A multiscale method for periodic structures using domain decomposition and ECM-hyperreduction

J.A. Hernández[a,b,*]

[a] *Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Technical University of Catalonia, Edificio C1, Campus Norte, Jordi Girona 1-3, Barcelona 08034, Spain*

[b] *E.S. d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa, Technical University of Catalonia, C/ Colom, 11, Terrassa 08222, Spain*

Abstract

This paper presents a nonlinear multiscale approach for periodic structures in the quasi-static, small strain regime. The approach consists in combining a domain decomposition method in which interface conditions are established through a "fictitious" frame with reduced-order modeling (ROM). We propose to approximate interface displacements, subdomain displacements and Lagrange Multipliers as linear combinations of reduced sets of dominant modes, and to assign to the coefficients of such linear combinations the role of coarse-scale displacements, strains and stresses, respectively. We discuss and propose a method based on *subspace rotations* that ensures that these three modal approximations lead to stable formulations. The modes of such expansions are determined in an offline "training" stage by applying the truncated *Singular Value Decomposition* (SVD) to the solutions obtained from Finite Element (FE) analyses. In contrast to other multiscale approaches, such as *FE-ROM* homogenization schemes, which uses a single unit cell with periodic conditions, here the "training" structures are formed by several unit cells. This way, we allow the SVD to extract also dominant patterns of interaction between subdomains in terms of reactive forces and deformations. This original feature confers to the proposed approach three unique advantages over FE-ROM homogenization schemes, namely: 1) It can deal with unit cells of arbitrary size (no need for scale separation). 2) It can model, not only how forces are transmitted through the structure, but also *local effects*. 3) It can also handle domains which are only periodic along one or two directions (beam-like or shell structures, respectively). To deal with material nonlinearities, element-wise Gauss integration of reduced internal forces is replaced by an algorithmically improved version of the *hyperreduction* scheme recently proposed by the author elsewhere, called the *Empirical Cubature Method* (ECM). Furthermore, we demonstrate that the coarse-scale Degrees of Freedom (DOFs) of a subdomain can be expressed directly in terms of the (fine-scale) stresses and strains at the ECM integration points. This feature dispenses with the need of deploying a special algorithmic infrastructure of intertwined local/global problems, as it occurs in FE-ROM schemes, since the unit cell can be treated as a special type of finite element, in which the centroids of the interfaces and the ECM points play the role of nodes and Gauss points, respectively. To illustrate all these advantages, we present 4 distinct examples: two beam-like structures of rectangular-shape and I-shaped cross-sections, a 2D hexagonal cellular material, and a cylindrical shell made of a porous composite cell. In all 4 cases the proposed method is able to produce coarse-scale models reducing the number of DOFs and integration points by over two orders of magnitude, with errors below 5 %. Interestingly, in the case of the beam-like structures, we show that the method provides 2-node beam finite elements whose kinematics are identical to that predicted by "analytical" beam theories (6 DOFs per node in the case of the rectangular beam), and totally consistent with their 3D full-order counterparts.

Keywords: multiscale, reduced-order modeling, Singular Value Decomposition, SVD, Empirical Cubature Method, ECM, Hyperreduction, Data Driven, Machine learning, Subspace rotation, Domain Decomposition

1. Introduction

Hierarchical multiscale approaches [28] are based on the general principle of *divide-and-conquer*: the domain at the highest scale is divided into smaller subdomains; then, if possible, these subdomains are further subdivided into smaller regions, and so on until arriving at the deepest level of the hierarchy, at which no more partitioning is possible, and

*Corresponding author
Email address: jhortega@cimne.upc.edu (J.A. Hernández)

recourse to experimentally derived relationships is to be made. However, this divide-and-conquer strategy *per se* does not lead to any reduction in complexity —the total number of unknowns remains the same. The actual feature that distinguishes a hierarchical multiscale model from a standard domain decomposition problem is the introduction of *simplifying assumptions in the equations governing the interaction between contiguous subdomains* at each scale. In structural problems, these simplifying assumptions translate into *low-dimensional parametrization* of the deformation and force information transmitted between contiguous subdomains. Compatibility and equilibrium equations at the immediate coarser scale are then formulated in terms of this *reduced* set of force and deformation variables.

1.1. Low-dimensional parametrization of interface forces and deformations

To illustrate this point, consider the case of first- and second-order *homogenization* of *periodic* heterogeneous materials (see Ref. [17] for a survey on these two methods). It may be argued that *first-order homogenization* represents a limiting case of two-scale method in which the subdomains (the repeating *unit cells*) are, loosely speaking, so small in comparison with the characteristic length of the coarse scale that they can be regarded as *infinitesimal* and, therefore, be treated as points in the classical sense of Cauchy's continuum mechanics. This hypothesis is often referred to as *scale separation*. In assuming scale separation, one accepts as axiomatic the precepts upon which Cauchy's theory of continuum rests, namely, that the deformation and the interface forces on the subdomains are solely characterized by the components of the strain and stress tensors —i.e., by just 6 parameters each. In *second-order homogenization* theory, this scale separation condition is relaxed, and the unit cells are considered of finite size. As a consequence, not only the stresses, but also the first moment of the stresses, come into play in the *parametrization* of the interface forces. Likewise, the gradient of the strain tensor is required in the description of boundary deformations. Classical beam theory, applied to prismatic beams of constant cross-section, can be also viewed as a two-scale hierarchical model —the repeating unit cells in this case are "slices" of infinitesimal width. Simplifications such as the assumption that the faces of the slices remain plane after deformation allow one to parametrize the displacement of such faces in terms of 6 variables (three translations, and three rotations). Likewise, interface forces become describable by just 6 variables, i.e, their resultants, (axial and shear forces) and their moment resultants (bending and torsion).

1.2. Simplifications in local equilibrium problems: reduced-order modeling

Once the simplifying hypothesis regarding the way subdomains interact have been established, one is faced with the problem of solving the *local* equilibrium problems —one for each subdomain. In these equilibrium problems, the aforementioned deformation parameters play the role of inputs —in the form of Dirichlet boundary conditions—, while the interface force parameters are the outputs. In turn, these equilibrium problems may be solved *exactly*, or one may resort, again, to some kind of approximation. The FE-2 method [15], in first-order homogenization, is a prominent example of the former category, since the fine-scale problem is solved with full details using the Finite Element (FE) method. Approximate methods, on the other hand, may range from closed-form solutions, like the simplistic rule of mixtures, to more sophisticated techniques such as the Transformation Field Analysis [10] or variants thereof [16]. A class of approximated methods that is receiving increasing attention in recent years are those combining the FE method at the coarse-scale and Reduced-Order Modeling (ROM) at the fine scale [22]. Because of its relevance within the present work, we describe in the following the basic tenets of these approaches ——that we shall refer henceforth as *FE-ROM methods*.

Like classical model reduction methods for linear modal analysis, FE-ROM methods rely on the approximation of the displacement field of the unit cell as a linear combination of a few characteristic *modes*. The distinct feature here is that these modes are not obtained from the FE stiffness and mass matrices of the subdomain, but rather determined from *computational experiments* (performed in an *offline* stage): FE analyses of the unit cell subjected to representative deformation histories are conducted; and then the set of computed solutions is processed by *dimensionality reduction* tools in order to identify and unveil the most statistically dominant displacement modes. Galerkin projection of the FE equilibrium equations onto the subspace spanned by these modes leads to a reduced-order model with far fewer equations that the original FE model. To complete the reduction process, the projection has to be accompanied by an efficient integration rule, tailored to the physics of the problem —ROM with efficient integration rules are known as *Hyperreduced-Order Models* (HROM).

1.3. Overall goal

From a broader perspective, the use of the these dimensionality reduction techniques for solving the local cell problems may be viewed as a sort of *machine learning* procedure, in the sense that the *simplifications* in such local equilibrium problems are not derived following the old physical-insight based recipe, but rather are automatically "learned" from the data obtained from a set of representative computational experiments. To put it more emphatically, the *central task of modeling* —which is, quoting M. Ashby [1], to "unashamedly distort the inessentials in order to capture the features

that really matter"— is *delegated in this case to computer algorithms*, and therefore, it is not subject to the limitations of human judgment.

The present work attempts to move multiscale structural analysis a step closer to this machine learning paradigm. We propose a multiscale approach in which dimensionality reduction is used, not only for efficiently solving the local cell problems, but also for inferring *low-dimensional parameterizations for deformations and forces at the unit cell interfaces*. As we argued before, in standard multiscale models, such parameterizations are given *a priori*, and are predicated on hypotheses derived from experimental observations and/or purely analytical deductions. Furthermore, these hypotheses imply severe restrictions on the size of the unit cells, and ignore local effects caused by the application of external forces and/or displacement constraints.

1.4. Methodology

The idea proposed here to overcome these intrinsic limitations is to address the modeling of *periodic* structures as a *reduced-order* domain decomposition problem, the qualifier "reduced-order" meaning that interface equations (equilibrium and compatibility) and subdomain equilibrium equations are projected onto *low-dimensional subspaces* —subspaces of the FE spaces associated to the spatial discretization of the subdomains. Similarly to the above mentioned FE-ROM methods for the unit cells problems, the basis vectors (i.e., the dominant modes) for such subspaces are extracted from FE analyses using dimensionality reduction techniques —here we use for this purpose the ubiquitous *Singular Value Decomposition*, SVD. However, as opposed to the standard FE-ROM methods, in which the "training" entails FE analyses of one single unit cell under prescribed macro-deformations (using periodic conditions), in the proposed approach these FE training tests involve several subdomains, for the interest now lies in *learning characteristic patterns of collective behavior*, including the way subdomains interact with each other in terms of force and displacements.

The partitioning framework adopted in the present work is the *Localized Lagrange Multiplier* (LLM) method, proposed by Park et al. in Refs. [32, 33]. The reason for favoring this method over other partitioning techniques is that its three-field formulation provides the ideal framework for our reduced-order multiscale approach. Indeed, the salient feature of the LLM method is the use of a (fictitious) *partition frame*, whose displacement is assumed independent from the displacements of the subdomains. To "glue" such partition frame to the subdomains, Lagrange multipliers are employed, resulting in a three-field formulation (frame displacements, subdomain displacements and Lagrange multipliers). We shall approximate these three variables by modal expansions, and the amplitudes of these expansions will become the *coarse-scale* variables in our multiscale method —in particular, the amplitude of the fictitious frame displacement modes will play the role of *coarse-scale Degrees of Freedom* (DOFs).

1.5. Domain of validity

The multiscale formulation presented in what follows is valid for any periodic structure working within the small, quasi-static deformation regime. Thus, it can cope with nonlinear material behavior, but not with large rotations and/or strains. It should be emphasized that the notion of periodicity is to be understood here in a broader sense than in classical multiscale methods such as the FE2: we regard as periodic any structure formed by repetition of a given building block —the *unit cell*— either along a straight or constant-radius line (1D periodicity, Figures 1.a to 1.d); a plane or a constant-radius cylinder (2D periodicity, Figure 1.e); or along 3 mutually orthogonal directions (3D periodicity, Figure 1.f). In the case of 1D periodicity, the resulting domains can be either a typical prismatic structure (Figure 1.a), an helix structure (Figure 1.c), or, if the unit block is allowed to rotate around the directrix, twisted structures such as turbine blade or screw-like bars (Figures 1.b and 1.d, respectively).

1.6. Related works

The present paper can be deemed as a natural extension to general multiscale scenarios of the above mentioned FE-HROM approaches, pioneered by the author and co-workers in Ref. [22] in the context of structural problems with small strains and mild nonlinearities —and later refined to cover also problems with more severe nonlinearities [31, 5, 35] as well as to other engineering applications such as magneto-mechanics [4], among others. On the other hand, the idea of addressing the multiscale problem as a substructuring problem combined with model reduction is borrowed from the modal synthesis technique known as the Craig-Bampton method [9]. Indeed, similarly to the Craig-Bampton method, we do approximate the displacements of each subdomain as a linear combination of modes; however, the way the modes are obtained (we use non-bubble SVD modes), as well as the treatment of the interface conditions, are totally different in our approach from those in the Craig-Bampton technique. Furthermore, the Craig-Bampton method and its variants (cf. Ref. [34]) require the explicit computation of the full-order stiffness matrices of each subdomain, a fact that restricts its scope to linear problems. By contrast, the approach presented here is valid also when material nonlinearities are present. This key advantage is in turn provided by the aforementioned *hyperreduction*, which, in the present work, is

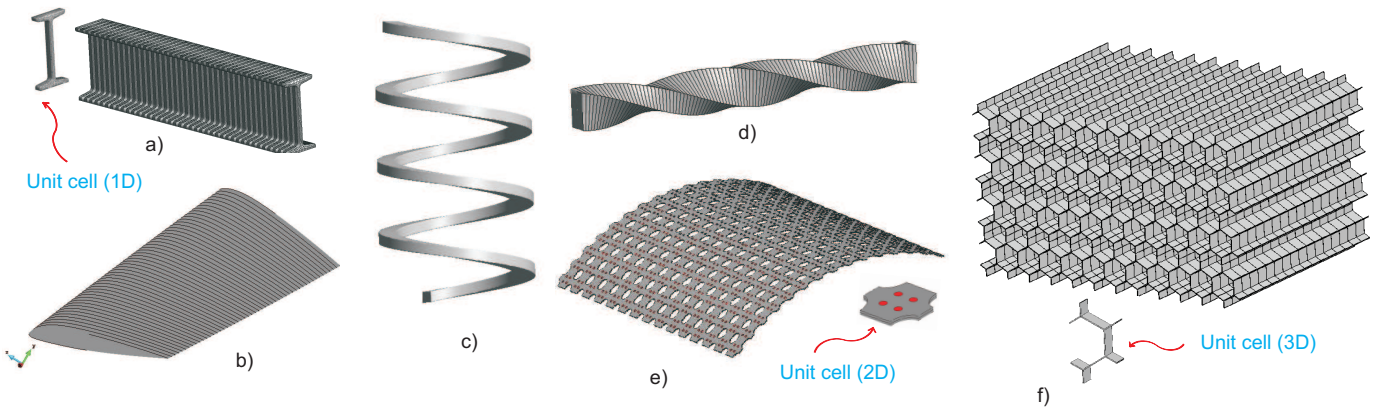

Figure 1: Class of periodic structures that can be analyzed with the proposed multiscale approach. (a) to (d) 1D periodicity, (e) 2D periodicity (shell structure) (f) 3D periodicity

applied to the efficient integration of the (reduced) internal forces of each subdomain. We employ for this integration an algorithmically improved version of the *Empirical Cubature Method* (ECM), put forward by the author in Ref. [21]. It should be mentioned that hyperreduction in the context of domain decomposition problems has been already explored by Kerfriden et al. [24]. However, Kerfriden's work does not qualify as "multiscale" in the sense defined here, because the interface conditions are resolved in its original, full-order format (the only variable subject to modal approximation in Ref. [24] is the displacement of the interior DOFs).

An approach that does qualify as multiscale, in the sense defined here, and that bears some resemblance with our proposal, is the so-called *Generalized Multiscale Finite Element Method* (GmsFEM), originally developed by Efendiev and co-workers in Ref. [11]. Efendiev's approach is a multigrid method whose major ingredients are a set of multiscale basis functions that captures fine-scale features, and a global variational formulation that couples these multiscale basis functions. As in our proposal, such multiscale basis functions are pre-computed in an offline stage by solving local problems and applying dimensionality reduction over the resulting solutions. However, Ref. [11] does not use hyperreduction to alleviate the cost of evaluating the nonlinear terms. Furthermore, the bridging between the coarse and fine scales is made by defining standard polynomial shape functions at the coarse-scale, and then making the multiscale basis functions to coincide with such shape functions on the boundaries of the coarse-grid blocks. This implies, thus, that in the GmsFEM, in the context of equilibrium problems, the *coarse-scale kinematics is defined a priori*. This is in sharp contrast to our approach, in which, as pointed out earlier, we explicitly refrain from making any assumption on the coarse-scale kinematics —rather, we let the model to "learn" from the training *data*[1] which is the appropriate kinematics for each case.

1.7. Organization of the paper

The remainder of the paper is organized as follows. Sections 2 and 3 present the employed LLM domain decomposition framework, describing the geometric setup as well as the involved fine-scale variables. In Section 4, we explain how to obtain the coarse-scale variables from the modal expansion of their fine-scale counterparts, whereas Section 5 is devoted to both the variational derivation of the corresponding coarse-scale governing equations, on the one hand, and the incorporation of the hyperreduction for the term of internal forces, on the other hand. The pseudo-code of the employed hyperreduction method (the Empirical Cubature Method) is provided in Appendix A. Section 6 is concerned with the solution strategy for the resulting set of governing, upscaling and downscaling equations. We show that such equations can be cast into a one-scale format which, remarkably, is amenable to implementation in existing finite element codes; the guidelines for adapting the implementation to such codes are provided later on, in both Section 8 and Appendix (B). In section 7, we describe thoroughly all the *offline* steps necessary for generating the training data and computing the reduced-order operators, including also the crucial issue of how to ensure the *solvability* of the coarse-scale equations. Finally, Section 9 contains a numerical assessment containing 4 distinct types of structures.

Given the length of the paper, and in order not to discourage the more programming-oriented reader, we have summarized all the implementation steps in Boxes 8.1 (offline stage) and 8.2 (online stage). The logic of the proposed methodology

[1]Incidentally, it should be noticed that the idea that the coarse-scale kinematics is not pre-defined by the user but extracted from a set of (computational) experiments is very much aligned with the spirit of the "data-driven" paradigm recently proposed by Kirchdoerfer et Ortiz in Ref. [25].

can be followed without the finer details from the information in such Boxes.

2. Geometric setup

2.1. Unit cell

Let $\Omega \subset \mathbb{R}^{n_{sd}}$ denote the domain under study ($n_{sd} = 2, 3$). As illustrated in Figure 2.a, this domain is assumed to consist of M non-overlapping, geometrically identical subdomains Ω^e ($e = 1, 2 \dots M$). This means that each Ω^e can be mapped onto a single reference subdomain $\bar{\Omega}$ —the *unit cell*—by means of translations and rotations. On the other hand, the boundary of the unit cell, denoted by $\partial\bar{\Omega}$, is assumed to be divided as follows (see Figure 2.d):

$$\partial\bar{\Omega} = \overbrace{\partial\bar{\Omega}_1 \cup \partial\bar{\Omega}_2 \cdots \partial\bar{\Omega}_l}^{\text{Interface boundaries}} \cup \partial\bar{\Omega}^{non}. \tag{1}$$

Here $\{\partial\bar{\Omega}_i\}_{i=1}^{l}$ denotes the *interface* boundaries —the boundaries that can be potentially connected to other subdomains—, while $\partial\bar{\Omega}^{non}$ designates the remaining portions —that we call *non-interface boundaries*. The number of interface boundaries can be either $l = 2$, $l = 4$ or $l = 6$ (1D, 2D o 3D periodicity, respectively). The interface boundaries are assumed to be plane and not to intersect with each other[2]($\partial\bar{\Omega}_i \cap \partial\bar{\Omega}_j = \emptyset, \quad i \neq j$). Furthermore, geometrical compatibility of contiguous subdomains (periodicity) requires that the boundary interfaces be pairwise identical, i.e.:

$$\{\partial\bar{\Omega}_i\}_{i=1}^{l} = \partial\bar{\Omega}^{+} \cup \partial\bar{\Omega}^{-}, \tag{2}$$

such that for any $\partial\bar{\Omega}_i^{+}$ ($i = 1, 2 \dots l/2$), there is a $\partial\bar{\Omega}_i^{-}$ which can be transformed into $\partial\bar{\Omega}_i^{+}$ by means of a translation and a rotation.

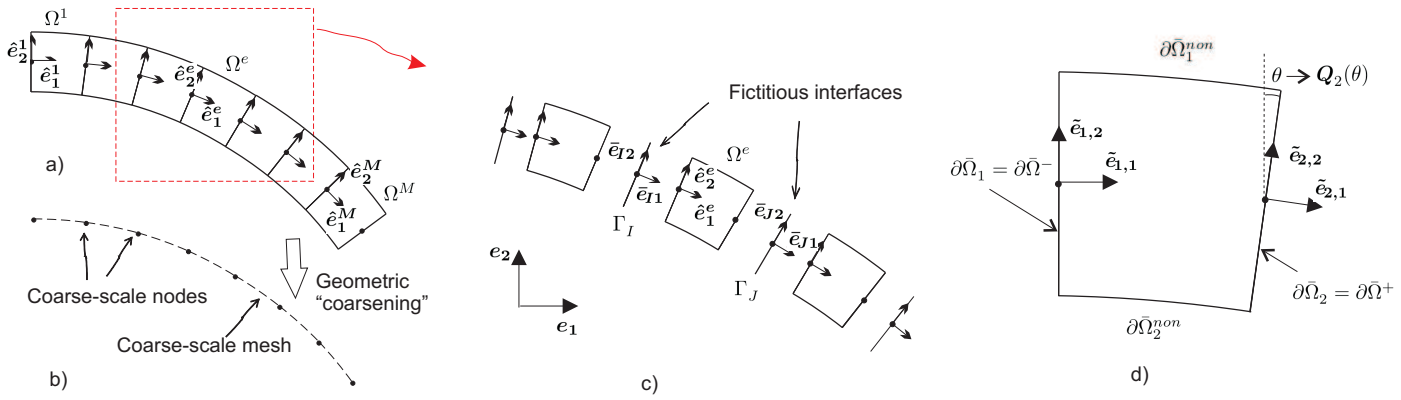

Figure 2: a) Partition of the domain into M identical domains. b) Coarse-scale representation of the structure of Figure a. The coarse-scale nodes are located at the centroids of the interface boundaries. c) Domain partition scheme. The connection between domains are established through fictitious interfaces, whose displacements are assumed to be independent from those of the neighboring domains. d) Geometric definition of the *unit cell*, showing the distinction between interface boundaries ($\partial\bar{\Omega}_1$ and $\partial\bar{\Omega}_2$) and non-interface boundaries ($\partial\bar{\Omega}_1^{non}$ and $\partial\bar{\Omega}_2^{non}$).

Two coordinate systems are employed to describe the displacement of each subdomain: a global, common system, characterized by the triad of unit orthogonal vectors $\{\boldsymbol{e_i}\}_{i=1}^{n_{sd}}$; and a local reference system intrinsic to each subdomain Ω^e, and determined by the triad $\{\hat{\boldsymbol{e}}_{\boldsymbol{i}}^{\boldsymbol{e}}\}_{i=1}^{n_{sd}}$ ($e = 1, 2 \dots M$). The vector $\hat{\boldsymbol{e}}_{\boldsymbol{1}}^{\boldsymbol{e}}$ is chosen to be normal to the interface boundary $\partial\Omega_1^e$, and pointing toward the inside of the domain —so that the normal of boundary $\partial\Omega_1^e$ is $-\hat{\boldsymbol{e}}_{\boldsymbol{1}}^{\boldsymbol{e}}$, see Figure 2.a. The coordinates of the local triad on the global reference system are stored in a rotation matrix $[\hat{\boldsymbol{Q}}^e]_{ij} = \boldsymbol{e_i}^T \hat{\boldsymbol{e}}_{\boldsymbol{j}}^{\boldsymbol{e}}$. We also attach a local coordinate system $\{\tilde{\boldsymbol{e}}_{\boldsymbol{i,j}}\}_{j=1}^{n_{sd}}$ ($i = 1, 2 \dots l$) to each of the interface boundaries of the unit cell (see Figure 2.d). The coordinates of this triad in the local reference system of the unit cell are stored in a rotation matrix $\boldsymbol{Q}_i$ ($i = 1, 2 \dots l$). By convention, we set $\tilde{\boldsymbol{e}}_{\boldsymbol{1,j}} = \hat{\boldsymbol{e}}_{\boldsymbol{j}}^{\boldsymbol{e}}$ ($j = 1, 2 \dots n_{sd}$); thus $\boldsymbol{Q}_1 = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $n_{sd} \times n_{sd}$ identity matrix. Accordingly, in 1D problems, $\boldsymbol{Q}_2$ will define the rotation of the interface boundary $\partial\bar{\Omega}_2$ with respect to $\partial\bar{\Omega}_1$, as illustrated in Figure 2.d. On the other hand, in 2D cylindrical structures, such as the one shown previously in Figure 1.e, we shall adopt the convention of labeling the interface boundaries such that $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{Q}_4$ and $\boldsymbol{Q}_3 \neq \boldsymbol{I}$ (this convention will be further explained later on, in the example of Section 9.5).

[2]This prerequisite is introduced for simplicity in dealing with the underlying domain decomposition problem (it avoids special treatment of intersection entities in 2D and 3D periodicity). In theory, this does not restrict the scope of the approach, since unit cells with continuous interface boundaries may be considered as limiting cases.

2.2. Fictitious interfaces and "coarse-scale mesh"

As pointed out in the introduction, the interface conditions between adjacent domains are not established directly but through a "fictitious" *partition frame* [33], denoted here by $\{\Gamma_I\}_{I=1}^N$ (global description), or by $\{\bar{\Gamma}_i\}_{i=1}^l$ (local description). In the undeformed configuration, the fictitious interfaces coincide with the interface boundaries of the subdomains, and their connectivity is given by an array $\boldsymbol{\mathcal{T}}$; the (e,k) entry of this array indicates the global numbering of the fictitious interface connected to the interface boundary $\partial\Omega_k^e$. For instance, in the example given in Figure 2.c, we see that $\boldsymbol{\mathcal{T}}(e,1)=I$ and $\boldsymbol{\mathcal{T}}(e,2)=J$.

The description of the interconnection is completed by providing the position in the global reference system of the centroids of each fictitious interface, denoted by $\{\mathbf{X}^*_{\boldsymbol{cI}}\}_{I=1}^N$. These position vectors, along with the connectivity array $\boldsymbol{\mathcal{T}}(\cdot,\cdot)$, constitute what we call the *coarse-scale* description of the domain Ω. This coarse-scale description, thus, ignores the exact geometry and composition of the subdomains, and only takes into consideration the position and geometry of the interfaces. Symbolically, we may represent it as the union of all fictitious interfaces: $\Gamma=\cup_{I=1}^N\Gamma_I$. Its boundary, denoted by $\partial\Gamma$, is formed by those interfaces only connected with one subdomain, i.e., those $J\in\{1,2\dots N\}$ that only appears in one row of $\boldsymbol{\mathcal{T}}$. This set is denoted by $\mathbf{b}$. For pre- and post-processing purposes, it proves advantageous to go further in this geometric "coarsening" and portray the domain as a finite element mesh in which (some of) their nodes coincide with the centroids of the interfaces —therefore ignoring as well the geometrical features of the interfaces. Accordingly, in 1D periodicity, the resulting coarse-scale mesh would be a 1D discretization of a line or curve (see Figure 2.b), while in 2D periodicity, the coarse-scale mesh would be a quadrilateral mesh (with the nodes located on the sides).

Lastly, a local coordinate system $\{\bar{\boldsymbol{e}}_{\boldsymbol{Ij}}\}_{j=1}^{n_{sd}}$ is attached to each fictitious interface Γ_I $(I=1,2\dots N)$. The coordinates of the interface triad in the global reference system are given by the columns of the matrix $[\bar{\boldsymbol{Q}}_I]_{kj}:=\boldsymbol{e_k}^T\bar{\boldsymbol{e}}_{\boldsymbol{Ij}}$. We shall assume this triad is oriented as the local reference axes for the interface boundaries of the unit cell; thus, it follows that

$$\bar{\boldsymbol{Q}}_I=\hat{\boldsymbol{Q}}^e\boldsymbol{Q}_i, \tag{3}$$

where $I=\boldsymbol{\mathcal{T}}(e,i)$.

2.3. Fine-scale mesh

The unit cell $\bar{\Omega}$ is discretized by a finite element mesh featuring n nodes and m elements. We shall refer to this spatial discretization as the *fine-scale mesh*. The DOFs of the interface boundary $\partial\bar{\Omega}_i$ are denoted by $\mathbf{f_i}$, and the vector formed by stacking all these vectors in a single column matrix by $\mathbf{f}$:

$$\mathbf{f}=\begin{bmatrix}\mathbf{f_1}^T & \mathbf{f_2}^T & \cdots & \mathbf{f_l}^T\end{bmatrix}^T. \tag{4}$$

We assume that the discretization is such that the mesh at the interface boundary $\partial\bar{\Omega}_i^+$ matches the mesh at $\partial\bar{\Omega}_i^-$ $(i=1\dots l/2)$. Furthermore, for simplicity, it is assumed that the $k-th$ entry of $\mathbf{f_i}^+$ corresponds to the $k-th$ entry of $\mathbf{f_i}^-$. The nodal coordinates of the mesh in the local reference system are denoted by $\{\boldsymbol{X'}_i\}_{i=1}^n$ $(\boldsymbol{X'}\in\mathbb{R}^{n_{sd}})$, and the positions of all the Gauss points of the mesh by $\{\boldsymbol{x}_g\}_{g=1}^{m_{gs}}$.

Likewise, the fictitious interfaces $\{\bar{\Gamma}_i\}_{i=1}^l$ are discretized using a mesh which is conforming to the discretization of the unit cell at the corresponding interface boundaries. The coordinates of the $k-th$ node of the mesh of $\bar{\Gamma}_i$ in the local system attached to the interface (located at the centroid, see Figure 2.c) is denoted by $\boldsymbol{Y'}_{i(k)}$. We assume again for simplicity that the ordering of the DOFs of the domain and fictitious interfaces is direct, meaning that the positions in the undeformed configuration of the $k-th$ node of the fictitious interface $\bar{\Gamma}_i$ and the $k-th$ node of the domain interface $\partial\bar{\Omega}_i$ coincide.

3. Fine-scale variables

Next we describe the set of variables defined on the finite element mesh of each subdomain and each fictitious interface. This set includes four nodal variables (domain displacements, Lagrange multipliers, interface displacements and external forces); and two variables defined at Gauss points (stresses and strains). In our multiscale framework, these magnitudes will play the role of *fine-scale variables*.

3.1. Domain variables

3.1.1. Displacements and Lagrange multipliers

The vectors of nodal displacements and Lagrange multipliers of domain Ω^e (expressed in the subdomain reference system), are denoted by $\boldsymbol{d}^e \in \mathbb{R}^{n_{dof}}$ and $\boldsymbol{\lambda}^e \in \mathbb{R}^{n_{dof}}$, respectively ($n_{dof} = n \cdot n_{sd}$). These vectors are partitioned as follows

$$\boldsymbol{d}^e = \begin{bmatrix} \boldsymbol{d}^e_{\boldsymbol{f}} \\ \boldsymbol{d}^e_{\boldsymbol{S}} \end{bmatrix}, \quad \text{where } \boldsymbol{d}^e_{\boldsymbol{f}} = \begin{bmatrix} \boldsymbol{d}^e_{\boldsymbol{f_1}} \\ \boldsymbol{d}^e_{\boldsymbol{f_2}} \\ \vdots \\ \boldsymbol{d}^e_{\boldsymbol{f_l}} \end{bmatrix}, \qquad \boldsymbol{\lambda}^e = \begin{bmatrix} \boldsymbol{\lambda}^e_{\boldsymbol{f}} \\ \boldsymbol{0} \end{bmatrix}, \quad \text{where } \boldsymbol{\lambda}^e_{\boldsymbol{f}} = \begin{bmatrix} \boldsymbol{\lambda}^e_{\boldsymbol{f_1}} \\ \boldsymbol{\lambda}^e_{\boldsymbol{f_2}} \\ \vdots \\ \boldsymbol{\lambda}^e_{\boldsymbol{f_l}} \end{bmatrix}. \tag{5}$$

Here, $\boldsymbol{d}^e_{\boldsymbol{f}}$ and $\boldsymbol{\lambda}^e_{\boldsymbol{f}}$ denote the nodal displacements and nodal Lagrange multipliers of the interface boundaries, while $\boldsymbol{d}^e_{\boldsymbol{S}}$ stands for the displacements of the remaining nodes —note that at such nodes the Lagrange multipliers are zero, i.e. $\boldsymbol{\lambda}^e_{\boldsymbol{S}} = \boldsymbol{0}$.

3.1.2. Strains, stresses and internal forces

The infinitesimal strain vector at the $g-th$ Gauss point of the mesh of domain Ω^e is denoted by $\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g) \in \mathbb{R}^s$ ($s = 4, 6$ for 2D and 3D problems, respectively); the relation between this vector and the nodal displacements of the domain is established through the expression:

$$\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g) = \boldsymbol{B}(\boldsymbol{x}_g)\boldsymbol{d}^e, \tag{6}$$

where $\boldsymbol{B}(\boldsymbol{x}_g)$ is the strain-displacement matrix (in its global, sparse format) associated to the $g-th$ Gauss point. On the other hand, the stress vector at Gauss point $\boldsymbol{x}_g$ is designated by $\boldsymbol{\sigma}^e(\boldsymbol{x}_g) \in \mathbb{R}^s$. Its relation to the strain vector and its past history at a given instant is assumed to be given by a general constitutive relationship, symbolically represented here as

$$\boldsymbol{\sigma}^e(\boldsymbol{x}_g) = \mathcal{H}\big(\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g), \boldsymbol{\xi}^e(\boldsymbol{x}_g); \boldsymbol{x}_g\big), \tag{7}$$

$\boldsymbol{\xi}^e(\boldsymbol{x}_g)$ being a vector of internal variables. Lastly, the vector of internal forces is computed by

$$\boldsymbol{F}^e_{\boldsymbol{int}} = \sum_{g=1}^{m_{gs}} \boldsymbol{B}^T(\boldsymbol{x}_g) W_g \boldsymbol{\sigma}^e(\boldsymbol{x}_g), \tag{8}$$

where $W_g \in \mathbb{R}$ stands for the product of the Gauss weight and the Jacobian at the Gauss point $\boldsymbol{x}_g$.

3.1.3. External forces

The vector of nodal external forces acting on subdomain Ω^e is denoted by $\boldsymbol{F}^e_{\boldsymbol{ext}}$ (expressed in the domain reference system). In the proposed partitioning framework, this vector is assumed to contain body forces, on the one hand, and, on the other hand, surface tractions acting on the non-interface boundaries $\partial\bar{\Omega}_{non}$:

$$\boldsymbol{F}^e_{\boldsymbol{ext}} = \overbrace{\boldsymbol{F}^e_{\boldsymbol{b}}}^{\text{Body forces}} + \overbrace{\boldsymbol{F}^e_{\boldsymbol{tr}}}^{\text{Tractions on } \partial\bar{\Omega}_{non}}. \tag{9}$$

As will be explained later, forces acting on interface boundaries are regarded as applied on their associated fictitious interfaces rather than on the subdomains themselves.

3.2. Fictitious interface variables

3.2.1. Interface displacements

In analogy to standard finite element formulations, we shall use global and local descriptions for referring to the vector of nodal displacements for the fictitious interfaces. On the one hand, the displacements of a given interface Γ_I ($I = 1, 2 \dots N$) is denoted by $\boldsymbol{u}_I$ —this is the global description. On the other hand, the vector of interface displacements of a given subdomain Ω^e ($e = 1, 2 \dots M$) is defined by

$$\boldsymbol{u}^e := \begin{bmatrix} \boldsymbol{u}^e_1 \\ \boldsymbol{u}^e_2 \\ \vdots \\ \boldsymbol{u}^e_l \end{bmatrix}, \tag{10}$$

(this is the local point of view). Both global and local vectors are assumed to be expressed in the reference frame attached to the interfaces. The connection between global and local descriptions is established via the coarse-scale connectivity matrix, i.e.:

$$\boldsymbol{u}_i^e = \boldsymbol{u}_I, \quad \text{if} \quad I = \boldsymbol{\mathcal{T}}(e,i). \tag{11}$$

The local point of view allow us to compactly write the compatibility conditions between subdomains and fictitious interfaces. Indeed, the difference between the displacements of the interface boundaries of a given domain and the nodal displacements of the surrounding fictitious interfaces can be expressed as $\boldsymbol{d}_{\boldsymbol{f}}^e - \boldsymbol{\mathcal{Q}}\boldsymbol{u}^e$, where

$$\boldsymbol{\mathcal{Q}} := \text{diag}\,(\boldsymbol{\mathcal{Q}}_1, \boldsymbol{\mathcal{Q}}_2, \cdots, \boldsymbol{\mathcal{Q}}_l) \tag{12}$$

and

$$\boldsymbol{\mathcal{Q}}_j = \text{diag}\,(\boldsymbol{Q}_j, \boldsymbol{Q}_j, \cdots, \boldsymbol{Q}_j). \tag{13}$$

3.2.2. Interface external forces

The interface external forces are those traction loads applied on interface boundaries (more specifically, on interface boundaries not connected with other subdomains). We denote by $\bar{\boldsymbol{F}}_J$ the nodal vector, expressed in the local reference system attached to the interface, of the traction loads acting on the interface boundary associated to Γ_J ($J \in \mathbf{b}$).

4. Low-dimensional modal approximations

4.1. Domain displacements

Now we address the low-dimensional parametrization of the fine scale variables defined in the foregoing. We begin by the domain nodal displacements $\boldsymbol{d}^e$. As is customary in partitioning problems in the small strains regime [33], the total displacements is decomposed as the sum of *rigid body and straining components* (see Figure 3.a):

$$\boldsymbol{d}^e = \overbrace{\boldsymbol{R}\boldsymbol{\alpha}^e}^{\text{Rigid body}} + \overbrace{\boldsymbol{\Phi}\boldsymbol{q}^e}^{\text{Straining}}. \tag{14}$$

Here, $\boldsymbol{R} \in \mathbb{R}^{n_{dof} \times n_{rb}}$ ($n_{rb} = 3, 6$ for 2D and 3D problems, respectively) and $\boldsymbol{\Phi} \in \mathbb{R}^{n_{dof} \times n^*}$ stand for the matrices of rigid body and *straining* modes, respectively. The number of straining modes, n^*, is assumed to be much smaller than the number of DOFs of the domain ($n^* << n_{dof}$).

The matrix of rigid body modes $\boldsymbol{R}$ can be further partitioned into translational ($\boldsymbol{R_t}$) and rotational ($\boldsymbol{R_r}$) modes. Taking the centroid $\boldsymbol{X'}_c$ as reference point, the expression for such matrices for a given node i reads (for 3D problems)

$$\boldsymbol{R_{ti}} = \boldsymbol{I}, \qquad \boldsymbol{R_{ri}} = -\mathbf{spin}(\Delta \boldsymbol{X'}_i) = -\begin{bmatrix} 0 & -(\Delta \boldsymbol{X'}_i)_3 & (\Delta \boldsymbol{X'}_i)_2 \\ (\Delta \boldsymbol{X'}_i)_3 & 0 & -(\Delta \boldsymbol{X'}_i)_1 \\ -(\Delta \boldsymbol{X'}_i)_2 & (\Delta \boldsymbol{X'}_i)_1 & 0 \end{bmatrix} \tag{15}$$

(here $\Delta \boldsymbol{X'}_i = \boldsymbol{X'}_i - \boldsymbol{X'}_c$). The corresponding amplitudes $\boldsymbol{\alpha}_{\boldsymbol{t}}^e \in \mathbb{R}^{n_{sd}}$ and $\boldsymbol{\alpha}_{\boldsymbol{r}}^e \in \mathbb{R}^{n_{rot}}$ ($n_{rot} = 1, 3$ for 2D and 3D) are thus the translation of the centroid and the (infinitesimal) rotation angles, respectively. We further postulate that the straining modes $\boldsymbol{\Phi}$ are mutually *orthogonal*, and, besides, *M-orthogonal* to the rigid body modes:

$$\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{I}, \qquad \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{R} = \mathbf{0}, \tag{16}$$

$\boldsymbol{M}$ being the (geometric[3]) nodal mass matrix of the unit cell $\bar{\Omega}$. Furthermore

4.1.1. Coarse-scale strains

Substitution of Eq.(14) into Eq.(6) yields

$$\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g) = \boldsymbol{B}(\boldsymbol{x}_g)\boldsymbol{R}\boldsymbol{\alpha}^e + \boldsymbol{B}(\boldsymbol{x}_g)\boldsymbol{\Phi}\boldsymbol{q}^e, \qquad g = 1, 2 \ldots m_{gs}. \tag{17}$$

Since rigid body motions produce no strain, we have that $\boldsymbol{B}(\boldsymbol{x}_g)\boldsymbol{R} = \mathbf{0}$, and therefore

$$\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g) = \boldsymbol{B}^*(\boldsymbol{x}_g)\boldsymbol{q}^e, \qquad g = 1, 2 \ldots m_{gs} \tag{18}$$

[3]Similar to the standard nodal mass matrix, but with density equal to one. Using the variables defined later in Appendix B, it can be calculated as $\boldsymbol{M} = \sum_{g=1}^{m_{gs}} \boldsymbol{N}^T(\boldsymbol{x}_g) W_g \boldsymbol{N}(\boldsymbol{x}_g)$.

where

$$\boldsymbol{B}^*(\boldsymbol{x}_g) := \boldsymbol{B}(\boldsymbol{x}_g)\boldsymbol{\Phi}. \tag{19}$$

Notice that equation (18) represents a low-dimensional parametrization of the fine-scale strains of the subdomain Ω^e in terms of the amplitudes of the straining modes of such a domain. For this reason, one may interpret $\boldsymbol{q}^e$ as the vector of generalized *coarse-scale strains*. In turn, in the light of this interpretation, Eq.(18) may be viewed as a *strain downscaling equation*, for it allows us to, once $\boldsymbol{q}^e$ has been calculated, recover the (fine-scale) strains at any Gauss point of the domain Ω^e.

4.2. Lagrange multipliers (reactive forces)

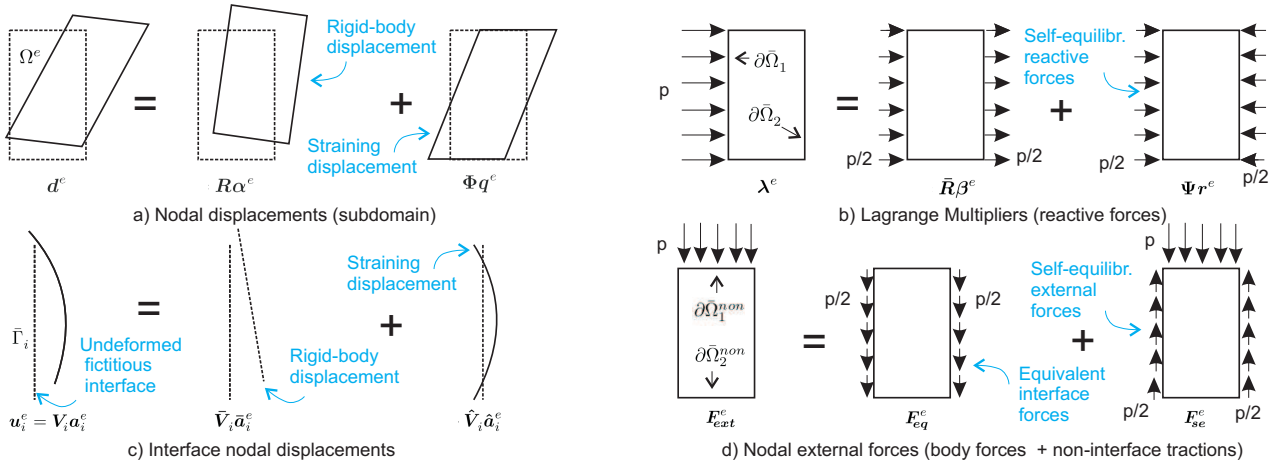

Figure 3: Proposed decompositions for: a) Nodal displacements $\boldsymbol{d}^e$ of each subdomain Ω^e into rigid-body and straining parts (see Eq. 14). The amplitude $\boldsymbol{q}^e$ of the straining modes are the generalized *coarse-scale strains* in our multiscale framework. b) Nodal reactive forces $\boldsymbol{\lambda}^e$ into resultant and self-equilibrating components (see Eq. 20). The amplitudes $\boldsymbol{r}^e$ of the self-equilibrating modes will play the role of generalized *coarse-scale stresses*. c) Nodal displacements of each fictitious interface into rigid-body and straining components (see Eq. 32). The amplitude $\boldsymbol{a}_i^e$ of both rigid-body and straining modes constitute the generalized *coarse-scale DOFs* of our model. d) Nodal external forces into equivalent interface and self-equilibrating components (see Eq. 28).

For the vector of nodal Lagrange multipliers of domain Ω^e, we propose a decomposition analogous to Eq.(14), i.e.:

$$\boldsymbol{\lambda}^e = \bar{\boldsymbol{R}}\boldsymbol{\beta}^e + \overbrace{\boldsymbol{\Psi}\boldsymbol{r}^e}^{\text{Self-equilibrating}} \tag{20}$$

where

$$\boldsymbol{R}^T\boldsymbol{\Psi} = \boldsymbol{0}, \quad \text{and} \quad \boldsymbol{\Psi}^T\boldsymbol{\Psi} = \boldsymbol{I}. \tag{21}$$

In Eq.(20), $\bar{\boldsymbol{R}}$ stands for the following matrix

$$\bar{\boldsymbol{R}} = \begin{bmatrix} \bar{\boldsymbol{R}}_{\boldsymbol{f}} \\ \bar{\boldsymbol{R}}_{\boldsymbol{S}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{\boldsymbol{f}}\boldsymbol{R}_{\boldsymbol{f}} \\ \boldsymbol{0} \end{bmatrix}, \tag{22}$$

i.e., a matrix in which the rows corresponding to the interface boundaries DOFs are equal to the rigid body modes times the mass matrix —and zero at the remaining rows. On the other hand, $\boldsymbol{\Psi} \in \mathbb{R}^{n_{dof} \times p}$ designates the *matrix of self-equilibrating reactive modes*. Like the straining modes $\boldsymbol{\Phi}$, this matrix is determined empirically from FE, full-order analyses via dimensionality reduction. The qualifier "self-equilibrating" refers to the fact that, according to condition (21.a), each column of this matrix forms a system of nodal forces whose resultant and moment resultant are zero (see Figure 3.b). This follows easily from Eq.(5) and Eq.(15); indeed, for the $i-th$ mode we have that

$$\boldsymbol{R}_{\boldsymbol{t}}^T\boldsymbol{\Psi}_{\boldsymbol{i}} = \sum_{j=1}^{n} \boldsymbol{\Psi}_{\boldsymbol{ij}} = \boldsymbol{0}, \quad \text{and} \quad \boldsymbol{R}_{\boldsymbol{r}}^T\boldsymbol{\Psi}_{\boldsymbol{i}} = -\sum_{j=1}^{n} \mathbf{spin}(\Delta\boldsymbol{X}'_j)\boldsymbol{\Psi}_{\boldsymbol{ij}} = -\sum_{j=1}^{n} \Delta\boldsymbol{X}'_j \times \boldsymbol{\Psi}_{\boldsymbol{ij}} = \boldsymbol{0}. \tag{23}$$

4.2.1. Coarse-scale stresses

In much the same way we have interpreted the amplitudes of the straining modes $\boldsymbol{q}^e$ as the generalized coarse-scale strains, we may view $\boldsymbol{r}^e$ as the vector of *generalized coarse-scale stresses* of the domain Ω^e, for they represent a *low-dimensional parametrization of the reactive forces* between Ω^e and its neighboring subdomains. In fact, the reason for enforcing the condition that $\boldsymbol{\Psi}$ must be a system of self-equilibrating forces is to ensure that $\boldsymbol{r}^e$ is, in some sense, the work conjugate of $\boldsymbol{q}^e$. Indeed, the work done by these modes over virtual rigid body displacements is zero, and therefore

$$\delta \boldsymbol{d}^{e^T} \boldsymbol{\Psi} \boldsymbol{r}^e = \delta \boldsymbol{q}^{e^T} \boldsymbol{H} \boldsymbol{r}^e, \tag{24}$$

where

$$\boldsymbol{H} := \boldsymbol{\Phi}^T \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{f}}^T & \boldsymbol{\Phi}_{\boldsymbol{S}}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_{\boldsymbol{f}} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{\Phi}_{\boldsymbol{f}}^T \boldsymbol{\Psi}_{\boldsymbol{f}} = \sum_{i=1}^{l} \boldsymbol{\Phi}_{\boldsymbol{f_i}}^T \boldsymbol{\Psi}_{\boldsymbol{f_i}}. \tag{25}$$

The matrix defined above will play a critical role in the stability and the well-posedness of the overall problem. Physically, $\boldsymbol{H}_{ij}$ represents the work done by the $j-th$ self-equilibrating force mode when the interface boundary nodes moves along the $i-th$ straining pattern. It can be argued thus that, in order to avoid having coarse-scale stresses that do not contribute to the coarse-scale work, the *kernel of this matrix must be zero*

$$\ker(\boldsymbol{H}) = \{\boldsymbol{0}\}. \tag{26}$$

This condition implies that $\texttt{rank}(\boldsymbol{H}) = p$, which means, in turn, that

$$p \leq n^*, \tag{27}$$

that is, in our multiscale formulation, the *number of coarse-scale stress components must be equal or less than the number of coarse-scale strain components.*

4.2.2. Equivalent and self-equilibrating forces

It proves convenient to introduce at this point a decomposition similar to that in Eq.(20) for the input vector of nodal external forces on domain Ω^e:

$$\boldsymbol{F}^e_{\boldsymbol{ext}} = \overbrace{\boldsymbol{F}^e_{\boldsymbol{eq}}}^{\text{Equiv. interf. forces}} + \overbrace{\boldsymbol{F}^e_{\boldsymbol{se}}}^{\text{Self-equilib.}} \tag{28}$$

where

$$\boldsymbol{F}^e_{\boldsymbol{eq}} := \bar{\boldsymbol{R}}(\boldsymbol{R}^T \bar{\boldsymbol{R}})^{-1} (\boldsymbol{R}^T \boldsymbol{F}^e_{\boldsymbol{ext}}) \tag{29}$$

and

$$\boldsymbol{F}^e_{\boldsymbol{se}} := \boldsymbol{F}^e_{\boldsymbol{ext}} - \boldsymbol{F}^e_{\boldsymbol{eq}}. \tag{30}$$

As illustrated in Figure 3.d, the term $\boldsymbol{F}^e_{\boldsymbol{eq}}$ in Eq.(29) is a vector of nodal forces which are different from zero only at the nodes of the interface boundaries (this follows from the definition of $\bar{\boldsymbol{R}}$ in Eq. 22), and whose resultant and moment resultant are identical[4] to that of $\boldsymbol{F}^e_{\boldsymbol{ext}}$, i.e.:

$$\boldsymbol{R}^T \boldsymbol{F}^e_{\boldsymbol{eq}} = \boldsymbol{R}^T \boldsymbol{F}^e_{\boldsymbol{ext}}. \tag{31}$$

For this reason, we shall refer to this variable as the vector of *equivalent interface nodal forces* on domain Ω^e. Conversely, $\boldsymbol{F}^e_{\boldsymbol{se}}$ will be called the vector of *self-equilibrating nodal forces* on domain Ω^e, since its resultant and moment resultant are zero (this is readily seen by multiplying Eq.(30) by $\boldsymbol{R}^T$).

4.3. Fictitious interface displacements

Now we address the parametrization of the nodal displacements of the fictitious interfaces of domain Ω^e ($e = 1, 2 \ldots M$). We postulate for $\boldsymbol{u}^e_i$ ($i = 1, 2 \ldots l$) a decomposition similar to the one used in Eq.(14) for domain displacements (see Figure 3.c):

$$\boldsymbol{u}^e_i = \overbrace{\bar{\boldsymbol{V}}_i \bar{\boldsymbol{a}}^e_i}^{\text{Rigid body}} + \overbrace{\hat{\boldsymbol{V}}_i \hat{\boldsymbol{a}}^e_i}^{\text{Straining}} = \boldsymbol{V}_i \boldsymbol{a}^e_i, \quad i = 1, 2 \ldots l \tag{32}$$

[4]Formally, $\boldsymbol{F}^e_{\boldsymbol{eq}}$ is an (oblique) projection of $\boldsymbol{F}_{\boldsymbol{ext}}$ onto the column space of $\bar{\boldsymbol{R}}$.

where

$$\boldsymbol{V}_i = \begin{bmatrix} \bar{\boldsymbol{V}}_i & \hat{\boldsymbol{V}}_i \end{bmatrix}, \qquad \boldsymbol{a}_i^e = \begin{bmatrix} \bar{\boldsymbol{a}}_i^e \\ \hat{\boldsymbol{a}}_i^e \end{bmatrix}. \tag{33}$$

Here $\bar{\boldsymbol{V}}_i \bar{\boldsymbol{a}}_i^e$ stands for the rigid body component, whereas $\hat{\boldsymbol{V}}_i \hat{\boldsymbol{a}}_i^e$ represents the straining part. The translational $\bar{\boldsymbol{V}}_{\boldsymbol{ti}}$ and rotational $\bar{\boldsymbol{V}}_{\boldsymbol{ri}}$ rigid body modes are calculated as in Eq.(15):

$$(\bar{\boldsymbol{V}}_{\boldsymbol{ti}})_j = \boldsymbol{I}, \qquad (\bar{\boldsymbol{V}}_{\boldsymbol{ri}})_j = -\mathbf{spin}(\boldsymbol{Y'}_{i_j} - \boldsymbol{Y'}_{ic}). \tag{34}$$

The straining modes $\hat{\boldsymbol{V}}_i$, on the other hand, are assumed to be mutually M-orthogonal, as well as M-orthogonal to the rigid body modes:

$$\hat{\boldsymbol{V}}_i^T \bar{\boldsymbol{M}}_{\boldsymbol{i}} \hat{\boldsymbol{V}}_i = \boldsymbol{I}, \qquad \hat{\boldsymbol{V}}_i^T \bar{\boldsymbol{M}}_{\boldsymbol{i}} \bar{\boldsymbol{V}}_i = \boldsymbol{0}, \qquad i = 1, 2 \dots l \tag{35}$$

where $\bar{\boldsymbol{M}}_{\boldsymbol{i}}$ denotes the "geometric" mass matrix of interface $\bar{\Gamma}_i = \partial\bar{\Omega}_i$. Lastly, periodicity dictates that the modes of $\partial\bar{\Omega}^-$ be equal to the modes of $\partial\bar{\Omega}^+$. We symbolically express this condition as

$$\boldsymbol{V}_i^+ = \boldsymbol{V}_i^- \qquad i = 1, 2 \dots l/2. \tag{36}$$

For 1D periodicity, for instance, this implies that $\boldsymbol{V}_1 = \boldsymbol{V}_2$, whereas for 2D periodicity, we have $\boldsymbol{V}_1 = \boldsymbol{V}_3$, and $\boldsymbol{V}_2 = \boldsymbol{V}_4$.

4.3.1. Coarse-scale DOFs

We pointed out in Section 2.2 that, in the proposed multiscale framework, the coarse-scale description only accounts for the kinematics of the fictitious interfaces, ignoring the subdomains. Accordingly, we shall assign to the amplitudes of the interface displacement modes the role of *coarse-scale DOFs*. The basic unknown of the coarse-scale problem will be therefore the vector formed by gathering all the interface contributions in a single column matrix:

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_N \end{bmatrix}, \tag{37}$$

where

$$\boldsymbol{a}_I = \boldsymbol{a}_i^e, \qquad \text{if } I = \boldsymbol{\mathcal{T}}(e, i). \tag{38}$$

From definition (10), it follows that the nodal fictitious displacements associated to domain Ω^e are expressible in terms of coarse-scale DOFs as

$$\boldsymbol{u}^e = \boldsymbol{V}\,\boldsymbol{a}^e, \tag{39}$$

where

$$\boldsymbol{V} := \mathrm{diag}\,(\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_l), \qquad \boldsymbol{a}^e := \begin{bmatrix} \boldsymbol{a}_1^e \\ \boldsymbol{a}_2^e \\ \vdots \\ \boldsymbol{a}_l^e \end{bmatrix}. \tag{40}$$

The vector $\boldsymbol{a}^e$ defined in the preceding equation is the *local* coarse-scale DOFs of domain Ω^e. In analogy to standard finite element formulations, we introduce a coarse-scale Boolean matrix $\mathbb{L}^e$ to relate this *local* domain vector to the *global* vector defined in Eq.(37):

$$\boldsymbol{a}^e = \mathbb{L}^e \boldsymbol{a} \tag{41}$$

5. Variational principle

Having defined the fine-scale and coarse-scale variables, the next step consists in determining the constitutive, compatibility and equilibrium equations for the coarse-scale variables, as well as the stress upscaling equation. According to the employed variational partitioning framework [33], these equations emerge from the stationary conditions of the following energy functional:

$$\Pi = \Pi_{\boldsymbol{int}} - \Pi_{\boldsymbol{ext}} - \pi. \tag{42}$$

Here, $\Pi_{\boldsymbol{int}}$ denotes the contribution of strain energy, $\Pi_{\boldsymbol{ext}}$ stands for the part due to external forces, while π is the so-called *interface* or dislocation potential. Taking variations of Eq.(42), we get

$$\delta\Pi = \delta\Pi_{\boldsymbol{int}} - \delta\Pi_{\boldsymbol{ext}} - \delta\pi. \tag{43}$$

In the following, we elaborate on each of the three terms of the right-hand side of the preceding equation.

5.1. Internal virtual work

The total internal virtual work in Eq.(43) is calculated by summing the contributions of all the subdomains:

$$\delta\Pi_{\boldsymbol{int}} = \sum_{e=1}^{M} \delta\Pi^e_{\boldsymbol{int}} \tag{44}$$

In turn, the contribution of a given subdomain Ω^e is given by $\delta\Pi^e_{\boldsymbol{int}} = \delta\boldsymbol{d}^{e^T}\boldsymbol{F}^e_{\boldsymbol{int}}$. Taking variations of Eq. (14); using Eq.(8) and Eq.(19); and considering that rigid body motions do not produce straining, we obtain

$$\begin{aligned}\delta\Pi^e_{\boldsymbol{int}} &= (\delta\boldsymbol{\alpha}^{e^T}\boldsymbol{R}^T + \delta\boldsymbol{q}^{e^T}\boldsymbol{\Phi}^T)\sum_{g=1}^{m_{gs}}\boldsymbol{B}^T(\boldsymbol{x}_g)W_g\boldsymbol{\sigma}^e(\boldsymbol{x}_g)\\ &= \delta\boldsymbol{q}^{e^T}\sum_{g=1}^{m_{gs}}\boldsymbol{B}^{*T}(\boldsymbol{x}_g)W_g\boldsymbol{\sigma}^e(\boldsymbol{x}_g).\end{aligned} \tag{45}$$

5.1.1. Empirical cubature of virtual work

Note that the evaluation of the virtual work in the preceding equation requires summing up the contribution of all the m_{gs} Gauss points of the underlying (fine-scale) mesh. This is because we are using the same integration rule as for the nodal internal forces in Eq.(8), that is, element-wise Gauss quadrature. This dependence on the total number of Gauss-points of the unit cell mesh would render the overall approach ostensibly inefficient and impractical in nonlinear scenarios, for one would be forced to track and store the strains, the stresses and the internal variables at all the Gauss points of each subdomain.

Fortunately, there is no need to use all such Gauss points for the integration. As shown by the author in Ref. [22] (in the context of multiscale homogenization problems), since displacements are constrained to lie in a subspace of dimension $n^* << n_{dof}$, then the corresponding strains and stresses are also expected to reside in subspaces of dimension $\mathcal{O}(n^*)$. This implies that the virtual work per unit volume —the product of stresses and strains— is of dimensionality $\mathcal{O}(n^{*2})$, and therefore, $\mathcal{O}(n^{*2})$ points[5] suffice to evaluate $\delta\Pi^e_{\boldsymbol{int}}$ in Eq.(45).

In the present work, such integration points are optimally chosen among the set of mesh Gauss points by means of a constrained optimization procedure that guarantees the positiveness of the weights —the *Empirical Cubature Method*, ECM for short, developed by the author in Ref. [21]. We present in Appendix A an algorithmically improved version of this ECM algorithm. Let $\mathbf{Z} \subset \{1, 2 \dots m_{gs}\}$ denote the set of m^*_{gs} indexes corresponding to the chosen points, and[6] $\{\omega(\boldsymbol{x}_g)\}_{g\in\mathbf{Z}}$ the set of corresponding positive weights. With these two variables at hand, we can approximate the virtual internal work in Eq.(45) by

$$\delta\Pi^e_{\boldsymbol{int}} = \delta\boldsymbol{q}^{e^T}\sum_{g\in\mathbf{Z}}\boldsymbol{B}^{*T}(\boldsymbol{x}_g)\omega(\boldsymbol{x}_g)\boldsymbol{\sigma}^e(\boldsymbol{x}_g). \tag{46}$$

Note that this expression is formally identical to Eq.(45); the only differences are that, on the one hand, the sum is taken over the Gauss points belonging to the set $\mathbf{Z}$, and that, on the other hand, we have replaced the FE weights W_g by the ECM weights $\omega(\boldsymbol{x}_g)$.

5.2. External virtual work

The external work term $\delta\Pi_{\boldsymbol{ext}}$ in Eq.(43) is composed of two contributions: work done by body forces and tractions applied on the non-interface boundaries (described in Section 3.1.3); and work done by tractions applied on the coarse-scale boundary ($\bar{\boldsymbol{F}}_I$), described in Section (3.2.2). Taking variations of Eq.(14) and Eq.(32), and multiplying by the corresponding forces, we can express the total virtual external work as[7]

$$\delta\Pi_{\boldsymbol{ext}} = \sum_{e=1}^{M}\delta\boldsymbol{\alpha}^{e^T}\boldsymbol{R}^T\boldsymbol{F}^e_{\boldsymbol{ext}} + \sum_{e=1}^{M}\delta\boldsymbol{q}^{e^T}\boldsymbol{\Phi}^T\boldsymbol{F}^e_{\boldsymbol{ext}} + \delta\boldsymbol{a}^T\sum_{I\in\mathbf{b}}\bar{\boldsymbol{L}}_I^T\boldsymbol{V}_I^T\bar{\boldsymbol{F}}_I \tag{47}$$

where $\bar{\boldsymbol{L}}_I$ is a boolean matrix such that $\boldsymbol{a}_I = \bar{\boldsymbol{L}}_I\boldsymbol{a}$.

[5] The number of points needed to integrate a function residing in a subspace of dimension n is of the order of $\mathcal{O}(n)$ points. For instance, for a polynomial of order p (and therefore of dimensionality $p+1$), the number of points is equal to $(p+1)/2$ when using Gauss quadrature (the optimal quadrature rule), and $p+1$ when employing Newton-Cotes.

[6] $\omega(\boldsymbol{x}_g)$ indicates the weight associated to the ECM point located at $\boldsymbol{x}_g$.

[7] With some abuse of notation, we employ the same symbol for denoting the local and global counterpart of the interface modes matrix. To distinguish them, we use small letter indexes for local description ($\boldsymbol{V}_i$), ,and capital letter indexes for global description ($\boldsymbol{V}_I$).

5.3. Interface potential

The interface potential π is defined by

$$\pi = \sum_{e=1}^{M} \pi^e = \sum_{e=1}^{M} \boldsymbol{\lambda}^{e^T} (\boldsymbol{d}_{\boldsymbol{f}}^e - \boldsymbol{\mathcal{Q}} \boldsymbol{u}^e), \tag{48}$$

Taking variations we get

$$\delta\pi^e = \delta\boldsymbol{\lambda}^{e^T} (\boldsymbol{d}_{\boldsymbol{f}}^e - \boldsymbol{\mathcal{Q}} \boldsymbol{u}^e) + (\delta\boldsymbol{d}_{\boldsymbol{f}}^{e^T} - \delta\boldsymbol{u}^{e^T} \boldsymbol{\mathcal{Q}}^T)\boldsymbol{\lambda}^e. \tag{49}$$

Next we replace the expressions for $\boldsymbol{\lambda}^e$, $\boldsymbol{d}_{\boldsymbol{f}}^e$ and $\boldsymbol{u}^e$ (Eq. 20, Eq. 14 and Eq. 39, respectively), as well as their variations, in Eq.(49); this leads, upon factoring out the variations of each variable, to

$$\begin{aligned}
\delta\pi^e = {} & \delta\boldsymbol{\alpha}^{e^T} (\boldsymbol{R}_{\boldsymbol{f}}^T \bar{\boldsymbol{R}}_{\boldsymbol{f}} \boldsymbol{\beta}^e) + \delta\boldsymbol{q}^{e^T} \left(\boldsymbol{\Phi}_{\boldsymbol{f}}^T \bar{\boldsymbol{R}}_{\boldsymbol{f}} \boldsymbol{\beta}^e + \boldsymbol{\Phi}_{\boldsymbol{f}}^T \boldsymbol{\Psi}_{\boldsymbol{f}} \boldsymbol{r}^e \right) \\
& + \delta\boldsymbol{r}^e \left(\boldsymbol{\Psi}_{\boldsymbol{f}}^T \boldsymbol{\Phi}_{\boldsymbol{f}} \boldsymbol{q}^e - \boldsymbol{\Psi}_{\boldsymbol{f}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{V} \boldsymbol{a}^e \right) \\
& + \delta\boldsymbol{\beta}^e \left(\bar{\boldsymbol{R}}_{\boldsymbol{f}}^T \boldsymbol{R}_{\boldsymbol{f}} \boldsymbol{\alpha}^e + \bar{\boldsymbol{R}}_{\boldsymbol{f}}^T \boldsymbol{\Phi}_{\boldsymbol{f}} \boldsymbol{q}^e - \bar{\boldsymbol{R}}_{\boldsymbol{f}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{V} \boldsymbol{a}^e \right) \\
& - \delta\boldsymbol{a}^{e^T} \left(\boldsymbol{V}^T \boldsymbol{\mathcal{Q}}^T \bar{\boldsymbol{R}}_{\boldsymbol{f}} \boldsymbol{\beta}^e + \boldsymbol{V}^T \boldsymbol{\mathcal{Q}}^T \boldsymbol{\Psi}_{\boldsymbol{f}} \boldsymbol{r}^e \right).
\end{aligned} \tag{50}$$

For notational compactness, we define the following reduced-order matrices

$$\bar{\boldsymbol{H}} := \boldsymbol{\Phi}_{\boldsymbol{f}}^T \bar{\boldsymbol{R}}_{\boldsymbol{f}}, \qquad \boldsymbol{G} := \boldsymbol{R}_{\boldsymbol{f}}^T \bar{\boldsymbol{R}}_{\boldsymbol{f}} \tag{51}$$

$$\boldsymbol{T} := \boldsymbol{\Psi}_{\boldsymbol{f}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{V} = \sum_{i=1}^{l} \boldsymbol{\Psi}_{\boldsymbol{f}_i}^T \boldsymbol{\mathcal{Q}}_i \boldsymbol{V}_i, \qquad \bar{\boldsymbol{T}} := \bar{\boldsymbol{R}}_{\boldsymbol{f}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{V}. \tag{52}$$

With these matrices at hand, and using Eq.(41), Eq.(25) and Eq.(50), the variation of Eq.(48) can be finally written as

$$\begin{aligned}
\delta\pi = \sum_{e=1}^{M} \delta\pi^e = {} & \sum_{e=1}^{M} \left(\delta\boldsymbol{\alpha}^{e^T} \boldsymbol{G} \boldsymbol{\beta}^e + \delta\boldsymbol{q}^{e^T} \left(\bar{\boldsymbol{H}} \boldsymbol{\beta}^e + \boldsymbol{H} \boldsymbol{r}^e \right) + \delta\boldsymbol{r}^e \left(\boldsymbol{H}^T \boldsymbol{q}^e - \boldsymbol{T} \boldsymbol{a}^e \right) \right) \\
& + \sum_{e=1}^{M} \left(\delta\boldsymbol{\beta}^e \left(\boldsymbol{G}^{e^T} \boldsymbol{\alpha}^e + \bar{\boldsymbol{H}}^T \boldsymbol{q}^e - \bar{\boldsymbol{T}} \boldsymbol{a}^e \right) - \delta\boldsymbol{a}^T \mathbb{L}^{e^T} \left(\bar{\boldsymbol{T}}^T \boldsymbol{\beta}^e + \boldsymbol{T}^T \boldsymbol{r}^e \right) \right).
\end{aligned} \tag{53}$$

5.4. Stationary points of the functional

The stationary points of the proposed functional follow from equating to zero the terms multiplying $\delta\boldsymbol{\alpha}^e$, $\delta\boldsymbol{q}^e$, $\delta\boldsymbol{r}^e$, $\delta\boldsymbol{a}$ and $\delta\boldsymbol{\beta}^e$ in Eq.(44), Eq.(47) and Eq.(53).

5.4.1. Resultants of domain external forces

We begin with the variations of the amplitudes of the rigid body domain displacement $\delta\boldsymbol{\alpha}^e$. Gathering the terms from Eq.(47) and Eq.(53), we get

$$-\boldsymbol{R}^T \boldsymbol{F}_{\boldsymbol{ext}}^e - \boldsymbol{G} \boldsymbol{\beta}^e = \boldsymbol{0}, \qquad e = 1, 2 \ldots M. \tag{54}$$

Solving the above for $\boldsymbol{\beta}^e$, and using Eq.(51).b, we obtain

$$\boldsymbol{\beta}^e = -(\boldsymbol{R}_{\boldsymbol{f}}^T \bar{\boldsymbol{R}}_{\boldsymbol{f}})^{-1} \boldsymbol{R}^T \boldsymbol{F}_{\boldsymbol{ext}}^e. \tag{55}$$

5.4.2. Coarse-scale equilibrium

As for the terms multiplying $\delta\boldsymbol{a}$, we get from Eq.(47) and Eq.(53) that

$$\sum_{e=1}^{M} \mathbb{L}^{e^T} \boldsymbol{T}^T \boldsymbol{r}^e = \overbrace{\sum_{e=1}^{M} \mathbb{L}^{e^T} \boldsymbol{\mathcal{F}}_{\boldsymbol{eq}}^e}^{\text{Equiv. interf. forces}} + \overbrace{\sum_{I \in \mathbf{b}} \bar{\boldsymbol{L}}_I^T \bar{\boldsymbol{\mathcal{F}}}_I}^{\text{Boundary forces}}. \tag{56}$$

where

$$\bar{\boldsymbol{\mathcal{F}}}_I := \boldsymbol{V}_I^T \bar{\boldsymbol{F}}_I, \qquad I \in \mathbf{b} \tag{57}$$

and

$$\boldsymbol{\mathcal{F}}^e_{\boldsymbol{eq}} := -\bar{\boldsymbol{T}}^T\boldsymbol{\beta}^e = (\boldsymbol{V}^T\boldsymbol{\mathcal{Q}}^T\bar{\boldsymbol{R}}_{\boldsymbol{f}})(\boldsymbol{R}^T_{\boldsymbol{f}}\bar{\boldsymbol{R}}_{\boldsymbol{f}})^{-1}\boldsymbol{R}^T\boldsymbol{F}^e_{\boldsymbol{ext}} = \boldsymbol{V}^T\boldsymbol{\mathcal{Q}}^T\boldsymbol{F}^e_{\boldsymbol{eq,f}}, \qquad e = 1, 2 \dots M. \tag{58}$$

Expression (56) constitutes the *coarse-scale equilibrium equation* of our multiscale problem, for it represents a balance between *coarse-scale internal forces* (depending on the generalized coarse-scale stresses $\boldsymbol{r}^e$, left-hand side) and *coarse-scale external forces* (right-hand side). Such external forces are in turn formed by assembling the contribution of two distinct terms, namely: (coarse-scale) *boundary forces* and (coarse-scale) *equivalent interface forces*. The latter is called this way because, as it follows from Eq.(58), it emanates from *upscaling* the fine-scale vector of equivalent interface nodal forces $\boldsymbol{F}^e_{\boldsymbol{eq}}$ (given in turn in Eq.(29)). Likewise, $\bar{\boldsymbol{\mathcal{F}}}_{\boldsymbol{I}}$ in Eq.(57) is called the vector of *coarse-scale boundary forces* because it arises from projecting the nodal forces $\bar{\boldsymbol{F}}_{\boldsymbol{I}}$ acting on the boundary of the coarse-scale geometry onto the span of $\boldsymbol{V}_{\boldsymbol{I}}$.

5.4.3. Stress upscaling equation

Next we group the terms corresponding to the variations of the straining amplitudes $\delta\boldsymbol{q}^e$ (from Eqs. 44, 46, 47 and 53). Upon some manipulation (using Eqs. 51, 54 and 30), we arrive at

$$\boldsymbol{H}\boldsymbol{r}^e = \sum_{g\in\mathbf{Z}} \boldsymbol{B}^{*T}(\boldsymbol{x}_g)\omega(\boldsymbol{x}_g)\boldsymbol{\sigma}^e(\boldsymbol{x}_g) - \boldsymbol{\Phi}^T\boldsymbol{F}^e_{\boldsymbol{se}}, \qquad e = 1, 2 \dots M. \tag{59}$$

The above equation is our *stress upscaling equation*, because it relates the generalized coarse-scale stresses $\boldsymbol{r}^e$ (left-hand side) to its fine-scale counterpart $\{\boldsymbol{\sigma}^e(\boldsymbol{x}_g)\}_{g\in\mathbf{Z}}$ (right-hand side). Notice that the external forces also appears in this upscaling equation —through its *self-equilibrating component* $\boldsymbol{F}^e_{\boldsymbol{se}}$, defined in Eq.(30), and pictorially depicted in Figure 3.d. It should be highlighted that this is a distinguishing feature of our approach, for standard multiscale theories, such as FE beam/shell theories or first- and second-order homogenization, ignore this self-equilibrating component, and only consider external domain forces through its equivalent interface component $\boldsymbol{F}^e_{\boldsymbol{eq}}$ in the coarse-scale equilibrium equation —hence their inability to capture *local effects*[8].

5.4.4. Coarse-scale strain-displacement equation

The only term multiplying the variations of the amplitudes of the self-equilibrating modes $\delta\boldsymbol{r}^e$ appears in Eq.(53); equating this term to zero, we obtain the relationship between *coarse-scale strains and displacements*:

$$\boldsymbol{H}^T\boldsymbol{q}^e = \boldsymbol{T}\boldsymbol{a}^e, \qquad e = 1, 2 \dots M. \tag{60}$$

5.4.5. Rigid-body amplitudes

The remaining coarse-scale kinematic condition is the relationship between displacements, strains and rigid-body motion of each subdomain. This equation can be deduced from gathering and equating to zero the terms multiplying $\delta\boldsymbol{\beta}^e$ in Eq.(53):

$$\boldsymbol{G}^T\boldsymbol{\alpha}^e + \bar{\boldsymbol{H}}^T\boldsymbol{q}^e - \bar{\boldsymbol{T}}\boldsymbol{a}^e = \boldsymbol{0}, \qquad e = 1, 2 \dots M. \tag{61}$$

Note that, since $\boldsymbol{G}$ is by construction invertible, the rigid body vector $\boldsymbol{\alpha}^e$ can be obtained a posteriori, once $\boldsymbol{q}^e$ and $\boldsymbol{a}^e$ have been determined.

6. Solution strategy

Prior to launching into the specific details of the solution strategy, let us first summarize the equations and unknowns of the multiscale problem we have described in the preceding sections. We have a total of *five* sets of unknowns, namely, the *coarse-scale* strains, stresses and displacements ($\boldsymbol{q}^e, \boldsymbol{r}^e$ and $\boldsymbol{a}$, respectively); and the *fine-scale* strains and stresses ($\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g)$, $\boldsymbol{\sigma}^e(\boldsymbol{x}_g)$, respectively). The total number of governing equations is *five* as well: 2 coarse-scale equations (equilibrium and the stress-displacements relationship, see Eqs. 56 and 60); 2 upscaling/downscaling equations (the stress upscaling and strain downscaling mappings, see Eqs. 59 and 18); and one *fine-scale* equation (the constitutive law, see Eq. 7).

[8]For instance, in beam theories, if a longitudinal "slice" is subjected to a system of lateral self-equilibrating forces, then the slice is (incorrectly) considered unstressed. By contrast, our approach can accurately deal with this type of local effects, as we shall demonstrate later in the example of Section 9.3

6.1. Hierarchical versus inter-scale approach

In principle, the strategy for solving this set of five matrix equations would be similar to that of other *hierarchical* two-scale methods such as the FE2 or the FE-HROM. Indeed, notice that four of the involved variables (namely, $\boldsymbol{q}^e, \boldsymbol{r}^e, \boldsymbol{\varepsilon}^e(\boldsymbol{x}_g)$, $\boldsymbol{\sigma}^e(\boldsymbol{x}_g)$) are *local* —they are defined at subdomain level— whereas just one of them is global (the coarse-scale DOFs $\boldsymbol{a}$). Likewise, there are 4 *local* and 1 global equations. The global equation is the coarse-scale equilibrium (56), which depends solely on the coarse-scale stresses $\boldsymbol{r}^e$. Thus, the gist of the strategy would be to, given a tentative coarse-scale displacement $\boldsymbol{a}$, solve first the *local* equations in order to determine the corresponding coarse-scale stresses $\boldsymbol{r}^e$ $(e = 1, 2 \dots M)$. If such stresses are not in equilibrium with the applied external actions, then the linearized version of the equilibrium equation would be used to determine a correction for the coarse-scale displacements —and so on until convergence.

The computer implementation of this type of hierarchical approaches is a cumbersome, laborious task —as anyone familiar with the intricacies of the FE2 will attest. The fact that stresses and strains at the scale of interest are not related through an empirical constitutive law compels one to deploy a special algorithmic infrastructure of intertwined local/global problems, wherein convergence issues are difficult to spot and debug.

Fortunately, in the proposed multiscale framework, it is possible to circumvent this nested sequence of local/global, and produce a simplified "inter-scale" formulation which, remarkably, is amenable to *implementation in a standard finite element code*. The key insight for understanding how to arrive at this simplified strategy is concealed in the condition $\texttt{rank}(\boldsymbol{H}) = p$, presented in Section 4.2.1. Physically, such a condition indicates that one may include more straining modes than reactive modes, provided that at least p of such straining modes *effectively* contribute to the total coarse-scale work. An immediate corollary is that if $n^* > p$, then there is a straining subspace of dimension $n^* - p$ whose motions are either *energetically irrelevant or redundant* to the coarse-scale[9]. In the light of this observation, it is natural to wonder whether there would be any algorithmic advantage in excluding these irrelevant or redundant modes from the straining modes matrix $\boldsymbol{\Phi}$. We next demonstrate that this question has an affirmative answer.

Suppose that, given a matrix of p self-equilibrated reactive modes $\boldsymbol{\Psi}$, we manage to determine a matrix of $n^* = p$ *effective* straining modes $\boldsymbol{\Phi}$ such that $\boldsymbol{H} = \boldsymbol{\Phi}^T\boldsymbol{\Psi}$ is invertible —we shall explain later, in Section 7.3, how to ensure this condition. In doing so, both Eq.(60) and Eq.(59) can be solved for $\boldsymbol{q}^e$ and $\boldsymbol{r}^e$, yielding

$$\boldsymbol{q}^e = \boldsymbol{H}^{-T}\boldsymbol{T}\boldsymbol{a}^e, \qquad e = 1, 2 \dots M, \tag{62}$$

and

$$\boldsymbol{r}^e = \boldsymbol{H}^{-1}\sum_{g\in\mathbf{Z}} \boldsymbol{B}^{*T}(\boldsymbol{x}_g)\omega(\boldsymbol{x}_g)\boldsymbol{\sigma}^e(\boldsymbol{x}_g) - \boldsymbol{H}^{-1}\boldsymbol{\Phi}^T\boldsymbol{F}^e_{\boldsymbol{se}}, \qquad e = 1, 2 \dots M, \tag{63}$$

respectively. If we now substitute Eq.(62) into the strain downscaling expression (18), we obtain

$$\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g) = \boldsymbol{\mathcal{B}}(\boldsymbol{x}_g)\boldsymbol{a}^e, \qquad e = 1, 2 \dots M, \tag{64}$$

where

$$\boldsymbol{\mathcal{B}}(\boldsymbol{x}_g) := \boldsymbol{B}^*(\boldsymbol{x}_g)\boldsymbol{H}^{-T}\boldsymbol{T}, \qquad g = 1, 2 \dots m_{gs}. \tag{65}$$

The above matrix will be referred to as the *interscale "B" matrix* at Gauss point $\boldsymbol{x}_g$, for it allows one to determine the fine-scale strain vector in terms of the coarse-scale DOFs of the domain. On the other hand, inserting Eq.(63) into the coarse-scale equilibrium equation (56), we arrive at

$$\sum_{e=1}^{M} \mathbb{L}^{e^T}\boldsymbol{T}^T\boldsymbol{H}^{-1}\sum_{g\in\mathbf{Z}} \boldsymbol{B}^{*T}(\boldsymbol{x}_g)\omega(\boldsymbol{x}_g)\boldsymbol{\sigma}^e(\boldsymbol{x}_g) = \sum_{e=1}^{M} \mathbb{L}^{e^T}\boldsymbol{T}^T\boldsymbol{H}^{-1}\boldsymbol{\Phi}^T\boldsymbol{F}^e_{\boldsymbol{se}} + \sum_{e=1}^{M} \mathbb{L}^{e^T}\boldsymbol{\mathcal{F}}^e_{\boldsymbol{eq}} + \sum_{I\in\mathbf{b}} \bar{\boldsymbol{L}}_I^T\bar{\boldsymbol{\mathcal{F}}}_I \tag{66}$$

Using Eq.(65), the preceding equation can be recast into a format reminiscent of that of typical finite element formulations:

$$\overbrace{\sum_{e=1}^{M} \mathbb{L}^{e^T}\boldsymbol{\mathcal{F}}^e_{\boldsymbol{int}}}^{\text{Internal forces}} = \overbrace{\sum_{e=1}^{M} \mathbb{L}^{e^T}(\boldsymbol{\mathcal{F}}^e_{\boldsymbol{se}} + \boldsymbol{\mathcal{F}}^e_{\boldsymbol{eq}})}^{\text{Body forces}} + \overbrace{\sum_{I\in\mathbf{b}} \bar{\boldsymbol{L}}_I^T\bar{\boldsymbol{\mathcal{F}}}_I}^{\text{Boundary forces}} \tag{67}$$

[9]Recall that the Lagrange multipliers are only nonzero at the DOFs of interface boundaries, while the displacement domains are non-zero at all DOFs. Accordingly, it may occur that certain strain modes only affect the interior of the domain, being its impact on interface boundary DOFS —and therefore on the coarse-scale— negligible.

where

$$\boldsymbol{\mathcal{F}}^e_{\boldsymbol{int}} := \sum_{g\in\mathbf{Z}} \boldsymbol{\mathcal{B}}^T(\boldsymbol{x}_g)\omega(\boldsymbol{x}_g)\boldsymbol{\sigma}^e(\boldsymbol{x}_g) \tag{68}$$

and

$$\boldsymbol{\mathcal{F}}^e_{\boldsymbol{se}} := (\boldsymbol{T}^T\boldsymbol{H}^{-1}\boldsymbol{\Phi}^T)\boldsymbol{F}^e_{\boldsymbol{se}} \tag{69}$$

($\boldsymbol{\mathcal{F}}^e_{\boldsymbol{eq}}$ was defined in Eq. 58).

6.2. The coarse-scale finite element

Eqs. (64) and (67) are of crucial importance in the proposed method. By simply enforcing that $\boldsymbol{H} = \boldsymbol{\Phi}^T\boldsymbol{\Psi}$ be invertible, we have managed to couch the kinematics and the equilibrium of the coarse-scale problem solely in terms of the infinitesimal strain vector $\boldsymbol{\varepsilon}^e(\boldsymbol{x}_g)$, the Cauchy stress vector $\boldsymbol{\sigma}^e(\boldsymbol{x}_g)$ (both at the ECM integration points, $g \in \mathbf{Z}$) and the vector of coarse-scale DOFs. Furthermore, both Eqs. (64) and (67) have practically the same format as those of standard finite element formulations. Thus, for implementational purposes, it proves useful to exploit this resemblance and conceptualize the unit cell as a *special type of finite element*. The main features of this coarse-scale element are summarized below.

1. The nodes are located at the centroids of the l interface boundaries.
2. The integration points of this finite element are located at the positions computed by the Empirical Cubature Method $\{\boldsymbol{x}_g \in \bar{\Omega}\}_{g\in\mathbf{Z}}$.
3. The B-matrices relating the strains at the ECM integration points to the element DOFs are given in Eq.(65). There is one of such matrices for each coarse-scale node, that is,

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix}\boldsymbol{\mathcal{B}}_1 & \boldsymbol{\mathcal{B}}_2 & \cdots & \boldsymbol{\mathcal{B}}_l\end{bmatrix}, \tag{70}$$

   where, according to Eq.(52).a:

$$\boldsymbol{\mathcal{B}}_i(\boldsymbol{x}_g) = \boldsymbol{B}^*(\boldsymbol{x}_g)\boldsymbol{H}^{-T}(\boldsymbol{\Psi}^T_{\boldsymbol{f}_i}\boldsymbol{\mathcal{Q}}_i\boldsymbol{V}_i). \tag{71}$$

4. The *element internal force* vector (see Eq. 68) is obtained as in a standard FE formulation, i.e., by summing the internal forces per unit volume ($\boldsymbol{\mathcal{B}}^T\boldsymbol{\sigma}^e$) at all the integration points. In this case, however, the volumetric weights are not the Gauss weights times the Jacobian, but the weights computed by the ECM ($\omega(\boldsymbol{x}_g), g \in \mathbf{Z}$). The global internal force vector $\boldsymbol{\mathcal{F}}_{\boldsymbol{int}}$ is computed by standard FE assembly operations (using the coarse-scale connectivity array $\boldsymbol{\mathcal{T}}(\cdot,\cdot)$).
5. The *element tangent stiffness matrix* also exhibits the same format as that of FE formulations. Indeed, linearization of Eq.(68) yields

$$\mathcal{D}_{\Delta\boldsymbol{a}^e}\boldsymbol{\mathcal{F}}^e_{\boldsymbol{int}} = \boldsymbol{\mathcal{K}}^e\Delta\boldsymbol{a}^e \tag{72}$$

   where

$$\boldsymbol{\mathcal{K}}^e := \sum_{g\in\mathbf{Z}} \omega(\boldsymbol{x}_g)\boldsymbol{\mathcal{B}}^T(\boldsymbol{x}_g)\boldsymbol{C}^e(\boldsymbol{x}_g)\boldsymbol{\mathcal{B}}(\boldsymbol{x}_g) \tag{73}$$

   (here $\boldsymbol{C}^e(\boldsymbol{x}_g)$ stands for the algorithmic tangent modulus at integration point $\boldsymbol{x}_g$). In linear problems, $\boldsymbol{\mathcal{K}}^e$ does not change and therefore, it may be directly pre-computed (without recourse to reduced-order integration) from the fine-scale element stiffness matrix $\boldsymbol{K}^e$ by the following expression

$$\boldsymbol{\mathcal{K}}^e = (\boldsymbol{T}^T\boldsymbol{H}^{-1}\boldsymbol{\Phi}^T)\boldsymbol{K}^e(\boldsymbol{\Phi}\boldsymbol{H}^{-T}\boldsymbol{T}). \tag{74}$$

6. The *element body force vector*[10] $\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}}$ (first term of the right-hand side of Eq. 67) comprises two terms, $\boldsymbol{\mathcal{F}}^e_{\boldsymbol{eq}}$, defined in Eq. 58, and $\boldsymbol{\mathcal{F}}^e_{\boldsymbol{se}}$, defined in Eq. 69. In Appendix B, we show that these body coarse-scale force admits also a format which resembles that of external force vector in standard finite element formulations. Indeed, by assuming that the only body force per unit volume is self-weight, and that the non-interface boundary $\partial\bar{\Omega}^{non}$ is subdivided into b_{non} portions upon which the applied tractions are constant, it can be demonstrated that (see Eq. 143 in Appendix B) the coarse-scale body force at Ω^e becomes expressible as

$$\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}} = \sum_{k=1}^{n_{mat}} \boldsymbol{\mathcal{N}}^{k^T}\left(\rho^k\hat{\boldsymbol{Q}}^{e^T}\boldsymbol{g}\right) + \sum_{g=1}^{b_{non}} \bar{\boldsymbol{\mathcal{N}}}^{g^T}\boldsymbol{t}'^{e(g)}. \tag{75}$$

[10] We call it "body force vector" because it arises from upscaling those nodal forces not applied on the interface boundaries.

Here, ρ^k denotes the density of the k-th material of the unit cell ($k = 1, 2 \dots n_{mat}$), $\boldsymbol{g} \in \mathbb{R}^{n_{sd}}$ is the vector of acceleration of gravity (in global coordinates) and $\boldsymbol{t'}^{e(g)}$ stands for the traction vector (in local, surface coordinates) acting on the $g-th$ portion of $\partial\Omega^e_{non}$ ($g = 1, 2 \dots b_{non}$). The expressions for the reduced-order matrices $\boldsymbol{\mathcal{N}}^k$ and $\bar{\boldsymbol{\mathcal{N}}}^g$, on the other hand, are given in Eqs. (141) and (142), respectively. Such matrices depend only on the geometry and basis modes of the unit cell, and hence they can be precomputed in the offline phase. To adapt an existing finite element code to compute $\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}}$, one would just need to replace the standard polynomial shape functions by these matrices.

6.2.1. Dirichlet boundary conditions

In Section 4.3, we saw that each coarse-scale node possesses translational ($\bar{\boldsymbol{a}}_{\boldsymbol{t}I}$), rotational ($\bar{\boldsymbol{a}}_{\boldsymbol{r}I}$) and straining ($\hat{\boldsymbol{a}}_I$) DOFs ($I \in \mathbf{b}$ is the global index of the coarse-scale node). If there are no straining DOFs, then imposing Dirichlet boundary conditions is as simple as specifying the prescribed rigid-body DOFs of each node —as it is done in FE beam and shell formulations. For instance, a "fixed node" corresponds to a node in which $\bar{\boldsymbol{a}}_{\boldsymbol{t}I} = \mathbf{0}$ and $\bar{\boldsymbol{a}}_{\boldsymbol{r}I} = \mathbf{0}$, whereas a "hinged node" is that in which $\bar{\boldsymbol{a}}_{\boldsymbol{t}I} = \mathbf{0}$ and $\bar{\boldsymbol{a}}_{\boldsymbol{r}I}$ is unknown.

When straining DOFs come into play, things become a little bit more complicated. Such straining modes are empirically determined and therefore have no apparent physical interpretation. The easiest route is to set them to zero ($\hat{\boldsymbol{a}}_I = \mathbf{0}$) when the node is fixed or with prescribed rigid-body motions, or include them in the set of unknowns when the node is unrestricted. If the input nodal displacements, designated by $\bar{\boldsymbol{u}}_I$, are given, then one can compute the DOFs by solving the minimization problem $\min_{\boldsymbol{a}_I} \|\boldsymbol{V}_I\boldsymbol{a}_I - \bar{\boldsymbol{u}}_I\|_{\bar{\boldsymbol{M}}_{\boldsymbol{I}}}$; this gives

$$\boldsymbol{a}_I = (\boldsymbol{V}_I^T \bar{\boldsymbol{M}}_{\boldsymbol{I}} \boldsymbol{V}_I)^{-1} \boldsymbol{V}_I^T \bar{\boldsymbol{M}}_{\boldsymbol{I}} \bar{\boldsymbol{u}}_I. \tag{76}$$

The most general case is when the boundary condition is of the general affine type $\boldsymbol{A}_I\boldsymbol{V}_I\boldsymbol{a}_I = \bar{\boldsymbol{u}}_I$, where $\boldsymbol{A}_I$ is some matrix with at least as many columns as rows. This formulation covers, for instance, the case of supports in which the interface is free to move in one plane, but is restricted in the other direction. To incorporate this type of boundary conditions, we proceed by finding a full-rank block matrix of $\boldsymbol{V}^*_I = \boldsymbol{A}_I\boldsymbol{V}_I$. Denoting by $\boldsymbol{V}^*_{Ir}$ this block matrix, we can write the boundary condition as

$$\boldsymbol{V}^*_{Ir}\boldsymbol{a}_{Ir} = \bar{\boldsymbol{u}}_I - \boldsymbol{V}^*_{Im}\boldsymbol{a}_{Im}. \tag{77}$$

We refer to $\boldsymbol{a}_{Ir}$ and $\boldsymbol{a}_{Im}$ as *the slave and master DOFs*, respectively, of the (coarse-scale) node I. Next we express the slave DOFs as an *affine* function of the master DOFs by solving the corresponding minimization problem; this yields

$$\boldsymbol{a}_{Ir} = \overbrace{(\boldsymbol{V}^{*T}_{Ir}\bar{\boldsymbol{M}}_{Ir}\boldsymbol{V}^*_{Ir})^{-1}\boldsymbol{V}^{*T}_{Ir}\bar{\boldsymbol{M}}_{Ir}\bar{\boldsymbol{u}}_I}^{\bar{\boldsymbol{v}}_I} - \overbrace{(\boldsymbol{V}^{*T}_{Ir}\bar{\boldsymbol{M}}\boldsymbol{V}^*_{Ir})^{-1}\boldsymbol{V}^{*T}_{Ir}\bar{\boldsymbol{M}}_{Ir}\boldsymbol{V}^*_{Im}}^{-\boldsymbol{J}_I}\boldsymbol{a}_{Im}. \tag{78}$$

Gathering the equations for all the boundary nodes, we can express the boundary condition in a compact fashion as

$$\boldsymbol{a}_r = \bar{\boldsymbol{v}} + \boldsymbol{J}\boldsymbol{a}_m \tag{79}$$

The unknown DOFs in the problem are therefore $\boldsymbol{a}_l = [\boldsymbol{a}_m^T, \boldsymbol{a}_f^T]^T$, $\boldsymbol{a}_f$ being the remaining DOFs.

6.3. Solvability requirements

The matrices of straining modes $\boldsymbol{\Phi}$, self-equilibrating reactive modes $\boldsymbol{\Psi}$ and interface modes $\boldsymbol{V}$ cannot be chosen independently from each other, guided only by accuracy considerations. As in any other multifield variational problem, such matrices have to observe certain conditions that guarantee the well-posedness of the overall problem. One of such conditions have been already stated —the invertibility of $\boldsymbol{H} = \boldsymbol{\Phi}^T\boldsymbol{\Psi}$. The remaining conditions pose constraints in the number and form of the interface modes $\boldsymbol{V}$. In what follows, we list these additional conditions, along with other well-posedness considerations. Later on, in Sections 7.3 and 7.4, we shall discuss how to guarantee that these conditions are met when determining the basis matrices from the FE "training" data.

6.3.1. Rigid-body condition

We postulated in Eq.(32) that each fictitious interface possesses rigid-body and straining modes ($\boldsymbol{V}_i = \begin{bmatrix}\bar{\boldsymbol{V}}_i & \hat{\boldsymbol{V}}_i\end{bmatrix}$). The rigid body modes *must be included* because otherwise the coarse-scale model would not be able to exactly represent rigid-body motions without straining. Thus, we can write that

$$n_i^{int} \geq n_{rb}, \qquad i = 1, 2 \dots l \tag{80}$$

where n_i^{int} denotes the number of columns of $\boldsymbol{V}_i$. The number of interface straining modes, denoted hereafter by $\hat{n}_i^{int} = n_i^{int} - n_{rb}$, depends on the geometry of the unit cell and the "training" cases used for constructing the modes. In many cases $\hat{n}_i^{int} = 0$ ($i = 1, 2 \dots l$), meaning that the only coarse-scale DOFs are the amplitude of the rigid-body modes.

6.3.2. Interface work conditions

Suppose that we constrain the displacement of, at least, one of the fictitious interfaces of a given domain Ω^e. In such a case, the coarse-scale strain-displacement relation (62) boils down to $\boldsymbol{q}^e = \boldsymbol{H}^{-T}\boldsymbol{T}_L\boldsymbol{a}_L^e$, where the subscript L refers to the remaining coarse-scale DOFs. Since rigid-body motions are prevented, the domain must necessarily undergo straining (i.e., $\boldsymbol{q}^e \neq \boldsymbol{0}$), which requires that $\ker(\boldsymbol{H}^{-T}\boldsymbol{T}_L) = \{\boldsymbol{0}\}$, and this in turn translates into

$$\ker(\boldsymbol{T}_L) = \{\boldsymbol{0}\} \tag{81}$$

that is, the *kernel of the block matrix $\boldsymbol{T}_L$ must be zero*. This condition guarantees the invertibility of the coarse-scale stiffness matrix $\boldsymbol{\mathcal{K}}^e_{\boldsymbol{LL}}$, and therefore, the solvability of the coarse-scale problem (this follows from Eq. 74).

Finally, let us examine the physical interpretation of this solvability requirement. For the sake of argument, suppose that in this case we restrict the displacements of all interfaces but one ($\boldsymbol{a}_i^e \neq \boldsymbol{0}$), so that now the condition is $\ker(\boldsymbol{T}_i) = \{\boldsymbol{0}\}$. By virtue of the *dimension theorem*[11], and using the definition of $\boldsymbol{T}$, see Eq.(52).a, it follows that

$$n_i^{int} \leq p \tag{82}$$

and

$$\texttt{rank}(\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}\boldsymbol{V}_i) = \texttt{rank}(\begin{bmatrix}\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}\bar{\boldsymbol{V}}_i & \boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}\hat{\boldsymbol{V}}_i\end{bmatrix}) = \overbrace{n_{rb} + \hat{n}_i^{int}}^{n_i^{int}} \tag{83}$$

where[12] $\boldsymbol{\Psi}'_{\boldsymbol{f_i}} := \boldsymbol{\mathcal{Q}}_i^T\boldsymbol{\Psi}_{\boldsymbol{f_i}}$. Physically, the preceding condition is telling us that both rigid-body modes $\bar{\boldsymbol{V}}_i$ and straining body modes $\hat{\boldsymbol{V}}_i$ (if any) must *contribute to the work done by the reactive nodal forces $\boldsymbol{\Psi}'_{\boldsymbol{f_i}}$*. This condition, thus, provides a useful criterion for discriminating which straining modes should be incorporated into the set of interface modes (we shall explain this in Section 7.3). Interestingly, it also gives us hints on which type of FE "training" tests should be conducted to construct $\boldsymbol{\Psi}$. Indeed, for Eq.(83) to hold, it is necessary that $\texttt{rank}(\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}\bar{\boldsymbol{V}}_i) = n_{rb}$. For a 3D problem, for instance, this means that

$$\texttt{rank}([\overbrace{\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}(\bar{\boldsymbol{V}}_i)_1}^{\text{Axial}}, \overbrace{\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}(\bar{\boldsymbol{V}}_i)_2, \boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}(\bar{\boldsymbol{V}}_i)_3}^{\text{Shear}}, \overbrace{\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}(\bar{\boldsymbol{V}}_i)_4}^{\text{Torsion}}, \overbrace{\boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}(\bar{\boldsymbol{V}}_i)_5, \boldsymbol{\Psi}'^T_{\boldsymbol{f_i}}(\bar{\boldsymbol{V}}_i)_6}^{\text{Bending}}]) = 6. \tag{84}$$

The preceding expression indicates that the FE "training" tests should be designed so that, at the very least, the self-equilibrated modes at each interface $i = 1, 2 \ldots l$ have components in tension/compression, bending, shear and torsion. In doing so, one ensures that the *coarse-scale model is able to transmit across any interface boundary the resultants and moment resultants of the external forces* . In turn, this implies that the minimum number of straining/reactive modes in 1D, 2D and 3D periodicity is $p = 6$, 12 and 18, respectively.

7. Offline stage

As is customary in reduced-order modeling, the solution procedure is split into two distinct steps—known as the *offline and online stages*. In the *offline* stage, the basis matrices and reduced-order operators required for constructing the coarse-scale equations are computed. Such computations have to be performed *just once*, and its outcome may be reused in subsequent analyses. In the case at hand, the information required for constructing the coarse-scale problem depends, on the one hand, on a set of geometric matrices, such as the matrix of rigid body modes $\boldsymbol{R}$, and on the other hand, on three *empirical* matrices, namely: the matrix of straining modes $\boldsymbol{\Phi}$, the matrix of self-equilibrated modes $\boldsymbol{\Psi}$ and the matrix of stress modes $\boldsymbol{\Lambda}$. The qualifier *empirical* here means that such matrices are determined from the outcome of FE simulations (i.e., from *computational experiments*).

7.1. FE "training" tests

The need for such computational experiments (the *FE training tests*) is common to other multiscale methods such as the FE-HROM. However, there is a notable difference between the approach followed in such methods and the one employed here. For instance, in the FE-HROM in the small strains regime, the deformational modes are obtained by analyzing the equilibrium of a *single unit cell* under appropriate periodic boundary conditions; the infinitesimal strain

[11] The dimension theorem states that $\dim\ker(\boldsymbol{T}_i) + \texttt{rank}(\boldsymbol{T}_i) = \texttt{ncol}(\boldsymbol{T}_i)$, consult for instance [23]

[12] $\boldsymbol{\Psi}'_{\boldsymbol{f_i}}$ is the matrix of reactive modes on interface boundary $\boldsymbol{f}_i$ expressed in the reference system attached to interface. We use this variable instead of $\boldsymbol{\Psi}_{\boldsymbol{f_i}}$ to avoid the nuisance of explicitly writing the global rotation matrix $\boldsymbol{\mathcal{Q}}_i$

tensor (at the coarse scale) plays the role of input parameters in this equilibrium problem. In our multiscale approach, we cannot follow this approach of training with just one unit cell simply because *we do not know a priori which are the coarse-scale strains*. Of course we may contrive sophisticated simplifying assumptions and tailor consistent *periodic boundary conditions* to each specific case[13]. However, this approach is at odds with the very spirit of the proposed methodology, in which the simplifications leading to the coarse-scale model are presumed to be automatically *discovered* by the dimensionality reduction algorithms. The most general approach is, thus, to conduct FE training analyses of arrangements of several unit cells. This way, we allow the method to extract the most prominent patterns of *collective behavior*, without assuming any kinematic simplification at the outset.

The number of unit cells of the training structures depends on the purpose and scope of the reduced-order model. The only consideration to keep in mind is that, at the very least, the corresponding reactive modes should have component in tension/compression, shear/bending and torsion, as deduced in Eq.(84). This suggests to perform, as a rule, at least 4 training tests (for 3D problems) for each periodicity direction (1 for axial behavior, 2 for shear/bending, and 1 for torsion).

7.1.1. Snapshot matrices

Suppose we run, following the guidelines described above, a battery of FE training tests. The information of the displacements of each unit cell, the corresponding reaction forces between domains, and the stresses (all in domain coordinates) is to be stored in three distinct snapshot matrices, namely, the matrix of displacements:

$$\mathbf{A}^{\boldsymbol{d}} = \left[\boldsymbol{d}(\boldsymbol{\mu}^1) \quad \boldsymbol{d}(\boldsymbol{\mu}^2) \quad \cdots \quad \boldsymbol{d}(\boldsymbol{\mu}^P)\right]; \tag{85}$$

the matrix of reactions (for only the entries $\mathbf{f}$ corresponding to the interface boundary nodes)

$$\mathbf{A}^{\boldsymbol{\lambda}} = \left[\boldsymbol{\lambda}_{\boldsymbol{f}}(\boldsymbol{\mu}^1) \quad \boldsymbol{\lambda}_{\boldsymbol{f}}(\boldsymbol{\mu}^2) \quad \cdots \quad \boldsymbol{\lambda}_{\boldsymbol{f}}(\boldsymbol{\mu}^P)\right]; \tag{86}$$

and the matrix of stresses

$$\mathbf{A}^{\boldsymbol{\sigma}} = \left[\boldsymbol{S}(\boldsymbol{\mu}^1) \quad \boldsymbol{S}(\boldsymbol{\mu}^2) \quad \cdots \quad \boldsymbol{S}(\boldsymbol{\mu}^P)\right]. \tag{87}$$

Here $\boldsymbol{\mu}$ denotes collectively the set of *input parameters* of the problem, which may encompass the location within the structure, the employed boundary conditions, the pseudo-time (in nonlinear problems), and/or the variations of material properties, among others. The total number of snapshots is denoted by P. On the other hand, $\boldsymbol{S}$ is a vector formed by stacking the Cauchy stress vector at all Gauss points of a given domain.

Recall (see Eq.(16).b and Eq.(21).a) that we are only interested in the straining and self-equilibrated part of $\mathbf{A}^{\boldsymbol{d}}$ and $\mathbf{A}^{\boldsymbol{\lambda}}$, respectively; thus, we have to purge the rigid-body and resultant components, respectively, of each column of such matrices:

$$\mathbf{A}^{\boldsymbol{d}} \leftarrow \mathbf{A}^{\boldsymbol{d}} - \boldsymbol{R}(\boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R})^{-1}(\boldsymbol{R}^T\boldsymbol{M}\,\mathbf{A}^{\boldsymbol{d}}) \tag{88}$$

$$\mathbf{A}^{\boldsymbol{\lambda}} \leftarrow \mathbf{A}^{\boldsymbol{\lambda}} - \boldsymbol{R}_{\boldsymbol{f}}(\boldsymbol{R}_{\boldsymbol{f}}^T\boldsymbol{R}_{\boldsymbol{f}})^{-1}(\boldsymbol{R}_{\boldsymbol{f}}^T\mathbf{A}^{\boldsymbol{\lambda}}). \tag{89}$$

The stress snapshot matrix in Eq.(87), on the other hand, is needed for constructing the objective function of the minimization problem associated to the choice of the ECM points (see Eq. 123 in Appendix A).

7.2. Self-equilibrating reactive modes

To determine $\boldsymbol{\Psi}_{\boldsymbol{f}}$ from $\mathbf{A}^{\boldsymbol{\lambda}}$, we employ the *Singular Value Decomposition* (SVD). The *truncated* SVD of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ is symbolically represented here as the function

$$[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{W}] = \texttt{SVD}(\boldsymbol{A}, \epsilon), \tag{90}$$

which takes the matrix $\boldsymbol{A}$ and the relative tolerance $0 \leq \epsilon \leq 1$ as arguments, and returns the matrix of left singular vectors $\boldsymbol{U} \in \mathbb{R}^{n\times r}$, the (diagonal) matrix of positive singular values $\boldsymbol{S} \in \mathbb{R}^{r\times r}$ and the matrix of right singular vectors $\boldsymbol{W} \in \mathbb{R}^{m\times r}$. These matrices meet the following relationships

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{W}^T + \boldsymbol{E}, \qquad \|\boldsymbol{E}\|_F \leq \epsilon\|\boldsymbol{A}\|_F, \qquad \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_{r\times r}, \qquad \boldsymbol{S}_{(i+1,i+1)} \geq \boldsymbol{S}_{(i,i)} > 0, \tag{91}$$

where $\|\boldsymbol{E}\|_F$ denotes the Frobenius norm of the truncation error.

Applied to the matrix $\mathbf{A}^{\boldsymbol{\lambda}}$, using a tolerance ϵ_λ (typically around 10^{-3}), the function SVD furnishes the desired matrix of self-equilibrated modes:

$$[\boldsymbol{\Psi}_{\boldsymbol{f}}, \boldsymbol{\Sigma}^{\boldsymbol{\lambda}}, \bullet] = \texttt{SVD}(\mathbf{A}^{\boldsymbol{\lambda}}_{\boldsymbol{f}}, \epsilon_\lambda). \tag{92}$$

If the matrix is divided into blocks, one can use the partitioned version of the SVD proposed by the author in Ref. [21]. This partitioned version allows one to specify a distinct tolerance for each submatrix.

[13] We have explored this possibility in the case of 1D periodicity, but results are only satisfactory for certain types of geometries in the elastic range

Algorithm 1: Algorithm for maximizing the alignment between subspaces

Function $\boldsymbol{C}$ =*ALIG* $(\boldsymbol{A}, \boldsymbol{B})$ **:**

Data: $\boldsymbol{A} \in \mathbb{R}^{n\times r}$, $\boldsymbol{B} \in \mathbb{R}^{n\times p}$, $r \geq p$

Result: $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{h} \in \mathbb{R}^{n\times p}$, where $\boldsymbol{h} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{B} - \boldsymbol{A}\boldsymbol{c}\|_F^2$ subject to $\boldsymbol{h}^T\boldsymbol{h} = \boldsymbol{I}$

$[\boldsymbol{u}, \boldsymbol{s}, \boldsymbol{v}] = \texttt{SVD}(\boldsymbol{A}^T\boldsymbol{B})$

$\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{u}\boldsymbol{v}^T)$

7.3. Straining modes

Having determined $\boldsymbol{\Psi} \in \mathbb{R}^{n_{dof}\times p}$, the next step is to compute the corresponding matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n_{dof}\times p}$ of straining modes. The simplest route would be to apply the SVD on $\mathbf{A}^{\boldsymbol{d}}$

$$[\bar{\boldsymbol{U}}, \bullet, \bullet] = \texttt{SVD}(\mathbf{A}^{\boldsymbol{d}}, \epsilon_d) \tag{93}$$

(with a tolerance ϵ_d significantly smaller than ϵ_λ), and then make $\boldsymbol{\Phi} = \bar{\boldsymbol{U}}(:, 1:p)$, that is, take $\boldsymbol{\Phi}$ as the matrix formed by the p leading left singular vectors of $\mathbf{A}^{\boldsymbol{d}}$. However, this procedure does not guarantee one of the solvability requirements in our approach, namely, that $\boldsymbol{H} = \boldsymbol{\Phi}_{\boldsymbol{f}}^T\boldsymbol{\Psi}_{\boldsymbol{f}}$ *must be nonsingular* (see Section 6.1). It becomes necessary, thus, to develop a strategy that explicitly takes into consideration this condition.

7.3.1. Alignment of subspaces

Let us begin by assuming that the p columns of the desired basis matrix are a linear combination of the r columns ($r > p$) of $\bar{\boldsymbol{U}}$ (computed from the SVD (93)), and let us denote by $\boldsymbol{c}$ the coefficient matrix in this linear combination. The problem can be rephrased, thus, as that of finding a matrix $\boldsymbol{c} \in \mathbb{R}^{r\times p}$ such that $(\bar{\boldsymbol{U}}_{\boldsymbol{f}}\boldsymbol{c})^T\boldsymbol{\Psi}_{\boldsymbol{f}}$ is non-singular. As pointed out earlier, this matrix represents the work done by the reactive modes $\boldsymbol{\Psi}_{\boldsymbol{f}}$ when the interface boundary nodes move along the displacement patterns dictated by $\bar{\boldsymbol{U}}_{\boldsymbol{f}}\boldsymbol{c}$. It appears reasonable, therefore, to enforce that the matrix is nonsingular by, loosely speaking, incorporating straining modes that contribute "as much as possible" to the interface boundary work. Intuitively, if we think of $\bar{\boldsymbol{U}}_{\boldsymbol{f}}\boldsymbol{c}$ and $\boldsymbol{\Psi}_{\boldsymbol{f}}$ as vectors of displacements and forces, respectively, in a high-dimensional space, then maximizing the work (in absolute value) is equivalent to make $\bar{\boldsymbol{U}}_{\boldsymbol{f}}\boldsymbol{c}$ "as aligned as possible" to $\boldsymbol{\Psi}_{\boldsymbol{f}}$. This intuitive idea can be shaped into a more rigorous form by appealing to the notion of *rotation of subspaces* (cf. Golub et al. [19]). In particular, the problem at hand can be posed as follows: find $\boldsymbol{c} \in \mathbb{R}^{r\times p}$ such that

$$\min_{\boldsymbol{c}} \|\boldsymbol{\Psi}_{\boldsymbol{f}} - \bar{\boldsymbol{U}}_{\boldsymbol{f}}\boldsymbol{c}\|_F^2 \tag{94}$$

subject to $\boldsymbol{c}^T\boldsymbol{c} = \boldsymbol{I}$. The solution of this constrained minimization problem can be readily obtained as

$$\boldsymbol{c} = \boldsymbol{u}\boldsymbol{v}^T \tag{95}$$

where

$$[\boldsymbol{u}, \bullet, \boldsymbol{v}] = \texttt{SVD}(\bar{\boldsymbol{U}}_{\boldsymbol{f}}^T\boldsymbol{\Psi}_{\boldsymbol{f}}). \tag{96}$$

(When the second argument of the function SVD() is omitted, it is tacitly assumed that the tolerance is equal to the *machine epsilon*). For later purposes, this operation of maximizing the alignment between two subspaces will be compactly represented by the function $\texttt{ALIG}(\bullet, \bullet)$, described in Algorithm 1.

According to condition 16.a, the desired basis matrix $\boldsymbol{\Phi}$ is to be columnwise orthogonal; hence, a last step is required to turn $\bar{\boldsymbol{U}}\boldsymbol{c}$ orthogonal:

$$[\boldsymbol{\Phi}, \bullet, \bullet] = \texttt{SVD}(\bar{\boldsymbol{U}}\boldsymbol{c}). \tag{97}$$

7.4. Interface modes

Now we come to grips with arguably the most critical and challenging part of the offline stage: the determination of the interface modes. More specifically, since rigid body modes are to be invariably included, the task boils down to determine the straining modes $\hat{\boldsymbol{V}}_i$ $(i = 1, 2 \ldots l)$ of each fictitious interfaces. The conditions that these modes have to satisfy, discussed in previous Sections, are summarized below.

1. Orthogonality conditions (see Eq. 35)

$$\hat{\boldsymbol{V}}_i^T\bar{\boldsymbol{M}}_{\boldsymbol{i}}\hat{\boldsymbol{V}}_i = \boldsymbol{I}, \qquad \hat{\boldsymbol{V}}_i^T\bar{\boldsymbol{M}}_{\boldsymbol{i}}\bar{\boldsymbol{V}}_i = \boldsymbol{0}, \quad i = 1, 2 \ldots l \tag{98}$$

Algorithm 2: Weighted (truncated) Singular Value decomposition

Function $[\boldsymbol{U},\boldsymbol{S},\boldsymbol{V}]$ =WSVD($\boldsymbol{A},\boldsymbol{M},TOL$):
  Data: $\boldsymbol{A}\in\mathbb{R}^{n\times m}$, $\boldsymbol{M}\in\mathbb{R}^{n\times n}$ (positive definite), tolerance $0\leq TOL\leq 1$
  Result: $\boldsymbol{U}\in\mathbb{R}^{n\times r}$, $\boldsymbol{S}\in\mathbb{R}^{r\times r}$, $\boldsymbol{V}\in\mathbb{R}^{m\times r}$ ($r\leq min(n,m)$), with $\boldsymbol{U}^T\boldsymbol{M}\boldsymbol{U}=\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}_{r\times r}$,
    $\boldsymbol{S}_{(i+1,i+1)}\geq\boldsymbol{S}_{(i,i)}>0$, such that $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{W}^T+\boldsymbol{E}$, $\|\boldsymbol{E}\|_{\boldsymbol{M}}\leq\epsilon\|\boldsymbol{A}\|_{\boldsymbol{M}}$
  $\bar{\boldsymbol{M}}$ =chol($\boldsymbol{M}$) // Cholesky decomposition of $\boldsymbol{M}$
  $\boldsymbol{A}\leftarrow\bar{\boldsymbol{M}}\boldsymbol{A}$
  $[\bar{\boldsymbol{U}},\boldsymbol{S},\boldsymbol{V}]$ = SVD($\boldsymbol{A},TOL$) // *Thin* SVD of $\boldsymbol{A}$.
  $\boldsymbol{U}=\bar{\boldsymbol{M}}^{-1}\bar{\boldsymbol{U}}$

2. Periodicity

$$\hat{\boldsymbol{V}}_i^+=\hat{\boldsymbol{V}}_i^- \qquad i=1,2\ldots l/2 \tag{99}$$

3. Interface work conditions (Eq. 81)

$$\ker(\boldsymbol{T}_L)=\{\boldsymbol{0}\} \tag{100}$$

where $\boldsymbol{T}=\boldsymbol{\Psi}_{\boldsymbol{f}}'^T\boldsymbol{V}$. The subscript L refers to the set of unconstrained coarse-scale DOFs of the unit cell. Condition Eq.(100) must hold for all L such that, at least, one of the interfaces is prevented to move (by constraining all its nodal DOFs).

7.4.1. Candidate modes

Let us consider a given pair of periodic interfaces $\partial\bar{\Omega}_i^+$ and $\partial\bar{\Omega}_i^-$ ($i=1,2\ldots l/2$), and let us construct the matrix containing the entries of the (domain) straining modes at the nodes of the concerned interface boundaries (in the local reference system attached to the interfaces):

$$\boldsymbol{C}_i=[\boldsymbol{\Phi}'_{\boldsymbol{f}_i^+},\boldsymbol{\Phi}'_{\boldsymbol{f}_i^-}] \tag{101}$$

where $\boldsymbol{\Phi}'_{\boldsymbol{f}_i}:=\boldsymbol{\mathcal{Q}}_i^T\boldsymbol{\Phi}_{\boldsymbol{f}_i}$. Next we make the above matrix M-orthogonal to the rigid body modes, as required by Eq.(35).b:

$$\boldsymbol{C}_i\leftarrow\boldsymbol{C}_i-\bar{\boldsymbol{V}}_i(\bar{\boldsymbol{V}}_i^T\bar{\boldsymbol{M}}_{\boldsymbol{i}}\bar{\boldsymbol{V}}_i)^{-1}\bar{\boldsymbol{V}}_i^T\bar{\boldsymbol{M}}_{\boldsymbol{i}}\boldsymbol{C}_i. \tag{102}$$

Finally, to turn $\boldsymbol{C}_i$ columnwise M-orthogonal, as specified by Eq.(35).b, we employ the weighted SVD described in Algorithm[14] 2:

$$[\bar{\boldsymbol{C}}_i,\bullet,\bullet]=\texttt{WSVD}(\boldsymbol{C}_i,\bar{\boldsymbol{M}}_{\boldsymbol{i}},\epsilon_c). \tag{103}$$

To filter out irrelevant, noisy modes, we set $\epsilon_c\sim10^{-6}$. With $\bar{\boldsymbol{C}}_i$ at hand, we can state that

$$span(\boldsymbol{V}_i)\subseteq span([\bar{\boldsymbol{V}}_i,\bar{\boldsymbol{C}}_i]) \tag{104}$$

that is, the *column space* of the desired basis matrix $\boldsymbol{V}_i$ lies on the column space of the matrix containing $\bar{\boldsymbol{C}}_i$ and the rigid-body matrix $\bar{\boldsymbol{V}}_i$. We call $\bar{\boldsymbol{C}}_i$ the matrix of *candidate interface modes* of the $i-th$ fictitious interface. The remainder of this section is devoted to examine which of these candidate modes is to be incorporated to the set of interface modes. The guiding criterion is the stability condition (100). We begin by studying the stability at the level of two interacting interfaces; this will provide sufficient solvability conditions for 1D periodicity. Next, in Section (7.4.3), we shall address the more general case of 2D and 3D periodicity.

7.4.2. Solvability condition for each pair of interfaces

Particularization of the solvability requirement (100) for the cases in which all interfaces are fixed except for either $\partial\bar{\Omega}_i^+$ or $\partial\bar{\Omega}_i^-$ yields the conditions

$$\ker(\boldsymbol{\Psi}'^T_{\boldsymbol{f}_i^+}\boldsymbol{V}_i)=\{\boldsymbol{0}\},\qquad\text{and}\qquad\ker(\boldsymbol{\Psi}'^T_{\boldsymbol{f}_i^-}\boldsymbol{V}_i)=\{\boldsymbol{0}\}. \tag{105}$$

[14] In essence, it is similar to the standard SVD, except for the fact that the underlying minimization problem is posed in terms of the norm induced by a positive definite matrix $\boldsymbol{M}$.

These two conditions are formally identical to that discussed in Section 7.3, a fact that suggests that we may follow the "subspace alignment recipe" for determining $\boldsymbol{V}_i$. However, there are two additional difficulties now. Firstly, we do not know yet how many interface modes (n_i^{int}) are to be included; and secondly, in general $span(\boldsymbol{\Psi}'^T_{\boldsymbol{f}_i^+}) \neq span(\boldsymbol{\Psi}'^T_{\boldsymbol{f}_i^-})$ (both subspaces are only equal in 1D periodic problems when the training is made under strict periodicity conditions), and therefore, if we find one $\boldsymbol{V}_i$ such that Eq.(105).a is satisfied, this does not imply that Eq.(105).b holds.

The key to resolve this double conundrum lies in the notion of *intersection of subspaces*. Suppose we calculate a matrix $\boldsymbol{D}$ such that its column space is the intersection of the column spaces of $\boldsymbol{\Psi}'_{\boldsymbol{f}_i^+}$ and $\boldsymbol{\Psi}'_{\boldsymbol{f}_i^-}$:

$$span(\boldsymbol{D}) = span(\boldsymbol{\Psi}'_{\boldsymbol{f}_i^+}) \cap span(\boldsymbol{\Psi}'_{\boldsymbol{f}_i^-}). \tag{106}$$

With $\boldsymbol{D}$ at one's disposal, conditions in 105.a and 105.b can be replaced by a single requirement:

$$\ker(\boldsymbol{D}^T \boldsymbol{V}_i) = \{\boldsymbol{0}\}. \tag{107}$$

It follows from the preceding condition that the number of interface modes n_i^{int} cannot be greater than the dimension of the intersection space (number of columns of $\boldsymbol{D}$). As a rule, we shall take $n_i^{int} = \texttt{ncol}(\boldsymbol{D})$. Appealing again to the reasoning used in Section 7.3.1, the problem can be thus phrased as that of finding a matrix $\boldsymbol{U}$ with $\texttt{ncol}(\boldsymbol{D})$ columns such that, on the one hand, $span(\boldsymbol{U}) \subseteq span([\bar{\boldsymbol{V}}_i, \bar{\boldsymbol{C}}_i])$ and, on the other hand, $span(\boldsymbol{U})$ is "as aligned as possible" to the intersection subspace $span(\boldsymbol{D})$:

$$\boldsymbol{U} = \texttt{ALIG}(\boldsymbol{D}, [\bar{\boldsymbol{V}}_i, \bar{\boldsymbol{C}}_i]). \tag{108}$$

Algorithm 3: Straining modes of interfaces $\partial\bar{\Omega}_i^+$ and $\partial\bar{\Omega}_i^-$

1 **Function** $[\hat{\boldsymbol{V}}, \bar{\boldsymbol{C}}]$ =STRMODP($\boldsymbol{\Psi}^+$,$\boldsymbol{\Psi}^-$,$\boldsymbol{\Phi}^+$,$\boldsymbol{\Phi}^-$,$\bar{\boldsymbol{V}}$,$\bar{\boldsymbol{M}}$,TOL_{ang}):

Data: $\boldsymbol{\Psi}^+, \boldsymbol{\Psi}^- \in \mathbb{R}^{n\times p}$ (domain reaction modes at the nodes of the two periodic boundary interfaces), $\boldsymbol{\Phi}^+, \boldsymbol{\Phi}^- \in \mathbb{R}^{n\times p}$ (corresponding domain straining modes), $\bar{\boldsymbol{V}} \in \mathbb{R}^{n\times n_{rb}}$ (rigid-body matrix of the interface), $\bar{\boldsymbol{M}} \in \mathbb{R}^{n\times n}$ (geometric mass matrix of the interface), $0 \leq TOL \leq 90$ (angle tolerance for the intersection subspace, in degrees)

Result: $\hat{\boldsymbol{V}} \in \mathbb{R}^{n\times m}$ $(0 \leq m \leq p - n_{rb})$: Matrix of straining interface modes. $\bar{\boldsymbol{C}} \in \mathbb{R}^{n\times m}$, matrix of candidate for straining interface modes ($span(\hat{\boldsymbol{V}}) \subseteq span(\bar{\boldsymbol{C}})$)

2 $\bar{\boldsymbol{C}} \leftarrow [\boldsymbol{\Phi}^+, \boldsymbol{\Phi}^-]$ // Matrix of candidate interface modes (initialization)

3 $\bar{\boldsymbol{C}} \leftarrow \bar{\boldsymbol{C}} - \bar{\boldsymbol{V}}(\bar{\boldsymbol{V}}^T\bar{\boldsymbol{M}}\bar{\boldsymbol{V}})^{-1}\bar{\boldsymbol{V}}^T\bar{\boldsymbol{M}}\bar{\boldsymbol{C}}$ // Make $\bar{\boldsymbol{C}}$ M-orthogonal to rigid-body modes

4 $[\bar{\boldsymbol{C}}, \bullet, \bullet] \leftarrow \texttt{WSVD}(\bar{\boldsymbol{C}}, \bar{\boldsymbol{M}}, TOL_c)$ // Make $\bar{\boldsymbol{C}}$ columnwise M-orthogonal, $TOL_c \sim 10^{-6}$

5 $\boldsymbol{D} \leftarrow \texttt{INTS}\ (\boldsymbol{\Psi}^+, \boldsymbol{\Psi}^-, \boldsymbol{I}, TOL_{ang})$ // Basis matrix for $span(\boldsymbol{\Psi}^+) \cap span(\boldsymbol{\Psi}^-)$, see Alg. 4

6 $\boldsymbol{U} \leftarrow \texttt{ALIG}\ (\boldsymbol{D}, [\bar{\boldsymbol{V}}, \bar{\boldsymbol{C}}])$. // Alignment of subspaces, see Alg. 1

7 $\hat{\boldsymbol{V}} \leftarrow \texttt{INTS}\ (\boldsymbol{U}, \bar{\boldsymbol{C}}, \bar{\boldsymbol{M}}, TOL_{int})$ // Basis matrix for $span(\boldsymbol{U}) \cap span(\bar{\boldsymbol{C}})$, $TOL_{int} \sim 10^{-3}$, see Alg. 4

Physically, $span(\boldsymbol{U})$ is the subspace of $span([\bar{\boldsymbol{V}}_i, \bar{\boldsymbol{C}}_i])$ that contributes the most to the work done by the reactive modes whose patterns are identical in both $\partial\bar{\Omega}_i^+$ and $\partial\bar{\Omega}_i^-$ —somehow, thus, in proceeding this way, we are imposing *periodicity* in terms of reactive forces. Once $\boldsymbol{U}$ is at our disposal, we have to calculate the intersection between the column space of $\boldsymbol{U}$ and $\bar{\boldsymbol{C}}_i$ (in order to determine which is the *straining* component of the subspace):

$$span(\hat{\boldsymbol{V}}_i) = span(\boldsymbol{U}) \cap span(\bar{\boldsymbol{C}}_i). \tag{109}$$

It should be noted that $\hat{\boldsymbol{V}}_i$ will be non-empty only if $\texttt{ncol}(\boldsymbol{D}) > n_{rb}$. If $\texttt{ncol}(\boldsymbol{D}) = n_{rb}$, the interfaces will have only rigid-body DOFs. The case $\texttt{ncol}(\boldsymbol{D}) < n_{rb}$, on the other hand, evidences a deficient training set —in the sense that condition (84) is not observed.

The above described steps for determining the straining modes of each pair of periodic interfaces are synthesized in function

$$[\hat{\boldsymbol{V}}_i, \bar{\boldsymbol{C}}_i] = \texttt{STRMODP}(\boldsymbol{\Psi}'_{\boldsymbol{f}_i^+}, \boldsymbol{\Psi}'_{\boldsymbol{f}_i^-}, \boldsymbol{\Phi}'_{\boldsymbol{f}_i^+}, \boldsymbol{\Phi}'_{\boldsymbol{f}_i^-}, \bar{\boldsymbol{V}}_i, \bar{\boldsymbol{M}}_{\boldsymbol{i}}, \epsilon_{rint}), \tag{110}$$

outlined in Algorithm 3. Line 5 in this Algorithm corresponds to the computation of the intersection subspace of Eq.(106), which is in turn carried out following the steps summarized in Algorithm 4. Such computations are based on the concept of *principal angles* between two subspaces (cf. Ref. [19]). Strictly speaking, the dimension of the intersection is equal to the number of principal angles (calculated in line 8 of Algorithm 4) equal to zero. However, we relax this condition and define the intersection in a weaker sense, by selecting those principal vectors whose associated angles are below a specified tolerance ϵ_{rint} (the last input argument of function STRMODP()). In the examples shown later we have set $\epsilon_{rint} = 1$ degree.

Algorithm 4: Intersection of subspaces

Function $\boldsymbol{U}$ =INTS($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{M}, \theta_{tol}$):
  Data: Matrices $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, $\boldsymbol{B} \in \mathbb{R}^{n\times p}$, $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ (positive definite). Maximum angle (degr.) $0 \leq \theta_{tol} \leq 90$
  Result: $\boldsymbol{U} \in \mathbb{R}^{n\times r}$, $r \leq min(m,p)$, with $\boldsymbol{U}^T\boldsymbol{M}\boldsymbol{U} = \boldsymbol{I}$ and $span(\boldsymbol{U}) = span(\boldsymbol{A}) \cap span(\boldsymbol{B})$
  if $\boldsymbol{M} \neq \boldsymbol{I}$ **then** $\boldsymbol{M} \leftarrow$chol($\boldsymbol{M}$) // Cholesky decomposition of $\boldsymbol{M}$
  $\boldsymbol{A} \leftarrow \boldsymbol{M}\boldsymbol{A}$; $\boldsymbol{B} \leftarrow \boldsymbol{M}\boldsymbol{B}$
  $[\boldsymbol{Y}_A, \cdot, \cdot] \leftarrow$ SVD($\boldsymbol{A}$) // *Thin* SVD of $\boldsymbol{A}$. $\boldsymbol{Y}_A$ is the matrix of left singular vectors
  $[\boldsymbol{Y}_B, \cdot, \cdot] \leftarrow$ SVD($\boldsymbol{B}$)
  $[\boldsymbol{Z}, \boldsymbol{S}, \cdot] \leftarrow$ SVD($\boldsymbol{Y}_A^T\boldsymbol{Y}_B$) // $\boldsymbol{S}$ is a vector containing the singular values.
  $\theta_i \leftarrow$ acosd(S_i) $(i = 1, 2 \ldots$ length $(\boldsymbol{S}))$ // Principal angles, in degrees
  Find s such that $\theta_s > \theta_{tol}$. Set $r \leftarrow s - 1$
  $\boldsymbol{U} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{Y}_A\boldsymbol{Z}(:, 1:r)$ // $\boldsymbol{Z}(:, 1:r)$ is the block matrix of $\boldsymbol{Z}$ formed by columns $1, 2 \ldots r$

7.4.3. General solvability conditions for 2D and 3D periodicity

The procedure described in the foregoing (summarized in Algorithm 3) provides a set of straining modes that meet the necessary conditions stated in Eq.(105), in which all the interfaces of the unit cell but one are assumed to be fixed. For the case of 1D periodicity, these conditions are sufficient to ensure the solvability of the problem. However, for 2D and 3D periodicity, there is no guarantee that these straining modes furnish a stable solution. The reason is that these modes have been obtained without accounting for possible interrelations between the motion of contiguous interface boundaries, as demanded by condition (100). The criterion for checking whether such interrelations exist reads

$$\sum_{i=1}^{l}(n_{rb} + \hat{n}_i^{int}) \leq n_{rb} + p. \tag{111}$$

If this condition is not satisfied, that is, if the total number of modes of the fictitious interfaces (left-hand side) exceeds the total number of modes of the unit cell (right-hand side), it means that there are straining modes that are interrelated and, thus, the problem may become ill-posed. In what follows, we propose a method for dealing with such situations.

The method is based on a reformulation of the stability condition (100). Now, rather than prescribing all the nodal displacements of one of the interfaces, we set to zero the amplitudes of its rigid body modes, allowing the interface to freely strain. To put it alternatively, the subscript $\boldsymbol{L}$ in (100) (reprinted below)

$$\ker(\boldsymbol{\Psi}_{\boldsymbol{f}}'^T\boldsymbol{V}_L) = \{\boldsymbol{0}\}, \tag{112}$$

refers now to the indexes of all the coarse-scale DOFs except for the n_{rb} rigid body DOFs of any of the fictitious interfaces. The resulting condition is more stringent (it is sufficient, but not necessary for solvability purposes), but facilitates the determination of the interface straining modes. Indeed, this expression bears a close resemblance to the condition observed by the straining modes of the domain (see Eq. 26). For convenience, we reprint this equation below:

$$\ker(\boldsymbol{\Psi}_{\boldsymbol{f}}^T\boldsymbol{\Phi}_{\boldsymbol{f}}) = \ker(\boldsymbol{\Psi}_{\boldsymbol{f}}'^T\boldsymbol{\Phi}_{\boldsymbol{f}}') = \{\boldsymbol{0}\}. \tag{113}$$

This resemblance suggests that, since the domain straining modes $\boldsymbol{\Phi}_{\boldsymbol{f}}'$ already fulfills the preceding *domain* stability condition, all we have to do is to determine $\boldsymbol{V}$ such that $\boldsymbol{V}_L$ can reproduce as close as possible any straining displacement of the interface boundaries (any linear combination of $\boldsymbol{\Phi}_{\boldsymbol{f}}'$). In doing so, the interface condition (112) will be automatically observed. We describe in what follows a *heuristic* procedure for achieving such a goal.

Let us begin by considering the matrix

$$\boldsymbol{S} = \begin{bmatrix}\boldsymbol{R}_{\boldsymbol{f}}' & \boldsymbol{\Phi}_{\boldsymbol{f}}'\end{bmatrix} \tag{114}$$

where $\boldsymbol{R}_{\boldsymbol{f}_i}' := \boldsymbol{\mathcal{Q}}_i^T\boldsymbol{R}_{\boldsymbol{f}_i}$, $i = 1, 2 \ldots l$. The column space of this matrix represents the set of all possible displacements of the interface boundaries of the unit cell (expressed in the local reference system of each interface). On the other hand, the column space of the matrix

$$\boldsymbol{U} = \text{diag}\left([\bar{\boldsymbol{V}}_1, \bar{\boldsymbol{C}}_1], [\bar{\boldsymbol{V}}_2, \bar{\boldsymbol{C}}_2], \cdots, [\bar{\boldsymbol{V}}_l, \bar{\boldsymbol{C}}_l]\right) \tag{115}$$

represents the set of all possible configurations of the fictitious interfaces —as a function of the candidate straining modes $\bar{\boldsymbol{C}}_i$, computed by means of function STRMODP() in (110). On average, each fictitious interface has p candidate straining modes, so the total number of independent configurations is ncol($\boldsymbol{U}$) $\sim l(n_{rb}+p)$, a figure that largely exceeds the number

of possible configurations of the interface boundaries of the unit cell ($\texttt{ncol}(\boldsymbol{S}) = n_{rb} + p$). The idea put forward above is that, in order to guarantee the solvability of the problem, the interface basis matrix $\boldsymbol{V}$ can only include those columns of $\boldsymbol{U}$ that are, in the sense defined in what follows, *kinematically compatible* with the modes of the unit cell (the columns of $\boldsymbol{S}$). These kinematically compatible modes (a total of $n_{rb} + p$) may be selected from the columns of $\boldsymbol{U}$ by minimizing the gap between the fictitious interfaces and the interface boundaries of the unit cell for any displacement of the latter. The underlying minimization problem is as follows: find $\boldsymbol{a} \in \mathbb{R}^{n_{rb}+p}$ and $\mathcal{I} \subseteq \{1, 2 \dots \texttt{ncol}(\boldsymbol{U})\}$ such that

$$\{\boldsymbol{a}, \mathcal{I}\} = \arg \min_{\bar{\mathcal{I}}, \bar{\boldsymbol{a}} \in \mathbb{R}^{n_{rb}+p}} \|\boldsymbol{S}\bar{\boldsymbol{q}} - \boldsymbol{U}_{\bar{\mathcal{I}}}\, \bar{\boldsymbol{a}}\|_{\boldsymbol{M}}^2 \tag{116}$$

for all $\bar{\boldsymbol{q}} \in \mathbb{R}^{n_{rb}+p}$ (here $\boldsymbol{U}_{\bar{\mathcal{I}}} = \boldsymbol{U}(:, \bar{\mathcal{I}})$, that is, it is the block of $\boldsymbol{U}$ formed by the columns corresponding to indexes $\bar{\mathcal{I}}$). This (discrete) minimization problem is subjected to two constraints. Firstly, $\mathcal{I}$ must include the indexes of the rigid-body modes (as stated in Section 6.3.1); and, secondly, $\mathcal{I}$ must observe the periodicity conditions, in the sense that there should be a disjoint partition $\mathcal{I} = \mathcal{I}^+ \cup \mathcal{I}^-$ such that the modes of [15] $\boldsymbol{U}(:, \mathcal{I}^+)$ are identical to that of $\boldsymbol{U}(:, \mathcal{I}^-)$.

Algorithm 5: Selection of kinematically compatible interface modes

Function $\mathcal{I}$ =KINEMDOF($\boldsymbol{D}, I^{rb}, I^+, I^-$):
- **Data:** Unconstrained minimization matrix $\boldsymbol{D} \in \mathbb{R}^{2m \times 2p}$ (see Eq. 117), where $m > p$, and $\texttt{rank}(\boldsymbol{D}) = 2p$. Rigid-body indexes $I^{rb} \subset \{1, 2 \dots 2m\}$, $\texttt{length}(I^{rb}) \leq 2p$. Periodic indexes I^+, I^-, where $I^+ \cup I^- = \{1, 2 \dots 2m\}$ ($I^+(k)$ is the periodic counterpart of $I^-(k)$, for $k = 1, 2 \dots m$).
- **Result:** $\mathcal{I} \subset \{1, 2 \dots 2m\}$: set of $2p$ linearly independent rows of $\boldsymbol{D}$ such that $I^{rb} \subseteq \mathcal{I}$ and $\mathcal{I} = \mathcal{I}^+ \cup \mathcal{I}^-$
- $J \leftarrow \texttt{DEIM}(\boldsymbol{D}(I^{rb}, :)^T)$ // Apply DEIM (see Alg.(6)) to select J so that $\boldsymbol{D}(I^{rb}, J)$ is invertible
- $L \leftarrow \{1, 2 \dots 2p\} \setminus J$; $\boldsymbol{A} \leftarrow \boldsymbol{D}(:, J)$; $\mathcal{I} \leftarrow I^{rb}$; $\mathcal{I}^+ \leftarrow \mathcal{I} \cap I^+$; $\mathcal{I}^- \leftarrow \mathcal{I} \cap I^-$ // Initializations
- **for** $j = 1$ *to* $\texttt{length}(L)/2$ **do**
  - // Approximate the next two columns of $\boldsymbol{D}(:, L)$ as an oblique projection onto $\boldsymbol{D}(:, J)$
  - $\boldsymbol{r} \leftarrow \boldsymbol{0}$ // Initialization residual
  - $C^+ \leftarrow I^+ \setminus \mathcal{I}^+$; $C^- \leftarrow I^- \setminus \mathcal{I}^-$ // Candidate indexes
  - **for** $i = 1$ *to* 2 **do**
    - $k \leftarrow 2(j-1) + i$
    - $\boldsymbol{s} \leftarrow \boldsymbol{D}(:, L(k)) - \boldsymbol{A}\boldsymbol{A}(\mathcal{I}, :)^{-1} \boldsymbol{D}(\mathcal{I}, L(k))$ // Residual (oblique projection)
    - $\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{s} \circ \boldsymbol{s}$ // $\boldsymbol{s} \circ \boldsymbol{s}$ is the Hadamard (element wise) product
  - **end**
  - $\boldsymbol{r} \leftarrow \sqrt{\boldsymbol{r}}$
  - $\boldsymbol{a} \leftarrow \sqrt{\boldsymbol{r}(C^+) \circ \boldsymbol{r}(C^+) + \boldsymbol{r}(C^-) \circ \boldsymbol{r}(C^-)}$ // Residual associated to each pair of faces
  - $d \leftarrow \arg\max_e \boldsymbol{a}(e)$ // Candidate index at which the residual is maximum
  - $\mathcal{I}^+ \leftarrow \mathcal{I}^+ \cup C^+(d)$; $\mathcal{I}^- \leftarrow \mathcal{I}^- \cup C^-(d)$; $\mathcal{I} \leftarrow \mathcal{I}^+ \cup \mathcal{I}^-$;
- **end**

Problem (116) may be expressed in a different format, with a single unknown, by using the solution of the unconstrained minimization problem. Assuming one takes all the columns of $\boldsymbol{U}$, this solution reads $\boldsymbol{a} = \boldsymbol{D}\bar{\boldsymbol{q}}$, where

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_1^T & \boldsymbol{D}_2^T & \cdots & \boldsymbol{D}_l^T \end{bmatrix}^T, \qquad \boldsymbol{D}_i = (\boldsymbol{U}_i^T \bar{\boldsymbol{M}}_i \boldsymbol{U}_i)^{-1} \boldsymbol{U}_i^T \bar{\boldsymbol{M}}_i [\boldsymbol{R}'_{\boldsymbol{f}_i}, \boldsymbol{\Phi}'_{\boldsymbol{f}_i}]. \tag{117}$$

By construction, $span(\boldsymbol{S}) \subseteq span(\boldsymbol{U})$. Therefore, we can write $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{D}$, and cast Eq.(116) as

$$\mathcal{I} = \arg \min_{\bar{\mathcal{I}} \in \mathbb{N}^{n_{rb}+p}} \|\boldsymbol{U}(\boldsymbol{D} - \mathcal{P}_{\bar{\mathcal{I}}}^T \mathcal{P}_{\bar{\mathcal{I}}} \boldsymbol{D})\|_{\boldsymbol{M}} \tag{118}$$

where $\mathcal{P}_{\bar{\mathcal{I}}}$ denotes the boolean "selection" operator associated to the entries $\bar{\mathcal{I}}$. Owing to both its discrete nature and the presence of the periodicity constraints, problem (118) is a *NP-hard* optimization problem, and recourse to sub-optimal solutions is to be made. Here we propose a "greedy" method that operates on the matrix $\boldsymbol{D}$ defined in (117). The format of the objective function in problem (118) suggests that $\mathcal{I}$ may be selected so that, loosely speaking, the resulting block matrix $\boldsymbol{D}(\mathcal{I}, :)$ represents as much as possible the information contained in the original matrix. In this light, the problem

[15] To fulfill this condition, the truncation of reactions modes in the SVD (92) is to be made such that $p + n_{rb}$ is even.

Algorithm 6: Discrete Empirical Interpolation Method

Function J =DEIM($\boldsymbol{A}$):
  Data: $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m \geq n$, $\texttt{rank}(\boldsymbol{A}) = n$.
  Result: $J \subset \{1, 2 \dots m\}$: subset of n maximally independent rows of $\boldsymbol{A}$, such that $\boldsymbol{A}(J, :)$ is invertible.
  $\boldsymbol{U} \leftarrow \boldsymbol{A}(:, 1)$; $J \leftarrow \arg\max_e |\boldsymbol{U}(e)|$ // Initializations
  for $k = 2$ *to* n **do**
    $\boldsymbol{r} \leftarrow \boldsymbol{A}(:, k) - \boldsymbol{U}\boldsymbol{U}(J, :)^{-1}\boldsymbol{A}(J, k)$ // Residual (oblique projection)
    $i \leftarrow \arg\max_e |\boldsymbol{r}(e)|$ // New index
    $J \leftarrow J \cup i$
  end

translates into that of selecting the set of *maximally independent* rows of $\boldsymbol{D}$. This problem is analogous to that addressed by *discrete interpolation methods*, the only difference being the existence of the rigid-body and periodicity constraints. Accordingly, we have adapted one of such methods —more specifically, the (Discrete) Empirical Interpolation Method [2, 7], reproduced for convenience in Algorithm (6)— to account for the above mentioned constraints. The resulting scheme is shown in Algorithm (5). Once the set of indexes $\mathcal{I}$ have been computed using this algorithm, the desired interface basis matrix is obtained by simply setting $\boldsymbol{V} = \boldsymbol{U}(:, \mathcal{I})$.

7.5. Summary of offline operations

For the reader's convenience, we have summarized the set of operations to be carried out in the offline stage in Box 8.1.

8. Online stage: the coarse-scale problem

1. *Geometric information of the unit cell.* Compute and store in memory all the FE variables that solely depend on the geometry of the unit cell (and on its FE discretization). This includes, among others, the rigid body modes of the domain ($\boldsymbol{R}$, see Eq. 15), the strain-displacement matrix at each Gauss point ($\boldsymbol{B}(\boldsymbol{x}_g)$, see Eq. 6), the FE integration weights W_g (Eq. 8), the geometric mass matrix $\boldsymbol{M}$ (Eq. 16), the set of boundary DOFs $\boldsymbol{f}$ (Eq. 4) and $\bar{\boldsymbol{R}}_{\boldsymbol{f}} = \boldsymbol{M}_{\boldsymbol{f}}\boldsymbol{R}_{\boldsymbol{f}}$ (Eq. 22).
2. *Geometric information of the boundary interfaces.* Compute the geometric mass matrices $\bar{\boldsymbol{M}}_{\boldsymbol{i}}$ (Eq. 35), relative rotation matrices $\boldsymbol{Q}_i$ (see Section 2.1) and rigid body modes $\bar{\boldsymbol{V}}_i$ (Eq. 34) ($i = 1, 2 \dots l$).
3. *Training stage.* Run pertinent FE analyses following the guidelines given in Section (7.1). Store solutions in matrices $\mathbf{A}^{\boldsymbol{d}}$, $\mathbf{A}^{\boldsymbol{\lambda}}$ and $\mathbf{A}^{\boldsymbol{\sigma}}$ (see Eqs. 85, 86 and 87). Determine the straining and self-equilibrating components of $\mathbf{A}^{\boldsymbol{d}}$ and $\mathbf{A}^{\boldsymbol{\lambda}}$ using Eqs. (88) and (89).
4. *Unit cell modes.* Determine the basis matrix for self-equilibrating reactions $\boldsymbol{\Psi}$ using the truncated SVD on $\mathbf{A}^{\boldsymbol{\lambda}}$ (see Expr. 92). Using $\boldsymbol{\Psi}$ and $\mathbf{A}^{\boldsymbol{d}}$, compute the matrix of straining modes $\boldsymbol{\Phi}$ following the "alignment" recipe described in Section 7.3.
5. *Reduced set of integration points.* Using as inputs the reduced B-matrices $\{\boldsymbol{B}^*(\boldsymbol{x}^g)\}_{g=1}^{m_{gs}}$ (see Eq. 19), the FE volumetric weights $\{W_g)\}_{g=1}^{m_{gs}}$ and the snapshot matrix of stresses $\mathbf{A}^{\boldsymbol{\sigma}}$, apply the Empirical Cubature Method, explained in Appendix A, to determine the subset of *ECM* integration points $\mathbf{Z} \subset \{1, 2 \dots m_{gs}\}$, as well as their associated volumetric weights $\{\omega(\boldsymbol{x}_g)\}_{g\in\mathbf{Z}}$.
6. *Straining interface modes.* Use function STRMODP() in Expr. (110) to determine the (tentative) straining modes matrices $\hat{\boldsymbol{V}}_i$ for the fictitious interfaces ($i = 1, 2 \dots l$). For 2D and 3D periodicity, check whether inequality (111) holds. If not, compute matrix $\boldsymbol{D}$ in Eq.(117) and invoke function KINEMDOF() (described in Algorithm 5) to recompute the basis matrices for the interfaces.
7. *Reduced-order matrices.* For each ECM integration point, compute the *interscale* B-matrices defined in Eq. 65: $\{\boldsymbol{\mathcal{B}}(\boldsymbol{x}_g)\}_{g\in\mathbf{Z}}$. Likewise, for each distinct material, and for each non-interface boundary, determine $\{\boldsymbol{\mathcal{N}}^k\}_{k=1}^{n_{mat}}$ (Eq. 141) and $\{\bar{\boldsymbol{\mathcal{N}}}^g\}_{g=1}^{b_{non}}$ (Eq. 142), respectively.

Box 8.1: Offline operations

Once the reduced-order matrices outlined in Box 8.1 have been computed, the solution of the non-linear coarse-scale problem (the *online* stage) may be addressed like any standard nonlinear finite element problem. Consider a time discretization of the interval of interest $[t_0, t_f] = \bigcup_{n=1}^{N_{stp}} [t_n, t_{n+1}]$. The initial data at a given time step $t = t_n$ includes the coarse-scale DOFs vector $\boldsymbol{a}^n$ and the fine-scale stresses, strains and internal variables at the ECM integration points of each coarse-scale element: $\{\boldsymbol{\sigma}_n^e(\boldsymbol{x}_g), \boldsymbol{\varepsilon}_n^e(\boldsymbol{x}_g), \boldsymbol{\xi}_n^e(\boldsymbol{x}_g)\}_{g\in\mathbf{Z}}$ $(e = 1, 2 \dots M)$. On the other hand, the external inputs at $t = t_{n+1}$ comprise the prescribed displacements $\bar{\boldsymbol{v}}^{n+1}$ (c.f Eq. 79), the tractions applied on non-interface boundaries $\{\boldsymbol{t'}_{n+1}^{e(k)}\}_{k=1}^{b_{non}}$ $(e = 1, 2 \dots M)$, and the vector of coarse-scale boundary forces $\bar{\boldsymbol{\mathcal{F}}}_I^{n+1}$ $(I = 1, 2 \dots N)$. The iterative procedure to update the solution is sketched in Box 8.2. In step 8, $\boldsymbol{\mathcal{K}}_{\mathbf{ll}}^{(k)}$ and $\bar{\boldsymbol{\mathcal{R}}}_{\mathbf{l}}^{(k)}$ refer to the block matrices of the tangent stiffness matrix and residual, respectively, associated to the unknown DOFs $\mathbf{l}$. If the boundary conditions are affine, that is, if $\boldsymbol{J} \neq \mathbf{0}$ in Eq. (79), the expression for these matrices have to be modified as follows

$$\begin{aligned} &\boldsymbol{\mathcal{K}}_{\mathbf{ll}} = \begin{bmatrix} \boldsymbol{\mathcal{K}}_{\boldsymbol{mm}} & \boldsymbol{\mathcal{K}}_{\boldsymbol{mf}} \\ \boldsymbol{\mathcal{K}}_{\boldsymbol{fm}} & \boldsymbol{\mathcal{K}}_{\boldsymbol{ff}} \end{bmatrix}; \quad \boldsymbol{\mathcal{K}}_{\boldsymbol{mm}} \leftarrow \boldsymbol{\mathcal{K}}_{\boldsymbol{mm}} + (\boldsymbol{J}^T \boldsymbol{\mathcal{K}}_{\mathbf{rm}} + \boldsymbol{\mathcal{K}}_{\mathbf{mr}} \boldsymbol{J}) + \boldsymbol{J}^T \boldsymbol{\mathcal{K}}_{\mathbf{rr}} \boldsymbol{J} \\ &\boldsymbol{\mathcal{K}}_{\boldsymbol{mf}} \leftarrow \boldsymbol{\mathcal{K}}_{\boldsymbol{mf}} + \boldsymbol{J}^T \boldsymbol{\mathcal{K}}_{\mathbf{rf}}; \quad \boldsymbol{\mathcal{K}}_{\boldsymbol{fm}} \leftarrow \boldsymbol{\mathcal{K}}_{\boldsymbol{fm}} + \boldsymbol{\mathcal{K}}_{\boldsymbol{fr}} \boldsymbol{J}; \quad \bar{\boldsymbol{\mathcal{R}}}_{\mathbf{l}} \leftarrow \begin{bmatrix} \bar{\boldsymbol{\mathcal{R}}}_{\boldsymbol{m}} + \boldsymbol{J}^T \bar{\boldsymbol{\mathcal{R}}}_{\mathbf{r}} \\ \bar{\boldsymbol{\mathcal{R}}}_{\boldsymbol{f}} \end{bmatrix} \end{aligned} \tag{119}$$

Given $\boldsymbol{a}^n$, $\bar{\boldsymbol{v}}^{n+1}$, $\{\boldsymbol{\sigma}_n^e, \boldsymbol{\varepsilon}_n^e, \boldsymbol{\xi}_n^e\}(\boldsymbol{x}_g)_{g\in\mathbf{Z}}$, $\{\boldsymbol{t'}_{n+1}^{e(k)}\}_{k=1}^{b_{non}}$ $(e = 1, 2 \dots M)$, $\bar{\boldsymbol{\mathcal{F}}}_I^{n+1}$ $(I = 1, 2 \dots N)$.

1. Initialization. $k = 0$, $\boldsymbol{a}_{\boldsymbol{m}}^{(k)} = \boldsymbol{a}_{\boldsymbol{m}}^n$, $\boldsymbol{a}_{\boldsymbol{f}}^{(k)} = \boldsymbol{a}_{\boldsymbol{f}}^n$. Set $\boldsymbol{a}_{\mathbf{l}}^{(k)} = [\boldsymbol{a}_{\boldsymbol{m}}^{(k)^T}, \boldsymbol{a}_{\boldsymbol{f}}^{(k)^T}]^T$.
2. Slave DOFs: $\boldsymbol{a}_{\mathbf{r}}^{(k)} = \bar{\boldsymbol{v}}^{n+1} + \boldsymbol{J}\boldsymbol{a}_{\boldsymbol{m}}^{(k)}$ (cf. Eq. 79). Set $\boldsymbol{a}^{(k)} = [\boldsymbol{a}_{\mathbf{r}}^{(k)^T}, \boldsymbol{a}_{\mathbf{l}}^{(k)^T}]^T$
3. For each coarse-scale element $e = 1, 2 \dots M$, compute
   (a) Strains: $\boldsymbol{\varepsilon}_{(k)}^e(\boldsymbol{x}_g) = \boldsymbol{\mathcal{B}}(\boldsymbol{x}_g)\boldsymbol{a}^{e(k)}$, $g \in \mathbf{Z}$, where $\boldsymbol{a}^{e(k)} = \mathbb{L}^e \boldsymbol{a}^{(k)}$.
   (b) Stresses, algorithmic tangent moduli and internal variables (via constitutive Eq. 7):
   $$\{\boldsymbol{\sigma}_{(k)}^e, \boldsymbol{C}_{(k)}^e, \boldsymbol{\xi}_{(k)}^e\}(\boldsymbol{x}_g)_{g\in\mathbf{Z}} = \mathcal{H}\Big(\{\boldsymbol{\varepsilon}_{(k)}^e, \boldsymbol{\sigma}_n^e, \boldsymbol{\xi}_n^e\}(\boldsymbol{x}_g)_{g\in\mathbf{Z}}\Big)$$
   (c) Internal forces (Eq. 68): $\boldsymbol{\mathcal{F}}_{\boldsymbol{int}}^{e(k)} := \sum_{g\in\mathbf{Z}} \boldsymbol{\mathcal{B}}^T(\boldsymbol{x}_g)\omega(\boldsymbol{x}_g)\boldsymbol{\sigma}_{(k)}^e(\boldsymbol{x}_g)$
   (d) Body forces (if $k = 0$, Eq. 75): $\boldsymbol{\mathcal{F}}_{\boldsymbol{b}}^e = \sum_{i=1}^{n_{mat}} \boldsymbol{\mathcal{N}}^{i^T} (\rho^i \hat{\boldsymbol{Q}}^{e^T} \boldsymbol{g}) + \sum_{g=1}^{b_{non}} \boldsymbol{\mathcal{N}}^{g^T} \boldsymbol{t'}_{n+1}^{e(g)}$.
   (e) Residual forces: $\boldsymbol{\mathcal{R}}^{e(k)} = \boldsymbol{\mathcal{F}}_{\boldsymbol{int}}^{e(k)} - \boldsymbol{\mathcal{F}}_{\boldsymbol{b}}^e$
   (f) Tangent stiffness matrix (Eq. 73): $\boldsymbol{\mathcal{K}}^{e(k)} = \sum_{g\in\mathbf{Z}} \omega(\boldsymbol{x}_g)\boldsymbol{\mathcal{B}}^T(\boldsymbol{x}_g)\boldsymbol{C}_{(k)}^e(\boldsymbol{x}_g)\boldsymbol{\mathcal{B}}(\boldsymbol{x}_g)$
4. Assemble the vector of element residual forces: $\boldsymbol{\mathcal{R}}^{(k)} = \sum_{e=1}^{M} \mathbb{L}^{e^T} \boldsymbol{\mathcal{R}}^{e(k)}$.
5. Assemble the vector of boundary forces: $\bar{\boldsymbol{\mathcal{F}}} = \sum_{I=1}^{N} \bar{\boldsymbol{L}}_I^T \bar{\boldsymbol{\mathcal{F}}}_I^{n+1}$
6. Compute the global residual $\bar{\boldsymbol{\mathcal{R}}}^{(k)} = \boldsymbol{\mathcal{R}}^{(k)} - \bar{\boldsymbol{\mathcal{F}}}$. If $\|\bar{\boldsymbol{\mathcal{R}}}_{\mathbf{l}}^{(k)}\| \leq TOL$, go to step 9.
7. Assemble the global tangent stiffness matrix: $\boldsymbol{\mathcal{K}}^{(k)} = \sum_{e=1}^{M} \mathbb{L}^{e^T} \boldsymbol{\mathcal{K}}^{e(k)} \mathbb{L}^e$.
8. Update coarse-scale displacements: $\boldsymbol{a}_{\mathbf{l}}^{(k+1)} = \boldsymbol{a}_{\mathbf{l}}^{(k)} - \boldsymbol{\mathcal{K}}_{\mathbf{ll}}^{(k)^{-1}} \bar{\boldsymbol{\mathcal{R}}}_{\mathbf{l}}^{(k)}$. Set $k \leftarrow k + 1$, and go to step 2.
9. Converged step. Set $\boldsymbol{a}^{n+1} = \boldsymbol{a}^{(k)}$ and $\{\boldsymbol{\sigma}_{n+1}^e, \boldsymbol{\varepsilon}_{n+1}^e, \boldsymbol{\xi}_{n+1}^e\}(\boldsymbol{x}_g)_{g\in\mathbf{Z}} = \{\boldsymbol{\sigma}_{(k)}^e, \boldsymbol{\varepsilon}_{(k)}^e, \boldsymbol{\xi}_{(k)}^e\}(\boldsymbol{x}_g)_{g\in\mathbf{Z}}$. Determine $\boldsymbol{q}^e$ and $\boldsymbol{\alpha}^e$ from Eq.(62) and Eq.(61), respectively $(e = 1, 2 \dots M)$.

Box 8.2: Coarse-scale, nonlinear problem

It should be remarked that all the involved operations, as well as the input and output variables, are of reduced size (independent of the size of the FE discretization of the unit cell). Nevertheless, for post-processing purposes, one can reconstruct the fine-scale displacement and stresses at any time step of the analysis. Displacements can be recovered using Eq. 14. Recovery of stresses, on the other hand, can be carried out by a least-square fitting (see e.g., [21]):

$$\boldsymbol{\sigma}^e(\boldsymbol{x}_g) = \boldsymbol{\mathcal{R}}(\boldsymbol{x}_g)\boldsymbol{S}_{\mathbf{Z}}^e, \quad g = 1, 2 \dots m_{gs}, \tag{120}$$

where $\boldsymbol{S}_{\mathbf{Z}}^e = [\boldsymbol{\sigma}^{e^T}(\boldsymbol{x}_{\boldsymbol{Z}_1}), \boldsymbol{\sigma}^{e^T}(\boldsymbol{x}_{\boldsymbol{Z}_2}) \dots \boldsymbol{\sigma}^{e^T}(\boldsymbol{x}_{\boldsymbol{Z}_{m_{gs}^*}})]^T$; and $\boldsymbol{\mathcal{R}}(\boldsymbol{x}_g) = \boldsymbol{\Lambda}_{\boldsymbol{x}_g}(\boldsymbol{\Lambda}_{\mathbf{Z}}^T \boldsymbol{\Lambda}_{\mathbf{Z}})^{-1} \boldsymbol{\Lambda}_{\mathbf{Z}}^T$ (stress reconstruction operator at Gauss point $\boldsymbol{x}_g$). Needless to say, one need not perform the reconstruction for all domains. Rather, stresses and/or strains at the ECM integration points may be employed to discriminate which are the "critical" domains and apply to such domains the corresponding downscaling equations.

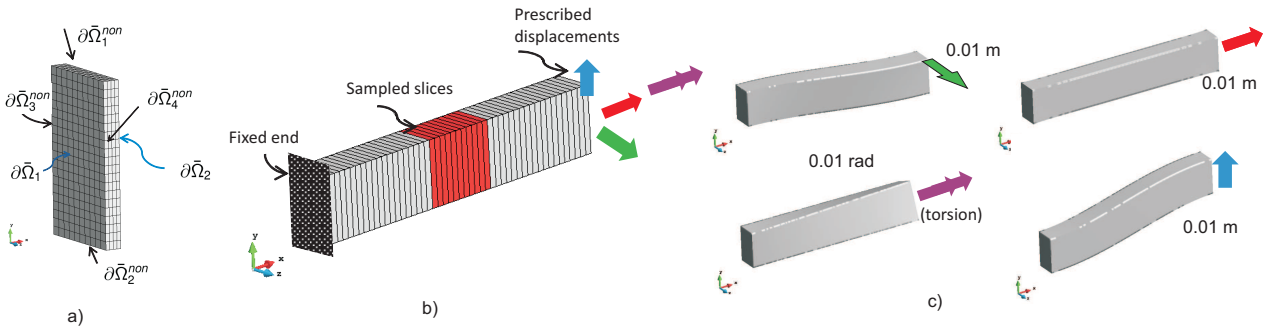

Figure 4: Rectangular cross-section beam problem. a) FE discretization of the unit cell. Definition of interface ($\partial\bar{\Omega}_1$ and $\partial\bar{\Omega}_2$) and non-interface ($\partial\bar{\Omega}^i_{non}$, $i = 1, 2, 3, 4$) boundaries . b) Structure and boundary conditions using for "training". The unit cells whose displacement, reactions and stress solutions are included in the snapshot matrices are highlighted in red. c) Deformed shapes of the four tests used for training.

9. Numerical assessment

9.1. Rectangular cross-section beam

We begin the numerical assessment by deriving a coarse-scale model for the linear elastic behavior of a 3D prismatic structure made up by repeating, along the x-axis, the rectangular slice shown in Figure 4.a (1D periodicity). The transverse dimensions are $h_y = 0.1\ m$ and $h_z = 0.06\ m$, while the thickness is $w = h_y/10 = 0.01\ m$. The material is isotropic, with Young's Modulus $E = 70000$ MPa, and Poisson's ration $\nu = 0.3$. The unit cell is discretized using a structured mesh of $m = 4 \times 20 \times 10 = 800$ eight-nodes hexahedral elements, resulting in $n = 1155$ "fine-scale" nodes, and $m_{gs} = 800 \cdot 8 = 6400$ Gauss points. The non-interface boundaries $\partial\bar{\Omega}_{non}$, on the other hand, are divided into the 4 portions shown in Figure 4.a.

9.1.1. Offline stage

As pointed out in Section 7.1, the number of unit cells and the boundary conditions for "training" the reduced-order model depends on the scope of the analysis. In this case, we are interested only in how loads are transmitted along the span of the structure (we shall deal with local effects later). Therefore, by virtue of Saint Venant's principle, the prismatic structure needs to be sufficiently large so as to ignore local effects due to the application of boundary conditions on its ends. We have taken a span of $L = 5h_y = 0.5$ m (thus, the structure has $L/w = 50$ slices). Four distinct 3D FE analyses have been carried out (to meet the requirement stated in Eq. 84). In each of them, the nodal displacements of one the ends of the prismatic structure are set to zero, while the other end is subjected to a rotation around the axis (to capture torsion behavior), a displacement along the x-axis (to capture axial behavior) and transversal displacements in the y and z directions (to capture bending behavior in both axes). These loading conditions are depicted in Figure 4.b, and the corresponding FE deformed shapes in Figure 4.c. The solution in displacements, reactive forces between domains and stresses of the slices of the central portion of the structure, highlighted in red in Figure 4.b, are stored in the corresponding snapshot matrices (step 3 in Box 8.1). Next the SVD is applied on the reactive and displacement matrices (step 4 in Box 8.1). We use for this the partitioned SVD proposed by the author in Ref. [21], using a relative tolerance of $\epsilon_\lambda = 10^{-3}$ for the four block matrices; this yields a total of $p = 6$ self-equilibrating interface force modes. These modes are shown in Figure 5 (top). Next we apply the SVD to the straining displacement matrix, with a tolerance $\epsilon_d = 10^{-4}$, obtaining 8 modes. Following the "alignment" method explained in Section 7.3.1, we determine from the space spanned by these 8 modes the subspace energetically associated with the $p = 6$ reactive modes $\boldsymbol{\Psi}$. In Figure 5 (bottom) we display the deformed shapes of the obtained $p = 6$ straining modes. It is worth noting that, albeit both the SVD and the alignment algorithm are purely "data-driven" procedures, the outcome modes do have a distinct physical interpretation: $\boldsymbol{\Phi_1}$ is clearly the *axial* mode (pure dilatational behavior due to Poisson's effect); $\boldsymbol{\Phi_2}$ displays the typical out-of-plane warping caused by torsion; $\boldsymbol{\Phi_3}$ and $\boldsymbol{\Phi_4}$ may be identified as pure *bending* modes (the cross-section remains approximately plane); while $\boldsymbol{\Phi_5}$ and $\boldsymbol{\Phi_6}$ exhibit the characteristic cubic warping of *shear* deformation produced by simple bending.

Having determined $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$, we can move to the next *offline* step of Box 8.1: the determination of the reduced set of integration points (step 5). The SVD of the stress snapshot matrix (Expr. 123), with tolerance $\epsilon_\sigma = \epsilon_\lambda = 10^{-3}$ gives, as expected, $r = 6$ stress modes. With these modes, and the reduced B-matrices $\boldsymbol{B}^*$ at the $m_{gs} = 6400$ Gauss points of the mesh, we build the internal forces matrix $\boldsymbol{A^f}$ of Eq. (124). This matrix has $p \cdot r = 36$ columns, out of which only $q = 21$

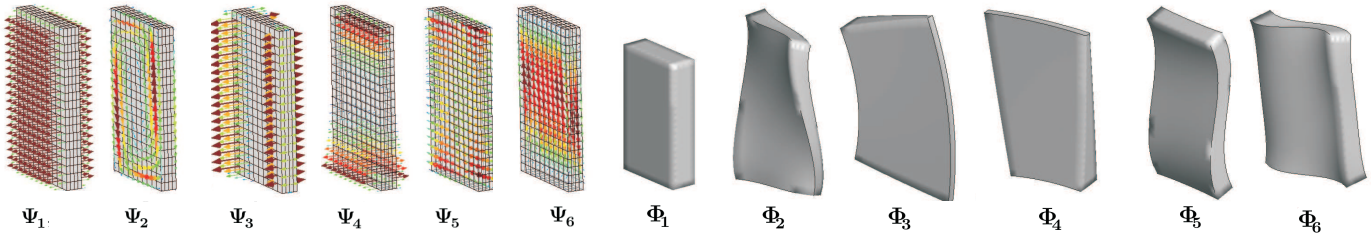

Figure 5: Modes obtained from the training shown in Fig. 4.b. Left: Self-equilibrating reactive modes ($\boldsymbol{\Psi_1}, \boldsymbol{\Psi_2} \ldots \boldsymbol{\Psi_6}$). Right: Straining modes ($\boldsymbol{\Phi_1}, \boldsymbol{\Phi_2} \ldots \boldsymbol{\Phi_6}$).

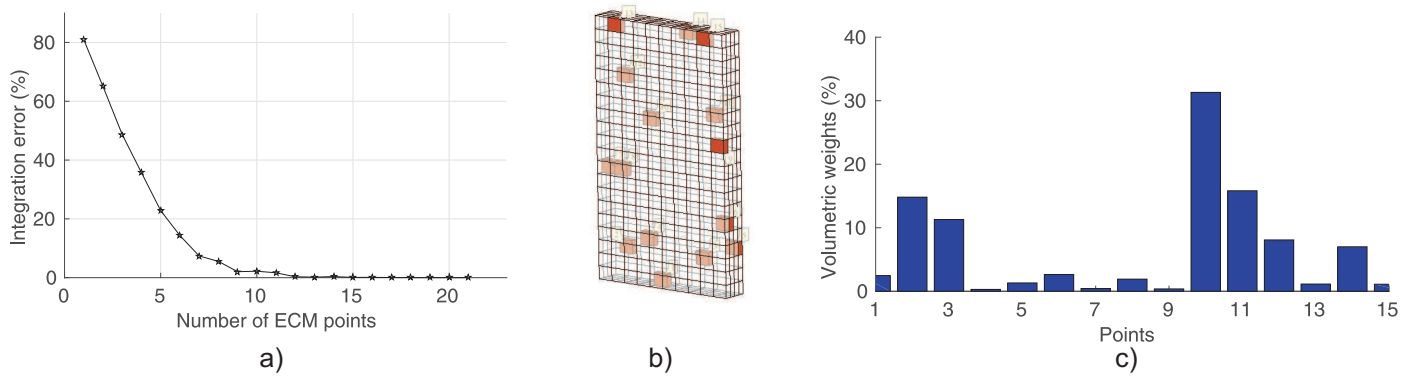

Figure 6: Determination, via the *Empirical Cubature Method* (see Appendix A), of the reduced set of integration weights and their associated volumetric weights. a) Integration error (%) versus number of selected points. b) Location of the finite elements containing the $m^*_{gs} = 15$ points used for integration (1 point per element). c) Volumetric weights, in % of the total volume of the unit cell.

are linearly independent[16] (this follows from applying the SVD, see Expr. 123, with tolerance $\epsilon_f = 10^{-6}$).

Then we employ the *Empirical Cubature Method* (ECM), presented in Appendix A, to select, among the $m_{gs} = 6400$ Gauss points of the FE mesh, a set of m^*_{gs} points such that the integration error is below a certain tolerance (in this case we have set $\epsilon_{ecm} = 10^{-3}$). Figure 6.a contains the graph of relative integration error versus number of points. Zero integration error is achieved when the number of points coincides with the number of internal force modes $q = 21$; on the other hand, for rendering the error below the prescribed tolerance, $m^*_{gs} = 15$ points are required. The location of the finite elements containing these $m^*_{gs} = 15$ points are shown in Figure 6.b, and the associated volumetric weights (in % of the total volume) in Figure 6.c.

The next offline operation (step 6 in Box 8.1) is the determination of how many and which (if any) *straining modes* $\hat{\boldsymbol{V}}_1 = \hat{\boldsymbol{V}}_2$ for the fictitious interfaces are to be incorporated in the formulation. However, since the number of reaction modes is $p = 6$, there can be no straining interface modes (this follows from Eq. 107). Thus, the fictitious interfaces will only experience rigid-body motions, or, in other words, our coarse-scale element only possesses 6 DOFs per node (3 translations and 3 rotations) —just like a standard Timoshenko's beam element.

9.1.2. Online analyses

We now focus on examining the predictive capabilities of the above derived coarse-scale model. We begin by conducting the coarse-scale counterparts of the the FE analyses used to "train" the model. The 1D coarse-scale mesh is formed in this case by $M = 50$ two-node elements (and thus $N = 51$ nodes). The Dirichlet boundary conditions are introduced by prescribing the 6 DOFs of the two end nodes: $\boldsymbol{a}_1 = \boldsymbol{0}$ (fixed end at $x = x_{min}$); and $\boldsymbol{a}_N(i) = \gamma$ for the other end at $x = x_{max}$, with $\boldsymbol{a}_N(j) = 0$ for $j \neq i$ ($i = 1$ and $\gamma = 0.01$ m for the axial tests; $i = 2, 3$ and $\gamma = 0.01$ m for the y-bending and z-bending tests, respectively; lastly, $i = 4$ and $\gamma = 0.01$ rad for the torsion tests). Figure 7 shows, for these training tests, the deformed shapes computed by the standard FE (top) and coarse-scale (bottom) models. For comparison purposes,

[16] Interestingly, this figure coincides with the number of independent entries of the $p \times p$ reduced stiffness matrix $\boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi}$, which is symmetric and therefore: $q = (p+1)p/2 = 7 \cdot 6/2 = 21$.

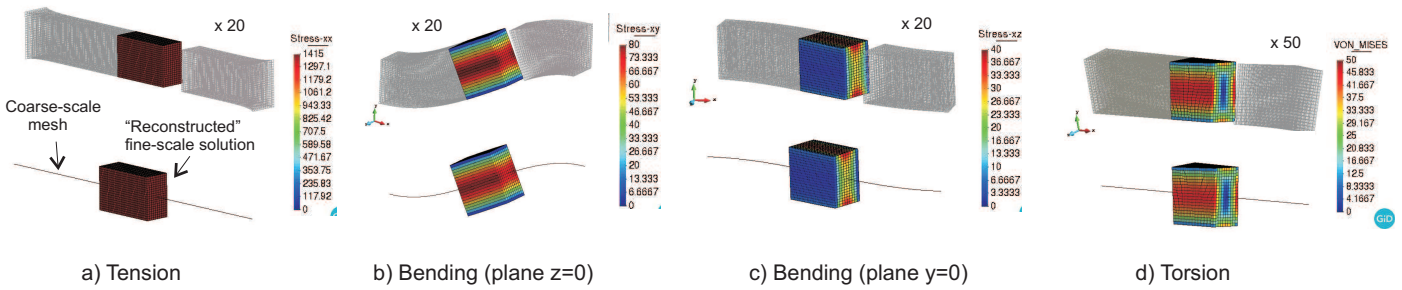

Figure 7: Top: Deformed shapes corresponding to the four FE training tests ($140 \cdot 10^3$ DOFs). The portion displayed with stress values (σ_x, τ_{xy}, τ_{xz} and equivalent Von Mises, in MPa) is the one from which the training snapshots are extracted. Bottom: Deformed coarse-scale mesh ($51 \cdot 6 = 306$ DOFs), along with the *reconstructed* stress distributions (in MPa).

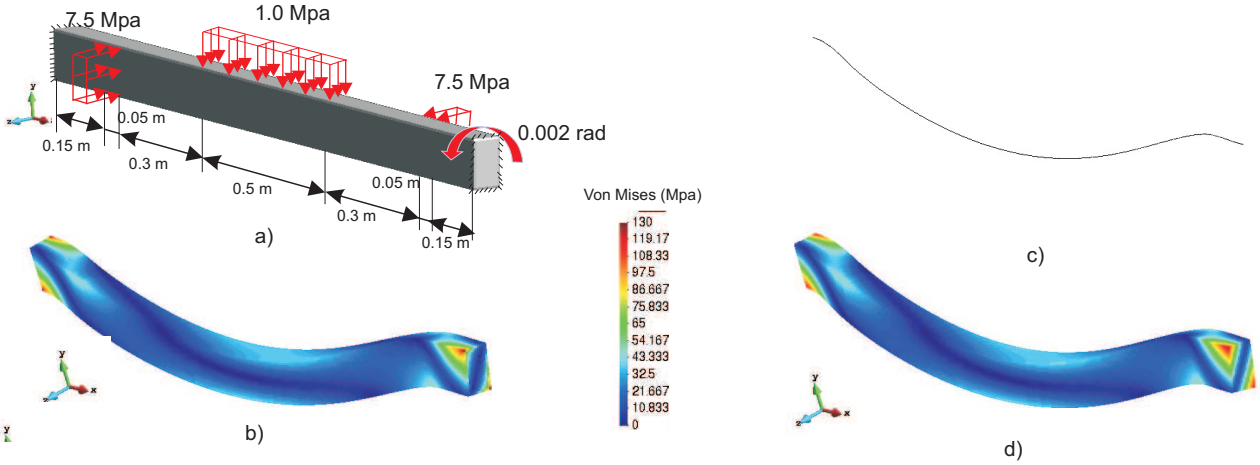

Figure 8: a) Prismatic structure (150 slices) and applied loads employed to test the coarse-scale model. b) Deformed shape (×200) and Von Mises stress computed by the standard FE (fine-scale solution, $\approx 420 \cdot 10^3$ DOFs). c) Deformed shape of the coarse-scale mesh ($6 \cdot 151 = 906$ DOFs). d) Deformed shape and Von Mises equivalent stress computed by *reconstruction* of the coarse-scale solution.

the stress distributions (σ_x, τ_{xy}, τ_{xz} and Von Mises equivalent stress, respectively) corresponding to the slices used for training are displayed in both cases (in the coarse-scale model, these distributions are obtained by the "reconstruction" procedure explained in Section 8). Visually, there are no discernible differences between the two contour plots in any of the four tests. A more quantitative assessment is carried out in Table 1, wherein we compare the non-zero entries of the coarse-scale reactive forces at the ends of the beam with the resultant and moment resultant of the fine-scale nodal reactive forces. In all the four cases, discrepancies are below 3%.

	F_x (test a)	F_y (test b)	M_z (test b)	F_z (test c)	M_y (test c)	M_x (test d)
Fine-scale model	8.48	0.305	$7.62 \cdot 10^{-2}$	0.119	$2.98 \cdot 10^{-2}$	$2.51 \cdot 10^{-3}$
Coarse-scale model	8.40	0.298	$7.47 \cdot 10^{-2}$	0.116	$2.90 \cdot 10^{-2}$	$2.44 \cdot 10^{-3}$
Error (%)	0.94	2.29	1.97	2.52	2.68	1.24

Table 1: Rectangular cross-section cell. Comparative study of the reactive forces (MN) and moments ($MN \cdot m$) at the ends of the beam for the training tests, computed by the standard FE (fine-scale) model, and the "trained" coarse-scale model.

The previous analyses demonstrate that the reduced-order model is *consistent* with its full-order counterpart, in the sense that, at least, it can reproduce the results employed in the training process. But the actual usefulness of a reduced-order model is to be examined in scenarios in which the geometry and the loading conditions are notably different from that used in the training stage. To explore the capabilities of our coarse-scale model in this regard, we have devised the test sketched in Figure 8.a. The prismatic beam is 3 times longer that the one used for training, and it is subjected to lateral loads in the y and z direction (recall that the training tests have no external forces), as well as to a rotation around the x-axis on the surface at $x = 1.5$ m, while the other end, at $x = 0$, remains fixed. Figures 8.b and 8.d display the deformed shape and the contour plot of Von Mises stress computed by the standard FE method and the coarse-scale

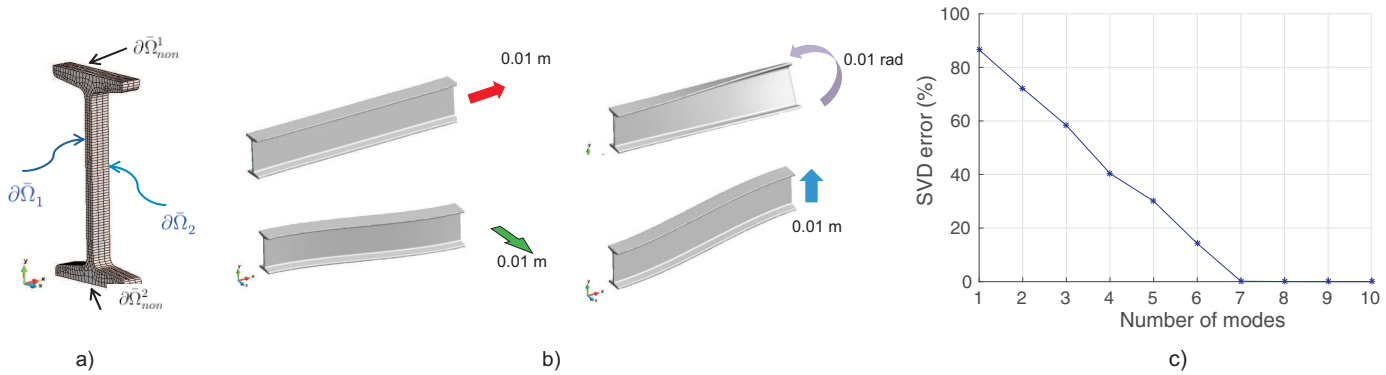

Figure 9: I-shaped cross-section beam. a) FE discretization of the unit cell ($437 \times 3 = 1311$ hexahedral elements and 2292 nodes.). b) Deformed shapes of the four tests used for training. c) SVD error (%) versus number of Lagrange multipliers modes.

model, respectively. The coarse-scale mesh (Figure 8.c) is formed in this case by $M = 150$ two-node elements. The qualitatively resemblance between the deformed shapes and stress values predicted by the FE model and the coarse-scale model (by reconstruction) is huge —despite the fact that the number of DOFs has been reduced by a factor over 500. The only perceptible differences in stresses are detected at the ends, and this is because the reduced-order model has not been trained for capturing the local effects due to Dirichlet boundary conditions (the reduced-order model fails to capture the stress concentrations at the corners). In more quantitative terms, the maximum displacements (euclidean norm) predicted by the standard FE is $d_{max} = 1.46\ mm$, while the coarse-scale model yields $d_{max} = 1.47\ mm$, just 0.68 % above.

9.2. I-shaped cross-section beam

The procedure explained in the foregoing for the slice of rectangular cross-section can be applied to any unit cell, regardless of its geometrical complexity. To illustrate this statement, we address in what follows the case of a straight beam with the I-shaped cross-section (IPN-100 profile, with $h_y = 0.1\ m$ and $h_z = 0.05\ m$) shown in Figure 9. The thickness is $w = h_y/10 = 0.01\ m$, and the material is isotropic, with Young's Modulus $E = 70000$ MPa, and Poisson's ration $\nu = 0.3$. The training parameters (sampled slices, tolerances for the SVD and ECM, and so forth) are exactly the same as the ones employed for the rectangular cross-section beam of Section 9.1. The deformed shapes corresponding to the four training tests are displayed in Figure 9.b. On the other hand, Figure 9.c represents the SVD truncation error[17] for the reactive forces versus the number of modes. According to this graph, the number of modes needed to achieve the required accuracy ($\epsilon_\lambda = 10^{-3}$) is $p = 8$, that is, two more modes than in the case of the rectangular cross-section. These $p = 8$ self-equilibrating reactive modes, together with the deformed shapes of its work-conjugate straining modes are shown in Figure 10.a. Using the same procedure explained for the rectangular cross-section slice, we determine the reduced set of integration points. In this case $m^*_{gs} = 32$ ECM points are required, out of a total of $m_{gs} = 1311 \cdot 8 = 10488$ Gauss points; the location of these ECM points are shown in Figure 10.b.

As for the interface straining modes (step 6 of Box 8.1), application of Algorithm 3 to the $p = 8$ straining and reactive modes of Figure 10.a indicates that, in this case, and as opposed to the rectangular cross-section beam, the fictitious interface does possess *one straining mode*[18]. The deformed shape of such an interface straining mode, displayed in Figure 10.c, has the characteristic signature of a torsion, out-of-plane *warping deformation* —the flanges rotate around the web axis in opposite directions. The coarse-scale element in the case at hand, thus, possesses *7 DOFs per node*, namely, the amplitudes of the $n_{rb} = 6$ rigid-body modes, plus the amplitude of this additional straining mode. It is instructive to note that this result concurs with the predictions of advanced beam theories (cf. Ref. [8] or [30]), according to which standard 6 DOFs-per-node finite beam elements must be supplemented by a "warping" degree-of-freedom in order to accurately capture torsion of thin-walled open sections.

The accuracy of this 7-DOFs, 2-node element is evaluated following the procedure employed for the rectangular cross-section. Qualitative consistency with the FE training tests in terms of stress distribution can be appreciated in Figure 11. Likewise, we can see in Table 2 that the discrepancies between the non-zero entries of the coarse-scale reactive forces at the ends of the beam and the resultant and moment resultant of the fine-scale nodal reactive forces are all below 1.2 %.

[17] This SVD truncation error is defined in Eq. 91.

[18] Indeed, the intersection of the subspaces of reactive modes of both interface boundaries turns out to be of dimension 7, which implies, according to Eq.(107), that $\hat{n}^{int} = 7 - n_{rb} = 1$.

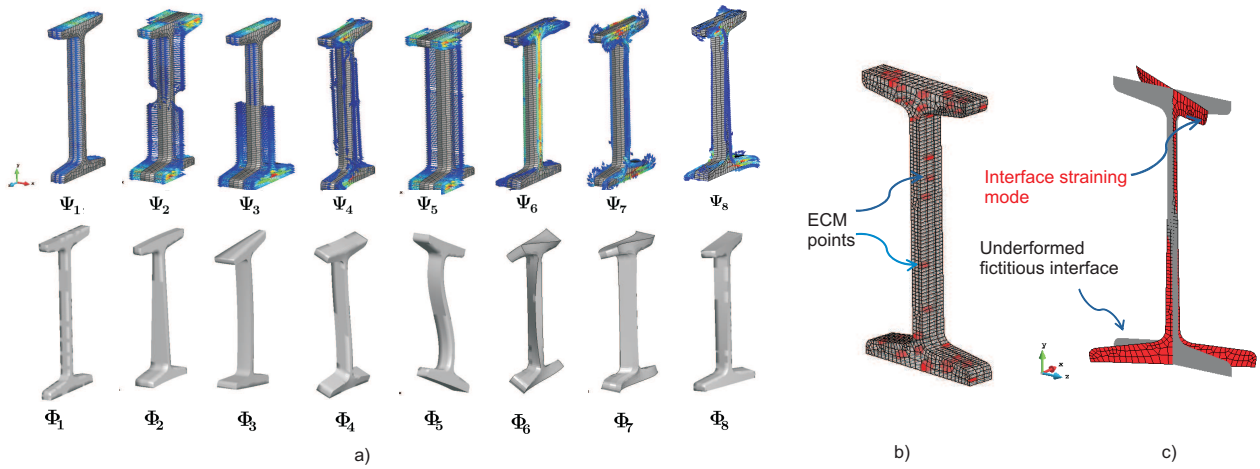

Figure 10: a) Modes obtained from the training shown in Fig. 9.b. Top: Self-equilibrating reactive modes. Bottom: Straining modes. b) Location of the finite elements containing the $m^*_{gs} = 32$ ECM integration points (out of a total of $m_{gs} = 1311 \cdot 8 = 10488$ Gauss points). c) Interface straining mode ($\hat{\boldsymbol{V}}_1 = \boldsymbol{V}_7$) computed by the method described in Algorithm 3.

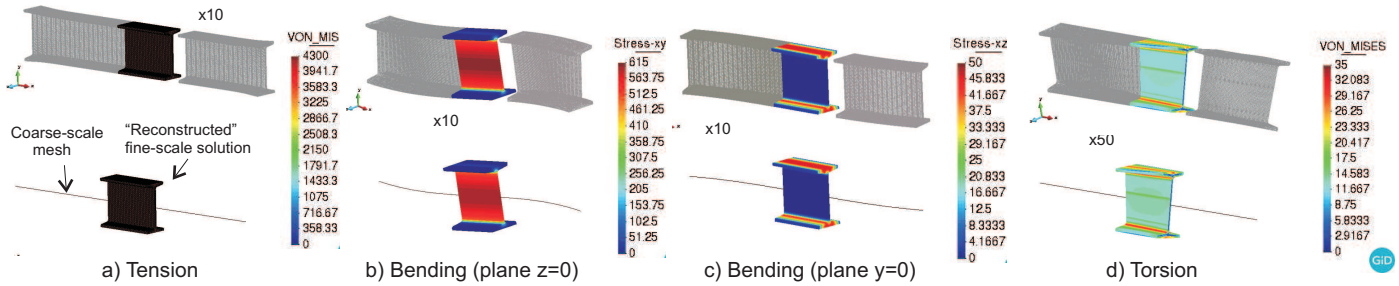

Figure 11: I-shaped cross-section cell. Top: Deformed shapes corresponding to the four FE training tests ($\sim 261 \cdot 10^3$ DOFs). The portion displayed with stress values (Von Mises, τ_{xy}, τ_{xz} and Von Mises, in MPa) is the one from which the training snapshots are extracted. Bottom: Deformed coarse-scale mesh ($51 \cdot 7 = 357$ DOFs), along with the *reconstructed* stress distributions.

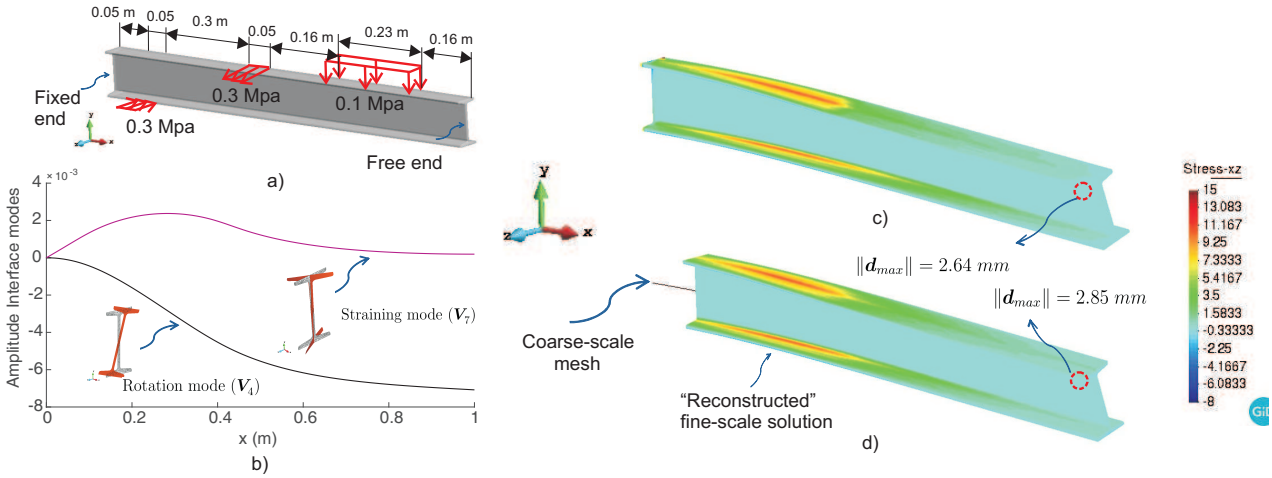

Figure 12: a) Prismatic structure (100 slices) and applied loads employed to test the coarse-scale model for the I-shaped unit cell. b) Evolution of the twist angle and the amplitude of the straining mode (multiplied by a factor 10^4) along the length of the beam. c) Deformed shape ($\times 50$) and tangential stress (MPa) computed by the standard FE d) Deformed shape and (reconstructed) tangential stress (MPa) computed by the the coarse-scale model.

	F_x (test a)	F_y (test b)	M_z (test b)	F_z (test c)	M_y (test c)	M_x (test d)
Fine-scale model	4.49	0.233	$5.85 \cdot 10^{-2}$	$2.41 \cdot 10^{-2}$	$6.02 \cdot 10^{-2}$	$7.77 \cdot 10^{-5}$
Coarse-scale model	4.46	0.231	$5.78 \cdot 10^{-2}$	$2.39 \cdot 10^{-2}$	$5.98 \cdot 10^{-2}$	$7.68 \cdot 10^{-5}$
Error (%)	0.67	0.86	1.20	0.83	0.66	1.15

Table 2: I-shaped cross-section cell. Comparative study of the reactive forces (MN) and moments ($MN \cdot m$) at the ends of the beam for the training tests, computed by the standard FE (fine-scale) model, and the "trained" coarse-scale model.

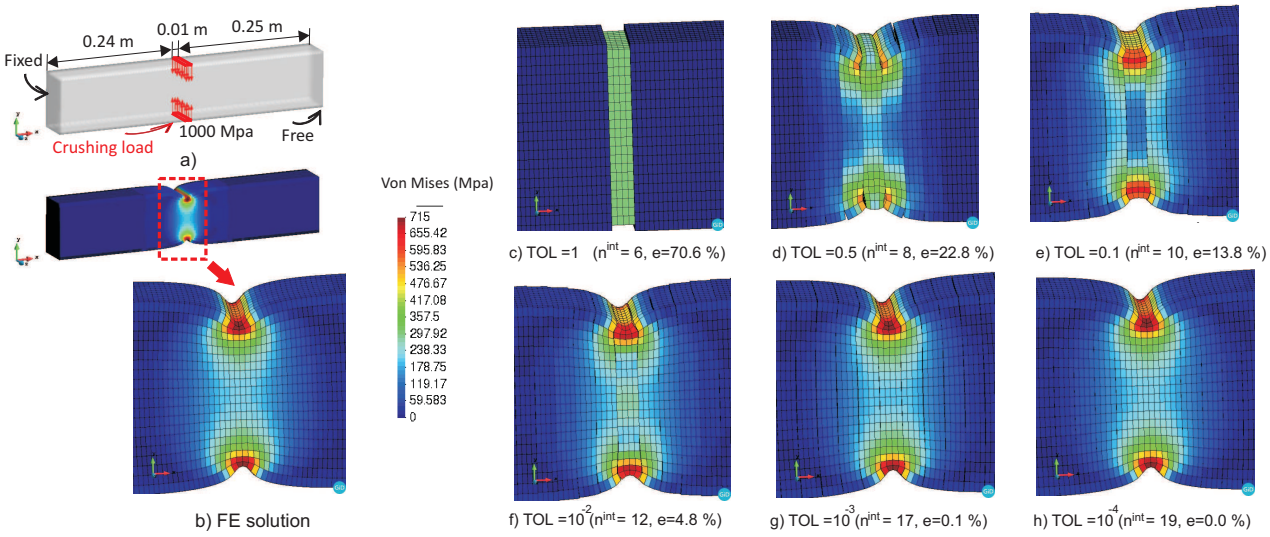

Figure 13: a) Geometry and boundary conditions of the training test devised to "teach" the coarse-scale model how to behave under crushing loads in the y-direction. b) Deformed shaped ($\times$ 50) and contour plot of Von Mises stress computed by the standard FE method. c) to h) Deformed shapes ($\times$50) and (reconstructed) contour plots of Von Mises stress computed by the coarse-scale model for 6 distinct SVD tolerances. In each case, we provide the number of coarse-scale DOFs per node n^{int} as well as the error with respect to the FE solution in capturing the maximum displacement (in %).

We complete the accuracy assessment by carrying out a comparative analysis with boundary conditions and geometry different from those employed in training. Such conditions are depicted in Figure 12.a. The prismatic beam is 1 m longer (twice the length of the training structure), it is fixed on one end, with the other end completely unconstrained[19], and it is subjected to tangential tractions and normal tractions along its length. The deformed shaped (amplified by a factor $f = 50$) and the stress τ_{xz} (in MPa) computed by the standard FE method ($\sim 520 \cdot 10^3$ DOFs) is shown in Figure 12.c. Its coarse-scale counterpart ($7 \cdot 101 = 707$ DOFs), on the other hand, is displayed in Figure 12.d. Notably, the coarse-scale model is able to capture the stress concentration occurring at the intersection between the top flange and the web —in the portion of beam subjected to the torsion caused by the tangential tractions. In terms of displacements, the coarse-scale model overpredicts the maximum displacement (which takes at the top-left corner of the free end) by a factor of $(2.85 - 2.64)/2.64 \cdot 100 \approx 8\%$, an error which is notably higher than in the rectangular cross-section case, but which is still within acceptable accuracy margins. Lastly, we plot in Figure 12.b the evolution along the length of the beam of the amplitude of both this mode ($\boldsymbol{V}_7 = \hat{\boldsymbol{V}}_1$) and the rigid-body mode $\boldsymbol{V}_4$. Observe that both curves exhibit a markedly nonlinear behavior, a fact that is in accordance with the analytical predictions of advanced torsion theories —according to these theories, the evolution of the twist angle is ruled by hyperbolic functions, see e.g. Ref. [37], pag. 436.

9.3. Local effects

The two examples discussed in the foregoing were intended to illustrate the ability of the proposed approach in automatically generating 2-node finite elements *consistent*, in the *Saint Venant* sense (i.e. ignoring *local effects*[20]), with the underlying 3D elasticity theory. In what follows, we describe how the methodology can be easily adapted to cope also with cases in which *local effects* become relevant and, thus, are to be incorporated.

[19]Note that the training structure is fixed on both ends

[20]The Saint Venant problem (see e.g. Refs. [26, 36]) consists in finding the stress and displacement field that appear in transmitting loads along prismatic beams of constant cross-section under the hypothesis that the local effects caused by such loads are negligible.

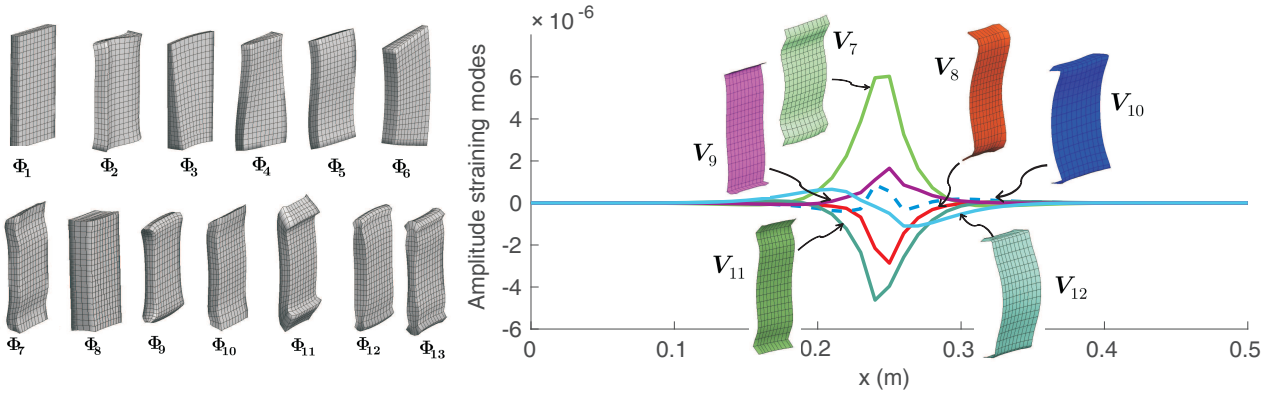

Figure 14: Left: $p = 13$ domain straining modes corresponding to the coarse-scale model of Figure 13.f. Right: For this coarse-scale mode, longitudinal evolution of the amplitude of the $\hat{n}^{int} = n^{int} - n_{rb} = 6$ interface straining modes ($\boldsymbol{V}_7$, $\boldsymbol{V}_8$, ... $\boldsymbol{V}_{12}$) .

To this end, let us return to the rectangular cross-section slice of Section 9.1. Suppose we wish to devise a coarse-scale element able to capture, aside from torsion, axial and bending behaviors, the deformations and stresses caused by *crushing* loads in the y-direction[21]. To "teach" the coarse-scale model how to behave under such loads, we simply enrich the training database used in the example of Section 9.1 with the FE test shown in Figure 13.a, in which a crushing pressure of 1000 MPa is applied to the central slice. Following the offline protocol of Box 8.1, the solution of displacement, reactive forces and stresses of the 20 slices of the central portion of the structure (see Figure 13.b) are stored in pertinent block matrices, that are added to the corresponding 4 block matrices obtained from the basic training of Figure 4.c. Then we apply the already mentioned blockwise SVD to the resulting five block matrices. To gain insight on how the increase of interface straining modes contributes to the accuracy of the coarse-scale predictions, we launch 6 distinct coarse-scale analyses with varying values of the SVD tolerance for the block matrix corresponding to the reactive forces in this crushing test (from $\epsilon_\lambda^{crush} = 1$ to $\epsilon_\lambda^{crush} = 10^{-4}$); all the other parameters remain fixed and equal to the values used in Section 9.1.

Figures 13.c to 13.h show the resulting deformed shapes (×50) and the (reconstructed) contour plots of Von Mises stress in the central portion of the beam. Figure 13.c corresponds to the limiting case in which the crushing tests is ignored, and therefore $n^{int} = 6$. Observe that the loaded slice is squeezed under the applied loads, yet the resulting deformation is not transmitted to the adjacent slices, which remain unaltered—this is because this 6 DOFs-per-node element is able to transmit only the resultant and moment resultant of the applied loads, which in this case are zero. As the SVD tolerance decreases, the number of interface modes n^{int} increases, and, as a consequence, the displacement and stresses discontinuities between slices become less pronounced. For $TOL = 10^{-4}$ (Figure 13.h), the jumps in displacements are imperceptible, and the resulting stress distribution is practically identical to the "exact" FE distribution shown in Figure 13.b. The *local* character of the straining DOFs can be appreciated in Figure 14 (right), where we plot the amplitude along the beam of the $\hat{n}^{int} = 12 - n_{rb} = 6$ interface straining modes of the coarse-scale model of Figure 13.f —observe that the amplitudes decay in an exponential fashion as we move away from the loaded slice. For completeness, in Figure 14 (left) we also plot for this case the deformed shapes of the $p = 13$ (cell) straining modes.

To evaluate the performance of the coarse-scale model in a case distinct from the training problems, we employ the structure (150 slices) and combined loads shown in Figure 15.a. Notice that now the crushing loads are not applied on a single slice, but varies along the the central portion of the beam in a piece-wise constant fashion. Figures 15.b, 15.c and 15 display the solutions corresponding to the standard FE, and the coarse-scale model with $n^{int} = 12$ and $n^{int} = 19$ DOFs per (coarse-scale) node, respectively. It can be seen that the stress distributions computed by both coarse-scale models bear a close resemblance to its full-order counterpart. In the case of $n^{int} = 19$ DOFs (Figure 15.d), they are practically indistinguishable, while in the $n^{int} = 12$ DOFs (Figure 15.c) mild discontinuities are observed in the part in which the crushing loads are applied (this can be also appreciated in terms of displacements in the enlarged view of 15.d). Nevertheless, it should be highlighted that the maximum displacement predicted by the 12 DOFs-per-node model is identical to that of the 19 DOFs-per-node model, a fact that shows that, remarkably, the reduced-order solution can tolerate a certain degree of discontinuities at the fine-scale without affecting the overall quality of coarse-scale solutions.

[21] Uniform tractions of equal value and opposite sign normal to the top and bottom boundary surfaces of the beam (the non-interface boundaries $\partial\bar{\Omega}^1_{non}$ and $\partial\bar{\Omega}^2_{non}$).

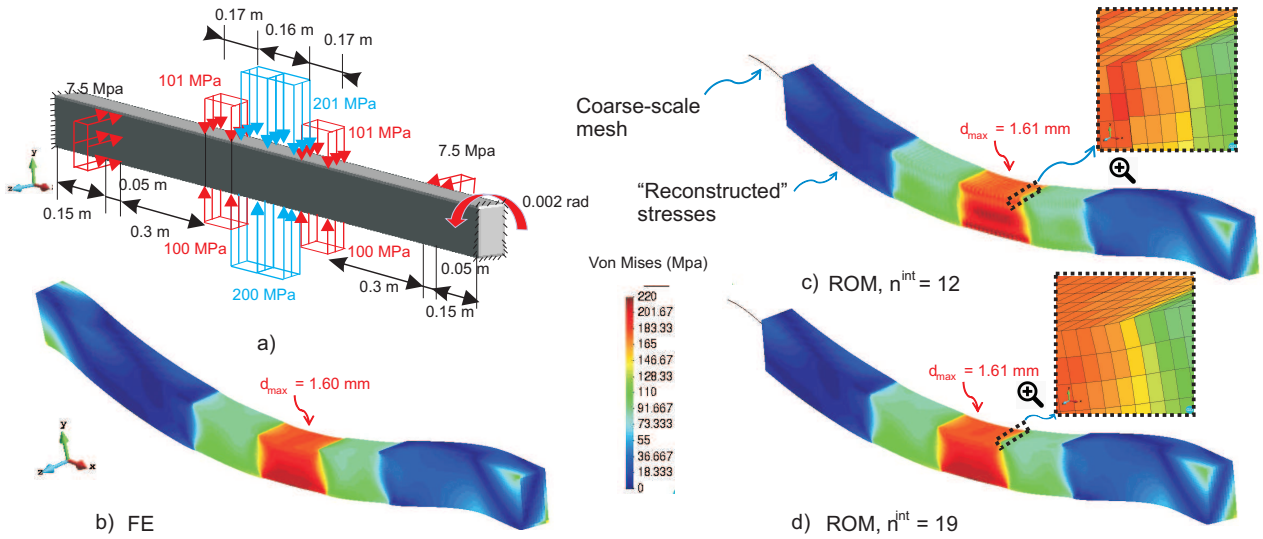

Figure 15: a) Prismatic structure (150 slices) and applied loads employed to test the coarse-scale model for the local effects case. b) Deformed shape (×75) and Von Mises stresses computed by the standard FE (∼ $420 \cdot 10^3$ DOFs) c) Deformed shape and (reconstructed) Von Mises stresses computed by the the coarse-scale model using $n^{int} = 12$ DOFs per node $(151 \cdot 12 = 1812$ DOFs). d) Idem for $n^{int} = 19$ $(151 \cdot 19 = 2869$ DOFs)

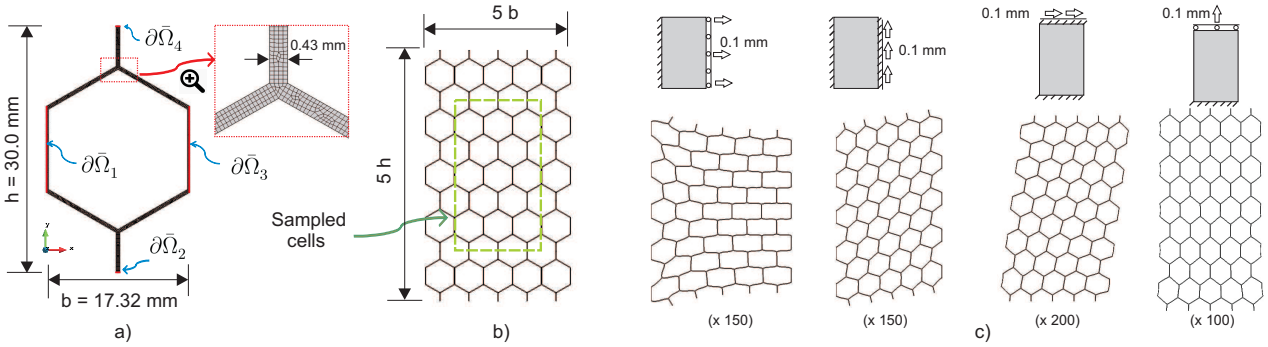

Figure 16: a) FE discretization of the hexagonal cell. b) 5-by-5 cell structure used for "training" (in the linear range). The unit cells whose displacement, reactions and stress solutions are included in the snapshot matrices are the cells enclosed by the dashed line. c) Boundary conditions and FE deformed shapes corresponding to the four training tests.

9.4. Cellular solid (2D)

9.4.1. Linear regime

We next address the derivation of a coarse-scale model for the linear elastic behavior —inelastic behavior will be treated in the next Section— of 2D hexagonal structures (plane strain), made up by tiled copies of the regular hexagonal cell of Figure 16.a. The material is isotropic, with Young's Modulus $E = 70000$ MPa, and Poisson's ratio $\nu = 0.3$. The unit cell is discretized using $m = 2622$ four-node quadrilateral elements (with 4 Gauss points per element), resulting in $n = 3314$ nodes. To train the coarse-scale model, we use the 5-by-5 cells[22] structure shown in Figure 16.b. Likewise, to satisfy the requirement embodied in Eq.(84), we carry out the four FE tests depicted in Figure 16 —2 tests for each periodicity direction. We then apply the partitioned SVD on the self-equilibrating snapshots obtained from these FE analyses (using a tolerance $\epsilon_\lambda = 10^{-4}$ for each of the 4 block matrices). This leads to $p = 15$ modes. We show in Figure 17.a the corresponding work-conjugate straining patterns (computed by the alignment method of Section 7.3.1). As for the ECM integration points (step 5 of Box 8.1), we employ the following tolerances for stresses, internal forces and error integration: $\epsilon_\sigma = \epsilon_f = 10^{-4}$ and $\epsilon_{ecm} = 10^{-3}$, respectively. This results in $r = 15$ stress modes, $q = 64$ internal force

[22]The first question that arises when dealing with the *training* process is how many unit cells should be employed to construct the training domain. To answer this question, we carried out analyses with structures of increasing number of domains (from 3-by-3 to 10-by-10 cells), including in the snapshot matrices only the interior cells. The conclusion was that the computed modes are the same for domains of size equal or larger than 5-by-5.

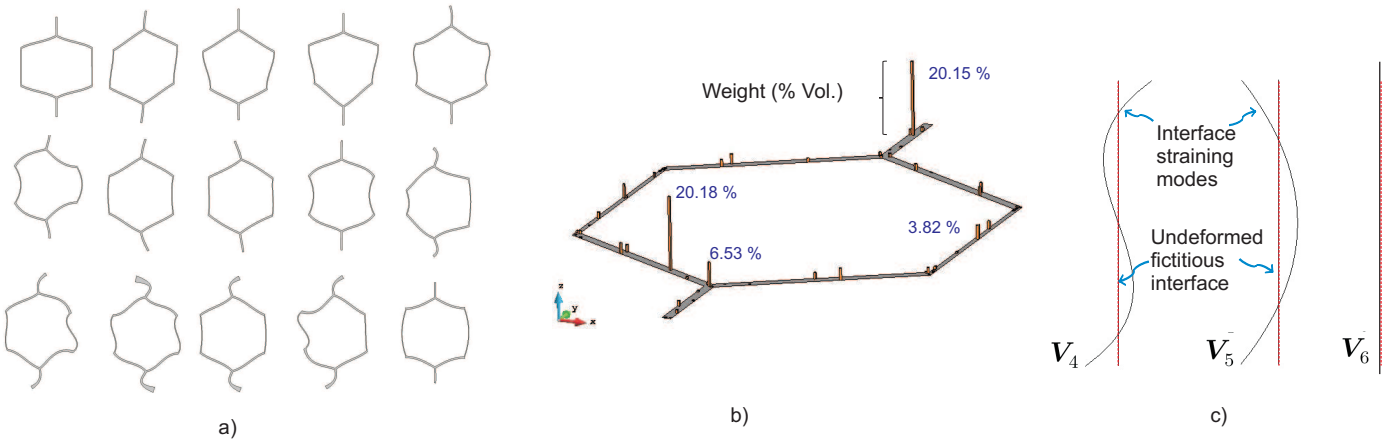

Figure 17: a) $p = 15$ straining modes obtained from the training tests of Figure 17.c. b) Location of the $m^*_{gs} = 60$ integration points chosen by the Empirical Cubature Method (among a total of $m_{gs} = 4 \cdot 2622 = 10488$ Gauss points). The bars represent the volumetric weights of each point (in % of the total volume). c) Straining modes computed by Algorithm 5. for the boundary interfaces $\bar{\Gamma}_1$ and $\bar{\Gamma}_3$

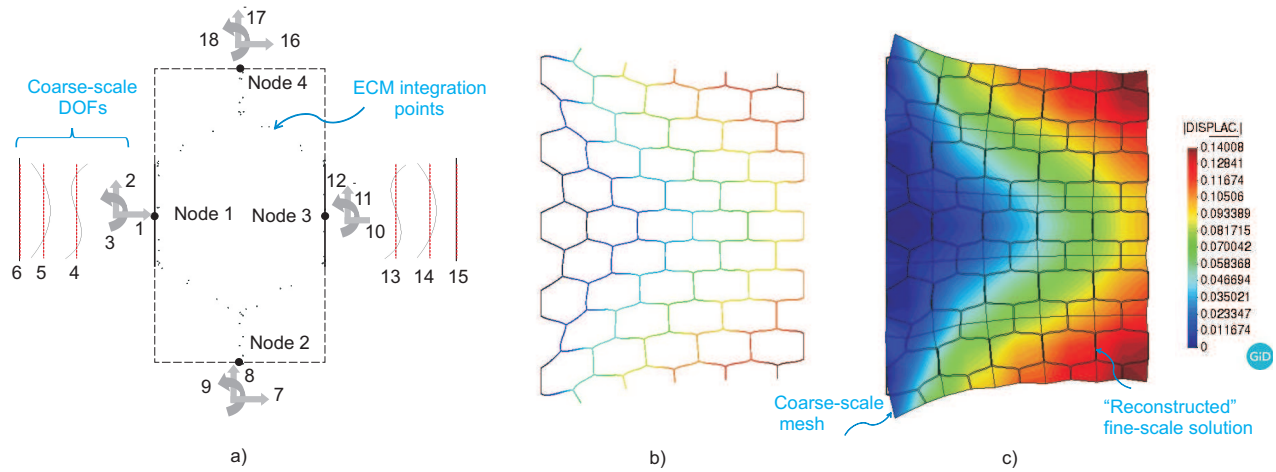

Figure 18: Linear elastic regime. a) Resulting 4-node coarse-scale finite element for the hexagonal unit cell (18 DOFs and 60 integration points). b) and c) Deformed shape (x150) and contour plot of the norm of displacements (in m) corresponding to the stretching training test (in the x-direction). b) is the solution computed by the standard FE, while c) is the solution computed using the 5-by-5 coarse-scale finite element mesh (along with the "reconstructed" fine-scale deformed mesh).

modes, and $m^*_{gs} = 60$ ECM points (out of a total of $m_{gs} = 4 \cdot 2622 = 10488$ Gauss points). The location and associated volumetric weights of these $m^*_{gs} = 60$ ECM points are displayed in the bar graph of Figure 17.b. Observe that the four points with highest weights account for more than 50 % of the total volume[23].

Regarding the straining modes for each pair of fictitious interfaces (step 6 in Box 8.1), application of Algorithm 3 yields $\hat{n}^{int}_1 = \hat{n}^{int}_3 = 6$ and $\hat{n}^{int}_2 = \hat{n}^{int}_4 = 0$, a number of modes that violates the stability condition (111), and consequently, recourse to the (more conservative) procedure outlined in Algorithm 5 is to be made. This, in turn, furnishes no straining modes for the top and bottom interfaces, and, for the other two interfaces, the three straining modes (cubic, parabolic and extensional) shown in Figure 17.c. The resulting *coarse-scale finite element* is pictorially represented in Figure 18.a. It is conceptualized as a *quadrilateral* element with four midside nodes (the centroids of the interface boundaries). The number of DOFs at nodes 1 and 3 are 6 (2 translations, 1 rotation, and the 3 straining modes described above), whereas nodes 2 and 4 only possess the 3 rigid-body DOFs. On the other hand, the integration points (for internal forces) are the $m^*_{gs} = 60$ points computed by the ECM (see Figure 17.b).

Having defined this (admittedly unorthodox) finite element, a typical coarse-scale analysis would proceed as any

[23] This uneven distribution of weights (also observed in Figure 5.c) is due to the "greedy" character of the employed selection algorithm (the ECM, explained in Appendix A)

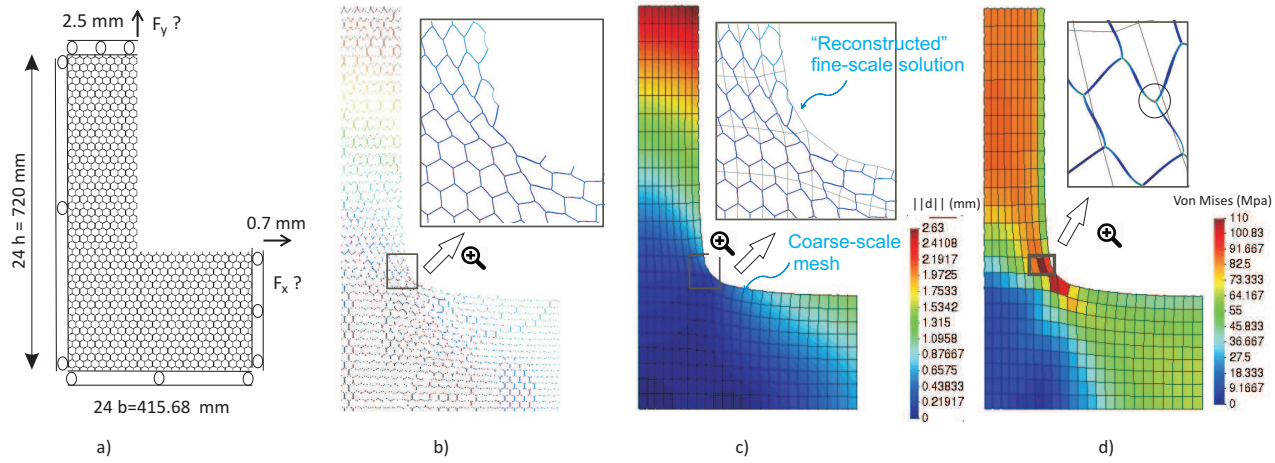

Figure 19: a) Geometry and boundary conditions employed for assessing the coarse-scale model. b) Deformed shape (×40) and contour plot of displacements (euclidean norm) computed by the standard finite element method (around 1 million standard quadrilateral elements). c) Idem to b), but computed by the coarse-scale model (351 18-DOFs quadrilateral elements). The enlarged view shows both the deformed coarse-scale and fine-scale meshes. d) (Coarse-scale) contour plot of maximum Von Mises stress (MPa), together with the fine-scale solution corresponding to the most critical coarse-scale element (enlarged view).

standard FE analysis. The only consideration to bear in mind is that the domain must be meshed with *quadratic* quadrilateral elements of the same height and width as the unit cell, solely keeping in the connectivity matrix the midside nodes—corner nodes are only used for post-processing purposes. We show in Figures 18.b and 18.c the fine-scale (FE) and coarse-scale solutions of the stretching test (first test in Figure 17.c). To facilitate the interpretation of results, the coarse-scale displacements are portrayed as if the structure were continuous —this is done by extrapolating the values of the midside nodes to the corner nodes. For comparison purposes, we also show the fine-scale, deformed mesh recovered from the coarse-scale solution. Visually, there are no discernible differences between this deformed mesh and the *exact* one computed by the FE method. In term of forces, the accuracy is equally notable: the resultant of forces in the x-direction predicted by the FE is $F_x^{fe} = 3.0398$ N, while the coarse-scale model gives $F_x^{rom} = 3.0396$ N (an error of just 0.006 %). The level of accuracy in the other 3 tests is less astonishing, yet equally satisfactory from a practical point of view: the shear test (y direction) requires $F_y^{fe} = 0.5307$ and $F_y^{rom} = 0.5309$ N (error 0.03 %); the stretching test (y direction) $F_y^{fe} = 0.777$ and $F_y^{rom} = 0.768$ N (error 1.22 %); and the shear test (x direction) $F_x^{fe} = 0.0994$ and $F_x^{rom} = 0.0981$ N (error 1.30 %).

To complete the assessment, we carry out the analysis of the L-shaped geometry of Figure 19.a under the indicated prescribed displacements. The deformed shapes and the displacement norm computed by the standard FE method and the coarse-scale model are displayed in Figures 19.b and 19.c. The former requires around 1 million standard 8-DOFs quadrilateral elements, while the latter contains only 351 quadrilateral elements of the type defined in the foregoing (18-DOFs per element). Despite this startling reduction in the number of elements, the deformed boundaries of both geometries are practically identical. The same degree of accuracy is observed in the critical corner region, displayed in magnified form for both cases, as well as in terms of resultant forces, summarized in Table 3.

	Fine-scale (FE) model	Coarse-scale model	**Error**	Homogenization (FE^2)	**Error**
$F_x(N)$	15.12	15.14	**0.13 %**	18.41	**21.76 %**
F_y (N)	11.96	11.99	**0.25 %**	14.36	**16.1 %**

Table 3: Resultant of horizontal and vertical forces in the L-shaped test of Figure 19.a. Values computed by the standard FE model (1 million 8-DOFs quadrilateral elements), the coarse-scale model (351 18-DOFs quadrilateral element, see Figure 18.a); and the first-order homogenization model (351 8-DOFs quadrilateral elements, using the homogenized elasticity matrix of Eq.(121)).

We also provide in this table the resultant forces computed by using the standard first-order homogenization (the FE^2

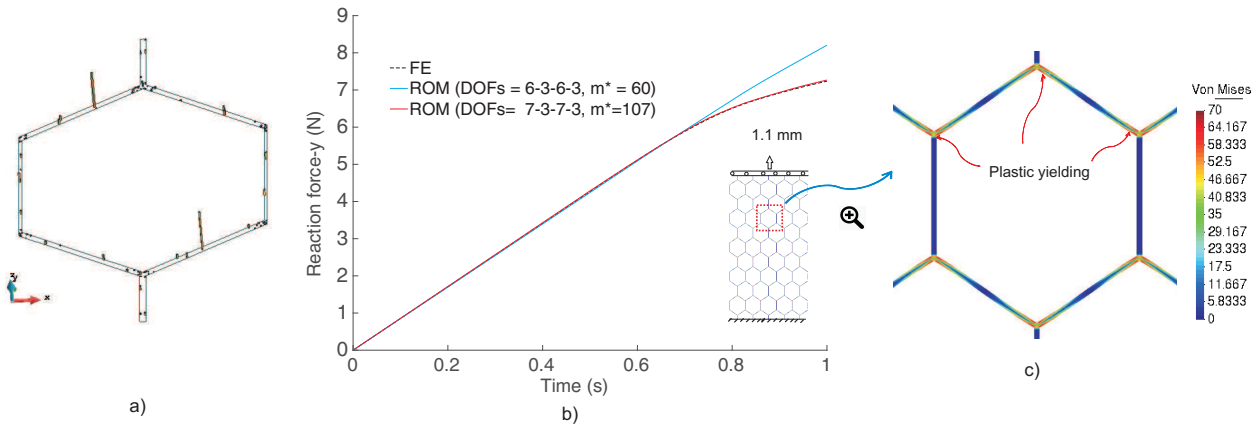

Figure 20: Nonlinear regime. a) Location of the $m^*_{gs} = 107$ of integration points computed by the ECM in this case. b) Comparison of the reaction forces in the nonlinear training test computed by the standard FE method and the elastic and inelastic coarse-scale elements ($m^*_{gs} = 60$ and 107 ECM points per element, respectively). c) and d) Enlarged view showing the contour plots of Von Mises stresses (in MPa)

method); the homogenized elasticity matrix[24] for this hexagonal unit cell is given by (in MPa)

$$\boldsymbol{C}^{homog} = \begin{bmatrix} 984.69 & 977.02 & 588.51 & 0.054 \\ 977.02 & 984.63 & 588.5 & 0.052 \\ 588.51 & 588.5 & 3809.4 & 0.032 \\ 0.054 & 0.052 & 0.032 & 3.81 \end{bmatrix} \tag{121}$$

We see that first-order homogenization grossly overpredicts both horizontal and vertical forces (22.76 % and 16.1 % of error); by contrast, our multiscale approach furnishes a remarkable accurate prediction (error below 0.25 % for both forces). These results emphasize one of the major advantages of the proposed multiscale method: its ability of *providing accurate results in scenarios in which the hypothesis of scale separation clearly does not hold*.

Lastly, Figure 19.d exemplifies how the proposed approach allows one to easily perform a *multiscale stress analysis*. Firstly, at the coarse scale, one can identify the critical regions by plotting the *maximum* Von Mises stress at each element (maximum among the $m^*_{gs} = 60$ ECM integration points). Then, by simple matrix multiplications (see Eq. 120), one can reconstruct the stresses at all the Gauss points of the underlying unit cell mesh (the fine-scale), and from this information, spot the areas with highest stresses (this is shown in the enlarged view of Figure 19.d).

9.4.2. Nonlinear regime

We carry out next a coarse-scale analyses in the *nonlinear regime*, by assuming that the material of the hexagonal cells obeys a rate-independent Von Mises elastoplastic model, endowed with a linear, isotropic hardening law (yield stress $\sigma_y = 60$ MPa, and hardening modulus $H = 700$ MPa). Our interest lies in detecting the onset of formation of *plastic hinges* (as described in Ref. [18]) at the intersection between the walls of the cell. To this end, we carry out the test depicted in Figure 20.b, in which the same 5-by-5 domain employed in the linear tests is stretched 1.1 *mm* in the y-direction. The constitutive differential equations are integrated in time (with $n_{steps} = 40$ time steps) using the classical, fully implicit backward-Euler scheme [38].

To derive a coarse-scale element that accounts for the nonlinear effects occurring in this test, we proceed as in Example 9.3, that is, by simply enriching the training data base with the solutions (a total of $9 \cdot 40 = 360$ snapshots) coming from this nonlinear problem. We then apply the operations specified in steps 4, 5 and 6 of the offline procedure of Box 8.1, using $\epsilon_\lambda = \epsilon_\sigma = 10^{-3}$ as truncation tolerances for the additional matrices, while keeping the remaining parameters to the values used in the elastic case. This gives $p = 17$, $r = 27$, $m^*_{gs} = 107$ and 7-3-7-3 coarse-scale DOFs. It is interesting to note that, in passing from the linear to the nonlinear regime, the kinematics of the coarse-scale element barely changes (there is only two additional DOFs), whereas the number of integration points computed by the ECM practically doubles (from 60 to 107). As can be seen in Figure 20.a, such points are mainly concentrated at the intersection between the walls of the cell, which is precisely where plastic yielding takes place (see Figure 20.c). To assess the effect of the number of ECM

[24] The homogenized elasticity matrix has been calculated following the standard procedure of running 3 FE simulations of a single unit cell under periodic boundary conditions (the 3rd row of the matrix corresponds to the stresses in the z-direction).

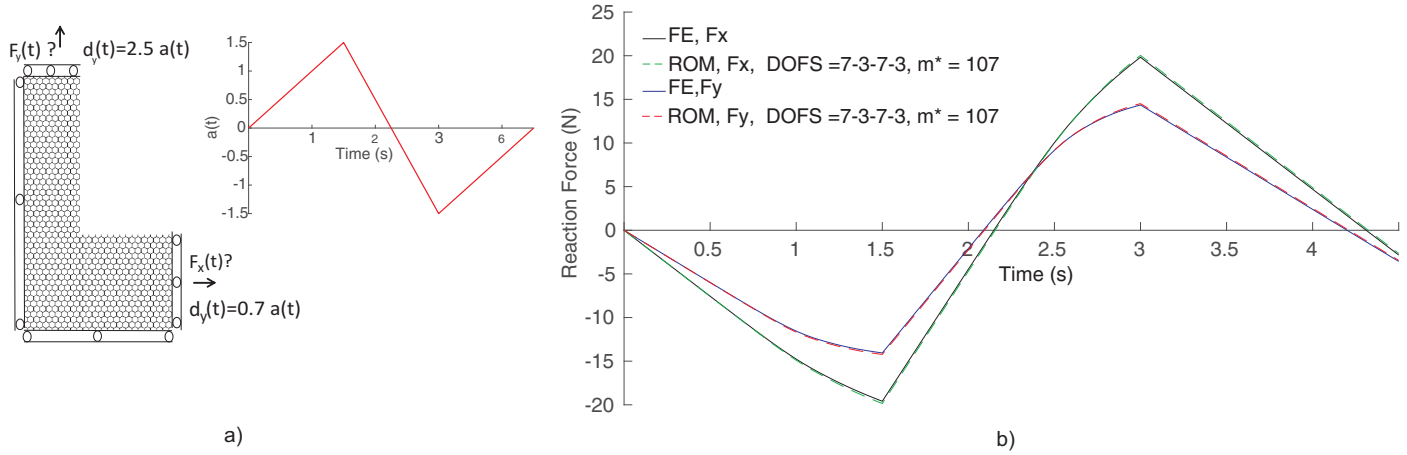

Figure 21: a) Geometry and boundary conditions employed for assessing the coarse-scale model in the elastoplastic regime. b) Resultants of vertical and horizontal forces versus time computed by the standard FE method and the coarse-scale model.

points in the accuracy of the predictions, we plot in Figure 20.b the graphs of the reaction force versus time computed by the FE model and the linear and nonlinear coarse-scale models ($m^*_{gs} = 60$ and $m^*_{gs} = 107$, respectively). Observe that the graph of the coarse-scale model with $m^*_{gs} = 107$ ECM points is practically identical to the full-order graph, the maximum discrepancy at the end being only 0.4 %. By contrast, the graph of the coarse-scale model with $m^*_{gs} = 60$ does capture the linear branch, but clearly fails to detect the onset of plastic yielding.

As customary, the evaluation of the predictive capabilities of this coarse-scale element is completed by examining a problem whose geometry and boundary conditions are markedly distinct to that used in training. We use for this purpose the L-shaped geometry described earlier (see Figure 19.a) under the cycle of horizontal and vertical prescribed displacement depicted in Figure 21.a (discretized in 300 time steps). The resultants of the vertical and horizontal reaction forces versus time computed by the standard FE method and the ROM are shown in Figure 21.b. We can see that the ROM is indeed able to accurately reproduce the whole loading/unloading cycle for both vertical and horizontal forces, with a maximum deviation at the peak forces of 1.3 %, a fact that demonstrates the applicability of the model to accurately predict the response for scenarios different from the training cases. As for the issue of computational efficiency, the FE simulation (around 1 million standard quadrilateral elements) took around 5 hours in an in-house, vectorized MATLAB code operating in a linux platform[25], while the reduced-order model (357 coarse-scale elements) required about 2 minutes for the same number of steps —the speedup[26] factor is, thus, around 300.

9.5. Composite shell

We conclude the assessment of the proposed methodology by deriving a coarse-scale model for the linear elastic behavior of a *cylindrical shell*, made up by tiling copies of the porous, composite cell of Figure 22.a. The cell is formed by two materials, one with Young's Modulus $E_m = 70000$ MPa and Poisson's ratio $\nu_m = 0.3$ (the matrix); and the other one with $E_i = 3E_m$ and $\nu_i = \nu_m$ (the four cylindrical inclusions). The thickness is $t = 0.24$ mm, and the radius of the midplane $r = 42.452$ mm. Likewise, the rotation of the interface boundary $\partial\bar{\Omega}_3$ with respect to $\partial\bar{\Omega}_1$ is given by the matrix

$$\boldsymbol{Q}_3 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{122}$$

where, as indicated in Figure 22.a, $\theta = 2\pi/80$ —thus, 80 unit cells are necessary to complete a revolution around the y-axis. The 3D geometry is discretized using $m = 4 \cdot 1431 = 5724$ (8-node) hexahedral elements, resulting in $n = 7680$ nodes[27].

[25] Intel(R) Core(TM) i7, with 8 processors at 2.80GHz

[26] This speedup factor is calculated without taking into account the offline cost of running the FE training analyses and constructing the bases, since such computations have to be performed just once. In the case at hand, the FE training process required around 3 minutes (220 seconds for the FE training tests, and 50 seconds for constructing the bases and determining the ECM points).

[27] A word concerning the meshing procedure is in order here. To ensure conformity between meshes of opposite boundary interfaces ($\partial\bar{\Omega}_1$ with $\partial\bar{\Omega}_3$, and $\partial\bar{\Omega}_2$ with $\partial\bar{\Omega}_4$), the discretization was done in an "unbent" configuration, with $\partial\bar{\Omega}_3$ parallel to the z-y plane; then, by applying a linearly varying rotation around the y-axis, the cell was transformed into the curved geometry shown in Figure 22.a.

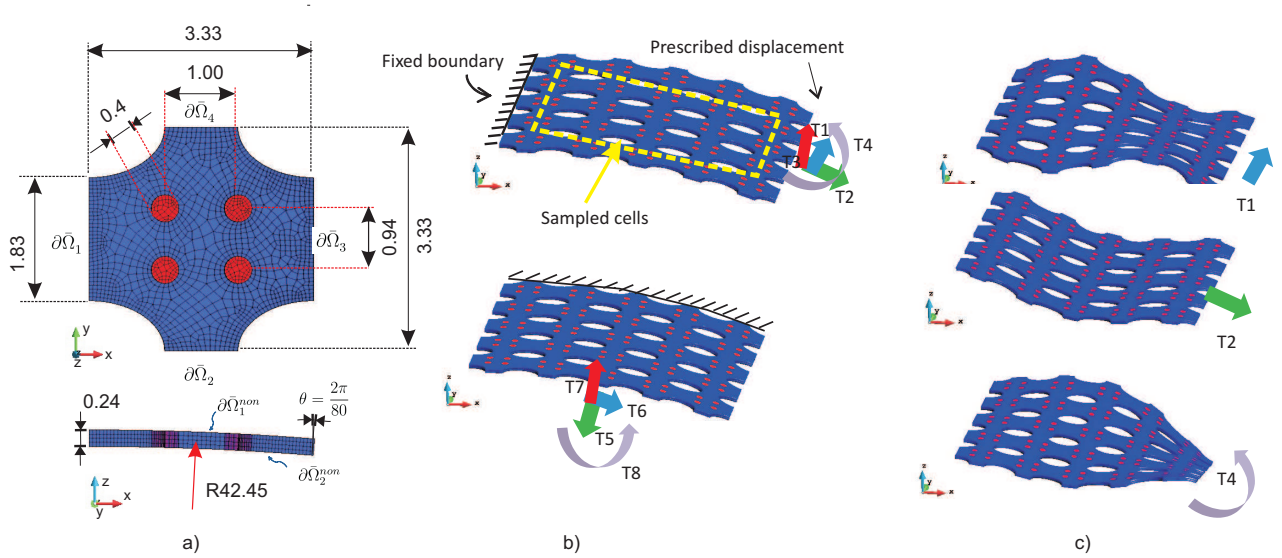

Figure 22: a) Geometry and FE discretization of the shell composite cell (top and front views). Dimensions are given in *mm*. b) 5-by-5 cell structure and boundary conditions used for "training" the coarse scale model. The unit cells whose displacement, reactions and stress solutions are included in the snapshot matrices are the 9 cells enclosed by the dashed line. c) Amplified FE deformed shapes corresponding to three (T1, T2 and T4) of the 8 training tests.

To train the coarse-scale model, we use the 5-by-5 cells structure and the boundary conditions depicted in Figure 22.b. For each of the two periodicity directions, the nodal displacements of one of the interface boundaries are set to zero, while the opposite surface is subjected to prescribed displacements (0.1 mm) in 3 mutually orthogonal directions (tests T1,T2,T3,T5,T6,T7), and torsional rotations (0.1 rad, tests T4 and T8). The solution in displacements, reactive forces between domains and stresses of the 3-by-3 inner cells —enclosed by the dashed line in Figure 22.b—- are stored in the corresponding snapshot matrices[28]. We then follow the offline procedure of Box 8.1, using truncation tolerances $\epsilon_\lambda = \epsilon_\sigma = \epsilon_{ecm} = 10^{-3}$ and $\epsilon_f = 10^{-4}$. This gives $p = 22$ straining and reactive modes (four of such modes are shown in Figures 23.a and 23.b), 7 interface modes for each interface, and $m^*_{gs} = 179$ ECM points. The resulting coarse-scale element, displayed in Figure 23.c, consists, thus, in a cylindrical shell quadrilateral featuring 4 midside nodes with 7 DOFs each (3 translations, 3 rotations and 1 straining DOF), and 179 ECM integration points. The reduction, with respect to the original 3D mesh, in number of DOFs and integration points is therefore equal to $3 \cdot 7680/28 \approx 823$ and $8 \cdot 5724/179 \approx 256$, respectively. Yet, the coarse-scale model still manages to reproduce the training test with acceptable degree of accuracy: in terms of resultant reaction forces, the error ranges from 0.10 % in test T6 to 3.28 % in test T2. The level of approximation can be better appreciated in Figure 24, where we display the deformed shape and Von Mises distribution of test T1, along with its coarse-scale counterparts (Figure 24.b is the deformed coarse-scale mesh, of 5-by-5 quadrilaterals, together with the cloud of ECM points used for integration, and Figure 24.c is the reconstructed fine-scale solution).

As usual, we next evaluate the predictive capabilities of the coarse-model in scenarios in which the input parameters are markedly different from the input parameters used in the training process. Two examples will be employed for this. The first one is depicted in Figure 25.a. It consists of a domain of 15 cells in both the axial and circumferential direction, clamped at the four edges, and subjected to a uniform pressure of 0.1 MPa, distributed over an area spanning 5-by-5 cells. Figure 25.b shows the deformed shape and contour plot of displacements computed by the FE method (featuring 1.3 million elements and 5.1 million DOFs). Figures 25.b, 25.c and 25.d, on the other hand, display the displacement field predicted by the coarse-scale model (225 elements of the type defined in Figure 23.c, and 3360 DOFs); Figure 25.b is the continuous 2D representation, while Figures 25.c and 25.d contain the reconstructed 3D field. The qualitative resemblance of the FE and ROM 3D fields (Figures 25.a and Figures 25.c) is notorious, the only subtle difference being that the area with highest displacements appears to be more dispersed in the FE case. In terms of maximum displacement, the FE yields 0.142 mm, while the ROM predicts 0.145 mm (only 2.1 % above). These small deviations are largely compensated by the enormous savings in computational time afforded by the coarse-model: assembling and solving the coarse-scale

[28]In this case, since the structure is curved, such quantities have to be rotated back to the local reference system of the unit cells

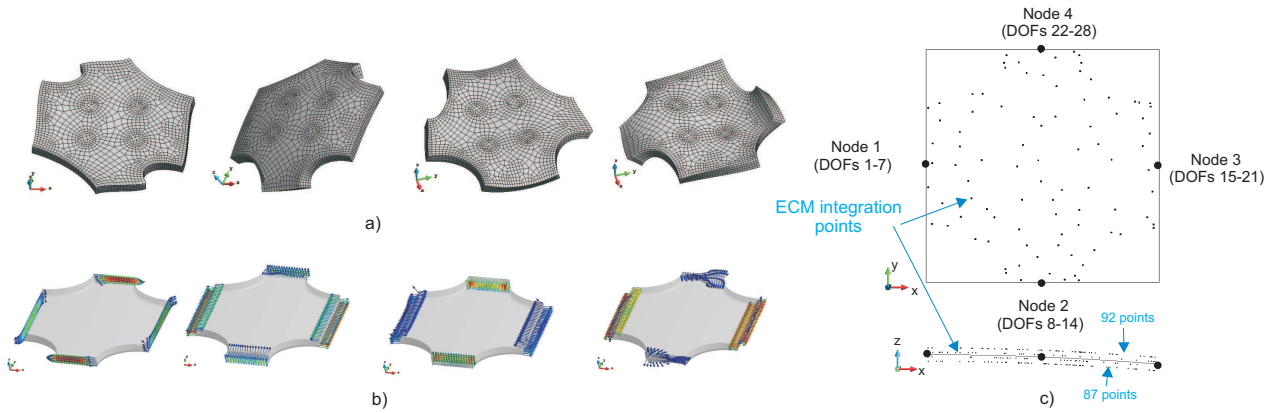

Figure 23: a) and b) Four of the $p = 22$ straining and reactive modes, respectively, obtained from the training set of Figure 22.b. c) Resulting coarse-scale element, featuring 4 nodes (with 7 DOFs per node) and $m^*_{gs} = 179$ ECM points.

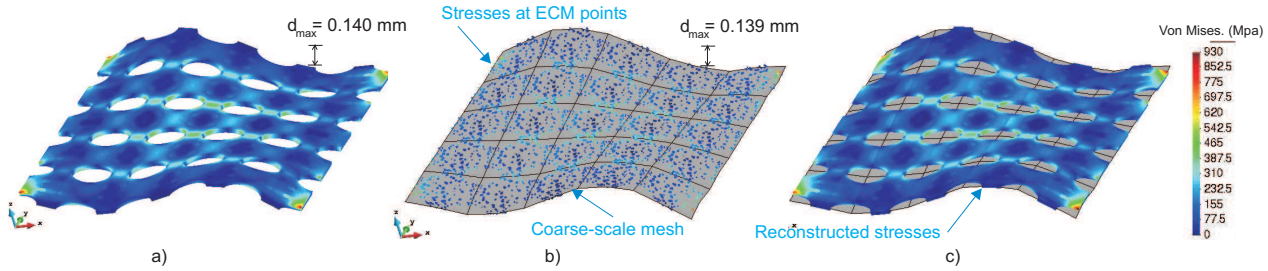

Figure 24: Training test T1. a) Deformed shape (×10) and Von Mises stress contour plot computed by the FE. b) 5-by-5 quadrilateral, coarse-scale deformed mesh (×10), along with the ECM integration points ($m^*_{gs} = 179$ for each coarse-scale element) . c) Deformed mesh and contour plot of the reconstructed Von Mises stresses (from the information of the ECM points).

equations take less than 1 second in the employed computer; by contrast, the FE requires more than 20 minutes to do the same operations —hence the speedup factor in this case is above 1200.

The second —and last— example used to test the performance of the shell coarse-scale element of Figure 23.c is described in Figure 26. The structure is a holed cylinder tube comprising 128 cells in the axial direction —and 80 in the circumferential direction. The hole is located midway between the ends of the cylinder, and it is made by removing a patch of 5-by-6 cells, as indicated in Figure 26.a. The cylinder is fixed at one end, and it is stretched by displacing a distance 0.1 mm the other end in the axial direction. The goal is to study the stress concentration caused by the presence of the hole. As done previously in the cellular solid problem (see Section 9.4.1), this stress analysis is conducted in a multiscale fashion. Firstly, we solve the system of coarse-scale equations and construct a 2D coarse-scale representation of the Von Mises equivalent stress (by plotting, for each element, the maximum Von Mises stress among the $m^*_{gs} = 179$ ECM points). The resulting contour plot is shown in Figure 26.b. The peak value is located, as one may expect, at one of the corners of the hole. Then, we reconstruct the 3D Von Mises stress field by the post-processing procedure of Section 8. This is done in Figure 26.c, where we show the 2D deformed geometry together with the reconstructed contour plot corresponding to the 500 cells with highest values in the plot of Figure 26.b. A closer inspection of the enlarged view of Figure 26.d allows us to finally reveal the actual fine-scale mechanism producing the peak stress observed at the coarse-scale —it is caused by stress concentration at the pore located at the corner.

To rigorously complete the assessment, it would be necessary to compare these results with a direct simulation using the standard FE —as it has been done with the rest of the problems discussed in previous sections. Unfortunately, in this case, the sheer number of elements of the resulting FE mesh[29] (more than 58.5 million) makes this assessment prohibitively costly using the employed computer and FE software (which, recall, is an in-house Matlab code). By contrast, the coarse-scale simulation takes *less than 5 seconds* in the same computer, a fact that emphasizes the tremendous gains in performance afforded by the proposed multiscale scheme. Concerning the accuracy issue, a glance at both Figures 26.c and 26.d indicate that both the reconstructed displacements and stress fields are fairly continuous; this can be taken as

[29]There are $M = 128 \cdot 80 - 6 \cdot 5 = 10212$ cells, and $m = 5724$ hexahedral elements per cell.

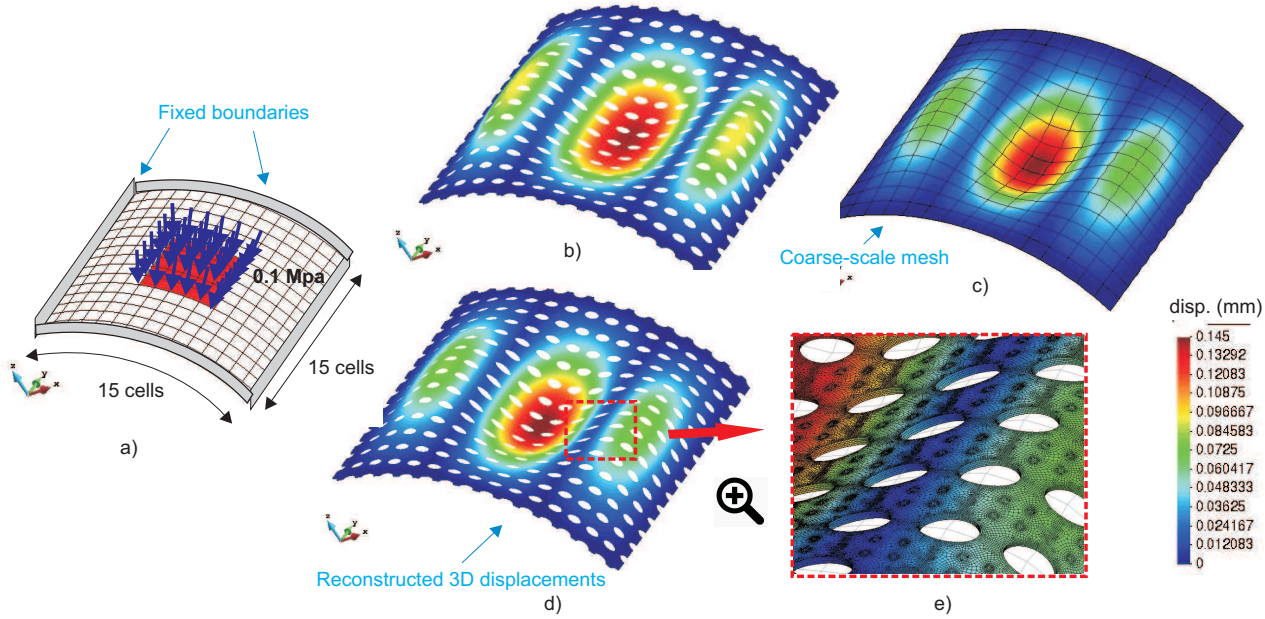

Figure 25: a) 15-by-15 cell domain and boundary conditions used to test the coarse-scale element defined in Figure 25.c. b) Deformed shape (×40) and contour plot of displacements (euclidean norm) computed by the standard FE (1.3 million elements and 5.1 million DOFs). c) Idem to b), but computed with the coarse-scale model (15 × 15 =225 elements and 3360 DOFs). d) and e) Reconstructed fine-scale displacement solution

an evidence that the results may be trusted, because, as we saw in dealing with the local effects problem of Figure 13.c), inaccuracies of the coarse-scale model typically manifest themselves in ostensible jumps in the reconstructed fine-scale fields .

10. Concluding remarks

At the heart of this work lies the conviction that any simplified structural model can be automatically derived from a more complex one by combining domain decomposition and dimensionality reduction. This conviction is grounded in an inescapable truth: dimensionality reduction algorithms are far better than humans in uncovering dominant patterns from data, and therefore, they can be harnessed to identify, from solutions generated with the complex model, the force/deformation low-dimensional parameterizations that characterize simplified structural models. The task of the modeler in this new paradigm is, thus, limited to that of judiciously designing the computational tests from which the "training" data are obtained.

We have chosen as vehicle for exposing this idea the two-scale modeling of periodic structures with unit cells possessing disjoint interface boundaries —and operating in the quasi-static, small deformations regime. Nevertheless, the employed ROM partitioning framework is rather general, and extensions to continuous interface boundaries, as well as geometrically distinct subdomains, should be feasible with minor modifications. To consider continuous interface boundaries, it would be necessary to include constraint equations accounting for the intersection entities, which in the proposed framework would translate into simply adding *vertex nodes* to the coarse-scale elements (we saw in the examples of Figure 18 and Figure 23 that, in 2D periodicity, the coarse-scale elements are quadrilateral with solely midside nodes). Departures from the periodicity condition, on the other hand, would require a redefinition of the training procedure (specific training tests should be devised for each family of geometrically related subdomains), and a revision of Algorithms 3 and 5 for extracting the interface modes from the domain displacement modes. The remaining elements of the approach would remain unchanged.

Extensions to dynamic and/or large deformation regimes, on the other hand, will offer varying degrees of difficulty depending on the situation. The adaption to scenarios with small strains but large rotations of the subdomain (for beam- and shell-like structures) should not pose significant conceptual challenges —albeit it will surely prove algorithmically intricate. The training process would be the same as in the small deformation regime, and one would have just to replace the additive decomposition with infinitesimal rotations of Eq.(14) by Eq.(32) by their finite rotation counterparts, as it is done in co-rotational finite element approaches (cf. Ref. [14]). Moderate strains and incorporation of inertial effects

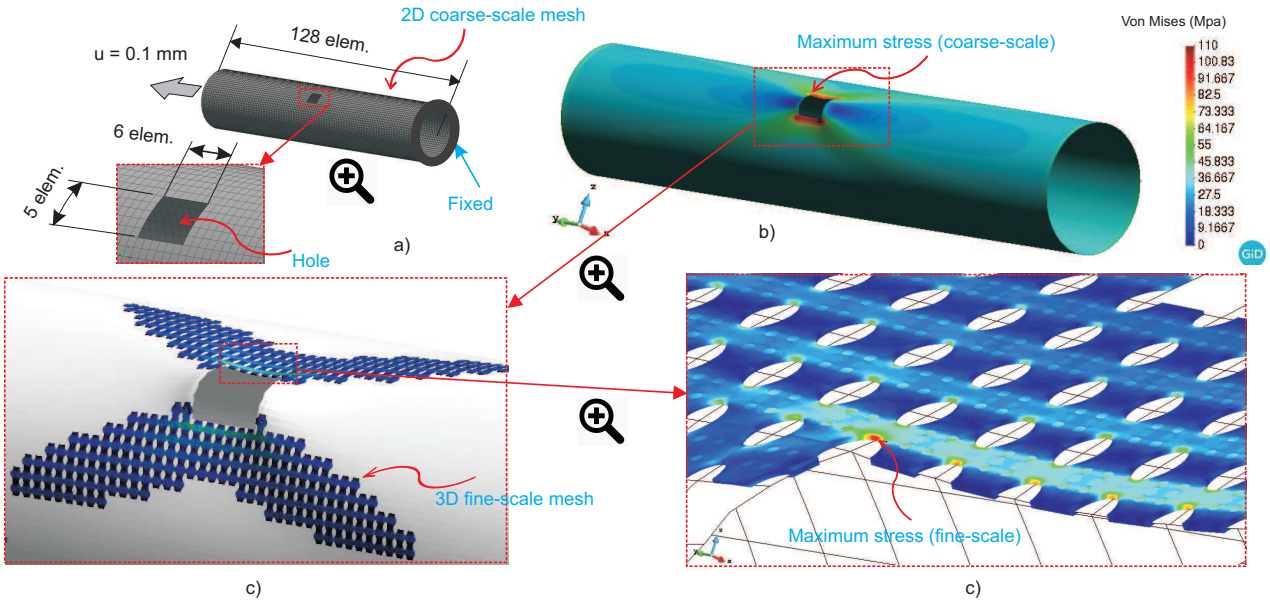

Figure 26: Tensile test on a holed tube formed by 128 × 80 cells (the hole occupies an area of 6×5 cells). a) Coarse-scale mesh ($M = 128 \cdot 80 - 6 \cdot 5 = 10212$ elements) and employed boundary conditions. b) Deformed shape (×40) and contour plot of coarse-scale Von Mises stresses. This contour plot is made by displaying the maximum stress among the $m^*_{gs} = 179$ ECM points of each coarse-scale element. c) and d) Fine-scale mesh reconstructed at the 500 elements with highest maximum stresses.

should not entail unsurmountable difficulties either—in fact, the employed partitioning framework was originally devised for dynamic problems [33]. By contrast, severe obstacles might be encountered when moving to cases in which the unit cells undergo relatively large deformations (especially if there is buckling), since in such cases the premise that the straining part of the domain displacements admits a low-dimensional *linear* expansion ceases to hold. Such a roadblock may be partially overcome by replacing the linear approximations by *nonlinear* mappings, which may be determined in turn by some type of *manifold learning algorithm*, as done, for instance, in Ref. [29] for one-scale dynamic problems.

Of paramount importance for the consistency of the overall approach has been to adopt a partitioning framework equipped with fictitious interfaces, and to assign the role of coarse-scale DOFs to the amplitude of the modes of such interfaces. Furthermore, this choice has enabled us to establish a direct link with existing finite element beam theories. Indeed, we have seen in the rectangular beam example of Section 9.1 that our method leads to a 2-node element with 6 DOFs, as in a standard Timoshenko's element. The difference is that, in our approach, *it is the fictitious interfaces that remain plane*, not the faces of the slices, which are free to warp due to torsion and shear. This is why our 2-node beam element is totally *consistent* with the underlying 3D theory —as opposed to Timoshenko's or other analytical theories. Besides, the proposed methodology is able to automatically detect from the provided data when it is necessary to enrich the kinematics with more DOFs. This has been fittingly illustrated by the I-shaped cross-section beam problem, in which Algorithm 3 has "discovered" the fact that the kinematics of thin-walled open cross-section cannot be accurately represented with just translation and rotations, but require an additional warping DOF (see Figure 10). Another pivotal ingredient of the proposed approach has been the *hyperreduction* of the internal forces of the subdomains via the Empirical Cubature Method. In this regard, we have observed in the hexagonal cell example of Section of 9.3 that the number of ECM points increases considerably when passing from the linear to the nonlinear regime, even if the kinematics remains practically the same. Therefore, it would be worthy to explore whether it is possible to further improve the ECM so that it returns less integration points for the same accuracy level. In principle, there is room for improvement, because the ECM, as well are other similar algorithms either based on nonnegative least-squares (such as the method advocated by Farhat et co-workers [12]), or on linear programming (such as the Empirical Quadrature Method of Patera et co-workers [39]), are not optimal, for they do not exploit the topology of the function they intend to integrate.

Acknowledgements

The research leading to these results has received funding from ANACONDA BIOMED S.L. and the Spanish Ministry of Science, Innovation and Universities via grant RTC-2017-6749-1. The author also acknowledges financial support from

the Spanish Ministry of Economy and Competitiveness, through the "Severo Ochoa Programme for Centres of Excellence in R&D" (CEX2018-000797-S).

A. Empirical Cubature Method

A key offline step (see Box 8.1, item 5) is the selection among the Gauss points of the unit cell mesh of a reduced set of points able to accurately integrate (when equipped with especially tailored weights) the internal virtual work, see Eq. (46). Here this task is carried out using an algorithmically improved version —in a sense that will be defined later— of the *Empirical Cubature Method* (ECM), proposed by the author in Ref. [21]. As explained therein, the ECM operates on a snapshot matrix $\boldsymbol{A^f}$ that depends on the value of the integrand at all the m_{gs} Gauss points and for all the P configurations . In this case, the integrand is the reduced internal work per unit volume $\boldsymbol{B}^{*T}(\boldsymbol{x}_g)\boldsymbol{\sigma}(\boldsymbol{x}_g,\boldsymbol{\mu})$. Since in the small strain regime $\boldsymbol{B}^{*T}(\boldsymbol{x}^g) \in \mathbb{R}^{p\times s}$ does not depend on the input parameters $\boldsymbol{\mu}$, rather than storing the internal forces at all Gauss points and for all configurations (which requires a matrix of size $m_{gs}\times P\cdot p$), it proves more efficient to solely store the stress solutions in a snapshot matrix $\mathbf{A^\sigma} \in \mathbb{R}^{s\,m_{gs}\times P}$ (see Eq. 87). Then, by applying the truncated SVD (with a tolerance $\epsilon_\sigma \sim \epsilon_\lambda$), one can obtain a basis matrix for the stresses:

$$[\boldsymbol{\Lambda},\bullet,\bullet] = \texttt{SVD}(\mathbf{A^\sigma},\epsilon_\sigma), \tag{123}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{s\cdot m_{gs}\times r}$, r being the number of stress modes (which should be of the same order as the number of straining and self-equilibrating modes p). Having the stress basis matrix at one's disposal, the desired matrix of internal forces is determined as

$$\boldsymbol{A^f} := \begin{bmatrix}\boldsymbol{\mathcal{F}}_1^1 & \cdots & \boldsymbol{\mathcal{F}}_1^p & \boldsymbol{\mathcal{F}}_2^1 & \cdots & \boldsymbol{\mathcal{F}}_2^p & \cdots & \boldsymbol{\mathcal{F}}_r^1 & \cdots & \boldsymbol{\mathcal{F}}_r^p\end{bmatrix}_{m_{gs}\times p\cdot r} \tag{124}$$

where the expression for $\boldsymbol{\mathcal{F}}_j^I \in \mathbb{R}^{m_{gs}}$ reads

$$\boldsymbol{\mathcal{F}}_j^I := \begin{bmatrix}\sqrt{W_1}\boldsymbol{B}_I^{*T}(\boldsymbol{x}_1)\boldsymbol{\Lambda}(\boldsymbol{b}_1,j)\\ \sqrt{W_2}\boldsymbol{B}_I^{*T}(\boldsymbol{x}_2)\boldsymbol{\Lambda}(\boldsymbol{b}_2,j)\\ \vdots \\ \sqrt{W_{m_{gs}}}\boldsymbol{B}_I^{*T}(\boldsymbol{x}_{m_{gs}})\boldsymbol{\Lambda}(\boldsymbol{b}_{m_{gs}},j)\end{bmatrix} \tag{125}$$

(here $\boldsymbol{b}_g = \{s\,(g-1)+1, s\,(g-1)+2\ldots s\,g\}$). Next we apply again the truncated SVD on $\boldsymbol{A^f}$ to eliminate redundancies and achieve an orthogonal basis matrix for the internal forces

$$[\boldsymbol{\Upsilon},\bullet,\bullet] = \texttt{SVD}(\boldsymbol{A^f},\epsilon_f). \tag{126}$$

Having at hand matrix $\boldsymbol{\Upsilon} \in \mathbb{R}^{m_{gs}\times q}$, the error incurred in evaluating the internal forces for any training configuration using the subset of integration points with indexes $\mathbf{Z} \subset \{1,2\ldots M\}$) and associated positive weights $\boldsymbol{\alpha} \in \mathbb{R}^{+m^*_{gs}}$ is given by[30]

$$e(\boldsymbol{\alpha},\mathbf{Z}) = \|\boldsymbol{\Upsilon}^T\sqrt{\boldsymbol{W}} - \boldsymbol{\Upsilon}^T(:,\mathbf{Z})\boldsymbol{\alpha}\|. \tag{127}$$

Given a tolerance $0 \le \epsilon_{ecm} \le 1$, we seek the smallest set of points $\mathbf{Z}$ and associated positive weights $\boldsymbol{\alpha}$ such that

$$\|\boldsymbol{G}\sqrt{\boldsymbol{W}} - \boldsymbol{G}(:,\mathbf{Z})\boldsymbol{\alpha}\| \le \epsilon_{ecm}\|\boldsymbol{G}\sqrt{\boldsymbol{W}}\| \tag{128}$$

where $\boldsymbol{G} = \boldsymbol{\Upsilon}^T$ ($\boldsymbol{G}\sqrt{\boldsymbol{W}}$ and $\boldsymbol{G}(:,\mathbf{Z})\boldsymbol{\alpha}$ represent the vectors of exact and approximate integrals, respectively). To solve this minimization problem, we use the function:

$$[\boldsymbol{Z},\boldsymbol{\omega}] = \text{ECM}(\boldsymbol{G},\boldsymbol{W},\epsilon_{ecm}) \tag{129}$$

described in Algorithm 7. As pointed out previously, this (greedy) method is an algorithmically improved version of the one put forward by the author in Ref [21]. The improvement goes as follows: at each iteration, the algorithm has to cope with a least-squares problem to obtain the corresponding weights: $\boldsymbol{\alpha} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$, where $\boldsymbol{A} = [\boldsymbol{G}(:,\boldsymbol{z}),\boldsymbol{G}(:,i)]$ is the matrix of coefficients formed by incorporating the incoming column $\boldsymbol{G}(:,i)$. In principle, this takes $\mathcal{O}(qm^2)$ floating-points

[30] If $\sqrt{\boldsymbol{W}}^T\boldsymbol{\Upsilon} = \mathbf{0}$, the problem becomes ill-posed (it admits the trivial solution $\boldsymbol{\alpha} = \mathbf{0}$). To eliminate the issue, it suffices to *expand* $\boldsymbol{\Upsilon}$ as $\boldsymbol{\Upsilon} \leftarrow [\boldsymbol{\Upsilon},\sqrt{\boldsymbol{W}}]$ (consult Ref. [21] for further details).

Algorithm 7: Empirical Cubature Method (enhanced version of the algorithm in Ref. [21])

Function $[\boldsymbol{z}, \boldsymbol{\omega}]$ =ECM($\boldsymbol{G}$,$\boldsymbol{W}$,TOL)

Data: $\boldsymbol{G} \in \mathbb{R}^{p\times M}$, where $\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{I}$; $\boldsymbol{W} \in \mathbb{R}^M$, $W_i > 0$; tolerance $0 \leq TOL \leq 1$

Result: $\boldsymbol{z} \subset \{1, 2 \dots M\}$, $\boldsymbol{\omega} > \boldsymbol{0} \in \mathbb{R}^m$, such that $\|\boldsymbol{G}(:, \boldsymbol{z})\boldsymbol{\alpha}\| \leq TOL\|\boldsymbol{G}\sqrt{\boldsymbol{W}}\|$, where $\omega_i = \sqrt{\boldsymbol{W}(\boldsymbol{z}_i)}\alpha_i$

$\boldsymbol{z} \leftarrow \emptyset$; $\boldsymbol{y} \leftarrow \{1, 2 \dots M\}$; $\boldsymbol{b} \leftarrow \boldsymbol{G}\sqrt{\boldsymbol{W}}$; $\boldsymbol{r} \leftarrow \boldsymbol{b}$; $\boldsymbol{\alpha} \leftarrow \emptyset$; $\boldsymbol{H} \leftarrow \emptyset$ // Initializations

$\boldsymbol{y} \leftarrow \boldsymbol{y} \setminus \boldsymbol{h}$, where $\boldsymbol{h} = [h_1, h_2 ...]$ such that $\|\boldsymbol{G}(:, h_i)\| \leq \epsilon$ // Remove points whose associated values in $\boldsymbol{G}$ are lower than a given tolerance ($\epsilon \sim 10^{-6}$)

while $\|\boldsymbol{r}\|/\|\boldsymbol{b}\| > TOL$ ***AND*** length($\boldsymbol{z}$)$< p$ ***AND*** length ($\boldsymbol{y}$)> 0 **do**

  $i = \arg\max_{i\in\boldsymbol{y}} \boldsymbol{g}_{\boldsymbol{y}}^T\boldsymbol{r}$, where $\boldsymbol{g}_j = \boldsymbol{G}(:, j)/\|\boldsymbol{G}(:, j)\|$ // Select the column most ''positively'' parallel to the residual $\boldsymbol{r}$

  if $\boldsymbol{z} = \emptyset$ **then**

    $\boldsymbol{H} \leftarrow (\boldsymbol{G}(:, i)^T\boldsymbol{G}(:, i))^{-1}$ // Inverse Hermittian matrix (first iteration)

    $\boldsymbol{\alpha} \leftarrow \boldsymbol{H}\boldsymbol{G}(:, i)^T\boldsymbol{b}$ // Weights computed through least-squares

  else

    $[\boldsymbol{\alpha}, \boldsymbol{H}] \leftarrow$ LSTONER($\boldsymbol{\alpha}, \boldsymbol{H}, \boldsymbol{G}(:, \boldsymbol{z}), \boldsymbol{G}(:, i), \boldsymbol{b}$) // Least-squares via one-rank update, see Algorithm 8.

  end

  $\boldsymbol{z} \leftarrow \boldsymbol{z} \cup i$; $\boldsymbol{y} \leftarrow \boldsymbol{y} \setminus i$; // Move index i from $\boldsymbol{y}$ to $\boldsymbol{z}$

  $\boldsymbol{n} \leftarrow$ Indexes such that $\boldsymbol{\alpha}_{\boldsymbol{n}} < 0$ // Identify negative weights

  if $\boldsymbol{n} \neq \emptyset$ **then**

    $\boldsymbol{y} \leftarrow \boldsymbol{y} \cup \boldsymbol{z}(\boldsymbol{n})$; $\boldsymbol{z} \leftarrow \boldsymbol{z} \setminus \boldsymbol{z}(\boldsymbol{n})$; // Remove indexes negative weights

    $\boldsymbol{H} \leftarrow$ UPHERM($\boldsymbol{H}, \boldsymbol{n}$) // Update inverse Hermitian Matrix (via recursive, one-rank operations, see Algorithm 9)

    $\boldsymbol{\alpha} = \boldsymbol{H}\boldsymbol{G}(:, \boldsymbol{z})^T\boldsymbol{b}$ // Recalculate weights

  end

  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \boldsymbol{G}(., \boldsymbol{z})\boldsymbol{\alpha}$ // Update the residual

end

$\omega_g \leftarrow \alpha_g\sqrt{W(z_g)}$, $g = 1, 2 \dots$ length($\boldsymbol{z}$)

Algorithm 8: Least-squares via one-rank update

Function $[\boldsymbol{\alpha}_{new}, \boldsymbol{H}_{new}]$ =LSTONER($\boldsymbol{\alpha}, \boldsymbol{H}, \boldsymbol{A}, \boldsymbol{a}, \boldsymbol{b}$)

Data: $\boldsymbol{A} \in \mathbb{R}^{p\times m}$, $\boldsymbol{a} \in \mathbb{R}^p$, $\boldsymbol{H} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}$, $\boldsymbol{b} \in \mathbb{R}^p$, $\boldsymbol{\alpha} = \boldsymbol{H}\boldsymbol{A}^T\boldsymbol{b}$,

Result: $\boldsymbol{H}_{new} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}$ and $\boldsymbol{\alpha}_{new} = \boldsymbol{H}_{new}\boldsymbol{B}^T\boldsymbol{b}$, where $\boldsymbol{B} = \begin{bmatrix}\boldsymbol{A} & \boldsymbol{a}\end{bmatrix}$, via one-rank update.

$\boldsymbol{c} = \boldsymbol{A}^T\boldsymbol{a}$; $\boldsymbol{d} = \boldsymbol{H}\boldsymbol{c}$; $s = \|\boldsymbol{a}\|^2 - \boldsymbol{c}^T\boldsymbol{d}$

$\boldsymbol{H}_{new} = \begin{bmatrix}\boldsymbol{H} + \frac{1}{s}\boldsymbol{d}\boldsymbol{d}^T & -\frac{1}{s}\boldsymbol{d} \\ -\frac{1}{s}\boldsymbol{d}^T & \frac{1}{s}\end{bmatrix}$

$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\alpha}$; $v = (\boldsymbol{a}^T\boldsymbol{r})/s$

$\boldsymbol{\alpha}_{new} = \begin{bmatrix}\boldsymbol{\alpha} - v\boldsymbol{d} \\ v\end{bmatrix}$

operations (FLOPs), which implies that, at each iteration, the cost increases by a factor $\mathcal{O}(m^2)$. This constitutes a serious bottleneck for high q, especially when m approaches q (the exact solution, for $\epsilon_{ecm} \approx 0$, is achieved for $m = q$, that is, when $\boldsymbol{A}$ becomes square). To amend this, we have modified the original algorithm in Ref. [21]. Now the the least-squares problem is solved using function LSTONER, described in Alg. 8. This function performs a rank-one update of $(\boldsymbol{A}^T\boldsymbol{A})^{-1}$ using the formula[31] for the inverse of a symmetric 2-by-2 block matrix (cf. Refs. [3, 20]), which requires only $\mathcal{O}(qm)$ FLOPs. The same strategy is used for updating the inverse of the Hermitian matrix upon removal of the columns with

[31] Alternatively, when the least-squares problem is solved using the QR decomposition, one can update the QR decomposition of the coefficient matrix. This is the procedure followed by Lawson and Hanson in their pioneering Nonnegative Least-Squares (NNLS) algorithm [27]. In this respect, Chapman et al. [6] has recently proposed a parallel updatable QR factorizer, which has been used in the context of the energy-conserving sampling and weighting (ECSW) procedure [12, 13].

Algorithm 9: Update of the inverse of an Hermitian matrix when several columns are removed

Function $[\boldsymbol{H}_{new}]$ =UPHERM$(\boldsymbol{H}, \boldsymbol{n})$
  Data: $\boldsymbol{H} \in \mathbb{R}^{m\times m}$, where $\boldsymbol{H}$ is of the form $\boldsymbol{H} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}$, $\boldsymbol{A} \in \mathbb{R}^{p\times m}$; $\quad \boldsymbol{n} \subset \{1, 2 \dots m\}$
  Result: $\boldsymbol{H}_{new} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}$, where $\boldsymbol{B} = \boldsymbol{A}(:, \boldsymbol{c})$, $\boldsymbol{c} = \{1, 2 \dots m\} \setminus \boldsymbol{n}$
  $\boldsymbol{n} \leftarrow$ sort$(\boldsymbol{n})$ // Sort the indexes in ascending order
  $\boldsymbol{H}_{new} \leftarrow \boldsymbol{H}$
  for $i = 1$ *to* length$(\boldsymbol{n})$ **do**
    $j = \boldsymbol{n}(i) - i + 1$
    $\boldsymbol{H}_{new} \leftarrow$ UPHERMone$(\boldsymbol{H}_{new}, j)$ // Hermitian matrix inverse when 1 column is removed (Alg. 10)
  end

Algorithm 10: Update of the inverse of an Hermitian matrix when one column is removed

Function $[\boldsymbol{C}_{new}]$ =UPHERMone$(\boldsymbol{C}, j)$ // From https://emtiyaz.github.io/Writings/OneColInv.pdf
  Data: $\boldsymbol{C} \in \mathbb{R}^{m\times m}$, where $\boldsymbol{C}$ is of the form $\boldsymbol{C} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}$, $\boldsymbol{A} \in \mathbb{R}^{p\times m}$; $\quad j \in \{1, 2 \dots m\}$
  Result: $\boldsymbol{C}_{new} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}$, where $\boldsymbol{B} = \boldsymbol{A}(:, \boldsymbol{c})$, $\boldsymbol{c} = \{1, 2 \dots m\} \setminus j$
  $\boldsymbol{D} \leftarrow \boldsymbol{C}$; $\quad \boldsymbol{r} \leftarrow \{1, 2 \dots m-1\}$
  if $j < m$ **then**
    $\boldsymbol{a} \leftarrow \{1, 2 \dots j-1\}$; $\quad \boldsymbol{b} \leftarrow \{j+1 \dots m\}$
    $\boldsymbol{D} \leftarrow \big[\boldsymbol{C}(:, \boldsymbol{a}) \quad \boldsymbol{C}(:, \boldsymbol{b}) \quad \boldsymbol{C}(:, j)\big]$; $\quad \boldsymbol{D} \leftarrow \big[\boldsymbol{D}(\boldsymbol{a}, :) \quad \boldsymbol{D}(\boldsymbol{b}, :) \quad \boldsymbol{D}(j, :)\big]$
  end
  $\boldsymbol{C}_{new} = \boldsymbol{D}(\boldsymbol{r}, \boldsymbol{r}) - \dfrac{\boldsymbol{D}(\boldsymbol{r}, m)\boldsymbol{D}(m, \boldsymbol{r})}{\boldsymbol{D}(m, m)}$

associated negative weights, see line 16 of Algorithm 7.

B. Coarse-scale body forces

The goal of this Section is to show how the expression for the coarse-scale body forces $\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}}$:

$$\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}} = \boldsymbol{\mathcal{F}}^e_{\boldsymbol{se}} + \boldsymbol{\mathcal{F}}^e_{\boldsymbol{eq}} \tag{130}$$

(see Eq. 67) can be cast in a reduced-order format, without the need of having at one's disposal the exact nodal expression for the FE nodal external forces $\boldsymbol{F}^e_{\boldsymbol{ext}}$.

B.1. Body forces

The vector of FE nodal body forces may be expressed as

$$\boldsymbol{F}^e_{\boldsymbol{b}} = \sum_{g=1}^{m_{gs}} \boldsymbol{N}^T(\boldsymbol{x}_g) W_g \boldsymbol{b}^e(\boldsymbol{x}_g), \tag{131}$$

where $\boldsymbol{N}(\boldsymbol{x}_g)$ is the shape function (in its global, sparse format) associated to the Gauss point located at $\boldsymbol{x}_g$ and $\boldsymbol{b}^e(\boldsymbol{x}_g)$ the corresponding body force per unit volume. If self-weight (gravity) is the only body force, then we may write Eq.(131) as

$$\boldsymbol{F}^e_{\boldsymbol{b}} = \sum_{k=1}^{n_{mat}} \boldsymbol{J}^k_{\boldsymbol{b}} \rho^k \hat{\boldsymbol{Q}}^{e^T} \boldsymbol{g}, \tag{132}$$

where

$$\boldsymbol{J}^k_{\boldsymbol{b}} := \sum_{g=1}^{m_{gs}(k)} \boldsymbol{N}^T(\boldsymbol{x}_{g(k)}) W_{g(k)}. \tag{133}$$

Here n_{mat} is the number of distinct materials, ρ_k designates the density of the $k-th$ material and $\boldsymbol{g} \in \mathbb{R}^{n_{sd}}$ is the vector of acceleration of gravity (in global coordinates).

B.2. Non-interface traction forces

As for non-interface tractions, suppose that the non-interface boundary $\partial\bar{\Omega}_{non}$ is subdivided into b_{non} portions. In such a case, the expression for the FE nodal contribution can be written as

$$\boldsymbol{F}^e_{\boldsymbol{tr}} = \sum_{k=1}^{b_{non}} \sum_{g=1}^{\bar{m}_{gs}(k)} \bar{\boldsymbol{N}}^T(\bar{\boldsymbol{x}}_{g(k)}) \bar{W}(\bar{\boldsymbol{x}}_{g(k)}) \boldsymbol{t}^e(\bar{\boldsymbol{x}}_{g(k)}), \tag{134}$$

where $\bar{\boldsymbol{N}}^T(\bar{\boldsymbol{x}}_{g(k)})$ stands for the matrix of boundary shape functions at the (boundary) integration point $\bar{\boldsymbol{x}}_{g(k)}$, $\bar{W}(\bar{\boldsymbol{x}}_{g(k)})$ is the corresponding volumetric weight and $\boldsymbol{t}^e(\bar{\boldsymbol{x}}_{g(k)}) \in \mathbb{R}^{n_{sd}}$ the traction vector in the domain reference axes. In the case of curved boundaries, the traction vector is often provided in the coordinate axes intrinsic to the surface: $\boldsymbol{t}^e(\bar{\boldsymbol{x}}_{g(k)}) = \boldsymbol{Q'}(\bar{\boldsymbol{x}}_{g(k)}) \boldsymbol{t'}^e(\bar{\boldsymbol{x}}_{g(k)})$ (here $\boldsymbol{Q'}(\bar{\boldsymbol{x}}_{g(k)})$ is the local rotation matrix at the boundary integration point $\bar{\boldsymbol{x}}_{g(k)}$), and Eq.(134) becomes expressible as

$$\boldsymbol{F}^e_{\boldsymbol{tr}} = \sum_{k=1}^{b_{non}} \boldsymbol{J}^k_{\boldsymbol{tr}} \boldsymbol{t'}^{e(k)}. \tag{135}$$

where

$$\boldsymbol{J}^k_{\boldsymbol{tr}} := \sum_{g=1}^{\bar{m}_{gs}(k)} \bar{\boldsymbol{N}}^T(\bar{\boldsymbol{x}}_{g(k)}) \bar{W}(\bar{\boldsymbol{x}}_{g(k)}) \boldsymbol{Q'}(\bar{\boldsymbol{x}}_{g(k)}) \tag{136}$$

In writing Eq.(135) we have assumed, without loss of generality, that the traction vector $\boldsymbol{t'}^{e(k)}$ is uniform over the non-interface boundary $\partial\bar{\Omega}^k_{non}$.

B.3. Final expressions

Next we determine the self-equilibrating and resultant components of each of the n_{sd} columns of $\boldsymbol{J}^k_{\boldsymbol{b}}$ and $\boldsymbol{J}^g_{\boldsymbol{tr}}$:

$$\boldsymbol{J}^k_{\boldsymbol{b}} = \boldsymbol{J}^k_{\boldsymbol{b,se}} + \boldsymbol{J}^k_{\boldsymbol{b,eq}} \tag{137}$$

$$\boldsymbol{J}^g_{\boldsymbol{tr}} = \boldsymbol{J}^g_{\boldsymbol{tr,se}} + \boldsymbol{J}^g_{\boldsymbol{tr,eq}} \tag{138}$$

where, according to Eq.(29):

$$\boldsymbol{J}^k_{\boldsymbol{b,eq}} = \bar{\boldsymbol{R}}(\boldsymbol{R}^T\bar{\boldsymbol{R}})^{-1}(\boldsymbol{R}^T\boldsymbol{J}^k_{\boldsymbol{b}}), \qquad \boldsymbol{J}^k_{\boldsymbol{b,se}} = \boldsymbol{J}^k_{\boldsymbol{b}} - \boldsymbol{J}^k_{\boldsymbol{b,eq}} \tag{139}$$

$$\boldsymbol{J}^g_{\boldsymbol{tr,eq}} = \bar{\boldsymbol{R}}(\boldsymbol{R}^T\bar{\boldsymbol{R}})^{-1}(\boldsymbol{R}^T\boldsymbol{J}^g_{\boldsymbol{tr}}), \qquad \boldsymbol{J}^g_{\boldsymbol{tr,se}} = \boldsymbol{J}^g_{\boldsymbol{tr}} - \boldsymbol{J}^g_{\boldsymbol{tr,eq}} \tag{140}$$

Using Eq.(58) and Eq.(69) we can readily compute the coarse-scale counterparts of $\boldsymbol{J}^k_{\boldsymbol{b}}$ and $\boldsymbol{J}^g_{\boldsymbol{tr}}$, denoted by $\boldsymbol{\mathcal{N}}^{T^k}$ and $\bar{\boldsymbol{\mathcal{N}}}^{T^g}$, respectively, by:

$$\boldsymbol{\mathcal{N}}^{k^T} = \boldsymbol{V}^T\boldsymbol{\mathcal{Q}}^T(\mathcal{P}_{\mathbf{f}}\boldsymbol{J}^k_{\boldsymbol{b,eq}}) + (\boldsymbol{T}^T\boldsymbol{H}^{-1}\boldsymbol{\Phi}^T)\boldsymbol{J}^k_{\boldsymbol{b,se}}, \qquad k = 1, 2 \ldots n_{mat} \tag{141}$$

$$\bar{\boldsymbol{\mathcal{N}}}^{g^T} = \boldsymbol{V}^T\boldsymbol{\mathcal{Q}}^T(\mathcal{P}_{\mathbf{f}}\boldsymbol{J}^k_{\boldsymbol{tr,eq}}) + (\boldsymbol{T}^T\boldsymbol{H}^{-1}\boldsymbol{\Phi}^T)\boldsymbol{J}^k_{\boldsymbol{tr,se}}, \qquad g = 1, 2 \ldots b_{non} \tag{142}$$

With $\boldsymbol{\mathcal{N}}^k$ and $\bar{\boldsymbol{\mathcal{N}}}^g$ at hand, the expression for the coarse-scale body forces can be finally expressed as

$$\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}} = \sum_{k=1}^{n_{mat}} \boldsymbol{\mathcal{N}}^{k^T}(\rho^k \hat{\boldsymbol{Q}}^{e^T}\boldsymbol{g}) + \sum_{g=1}^{b_{non}} \bar{\boldsymbol{\mathcal{N}}}^{g^T}\boldsymbol{t'}^{e(g)}. \tag{143}$$

According to the preceding equation, to compute the vector of coarse-scale body forces $\boldsymbol{\mathcal{F}}^e_{\boldsymbol{b}}$ of each subdomain Ω^e, one only has to specify the density ρ^k of the distinct materials forming the unit cell ($k = 1, 2 \ldots n^e_{mat}$); the orientation of the subdomain (through the rotation matrix $\hat{\boldsymbol{Q}}^e \in \mathbb{R}^{n_{sd}\times n_{sd}}$); and the traction vectors $\boldsymbol{t'}^{e(k)} \in \mathbb{R}^{n_{sd}}$ acting on each of the b_{non} non-interface boundaries of the unit cell. The reduced-order matrices $\boldsymbol{\mathcal{N}}^k$ and $\bar{\boldsymbol{\mathcal{N}}}^g$ depend only on the geometry and basis modes of the unit cell, and hence they can be precomputed in the offline phase.

References

[1] Ashby, M., 1992. Physical modelling of materials problems. Materials Science and Technology 8, 102–111.

[2] Barrault, M., Maday, Y., Nguyen, N., Patera, A., 2004. An empirical interpolation'method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique 339, 667–672.

[3] Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge Univ Pr.

[4] Brands, B., Davydov, D., Mergheim, J., Steinmann, P., 2019. Reduced-order modelling and homogenisation in magneto-mechanics: A numerical comparison of established hyper-reduction methods. Mathematical and Computational Applications 24, 20.

[5] Caicedo, M., Mroginski, J.L., Toro, S., Raschi, M., Huespe, A., Oliver, J., 2019. High performance reduced order modeling techniques based on optimal energy quadrature: application to geometrically non-linear multiscale inelastic material modeling. Archives of Computational Methods in Engineering 26, 771–792.

[6] Chapman, T., Avery, P., Collins, P., Farhat, C., 2017. Accelerated mesh sampling for the hyper reduction of nonlinear computational models. International Journal for Numerical Methods in Engineering 109, 1623–1654.

[7] Chaturantabut, S., Sorensen, D., 2010. Discrete empirical interpolation for nonlinear model reduction, in: Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, IEEE. pp. 4316–4321.

[8] Cook, R., 1995. Finite element modeling for stress analysis. John Wiley and Sons., New York.

[9] Craig Jr, R.R., Bampton, M.C., 1968. Coupling of substructures for dynamic analyses. AIAA journal 6, 1313–1319.

[10] Dvorak, G., Wafa, A., Bahei-El-Din, Y., 1994. Implementation of the transformation field analysis for inelastic composite materials. Computational Mechanics 14, 201–228.

[11] Efendiev, Y., Galvis, J., Hou, T., 2012. Generalized multiscale finite element methods (gmsfem) .

[12] Farhat, C., Avery, P., Chapman, T., Cortial, J., 2014. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. International Journal for Numerical Methods in Engineering 98, 625–662.

[13] Farhat, C., Chapman, T., Avery, P., 2015. 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, 1077–1110.

[14] Felippa, C., Haugen, B., 2005. A unified formulation of small-strain corotational finite elements: I. theory. Computer Methods in Applied Mechanics and Engineering 194, 2285–2335.

[15] Feyel, F., Chaboche, J., 2000. Fe-2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. Computer methods in applied mechanics and engineering 183, 309–330.

[16] Fish, J., Shek, K., Pandheeradi, M., Shephard, M., 1997. Computational plasticity for composite structures based on mathematical homogenization: Theory and practice. Computer Methods in Applied Mechanics and Engineering 148, 53–73.

[17] Geers, M., Kouznetsova, V., Brekelmans, W., 2010. Multi-scale computational homogenization: Trends and challenges. Journal of computational and applied mathematics 234, 2175–2182.

[18] Gibson, L.J., Ashby, M.F., 1999. Cellular solids: structure and properties. Cambridge university press.

[19] Golub, G.H., Van Loan, C.F., 2012. Matrix computations. volume 3. JHU press.

[20] Hager, W.W., 1989. Updating the inverse of a matrix. SIAM review 31, 221–239.

[21] Hernández, J.A., Caicedo, M.A., Ferrer, A., 2017. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. Computer Methods in Applied Mechanics and Engineering 313, 687–722.

[22] Hernández, J.A., Oliver, J., Huespe, A.E., Caicedo, M.A., Cante, J.C., 2014. High-performance model reduction techniques in computational multiscale homogenization. Computer Methods in Applied Mechanics and Engineering 276, 149–189.

[23] Hogben, L., 2006. Handbook of linear algebra. Chapman & Hall/CRC.

[24] Kerfriden, P., Goury, O., Rabczuk, T., Bordas, S.P.A., 2013. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. Computer methods in applied mechanics and engineering 256, 169–188.

[25] Kirchdoerfer, T., Ortiz, M., 2016. Data-driven computational mechanics. Computer Methods in Applied Mechanics and Engineering 304, 81–101.

[26] Lacarbonara, W., Paolone, A., 2007. On solution strategies to saint-venant problem. Journal of Computational and Applied Mathematics 206, 473–497.

[27] Lawson, C.L., Hanson, R.J., 1995. Solving least squares problems. volume 15. Siam.

[28] Lloberas-Valls, O., Rixen, D., Simone, A., Sluys, L., 2011. Multiscale domain decomposition analysis of quasi-brittle heterogeneous materials. International Journal for Numerical Methods in Engineering .

[29] Millán, D., Arroyo, M., 2013. Nonlinear manifold learning for model reduction in finite elastodynamics. Computer Methods in Applied Mechanics and Engineering .

[30] Nukala, P.K.V., White, D.W., 2004. A mixed finite element for three-dimensional nonlinear analysis of steel frames. Computer Methods in Applied Mechanics and Engineering 193, 2507–2545.

[31] Oliver, J., Caicedo, M., Huespe, A., Hernández, J., Roubin, E., 2017. Reduced order modeling strategies for computational multiscale fracture. Computer Methods in Applied Mechanics and Engineering 313, 560–595.

[32] Park, K., Felippa, C.A., 1998. A variational framework for solution method developments in structural mechanics. Journal of Applied Mechanics 65, 242–249.

[33] Park, K., Felippa, C.A., 2000. A variational principle for the formulation of partitioned structural systems. International Journal for Numerical Methods in Engineering 47, 395–418.

[34] Rixen, D.J., 2004. A dual craig–bampton method for dynamic substructuring. Journal of Computational and applied mathematics 168, 383–391.

[35] Rocha, I., van der Meer, F., Sluys, L., 2019. Efficient micromechanical analysis of fiber-reinforced composites subjected to cyclic loading through time homogenization and reduced-order modeling. Computer Methods in Applied Mechanics and Engineering 345, 644–670.

[36] Sadd, M.H., 2009. Elasticity: theory, applications, and numerics. Academic Press.

[37] Salmon, G., Johnson, J.E., Malhas, F.A., 1996. Steel structures: Design and behavior, 4th. Edition. Editorial Harper Collins .

[38] Simo, J.C., Hughes, T.J.R., 1998. Computational inelasticity. Springer, New York.

[39] Yano, M., Patera, A.T., 2019. An lp empirical quadrature procedure for reduced basis treatment of parametrized nonlinear pdes. Computer Methods in Applied Mechanics and Engineering 344, 1104–1123.