MOX-Report No. 68/2023

Nonlinear model order reduction for problems with microstructure using mesh informed neural networks

Vitullo, P.; Colombo, A.; Franco, N.R.; Manzoni, A.; Zunino, P.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

Nonlinear model order reduction for problems with microstructure using mesh informed neural networks

Piermario Vitullo[a], Alessio Colombo[b], Nicola Rares Franco[a], Andrea Manzoni[a], Paolo Zunino[a]

[a]*MOX, Department of Mathematics, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133, Milan, Italy*
[b]*Department of Civil and Environmental Engineering, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133, Milan, Italy*

Abstract

Many applications in computational physics involve approximating problems with microstructure, characterized by multiple spatial scales in their data. However, these numerical solutions are often computationally expensive due to the need to capture fine details at small scales. As a result, simulating such phenomena becomes unaffordable for many-query applications, such as parametrized systems with multiple scale-dependent features. Traditional projection-based reduced order models (ROMs) fail to resolve these issues, even for second-order elliptic PDEs commonly found in engineering applications. To address this, we propose an alternative nonintrusive strategy to build a ROM, that combines classical proper orthogonal decomposition (POD) with a suitable neural network (NN) model to account for the small scales. Specifically, we employ sparse mesh-informed neural networks (MINNs), which handle both spatial dependencies in the solutions and model parameters simultaneously. We evaluate the performance of this strategy on benchmark problems and then apply it to approximate a real-life problem involving the impact of microcirculation in transport phenomena through the tissue microenvironment.

Keywords: reduced order modeling, finite element approximation, neural networks, deep learning, embedded microstructure, microcirculation

1. Introduction

The repeated solution of differential problems required to describe, forecast, and control the behavior of a system in multiple virtual scenarios is a computationally extensive task if relying on classical, high-fidelity full order models (FOMs) such as, e.g., the finite element method, when very fine spatial grids and/or time discretizations are employed to capture the detailed behavior of the solution. The presence of multiple (spatial and/or temporal) scales affecting problem's data and its input parameters – such as, e.g., material properties or source terms included in the physical model – makes this task even more involved, and is a common issue whenever dealing with, e.g., biological models [1, 2, 3], structural mechanics [4, 5] as well as environmental flows [6, 7], just to make a few examples. Reduced order modeling techniques provide nowadays a wide set of numerical strategies to solve parametrized differential problems in a rapid and reliable way. For instance, in the case of problems where the microstructure is characterized by slender fibers immersed into a continuum, reduced order models based on dimensional reduction of the fibers have been successfully adopted, in the framework of mixed dimensional problem formulations, see, e.g., [8, 9, 10]. However, given the increasing complexity of the problems that need to be addressed, mixed dimensional formulations are no longer sufficient; indeed, they actually represent the starting point for a second level of model reduction that is addressed by our work.

Physics-based ROMs such as, e.g., the reduced basis (RB) method [11, 12], provide a mathematically rigorous framework to build ROMs involving a linear trial subspace, or reduced basis – obtained, e.g., through proper orthogonal decomposition (POD) on a set of FOM snapshots – to approximate the solution manifold, and a (Petrov-)Galerkin projection to derive the expression of the ROM. Provided the problem operators depend affinely on the input parameters, a suitable offline-online splitting ensures a very fast assembling and solution of the ROM during the online testing stage, once the linear subspace and the projected reduced operators have been computed and stored during the offline training stage.

The presence of involved, spatially-dependent, input parameters representing, e.g., diffusivity fields or distributed force fields within the domain where the problem is set, usually makes the offline-online splitting not so straightforward because of their high dimension and (potentially, highly) nonlinear nature of the parameter-to-solution map. To be consistent with the usual formulation of ROMs, we will refer, throughout the paper, to space-varying fields representing some problem inputs as to *input parameters*, despite these latter usually denote vectors of quantities. These features might impact on the reducibility of problems with microstructure with *linear* ROMs at least in two ways: *(i)* linear trial subspaces might have large dimension, thus not really reducing the complexity of the problem; *(ii)* classical hyper-reduction techniques, such as the (discrete) empirical interpolation method (DEIM) [13, 14, 15, 16] or the energy-conserving sampling and weighting (ECSW) method [17, 18], usually employed to speed up the ROM assembling, can also suffer from severe computational costs, and result in intrusive procedures.

To overcome these drawbacks, alternative data-driven methods can be used for the approximation of the RB coefficients without resorting to (Galerkin) projection. In these cases, the FOM solution is projected onto the RB space and the combination coefficients are approximated by means of a surrogate model, exploiting, e.g., radial basis function (RBF) approximation [19], polynomial chaos expansions [20], artificial neural networks (NNs) [21, 22, 23], or Gaussian process regression (GPR) [24, 25, 26, 27]. The high-fidelity solver is thus used only offline to generate the data required to build the reduced basis, and then to train the surrogate model. Non-intrusive POD-based approaches using RBFs to interpolate the ROM coefficients have been proposed in, e.g., [19, 28, 29]. In a seminal contribution, Hesthaven and Ubbiali [21] have instead employed NNs to build a regression model to compute the coefficients of a POD-based ROM, in the case of steady PDEs; a further extension to time-dependent nonlinear problems, i.e., unsteady flows, has been addressed in [30]; see also, e.g., [31, 32] for the use of alternative machine learning strategies to approximate the POD coefficients. We will refer hereon to this class of approaches as to *POD-NN* methods. POD with GP regression have been used to build ROMs by Guo et al. in [24] to address steady nonlinear structural analysis, as well as for time-dependent problems [25]. A detailed comparison among non-intrusive ROMs employing RBF, ANN, and GP regressions can be found in, e.g., [33]; see instead [34, 35] for an alternative use of NNs to perform regression in the context of multi-fidelity methods, capable of leveraging models with different fidelities. These latter may involve data-driven projection-based [36, 37] ROMs or more recently developed deep learning-based ROMs [38, 39, 40, 41]. In these latter cases, POD has been replaced by (e.g., convolutional) autoencoders to enhance dimensionality reduction of the solution manifold, relying on (deep) feedforward NNs to learn the reduced dynamics onto the reduced trial manifold.

Despite the advantages they provide compared with dense architectures – in terms of costs and size of the optimization problem to be solved during training – convolutional NNs cannot handle general geometries and they might become inappropriate as soon as the domain where the problem is set is not an hypercube, although some preliminary attempts to generalize CNN in this direction have recently appeared [42]. In the case of problems with microstructure, this issue usually arises when attempting at reducing the dimensionality of spatially distributed parameters using DNNs rather than POD as, e.g., the DEIM would do – provided a linear subspace built through POD is still employed to reduce the solution manifold. For this reason, in this paper we rely on Mesh-Informed Neural Networks (MINNs), a class of architectures recently introduced in [43] and specifically tailored to handle mesh based functional data. The driving idea behind MINNs is to embed hidden layers into discrete functional spaces of increasing complexity, obtained through a sequence of meshes defined over the underlying spatial domain. This approach leads to a natural pruning strategy, whose purpose is to inform the model with the geometrical knowledge coming from the domain. As shown through several examples in [43], MINNs offer reduced training times and better generalization capabilities, making them a competitive alternative to other operator learning approaches, such as DeepONets [44] and Fourier Neural Operators [45]. Our purpose is to employ MINNs to enable the design of sparse architectures aiming at feature extraction from space-varying parameters that define the problem's microstructure.

In this paper we propose a new strategy to tackle parametrized problems with microstructure, combining a POD-NN method to build a reduced order model in a nonintrusive way and MINNs to build a closure model capable of integrating in the resulting approximation the information coming from mesh based functional data, in order to avoid to deal with a very large number of POD modes in presence of complex microstructures. Such a problem arises, e.g., when describing oxygen transfer in the microcirculation by including blood flow and hematocrit transport coupled with the interstitial flow, oxygen transport in the blood and the tissue, described by the vascular-tissue exchange. The presence of microvasculature, described in terms of a (varying) graph-like structure within the domain, requires the use of a NN-based strategy to reduce the dimensionality of such data, thus calling into play MINNs, yielding a strategy we refer to as a POD-MINN method. To account for the neglected scales at the POD level, and correct the POD-MINN approximation to enhance its accuracy without further increasing its dimension, we equip the POD-MINN method with a closure model, ultimately yielding a strategy we refer to as a POD-MINN+ approximation.

The structure of the paper is as follows. In Sect. 2 we formulate the class of problems with microstructure we focus on in this paper, introducing their high-fidelity approximation by the finite element method, recalling how classical projection-based ROMs are formulated, and showing their main limitations. In Sect. 3 we describe how to take advantage of mesh-informed neural networks to tackle problems with microstructure, while in Sect. 4 we address the POD-MINN and the POD-MINN+ methods, showing results obtained in a series of simple numerical test cases. Finally, in Sect. 5 we consider the application of the proposed strategy to an oxygen transfer problem taking place in the microcirculation, and draw some conclusions in Sect. 6.

2. Formulation and approximation of PDEs with embedded microstructure

2.1. Problem setup

We start by describing, in general terms, the essential properties of a problem governed by a parametrized PDE affected by a microstructure. With the term *microstructure* we refer to some features, primarily of the forcing terms of the equations, that induce the coexistence of multiple characteristic scales in the solution. Despite this is a particular case of *multiscale* problem, developing ROMs for multiscale problems in the spirit of upscaling and/or numerical homogenization is not the scope of this work – this aspect has been addressed, at least under some simplifying assumptions, by several authors in the framework of reduced basis methods, see, e.g., [46, 47, 48]. Conversely, here we aim to develop a ROM that fully captures all the scales of the solution.

We restrict to steady problems governed by second order elliptic equations and we assume that the microstructure may influence the parameters of the operator, the forcing terms and the boundary conditions, whereas the domain is fixed and do not depend on parameters. Under these assumptions, in this paper we address parametrized PDEs of the form

$$\begin{cases} L_{\boldsymbol{\mu}} u_{\boldsymbol{\mu}} = f_{\boldsymbol{\mu}} & \text{in } \Omega, \\ B_{\boldsymbol{\mu}} u_{\boldsymbol{\mu}} = g_{\boldsymbol{\mu}} & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where the solution $u_{\boldsymbol{\mu}}$ depends on the parameter vector $\boldsymbol{\mu} = (\boldsymbol{\mu}_M, \boldsymbol{\mu}_m) \in \mathcal{P} = \mathcal{P}_M \times \mathcal{P}_m$, with $\dim(\mathcal{P}_M) = n_M$ and $\dim(\mathcal{P}_m) = n_m$. In other words, $u_{\boldsymbol{\mu}} = u(\boldsymbol{\mu})$. Both notations will be used in what follows, with a preference to the most compact one when the context allows it. By the distinction between *macroscale parameters* – denoted by $\boldsymbol{\mu}_M$ – and *microscale parameters* – denoted instead by $\boldsymbol{\mu}_m$ – we highlight that problem parameters satisfy a *scale separation property.* For example, in the biophysical application discussed at the end of this work, the physical parameters of the operator are affected by the macroscale parameter $\boldsymbol{\mu}_M$, namely $L_{\boldsymbol{\mu}} = L_{\boldsymbol{\mu}_M}$, while the geometry of the microstructure determines the forcing terms of the problem, that is, $f_{\boldsymbol{\mu}} = f_{\boldsymbol{\mu}_m}$ and $g_{\boldsymbol{\mu}} = g_{\boldsymbol{\mu}_m}$. As a consequence of these assumptions, we focus on the particular case of the abstract problem (1), that can be rewritten as follows,

$$\begin{cases} L_{\boldsymbol{\mu}_M} u_{\boldsymbol{\mu}} = f_{\boldsymbol{\mu}_m} & \text{in } \Omega, \\ B_{\boldsymbol{\mu}_m} u_{\boldsymbol{\mu}} = g_{\boldsymbol{\mu}_m} & \text{on } \partial\Omega. \end{cases} \tag{2}$$

In this particular context, we implicitly assume that the physical parameters belong to a low-dimensional space, whereas the geometry of the microstructure features high dimensionality – that is, $n_m \gg n_M$. Moreover, for the sake of presentation, in this section we assume that the operator $L_{\boldsymbol{\mu}_M}$ appearing in problem (2) is linear, although our methodology will also be applied to a nonlinear oxygen transfer problem, whose formulation is briefly described in Sect. 2.2.

2.2. An example of problem with microstructure inspired to microcirculation

We also address in this paper a mathematical model for oxygen transfer in the microcirculation on the basis of a comprehensive model described in [3, 49] that includes blood flow and hematocrit transport coupled with the interstitial flow, oxygen transport in the blood and the tissue, described by the vascular-tissue exchange. Indeed, our proposed methodology can be applied to any field described by the general microcirculation model, such as fluid pressure, velocity, oxygen concentration (or partial pressure), despite we only focus, in this paper, on the oxygen transport. The non-intrusive character of the POD-MINN (and POD-MINN+) method indeed allows us to approximate a single field, despite the coupled nature of the general model. Note that if we relied on a classical projection-based ROM we had to approximate all the variables simultaneously, thus requiring to face a much higher degree of complexity when constructing the ROM.

The general model describes the flow in two different domains, the tissue domain ($\Omega \subset \mathbb{R}^3$ with $\dim(\Omega) = 3$), where the unknowns are the fluid pressure p_t, the fluid velocity $\mathbf{v}_t$, and the oxygen concentration C_t, and the vascular domain ($\Lambda \subset \mathbb{R}^3$ with $\dim(\Lambda) = 1$), which is a metric graph describing a network of connected one-dimensional channels, where the unknowns are the blood pressure p_v, the blood velocity $\mathbf{v}_v$, and the vascular oxygen concentration C_v. The model for the oxygen transport employs the velocity fields $\mathbf{v}_v$ and $\mathbf{v}_t$ to describe blood flow in the vascular network and the plasma flow in the tissue. The governing equations of the oxygen transfer model read as follows:

$$\begin{cases} \nabla \cdot (-D_t \nabla C_t + \mathbf{v}_t \; C_t) + V_{max} \; \dfrac{C_t}{C_t + \alpha_t \; p_{m_{50}}} = \phi_{O_2} \, \delta_\Lambda & \text{on } \Omega \\ \pi R^2 \dfrac{\partial}{\partial s} \left(-D_v \dfrac{\partial C_v}{\partial s} + v_v \; C_v + v_v \; k_1 \; H \; \dfrac{C_v^\gamma}{C_v^\gamma + k_2} \right) = -\phi_{O_2} & \text{on } \Lambda \\ \phi_{O_2} = 2\pi R \; P_{O_2}(C_v - C_t) + (1 - \sigma_{O_2}) \left(\dfrac{C_v + C_t}{2} \right) \phi_v & \text{on } \Lambda \\ \phi_v = 2\pi R L_p \big((p_v - p_t) - \sigma(\pi_v - \pi_t) \big) & \\ C_v = C_{in} & \text{on } \partial\Lambda_{\text{in}} \\ -D_v \dfrac{\partial C_v}{\partial s} = 0 & \text{on } \partial\Lambda_{\text{out}} \\ -D_t \nabla C_t \cdot \mathbf{n} = \beta_{O_2} \; (C_t - C_{0,t}) & \text{on } \partial\Omega. \end{cases} \tag{3}$$

In particular, the first equation governs the oxygen in the tissue, the second describes how the oxygen is transported by the blood stream, and the third defines the oxygen transfer between the two domains Ω and Λ. In particular, the model for the flux ϕ_{O_2} is obtained assuming that the vascular wall acts as a semipermeable membrane. This model is complemented with a set of boundary conditions reported in the last three equations: at the vascular inlets $\partial\Lambda_{\text{in}}$ we prescribe the oxygen concentration; at the vascular endpoints $\partial\Lambda_{\text{out}}$ null diffusive flux is enforced; and for the boundary of the tissue domain $\partial\Omega$ we simulate the presence of an adjacent tissue domain with a boundary conductivity β_{O_2} and a far-field concentration $C_{0,t}$. The symbols $D_t, D_v, V_{max}, \alpha_t, p_{m_{50}}, k_1, k_2, C_v^\gamma, P_{O_2}, \sigma_{O_2}, L_p, \sigma, \pi_v, \pi_t$ are constants independent of the solution of the model. For a detailed description of the physical meaning of these quantities see, e.g., [49].

Comparing model (2) with (3), we observe that the operator $L_{\boldsymbol{\mu}_M}$ consists of the left hand side of the first equation, where the macroscale parameters are the physical parameters of the operator $\nabla \cdot (-D_t \nabla C_t + \mathbf{v}_t\ C_t) + V_{max} C_t/(C_t + \alpha_t\ p_{m_{50}})$, such as for example V_{max} that modulates the oxygen consumption by the cells in the tissue domain. Because of the last term, such operator is nonlinear. The solution $u_{\boldsymbol{\mu}}$ corresponds to the concentration C_t, and the flux $\phi_{O_2}\delta_\Lambda$ plays the role of the forcing term $f_{\boldsymbol{\mu}_m}$. As δ_Λ denotes a Dirac δ-function distributed on Λ, the term $\phi_{O_2}\delta_\Lambda$ represents an incoming mass flux for the oxygen concentration C_t, supported on the vascular network Λ. In other words, it is a concentrated source defined in Ω, acting as a microscale forcing term. According to the third equation, the intensity of the oxygen flux depends on both concentrations, namely $\phi_{O_2} = \phi_{O_2}(C_v, C_t)$. In turn, the vascular oxygen concentration C_v is governed by the second equation and by the inflow boundary conditions $C_v = C_{in}$ on $\partial\Lambda_{IN}$. This condition plays the role of $B_{\boldsymbol{\mu}_m} u_{\boldsymbol{\mu}} = g_{\boldsymbol{\mu}_m}$ on $\partial\Omega$ in problem (2), and as it depends on the inlet points of the vascular network on the domain boundary, it is classified as a microscale model. In conclusion, we can establish the following connections between problems (2) and (3):

$$\begin{aligned}
u_{\boldsymbol{\mu}} &\approx C_t, \\
L_{\boldsymbol{\mu}_M} u_{\boldsymbol{\mu}} &\approx \nabla \cdot (-D_t \nabla C_t + \mathbf{v}_t\ C_t) + V_{max}\ \frac{C_t}{C_t + \alpha_t\ p_{m_{50}}}, \\
f_{\boldsymbol{\mu}_m} &\approx \phi_{O_2}(C_v, C_t)\delta_\Lambda, \\
B_{\boldsymbol{\mu}_m} u_{\boldsymbol{\mu}} &\approx C_v|_{\partial\Lambda_{IN}}, \\
g_{\boldsymbol{\mu}_m} &\approx C_{in}|_{\partial\Lambda_{IN}}.
\end{aligned}$$

We remark that problem (3) is just an example among many other relevant problems characterized by a microstructure, such as flows through perforated or fractured domains, or the mechanics of fiber-reinforced materials, just to mention a few. Our proposed methodology does not depend on the specific problem and can be applied with suitable adaptations to all these problems thanks to its non-intrusive nature.

2.3. Numerical approximation by the finite element method

We consider the finite element method (FEM) as the high-fidelity FOM for the problem described in the previous section, as well as for the other test cases we propose. The central hypothesis underlying this work is that the FEM approximation of the microscale problem resolving all the scales of the microstructure is computationally too expensive for real-life applications. For the sake of simplicity, we present our methodology referring to the general, abstract problem (1). Before introducing its finite element discretization, we address the variational formulation of (1) that, given a suitable Hilbert space $(V, \|\cdot\|)$ depending on the boundary conditions imposed on the problem at hand, reads as follows: for any $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{n_p}$, find $u_{\boldsymbol{\mu}} \in V$ such that

$$a_{\boldsymbol{\mu}}(u_{\boldsymbol{\mu}}, v) = F_{\boldsymbol{\mu}}(v), \quad \forall v \in V, \tag{4}$$

being $a_{\boldsymbol{\mu}} : V \times V \mapsto \mathbb{R}$ and $F_{\boldsymbol{\mu}} : V \mapsto \mathbb{R}$ two parameter-dependent operators. Depending on the formulation of choice, Problems (1) and (4) naturally define a *parameter-to-solution map*, that is, a map that assigns to each input parameter $\boldsymbol{\mu} \in \mathcal{P}$ the corresponding solution $u_{\boldsymbol{\mu}} = u(\boldsymbol{\mu})$, *i.e.* ,

$$u_{\boldsymbol{\mu}} : \mathcal{P} \mapsto V; \quad \boldsymbol{\mu} \mapsto u(\boldsymbol{\mu}).$$

This also allows us to define the *solution manifold* $\mathcal{S} = \{u_{\boldsymbol{\mu}} = u(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\}$.

The high-fidelity FOM consists of the Galerkin projection of problem (4) onto a Finite Element (FE) space V_h of dimension $N_h = \dim(V_h)$, suitably chosen depending on the characteristics of the problem at hand. Assuming for simplicity a fully conformal approximation, given $V_h \subset V$, we aim to find $u_{\boldsymbol{\mu},h} \in V_h$ such that

$$a_{\boldsymbol{\mu}}(u_{\boldsymbol{\mu},h}, v_h) = F_{\boldsymbol{\mu}}(v_h) \qquad \forall v_h \in V_h. \tag{5}$$

As pointed out for the continuous case, problem (5) defines a mapping $\mathcal{P} \mapsto V_h$ that identifies the discrete solution manifold $\mathcal{S}_h = \{u_{\boldsymbol{\mu},h} : \boldsymbol{\mu} \in \mathcal{P}\}$. From the discrete standpoint, problem (5) is equivalent to a (large) system of algebraic equations of the form

$$A_{\boldsymbol{\mu},h}\mathbf{u}_{\boldsymbol{\mu},h} = \mathbf{F}_{\boldsymbol{\mu},h}$$

where $\mathbf{u}_{\boldsymbol{\mu},h} \in \mathbb{R}^{N_h}$ is the vector of degrees of freedom of the FE approximation. The need to, e.g., sample adequately the statistical variability of the microstructure, and assess its impact on the problem solution, would ultimately require to repeatedly query such FOM, whence the need of a rapid and reliable ROM.

2.4. Projection-based model order reduction

For second order elliptic PDEs, projection-based ROMs represent a consolidated approach that has been successful in many areas of application [50]. The main idea is to generate the ROM by projecting the FOM onto a low-dimensional linear subspace of V_h. Precisely, the *reduced basis* (RB) method aims at building a subspace $V_{rb} \subset V_h$ through the linear combination of a set of $n_{rb} \ll N_h$ basis functions, being n_{rb} independent of h and N_h:

$$V_{rb} = \text{span}\{\psi_1, \dots, \psi_{n_{rb}}\}.$$

The RB approximation $u^{rb}_{\boldsymbol{\mu},h} \in V_{rb}$ is then defined as the solution to the Galerkin projection of the FOM onto the RB space, that is,

$$a_{\boldsymbol{\mu}}(u^{rb}_{\boldsymbol{\mu},h}, v_{rb}) = F_{\boldsymbol{\mu}}(v_{rb}), \qquad \forall v_{rb} \in V_{rb}. \tag{6}$$

Equivalently, by expanding the RB solution over the reduced basis, the latter can be expressed as

$$u^{rb}_{\boldsymbol{\mu},h} = \sum_{i=1}^{n_{rb}} \mathrm{u}^{(i)}_{\boldsymbol{\mu},rb}\psi_i(x) \tag{7}$$

where $\mathbf{u}_{\boldsymbol{\mu},rb} = [\mathrm{u}^{(1)}_{\boldsymbol{\mu},rb}, \dots, \mathrm{u}^{(n_{rb})}_{\boldsymbol{\mu},rb}]^T$ are the *reduced coefficients*, which, in practice, are obtained by solving the algebraic counterpart of (6), namely

$$A_{\boldsymbol{\mu},rb}\mathbf{u}_{\boldsymbol{\mu},rb} = \mathbf{F}_{\boldsymbol{\mu},rb}. \tag{8}$$

Overall, equations (6), (7) and (8) illustrate different ways to represent the ROM approximation of the FOM solution. Precisely, $\mathbf{u}_{\boldsymbol{\mu},rb} \in \mathbb{R}^{n_{rb}}$ are the reduced coefficients that identify the ROM solution while $\mathbf{u}^{rb}_{\boldsymbol{\mu},h} \in \mathbb{R}^{N_h}$ is the representation of the same solution in the high-fidelity (FEM) space. We note that, although they have the same dimension, $\mathbf{u}_{\boldsymbol{\mu},h}$ and $\mathbf{u}^{rb}_{\boldsymbol{\mu},h}$ do not coincide.

We stress that this projection determines an approximation of the FOM solution $u_{\boldsymbol{\mu},h}$ onto a *linear subspace* of the discrete solution manifold $\mathcal{S}_h$. As a consequence the corresponding error behaves – at least, theoretically – following the decay of the Kolmogorov n_{rb}-width [12]. Indeed, a fast-decaying Kolmogorov n_{rb}-width reflects the approximability of the solution manifold by finite-dimensional linear spaces.

From an algebraic standpoint, the reduced basis is encoded in the matrix $\mathbb{V} = [\boldsymbol{\psi}_1 \mid \dots \mid \boldsymbol{\psi}_{n_{rb}}] \in \mathbb{R}^{N_h \times n_{rb}}$, where $\boldsymbol{\psi}_i \in \mathbb{R}^{N_h}$ are the FOM degrees of freedom of the i-th basis ψ_i, so that (7) can be equivalently rewritten as $\mathbf{u}^{rb}_{\boldsymbol{\mu},h} = \mathbb{V}\mathbf{u}_{\boldsymbol{\mu},rb}$ and the RB problem can be assembled projecting the FOM matrix and right hand side onto the reduced space,

$$A_{\boldsymbol{\mu},rb} = \mathbb{V}^T A_{\boldsymbol{\mu},h}\mathbb{V}, \quad \mathbf{F}_{\boldsymbol{\mu},rb} = \mathbb{V}^T\mathbf{F}_{\boldsymbol{\mu},h}. \tag{9}$$

For a special category of problems, characterized by an affine parameter dependence, the algorithm presented to build and solve (8) can be split into a parameter-independent *offline* and a parameter-dependent *online* stage [12]. However, if the affine parameter dependence does not hold, suitable hyper reduction techniques must be called into play to restore it, at least in an approximate way; nevertheless, this is not a focus of this work.

Regarding the construction of the subspace V_{rb}, a classical choice is to rely on POD, starting from a collection of FOM solutions, named *snapshots*, for suitably chosen parameters $\boldsymbol{\mu}^{(i)}$, such that the span of these functions, $\mathrm{span}\{u_h(\boldsymbol{\mu}^{(i)})\}_{i=1}^N$ is a reasonable approximation of the discrete solution manifold $\mathcal{S}_h$. By collecting the snapshots FOM degrees of freedom as the columns of a *data matrix*

$$\mathbb{S} = [\mathbf{u}_h(\boldsymbol{\mu}^{(1)})| \dots |\mathbf{u}_h(\boldsymbol{\mu}^{(i)})| \dots |\mathbf{u}_h(\boldsymbol{\mu}^{(N)})],$$

the singular value decomposition of $\mathbb{S}$,

$$\mathbb{S} = \mathbb{W} \begin{bmatrix} \widetilde{\mathbb{D}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbb{Z}^T$$

ensures the existence of three matrices $\mathbb{W} \in \mathbb{R}^{N_h \times N_h}$, $\widetilde{\mathbb{D}} = \mathrm{diag}[\sigma_1, \dots, \sigma_R]$, being $\sigma_1 \geq \dots \geq \sigma_i \geq \dots \geq \sigma_R$ the singular values of $\mathbb{S}$ and R is its rank, and $\mathbb{Z} \in \mathbb{R}^{N \times N}$. The columns of $\mathbb{W}$ and $\mathbb{Z}$ are named the left and right singular vectors, respectively. The POD basis is then defined as the set of the first n_{rb} left singular vectors of $\mathbb{S}$; their collection provides the matrix $\widetilde{\mathbb{W}}(n_{rb})$ that represents the *best* n_{rb}-rank approximation of $\mathbb{S}$ with respect to the following error [12],

$$E(\widetilde{\mathbb{W}}(n_{rb}); \mathbb{S}) = \min_{\mathbb{M} \in \mathbb{R}^{N_h \times N_h}} \sum_{j=1}^{N} \|\mathbb{S}(:, j) - \mathbb{M}\mathbb{M}^T \mathbb{S}(:, j)\|^2_{\mathbb{R}^{N_h}}.$$

In conclusion, a *POD-Galerkin* ROM consists of solving (8) using the matrix $\mathbb{V} = \widetilde{\mathbb{W}}(n_{rb})$ to define the reduced space.

2.5. Limitations of linear methods and nonlinear model order reduction

Despite mathematically rigorous, POD-Galerkin ROMs might feature some limitations when applied to problems with microstructure, as shown by some simple numerical test cases that we shall discuss in this section. As a prototype problem, we consider (1) with an operator independent on the parameters, *i.e.* $L_{\boldsymbol{\mu}} = -\Delta$, with homogeneous Dirichlet boundary conditions, and the action of the microstructure entirely carried out by the right hand side $f_{\boldsymbol{\mu}} = f_{\boldsymbol{\mu}_m}$. We consider a unit square domain $\Omega = (-1, 1)^2$ partitioned using a 50×50 computational mesh $\mathcal{T}_h$ of uniform triangular elements. For the sake of simplicity, we discretize the problem using piecewise linear, continuous, Lagrangian finite elements $V_h = X_h^1(\Omega)$, and consider two representative instances of the following characteristics of the microstructure: *(i)* a problem with continuously variable scales, and *(ii)* a problem with scale separation.

Problem with continuously variable scales. To model this feature of the microstructure, we consider

$$f_{\boldsymbol{\mu}_m}(x, y) = p(x)p(y), \quad \text{with} \quad p(z) = \sum_{k=1}^{n} (\alpha_k^z \sin(k\pi z) + \beta_k^z \cos(k\pi z))$$

and $\boldsymbol{\mu}_m = \{\alpha_k^z, \beta_k^z\}_{k=1}^n$ with $n = 6$ and $z = x, y$, for a total of 24 parameters modulating the linear combination of 144 trigonometric functions. For the application of the POD-Galerkin method to this test case, we sample the parameter space assuming a uniform probability distribution $\mathcal{U}[-1, 1]$ for all parameter, selecting a total amount of $N = 1000$ snapshots to build the data matrix $\mathbb{S}$. Equivalent results are obtained for a larger data matrix $\mathbb{S}$ made of 2000 snapshots– therefore, we keep $N = 1000$ as a reference value for the forthcoming tests.

Since we are interested to assess the capability of the POD basis to approximate the solution manifold, in this case we do not calculate the actual solution of (8), rather we consider the projection error obtained when projecting the FOM solution onto the POD basis for an increasing dimension n_{rb} of the reduced basis, that is,

$$E_{POD}(n_{rb}; \mathbf{u}_{\boldsymbol{\mu},h}) = \frac{\|(\mathbb{I} - \mathbb{V}(n_{rb})\mathbb{V}^T(n_{rb}))\mathbf{u}_{\boldsymbol{\mu},h}\|_{2,N_h}}{\|\mathbf{u}_{\boldsymbol{\mu},h}\|_{2,N_h}} \tag{10}$$

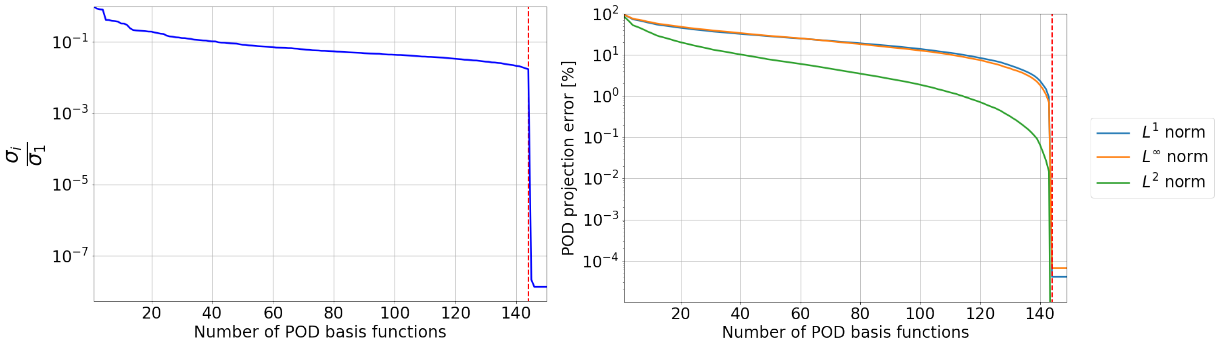


Figure 1: Decay of the singular values of $\mathbb{S}$ and the convergence rate of the projection error with respect to n_{rb} for a problem with continuously variable scales. The red line in the left panel denotes the threshold of $n = 144$ POD modes.

where $\mathbb{V}(n_{rb})$ denotes the matrix encoding the linear approximation space spanned by n_{rb} POD basis functions and $\|\mathbf{v}\|_{p,N} = \left(\sum_{i=1}^{N} |v_i|^p\right)^{1/p}$ is the p-norm in the generic vector space $\mathbb{R}^N$. We note that the solution of this problem depends linearly on the parameters, as a result if we apply the POD-Galerkin method with at least 144 basis functions, we shall represent exactly the discrete solution manifold – in other words, $E_{POD}(n_{rb} = 144; \mathbf{u}_{\boldsymbol{\mu},h}) = 0$ for any $\mathbf{u}_{\boldsymbol{\mu},h} \in \mathbb{R}^{N_h}$. Actually, the desired behavior of the ROM would be to achieve a satisfactory approximation with much less than 144 basis functions. The decay of the singular values of $\mathbb{S}$ and the convergence rate of the projection error with respect to n_{rb}, measured for a parameter value and corresponding finite element function not included in the snapshot matrix, are reported in Figure 1.

These results immediately show that, due to the slow decay of the singular values, the performance of the POD-Galerkin method cannot be satisfactory in this case.

Problem with scale separation. In this second test case, we discuss the behavior of the POD-Galerkin method for a problem where the forcing term is made of the superimposition of trigonometric functions with different frequencies, with a gap in the frequency spectrum. In the same computational setting described before, we consider the function

$$f(x,y) = f_{\boldsymbol{\mu}_M}(x,y) + f_{\boldsymbol{\mu}_m}(x,y), \quad \text{with} \;\; f_{\boldsymbol{\mu}_M}(x,y) = p_L(x)p_L(y),\, f_{\boldsymbol{\mu}_m}(x,y) = p_H(x)p_H(y)$$

and

$$P_L(z) = \sum_{k=1}^{2}(\alpha_k^z \sin(k\pi z) + \beta_k^z \cos(k\pi z)), \quad P_H(z) = \sum_{k=5}^{8}(\widetilde{\alpha}_k^z \sin(k\pi z) + \widetilde{\beta}_k^z \cos(k\pi z)),$$

for a total of 8 coefficients encoding the low frequency scales and 16 parameters for the high frequency scales, all sampled using a uniform probability function $\alpha_k^z, \beta_k^z, \widetilde{\alpha}_k^z, \widetilde{\beta}_k^z \sim \mathcal{U}[-1,1]$). These parameters modulate the forcing term made by the linear combination of 80 trigonometric modes.

The purpose of this test case is to investigate whether the POD basis with a number of entries comparable to the dimension of the low frequency source terms can approximate well the whole solution space. Figure 2 shows that the decay of the singular values reflects the gap in the frequency spectrum. However, after a reasonably good performance of the POD method in approximating the effect of the low frequencies on the solution, a large region of stagnation of the approximation properties can be highlighted.

Therefore, it is mandatory to include the effect of the microscale in the ROM approximation space, with a subsequent increase of the number of POD modes required to achieve a satisfactory approximation of the FOM.

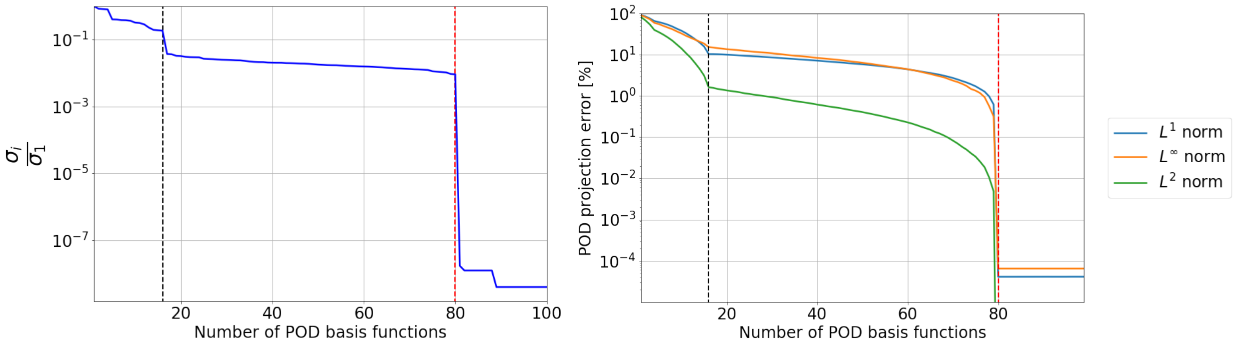


Figure 2: Decay of the singular values of $\mathbb{S}$ and the convergence rate of the projection error with respect to n_{rb} for a problem with scale separation. The vertical lines in the left panel denote the number of parameters of the low frequency function ($n = 16$) and the total number of parameters ($n = 80$).

2.5.1. Non intrusive, nonlinear reduced order modeling

The previous examples show that a Kolmogorov barrier arises due to the presence of the microstructure, this latter limiting the decay of the error achieved with projection-based ROMs that seek linear approximations in spaces. The main objective of this work is to overcome these limitations resorting to nonlinear model order reduction, implemented by combining a linear ROM approximation with nonlinear maps built through deep feedforward NNs, thus introducing a nonlinear, and non intrusive, ROM.

Nonlinear ROMs encompass a wide class of strategies aiming at replacing the linear approximation hypothesis encoded in (7) with more general, nonlinear maps. Rather than replacing the linear combination of POD modes with fully nonlinear approximations, as done, e.g., through deep learning-based ROMs [40, 38, 39], we express our approximation following the approach proposed in [51], as

$$\widetilde{u}^{rb}_{\boldsymbol{\mu},h} = \sum_{i=1}^{\widetilde{n}_{rb}} \widetilde{\mathrm{u}}^{(i)}_{\boldsymbol{\mu},rb} \widetilde{\psi}_i(x, \widetilde{\alpha}(\boldsymbol{\mu})), \tag{11}$$

where the main difference with respect to (7) is that the basis functions $\widetilde{\psi}_i$ now depend on the parameters of the problem through some *features*, named $\widetilde{\alpha}(\boldsymbol{\mu})$. Stated differently, the nonlinear approximation (11) is a linear combination of functions $\widetilde{\psi}_i(x, \widetilde{\alpha}(\boldsymbol{\mu}))$, $i = 1, \dots, \widetilde{n}_{rb}$ that depend through $\widetilde{\alpha}(\boldsymbol{\mu})$ on the element $u_{\boldsymbol{\mu},h}$ being approximated; this is indeed different from the linear approximation (7), where the basis functions are fixed and independent of which element $u_{\boldsymbol{\mu},h}$ of the (discrete) solution manifold is approximated. As we will see in the following, proceeding in this way can be extremely helpful in order to break the Kolmogorov barrier, leading to lower errors for the same number of degrees of freedom.

For the particular case of problems with microstructure, we put into action this general nonlinear ROM framework by using the nonlinear features as an additive correction, named *closure*, to the linear approximation obtained through (7). This consists in seeking a reduced approximation of the FOM based on the following representation,

$$\widetilde{u}^{rb}_{\boldsymbol{\mu},h} = \sum_{i=1}^{n_{rb}} \mathrm{u}^{(i)}_{\boldsymbol{\mu},rb} \psi_i(x) + \widetilde{\mathrm{u}}_{\boldsymbol{\mu},rb} \widetilde{\psi}(x, \widetilde{\alpha}(\boldsymbol{\mu})), \tag{12}$$

where the function $\widetilde{\psi}(x, \widetilde{\alpha}(\boldsymbol{\mu}))$ will be learned through artificial NNs (introduced in detail in the next section), mapping the parameter space into the full order FOM space, namely $\mathcal{N} : \mathcal{P} \to \mathbb{R}^{N_h} \equiv V_h$. In other terms, the linear approximation space is replaced by $\widetilde{V}_{rb} = V_{rb} + \mathcal{N}(\mathcal{P})$.

The presence of the microstructure poses however an additional challenge to the one of the Kolmogorov barrier discussed so far. Indeed, space-varying parameters deserve a similar dimensionality reduction as done for the state solution, given their dimension comparable to the number of high-fidelity FOM degrees of freedom. Moreover, the potentially involved nature of space-varying parameters makes the use of classical hyper reduction techniques – that would employ, e.g., a linear POD basis to approximate those data in order to comply with the ROM requirements – not feasible. For this reason, we resort to the use of NN-based approximations, exploiting a particular class of networks, the so-called *mesh informed neural networks* (MINNs) previously proposed by the authors [43], to handle mesh based functional data in a very efficient way; the resulting strategy, involving a POD-based dimensionality reduction for the state solution, and a MINN-based approximation of space-varying fields, will be referred to as POD-MINN method. A mesh informed architecture is also employed to approximate the ROM coordinates, similarly to the POD-NN approach [21], but extended here to the case of MINNs to handle the high-dimensional input data. The final approximation, obtained when the closure model is applied to the POD-MINN approximation, will be referred to as POD-MINN+.

In the next two sections we describe the formulation, the implementation details and the performance of these methodologies applied to problems with microstructure. Then, we will apply them to a real-life problem modeling that, despite some simplifying assumptions, is capable of representing the effect on microcirculation on the tissue microenvironment, involving multiple spatial scales and reaching very tiny detail levels.

3. Mesh-informed neural networks

Artificial neural networks (ANN) are parametrized functions between two (typically high-dimensional) vector spaces that have been recently employed also for the approximation of PDEs in several contexts. In this section, we discuss how they can be designed and exploited in particular case of problems with microstructure, to obtain non-intrusive and accurate ROMs. Before addressing mesh-informed neural networks, representing a key tool of our methodology, we briefly review feedforward neural networks for the sake of notation.

3.1. The architecture and the training of feedforward neural networks

ANNs are computational models obtained by stacking compositions of nonlinear transformations, acting on a collection of nodes called *neurons*, that are organized into building blocks called *layers*, these latter linked together through weighted connections. Given two vector spaces $V_m \subset \mathbb{R}^m$ and $\mathbb{R}^n$, a layer $\mathcal{L}$ that takes a vector in $\mathbb{R}^m$ as input and uses the *activation function* ρ, is a map of the form

$$\mathcal{L} : \mathbb{R}^m \to \mathbb{R}^n \text{ such that } \mathcal{L}(\mathbf{v}) = \rho(\mathbb{W}\mathbf{v} + \mathbf{b}),$$

where $\mathbb{W} \in \mathbb{R}^{n \times m}$ and $\mathbf{b} \in \mathbb{R}^m$ are the *weight matrix* and the bias of the layer, respectively. The activation function ρ, to be selected, is applied to the linear combination componentwise for each neuron of the same layer and ensures the nonlinearity of the approximation. The topology of the connections between the neurons determines the architecture of the ANN. The most common and known example is the *feedforward neural network*, defined as the composition of multiple layers $\mathcal{L}_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i+1}}$, with $i = 1, \ldots, l$, where $\mathcal{L}_{l+1}$ is the output layer and the remaining ones are called *hidden layers*. A feedforward neural network produces a map of the form $\mathcal{N} := \mathcal{L}_{l+1} \circ \mathcal{L}_l \ldots \circ \mathcal{L}_1$, so that, denoting with $\mathbb{W}_i$ the weight matrices between the i-th and the $i+1$-th layer, $\mathbf{b}_i$ the corresponding bias and ρ the activation function (generally the same for each layer):

$$\mathcal{N}(\mathbf{v}; \mathbb{W}_1, \ldots, \mathbb{W}_{l+1}, \mathbf{b}_1, \ldots, \mathbf{b}_{l+1}) = \rho_{l+1}(\mathbb{W}_{l+1}\rho_l(\ldots \mathbb{W}_2\rho_1(\mathbb{W}_1\mathbf{v} + \mathbf{b}_1) + \mathbf{b}_2 \ldots) + \mathbf{b}_{l+1}).$$

For the sake of clarity, we make a distinction between the notation related to an NN seen as an operator – that is, a parametrized function denoted with $\mathcal{N}$ and depending on the input vector $\mathbf{v} \in \mathbb{R}^{n_1}$, the weights and biases $\mathbb{W}_i, \mathbf{b}_i$ for $i = 1, \ldots, l+1$ – from the output of the same neural network, that is, a vector $\mathbf{w} \in \mathbb{R}^{n_{l+1}}$. To lighten the notation, we denote the hyperparameters of the neural network, namely the collection of weights and biases, with the general symbol $\boldsymbol{\theta}$. The choice of the activation function is left to the user and is problem-dependent. Furthermore, it can be neglected in the output layer [52, 53].

After determining the architecture of the neural network, the hyperparameters for the training process are set and tuned. Hinging upon the choice of the activation function for each layer, the initial state is provided: weights and biases are set to zero or randomly initialized in order to retain the variance of the activation across every layer, avoiding the exponential decrease or increase of the magnitude of input signals. In the context of *supervised learning*, the training of a neural network consists of tuning its weights and biases, simply put $\boldsymbol{\theta}$, by means of the minimization of a suitable *loss* function $\mathcal{E}$ that measures the discrepancy between a given dataset and the predictions of the neural network. The minimization is typically performed by means of a gradient descent algorithm [54], such as the L-BFGS method [55] (a quasi-Newton optimizer) or ADAM [56].

The computational costs of the training are heavily linked to the *complexity* of the neural network, that depends on its depth (the number of layers), on the number of neurons within each layer and on possible constraints on the weighted and biases. Regarding the latter, the architecture is *dense* if no constraints are imposed. In the following subsection, we discuss a different approach, that introduces sparsity in the weights matrix to efficiently approximate FE functions, exploiting their mesh-based information.

3.2. Mesh-informed neural networks for the approximation of finite element functions

Despite their incredible expressivity, plain NN architectures based on dense layers can incur into major limitations, especially when dealing with high-dimensional data. In fact, when the dimensions into play become fairly large, dense architectures quickly become intractable, as they result in models that are harder to optimize and often prone to overfitting (thus also requiring more data for their training). Unfortunately, within our context, these issues arise quite naturally. In fact, the geometrical features of the microstructure are encoded at a high-fidelity level: consequently, any architecture working with the microstructure information must be powerful enough to handle an input of dimension $O(N_h)$, with $N_h \gg 1$.

To overcome this difficulty, we rely on a particular class of sparse architectures called Mesh-Informed Neural Networks (MINNs), originally proposed in [43] as a way to handle high-dimensional data generated by FE discretizations. MINNs are obtained through an a priori pruning strategy that promotes local operations rather than global ones. We can summarize the general idea underlying MINNs as follows. Let $\mathbb{R}^{N_h} \cong V_h$ and $\mathbb{R}^{M_{h'}} \cong V_{h'}$ be two vector spaces, each associated to a given FE discretization. The two may refer to different discretizations in terms of mesh step size or polynomial degree, but they must be defined over a common spatial domain Ω.

Let

$$\{\mathbf{x}_j\}_{j=1}^{N_h} \subset \Omega \quad \text{and} \quad \{\mathbf{x}'_i\}_{i=1}^{M_{h'}} \subset \Omega,$$

be the coordinates of the nodes for the two FE spaces, respectively. Then, following the definition in [43], a mesh-informed layer with support $r > 0$ is a DNN layer $\mathcal{L} : \mathbb{R}^{N_h} \to \mathbb{R}^{M_{h'}}$ whose weight matrix $\mathbb{W}$ satisfies the sparsity constraint below

$$\mathbb{W}_{i,j} = 0 \quad \forall i, j, \text{ such that } \; d(\mathbf{x}_j, \mathbf{x}'_i) > r,$$

where $d : \Omega \times \Omega \to [0, +\infty)$ is a suitable distance function defined over the spatial domain, such as, e.g., the Euclidean distance $d(\mathbf{x}, \mathbf{x}') := |\mathbf{x} - \mathbf{x}'|$ – this choice, however, is not restrictive, making then possible to use also a geodesic distance. Here, the intuition is that each entry component j at input (resp. output) has been associated to some node $\mathbf{x}_j$ representing a degree of freedom in the FEspace at input (resp. output). Then, this interpretation allows one to construct a sparse DNN model whose weight matrix ultimately acts as a local operator in terms of the corresponding FE spaces. In general, for simplicial meshes in $\Omega \subset \mathbb{R}^d$, it can be shown that the weight matrix of a mesh-informed layer with support $r > 0$ has at most $O(r^d N_h M_{h'})$ nonzero entries [43]. The advantage of such construction is that it reduces the number of hyparameters to be optimized during the training stage without affecting the overall expressivity of the model.

In particular, training a MINN model can be far less demanding than doing the same with its dense counterpart; furthermore, the sparsity constraints naturally introduced in a MINN can have a beneficial impact on the generalization capabilities of the architecture, see, e.g., [43] for further details.

In light of these considerations, for what concerns our purposes, we shall exploit both mesh-informed and dense layers: the choice will depend, from time to time, on the nature of the data at hand. To distinguish between MINNs and general dense neural networks, we denote by $\mathcal{M}$ those architectures that feature the use of mesh-informed layers.

4. Non-intrusive and nonlinear ROMs: POD-MINN and POD-MINN+ strategies

In this section we illustrate a strategy to use mesh-informed neural networks for the construction of non-intrusive ROMs for problems with microstructure, detailing the approximation introduced in equation (12). We remark that we will exploit MINNs in two distinct steps:

1. in the first step, directly inspired to [21], MINNs are used to approximate the map from the parameters of the problem to the reduced coefficients using the network

$$\mathcal{M}_{rb}: \quad \mathcal{P} \to \mathbb{R}^{n_{rb}}, \quad \boldsymbol{\mu} \mapsto \widehat{\mathbf{u}}_{\boldsymbol{\mu},rb};$$

2. in the second step, MINNs are exploited for the *closure* model, namely a nonlinear correction to the previous ROM (based on a linear approximation space). To this purpose, we introduce the MINN $\mathcal{M}_c$ such that

$$\mathcal{M}_c: \quad \mathcal{P} \to \mathbb{R}^{N_h}, \quad \boldsymbol{\mu} \mapsto \widetilde{\mathbf{u}}_{\boldsymbol{\mu},c}.$$

The overall framework is summarized in Figure 3. We recall that parameters affecting the differential problem feature a *scale separation property*, with macroscale parameters showing a small dimensional and, conversely, microscale parameters that are instead high-dimensional – that is, $n_m \gg n_M$. Approximating quantities depending on the former can thus be done through plain feedforward neural networks, while we decide to encode the latter through a mesh-informed neural network to approximate spatially-dependent parametric fields related to the microscale. We shall detail how in the upcoming subsection.

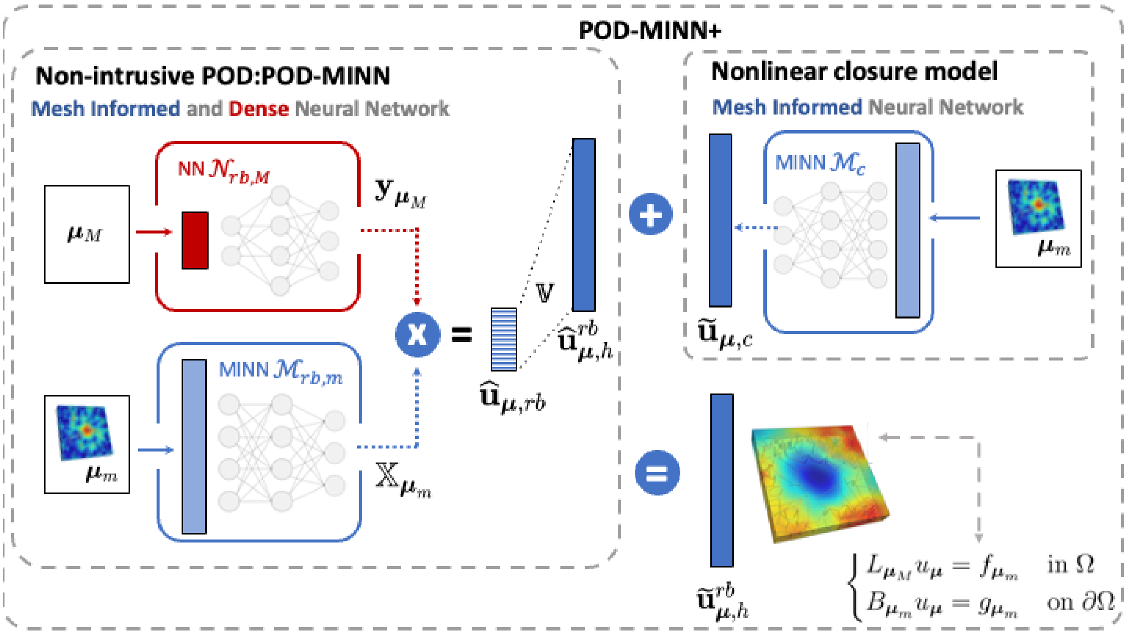

POD-MINN+
Non-intrusive POD:POD-MINN
Mesh Informed and Dense Neural Network
Nonlinear closure model
Mesh Informed Neural Network
$\boldsymbol{\mu}_M$
NN $\mathcal{N}_{rb,M}$
$\mathbf{y}_{\boldsymbol{\mu}_M}$
$\boldsymbol{\mu}_m$
MINN $\mathcal{M}_{rb,m}$
$\mathbb{X}_{\boldsymbol{\mu}_m}$
x
=
$\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}$
$\mathbb{V}$
$\widehat{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h}$
+
MINN $\mathcal{M}_c$
$\widetilde{\mathbf{u}}_{\boldsymbol{\mu},c}$
$\boldsymbol{\mu}_m$
=
$\widetilde{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h}$
$L_{\boldsymbol{\mu}_M} u_{\boldsymbol{\mu}} = f_{\boldsymbol{\mu}_m}$ in Ω
$B_{\boldsymbol{\mu}_m} u_{\boldsymbol{\mu}} = g_{\boldsymbol{\mu}_m}$ on $\partial\Omega$

Figure 3: A sketch of the POD-MINN+ method. The macroscale parameters, $\boldsymbol{\mu}_M$, and the microscale ones, $\boldsymbol{\mu}_m$, are fed to two separate architectures, $\mathcal{N}_{rb,M}$ and $\mathcal{M}_{rb,m}$, whose outputs are later combined to approximate the RB coefficients, cf. Eq. (15). The coefficients are then expanded over the POD basis, $\mathbb{V}$, and the ROM solution is further corrected with a closure term computed by a third network, $\mathcal{M}_c$, that accounts for the small scales.

4.1. Approximation of the POD coefficients using mesh informed neural networks

We observe that the best approximation of the FOM solution $u_{\boldsymbol{\mu},h}$ – whose degrees of freedom are collected in the vector $\mathbf{u}_{\boldsymbol{\mu},h} \in \mathbb{R}^{N_h}$ – in the POD space is given by its projection

$$\mathbb{V}\mathbb{V}^T \mathbf{u}_{\boldsymbol{\mu},h} = \sum_{i=1}^{n_{rb}} \left(\mathbb{V}^T \mathbf{u}_{\boldsymbol{\mu},h}\right)^{(i)} \psi_i(x).$$

The expression above defines a mapping

$$\pi : \quad \mathcal{P} \subset \mathbb{R}^p \to \mathbb{R}^{n_{rb}}, \quad \boldsymbol{\mu} \mapsto \mathbb{V}^T \mathbf{u}_{\boldsymbol{\mu},h}, \tag{13}$$

whose approximation could provide a way to generate, onto the POD space, the solution of the problem for unseen values of the input parameters. As shown in [21], if the dimension of the reduced space is small enough, it is possible to effectively approximate the map π by means of a neural network. Since for the case of our interest the parameter space $\mathcal{P}$ includes the high-dimensional data necessary to encode the microscale parameters $\boldsymbol{\mu}_m$, we use a MINN instead of a generic feedforward neural network.

We remark that this method overrides the projection step of the FOM onto the reduced space, (9), and the solution of the reduced problem (8). As a consequence, it turns out to be a completely non-intrusive reduced order modeling strategy that will be particularly advantageous to handle problems that do not enjoy an affine decomposition of the FOM operators, or in which one is interested to approximate only a subset of the variables involved in its formulation.

The dataset to train the network $\mathcal{M}_{rb}$ is the representation in the reduced space of the FOM solutions collected into the data matrix $\mathbb{S}$. In other words, the training set are the input-output couples $\{\boldsymbol{\mu}^{(i)}, \mathbb{V}^T\mathbb{S}(:,i)\}_{i=1}^N$, where $\boldsymbol{\mu}^{(i)}$ are the parameter values used to calculate the FOM solution vector $\mathbb{S}(:,i) = \mathbf{u}_{\boldsymbol{\mu}^{(i)},h} = \mathbf{u}_h(\boldsymbol{\mu}^{(i)})$. This dataset is used to define the loss function,

$$\mathcal{E}(\mathcal{M}_{rb}; \boldsymbol{\theta}_{rb}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbb{V}^T\mathbb{S}(:,i) - \mathcal{M}_{rb}(\boldsymbol{\mu}^{(i)}; \boldsymbol{\theta}_{rb})\|_{2,n_{rb}},$$

and the (perfectly) trained neural network is the one characterized by the paramaters $\boldsymbol{\theta}_{rb}^*$ that minimize the loss, precisely $\boldsymbol{\theta}_{rb}^* = \mathrm{argmin}_{\boldsymbol{\theta}} \mathcal{E}(\mathcal{M}_{rb}; \boldsymbol{\theta}_{rb})$.

Once the network has been trained, the model can be used in an online phase similarly to the baseline reduced basis approach: given a new set of parameters $\boldsymbol{\mu}$, the mapping through the network $\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} = \mathcal{M}_{rb}(\boldsymbol{\mu}; \boldsymbol{\theta}_{rb}^*)$ and a subsequent linear transformation, $\mathbb{V}\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}$, enable the approximation of the FOM solution $\mathbf{u}_{\boldsymbol{\mu},h}(\boldsymbol{\mu})$. The approximation provided by $\mathcal{N}_{rb}$ can be represented by the following function,

$$\widehat{u}_{\boldsymbol{\mu},h}^{rb}(x) = \sum_{i=1}^{n_{rb}} \widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}^{(i)} \psi_i(x) = \sum_{j=1}^{N_h} \widehat{\mathbf{u}}_{\boldsymbol{\mu},h}^{rb,(j)} \phi_j(x), \tag{14}$$

where $\widehat{\mathbf{u}}_{\boldsymbol{\mu},h}^{rb} = \mathbb{V}\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} \in \mathbb{R}^{N_h}$ is the vector collecting its degrees of freedom in the FEM space.

The POD-MINN method is based on specific architectures designed with the purpose to manage differently the parameters related to the *micro* and *macro* scales, as they belong to spaces with different dimension. The first step, $\mathcal{M}_{rb}$ combines a MINN that manages the high-dimensional information about $\boldsymbol{\mu}_m$ with a DNN that deals with $\boldsymbol{\mu}_M$, leading to the following formulation of an input-output map that approximates the reduced coefficients:

$$\mathcal{M}_{rb}(\boldsymbol{\mu}; \boldsymbol{\theta}_{rb}) = \mathcal{M}_{rb,m}(\boldsymbol{\mu}_m; \boldsymbol{\theta}_m) \mathcal{N}_{rb,M}(\boldsymbol{\mu}_M; \boldsymbol{\theta}_M), \tag{15}$$

where:

- The output of $\mathcal{N}_{rb,M} : \mathbb{R}^{n_M} \to \mathbb{R}^k$ is a vector $\mathbf{y}_{\boldsymbol{\mu}_M} = \mathcal{N}_{rb,M}(\boldsymbol{\mu}_M) \in \mathbb{R}^k$;

- The network $\mathcal{M}_{rb,m} : \mathbb{R}^{n_m} \to \mathbb{R}^{n_{rb} \times k}$ gives back a matrix $\mathbb{X}_{\boldsymbol{\mu}_m} \in \mathbb{R}^{n_{rb} \times k}$.

We note that the network $\mathcal{M}_{rb}$ approximates the mapping π defined in (13). The final output is then represented by

$$\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} = \mathbb{X}_{\boldsymbol{\mu}_m} \cdot \mathbf{y}_{\boldsymbol{\mu}_M},$$

where $\cdot$ emphasizes the presence of a matrix-vector multiplication. The training of the network defined in (15) is done all at once using the loss function $\mathcal{E}(\mathcal{M}_{rb}; \boldsymbol{\theta}_{rb})$. The respective weights and biases, collected in the vectors $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}_M$, are drawn from the normalized initialization proposed in [57].

Although the approach proposed here provides an efficient way to recover the main features of the FOM solution, it is characterised by two main error sources, in fact, regardless of the training of the network, by orthogonality of $\mathbb{V}$:

$$\|\mathbf{u}_{\boldsymbol{\mu},h} - \mathbb{V}\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}\|^2_{2,N_h} = \|\mathbf{u}_{\boldsymbol{\mu},h} - \mathbb{V}\mathbb{V}^T\mathbf{u}_{\boldsymbol{\mu},h}\|^2_{2,N_h} + \|\mathbb{V}^T\mathbf{u}_{\boldsymbol{\mu},h} - \widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}\|^2_{2,n_{rb}}.$$

This is to emphasize that the neural network only provides an estimate of the map π, meaning that $\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} \approx \mathbb{V}^T\mathbf{u}_{\boldsymbol{\mu},h}$; furthermore, on top of this, the representation of the FOM solution in the reduced space is affected by a projection error $\|\mathbf{u}_{\boldsymbol{\mu},h} - \mathbb{V}\mathbb{V}^T\mathbf{u}_{\boldsymbol{\mu},h}\|_{2,N_h}$, identifying a lower bound for the approximation of the FOM solution that can not be overcome by the POD-MINN approach. The approximation of problems with microstructure entails the choice of a higher number of reduced basis functions in order to keep the projection error sufficiently small and, consequently, the reconstruction error of the solution. We tackle these issues by means of a closure model, as shown in the following section.

4.2. Mesh informed neural networks for the closure model

Designing a closure model for problems with microstructure is particularly challenging for the following reasons: *(i)* the input of the network $\mathcal{M}_c$ must necessarily include the microscale parameters $\boldsymbol{\mu}_m$ that are represented in the high-dimensional FE space relative to the FOM model; *(ii)* the output of the closure model is also isomorphic to the FE space of the FOM; *(iii)* as a consequence of *(ii)*, the training data of $\mathcal{M}_c$ are also high-dimensional. For these reasons, MINNs play a crucial role for the successful application of the closure model.

Let us denote with $\widetilde{\mathbf{u}}_{\boldsymbol{\mu},c}$ the output of the closure model, namely $\widetilde{\mathbf{u}}_{\boldsymbol{\mu},c} = \mathcal{M}_c(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$. Then, following (12), the closure model is easily added to the reduced basis approximation,

$$\widetilde{u}^{rb}_{\boldsymbol{\mu},h}(x) = \sum_{j=1}^{N_h} \widetilde{\mathbf{u}}^{(j)}_{\boldsymbol{\mu},c}\phi_j(x) + \sum_{i=1}^{n_{rb}} \widehat{\mathbf{u}}^{(i)}_{\boldsymbol{\mu},rb}\psi_i(x), \tag{16}$$

where $\psi_i(x)$ are the reduced bases while $\phi_j(x)$ are the standard FE basis functions used in the FOM model. In vector form, the previous equation is equivalent to

$$\widetilde{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h} = \widetilde{\mathbf{u}}_{\boldsymbol{\mu},c} + \mathbb{V}\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}.$$

The dataset for training the closure model consists of the FOM snapshots collected in the data matrix $\mathbb{S}$, more precisely the couples $\{\boldsymbol{\mu}^{(i)}, \mathbb{S}(:,i)\}_{i=1}^N$. Moreover, we make the choice to feed the closure model only with the microscale input parameters. Precisely, we have

$$\mathcal{M}_c(\boldsymbol{\mu}; \boldsymbol{\theta}_c) = \mathcal{M}_c(\boldsymbol{\mu}_m; \boldsymbol{\theta}_c). \tag{17}$$

As a result, we look for a network $\mathcal{M}_c(\boldsymbol{\mu}_m; \boldsymbol{\theta}_c) : \mathbb{R}^{n_m} \to \mathbb{R}^{N_h}$ that minimizes the following loss function,

$$\mathcal{E}(\mathcal{M}_c; \boldsymbol{\theta}_c) = \frac{1}{N}\sum_{i=1}^{N} \|\big(\mathbb{S}(:,i) - \mathbb{V}\mathcal{M}_{rb}(\boldsymbol{\mu}^{(i)}; \boldsymbol{\theta}^*_{rb})\big) - \mathcal{M}_c(\boldsymbol{\mu}^{(i)}_m; \boldsymbol{\theta}_c)\|_{2,N_h},$$

where $\mathbb{S}(:,i) - \mathbb{V}\mathcal{M}_{rb}(\boldsymbol{\mu}^{(i)}; \boldsymbol{\theta}^*_{rb})$ represents the approximation error generated by the POD-MINN method, related to the i^{th}-snapshot, which is then corrected by $\mathcal{M}_c$ to further reduce it. The weights and biases of $\boldsymbol{\theta}_c$ are initialized to zero, since the optimization process starts from an initial state determined by the POD-MINN reconstruction of the solution.

As result, the POD-MINN+ approximation of the FOM solution is given by the following expression:

$$\widetilde{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h} = \mathbb{V}\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} + \widetilde{\mathbf{u}}_{\boldsymbol{\mu},c} = \mathbb{V}\mathbb{X}_{\boldsymbol{\mu}_m}\mathbf{y}_{\boldsymbol{\mu}_M} + \mathcal{M}_c(\boldsymbol{\mu}_m) \approx \mathbf{u}_{\boldsymbol{\mu},h}.$$

In conclusion, augmenting the POD-MINN trained map through the closure model allows to retrieve also the variability of local features in the solution manifold that POD-MINN can recover only considering a large number of reduced basis functions, including the small frequency POD modes. The method with closure is called POD-MINN+.

4.3. Numerical results obtained with the POD-MINN and POD-MINN+ methods

In this section we analyze the performance of the POD-MINN+ method in the approximation of the benchmark problems proposed in section 2.5. To assess the performance of the closure model, we compare the POD-MINN+ reconstruction error with the POD-MINN one and the projection error generated projecting the FOM solution onto the POD space, defined as in (10). The former two are defined as follows:

$$E_{POD-MINN}(n_{rb}, \mathbf{u}_{\boldsymbol{\mu},h}) = \frac{\|\mathbf{u}_{\boldsymbol{\mu},h} - \mathbb{V}(n_{rb})\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb}\|_{2,N_h}}{\|\mathbf{u}_{\boldsymbol{\mu},h}\|_{2,N_h}}, \tag{18}$$

$$E_{POD-MINN+}(n_{rb}, \mathbf{u}_{\boldsymbol{\mu},h}) = \frac{\|\mathbf{u}_{\boldsymbol{\mu},h} - \mathbb{V}(n_{rb})\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} - \widetilde{\mathbf{u}}_{\boldsymbol{\mu},c}\|_{2,N_h}}{\|\mathbf{u}_{\boldsymbol{\mu},h}\|_{2,N_h}}. \tag{19}$$

For both benchmark problems addressed in Sect. 2.5, we consider the same architectures for the POD-MINN method and the closure model, hinging upon the use of MINNs. The microscale input parameters $\boldsymbol{\mu}_m$ are the representations of the respective forcing terms $f(x, y)$ in the finite-element space of the solution V_h, with dimension N_h. Let $\mathcal{T}_h$ denote the computational mesh (collection of all geometric elements) related to the space V_h. We remind that we consider a unit square domain $\Omega = (-1, 1)^2$ partitioned using uniform triangles corresponding to a 50×50 grid named $\mathcal{T}_h$.

As a first step, related to the POD-MINN approach, we introduce the map approximating the reduced coefficients defining the architecture $\mathcal{M}_{rb}(\boldsymbol{\mu}_m; \boldsymbol{\theta}_{rb}) : \mathbb{R}^{N_h} \to \mathbb{R}^{n_{rb}}$, exploiting a mesh-informed layer L^{rb}_1 of support $r = 0.6$ and a dense layer L^{rb}_2:

$$L^{rb}_1 :\ V_h \xrightarrow{r=0.6} V_{2h}, \qquad L^{rb}_2 : V_{2h} \longrightarrow \mathbb{R}^{n_{rb}}$$

with $V_{2h} = X^1_{2h}(\Omega)$ finite-element space defined over the coarser computational mesh $\mathcal{T}_{2h}$ (stepsize $2h$, 25×25 grid), made by uniform triangles, of the same domain Ω. For the closure model, let $\mathcal{T}_{h'}$ a coarse mesh in Ω, with $h' > h$. We use a MINN architecture $\mathcal{M}_c(\boldsymbol{\mu}_m; \boldsymbol{\theta}_c) : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$, composed by 2 mesh-informed layers L^c_i of support $r = 0.6$:

$$L^c_1 :\ V_h \xrightarrow{r=0.6} V_{h'}, \qquad L^c_2 : V_{h'} \xrightarrow{r=0.6} V_h$$

where $V_{h'} = X^1_{h'}(\Omega)$ is the finite-element space defined from a 35×35 computational mesh $\mathcal{T}_{h'}$ of uniform triangles. As result, the MINNs $\mathcal{M}_{rb}(\boldsymbol{\mu}_m; \boldsymbol{\theta}_{rb})$ and $\mathcal{M}_c(\boldsymbol{\mu}_m; \boldsymbol{\theta}_c)$ are defined as follows:

$$\mathcal{M}_{rb} : V_h \xrightarrow{r=0.6} V_{2h} \longrightarrow \mathbb{R}^{n_{rb}}, \qquad \mathcal{M}_c :\ V_h \xrightarrow{r=0.6} V_{h'} \xrightarrow{r=0.6} V_h.$$

For what concerns the training of the neural networks, the $N = 1000$ snapshot functions, taken into account to build the data matrix $\mathbb{S}$, are partitioned to provide:

- a *training* set of $N_{train} = 750$ samples;
- a *validation* set consisting of $N_{valid} = 50$ snapshots;
- a *test* set of $N_{valid} = 200$ snapshots.

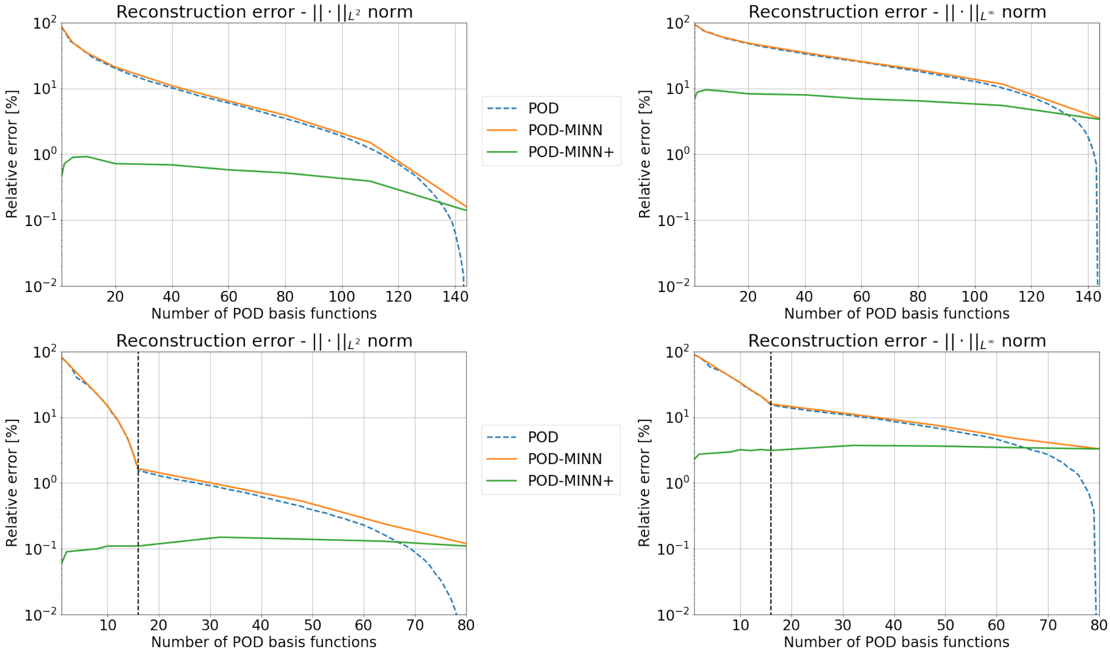


Figure 4: The performance of the POD-MINN+ method applied to the benchmark problems with continuous scales (top row) and separate scales (bottom row). We compare the errors $E_{POD-MINN}$ and $E_{POD-MINN+}$ defined in (18), to the one of the POD projection defined as E_{POD} in (10), measured in the norms $p = 2, \infty$. The case with $p = 2$ is reported on the left, the one with $p = \infty$ is on the right.

The optimization of the loss function is performed through the L-BFGS optimizer with learning rate equal to 1, without batching. The networks are trained for a total of 250 epochs, using an early stopping criterion based on the validation error, that is applied if the following conditions are met: the training error *decreases* but the validation error *increases* for at least two consecutive epochs.

The training times for the POD-MINN+ method in the two benchmark problems vary between from 98 to 130 seconds for the test case with continuous scales and from 89 to 188 seconds for the test case with continuously variables scales. As a general trend, we denote that an increase of the number of POD basis functions entails a decrease of the training time of the closure model.

The comparison between the POD projection error, namely E_{POD}, and $E_{PODMINN+}$ when varying the number of basis functions n_{rb} is reported in Figure 4 for the benchmark problems with continuously variable scales and with scale separation. These results confirm that the the POD-MINN approach is not performing well, because it can not provide a better approximation than the POD projection error. The POD-MINN error is indeed very close to the POD projection, but this approximation is qualitatively not satisfactory for both the L^2 and L^∞ norms, unless a large number of bases is used.

Conversely, the POD-MINN+ method is very effective especially in the regime with a low number of basis functions. In particular, for the case with continuously variable scales the POD-MINN+ method reduces of 2 orders of magnitude the POD error in the region $n_{rb} < 20$, when it is measured in the 2-norm. For the benchmark problem with the scale separation the gain increases to more than three orders of magnitude, see Figure 4 (bottom-left panel). This effect is particularly evident when we approximate this benchmark problem with basis functions only able to capture the low frequency modes, namely the case $n_{rb} \leq 16$. In this regime the closure model is entirely responsible for the approximation of the high frequency modes. The POD-MINN+ method effectively performs this task with a relative error of $0.1\% = 10^{-3}$.

We note that the closure model does not improve the rate of convergence with respect to n_{rb}, it only decreases the magnitude of the error. In general, the error of the POD-Galerkin method enjoys an optimal exponential decay with respect to n_{rb} that can not be increased resorting to the closure model. This observation confirms that the POD-MINN+ method is meaningful for those problems where the Kolmogorov n-width and consequently the POD projection error decays slowly with respect to n_{rb}.

5. Application to oxygen transfer in the microcirculation

In this section we present an application of the POD-MINN+ method to model oxygen transfer at the level of microcirculation, described by (3). After introducing the FOM model and its discretization, we discuss its parametrization and finally we present the numerical results that illustrate the advantages of the proposed reduced order model.

5.1. Numerical approximation of oxygen transfer in the microcirculation: the full order model

For the spatial approximation of the problem we use the finite-element method. We consider a 3D domain Ω, identified by a slab of edge $1\,mm$ and thickness $0.15\,mm$ and discretized through a $20 \times 20 \times 3$ structured computational mesh of tetrahedra. For the interstitial domain we introduce the space of the piecewise linear, continuous, Lagrangian finite-elements $V_{t,h} = X_h^1(\Omega)$, with dimension $N_h = 1764$. On the other hand, we assume that the 1D vascular network is immersed in the 3D slab and we discretize it by partitioning each vascular branch Λ_i into a sufficiently large number of linear segments. A piecewise linear, continuous, Lagrangian finite-element space $V_{v,h}^i = X_h^1(\Lambda_i)$ is employed for each branch Λ_i, where the index i spans through the vascular branches $i = 1, \dots, N_b$. Hence, we approximate the equations in the whole microvascular network through a finite element space $V_{v,h} = \left(\bigcup_{i=1}^{N_b} V_{v,h}^i \right) \bigcap C^0(\Lambda)$.

It is important to clarify that the finite-element approximation of the oxygen transfer model is performed downstream to the full comprehensive problem for the vascular microenvironment, described in [3, 49]. As it is shown in Figure 5, indeed, we firstly compute via finite-element method the velocity and the pressure in the tissue and the vascular network, together with the discharge hematocrit in the vascular network. We remark that this proposed ROM based on the POD-MINN+ method could have been equivalently applied to any variable in the domain Ω, such as the pressure or the velocity fields.

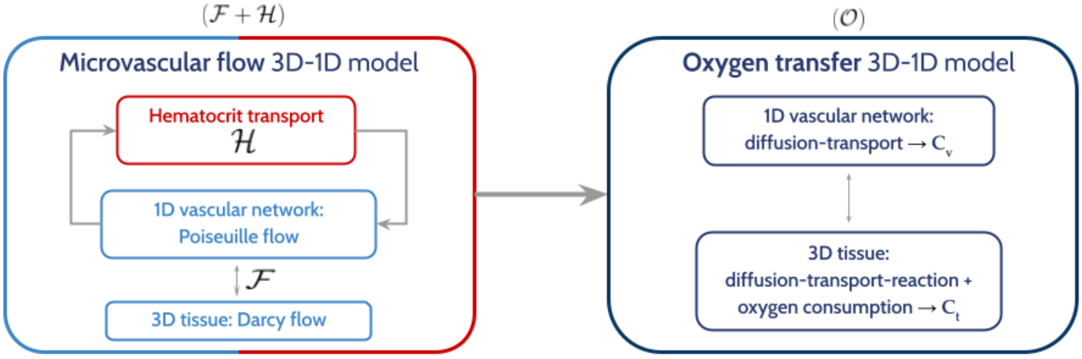

$(\mathcal{F}+\mathcal{H})$
Microvascular flow 3D-1D model
Hematocrit transport
$\mathcal{H}$
1D vascular network:
Poiseuille flow
$\mathcal{F}$
3D tissue: Darcy flow
$(\mathcal{O})$
Oxygen transfer 3D-1D model
1D vascular network:
diffusion-transport → C_v
3D tissue:
diffusion-transport-reaction +
oxygen consumption → C_t

Figure 5: General layout of the full order model for the whole vascular microenvironment.

5.2. The parametrization of the FOM: inputs and outputs

Since the model presented in Section 2 is driven by parametrized PDEs, a crucial aspect of the problem is the choice of which parameters to include as inputs of the model. We consider the parameter space $\mathcal{P} = \mathcal{P}_M \cup \mathcal{P}_m$, where the macro- and micro-scale subspaces are selected according to the results of an earlier sensitivity analysis [58]. We then select three physical parameters (see Table 1) among those appearing in the model (3), and two geometrical, mesh-based inputs, encoding the information related to the 1D vascular networks. The range of variation of each input parameter is determined consistently with respect to its physiological bounds.

In particular, the vascular architectures are obtained starting from a biomimetic algorithm that replicate the essential traits of angiogenesis, starting from some parameters that characterize the density and the distribution of small vessels in a vascularized tissue. These quantities are the vascular surface per unit volume of tissue, named S/V, and an indicator that governs the distribution of point seeds used to initialize the angiogenesis algorithm. The definitions and the range of these hyper-parameters of the vascular model are reported in Table 1. By sampling these hyper-parameters and applying the angiogenesis algorithm, we obtain a population of admissible vascular networks with sufficient variability. Figure 6 shows some examples of 1D vascular networks with different spatial distributions, spanning from regular structures to less ordered ones. For more details regarding the generation of the embedded microstructures, we refer to [58]. From the computational standpoint, the vascular networks are represented as metric graphs, the vertices of which are points in the 3D space. This description is not practical for the purpose of generating a reduced order model. It is more convenient to transform the vascular graph into two continuous functions. The main one is the distance function from the nearest point of the vascular network, defined on Ω_t and named $\mathbf{d}$. The second one is used to identify the intersections of the vascular network with the boundary and it is named $\boldsymbol{\eta}$. These are the input functions of the ROM and are presented below.

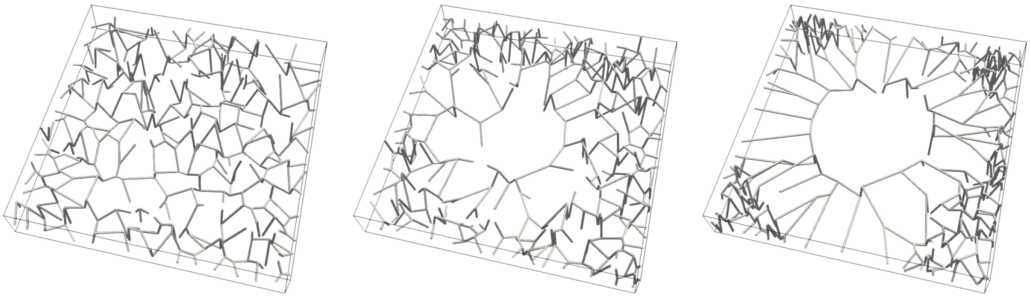


Figure 6: Examples of artificial vascular networks with extravascular distance $\mathbf{d}$ progressively increasing from left to right.

5.2.1. *Extravascular distance and inlet function*

As a way to retrieve an equivalent information of the 1D vascular graph we thus choose to introduce the *extravascular distance i.e.* the function mapping each point in the tissue slab to its distance from the closest point of the vascular network. We represent this function as a linear and piecewise continuous finite element function in $V_{t,h}$. The vector of degrees of freedom of such discrete function, corresponding to its nodal values, is $\mathbf{d} \in \mathbb{R}^{N_h}$. The vector $\mathbf{d}$ plays the role of the forcing term $f_{\boldsymbol{\mu}_m}$ in the model (2).

Furthermore, it is useful to provide the ROM with the information of the inlet points of the vascular network on the boundary. To this purpose, we introduce an *inlet characteristic function* that assigns a unit value to nodes of the tissue slab grid which are close to the inlets. Again, this function is defined at the discrete level using the space $V_{t,h}$. In practice, a vertex of the finite element mesh is marked as having non-zero value if and only if one of the surface elements is intersected by an inlet vessel. The degrees of freedom of this function are denoted by $\boldsymbol{\eta} \in \mathbb{R}^{N_h}$. Figure 8 shows the mesh-based data $\mathbf{d}$ and $\boldsymbol{\eta}$ for a particular instance of the vascular graph.

We point out, without loss of generality, that we can extend this approach encoding the geometry of the vascular network with high-dimensional data defined over coarser meshes (*super-resolution*) or refined ones (*sub-resolution*) [59].

5.2.2. Output of the full order model

The output of the FOM is the tissue oxygenation map C_t, one of the state variables of the oxygen transfer model, measured in mL_{O_2}/mL_B. In the equations (2) it represents the high-fidelity solution $u_{\boldsymbol{\mu}}$. An example of the oxygen map C_t is shown in Figure 7, where we notice that the embedded vascular microstructure has a significant influence of on the tissue oxygenation. We also note that for the development of the ROM the variable C_t is suitably rescaled by a factor 300. We remark that we can exploit the 3D-1D full order model of the vascular microenvironment to extend the proposed approach for the approximation of other state variables such as the interstitial pressure in the tissue domain Ω_t. On the contrary, the methodology has to be revised if the aim is to reconstruct outputs defined over the vascular network.

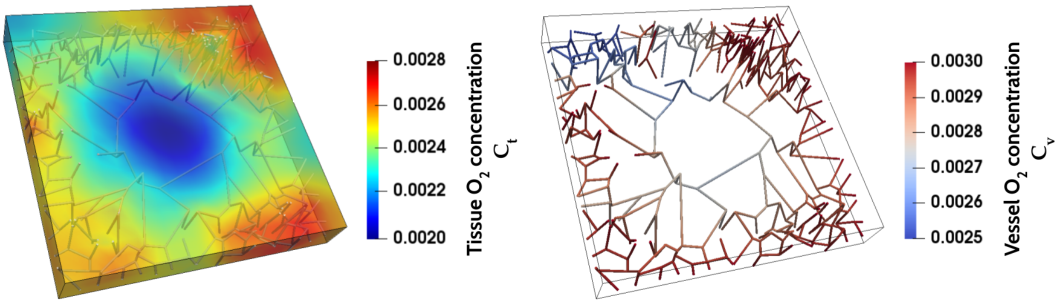


Figure 7: The FOM solution representing the tissue oxygenation map (mL_{O_2}/mL_B) is reported on the left panel. On the right we show the 1D embedded vascular microstructure that visibly influences the oxygen map, over which the vessel oxygen concentration C_v is represented.

SYMBOL	PARAMETER	UNIT	RANGE OF VARIATION
P_{O_2}	O_2 wall permeability	m/s	$3.5 \times 10^{-5} - 3.0 \times 10^{-4}$
V_{max}	O_2 consumption rate	$mL_{O_2}/cm^3/s$	$4.0 \times 10^{-5} - 2.4 \times 10^{-4}$
$C_{v,in}$	O_2 concentration at the inlets	mL_{O_2}/mL_B	$2.25 \times 10^{-3} - 3.75 \times 10^{-3}$
$\% \frac{SEEDS_{(-)}}{SEEDS_{(+)}}$	Seeds for angiogenesis	$[\%]$	$0 - 75$
S/V	Vascular surface per unit volume	$[m^{-1}]$	$5 \cdot 10^3 - 7 \cdot 10^3$

Table 1: The first three rows illustrate the biophysical parameters of the ROM and their ranges of variation. The last two rows report the hyper-parameters used to initialize the algorithm that generates the vascular network.

5.3. Implementation of the POD-MINN and POD-MINN+ methods

Once defined the macroscale and the microscale parameters, namely $\boldsymbol{\mu}_M = [V_{max}, C_{v,in}, P_{O_2}]$ and $\boldsymbol{\mu}_m = (\mathbf{d}, \boldsymbol{\eta})$, we apply a Monte Carlo sampling of the parameter space and collect the corresponding input-output pairs provided by the FOM model, named $\{(\boldsymbol{\mu}_M^{(i)}, \mathbf{d}^{(i)}, \boldsymbol{\eta}^{(i)}); \mathbf{u}_{\boldsymbol{\mu},h}^{(i)}\}_{i=1}^N$. We organize these data into the data matrix $\mathbb{S}$. In particular, for what concerns the microscale parameter space, the sampling of the vascular networks is performed with respect to the hyper-parameters introduced in Table 1. Thanks to the chosen settings, the extravascular distance $\mathbf{d}$ and the inlets function $\boldsymbol{\eta}$ share the same dimension and ordering of the snapshots in the matrix $\mathbb{S}$, so that the matching between inputs and outputs can be easily established. The entries of the physical parameters $\boldsymbol{\mu}_M$ are normalized between 0 and 1 with respect to their range of variation reported in Table 1. The available dataset of $N = 1600$ FOM snapshots and parameters samples is partitioned as follows:

- $N_{train} = 1220$ training data;

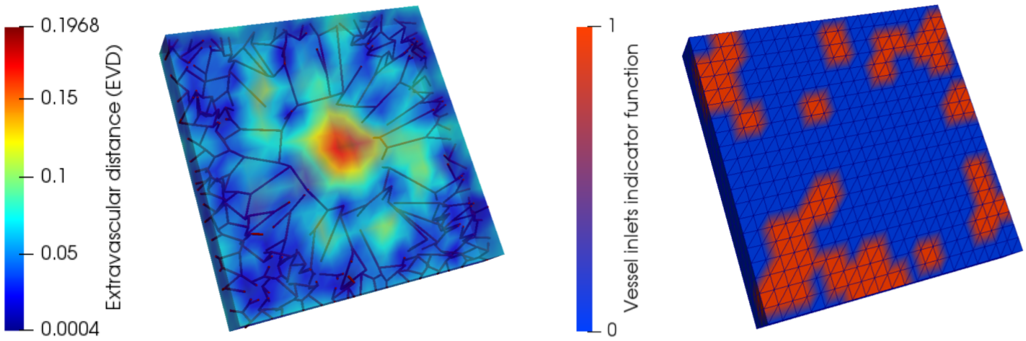

0.1968
0.15
0.1
0.05
0.0004
Extravascular distance (EVD)
1
0
Vessel inlets indicator function

Figure 8: The nodal distance from the nearest vessel of the vascular network is shown on the left. On the right we plot the indicator function of the vessel inlets on the boundary.

- $N_{valid} = 100$ validation data;
- $N_{test} = 280$ testing data.

We perform a singular value decomposition of the training dataset, building the projection matrix $\mathbb{V}$, collecting the discrete representation of n_{rb} basis functions. As a preliminary analysis, we study in Figure 9 the convergence rate of the projection error of the FOM output on the reduced space. These results immediately show that, since the problem is globally diffusion-driven and overall well described by the macroscale features at the low frequencies, the decay of the singular values is relatively quick, meaning that a POD approach is expected to yield good results, as shown in Figure 9 (*left*). Nonetheless, by comparing projection errors measured in the the L^2 and the L^∞ norms, reported in Figure 9 (*right*), we see that the L^∞ error dominates over the L^2 one, meaning that the local effects due to the microstructure still have a big impact on the whole solution. Rather than including a large number of POD modes, it is therefore preferable to retain just few of them, and then include a suitable closure model, with the aim to retrieve the local features of the oxygen map and improve the overall accuracy of the method.

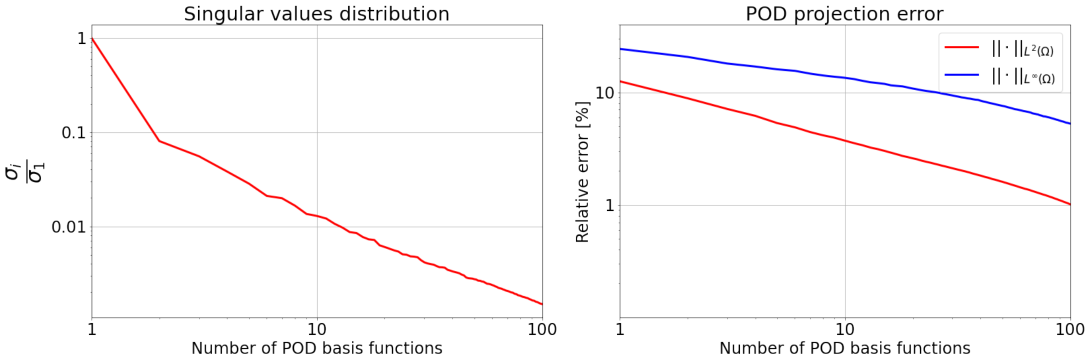


Figure 9: (*Left*) The decay of the singular values distribution for the microcirculation problem (*Right*) The POD projection error of the FOM solution on the reduced basis space varying the number of basis functions.

5.3.1. Architectures and training of the POD-MINN method

In order to implement the POD-MINN method, we exploit two independent MINNs to process the mesh-based geometrical input data $\mathbf{d}$ and $\boldsymbol{\eta}$, using the same architectures in both cases.

We recall the formulation of the input-output function $\mathcal{M}_{rb}$ in (15):

$$\mathcal{M}_{rb}(\boldsymbol{\mu}; \boldsymbol{\theta}_{rb}) = \mathcal{M}_{rb,m}(\boldsymbol{\mu}_m; \boldsymbol{\theta}_m)\mathcal{N}_{rb,M}(\boldsymbol{\mu}_M; \boldsymbol{\theta}_M), \tag{20}$$

where, in this case, the MINN that takes as input the geometrical parametrization is defined as

$$\mathcal{M}_{rb,m}(\boldsymbol{\mu}_m;\boldsymbol{\theta}_m) = \mathcal{M}_{rb,\eta}(\boldsymbol{\eta};\boldsymbol{\theta}_\eta) \odot \mathcal{M}_{rb,d}(\mathbf{d};\boldsymbol{\theta}_d),$$

splitting the contributions of the single geometrical input parameters, with the inlets function $\boldsymbol{\eta}$ weighting the effect of the extravascular distance $\mathbf{d}$ through the MINN $\mathcal{M}_{rb,\eta}$. Here, we denote with $\odot$ the Hadamard product between two matrices,

$$(A \odot B)_{i,j} = (A)_{i,j}(B)_{i,j} \qquad \forall A, B \in \mathbb{R}^{m\times n},\ m, n \geq 1.$$

The neural network $\mathcal{M}_{rb,d}(\mathbf{d};\boldsymbol{\theta}_d) : \mathbb{R}^{N_h} \to \mathbb{R}^{n_{rb}\times k}$ is assembled relying on two mesh-informed layers $L_1^{rb,d}$ and $L_2^{rb,d}$ of support $r = 0.3$ and hyperbolic tangent activation function, combined with a dense layer $L_3^{rb,d}$ complemented with a suitable reshape of the output, which results in the following structure:

- $L_1^{rb,d}$: $V_{t,h} \xrightarrow{r=0.3} V_{t,H}$,
- $L_2^{rb,d}$: $V_{t,H} \xrightarrow{r=0.3} V_{t,H}$,
- $L_3^{rb,d}$: $V_{t,H} \longrightarrow \mathbb{R}^{n_{rb}k} \xrightarrow{\text{reshape}} \mathbb{R}^{n_{rb}\times k}$,

where $V_{t,H} = X_H^1(\Omega)$ is the finite-element space determined from the computational mesh $\mathcal{T}_H$ of tetrahedrons, with stepsize $H > h$ (8×8×3 grid). The network $\mathcal{M}_{rb,d}(\mathbf{d};\boldsymbol{\theta}_d)$ generates a matrix in $\mathbb{R}^{n_{rb}\times k}$. Each column of such matrix is a vector in the space of the reduced coefficients. These $k = 10$ vectors are linearly combined through some hyper-parameters (described below) to obtain the reduced coefficients that best fit the loss function.

The same considerations holds true for the neural network taking the inlet function $\boldsymbol{\eta}$ as input, with the only difference that in the dense layer $L_3^{rb,d}$ no activation is applied, while the dense layer $L_3^{rb,\eta}$ of the MINN $\mathcal{M}_{rb,\eta}$ is composed with a hyperbolic tangent function. In conclusion:

$$\mathcal{M}_{rb,d}, \mathcal{M}_{rb,\eta} : V_{t,h} \xrightarrow{r=0.3} V_{t,H} \xrightarrow{r=0.3} V_{t,H} \longrightarrow \mathbb{R}^{n_{rb}\times 10},$$

giving back two matrices $\mathbb{X}_{\mathbf{d}} \in \mathbb{R}^{n_{rb}\times 10}$ and $\mathbb{X}_{\boldsymbol{\eta}} \in \mathbb{R}^{n_{rb}\times 10}$, respectively.

For what concerns the physical parameters $\boldsymbol{\mu}_M \in \mathbb{R}^3$, a shallow neural network $\mathcal{N}_{rb,M} : \mathbb{R}^3 \to \mathbb{R}^{10}$ with two dense layers $L_1^{rb,m}$ and $L_2^{rb,m}$ is introduced:

$$L_1^{rb,m} : \mathbb{R}^3 \longrightarrow \mathbb{R}^{15}, \qquad L_2^{rb,m} : \mathbb{R}^{15} \longrightarrow \mathbb{R}^{10},$$

where the input layer is composed with a hyperbolic tangent function and the output one has no activation. This leads to the architecture

$$\mathcal{N}_{rb,M} : \mathbb{R}^3 \longrightarrow \mathbb{R}^{15} \longrightarrow \mathbb{R}^{10},$$

that provides as output a vector $\mathbf{y}_{\boldsymbol{\mu}_M} = \mathcal{N}_{rb,M}(\boldsymbol{\mu}_M) \in \mathbb{R}^{10}$.

In order to obtain the approximation of the reduced coefficients, we hinge on the aforementioned data structures and we perform the following linear combination

$$\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} = (\mathbb{X}_{\boldsymbol{\eta}} \odot \mathbb{X}_{\mathbf{d}})\mathbf{y}_{\boldsymbol{\mu}_M} = \mathbb{X}_{\boldsymbol{\mu}_m}\mathbf{y}_{\boldsymbol{\mu}_M}.$$

The network $\mathcal{M}_{rb}$ of equation (20) is trained for at most 200 epochs, minimizing the loss function

$$\mathcal{E}(\mathcal{M}_{rb};\boldsymbol{\theta}_{rb}) = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \|\mathbb{V}^T\mathbb{S}(:,i) - \mathcal{M}_{rb}(\boldsymbol{\mu}^{(i)};\boldsymbol{\theta}_{rb})\|_{2,n_{rb}},$$

relying on the L-BFGS optimizer with learning rate equal to 1 and no batching and applying the same early stopping criterion presented in Section 4.3. We initialize the corresponding weights and biases with the normalized initialization proposed in [57].

5.3.2. Architectures and training of the closure model (POD-MINN+)

The augmented POD-MINN approach is carried out exploiting a closure model $\mathcal{M}_c(\boldsymbol{\mu}_m; \boldsymbol{\theta}_c) : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ that is fed only with the geometrical parameters $\mathbf{d}$ and $\boldsymbol{\eta}$, encoding the microscale features of the microcirculation problem. As in the previous case, we introduce two distinct MINNs that handle each of two inputs, i.e. $\mathcal{M}_{c,d}$ and $\mathcal{M}_{c,\eta}$, separating their individual effects on the correction of the POD-MINN model:

$$\mathcal{M}_c(\boldsymbol{\mu}_m; \boldsymbol{\theta}_c) = \mathcal{M}_{c,\eta}(\boldsymbol{\eta}; \boldsymbol{\theta}_{c,\boldsymbol{\eta}}) \odot \mathcal{M}_{c,d}(\mathbf{d}; \boldsymbol{\theta}_{c,d}).$$

Each of these MINNs is built with a single mesh-informed layer:

- $L_1^{c,d}$: $V_{t,h} \xrightarrow{r=0.5} V_{t,h}$, activation function $\rho_d(x) = 0.4\ \tanh(x)$;
- $L_1^{c,\eta}$: $V_{t,h} \xrightarrow{r=0.5} V_{t,h}$, activation function $\rho_\eta(x) = 0.05\ \tanh(x)$.

The outputs are the vectors $\mathbf{z}_{\mathbf{d}} = \mathcal{M}_{c,d}(\mathbf{d}) \in \mathbb{R}^{N_h}$ and $\mathbf{z}_{\boldsymbol{\eta}} = \mathcal{M}_{c,\eta}(\boldsymbol{\eta}) \in \mathbb{R}^{N_h}$, providing the following POD-MINN+ approximation:

$$\widetilde{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h} = \mathbb{V}\widehat{\mathbf{u}}_{\boldsymbol{\mu},rb} + \widetilde{\mathbf{u}}_{\boldsymbol{\mu},c} = \mathbb{V}\mathbb{X}_{\boldsymbol{\mu}_m}\mathbf{y}_{\boldsymbol{\mu}_M} + (\mathbf{z}_{\boldsymbol{\eta}} \odot \mathbf{z}_{\mathbf{d}}) \approx \mathbf{u}_{\boldsymbol{\mu},h}.$$

To train the closure model the following loss function is minimized:

$$\mathcal{E}(\mathcal{M}_c; \boldsymbol{\theta}_c) = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \Bigg(\|\big(\mathbb{S}(:,i) - \mathbb{V}\mathcal{M}_{rb}(\boldsymbol{\mu}^{(i)}; \boldsymbol{\theta}_{rb})\big) - \mathcal{M}_c(\boldsymbol{\mu}_m^{(i)}; \boldsymbol{\theta}_c)\|_{2,N_h} + \\ \|\big(\mathbb{S}(:,i) - \mathbb{V}\mathcal{M}_{rb}(\boldsymbol{\mu}^{(i)}; \boldsymbol{\theta}_{rb})\big) - \mathcal{M}_c(\boldsymbol{\mu}_m^{(i)}; \boldsymbol{\theta}_c)\|_{\infty,N_h} \Bigg),$$

where we have introduced a regularization term to adjust the loss to control the reconstruction error of the solution in the L_∞ norm.

The training consists of at most 200 epochs with the same early stopping criterion previously mentioned. Also in this case we rely on the L-BFGS optimizer (learning rate equal to 1 and no batching). Thanks to the fact that the closure model builds upon the POD-MINN method that is already trained, the weights and biases collected into the vector $\boldsymbol{\theta}_c$ are initialized to zero.

It is crucial to stress that for this real-life problem with complex input data, an approach based on training directly a parameter-to-solution neural network (based on MINNs) is not feasible. The combination of the linear projection achieved by the POD-MINN followed by the closure model is the key feature for the success of the method. We need firstly to perform a POD projection into a reduced basis space, so that the closure model acts as a correction of an already trained model.

5.3.3. Computational performance

The speed up provided by the POD-MINN (with or without closure) is significant, unlocking approaches that would be unfeasible for the full order model and allowing to employ the POD-MINN+ model as a surrogate for the numerical solver of the microcirculation problem. The trained parameter-to-solution POD-MINN+ model is able to compute a solution in approximately 0.001 seconds, while a single run of the FOM requires a higher wall computational time, that varies from 15 to 35 minutes, depending on the density of the embedded 1D microstructure.

The gain with respect to the FOM is relevant also if we include the training times. In average, the training time of the POD/MINN method oscillates between 50 and 100 seconds, depending on the number of basis functions (but the trend is not monotone). The training time for the closure model is higher, although the optimization process consists of a correction of a trained model, and it is almost independent of the number of the POD basis functions, varying from 244 to 276 seconds.

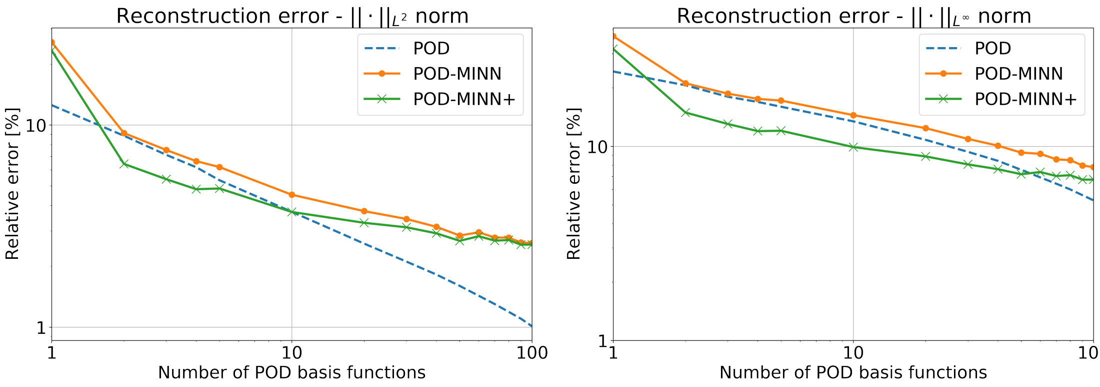


Figure 10: These plots compare the error $E_{POD-MINN}$ and $E_{POD-MINN+}$ defined in (18), to the one of the POD projection defined as E_{POD} in (10), measured in the norms $p = 2, \infty$ computed for the oxygen transfer problem (3). The case with $p = 2$ is reported on the left, the one with $p = \infty$ is on the right. The plots are shown in logarithmic scale.

5.4. Numerical results: comparing linear and nonlinear model order reduction

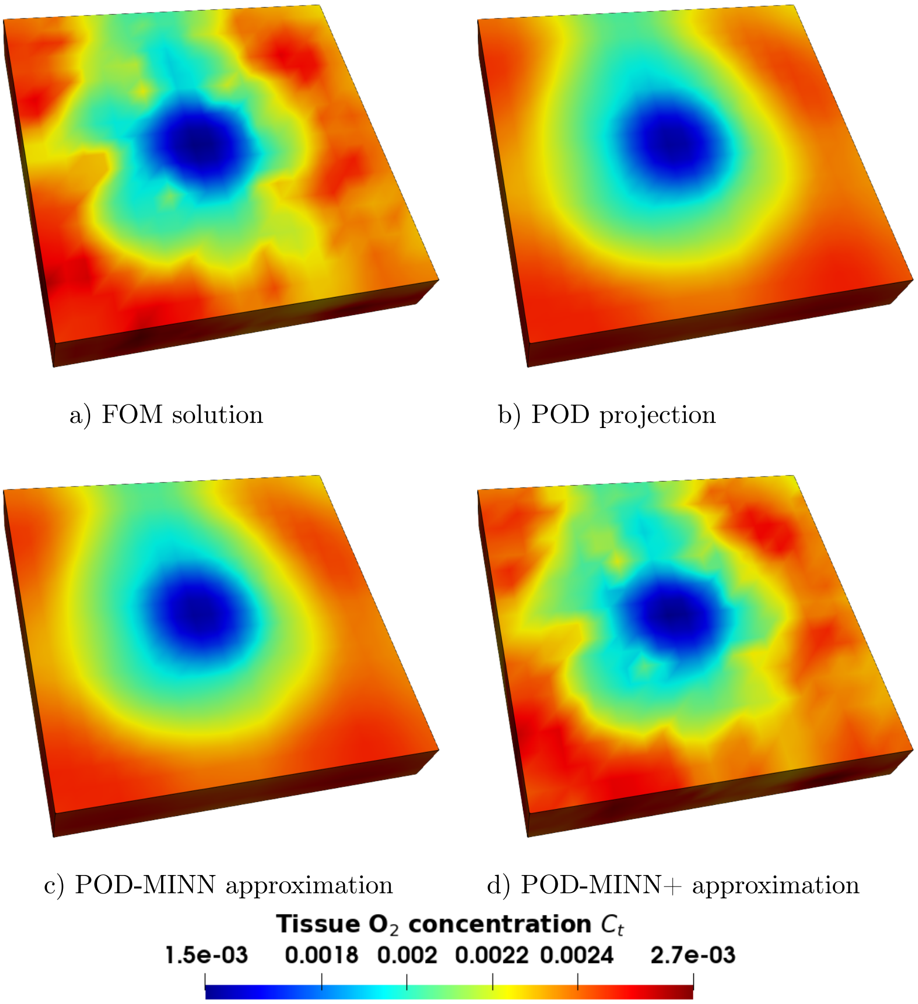
In order to measure the approximation properties of the POD-MINN+ method, we analyze the POD projection error E_{POD}, defined as in (10), and the reconstruction errors $E_{PODMINN}$ and $E_{PODMINN+}$. We compare the behaviour of each error varying the number of POD basis functions n_{rb} in Figure 10.

From a first analysis, we notice the good performance of both methods, being able to capture featuring good approximation results for the L^2 norm of the errors with few POD basis functions. Indeed oxygen transfer is diffusion dominated and consequently it is well suited for an approximation methods based on POD. This is confirmed by the fast decay of the singular values of $\mathbb{S}$ described before. On the other hand, a significant difference is detected between the L^2 and the L^∞ norm, as the POD-MINN+ shows overall higher accuracy with respect to POD and POD-MINN. The latter approach fails to retrieve the local effects due to the microscale data. Instead, the closure model is able to exploit the information about the microscale geometry to improve the smallest scales, provided that they are resolved by the FOM.

The efficacy of the POD/MINN method is more significant with low number of POD basis functions, as the closure model approximates the high-frequency modes that are neglected at POD level: see Table 2 to gain more comprehensive quantitative insights about each proposed method.

	Test Errors					
	5 **POD Modes**		10 **POD Modes**		20 **POD Modes**	
Method	$\|\cdot\|_{2,N_h}$	$\|\cdot\|_{\infty,N_h}$	$\|\cdot\|_{2,N_h}$	$\|\cdot\|_{\infty,N_h}$	$\|\cdot\|_{2,N_h}$	$\|\cdot\|_{\infty,N_h}$
POD	5.33%	16.00%	3.72%	13.45%	2.59%	10.80%
POD-MINN	6.19%	17.21%	4.52%	14.48%	3.75%	12.41%
POD-MINN+	4.85%	12.03%	3.71%	9.91%	3.28%	8.87%

Table 2: Comparison of POD, POD-MINN, and POD-MINN+ with respect to different choices of the number of reduced basis functions. We report the errors $\|\mathbf{u}_{\boldsymbol{\mu},h} - \widehat{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h}\|_{p,N_h}$ and $\|\mathbf{u}_{\boldsymbol{\mu},h} - \widetilde{\mathbf{u}}^{rb}_{\boldsymbol{\mu},h}\|_{p,N_h}$, for the POD-MINN and POD-MINN+ respectively, in the Euclidean ($p = 2$) and maximum norm ($p = \infty$).



a) FOM solution

b) POD projection

c) POD-MINN approximation

d) POD-MINN+ approximation

Figure 11: Plots of the high-fidelity FOM solution in a) is compared with the corresponding POD projection in b) and the reconstructions with the POD-MINN and POD-MINN+ approaches in c) and d) respectively: through the former method, given a small number of POD basis functions ($n_{rb} = 10$), it is possible to retrieve only the global features of the solution at the macroscale, while the local effects associated to the macroscale are captured thanks to the closure model.

This analysis becomes more evident when we compare in Figure 11 the visualization of a FOM solution with the three reconstructions, obtained respectively using POD, POD-MINN and POD-MINN+. It is clear that the POD and POD-MINN methods are over-diffusive, failing in representing the details of the solution at the smallest scale that are filtered out due to the projection on the reduced basis space of only $n_{rb} = 10$ basis functions. On the other hand, the POD-MINN+ ensures the microstructures are correctly captured, and yields a much more appreciable description of the true behaviour of the solution over the entire domain.

6. Conclusions

Reduced order modeling plays a crucial role in approximating the behaviour of continuous media with microstructure. These complex systems exhibit intricate geometries and intricate physics, making their direct numerical simulation computationally expensive and time-consuming.

Reduced order modeling offers an efficient approach to capture the essential features of the system while significantly reducing computational costs. However, well established techniques such as proper orthogonal decomposition (POD) and reduced basis methods, exploiting the projection of functions belonging to high-dimensional spaces onto a low-dimensional subspace, fail in preserving the typical high frequency modes of the true solution in presence of a microstructure. On the other hand, neural networks possess a remarkable ability to approximate functions in high dimensions. This flexibility allows neural networks to capture intricate patterns and relationships even in high-dimensional spaces. Leveraging on these properties, we have developed a correction of the POD-Galerkin approximation that restores the fine scale details into the reduced solution.

We perform this task in two steps: first we use MINNs to approximate the map from the parameters of the problem to the reduced POD coefficients, yielding the POD-MINN method; second we enhance the approach by including an additive closure model, that is a correction term ultimately providing the POD-MINN+ method. The whole procedure can be defined and successfully trained thanks to a new family of neural network architectures, called mesh-informed neural networks (MINNs), which take advantage of the mesh structure by considering connectivity information between neighboring points or elements in the mesh. This information can be used to design specialized layers that exploit the spatial relationships within the mesh. By incorporating the mesh structure into the network architecture, the model can learn more effectively from the data and capture spatial dependencies that would be difficult to capture using traditional neural network architectures.

The resulting ROM is accurate, efficient, and non intrusive, thus applicable to many scenarios where a FOM capable to capture the effect of microstructures is available but extremely expensive to query. The POD-MINN+ strategy not only accelerates simulations, but is also potentially able to enhance parametric studies such as sensitivity analysis and uncertainty quantification, making it a valuable tool for understanding the role of microstructure in many physical systems.

Acknowledgments

The present research is part of the activities of project Dipartimento di Eccellenza 2023-2027, funded by MUR, and of project FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence). The authors are members of Gruppo Nazionale per il Calcolo Scientifico (GNCS) of Istituto Nazionale di Alta Matematica (INdAM).

References

[1] W. K. El-Bouri, S. J. Payne, Multi-scale homogenization of blood flow in 3-dimensional human cerebral microvascular networks, Journal of Theoretical Biology 380 (2015) 40 – 47. doi:10.1016/j.jtbi.2015.05.011.

[2] E. Vidotto, T. Koch, T. Köppl, R. Helmig, B. Wohlmuth, Hybrid models for simulating blood flow in microvascular networks, Multiscale Modeling and Simulation 17 (3) (2019) 1076 – 1102. doi:10.1137/18M1228712.

[3] L. Possenti, S. di Gregorio, F. M. Gerosa, G. Raimondi, G. Casagrande, M. L. Costantino, P. Zunino, A computational model for microcirculation including fahraeus-lindqvist effect, plasma skimming and fluid exchange with the tissue interstitium, International Journal for Numerical Methods in Biomedical Engineering 35 (3) (2019). doi:10.1002/cnm.3165.

[4] C. Meier, M. J. Grill, W. A. Wall, A. Popp, Geometrically exact beam elements and smooth contact schemes for the modeling of fiber-based materials and structures, International Journal of Solids and Structures 154 (2018) 124 – 146. doi:10.1016/j.ijsolstr.2017.07.020.

[5] A. Fumagalli, A. Scotti, A numerical method for two-phase flow in fractured porous media with non-matching grids, Advances in Water Resources 62 (2013) 454 – 464. doi:10.1016/j.advwatres.2013.04.001.

[6] S. Tschisgale, B. Löhrer, R. Meller, J. Fröhlich, Large eddy simulation of the fluid-structure interaction in an abstracted aquatic canopy consisting of flexible blades, Journal of Fluid Mechanics 916 (2021). doi:10.1017/jfm.2020.858.

[7] N. Hagmeyer, M. Mayr, I. Steinbrecher, A. Popp, One-way coupled fluid–beam interaction: capturing the effect of embedded slender bodies on global fluid flow and vice versa, Advanced Modeling and Simulation in Engineering Sciences 9 (1) (2022). doi:10.1186/s40323-022-00222-y.

[8] C. D'Angelo, A. Quarteroni, On the coupling of 1d and 3d diffusion-reaction equations. application to tissue perfusion problems, Mathematical Models and Methods in Applied Sciences 18 (8) (2008) 1481 – 1504. doi:10.1142/S0218202508003108.

[9] M. Kuchta, F. Laurino, K.-A. Mardal, P. Zunino, Analysis and approximation of mixed-dimensional pdes on 3d-1d domains coupled with lagrange multipliers, SIAM Journal on Numerical Analysis 59 (1) (2021) 558 – 582. doi:10.1137/20M1329664.

[10] Y. Mori, A. Rodenberg, D. Spirn, Well-posedness and global behavior of the peskin problem of an immersed elastic filament in stokes flow, Communications on Pure and Applied Mathematics 72 (5) (2019) 887 – 980. doi:10.1002/cpa.21802.

[11] J. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer, 2016. doi:10.1007/978-3-319-22470-1.

[12] A. Quarteroni, A. Manzoni, F. Negri, Reduced basis methods for partial differential equations: An introduction, Springer, 2015. doi:10.1007/978-3-319-15431-2.

[13] M. Barrault, Y. Maday, N. Nguyen, A. Patera, An 'empirical interpolation'method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique 339 (9) (2004) 667–672. doi:10.1016/j.crma.2004.08.006.

[14] S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (5) (2010) 2737–2764. doi:10.1137/090766498.

[15] F. Negri, A. Manzoni, D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, Journal of Computational Physics 303 (2015) 431–454. doi:10.1016/j.jcp.2015.09.046.

[16] D. Bonomi, A. Manzoni, A. Quarteroni, A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics, Computer Methods in Applied Mechanics and Engineering 324 (2017) 300–326. doi:10.1016/j.cma.2017.06.011.

[17] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, International journal for numerical methods in engineering 102 (5) (2015) 1077–1110. doi:10.1002/nme.4820.

[18] C. Farhat, S. Grimberg, A. Manzoni, A. Quarteroni, Computational bottlenecks for PROMs: precomputation and hyperreduction, in: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. Silveira (Eds.), Snapshot-Based Methods and Algorithms, De Gruyter, 2020, pp. 181–244. doi:10.1515/9783110671490-005.

[19] C. Audouze, F. De Vuyst, P. Nair, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, Numerical Methods for Partial Differential Equations 29 (5) (2013) 1587–1628. doi:10.1002/num.21768.

[20] X. Sun, X. Pan, J. Choi, Non-intrusive framework of reduced-order modeling based on proper orthogonal decomposition and polynomial chaos expansion, Journal of Computational and Applied Mathematics 390 (2021) 113372. doi:10.1016/j.cam.2020.113372.

[21] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, Journal of Computational Physics 363 (2018) 55–78. doi:10.1016/j.jcp.2018.02.037.

[22] Z. Gao, Q. Liu, J. Hesthaven, B. Wang, W. Don, X. Wen, Non-intrusive reduced order modeling of convection dominated flows using artificial neural networks with application to rayleigh-taylor instability, Commun. Comput. Phys. 30 (1) (2021) 97–123. doi:10.4208/CICP.OA-2020-0064.

[23] M. Salvador, L. Dede', A. Manzoni, Non intrusive reduced order modeling of parametrized pdes by kernel pod and neural networks, Computers & Mathematics with Applications 104 (2021) 1–13. doi:10.1016/j.camwa.2021.11.001.

[24] M. Guo, J. Hesthaven, Reduced order modeling for nonlinear structural analysis using gaussian process regression, Computer methods in applied mechanics and engineering 341 (2018) 807–826. doi:10.1016/j.cma.2018.07.017.

[25] M. Guo, J. Hesthaven, Data-driven reduced order modeling for time-dependent problems, Computer Methods in Applied Mechanics and Engineering 345 (2019) 75–99. doi:https://doi.org/10.1016/j.cma.2018.10.029.

[26] M. Kast, M. Guo, J. Hesthaven, A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems, Computer Methods in Applied Mechanics and Engineering 364 (2020) 112947. doi:10.1016/j.cma.2020.112947.

[27] Z. Zhang, M. Guo, J. Hesthaven, Model order reduction for large-scale structures with local nonlinearities, Computer Methods in Applied Mechanics and Engineering 353 (2019) 491–515. doi:10.1016/j.cma.2019.04.042.

[28] S. Walton, O. Hassan, K. Morgan, Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions, Applied Mathematical Modelling 37 (20-21) (2013) 8930–8945. doi:10.1016/j.apm.2013.04.025.

[29] D. Xiao, F. Fang, C. Pain, I. Navon, A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications, Computer Methods in Applied Mechanics and Engineering 317 (2017) 868–889. doi:10.1016/j.cma.2016.12.033.

[30] Q. Wang, J. Hesthaven, D. Ray, Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem, Journal of computational physics 384 (2019) 289–307. doi:10.1016/j.jcp.2019.01.031.

[31] R. Swischuk, L. Mainini, B. Peherstorfer, K. Willcox, Projection-based model reduction: Formulations for physics-based machine learning, Computers & Fluids 179 (2019) 704–717. doi:10.1016/j.compfluid.2018.07.021.

[32] J. Yu, C. Yan, M. Guo, Non-intrusive reduced-order modeling for fluid problems: A brief review, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 233 (16) (2019) 5896–5912. doi:10.1177/0954410019890721.

[33] A. Bērziņš, J. Helmig, F. Key, S. Elgeti, Standardized non-intrusive reduced order modeling using different regression models with application to complex flow problems, arXiv preprint arXiv:2006.13706 (2020).

[34] M. Guo, A. Manzoni, M. Amendt, P. Conti, J. Hesthaven, Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities, Computer methods in applied mechanics and engineering 389 (2022) 114378. doi:10.1016/j.cma.2021.114378.

[35] P. Conti, M. Guo, A. Manzoni, J. Hesthaven, Multi-fidelity surrogate modeling using long short-term memory networks, Computer Methods in Applied Mechanics and Engineering 404 (2023) 115811. doi:10.1016/j.cma.2022.115811.

[36] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Computer Methods in Applied Mechanics and Engineering 306 (2016) 196–215. doi:10.1016/j.cma.2016.03.025.

[37] M. Guo, S. McQuarrie, K. Willcox, Bayesian operator inference for data-driven reduced-order modeling, Computer Methods in Applied Mechanics and Engineering 402 (2022) 115336. doi:10.1016/j.cma.2022.115336.

[38] S. Fresca, L. Dede', A. Manzoni, A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes, Journal of Scientific Computing 87 (2) (2021) 1–36. doi:10.1007/s10915-021-01462-7.

[39] S. Fresca, A. Manzoni, POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition, Computer Methods in Applied Mechanics and Engineering 388 (2022) 114181. doi:10.1016/j.cma.2021.114181.

[40] N. Franco, A. Manzoni, P. Zunino, A deep learning approach to reduced order modelling of parameter dependent partial differential equations, Mathematics of Computation 92 (340) (2023) 483–524. doi:10.1090/mcom/3781.

[41] N. Botteghi, M. Guo, C. Brune, Deep kernel learning of dynamical models from high-dimensional noisy data, Scientific Reports 12 (1) (2022) 21530. doi:10.1038/s41598-022-25362-4.

[42] H. Gao, L. Sun, J.-X. Wang, Phygeonet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain, Journal of Computational Physics 428 (2021) 110079. doi:10.1016/j.jcp.2020.110079.

[43] N. R. Franco, A. Manzoni, P. Zunino, Mesh-informed neural networks for operator learning in finite element spaces, arXiv preprint arXiv:2203.11648 (2022).

[44] L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, Nature machine intelligence 3 (3) (2021) 218–229. doi:10.1038/s42256-021-00302-5.

[45] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895 (2020).

[46] S. Boyaval, Reduced-basis approach for homogenization beyond the periodic setting, Multiscale Modeling & Simulation 7 (1) (2008) 466–494. doi:10.1137/070688791.

[47] A. Abdulle, Y. Bai, Reduced basis finite element heterogeneous multiscale method for high-order discretizations of elliptic homogenization problems, Journal of Computational Physics 231 (21) (2012) 7014–7036. doi:10.1016/j.jcp.2012.02.019.

[48] A. Abdulle, Y. Bai, G. Vilmart, An offline–online homogenization strategy to solve quasilinear two-scale problems at the cost of one-scale problems, International Journal for Numerical Methods in Engineering 99 (7) (2014) 469–486. doi:10.1002/nme.4682.

[49] L. Possenti, A. Cicchetti, R. Rosati, D. Cerroni, M. L. Costantino, T. Rancati, P. Zunino, A mesoscale computational model for microvascular oxygen transfer, Annals of Biomedical Engineering 29 (12) (2021) 3356–3373. doi:10.1007/s10439-021-02807-x.

[50] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM review 57 (4) (2015) 483–531. doi:10.1137/130932715.

[51] B. Peherstorfer, Breaking the kolmogorov barrier with nonlinear model reduction, Notices of the American Mathematical Society 69 (5) (2022) 725–733. doi:10.1090/noti2475.

[52] G. Kutyniok, P. Petersen, M. Raslan, R. Schneider, A theoretical analysis of deep neural networks and parametric pdes, Constructive Approximation 55 (2022) 73–125. doi:10.1007/s00365-021-09551-4.

[53] C. Schwab, J. Zech, Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in uq, Analysis and Applications 17 (07 2018). doi:10.1142/S0219530518500203.

[54] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117. doi:https://doi.org/10.1016/j.neunet.2014.09.003.

[55] D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Mathematical Programming 45 (1) (1989) 503–528. doi:10.1007/BF01589116.

[56] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2017). arXiv:1412.6980.

[57] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Journal of Machine Learning Research - Proceedings Track 9 (2010) 249–256.

[58] P. Vitullo, L. Cicci, L. Possenti, A. Coclite, M. L. Costantino, P. Zunino, Sensitivity analysis of a multi-physics model for the vascular microenvironment, International Journal for Numerical Methods in Biomedical Engineering (2023). doi:10.1002/cnm.3752.

[59] K. Fukami, K. Fukagata, K. Taira, Super-resolution analysis via machine learning: a survey for fluid flows, Theoretical and Computational Fluid Dynamics (2023). doi:10.1007/s00162-023-00663-0.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

66/2023 Fresca, S.; Gobat, G.; Fedeli, P.; Frangi, A.; Manzoni, A.
Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures

67/2023 Conti, P.; Guo, M.; Manzoni, A.; Hesthaven, J.S.
Multi-fidelity surrogate modeling using long short-term memory networks

65/2023 Gatti, F.; de Falco, C.; Perotto, S.; Formaggia, L.; Pastor, M.
A scalable well-balanced numerical scheme for the modelling of two-phase shallow granular landslide consolidation

Gatti, F.; de Falco, C.; Perotto, S.; Formaggia, L.; Pastor, M.
A scalable well-balanced numerical scheme for the modeling of two-phase shallow granular landslide consolidation

64/2023 Heltai, L.; Zunino, P.
Reduced Lagrange multiplier approach for non-matching coupling of mixed-dimensional domains

63/2023 Antonietti, P. F.; Matalon, P.; Verani M.
Iterative solution to the biharmonic equation in mixed form discretized by the Hybrid High-Order method

62/2023 Zappon, E.; Manzoni, A.; Quarteroni, A.
A staggered-in-time and non-conforming-in-space numerical framework for realistic cardiac electrophysiology outputs

61/2023 Afriaca, P.C.A; Piersanti, R.; Regazzoni, F.; Bucelli, M.; Salvador, M.; Fedele, M.; Pagani, S.; Dede', L.; Quarteroni, A.
lifex-ep: a robust and efficient software for cardiac electrophysiology simulations

60/2023 Negrini G.; Parolini N.; Verani M.
The Rhie-Chow stabilized Box Method for the Stokes problem

59/2023 Corti, M.; Bonizzoni, F.; Antonietti, P.F.
Structure Preserving Polytopal Discontinuous Galerkin Methods for the Numerical Modeling of Neurodegenerative Diseases