table 2.

It is widely known in the multiscale solid mechanics literature that the volume fraction $\phi = \frac{\sum_{i=1}^{N_b} \pi r_i^2}{(L^{HF}_\mu)^2}$ is a random variable that plays a crucial role in the final solution. Consequently it is worth looking at how this variable is indirectly sampled by the assumed random generation algorithm, which focuses only on individual radii. In Fig. 8, checking the histograms of ϕ for $\mathbb{P}^{(1)}$, $\mathbb{P}^{(2)}$, and $\mathbb{P}^{(3)}$, we can observe that, although not implicitly enforced, the resulting underlying probability distribution seems to be consistent along the several datasets. We can also perfectly use another sample strategy, provided that the resulting volume fractions remain close to the range that the model was designed for.

For obtaining the HF solutions, we use the FEM with quadratic polynomials in triangular meshes. For the mesh generation using *gmsh* [13], we have imposed 20 equally spaced divisions for each internal boundary segment (bottom, right, top and left of $\partial\Omega^R_\mu$), making a total of 160 degrees of freedom on the boundary, for all snapshots. The overall unstructured mesh on Ω^H_μ is then generated using the corresponding characteristic length size of the internal boundary. Accordingly, the number of functions in the RB will be at most the total number of degree of freedom on the boundary, and the associated POD-error should vanish close to this value. Indeed, the largest RB considered here is 140, which actually gives practically zero POD-error.

Concerning the computational time, all simulations were performed in a cluster with 80 (160 threads) Intel(R) Xeon(R) Gold 6148 CPU @ 2.40 GHz cores and the results of solving each HF simulation are resumed in Table 3. The *Total Time* displayed is an estimate based on the statistics computed over all snapshots, i.e., the number of snapshots times the mean time spent per snapshot. The Wall Time is computed supposing the use of 32 cores and such that the Total Time is split

Table 2
Summary datasets for the different combinations of “Load/Use” scenarios. Each “Use” is associated with one LHS run for a given `seed` and number of samples.

Index	Load	Use	Size (N_s)	Sample
1	Axial	Training/RB	$N_s^{trb} = 51200$	$\mathbb{P}^{(1)}$ (LHS(N_s^{trb}, $\texttt{seed} = 1$))
2	Shear	Training/RB	$N_s^{trb} = 51200$	$\mathbb{P}^{(1)}$ (LHS(N_s^{trb}, $\texttt{seed} = 1$))
3	Axial	Validation	$N_s^{val} = 10240$	$\mathbb{P}^{(2)}$ (LHS(N_s^{val}, $\texttt{seed} = 2$))
4	Shear	Validation	$N_s^{val} = 10240$	$\mathbb{P}^{(2)}$ (LHS(N_s^{val}, $\texttt{seed} = 2$))
5	Axial	Testing	$N_s^{tes} = 5120$	$\mathbb{P}^{(3)}$ (LHS(N_s^{tes}, $\texttt{seed} = 3$))
6	Shear	Testing	$N_s^{tes} = 5120$	$\mathbb{P}^{(3)}$ (LHS(N_s^{tes}, $\texttt{seed} = 3$))

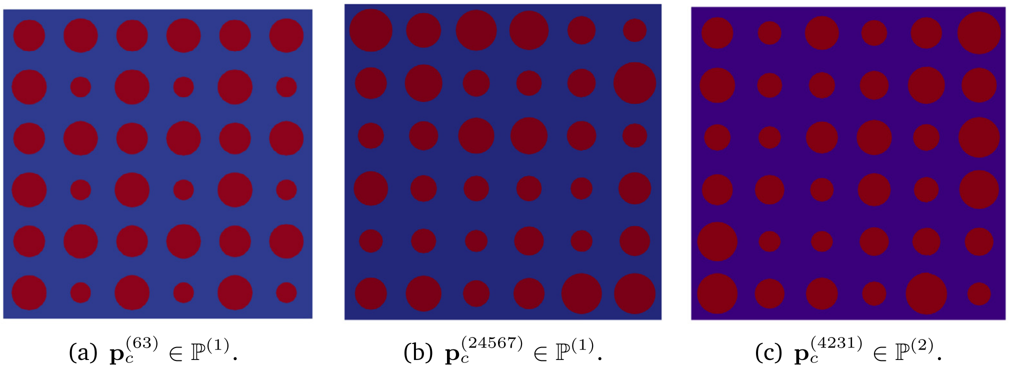

(a) $\mathbf{p}_c^{(63)} \in \mathbb{P}^{(1)}$.
(b) $\mathbf{p}_c^{(24567)} \in \mathbb{P}^{(1)}$.
(c) $\mathbf{p}_c^{(4231)} \in \mathbb{P}^{(2)}$.

Fig. 7. Some realisations of microstructures for the high-fidelity model showing the different phases of Ω_μ^H: $\Omega_\mu^{H,1}$ in blue and $\Omega_\mu^{H,2}$ in red (for interpretation of the colours in the figures, the reader is referred to the web version of this article).

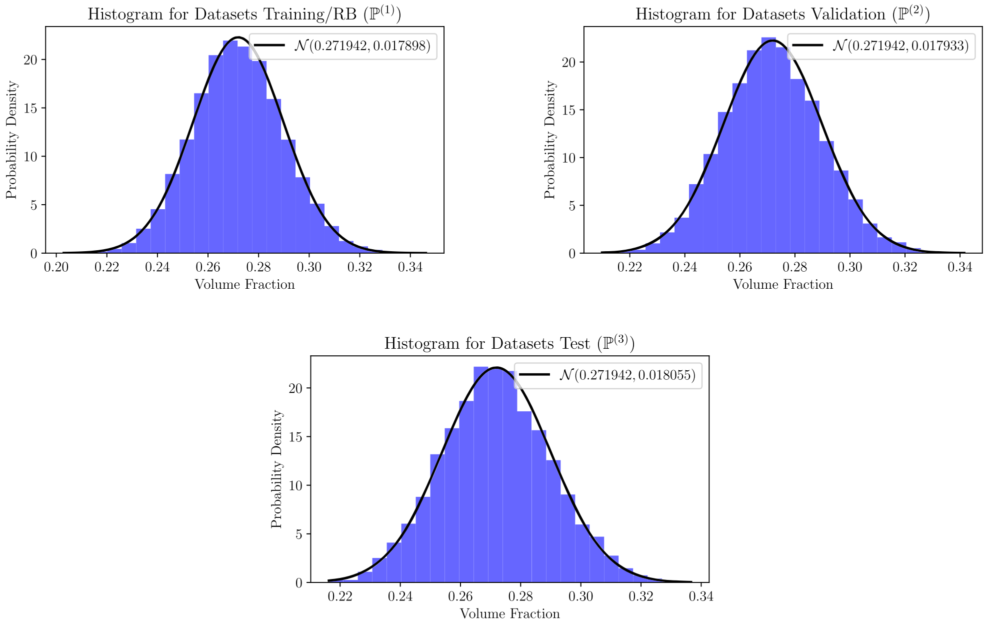


Fig. 8. Volume fraction histograms of three different types of datasets. The notation $\mathcal{N}(\cdot,\cdot)$ represents the fitted Normal distribution (do not mix with a DNN).

equally. These estimates make sense only in embarrassingly parallel scenarios, as it is the case of the problem at hand. Notwithstanding, the ultimate time expended depend on many other factors, which are out of the scope of our analysis, such as the implementation, number of cores/threads, memory available, etc. Furthermore, the time expended in the RB-ROM steps and the mesh generation is negligible compared to the Wall Time for the *training/RB* dataset, and for this reason, we do not report them.

Table 3
Computational cost of the dataset generation.

	Training/RB	Validation	Test
Time per snapshot(s)	43.16 ± 5.34	45.11 ± 10.60	47.91 ± 12.96
Total Time (h)	613.8	128.3	68.14
Wall Time (h) (32 Cores)	19.18	4.009	2.129

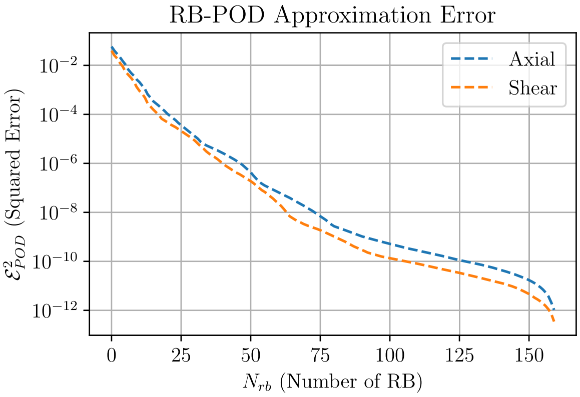


Fig. 9. Error committed in the RB-ROM approximation.

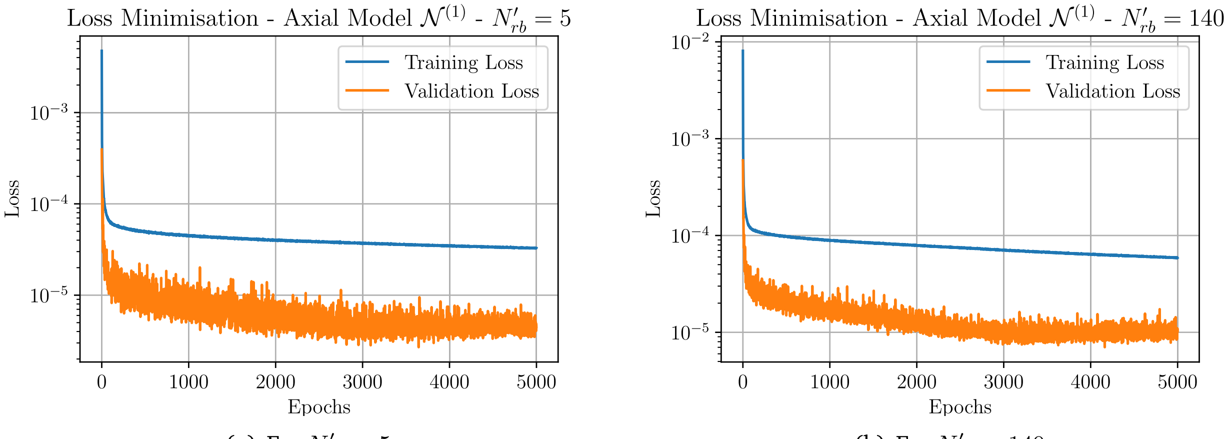
Finally, by applying POD procedure of Algorithm 1 to the datasets 1 and 2, we can evaluate the committed error in the sense of (10), as shown in Fig. 9. This error represents the best expected approximation of the BC in the sense of $L^2(\partial\Omega^R_\mu)$, i.e., with no error due to the DNN approximation. We can verify these two errors are comparable, with a slight advantage to the shear over the axial model, for a given number of RB functions. As result, we have adopted the same DNN architecture and number of RB in both models, since they have comparable complexity.

To define the models $\mathcal{N}^{(1)}$ and $\mathcal{N}^{(3)}$, it is worth studying the sensibility of the total error $\mathcal{E}_T$ with the number of RB (see (33)). It is reasonable to expect that at some point the decreasing of $\mathcal{E}_{POD}$ may not compensate the increasing of $\mathcal{E}_{DNN}$ as N'_{rb} gets larger. Therefore, we have considered $N'_{rb} \in \{5, 10, 20, 40, 80, 140\}$ and have trained 6 different DNN architectures (one for each N'_{rb}), which do not differ regarding the hidden layers, but only the number of entries on the output. Also, we have considered a DNN architecture of 3 hidden layers with 300 neurons each. As activation functions, we have used the Swish [31] in the hidden layers followed by a linear activation in the output layer. To regularise the model, we have used the dropout with probability of retention $p = 99,5\%$ in all hidden layers and $\lambda = 10^{-8}$ for l^2 regularisation for all weights and biases. The DNN has been trained using the ADAM optimiser with default parameters and mini-batches of 32 samples. An adaptive learning rate has been employed, decreasing linearly from 5.0×10^{-4} until 5.0×10^{-5} in epoch 5000 (last). It is worth mentioning that we reached these values by systematically testing combinations of smaller architectures, regularisation values, and activation functions. Therefore, such values should be understood as guidance for problems with similar complexity and dataset sizes, although there might be still room for small tunings. Finally, we used `Tensorflow` [1] for the implementation, and the final model is assumed to be the one yielding to the least loss function value in the validation dataset.

In Figs. 10(a) and 10(b) we can see the loss function history ($\mathcal{E}^2_{DNN}$) for the axial DNN model using $N'_{rb} = 5$ and $N'_{rb} = 140$, respectively. Loss function values are naturally larger for $N'_{rb} = 140$, since the output has more entries. The loss function evaluated on training and validation differ considerably as result of regularisation, which is not used in validation, the latter turning out to be smaller. For the last epochs, the validation reaches a plateau while the training error continues to decrease slightly. This indicates that the model reached its full capacity while avoiding overfitting. The very same comments apply to Figs. 11(a) and 11(b), for the shear DNN model. For the results with different N'_{rb}, check Appendix C.

Lastly, the final trained DNN models are compared in terms of the DNN, POD, and total errors (in test datasets) in Figs. 12(a) and 12(b) for the axial and shear cases, respectively. For the axial case we observe the least total error for $N'_{rb} = 140$ while in the shear case both $N'_{rb} = 80$ and $N'_{rb} = 80$ give equivalent quantities. Therefore, we choose to keep $N'_{rb} = 140$ to the numerical examples of next section.

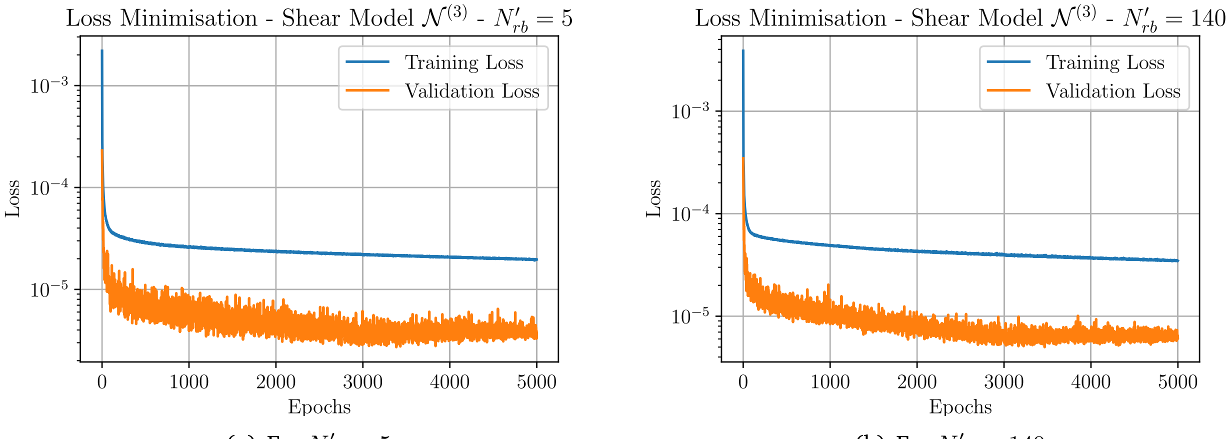
Remark 8. It is worth remembering since we are using relatively small MDs (Ω^H_μ and Ω^R_μ) if compared to the size for a Representative Volume Element (RVE) [40], homogenised quantities can significantly vary according to the specific microstructural random realisation [22,26], and also due to the applied BC influence [22,45]. The gap between Ω^H_μ and Ω^R_μ is motivated by the idea of weighted or truncated average [45] and aims at decreasing, practically eliminating, the bias of the classical BCs applied in $\partial\Omega^H_\mu$. The subsequent physically-based learning-prediction process derives a most suitable adaptive



(a) For $N'_{rb} = 5$.

(b) For $N'_{rb} = 140$.

Fig. 10. Historic for the loss function minimisation in Axial model $\mathcal{N}^{(1)}$. Training loss evaluated with regularisation (dropout and l^2), while validation loss in prediction mode (regularisation disabled).



(a) For $N'_{rb} = 5$.

(b) For $N'_{rb} = 140$.

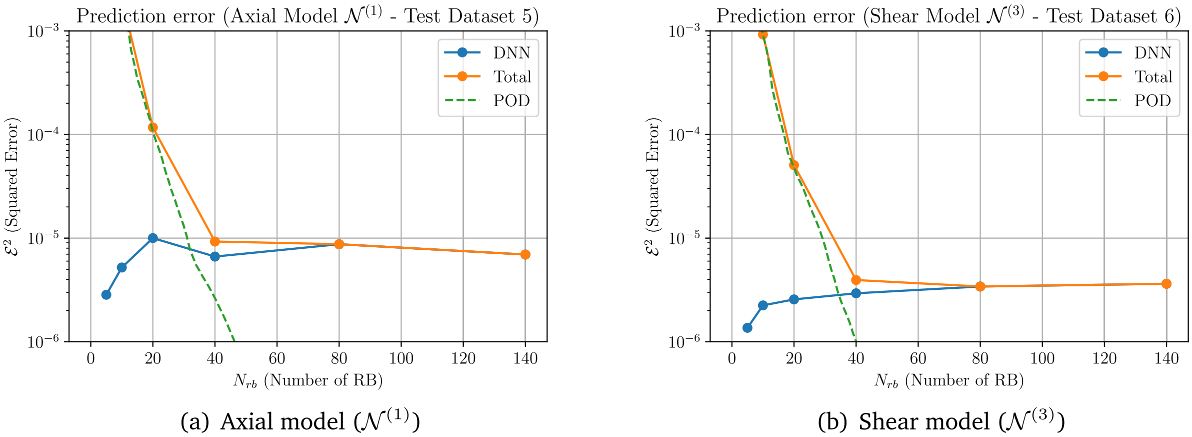
Fig. 11. Historic for the loss function minimisation in Shear model $\mathcal{N}^{(3)}$. Training loss evaluated with regularisation (dropout and l^2), while validation loss in prediction mode (regularisation disabled).

BC for $\partial\Omega_\mu^R$, which lies in the manuscript's core contributions. On the other hand, the usual remedy for the subsampling assumption is to resort to stochastic finite element techniques [22,44,29]. In the examples of Section 6.1 and 6.3.1, we mitigate the effect of subsampling by comparing results in terms of means and standard deviation values among several random microstructural realisations. However, the rigorous treatment of subsampling altogether with the BCs will is out of the scope of this contribution.

Remark 9. It should be mentioned that disregarding factors such as the DNN training and database sizes, in a broader sense, the chosen sizes for Ω_μ^H and Ω_μ^R sets a theoretical lower bound for the accuracy of the method. As already mentioned, it is known that this error decays with the MD size [45] in a ratio of $\frac{1}{L_\mu}$ with proportionality constants dependent on the type of classical BCs. A deterministic study for periodic media can easily be performed by replicating unit cells. Nevertheless, for non-periodic media, a precise statistical analysis that also accounts for the dispersion of heterogeneity is more challenging, and performing it is out of the scope of this work (see for instance [43,40]).

6. Numerical examples

In this section we aim at assessing the DeepBND method in practice, i.e., in coupled two-scale simulations, also known as FE^2 [10]. Two kinds of problems are considered (recall the notation of Fig. 1): i) the first with a strong scale separation, i.e., the size of heterogeneities is several orders of magnitude smaller than the dimension of the body such that $L_\mu \ll L$;



(a) Axial model ($\mathcal{N}^{(1)}$)

(b) Shear model ($\mathcal{N}^{(3)}$)

Fig. 12. Mean Squared Prediction error (test datasets) for the $[H^{1/2}(\partial\Omega_\mu)]^d$: DNN and POD contributions yields to the Total error.

and ii) the second with $L_\mu < L$ with about one or two orders of magnitude of difference. As already commented, the computational homogenisation approach is best suited for the i)-kind-problem, however due practical reasons ii)-kind-problem is also interesting since only in this situation direct numerical simulations (DNS) are feasible. The two kinds of problems differ substantially in terms of how the microstructures are generated at integration points. We postpone the presentation of further details to Section 6.1 and Section 6.2.

Before digging into the numerical results, it is worth mentioning that we aim at answering the following questions:

1. How does DeepBND compare in terms of accuracy with the periodic condition using the high-fidelity solution as reference?
2. How does DeepBND compare in terms of computational cost the high-fidelity model?
3. How does DeepBND compare in terms of accuracy with the periodic and high-fidelity approaches using the DNS as the reference solution?

Here, it is worth recalling that the DeepBND and periodic models refer to local problems posed in Ω_μ^R, thus with the BC imposed on $\partial\Omega_\mu^R$, while high-fidelity stands for local problems Ω_μ^H, with periodic conditions imposed on $\partial\Omega_\mu^H$. We decided to choose periodic BCs for comparisons, since it is widely accepted that this model usually leads to more accurate results in contrast with classical models for BCs. As a natural choice, we answer questions 1 and 2 above using a i)-kind-problem (Section 6.1), and question 3 using a ii)-kind-problem (Section 6.2).

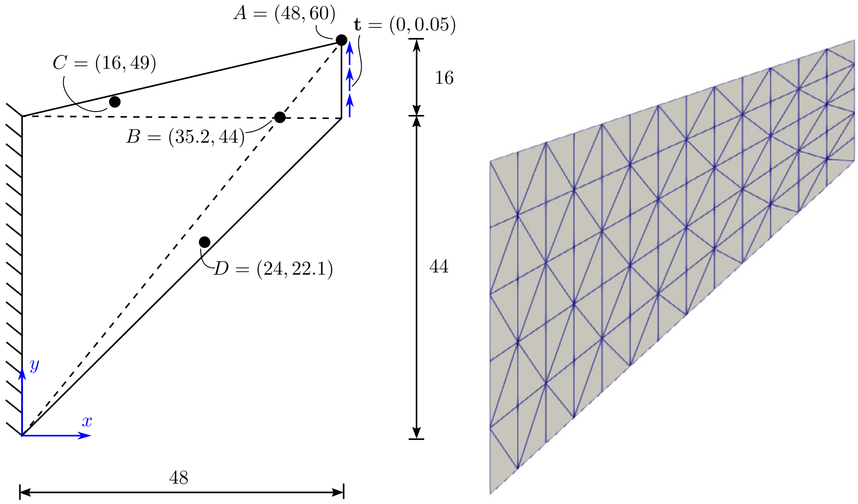
As a proof-of-concept, we also reconsider the i)-kind-problem with different kinds of inclusion patterns, such as randomly oriented ellipses and circular inclusions with random positions. The details of the database generation and results obtained are presented in Section 6.3.

Concerning the computational framework, in all numerical examples we use the *micmacsfenics* library [33], an open-source Fenics-based library for the implementation classical BCs in the FE^2 setting, which can be alternatively found as part of *multiphenics* [5]. As for the DeepBND part, we have used *deepbnd* [34].

6.1. Cook membrane

In this example for i)-kind-problem we apply the DeepBND to the so-called Cook Membrane, a classical benchmark in solid mechanics normally used to test numerical methods for incompressible material [41], but also used in computational homogenisation [37] and to test ML-based surrogate models [24]. It consists in a solid structure with dimensions as in Fig. 13(a) subjected to an homogeneous distributed vertical load $\mathbf{t}$ on the rightmost face. The magnitude of $\mathbf{t}$ and the material model itself varies among the literature considered. Therefore, for the sake of simplicity, we decided to consider the very same material as used in the dataset construction of Section 5 and $\mathbf{t} = (0, 0.05)$, which gives approximately the same vertical displacement of the tip A found in [41].

We aim at comparing the results obtained using the boundary conditions i) DeepBND and ii) Periodic against our reference solution, obtained from high-fidelity microstructures. For the sake of simplicity, we choose the MDs randomly from the *validation* and *test* datasets from Section 5, already at hand. They are sorted among the elements without repetition, which is possible since there are fewer integration points in the finer mesh than the pre-simulated structures microstructures available. To obtain more reliable results, we always report results considering the average over 10 realisations, where each realisation should be understood as a different sorting of MDs along the integration points. Importantly, also notice that due the scale separation, the MDs choices keeps no neighbouring relation.



(a) Schema showing dimensions and points of interest. (b) Coaser mesh considered, containing 5 elements on the vertical direction.

Fig. 13. Cook membrane.

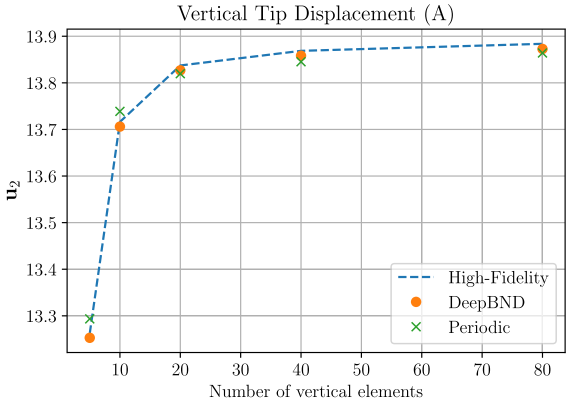


Fig. 14. Convergence of the vertical displacement at the tip (A) in averaged values for 10 realisations of the MDs sorting.

As first study, we consider the convergence of the vertical displacement of the tip A (Fig. 13(a)) with respect to the mesh divisions in the vertical direction. The coarser mesh looks like the one depicted in Fig. 13(b), with 5 regular divisions along the vertical direction and a the horizontal number of divisions such as we keep the same characteristic length of the rightmost side. We then build the other finer meshes with 10, 20, 40, and 80 divisions. We can see the results in Fig. 14, where a clear advantage of DeepBND against the periodic can be observed when compared with the high-fidelity results. In numerical terms, for the finest mesh we have gained between one and two orders of magnitude in the mean relative error norm (see Table 4). Moreover, the variability in the relative error is also decreased, corroborating that DeepBND introduces less boundary layer spurious biases on the results if compared with periodic condition.

It is worth remarking that the average procedure to report the results has been adopted to increase the confidence of the obtained results, mitigating the effect of the microstructure subsampling with small MDs (see Remark 8). We should also note that the reference case (High-Fidelity) is itself homogenised in Ω_μ^R, hence our comparisons focus exclusively on the effect of the BCs choice, being worthwhile regardless of the subsampling assumption, which is equally present for all the cases simulated (DeepBND, periodic and High-Fidelity).

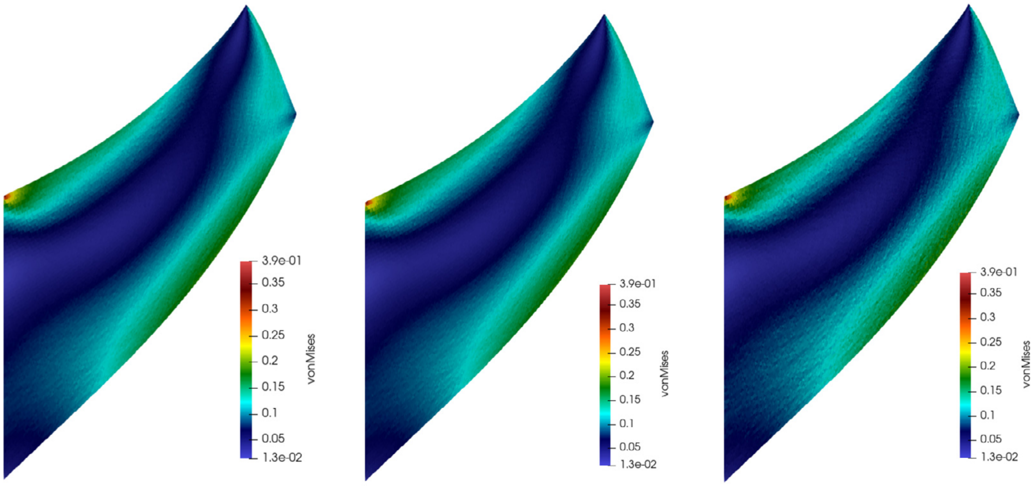
As a second comparison, Fig. 15 shows the von Mises equivalent stress field for the different models, where we can notice that the results for the periodic case (rightmost plot) are considerably noisier, which means worse, compared to the DeepBND simulation (central plot) that visually presents identical results to High-fidelity solution (leftmost plot). Such discrepancies become clearer by plotting errors of the Von Mises stresses, as depicted Fig. 16, in which DeepBND performs about 3 orders of magnitude better than the periodic case. Similar conclusions for the stress are also retrieved component-

Table 4
Relative errors in the finest mesh.

	DeepBND	Periodic
Vertical Displacement A	$7.923 \times 10^{-4} \pm 1.091 \times 10^{-4}$	$1.417 \times 10^{-3} \pm 5.496 \times 10^{-4}$
Von Mises B	$2.185 \times 10^{-4} \pm 1.758 \times 10^{-4}$	$3.126 \times 10^{-2} \pm 1.722 \times 10^{-2}$
Von Mises C	$2.444 \times 10^{-4} \pm 1.799 \times 10^{-4}$	$1.096 \times 10^{-2} \pm 1.060 \times 10^{-2}$
Von Mises D	$1.288 \times 10^{-4} \pm 9.262 \times 10^{-5}$	$6.096 \times 10^{-3} \pm 4.362 \times 10^{-3}$

Table 5
Computational cost for the simulations using DeepBND and High-Fidelity cases. To a fairer comparison in terms of computational times, note that the values reported for the High-Fidelity corresponds to the case that is implemented via homogeneous Dirichlet BCs (Linear BC space), thus differing from those from Table 3, implemented with periodic BCs.

	DeepBND	HF MD (Linear BC)
Time per snapshot(s)	0.5148 ± 0.09548	30.56 ± 7.238
Total Time Finer Mesh (h)	1.831	108.6
Wall Time Finer Mesh (h) (32 Cores)	0.05722	3.394
Speed-up (ratio time per snapshot)	59.36×	1×



(a) σ_{vm}^{HF}.

(b) σ_{vm}^{dnn}.

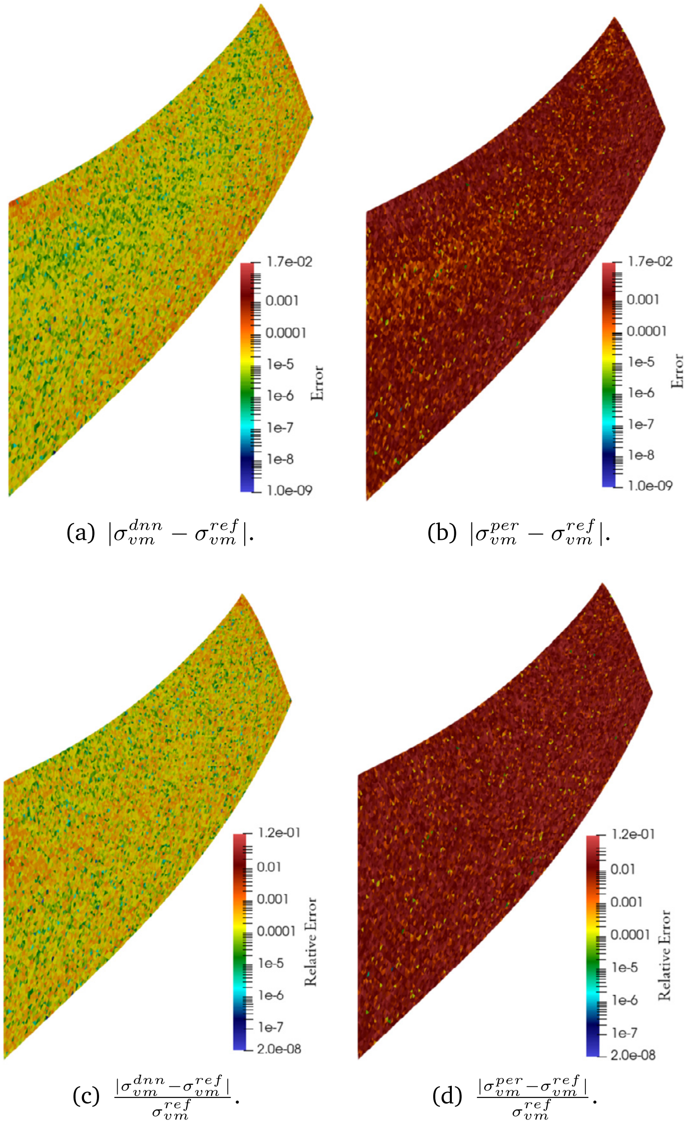
(c) σ_{vm}^{per}.

Fig. 15. Comparison for the Von Mises Stress (σ_{vm}) in the Cook membrane.

wise in Fig. 17. Analysing Fig. 15, 17, and 13(a), we can see that points B, C and D were placed where extreme values for stresses are retrieved. Comparing the Von Mises stress along all realisations, DeepBND gains between one and two orders of magnitude in accuracy with respect to the periodic case, depending on the point considered, as also seen in Table 4.

In terms of computational times, we can see in Table 5 that the computational speed-up reached by DeepBND with respect to HF with linear BCs was about 59×. This number refers only to the embarrassingly parallel part of the problem, which in this case is the assembly of FEM matrices for the macroscale problem, specifically the solution of the local problems. The remaining fixed time for solving the macroscale problem is the same, hence no reason of including it here. Also, the mesh generation is not taken into account, but if taken, the speed-up would be even higher. The DeepBND was not compared directly with periodic BCs times, since due to the more efficient native Fenics/Multiphenics implementation of enforcing Dirichlet BCs with DeepBND, instead of the periodicity which is imposed by Lagrange Multipliers with additional pure Python routines.

In contrast, an approach combining ROM and DNN aiming at fully replacing constitutive laws with surrogate models reports 20× speed-up (comparing with the same authors' customised implementation of the FE^2) [12]. Although not directly comparable, the aforementioned contribution and the DeepBND show the same order of magnitude for speed-ups, which at least corroborates positively with the competitiveness of our approach. On the other hand, the computational gains we reported previously are fairer in the sense the same computational implementation is concerned and it compares strategies targeting the same goal.



(a) $|\sigma_{vm}^{dnn} - \sigma_{vm}^{ref}|$.

(b) $|\sigma_{vm}^{per} - \sigma_{vm}^{ref}|$.

(c) $\frac{|\sigma_{vm}^{dnn} - \sigma_{vm}^{ref}|}{\sigma_{vm}^{ref}}$.

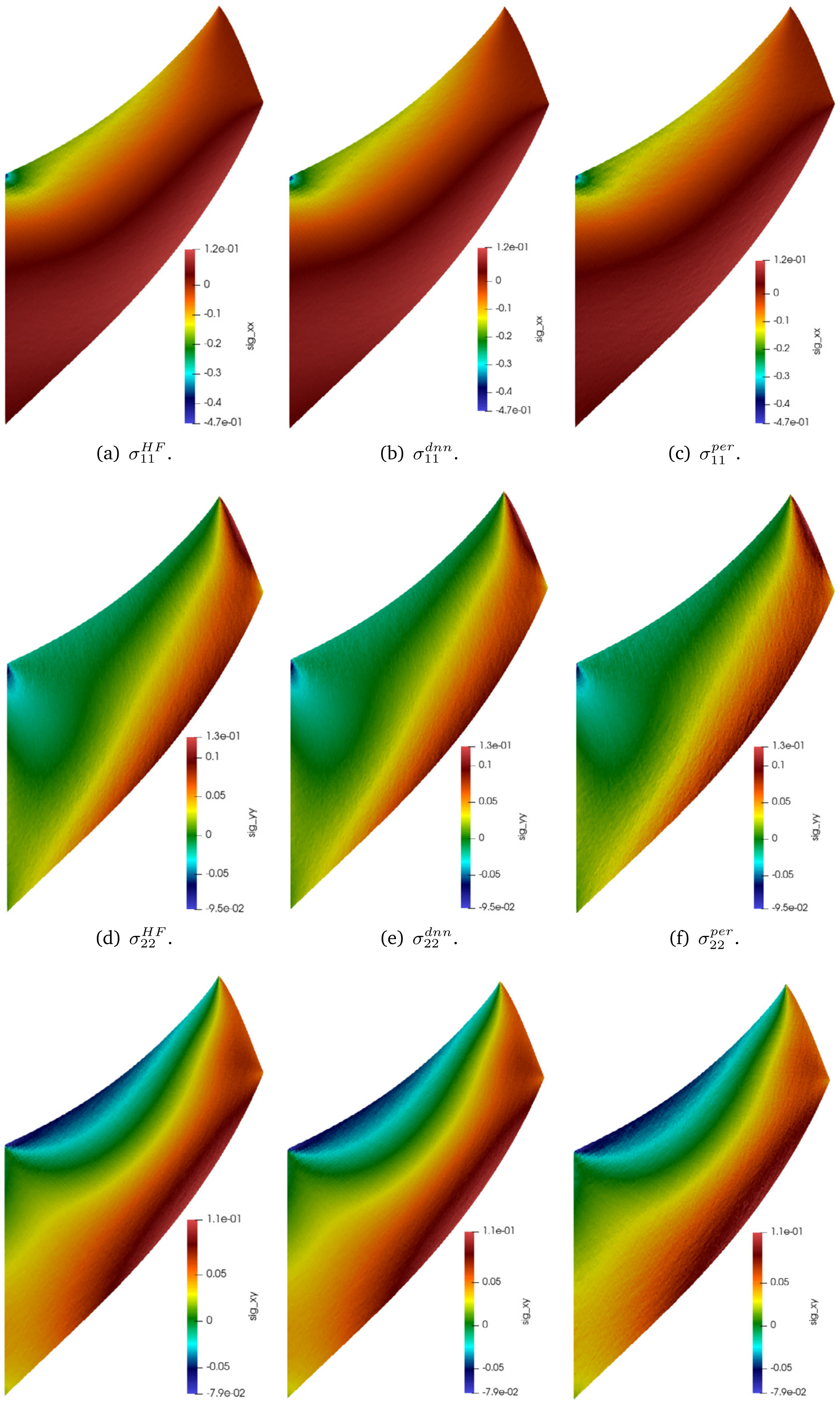
(d) $\frac{|\sigma_{vm}^{per} - \sigma_{vm}^{ref}|}{\sigma_{vm}^{ref}}$.

Fig. 16. Comparing DeepBND and periodic simulations in terms of absolute and relative errors of the Von Mises Stress (σ_{vm}) for the Cook membrane.

6.2. Clamped bar using DNS

The last test of the new multiscale methodology aims at assessing its performance in terms of accuracy and computational cost with respect to other multiscale methods in comparison to direct numerical simulations (DNS). In this second numerical example, for easiness of microstructures constructions, we adopt a rectangular bar $\Omega = [0, L_x = 4] \times [0, L_y = 1]$ as macroscopic domain such that it is clamped on the left and loaded with a uniform shear on the right $\mathbf{t} = (0, -0.2)$, as in Fig. 18. The material is organised by a $N_x^{DNS} \times N_y^{DNS}$ grid of $H^{DNS} = L_x/N_x^{DNS} = L_y/N_y^{DNS}$ squared blocks in which each block contains a random sized circular inclusion. The mesh for the DNS simulations is generated with the characteristic length $h \approx H^{DNS}/15$, which is similar to the one used in the dataset generation. We consider $N_y^{DNS} \in \{24, 72\}$, to study two different approximations with respect to the DNS, which is expected to be more suitable for larger N_y^{DNS}. Just for visualisation purposes, we show in Fig. 19 one realisation of a structure with $N_y^{DNS} = 7$. Moreover, it is worth remarking that the mesh generation for larger N_y^{DNS} becomes cumbersome and computationally costly. Finally, it is also worth mentioning that the bar geometric dimensions were inspired by [24], however as in that reference the results presented were merely visual, we decided to compare DeepBND against HF and periodic models only.

Concerning the multiscale simulations, we use a regular triangular mesh with 100 divisions vertically and 400 horizontally. Notice that this mesh is several orders of magnitude coarser than the DNS, which makes the FE^2 simulations still



(a) σ_{11}^{HF}. (b) σ_{11}^{dnn}. (c) σ_{11}^{per}.

(d) σ_{22}^{HF}. (e) σ_{22}^{dnn}. (f) σ_{22}^{per}.

(g) σ_{12}^{HF}. (h) σ_{12}^{dnn}. (i) σ_{12}^{per}.

Fig. 17. Comparison Cauchy stress components in the Cook membrane.

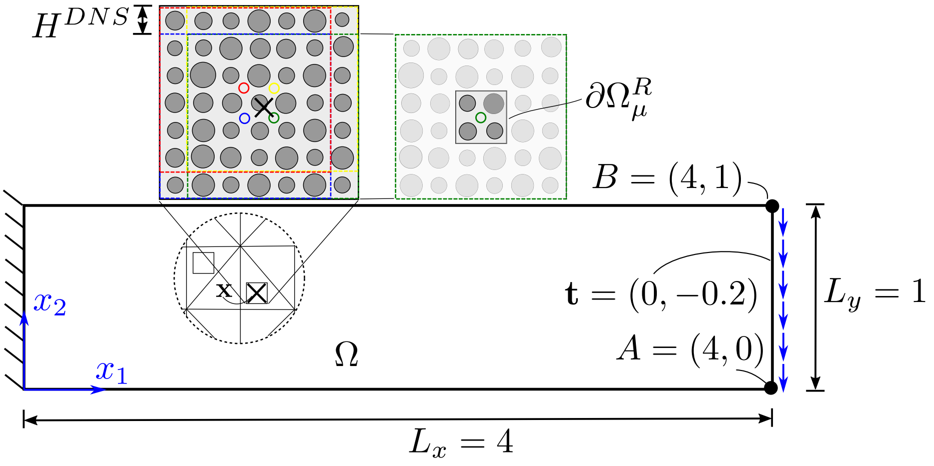


Fig. 18. Clamped bar scheme showing geometric dimensions and loads. The integration point is represented by (×) and underlying HF cell centres by circles. Colours indicate the correspondence between the centres and respective window.

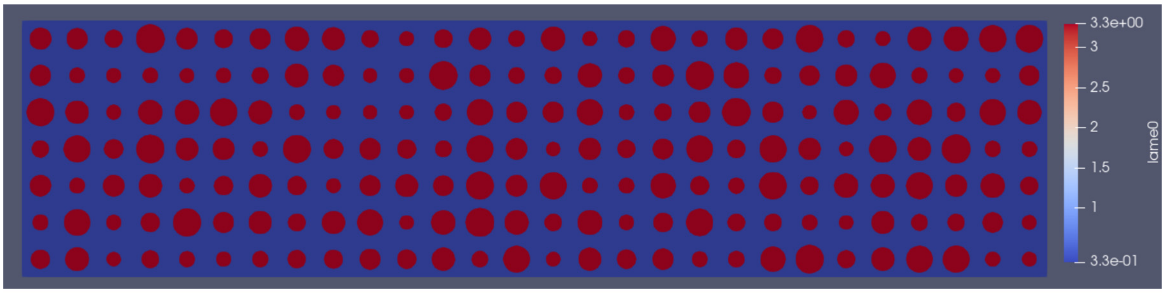


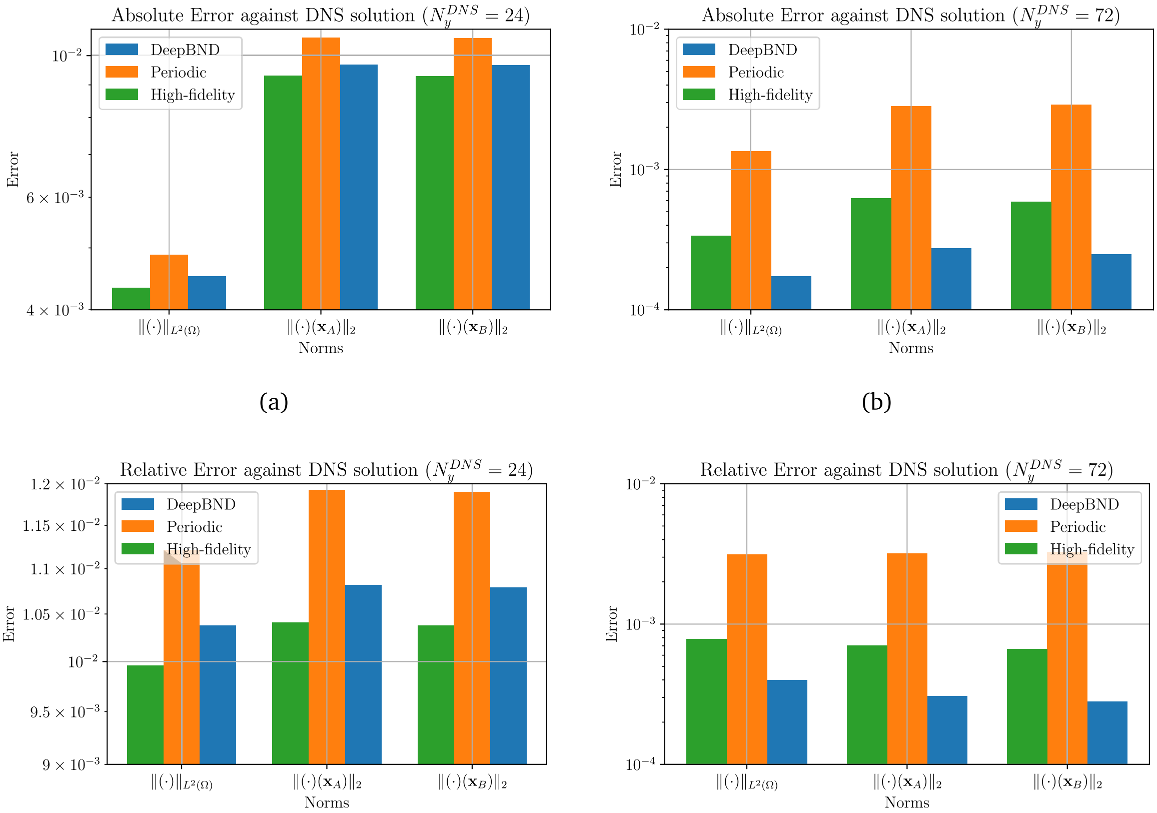
Fig. 19. Realisation of bar with $N_y^{DNS} = 7$ and $N_x^{DNS} = 28$.

appealing, despite its also elevated cost. As usual, one MD should be associated at each integration point. Differently from the previous case (Cook's Membrane, Section 6.1), now the microstructures have to keep some relation with its neighbourhood. Given an integration point, we choose the associated MD as the 6 × 6 window having the closest centre point, and then we proceed as before by identifying the reduced 2 × 2 window and predicting the BCs. This strategy is depicted in Fig. 18, where the dashed windows are associated with the respective centre (circles) of the same colour. Notice that in total the number of different MDs are $(N_x^{DNS} - 5) \times (N_y^{DNS} - 5)$, which makes 1729 and 18961 for $N_y^{DNS} = 24$ and $N_y^{DNS} = 72$, respectively.

Concerning the results, each structure has been solved using quadratic polynomial elements for the DNS mesh (DNS solution) and using the multiscale approach, with the MDs selected as aforementioned, for the DeepBND, periodic, and the HF cases. The latter cases have been compared with the DNS solution with respect to the L^2 norm, and the Euclidean norm for displacements at points A and B (see Fig. 18). We can see in Fig. 20(c) that relative errors for the $N_y^{DNS} = 24$ case are of order 1%, with slight advantages for the HF and DeepBND contrasted with the periodic case. On the other hand, for the $N_y^{DNS} = 72$ case, DeepBND model reaches 0.05% in relative error for the $L^2(\Omega)$-norm while the periodic model attains 0.2% in the same norm, i.e., a factor 4 of improvement. Similar gains were found for the point-wise displacement norms. Unexpectedly, comparing with the HF case, DeepBND performs even better, which shows that our approach does not only interpolate the HF targets, in the sense of averaging results taking close inputs, but it is also capable of learning more general relations associated to the whole assemble of data.

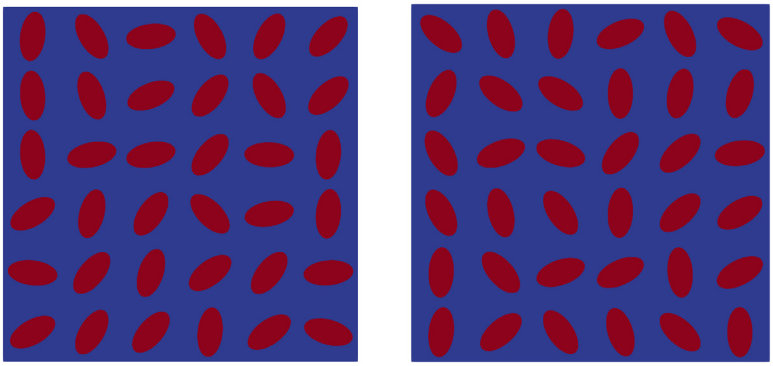
6.3. Cook membrane with different microstructural patterns

In the previous examples we have specialised the method framework for the case of microstructures composed of circular inclusions. The present example aims at showing that the DeepBND method can be seamlessly applicable to other kinds of microstructures, in our case: i) ellipsoidal inclusions and ii) misaligned circular inclusions. Such microscopic constitutive law parametrisations follow analogously as in Section 3.1.1. To shorten notation, we refer to Ellipsoidal Inclusions Dataset (EID) and Misaligned Circular Inclusions Dataset (MCID) for the datasets generated using two different parametrisations, in contrast with Circular Inclusions Dataset (CID) used in the previous examples. Note that in terms of BCs homogeneity,



(a) (b) (c) (d)

Fig. 20. Comparison errors for DNS solution as reference for the clamped bar problem.



(a) $\mathbf{p}_c^{(0)} \in \mathbb{P}^{(1)}$. (b) $\mathbf{p}_c^{(100)} \in \mathbb{P}^{(1)}$.

Fig. 21. Examples of ellipsoidal inclusions microstructures.

the EID and MCID give rise to less homogeneous fluctuations on the boundary when compared to CID, yielding to a more challenging situation for the method.

6.3.1. Ellipsoidal inclusions

Here, we have considered randomly oriented ellipsoidal inclusions of constant shapes, arranged in a 6×6 grid, as depicted in Fig. 21. As for the random generation, we have sampled the orientation of each ellipse uniformly in $[-\pi/2, \pi/2]$, also using the Latin Hypercube Sampling (LHS). To keep the number of parameters compared with the CID, we have fixed

Table 6
Relative errors in the finest mesh for the case of elliptical inclusions.

	DeepBND	Periodic
Vert. Disp. A	$1.12 \times 10^{-3} \pm 4.27 \times 10^{-5}$	$2.32 \times 10^{-3} \pm 9.18 \times 10^{-4}$
Von Mises B	$5.22 \times 10^{-4} \pm 1.17 \times 10^{-4}$	$1.03 \times 10^{-2} \pm 1.80 \times 10^{-3}$
Von Mises C	$2.84 \times 10^{-4} \pm 1.97 \times 10^{-4}$	$3.82 \times 10^{-3} \pm 1.98 \times 10^{-3}$
Von Mises D	$3.67 \times 10^{-4} \pm 6.68 \times 10^{-5}$	$1.38 \times 10^{-3} \pm 1.26 \times 10^{-3}$

$0.141 L_\mu^H$ and $0.0707 L_\mu^H$ as the major- and minor-axis sizes, respectively. These choices lead to a volume fraction close to mean values of CID (see Fig. 8) and ellipses with sufficient eccentricities to yield to a reasonably level of anisotropic response but also in such a way that the mesh is not degenerated. Hence, since only one parameter per ellipse is random, i.e., its orientation, the resulting parameter space is 36-dimensional. As the dimensions of the CID and EID parameter spaces are the same, we create datasets with the same number of samples accordingly to Table 2.

Concerning the DNN model, given the same size of databases and the supposedly similar complexity of the nonlinear mapping between geometric features and RB projections, we have used similar NN architectures regarding the hidden and output layers parts as in Section 5 (see comments in next paragraph concerning the input layer). Also, same analogy applies for the regularisation and optimisation algorithm employed. Moreover, the aforementioned setting is applied for both axial and shear modes submodels in the framework of Fig. 6. For the training of 5000 epochs, we have attained validation DNN errors of 3.23×10^{-5} and 2.15×10^{-5}, for the axial and shear models, respectively, which are between 2 to 3 times higher than for CID for both cases (see Fig. 10(b) and 11(b) respectively).

Finally, we should mention two small changes in how the NN inputs are structured. Firstly, we have used the mapping $\theta_i \mapsto (\cos\theta_i, \sin\theta_i)$, for $i = 1, \dots, 36$, where θ_i is the orientation of the ith ellipse, yielding to 72-dimensional inputs (neurons). This change, which yields to automatically normalised values in $[-1, 1]$, has led to slightly better training errors when contrasted with the use of orientations directly as input. Secondly, the clockwise permutation $\boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}$ (see (31)) should be redefined to include the rotation as follows $\boldsymbol{\pi}'^{-1}_{\frac{\pi}{2}}(\mathbf{p}_c) := \boldsymbol{\pi}_{\frac{\pi}{2}}^{-1}(\mathbf{p}_c - \frac{\pi}{2})$. As for the output layer, we have taken 140 (number of RB) neurons, which is largely sufficient to surpass the point at which the DNN error becomes prevalent to the POD error (remember Fig. 12 for the CID which is qualitatively for EID).

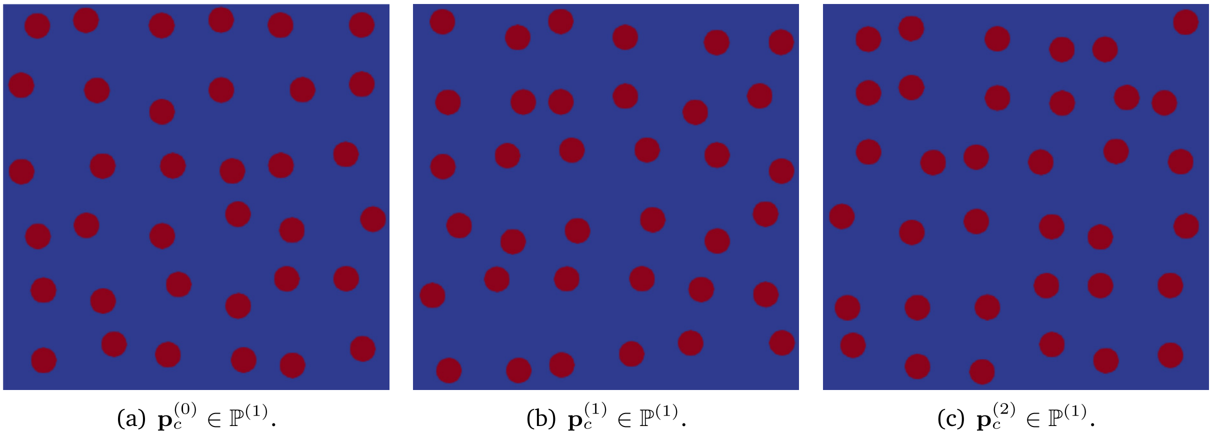
As for the numerical results, we measure relative errors for vertical displacement of point A and stresses at points B, C, and D compared to High-Fidelity solutions as shown in Table 6. For all metrics, DeepBND performs better than using periodic BCs, both for the mean and standard deviation values perspective. By comparing with the results for CID in Table 4, we have obtained slightly smaller accuracy gains for EID, which still accounts for improvements between 1 and 2 order of magnitude for stress, while for the vertical displacement, the error is halved. On the other hand, the results for coarser meshes display same qualitative behaviour as Fig. 14, therefore it is not repeated in the present section.

The slight decrease in performance can be attributed to the use of the same number of snapshots in the training database, and also to the choice of an almost identical DNN model and training, to a problem which is inherently slightly more complex, although the same number of parameters. Further tuning of the DNN model and training can be investigated to ameliorate such errors. Moreover, one can easily exploit data augmentation strategies by generating new snapshots at no cost by rotating one or more ellipses by an angle of π, i.e., by manipulating the input in multiple ways that leads to the same geometry and results. Notwithstanding, the DeepBND method has still been proved useful, by providing the same speed-ups as CID while the accuracy is considerably improved concerning naïve imposition of periodic boundary model.

6.3.2. Misaligned circular inclusions

For this example, we have generated randomly positioned circular inclusions of constant radius. More specifically, each MD has 36 randomly placed circular inclusions of radius $\frac{1}{30} L_\mu^H$. We have also constrained inclusions to be at least $\frac{1}{60} L_\mu^H$ apart from each other (inclusion-to-inclusion distance), and accordingly, such tolerance is halved to inclusions neighbouring the MD boundary. This is practically performed by uniformly sampling the coordinates (x and y) of each inclusion within disjoints squared subregions naturally matching these tolerances. For the sake of simplicity, for the example shown, we define the allowed subregions via a 6x6 grid, although other configurations are possible. Examples of microstructures are displayed in Fig. 22. Note that we have chosen a much smaller volume fraction of inclusions than CID and EID, allowing for larger ranges for moving inclusions without intersection.

As for the sampling, the above description yields 72-dimensional parametric input space, which was sampled using the LHS to generate 60000 snapshots for each loading scenario (axial tension and shearing). From the former dataset, training and validation partitions were also used to feed the training of a DNN model following the same guidelines of previous examples for architecture, regularisation, splitting, etc. For the training of 10000 epochs, we have attained validation DNN errors of 1.02×10^{-5} and 7.41×10^{-6}, for the axial and shear models, respectively, which are practically the same order of errors in the CID for both cases (see Fig. 10(b) and 11(b) respectively). As for the output layer, we have used 160 (number of RB) neurons, which is largely sufficient to surpass the point at which the DNN error becomes prevalent to the POD error (remember Fig. 12 for the CID which holds qualitatively for MCID). Finally, the clockwise permutation of input parameters that allows the cheap evaluation of the axial model in its perpendicular direction is easily achieved by rotation of coordinates.


(a) $\mathbf{p}_c^{(0)} \in \mathbb{P}^{(1)}$.

(b) $\mathbf{p}_c^{(1)} \in \mathbb{P}^{(1)}$.

(c) $\mathbf{p}_c^{(2)} \in \mathbb{P}^{(1)}$.

Fig. 22. Examples of misaligned circular inclusions microstructures.

Table 7
Relative errors in the finest mesh for the case of misaligned circular inclusions.

	DeepBND	Periodic
Vert. Disp. A	$1.37 \times 10^{-5} \pm 7.96 \times 10^{-7}$	$1.01 \times 10^{-4} \pm 3.50 \times 10^{-5}$
Von Mises B	$3.65 \times 10^{-5} \pm 4.33 \times 10^{-5}$	$1.45 \times 10^{-3} \pm 8.93 \times 10^{-4}$
Von Mises C	$2.66 \times 10^{-5} \pm 2.16 \times 10^{-5}$	$1.15 \times 10^{-3} \pm 8.12 \times 10^{-4}$
Von Mises D	$1.75 \times 10^{-5} \pm 8.67 \times 10^{-6}$	$6.51 \times 10^{-4} \pm 8.90 \times 10^{-4}$

Concerning the performance in the Cook's membrane problem, Table 7 shows relative errors at four selected points in terms of displacements and stresses (as used in the previous examples). We can see that DeepBND has improved performance with respect to periodic BCs by 1 to 2 orders of magnitude in all metrics. Moreover, by comparing with the results for CID in Table 4, we have obtained even more accurate results (by one order of magnitude), although the same tendency applies for the periodic case such that the relative gains are kept the same. The results shown are for the finest mesh tested, but the same conclusions were observed for coarser meshes.

Contrarily to the case EID case, in the MCID scenario we have observed more accurate results than in CID, even doubling the input parameter space. This suggests that the role played by the quality and density of sampling into the final accuracies is problem-dependent, such that the trade-off between number of parameters and dataset size is not the determinant for final accuracy. Our *a posteriori* interpretation is that as the volume fraction of inclusions is considerably lower and constant in this case, the mechanical response is less dominated by the hard inclusions, which leads to generally smaller errors, even though the microstructures have richer dispersion of material coefficients and we need more random variables to describe them.

Finally, it is also important to mention the strategy of subregions is important to create unequivocally geometrical meanings for the input parameters, i.e., each MD is associated to only one vector in the parameter space. Subregions can be freely placed in the MD and not necessarily should be distributed in a grid-like manner as in the example. Notwithstanding, the grid-like manner adopted is already sufficient to generate representative microstructures fibre reinforced material cross-sections, which a certain homogeneity in the fibre spatial density is desired.

7. Concluding remarks

In this work, we have introduced a novel ML-based technique, so-called DeepBND, to predict more suitable BCs for multiscale problems. Exploiting the combination between ROM-RB and DNN, our method delivers physically admissible BCs and can learn from previous HF simulations. As result, numerical examples have shown that DeepBND reaches improvements in accuracy up to two orders of magnitude, keeping the same computational cost, compared to classical methods. While the construction of the DeepBND model might be computationally expensive (offline phase), we reached speed-up gains of approximately 60 times in the real-use FE^2 case (online phase) when compared to HF simulations, which justifies the fixed additional effort. Indeed, as seen in Algorithm 2, DeepBND method only adds one new step to the online phase, namely the DNN model evaluation, which is negligible in terms of additional computational complexity concerning the standard methodologies. Consequently, DeepBND can be straightforwardly implemented in already existing FE^2 codes.

It is worth mentioning that our method presents clear advantages concerning FE^2 approaches with classical BCs and also those in which surrogate models replace the constitutive law. Concerning the former approach, as already discussed, the main asset is the computational speed-up by keeping similar accuracy. As for those using surrogate models, our method is naturally more precise since we still solve the physical problem. Indeed, DeepBND can be understood as an hybrid methodology, in the sense that it lies between the standard computational homogenisation with relatively large MDs and its complete replacement by surrogate models. Up to the best of authors' knowledge, our approach is unique in the literature. Our examples do not aim at generality but rather serve as a proof of concept that DeepBND is a viable technique to substantially increase accuracy versus computational effort ratio.

Finally, future developments should focus on extending the method to possible sharp parametric dependencies in the BCs descriptor, for example, in the case of composites where the inclusions cross the micro-cell boundary. As it is well known in the literature [30], RBs have poor performance for non-smooth parametric dependence and they should be replaced by another ROM with proven capabilities for dealing with this task, e.g., autoencoders. Moreover, less data-hungry alternatives to DNN, such as Gaussian processes, are worth exploring for larger problems, in particular tridimensional applications. Note that modifications in the ROM and ML methods are still covered, with minimal changes, by the generic framework set in this contribution.

As for the physics, it is worth remarking that although we have particularly focused on problems arising in multiscale solid mechanics, our method is sufficiently generic and remains essentially the same for other boundary value problems at the microscale. Other possible examples of applications are porous media with random porosity fields or fluid mechanics with random obstacles.

CRediT authorship contribution statement

Felipe Rocha: Conceptualization, Methodology, Software, Writing – original draft. **Simone Deparis:** Conceptualization, Funding acquisition, Writing – review & editing. **Pablo Antolin:** Conceptualization, Software, Writing – review & editing. **Annalisa Buffa:** Conceptualization, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported in part by the Swiss National Science Foundation under projects FNS-200021_197021 and 200021E-168311.

Appendix A. Multiscale modelling (additional details)

Here, for convenience, we provide some additional details concerning of the multiscale model of Section 2.1.2. For the interested reader, our presentation follows closely the Principle of Multiscale Virtual Power (PMVP) introduced in [8,7], which can be seen as a generalised version of the Hill-Mandel Principle [23].

To fully characterise kinematically admissible fluctuations, yielding to definition of the Minimally Constrained Space for fluctuations in (4), we need to consider two kinematical compatibility hypotheses below:

i) Compatibility of microscale displacements:

$$\mathbf{u} = \left\langle \mathbf{u}_\mu \right\rangle_{\Omega_\mu} . \tag{A.1}$$

The equivalent restriction on the fluctuation follows from (3) straightforwardly as

$$\begin{aligned} \left\langle \mathbf{u}_\mu \right\rangle_{\Omega_\mu} &= \langle \mathbf{u} \rangle_{\Omega_\mu} + \boldsymbol{\varepsilon} \, \langle \mathbf{y} \rangle_{\Omega_\mu} + \left\langle \tilde{\mathbf{u}}_\mu \right\rangle_{\Omega_\mu} \\ &= \mathbf{u} + \left\langle \tilde{\mathbf{u}}_\mu \right\rangle_{\Omega_\mu} , \end{aligned}$$

which yields

$$\left\langle \tilde{\mathbf{u}}_\mu \right\rangle_{\Omega_\mu} = \mathbf{0}.$$

ii) Compatibility of microscale strains:

$$\boldsymbol{\varepsilon} = \left\langle \boldsymbol{\varepsilon}_\mu(\mathbf{u}_\mu) \right\rangle_{\Omega_\mu} . \tag{A.2}$$

The equivalent restriction on the fluctuation follows from (3) and using standard vector calculus identities

$$\left\langle \boldsymbol{\varepsilon}_\mu(\mathbf{u}_\mu) \right\rangle_{\Omega_\mu} = \langle \boldsymbol{\varepsilon} \rangle_{\Omega_\mu} + \left\langle \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{u}}_\mu) \right\rangle_{\Omega_\mu} = \boldsymbol{\varepsilon} + \left\langle \tilde{\mathbf{u}}_\mu \otimes^s \mathbf{n} \right\rangle_{\partial\Omega_\mu},$$

which yields[3]

$$\left\langle \tilde{\mathbf{u}}_\mu \otimes^s \mathbf{n} \right\rangle_{\partial\Omega_\mu} = \mathbf{0},$$

where $\mathbf{n}$ is the outward unit vector along $\partial\Omega_\mu$.

Finally, to derive Problem 1, we enunciate the PMVP which states the virtual power balance between scales, as follows.

Problem 4 *(PMVP).* For a given $\boldsymbol{\varepsilon} \in \mathbb{R}^{d,d}_{sym}$, the pair $(\boldsymbol{\sigma}, \tilde{\mathbf{u}}_\mu) \in \mathbb{R}^{d,d}_{sym} \times V_\mu$ is such that

$$\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\varepsilon}} = \left\langle \boldsymbol{\sigma}_\mu \cdot \hat{\boldsymbol{\varepsilon}}_\mu \right\rangle_{\Omega_\mu}, \quad \forall \hat{\boldsymbol{\varepsilon}} \in \mathbb{R}^{d\times d}_{sym}, \tilde{\mathbf{v}}_\mu \in V_\mu, \tag{A.3}$$

where $\hat{\boldsymbol{\varepsilon}}_\mu = \hat{\boldsymbol{\varepsilon}} + \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{v}}_\mu)$ and $\boldsymbol{\sigma}_\mu = \mathbb{C}_\mu(\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}_\mu(\tilde{\mathbf{u}}_\mu))$. The tensors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ denote the macroscale strain and stress respectively.

As a result of the principle above we have the following variational consequences that constitute the microscale problem

1. Corrector problem: Taking $\hat{\boldsymbol{\varepsilon}} = \mathbf{0}$ in (A.3), it yields

$$\left\langle \boldsymbol{\sigma}_\mu \cdot \hat{\boldsymbol{\varepsilon}}_\mu(\tilde{\mathbf{v}}_\mu) \right\rangle_{\Omega_\mu} = 0, \quad \forall \tilde{\mathbf{v}}_\mu \in V_\mu,$$

which corresponds to (7).
2. Stress homogenisation: Taking $\tilde{\mathbf{v}}_\mu = \mathbf{0}$ in (A.3), it yields

$$\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\varepsilon}} = \left\langle \boldsymbol{\sigma}_\mu \cdot \hat{\boldsymbol{\varepsilon}} \right\rangle_{\Omega_\mu} = \left\langle \boldsymbol{\sigma}_\mu \right\rangle_{\Omega_\mu} \cdot \hat{\boldsymbol{\varepsilon}}, \quad \forall \hat{\boldsymbol{\varepsilon}} \in \mathbb{R}^{d\times d}_{sym},$$
$$\Rightarrow \boldsymbol{\sigma} = \left\langle \boldsymbol{\sigma}_\mu \right\rangle,$$

which corresponds to (8).

Appendix B. Remark on the admissibility of $\mathcal{V}^{\mathcal{N}}_\mu$

Let us retake the discussion initiated in Remark 6. Note that the functions of a given RB are generally such that $\langle \boldsymbol{\xi}_i \otimes \mathbf{n} \rangle_{\partial\Omega^R_\mu} \neq \mathbf{0}$, for all $i = 1, \ldots, N_{rb}$. Recalling the definition $V^{\mathcal{N}}_\mu$ in (27), from the former observation we have that $V^{\mathcal{N}}_\mu \not\subset V^M_\mu$, which renders Problem 3, just in principle, not consistent with (7). However, Problem 3 is still valid, since it derives from Problem 2 using well grounded variational arguments. To transform $V^{\mathcal{N}}_\mu(\mathbf{p})$ into a subset of V^M_μ, we should consider rearranging $\tilde{\mathbf{u}}^{\mathcal{N}}_\mu(\mathbf{p})$ as follows

$$\tilde{\mathbf{u}}^{\mathcal{N}}_\mu(\mathbf{p})(\mathbf{y}) = (\tilde{\mathbf{u}}^{\mathcal{N}}_\mu(\mathbf{p})(\mathbf{y}) - \left\langle \tilde{\mathbf{u}}^{\mathcal{N}}_\mu \otimes \mathbf{n} \right\rangle_{\partial\Omega^R_\mu} \mathbf{y}) + \left\langle \tilde{\mathbf{u}}^{\mathcal{N}}_\mu \otimes \mathbf{n} \right\rangle_{\partial\Omega^R_\mu}, \quad \text{for } \mathbf{y} \in \Omega^R_\mu. \tag{B.1}$$

Defining $\tilde{\boldsymbol{\varepsilon}}(\mathbf{p}) := \left\langle \tilde{\mathbf{u}}^{\mathcal{N}}_\mu(\mathbf{p}) \otimes \mathbf{n} \right\rangle_{\partial\Omega^R_\mu}$ and $(\tilde{\mathbf{u}}^{\mathcal{N}}_\mu)'(\mathbf{p}) := \tilde{\mathbf{u}}^{\mathcal{N}}_\mu(\mathbf{p})(\mathbf{y}) - \tilde{\boldsymbol{\varepsilon}}(\mathbf{p})\mathbf{y}$, we can see that $(\tilde{\mathbf{u}}^{\mathcal{N}}_\mu)' \in V^{ZA}_\mu$ and satisfies $\left\langle (\tilde{\mathbf{u}}^{\mathcal{N}}_\mu)'(\mathbf{p}) \otimes \mathbf{n} \right\rangle_{\partial\Omega^R_\mu} = \mathbf{0}$, so $(\tilde{\mathbf{u}}^{\mathcal{N}}_\mu)' \in V^M_\mu$ by construction. Applying the very same idea to each of RB $\boldsymbol{\xi}_i$, $i = 1, \ldots, N_{rb}$, we define the auxiliary basis $\mathcal{B}'_{N_{rb}} = \{\boldsymbol{\xi}'_i := \boldsymbol{\xi}_i - \langle \boldsymbol{\xi}_i \otimes \mathbf{n} \rangle_{\partial\Omega^R_\mu} \mathbf{y} \in L^2(\partial\Omega^R_\mu)\}$, yielding to

$$(V^{\mathcal{N}}_\mu)^*(\mathbf{p}) = \left\{ \boldsymbol{\eta} \in V^M_\mu;\, T_{\partial\Omega^R_\mu} \boldsymbol{\eta} = \sum_{i=1}^{N_{rb}} [\mathcal{N}(\mathbf{p})]_i \boldsymbol{\xi}'_i \right\} = V^L_\mu + (\tilde{\mathbf{u}}^{\mathcal{N}}_\mu)', \tag{B.2}$$

such that $(V^{\mathcal{N}}_\mu)^*(\mathbf{p}) \subset V^M_\mu$. It is also useful to rewrite (3) as follows

$$\mathbf{u}_\mu(\mathbf{y}) = \mathbf{u} + \boldsymbol{\varepsilon}\mathbf{y} + \tilde{\mathbf{u}}^0_\mu + \tilde{\mathbf{u}}^{\mathcal{N}}_\mu = \mathbf{u} + (\boldsymbol{\varepsilon} + \tilde{\boldsymbol{\varepsilon}})\mathbf{y} + \tilde{\mathbf{u}}^0_\mu + (\tilde{\mathbf{u}}^{\mathcal{N}}_\mu)', \tag{B.3}$$

[3] This constraint is valid if and only if the MD has no voids crossing $\partial\Omega_\mu$. On the contrary, a generalisation for this condition only accounting for integrals on the solid part of the boundary is necessary [36].

where the dependence on $\mathbf{p}$ has been omitted for the sake of simplicity. Note that the final product in the process of converting $V_\mu^{\mathcal{N}}(\mathbf{p}) \not\subset V_\mu^M$ to $(V_\mu^{\mathcal{N}}(\mathbf{p}))^* \subset V_\mu^M$ is an additional term added into the homogenised macroscale strain. The variational formulation of Problem 3 should be modified accordingly, resulting in the problem: find $\tilde{\mathbf{u}}_\mu \in (V_\mu(\mathbf{p}))^*$ such that

$$a'(\mathbf{p}; \tilde{\mathbf{u}}_\mu, \mathbf{v}) = b'(\mathbf{p}; \mathbf{v}) \quad \forall \mathbf{v} \in V_\mu^L, \tag{B.4}$$

where

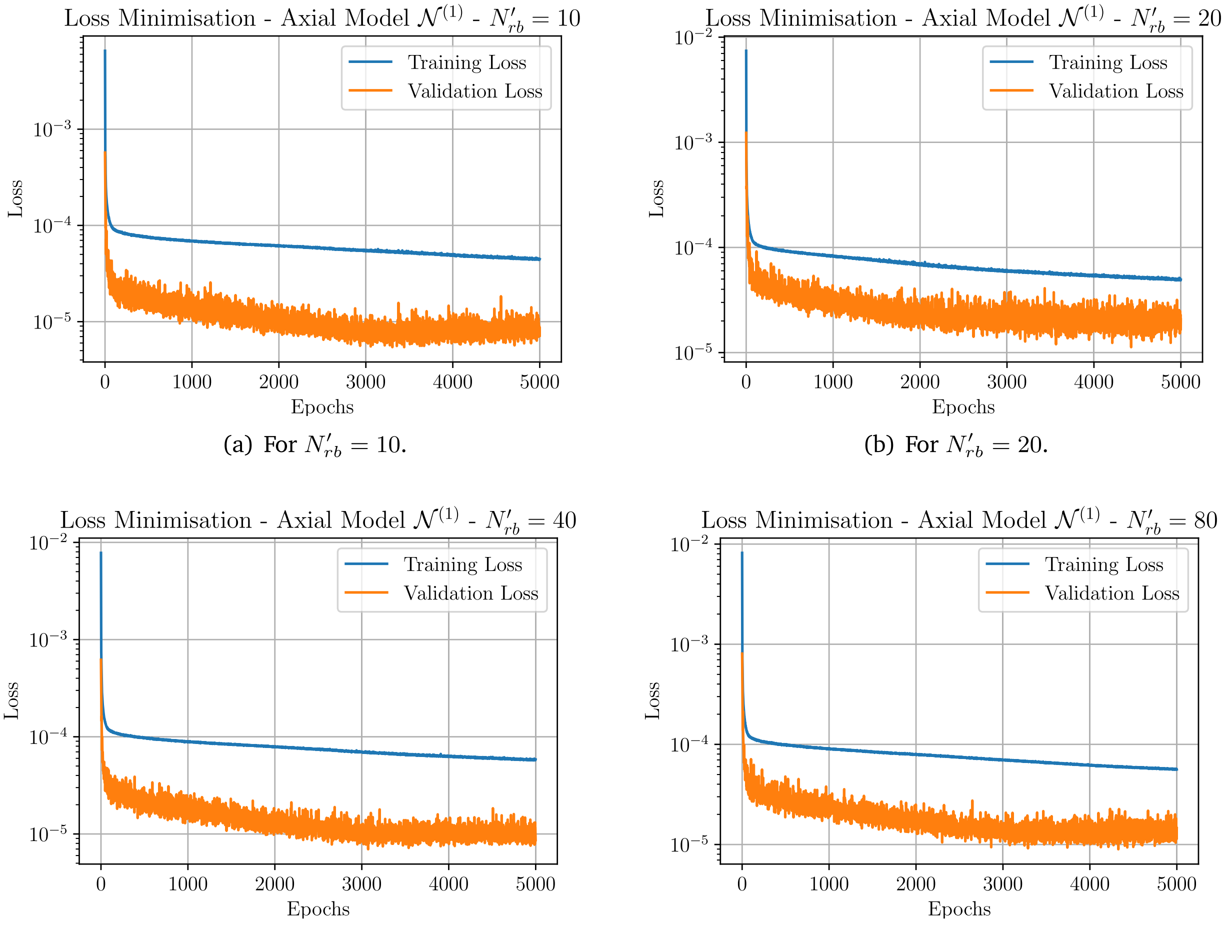
$$a'(\mathbf{p}; \tilde{\mathbf{u}}_\mu, \mathbf{v}) = (\mathbb{C}_\mu(\mathbf{p})\nabla^s \tilde{\mathbf{u}}_\mu, \nabla^s \mathbf{v})_{L^2(\Omega_\mu^R)} = a^R(\mathbf{p}; \tilde{\mathbf{u}}_\mu, \mathbf{v}), \tag{B.5a}$$

$$b'(\mathbf{p}; \mathbf{v}) = -(\mathbb{C}_\mu(\mathbf{p})(\boldsymbol{\varepsilon} + \tilde{\boldsymbol{\varepsilon}})(\mathbf{p}), \nabla^s \mathbf{v})_{L^2(\Omega_\mu^R)} = b^R(\mathbf{p}; \mathbf{v}) - (\mathbb{C}_\mu(\mathbf{p})\tilde{\boldsymbol{\varepsilon}}(\mathbf{p}), \nabla^s \mathbf{v})_{L^2(\Omega_\mu^R)}. \tag{B.5b}$$

In practical terms, Problem 3 can be still solved with no modifications. The homogenised stress also remains unchanged since the final microscale displacements in both formulations are the same by comparing both sides of (B.3).

Appendix C. Additional plots (training)

Here we provide, for the sake of completeness, additional plots concerning the training of the different architectures of Section 5 (see Figs. C.23 and C.24).



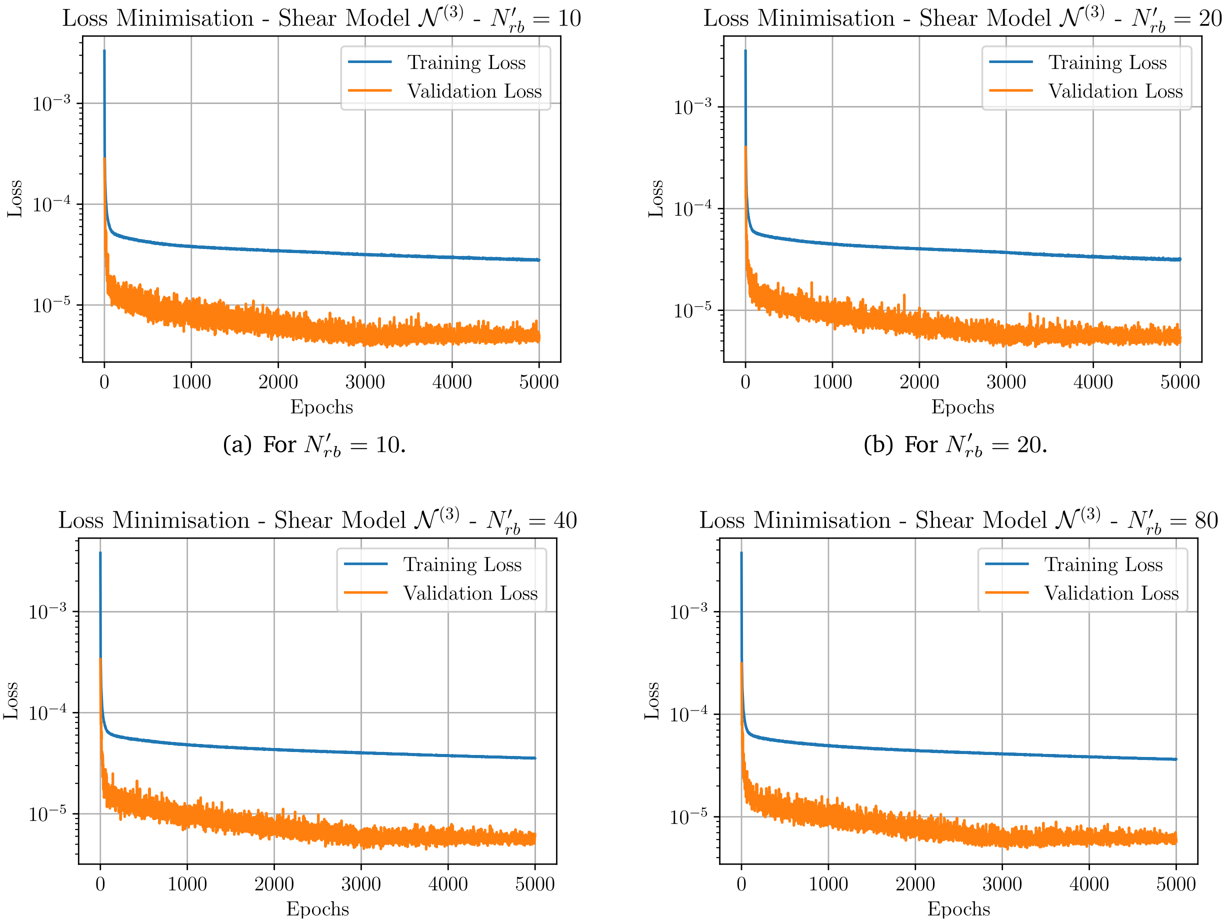
(a) For $N'_{rb} = 10$.

(b) For $N'_{rb} = 20$.

(c) For $N'_{rb} = 40$.

(d) For $N'_{rb} = 80$.

Fig. C.23. Historic for the loss function minimisation in Axial model $\mathcal{N}^{(1)}$. Training loss evaluated with regularisation (dropout and l^2), while validation loss in prediction mode (regularisation disabled).



(a) For $N'_{rb} = 10$.

(b) For $N'_{rb} = 20$.

(c) For $N'_{rb} = 40$.

(d) For $N'_{rb} = 80$.

Fig. C.24. Historic for the loss function minimisation in Shear model $\mathcal{N}^{(3)}$. Training loss evaluated with regularisation (dropout and l^2), while validation loss in prediction mode (regularisation disabled).

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, Tensorflow: A System for Large-Scale Machine Learning, 2016.

[2] A. Abdulle, D. Arjmand, E. Paganoni, Exponential decay of the resonance error in numerical homogenization via parabolic and elliptic cell problems, C. R. Math. 357 (2019) 545.

[3] A. Abdulle, B. Engquist, E. Vanden-Eijnden, The heterogeneous multiscale method, Acta Numer. 21 (2012) 1–87.

[4] The authors. Scikit-optimize: sequential model-based optimization in python, https://scikit-optimize.github.io/stable/, 2020.

[5] The authors. Multiphenics - easy prototyping of multiphysics problems in fenics, https://mathlab.sissa.it/multiphenics, 2021.

[6] A. Bensoussan, J.-L. Lions, G. Papanicolaou, Asymptotic Analysis for Periodic Structures, North-Holland, Amsterdam, 1978.

[7] P.J. Blanco, P.J. Sánchez, E.A. de Souza Neto, R.A. Feijóo, Variational foundations and generalized unified theory of RVE-based multiscale models, Arch. Comput. Methods Eng. 23 (2016) 191–253.

[8] E.A. de Souza Neto, P.J. Blanco, P.J. Sánchez, R.A. Feijóo, An RVE-based mutiscale theory of solids with micro-scale inertia and body force effects, Mech. Mater. 80 (2015) 136–144.

[9] B. Eidel, A. Fischer, The heterogeneous multiscale finite element method for the homogenization of linear elastic solids and a comparison with the fe2 method, Comput. Methods Appl. Mech. Eng. 329 (2018) 332–368.

[10] F. Feyel, A multilevel finite element method (fe2) to describe the response of highly non-linear structures using generalized continua, Comput. Methods Appl. Mech. Eng. 192 (2003) 3233.

[11] A. Fischer, B. Eidel, Convergence and error analysis of FE-HMM/FE^2 for energetically consistent micro-coupling conditions in linear elastic solids, Eur. J. Mech. A, Solids 77 (2019) 103735.

[12] Felix Fritzen, Mauricio Fernández, Fredrik Larsson, On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling, Front. Mater. 6 (2019).

[13] J.F. Geuzaine, C. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing, Int. J. Numer. Methods Eng. 79 (2009).

[14] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[15] I. Gühring, G. Kutyniok, P. Petersen, Error bounds for approximations with deep ReLU neural networks in $w^{s,p}$ norms, Anal. Appl. 18 (2020) 803–859.

[16] Z. Hashin, S. Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials, J. Mech. Phys. Solids 11 (1963) 127.

[17] J.S. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55.

[18] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer Briefs in Mathematics, Springer International Publishing, 2015.

[19] C.F. Higham, D.J. Higham, Deep learning: an introduction for applied mathematicians, SIAM Rev. 61 (2019) 860–891.

[20] R. Hill, A self-consistent mechanics of composite materials, J. Mech. Phys. Solids 13 (4) (1965) 213–222.

[21] G. Kutyniok, P. Petersen, M. Raslan, R. Schneider, A theoretical analysis of deep neural networks and parametric PDEs, Constr. Approx. 55 (2022) 73–125.

[22] Vincent Lucas, Jean-Claude Golinval, Stéphane Paquay, Van Dung Nguyen, Ludovic Noels, Ling Wu, A stochastic computational multiscale approach; application to mems resonators, Comput. Methods Appl. Mech. Eng. 294 (2015).

[23] J. Mandel, Plasticité classique et viscoplasticité, Springer-Verlag, 1972.

[24] V.M. Nguyen-Thanh, L. Trong, K. Nguyen, T. Rabczuk, X. Zhuang, A surrogate model for computational homogenization of elastostatics at finite strain using high-dimensional model representation-based neural network, Int. J. Numer. Methods Eng. 121 (2020) 4811–4842.

[25] M. Ostoja-Starzewski, Material spatial randomness: from statistical to representative volume element, Probab. Eng. Mech. 21 (2006) 112.

[26] Pierre Pineau, Frédéric Dau, Subsampling and homogenization to investigate variability of composite material mechanical properties, Comput. Methods Appl. Mech. Eng. 241–244 (2012) 238–245.

[27] A. Pinkus, Approximation theory of the MLP model in neural networks, Acta Numer. 8 (1999) 143–195.

[28] D. Pivovarov, R. Zabihyan, J. Mergheim, K. Willner, P. Steinmann, On periodic boundary conditions and ergodicity in computational homogenization of heterogeneous materials with random microstructure, Comput. Methods Appl. Mech. Eng. 357 (2019) 112563.

[29] Dmytro Pivovarov, Julia Mergheim, Kai Willner, Paul Steinmann, Stochastic local fem for computational homogenization of heterogeneous materials exhibiting large plastic deformations, Comput. Mech. 69 (2) (2022) 467–488.

[30] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Springer, 2016.

[31] P. Ramachandran, B. Zoph, Q.V. Le Swish, A Self-Gated Activation Function, 2017.

[32] F. Regazzoni, L. Dedè, A. Quarteroni, Machine learning for fast and reliable solution of time-dependent differential equations, J. Comput. Phys. 397 (2019) 108852.

[33] F. Rocha, micmacsfenics: a FEniCs-based implementation of two-level finite element simulations (FE2) using computational homogenization, https://doi.org/10.5281/zenodo.7643777, 2023.

[34] F. Rocha, deepbnd: an implementation of the DeepBND method based on FEniCs and Tensorflow., https://doi.org/10.5281/zenodo.7644075, 2023.

[35] F. Rocha, P.J. Blanco, P.J. Sánchez, E.A. de Souza Neto, R.A. Feijóo, Damage-driven strain localisation in networks of fibres: a computational homogenisation approach, Comput. Struct. 255 (2021) 106635.

[36] F. Rocha, P.J. Blanco, P.J. Sánchez, R.A. Feijóo, Multi-scale modelling of arterial tissue: linking networks of fibres to continua, Comput. Methods Appl. Mech. Eng. 341 (2018) 740–787.

[37] S. Saeb, P. Steinmann, A. Javili, Aspects of computational homogenization at finite deformations: a unifying review from Reuss' to Voigt's bound, Appl. Mech. Rev. 68 (2016) 050801.

[38] E. Sanchez-Palencia, Homogenization Method for the Study of Composite Media, 1983.

[39] N. Dal Santo, S. Deparis, L. Pegolotti, Data driven approximation of parametrized PDEs by reduced basis and neural networks, J. Comput. Phys. 416 (2020) 109550.

[40] Dimitrios Savvas, George Stefanou, Manolis Papadrakakis, Determination of rve size for random composites with local volume fraction variation, Comput. Methods Appl. Mech. Eng. 305 (2016) 340–358.

[41] J. Schröder, T. Wick, S. Reese, P. Wriggers, R. Müller, S. Kollmannsberger, M. Kästner, A. Schwarz, M. Igelbüscher, N. Viebahn, H.R. Bayat, S. Wulfinghoff, K. Mang, E. Rank, T. Bog, D. D'Angella, M. Elhaddad, P. Hennig, A. Düster, W. Garhuom, S. Hubrich, M. Walloth, W. Wollner, C. Kuhn, T. Heister, A selection of benchmark problems in solid mechanics and applied mathematics, Arch. Comput. Methods Eng. 28 (2) (2021) 713–751.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958.

[43] Patrizia Trovalusci, Martin Ostoja-Starzewski, Maria Laura De Bellis, Agnese Murrali, Scale-dependent homogenization of random composites as micropolar continua, Eur. J. Mech. A, Solids 49 (2015) 396–407.

[44] Ling Wu, Van-Dung Nguyen, Laurent Adam, Ludovic Noels, An inverse micro-mechanical analysis toward the stochastic homogenization of nonlinear random composites, Comput. Methods Appl. Mech. Eng. 348 (2019) 97–138.

[45] X. Yue, W. E, The local microscale problem in the multiscale modeling of strongly heterogeneous media: effects of boundary conditions and cell size, J. Comput. Phys. 222 (2007) 556.