CSMA 2026
17ème Colloque National en Calcul des Structures
18-22 Mai 2026, Presqu'île de Giens (Var)

NN-PGD for surrogate modeling of PDEs on parametrized domains

Kateřina Škardová[1,a], Alexandre Daby-Seesaram[1,b], Martin Genet[1,c]

[1] *Laboratoire de mécanique des solides, CNRS, École polytechnique, Institut Polytechnique de Paris, Palaiseau, France*
[a] *katerina.skardova@polytechnique.edu*
[b] *alexandre.daby-seesaram@polytechnique.edu*
[c] *martin.genet@polytechnique.edu*

Résumé — This work presents an extension of the Neural Network–Proper Generalised Decomposition (NN-PGD) framework for constructing surrogate models of PDEs defined on parametrized domains. Using a mapping onto a reference domain, the method allows a single model to provide solutions across a range of geometries. The framework combines the PGD with physics-informed training, enabling the modes to be learned directly from the governing equations. The approach is demonstrated on a 2D linear elasticity problem on a parametrized hexagonal domain.
Mots clés — Reduced-order modelling, FEM, Physics-informed machine learning, PGD

1 Introduction

Porous materials appear in a wide range of natural and engineered systems ; at the tissue scale, many organs such as the lung [1] or liver [2] can be modelled as porous media. Understanding the mechanics of such materials often requires linking microscopic features, such as pore size, to the resulting macroscopic behaviour. Detailed micromechanical modelling provides a way to study how these micro-scale characteristics influence the global response, but this approach is computationally demanding. A common strategy to alleviate this cost is to rely on simplified parametric microstructures that retain essential morphological features. However, even in such cases, each configuration – for instance, each value of porosity – still requires generating a new computational domain and solving new problem.

Reduced-order modelling (ROM) techniques offer a powerful way to accelerate the micro–macro coupling by reducing the computational burden associated with repeated microstructural simulations [3, 4]. Most classical ROM methods rely on solving the high-fidelity problem for a set of selected parameter values, producing so-called snapshots. These snapshots are then compressed into a reduced-order basis, onto which the governing equations are subsequently projected. The resulting reduced system has a significantly lower dimensionality and can therefore be solved more efficiently for new parameter configurations. The main example of this *a posteriori* approach is the Proper Orthogonal Decomposition (POD) [5]. However, the quality of the reduced model is highly dependent on the selection of snapshot set, and the generation of these snapshots requires substantial offline computational effort.

In contrast, *a priori* reduction methods such as the Proper Generalised Decomposition (PGD) [6, 7] build the reduced-order basis dynamically without the need for precomputed high-fidelity solutions. Once the modes have been constructed – commonly referred to as the "offline phase" – the "online phase" consists solely of evaluating the surrogate model for new parameter values. This online stage thus requires no further solution of the low-dimensional problems, enabling truly real-time evaluations.

We aim to construct a surrogate model for micro-scale simulations across domains with varying geometries. In this work, we present an extension of the existing NN-PGD framework [8], combining principles of Physics-Informed Neural Networks (PINNs) [9] with PGD, that allows the handling of parametrized geometries. Benefiting from back-propagation and optimisers used in machine learning applications, NN-PGD presents a new way to find the PGD tensor decomposition. In this contribution, we describe the method and present preliminary results of the resulting surrogate model.

2 Problem setting

The proposed method is demonstrated on a two-dimensional linear elasticity problem. The governing equations read

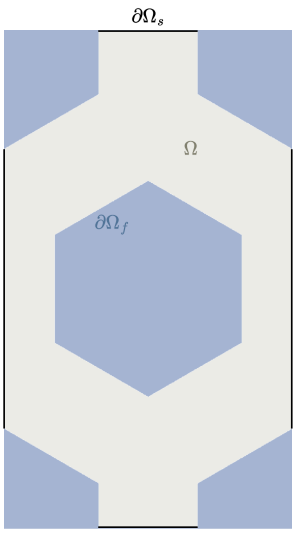
$$\begin{aligned} \nabla \cdot \underline{\underline{\sigma}}(\underline{u}) + \underline{f} = 0 \quad & \text{in } \Omega, \\ \underline{\underline{\sigma}}^T = \underline{\underline{\sigma}} \quad & \text{in } \Omega, \\ \underline{n} \cdot \underline{\underline{\sigma}}(\underline{u}) = \underline{t} \quad & \text{on } \partial\Omega_N, \\ \underline{u} = \underline{u}_D \quad & \text{on } \partial\Omega_d, \end{aligned}$$

where $\underline{u}$ is displacement, $\underline{\underline{\sigma}}$ is the Cauchy stress tensor, $\underline{f}$ is the prescribed body force, $\underline{u}_D$ and $\underline{t}$ are the displacement and surface force prescribed on the respective segments of the domain boundary. With the assumption of linear elasticity, the stress-strain relation is $\underline{\underline{\sigma}} = \underline{\underline{\underline{\underline{C}}}} : \underline{\underline{\varepsilon}}(\underline{u})$, where $\underline{\underline{\varepsilon}}(\underline{u}) := (\nabla \underline{u} + (\nabla \underline{u})^T)/2$ and $\underline{\underline{\underline{\underline{C}}}}$ is the stiffness tensor.

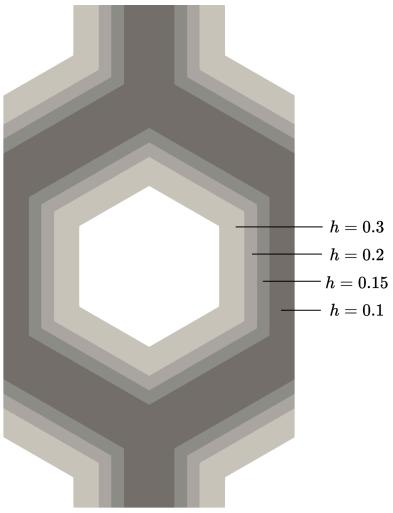
2.1 Reference & actual domain

We consider a domain $\Omega \subset \mathbb{R}^2$ designed as a cell containing hexagonal inclusions, inspired by the microstructure studied in [10]. The boundary of the domain consists of two types of interfaces : the fluid–solid interfaces, denoted by $\partial\Omega_f$, and the solid–solid interfaces on the outer boundary of the cell, denoted by $\partial\Omega_s$, as illustrated in Figure 1a.

Our objective is to develop a surrogate model capable of predicting the mechanical response of this microstructure for any porosity within a prescribed range. The variation in porosity is introduced by modifying a single parameter, the wall thickness h, which characterises the shape of the domain. Several examples of the geometry for different values of h are shown in Figure 1b.



(a) Reference domain Ω_R.



(b) Domain geometries for different values of parameters h.

We aim to construct a single surrogate model that can be evaluated across all considered porosities. To achieve this, the geometry of any actual domain associated with a specific porosity is mapped onto a single reference domain, denoted by Ω_R. This strategy allows us to discretize only the reference domain and use it to perform all computations. For the simple geometries considered in this study, an analytical mapping φ can be defined to transform the reference domain Ω_R into the actual domain Ω_A corresponding to a given wall thickness parameter h :

$$\varphi : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2 \tag{1}$$

3 Methods

3.1 Recall on Hierarchical Deep Learning Neural Network (HiDeNN)

The proposed method builds on the Hierarchical Deep Learning Neural Network (HiDeNN) framework [11]. HiDeNN provides a unified formulation in which finite element interpolation is embedded directly into the architecture of the neural network. In this setting, both the nodal coordinates and the nodal values appear as trainable parameters, allowing them to be optimized simultaneously.

This unified approach offers several advantages. First, the network architecture is very sparse and fully determined by the discretization of the problem. Second, all model parameters remain interpretable, a property that can be effectively leveraged in transfer learning strategies – such as the multi-level training procedure introduced in [12].

3.2 Recall on Neural Network–Proper Generalised Decomposition (NN-PGD)

This section summarises the Neural Network–Proper Generalised Decomposition (NN-PGD) framework introduced in our previous work [8]. NN-PGD is an efficient surrogate-modelling strategy for parametrised problems. The formulation follows the principles of classical PGD, in which the solution of a parametrised PDE is approximatedby a set of modes. Specifically, a displacement field $\underline{u}$ – a function of position $\underline{x}$ and a set of β parameters $\{\mu_i\}_{i=1}^{\beta}$ – is approximated by a reduced-order expression using m modes :

$$\underline{u}\big(\underline{x}, \{\mu_i\}_{i=1}^{\beta}\big) = \sum_{i=1}^{m} \underline{u}_i(\underline{x}) \prod_{j=1}^{\beta} \lambda_j^{(i)}(\mu_j) \tag{2}$$

In the NN-PGD framework, each PGD mode is represented by an individual HiDeNN network. The method learns these modes directly from the governing PDE, without the need for precomputed snapshots. Furthermore, the number of modes is not fixed *a priori*; instead, it increases during training until a prescribed convergence criterion is met. The greedy algorithm used to construct the PGD modes – and thereby adapt the network architecture on the fly – was detailed in [8]. In principle, representation of PGD modes by individual networks allows each mode to have different level of mesh resolution. In this work, the discretization is fixed and shared by all modes.

3.3 Loss function on actual and reference domain

Various types of physics-informed loss functions can be employed within the framework, as demonstrated in [12]. In this work, we adopt a loss function based on the potential energy of the system. For parametric PDEs, the loss to be minimized is defined as the potential energy averaged over the parametric space $\mathcal{B}$:

$$\mathcal{L} := \int_{\mathcal{B}} E_{\mathrm{p}}\Big(\underline{u}, \{\mu_i\}_{i=1}^{\beta}\Big)\, \mathrm{d}\beta. \tag{3}$$

The potential energy formulated on the actual domain Ω_A reads :

$$E_p = \frac{1}{2}\int_{\Omega_A} \underline{\underline{\varepsilon}}(\underline{u}(\underline{x}_A)) : \underline{\underline{C}} : \underline{\underline{\varepsilon}}(\underline{u}(\underline{x}_A))\, \mathrm{d}V_A - \int_{\Omega_A} \underline{f}(\underline{x}_A)\cdot\underline{u}(\underline{x}_A)\, \mathrm{d}V_A - \int_{\partial\Omega_{A,N}} \underline{t}(\underline{x}_A)\cdot\underline{u}(\underline{x}_A)\, \mathrm{d}S_A. \tag{4}$$

Our goal is to train a single framework that can be evaluated for different values of the parameter h, corresponding to different geometries. Since only the reference domain is discretized with a mesh, the loss must be expressed and evaluated on this reference domain. The mapping between the reference and actual domains, as well as its gradient $\underline{\underline{F}}_{\varphi}$, is assumed to be known. Using this mapping, $\varphi : \Omega_R \to \Omega_A$,

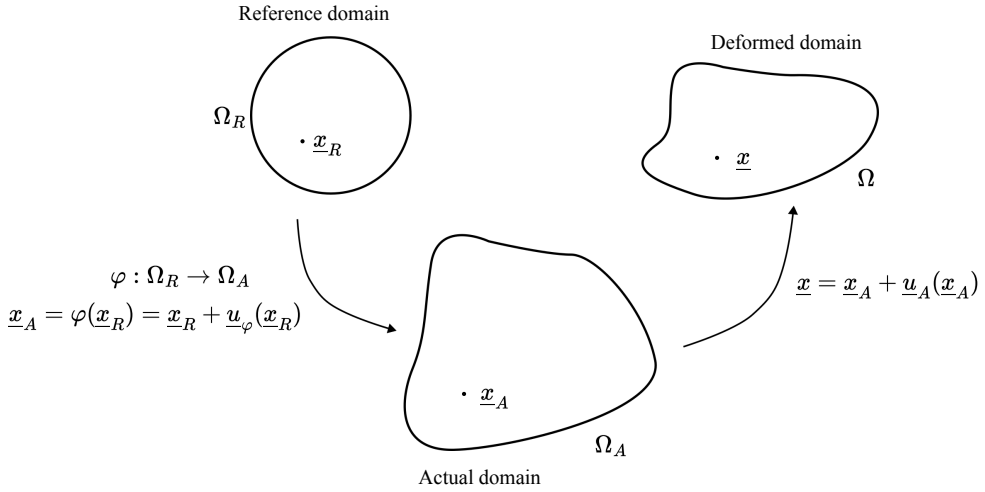


FIGURE 2 – Mapping between the reference and actual domain.

i.e. $\underline{x}_A = \varphi(\underline{x}_R)$ as illustrated in Figure 2, the integral can be rewritten over the reference domain Ω_R as :

$$E_p = \frac{1}{2}\int_{\Omega_R} \underline{\underline{\sigma}}(\underline{u}(\varphi(\underline{x}_R))) : \underline{\underline{\varepsilon}}(\underline{u}(\varphi(\underline{x}_R)))\, J_\varphi(\underline{x}_R)\, \mathrm{d}V_R \tag{5}$$

$$- \int_{\Omega_R} \underline{f}(\varphi(\underline{x}_R)) \cdot \underline{u}(\varphi(\underline{x}_R))\, J_\varphi(x_R)\, \mathrm{d}V_R \tag{6}$$

$$- \int_{\partial\Omega_{R,N}} \underline{t}(\varphi(x_R)) \cdot \underline{u}(\varphi(\underline{x}_R))\, J_{\varphi_\partial}(\underline{x}_R)\, \mathrm{d}S_R, \tag{7}$$

where $J_\varphi(x_R) = \det(\nabla\varphi(x_R))$ is the Jacobian of the transformation. Applying the chain rule to $\underline{u}_R(x_R) = \underline{u}_A(\varphi(x_R))$ yields :

$$\nabla_{x_R}\underline{u}_R = \nabla_{x_A}\underline{u}_A\, \nabla_{x_R}\varphi, \tag{8}$$

Denoting the mapping gradient $\underline{\underline{F}}_\varphi = \nabla_{x_R}\varphi(x_R)$, we can express the gradient with respect to the actual coordinates as :

$$\nabla_{x_A}\underline{u}_A(x_A) = \nabla_{x_R}\underline{u}_R(x_R) \cdot \underline{\underline{F}}_\varphi^{-1}, \tag{9}$$

Consequently, the strain tensor $\underline{\underline{\varepsilon}}(\underline{u}(\varphi(\underline{x}_R)))$, in shortened notation $\varepsilon(\underline{u}_A)$, transforms according to

$$\varepsilon(\underline{u}_A) = \frac{1}{2}\left[\nabla_{x_R}\underline{u}_R \cdot \underline{\underline{F}}_\varphi^{-1} + \left(\underline{\underline{F}}_\varphi^{-1}\right)^T \cdot \nabla_{x_R}\underline{u}_R^T\right]. \tag{10}$$

4 Numerical results

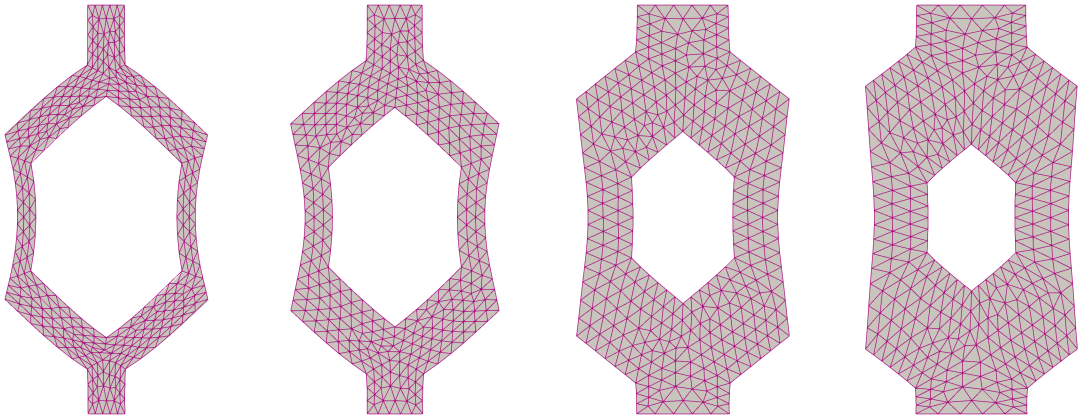
For the numerical examples, we consider a hexagonal domain Ω. The bottom boundary is fixed, and a displacement in the y-direction is prescribed on the top boundary, as illustrated in Figure 3. The prescribed displacement is set to $u_D = 0.1$ *mm*. Zero-traction boundary conditions are applied on the remaining boundaries. The material is modeled as linearly elastic, with Young's modulus $E = 1,\mathrm{kPa}$ and Poisson's ratio $\nu = 0.3$.

In this section, we first demonstrate the equivalence of the training process when the loss is evaluated on the actual versus the reference domain, and then we assess the accuracy of the resulting surrogate model.

4.1 Integration on the actual & reference domain

We compare the training in two settings : using the actual domain Ω_A generated for a three specific value of h, and using the reference domain Ω_R. The training was performed using the PyTorch implementation of the L-BFGS algorithm. The deformations obtained as a result of the training on the reference and actual domain are the same, as depicted in Figure 4.

FIGURE 3 – Setting of the problem.



(a) $h = 0.1$ (b) $h = 0.15$ (c) $h = 0.25$ (d) $h = 0.3$

FIGURE 4 – Comparison of the displacement computed on the actual domain (mesh) and reference domain (solid) for three values of parameter h.

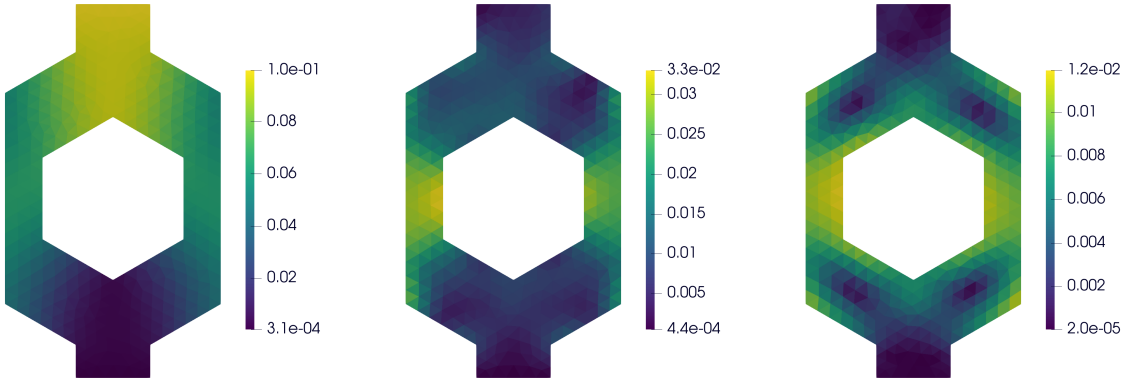
4.2 Accuracy of the surrogate model

The surrogate model was trained over the parameter range $H = [[0.1, 0.3]$. The training was conducted in two stages. In the first stage, the Adam optimizer was used to determine the optimal number of modes and, consequently, the architecture of the framework. In the second stage, the L-BFGS optimizer was employed to fine-tune the network parameters, without adding additional modes. The resulting surrogate model consists of three modes. The magnitudes of the space modes are shown in Figure 5.

A visual comparison between the solutions computed for multiple values of the parameter h and the corresponding evaluations from the surrogate model is presented in Figure 6. The normalized error between $\underline{u}$ – the solution of a model trained for specific h – and the surrogate model prediction $\underline{u}_{\text{ROM}}$ is defined as :

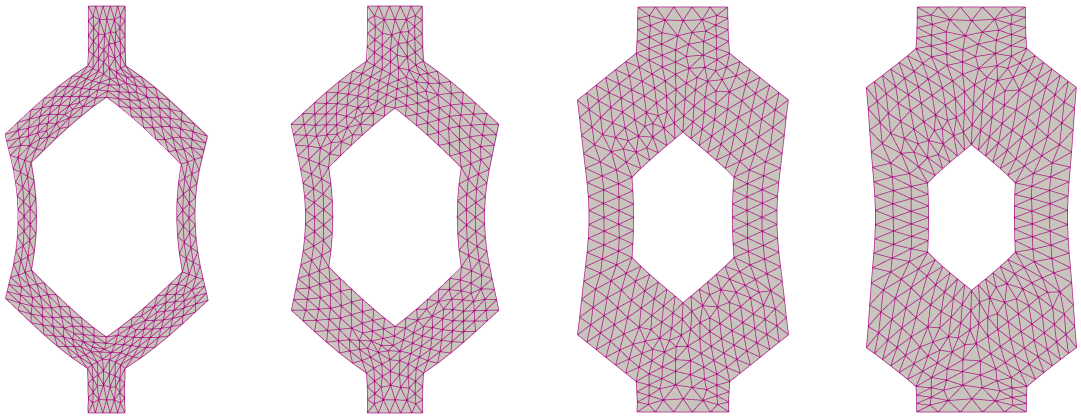
$$e_{\text{norm}} = \frac{\|\underline{u}_{\text{ROM}} - \underline{u}\|_2}{\|\underline{u}\|_2}. \tag{11}$$

The normalized error of the surrogate model is reported in Table 1. While the error varies across different values of h throughout the considered parameter range, it remains bellow 3%.



(a) Mode 1 (b) Mode 2 (c) Mode 3

FIGURE 5 – Magnitude of the four modes of the surrogate model.



(a) $h = 0.1$ (b) $h = 0.15$ (c) $h = 0.25$ (d) $h = 0.3$

FIGURE 6 – Comparison between the model trained for specific value of h and the surrogate model evaluated for this value of h.

5 Discussion & Perspectives

The proposed extension of the NN-PGD framework enables the efficient solution of PDEs defined on domains with parametrized geometries. By mapping all actual domains onto a single reference domain, the method avoids repeated mesh generation and allows for the training of a single surrogate model that is valid across a family of domain shapes. The numerical results presented in this study confirm that evaluating the loss on the actual or reference domain produces identical results. The resulting surrogate model achieves an average normalized error bellow 3 %, which is a promising preliminary outcome. An important feature of the NN-PGD framework is its ability to automatically determine the necessary number of PGD modes during training. Moreover, the surrogate model remains interpretable thanks to the HiDeNN-based representation of each mode, providing a advantage over fully black-box neural network architectures.

In the current study, the mapping φ between domains is analytical and specifically constructed for the simple hexagonal structure considered. Extending this approach to more complex geometries will require additional steps, such as shape registration, and may involve an increased number of geometric parameters beyond the single wall-thickness parameter h used here.

Future work will focus on improving the accuracy of the surrogate model by incorporating multi-level training strategies presented in [12, 8], extending the method to nonlinear elasticity, including non-geometrical parameters such as varying loads, and ultimately integrating the surrogate model into a full micro–macro computational homogenization framework.

Beyond the current micro–macro setting, the ability to handle families of geometries within a single surrogate model is also valuable at the macroscopic scale. In many applications, domain shapes naturally

	0.1	0.15	2	0.25	0.3
Normalized error	$2.7993 \cdot 10^{-2}$	$2.5398 \cdot 10^{-2}$	$1.6927 \cdot 10^{-2}$	$8.5778 \cdot 10^{-3}$	$5.9043 \cdot 10^{-3}$

TABLE 1 – Normalized displacement error for multiple value of h and solution.

vary across individuals or configurations. In such contexts, constructing separate models for each geometry might be inefficient. A surrogate model capable of accommodating geometric variability through a unified reference-domain formulation therefore offers a powerful alternative.

6 Conclusion

This work presents an extension of the NN-PGD surrogate modeling framework that enables solving PDEs on shapes of domain through a mapping to a single reference domain. The proposed approach eliminates the need for generating new geometries for each parameter value. By combining PGD, physics-informed training, and the interpretable HiDeNN architecture, the framework learns the required spatial and parametric modes directly from the governing equations.

The method was demonstrated on a 2D linear elasticity problem with a parametrized hexagonal domain. The resulting surrogate model successfully predicts displacement fields across a range of porosities, achieving an average normalized error of approximately 5 %. Future developments will focus on extending the method to more complex geometries, incorporating nonlinear physics, and integrating the surrogate model into full micro–macro computational homogenization procedures.

Références

[1] Mahdi Manoochehrtayebi, Martin Genet, and Aline Bel-Brunon. Micro-poro-mechanical modeling of the lung parenchyma : Theoretical modeling and parameters identification. *Journal of Biomechanical Engineering*, 148(1) :011001, 2026.

[2] Stéphanie Marchesseau, Simon Chatelin, and Hervé Delingette. Nonlinear biomechanical model of the liver. In *Biomechanics of living organs*, pages 243–265. Elsevier, 2017.

[3] Julien Yvonnet and Q-C He. The reduced model multiscale method (r3m) for the non-linear homogenization of hyperelastic media at finite strains. *Journal of Computational Physics*, 223(1) :341–368, 2007.

[4] Jean-Claude Michel, Hervé Moulinec, and Pierre Suquet. Effective properties of composite materials with periodic microstructure : a computational approach. *Computer methods in applied mechanics and engineering*, 172(1-4) :109–143, 1999.

[5] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, pages 808–817, 2000.

[6] Francisco Chinesta, Pierre Ladeveze, and Elias Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4) :395–404, 2011.

[7] Siamak Niroomandi, David González, Icíar Alfaro, Felipe Bordeu, Adrien Leygue, Elías Cueto, and Francisco Chinesta. Real-time simulation of biological soft tissues : a pgd approach. *International journal for numerical methods in biomedical engineering*, 29(5) :586–600, 2013.

[8] Alexandre Daby-Seesaram, Kateřina Škardová, and Martin Genet. Finite element neural network interpolation : Part II—hybridisation with the proper generalised decomposition for non-linear surrogate modelling. *Computational Mechanics*, August 2025.

[9] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks : A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378 :686–707, 2019.

[10] Mahdi Manoochehrtayebi, Aline Bel-Brunon, and Martin Genet. Finite strain micro-poro-mechanics : Formulation and compared analysis with macro-poro-mechanics. *International Journal of Solids and Structures*, 317 :113354, July 2025.

[11] Lei Zhang, Lin Cheng, Hengyang Li, Jiaying Gao, Cheng Yu, Reno Domel, Yang Yang, Shaoqiang Tang, and Wing Kam Liu. Hierarchical deep-learning neural networks : finite elements and beyond. *Computational Mechanics*, 67(1) :207–230, January 2021.

[12] Kateřina Škardová, Alexandre Daby-Seesaram, and Martin Genet. Finite element neural network interpolation : Part I—interpretable and adaptive discretization for solving PDEs. *Computational Mechanics*, August 2025.