A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics

P. Kerfriden[1*], O. Goury[1], T. Rabczuk[2], S.P.A. Bordas[1]

[1] Cardiff University, School of Engineering
Queen's Buildings, The Parade, Cardiff CF24 3AA, Wales, UK
[2] Institute of Structural Mechanics, Bauhaus-University Weimar
Marienstraße 15, 99423 Weimar, Germany

November 10, 2012

Abstract

We propose in this paper a reduced order modelling technique based on domain partitioning for parametric problems of fracture. We show that coupling domain decomposition and projection-based model order reduction permits to focus the numerical effort where it is most needed: around the zones where damage propagates. No *a priori* knowledge of the damage pattern is required, the extraction of the corresponding spatial regions being based solely on algebra. The efficiency of the proposed approach is demonstrated numerically with an example relevant to engineering fracture.

Keywords: model order reduction, proper orthogonal decomposition (POD), domain decomposition, nonlinear fracture mechanics, system approximation, parametric time-dependent problems

1 Introduction

Engineering problems are very often characterised by a large ratio between the scale of the structure and the scale at which the phenomena of interest need to be described. In fracture mechanics, the initiation and propagation of cracks is the result of localised microscopic phenomena. These phenomena are usually represented in a homogenised manner at a scale which is suitable for the simulation: the scale of the coarser material heterogeneities (meso-scale), or the engineering scale when such a coarse representation allows for predictive results. In any case, the local nature of fracture leads to large numerical models because sharp local gradients need to be correctly represented or because the meso-structure needs to be described in an explicit manner. To some extent, the availability of super-computing facilities alleviate this difficulty. However, in engineering design processes, a prohibitively high number of solutions might be of interest, for a range of values of design parameters, or to take into account the effect of randomness in the model for instance. Therefore, one needs to devise efficient strategies for the solution to parametric multiscale problems. In doing so, the availability of a range of efficient numerical methods for the solution to one particular realisation of the parametric problem (homogenisation techniques, advanced discretisation tools, domain decomposition and multiscale-based preconditioners for parallel computing) should not be ignored.

Model order reduction techniques that are based on the projection of fine scale problems in reduced spaces are a potential solution to this issue. Such strategies rely on the fact that the solutions to the fine-scale problem obtained for different values of the input parameters can be often represented accurately

*email: *kerfridenp@cardiff.ac.uk*, tel: +44 (0)29 20874071, fax: +44 (0)29 20874716

in low-dimensional subspaces spanned by well-chosen basis functions at the fine scale. Applying this idea, the numerous unknowns that arise from the discretisation of the fine-scale problem are reduced to a few state variables (i.e. the amplitude associated to each of the basis functions). Of course, obtaining the aforementioned global basis functions still requires heavy computations at the fine scale. Therefore, this class of methods is of interest if (i) the goal is to interact with a model (one can afford expensive "offline" computations in order to allow the user to interact with the reduced model in real or quasi-real time) or (ii) the cost of computing the global basis remains small when compared to the cost of solving the fine-scale problem for a large range of input parameters. This paper addresses the latter case, with a restriction to the design of structural components under extreme loading conditions.

Projection-based reduction methods have been extensively studied in system engineering (see the review proposed in [1]), fluid mechanics [2, 3, 4, 5, 6] and structural dynamics [7, 8, 9, 10, 11, 12]. The theory and applicability of various projection-based model order reduction methods such as component mode synthesis [13, 7], the reduced basis method [14, 15, 16], the proper orthogonal decomposition [17, 18, 2] which will be used in this work, the *a priori* hyperreduction method [19, 20] or the proper generalised decomposition [21, 22, 23] are now well-established in the linear to mildly nonlinear cases. Some attempts have been proposed to extend this concept to strong nonlinearities, in particular in structural mechanics [24, 19, 25, 26]. This background makes it conceivable to use such methods in complex engineering problems such as fracture mechanics.

Fracture mechanics is characterised by an intrinsic lack of separation of scales between the engineering scale and the scale at which damage initiation is described. Consequently, these problems are not directly reducible by the aforementioned methods (this fact will be illustrated in the core of the paper). More precisely, the level of reducibility of such multiscale problems depends on the region of the domain which is considered. Typically, the solution in the zones where damage initiates and propagates will not be correctly approximated in low-dimensional subspaces. To circumvent this difficulty, the idea followed in this work is to use a partition of the structural components into substructures and perform a reduction of the resulting subproblems only if such a reduction can be done without sacrificing accuracy.

The concept of local reduced basis itself is not new. It probably originates from the work of Craig and Bampton [7], who proposed a reduction by projection on a modal basis defined over predefined subdomains. This idea has been explored and improved in [27, 11, 12], or coupled with other reduction methods, as in the case of the proper generalised decomposition [21]. A closely related family of solvers uses this concept within local/global approaches: only part of the domain is reduced (sufficiently far away from the sources of nonlinearity) [10, 28, 29, 6], or the global reduced model is locally enriched by a fine-scale description [30, 31, 32] (these two approaches are equivalent when the reduced model is used as a preconditioner for the local fine-scale problem in the former group of methods [29]). The work presented here is novel in the sense that (i) it is the first formal coupling between Schur-based domain decomposition approaches and model order reduction by the Proper Orthogonal Decomposition and (ii) it is, to the authors' knowledge, the first application of systematic partitioned model order reduction for multiscale fracture.

Reduced order models obtained by the proper orthogonal decomposition (see for instance [33, 31, 34, 35, 36]) are powerful tools to reduce the computational burden associated with the repetitive analysis of parametrised nonlinear problems. The principle is to build the projection basis from the knowledge of a set of fine-scale solutions corresponding to a certain number of chosen values of the input parameters (the so-called "snapshots"). The proper orthogonal decomposition (POD) is used to extract attractive reduced spaces from these fine scale solutions in an "offline" phase (we use here the terminology developed for interactivity). Classical Galerkin-based reduction is finally deployed to compute a reliable approximation of the solution to the boundary value problem for arbitrary values of the input parameters at reduced cost ("online" phase). Let us emphasize the fact that, by construction, this family of reduction techniques rely on the "offline" computation of fine-scale solutions (like the reduced-basis method, and as opposed to the proper generalised decomposition and *a priori* hyperreduction methods, which only require cheap fine-scale predictors).

These "offline" computations are potentially expensive in the case of multiscale problems, and our

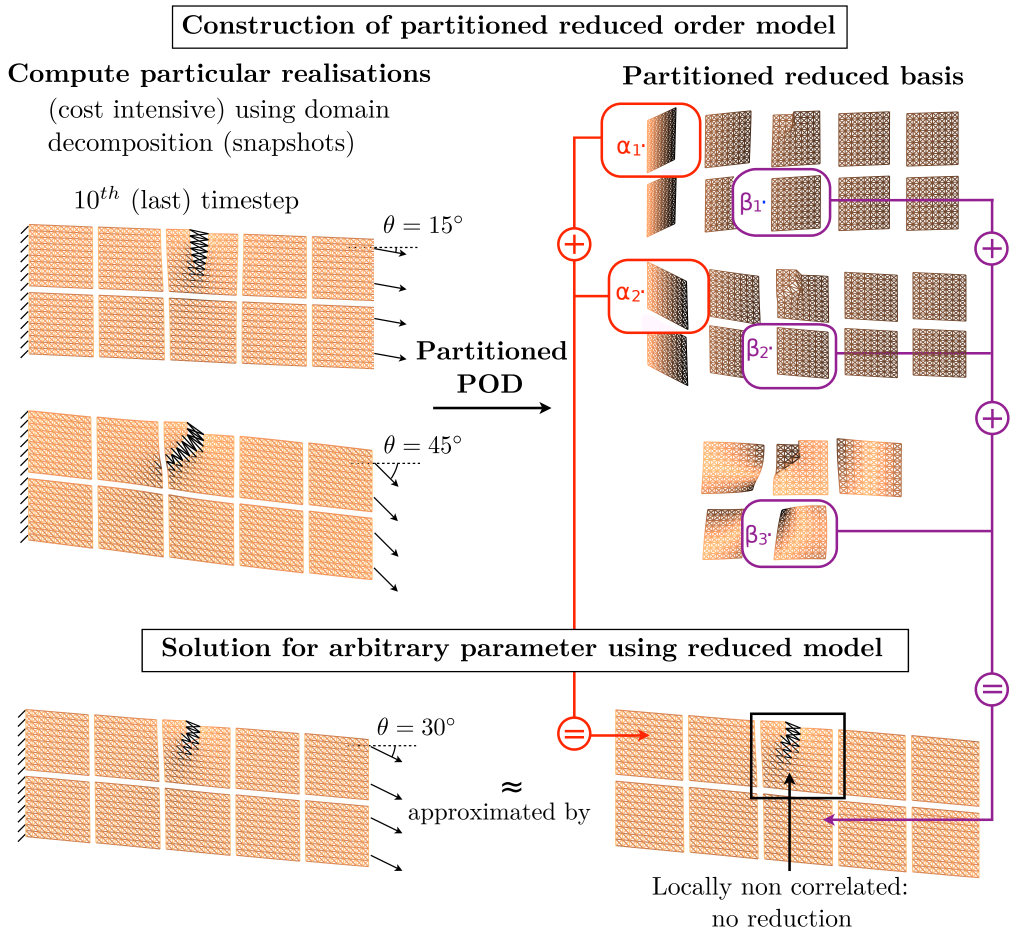


Figure 1: Schematic representation of the partitioned POD-based model order reduction strategy. A Snapshot POD is performed locally for each subdomain in an "offline" phase, which requires the "truth" solution corresponding to a set of particular parameter values. In the"online" phase, the solution corresponding to any value of the parameter is approximated by making use of a Galerkin projection of the governing equations in the local POD subspaces. If the convergence of the local POD transforms is not satisfying in the"offline" phase, the corresponding subproblems are systematically solved without reduction in the "online" phase (Galerkin projection of the governing equations in the local "truth" space). The darkest bars correspond to a completely damaged state of the material, while the lightest bars are undamaged

conception of the design process is that domain decomposition methods [37, 38, 39, 40], which are, to date, probably the most efficient family of parallel solvers, could be used to make them tractable. Examples of parallel computations using domain decomposition methods in the case of fracture can be found in [41, 42]. The purpose of this work is to reuse the substructured nature of the information generated during the "offline" stage to accelerate the solution process of the "online" stage. The choice of the domain decomposition method itself is not of prime interest here. Conceptually, we believe that the work presented in this paper can be extended to Schwartz-based methods, as done for the proper generalised decomposition in the LaTin framework [21], or to other Schur-dual based domain decomposition methods, as presented in [11] for component mode synthesis. We will focus in this work on the primal Schur-based domain decomposition method proposed in [37, 38]. This method relies on a static condensation of the subproblems on the interface degrees of freedom, and a solution of the resulting problem by a projected, preconditioned conjugate gradient in order to ensure a certain level of scalability. We propose to use the snapshot POD method to construct reduced models of the sub-problems corresponding to the interior degrees of freedom of each subdomain.

The proposed substructured approach to model order reduction (see a schematic representation in figure 1) is adapted to the multiscale nature of fracture problems and provides benefits in terms of applicability of POD-based reduction techniques, along the following lines. Firstly, the POD transform, even when using the snapshot technique proposed in [2] can be prohibitively expensive to compute. This issue was treated in [3] by preserving the distributed nature of the snapshot data and reconstructing an approximation of the first modes of the global POD transform from local transforms computed independently for each subdomain. In our case, the POD bases will be used locally, and therefore, their parallel construction is natural. Secondly, using local reduced bases means that the dimension of the reduced spaces, can be adapted to the level of nonlinearity of the subproblems (seen as a statistic correlation of the snapshot data by the POD transform). As mentioned previously, the domain decomposition framework makes it natural to switch from a model order reduction solver to a full scale solver for the solution of subproblems for which no relevant low-dimensional reduced space can be constructed. Notice that similar ideas have been used in the context of domain decomposition methods without reduction for the treatment of localised nonlinearities arising in fracture mechanics. In [43], subproblems corresponding to domains far away from the zones of interest are treated as linear, and the local finite element discretisation is coarsened to allow for computational savings. In [44] and [45], the preconditioner of the domain decomposition method is used for the coarse solution of subproblems that are far away from the process zones. At last, we believe that the systematic decomposition of the domain makes the solution of propagating nonlinearities by reduced order techniques more amenable than local refinements around evolving zones of interest.

The paper is organised as follows. In section 2, we give the general assumptions regarding the class of nonlinear problems which are addressed in this paper. Section 3 introduces classical model order reduction by projection. We focus on the snapshot POD methodology and establish the state-of-the-art of system approximations for nonlinear problems. An example of application of POD-based model order reduction in the case of fracture mechanics is presented to highlight the difficulties due to the local lack of correlation in the data. In section 4, we introduce the primal domain decomposition method, and formally develop a POD-based model order reduction of the sub-problems in a Galerkin context. An inductive method is proposed to determine the set of fine-scale solutions that should be used to obtain a certain level of accuracy in the partitioned snapshot POD. A system approximation strategy for the partitioned POD approach is developed in section 5. Finally, we propose results in terms of running time in section 6 (as a first step, the partitioned POD is used in a serial computing approach), and discuss further improvements for the proposed strategy.

2 General problem statement

We consider the evolution of a structure described by the partial differential equations of continuum mechanics (mechanical equilibrium and constitutive law with appropriate boundary conditions) on a bounded spatial domain Ω, over time interval $\mathcal{T} = [0, T]$. The evolution in time is supposed to be quasi-static. We focus on nonlinear constitutive material models representing the progressive failure of structures, such as plasticity or damage. We assume that the damage processes are rate-independent. The mechanical problem is parametrised by a set of real variables $\underline{\boldsymbol{\mu}}$ that evolves in parameter domain $\mathcal{P} \subset \mathbb{R}^{n_\mu}$.

Performing a space discretisation (finite element in our examples) of such a problem leads to a system of coupled nonlinear (ordinary differential of viscous effects were described) equations. We look for the parametric evolution of the state variables $\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}) \in \mathbb{R}^{n_\mathrm{u}}$ satisfying the following semi-discrete problem

$$\forall\,(t,\underline{\boldsymbol{\mu}}) \in \mathcal{T} \times \mathcal{P}, \qquad \underline{\mathbf{F}}_{\mathbf{int}}\left(\left(\underline{\mathbf{U}}(\tau;\underline{\boldsymbol{\mu}})\right)_{\tau\in[0,t]};\underline{\boldsymbol{\mu}}\right) + \underline{\mathbf{F}}_{\mathbf{ext}}(t;\underline{\boldsymbol{\mu}}) = \underline{\mathbf{0}}\,. \tag{1}$$

The vector of internal forces, $\underline{\mathbf{F}}_{\mathbf{int}} \in \mathbb{R}^{n_\mathrm{u}}$, is a non-linear function of the current state variables $\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})$ (e.g. vector of nodal values of the displacement field in finite element; we will therefore refer to it as "displacement"). n_u is the number of spatial unknowns in system (1). As we model structural damage, the vector of internal forces at time t also depends on the history of the state variables $\left(\underline{\mathbf{U}}(\tau;\underline{\boldsymbol{\mu}})\right)_{\tau\in[0,t[}$ over the past time interval $[0,t[$. Typically, the dependence of $\underline{\mathbf{F}}_{\mathbf{int}}$ to the history of the displacement is due to non-reversible material processes. In the context of parametric problems, $\underline{\mathbf{F}}_{\mathbf{int}}$ may additionally depend on the design variables (design-dependent elastic constants for instance). $\underline{\mathbf{F}}_{\mathbf{ext}} \in \mathbb{R}^{n_\mathrm{u}}$ is the vector of external forces, which may depend on time and on the design variables (design-dependant external load for instance).

A classical time discretisation of semi-discrete system (1) is performed. We search for a sequence of solutions $\left(\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})\right)_{t\in\mathcal{T}^\mathrm{h}}$, where we introduce the discrete time space $\mathcal{T}^\mathrm{h} = \{t_0,\, t_1,\, ...,\, t_{n_\mathrm{t}}\}$ such that $t_0 = 0$ and $t_{n_\mathrm{t}} = T$, which satisfies the fully discrete set of equations

$$\forall\,(t,\underline{\boldsymbol{\mu}}) \in \mathcal{T}^\mathrm{h} \times \mathcal{P}, \qquad \underline{\mathbf{F}}_{\mathbf{int}}\left(\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}), \left(\underline{\mathbf{U}}(\tau;\underline{\boldsymbol{\mu}})\right)_{\tau\in\mathcal{T}^h,\,\tau<t};\underline{\boldsymbol{\mu}}\right) + \underline{\mathbf{F}}_{\mathbf{ext}}(t;\underline{\boldsymbol{\mu}}) = \underline{\mathbf{0}} \tag{2}$$

System (2) is solved sequentially in time, and we assume that the structure is undamaged and at rest at t_0. At an arbitrary time $t \in \mathcal{T}^\mathrm{h}$, the discrete history of the displacement $\left(\underline{\mathbf{U}}(\tau;\underline{\boldsymbol{\mu}})\right)_{\tau\in\mathcal{T}^h,<t}$ is known, which allows to compute vector $\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})$. For readability, the dependence of the system of equations and of the solution vector to the discrete history of the variables, to the time and to the parameter will be explicitly written only if necessary.

The space and time discretisation are assumed to be sufficiently fine for our purpose (e.g.: extraction of an engineering quantity of interest). In this context, $\left(\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})\right)_{t\in\mathcal{T}^\mathrm{h}}$ is referred to as the "truth" solution as it is the one that will be approximated in the reduced order modelling approach.

Discrete system (2) at current time $t \in \mathcal{T}^\mathrm{h}$ is *a priori* nonlinear. It is solved by a usual Newton-Raphson algorithm. At iteration $i+1$ of the nonlinear solver, a tangent linear system is solved:

$$\text{Find } \underline{\mathbf{\Delta U}}^{i+1} \in \mathbb{R}^{n_\mathrm{u}} \text{ such that } \quad \underline{\underline{\mathbf{K}}}^i\, \underline{\mathbf{\Delta U}}^{i+1} = -\underline{\mathbf{R}}^i\,, \tag{3}$$

where $\underline{\mathbf{\Delta U}}^{i+1} = \underline{\mathbf{U}}^{i+1} - \underline{\mathbf{U}}^i$ is an increment in the displacement vector ($\underline{\mathbf{U}}^{i+1}$ is the actual solution of linear prediction $i+1$), $\underline{\underline{\mathbf{K}}}^i = \left.\frac{\partial \underline{\mathbf{F}}_{\mathbf{int}}(\underline{\mathbf{U}})}{\partial \underline{\mathbf{U}}}\right|_{\underline{\mathbf{U}}=\underline{\mathbf{U}}^i}$ is the tangent operator and $\underline{\mathbf{R}}^i = \underline{\mathbf{F}}_{\mathbf{int}}(\underline{\mathbf{U}}^i) + \underline{\mathbf{F}}_{\mathbf{ext}}$ is the residual of the fully discrete system of equations. The Newton algorithm is stopped if the relative euclidean norm of the residual at iteration $i+1$, $\frac{\|\underline{\mathbf{R}}^{i+1}\|_2}{\|\underline{\mathbf{F}}_{\mathbf{ext}}\|_2}$, is lower than a chosen tolerance ϵ_new.

3 Model Order Reduction and Proper Orthogonal Decomposition

Let us recall that our goal is to solve problem (2) for a range of admissible values of the design parameter. In this context, the property underlying the applicability of projection-based MOR is that variations in the design variables generate variations in the solution which can be represented in an attractive low-dimensional subspace of $\mathbb{R}^{n_u}$. Supposing that we can obtain a basis for this subspace, called ”reduced space”, for instance by a particular application of the Proper Orthogonal Decomposition (“offline phase” consisting of “truth” computations), then the evolution problem (2) can be solved approximately for any value of the parameter by looking for the solution in the reduced space (“online phase”, whose complexity must not depend on n_u).

3.1 Projection-based model order reduction

Let us write that the solution of (2) can be approximated, at any time $t \in \mathcal{T}^h$ and for any value of the parameter $\mu \in \mathcal{P}$, in a subspace of $\mathbb{R}^{n_u}$ spanned by (a few) identified basis vectors $(\underline{\mathbf{C}}_i(t;\mu))_{i\in[\![1,n_c]\!]}$ belonging to $\mathbb{R}^{n_u}$:

$$\forall\,(t,\underline{\boldsymbol{\mu}}) \in \mathcal{T}^h \times \mathcal{P}, \qquad \underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}) \approx \sum_{i=1}^{n_c} \underline{\mathbf{C}}_i(t;\underline{\boldsymbol{\mu}})\,\alpha_i(t,\underline{\boldsymbol{\mu}}) = \underline{\underline{\mathbf{C}}}(t;\underline{\boldsymbol{\mu}})\,\underline{\boldsymbol{\alpha}}(t,\underline{\boldsymbol{\mu}})\,. \tag{4}$$

where $\underline{\underline{\mathbf{C}}}(t;\underline{\boldsymbol{\mu}}) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_c}$ is a matrix whose columns are the basis vectors $(\underline{\mathbf{C}}_i(t;\underline{\boldsymbol{\mu}}))_{i\in[\![1,n_c]\!]}$ and $\underline{\boldsymbol{\alpha}}(t,\underline{\boldsymbol{\mu}})$ is a vector of reduced state variables $(\alpha_i(t,\underline{\boldsymbol{\mu}}))_{i\in[\![1,n_c]\!]}$ that needs to be computed “online”. We emphasize that the reduced space $\mathrm{Im}(\underline{\underline{\mathbf{C}}}(t;\underline{\boldsymbol{\mu}}))$ might depend on time and parameter, depending on the method of extraction performed “offline”.

Injecting this approximation into (2) at a particular point $(t,\underline{\boldsymbol{\mu}})$ of the time-parameter domain $\mathcal{T}^h \times \mathcal{P}$, one obtains an over-constrained set of equations in the n_c reduced state variables $\underline{\boldsymbol{\alpha}}$ ($n_c \ll n_u$). Let us define the residual of (2) by

$$\forall\,\underline{\boldsymbol{\alpha}}^\star \in \mathbb{R}^{n_c}, \quad \underline{\widetilde{\mathbf{R}}}(\underline{\boldsymbol{\alpha}}^\star) \stackrel{\text{def}}{=} \underline{\mathbf{R}}(\underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}) = \underline{\mathbf{F}}_{\mathbf{int}}\left(\underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\alpha}}^\star\right) + \underline{\mathbf{F}}_{\mathbf{ext}} \tag{5}$$

Determining optimal values for the reduced variables can be done in different ways in the “online phase”, depending on the physical quantities of interest and on computational tractability and stability issues. The most widely used methods are the Galerkin projection of the residual (5) and its least-square minimisation. The latter reads:

$$\underline{\boldsymbol{\alpha}} = \underset{\underline{\boldsymbol{\alpha}}^* \in \mathbb{R}^{n_c}}{\operatorname{argmin}} \left(\left\| \underline{\widetilde{\mathbf{R}}}(\underline{\boldsymbol{\alpha}}^*) \right\|_{\underline{\underline{\boldsymbol{\Theta}}}} \right)\,, \tag{6}$$

where $\|\,\underline{\widetilde{\mathbf{R}}}\,\|_{\underline{\underline{\boldsymbol{\Theta}}}} = \sqrt{\underline{\widetilde{\mathbf{R}}}^T\,\underline{\underline{\boldsymbol{\Theta}}}\,\underline{\widetilde{\mathbf{R}}}}$ denotes a Θ-norm of the residual vector $\underline{\widetilde{\mathbf{R}}}$ ($\underline{\underline{\boldsymbol{\Theta}}}$ is a symmetric, positive definite operator). Alternatively, in a Galerkin projection framework, $\underline{\boldsymbol{\alpha}}$ is defined as the solution of

$$\underline{\underline{\mathbf{C}}}^T\,\underline{\widetilde{\mathbf{R}}}(\underline{\boldsymbol{\alpha}}) = \underline{\mathbf{0}}\,. \tag{7}$$

We use the Galerkin approach. Nonlinear problem (7) can be solved by a classical Newton algorithm. The linearisation of reduced problem (7) at iteration $i+1$ of a Newton solver (see for instance [25] for more details) leads to the following problem:

$$\underline{\underline{\mathbf{C}}}^T\left(\underline{\widetilde{\mathbf{R}}}^i + \underline{\underline{\mathbf{K}}}^i\,\underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\Delta\alpha}}^{i+1}\right) = \underline{\mathbf{0}} \quad\Longleftrightarrow\quad \underline{\boldsymbol{\Delta\alpha}}^{i+1} = \underset{\underline{\boldsymbol{\Delta\alpha}}^* \in \mathbb{R}^{n_c}}{\operatorname{argmin}} \left(\left\| \underline{\widetilde{\mathbf{R}}}^i + \underline{\underline{\mathbf{K}}}^i\,\underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\Delta\alpha}}^* \right\|_{\left(\underline{\underline{\mathbf{K}}}^i\right)^{-1}} \right), \tag{8}$$

where $\underline{\boldsymbol{\Delta\alpha}}^{i+1} = \underline{\boldsymbol{\alpha}}^{i+1} - \underline{\boldsymbol{\alpha}}^i$ is the unknown quantity of the linear prediction and $\underline{\widetilde{\mathbf{R}}}^i \stackrel{\text{def}}{=} \underline{\widetilde{\mathbf{R}}}(\underline{\boldsymbol{\alpha}}^i)$. Linearised system (8) is a Galerkin reduction (or a least-square reduction as these two approaches are equivalent for the linearised problem when using a $\underline{\underline{\mathbf{K}}}^{-1}$-norm) of linearised equation (3) with the kinematic constraint $\underline{\boldsymbol{\Delta}\mathbf{U}}^{i+1} = \underline{\underline{\mathbf{C}}}\,\underline{\boldsymbol{\Delta\alpha}}^{i+1}$. The solution to (8) reads

$$\underline{\boldsymbol{\Delta\alpha}}^{i+1} = -\left(\underline{\underline{\mathbf{C}}}^T\,\underline{\underline{\mathbf{K}}}^i\,\underline{\underline{\mathbf{C}}}\right)^{-1}\underline{\underline{\mathbf{C}}}^T\,\underline{\widetilde{\mathbf{R}}}^i\,, \tag{9}$$

providing the reduced linearised operator $\underline{\underline{\mathbf{C}}}^T \, \underline{\underline{\mathbf{K}}}^i \, \underline{\underline{\mathbf{C}}}$ (of very small size n_c) is invertible.

At this point, we can notice the two following classical issues in projection-based model order reduction:

- The well-posedness of tangent problems (8) and the accuracy of the solution strongly depends on the choice of the reduced space.
- The Galerkin projection framework presented previously is inefficient. The tangent and residual of the initial problem of evolution must be evaluated at each iteration of the Newton solver. The evaluation of nonlinear function $\underline{\mathbf{F}}_{\mathbf{int}}$ requires a global integration over domain Ω. As a result, the numerical complexity of the reduction technique does not only depend on the dimension of the reduced space but also on the size of the initial problem, which results in insignificant speed-up.

Therefore, a reduction method should provide a "good" reduced space (in the sense of accuracy and stability of the solution), as well as an "efficient" strategy to obtain the "online" solution (significant speed-up compared to the full model, without sacrificing the accuracy expected when using a good reduced space). These two points are discussed in the following sections.

3.2 Proper Orthogonal Decomposition in projection-based model order reduction

3.2.1 Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD) is a popular transform which is classically used to generate relevant bases for projection-based reduced order models. Applied to our parametric evolution problem, the POD decomposes the solution of the problem over the full time-parameter domain $\tilde{\mathcal{P}} \overset{\text{def}}{=} \mathcal{T}^\mathrm{h} \times \mathcal{P}$ as

$$\forall\,(t,\underline{\boldsymbol{\mu}}) \in \tilde{\mathcal{P}}, \qquad \underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}) = \underline{\bar{\mathbf{U}}}(t;\underline{\boldsymbol{\mu}}) + \underline{\boldsymbol{\epsilon}}(t;\underline{\boldsymbol{\mu}}) \tag{10}$$

$$\underline{\bar{\mathbf{U}}}(t;\underline{\boldsymbol{\mu}}) = \sum_{i=1}^{n_\mathrm{p}} \underline{\boldsymbol{\phi}}_i \, \gamma_i(t,\underline{\boldsymbol{\mu}}) = \underline{\underline{\boldsymbol{\phi}}}\,\underline{\boldsymbol{\gamma}}(t,\underline{\boldsymbol{\mu}})\,,$$

such that $\underline{\bar{\mathbf{U}}}$ is the function of separable form (10) that is the closest to the exact solution,

$$\underline{\bar{\mathbf{U}}} = \underset{\underline{\bar{\mathbf{U}}}^* \in \{\underline{\mathbf{Z}} \,|\, \underline{\mathbf{Z}}(t;\underline{\boldsymbol{\mu}}) = \underline{\underline{\boldsymbol{\phi}}}\,\underline{\boldsymbol{\gamma}}(t,\underline{\boldsymbol{\mu}}),\, \forall\,(t,\underline{\boldsymbol{\mu}}) \in \tilde{\mathcal{P}}\}}{\operatorname{argmin}} d(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}^*)\,, \tag{11}$$

with the metric d defined on the space $\bar{\mathcal{U}}$ of functions defined over $\tilde{\mathcal{P}}$ with values in $\mathbb{R}^{n_\mathrm{u}}$:

$$\begin{array}{rccc} d: & \bar{\mathcal{U}} \times \bar{\mathcal{U}} & \to & \mathbb{R} \\ & (\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}) & \mapsto & d(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}) \end{array} \tag{12}$$

$$d(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}) = \int_{\underline{\boldsymbol{\mu}} \in \mathcal{P}} \sum_{t \in \mathcal{T}^\mathrm{h}} \|\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}) - \underline{\bar{\mathbf{U}}}(t;\underline{\boldsymbol{\mu}})\|_2^2 \; d\underline{\boldsymbol{\mu}}\,. \tag{13}$$

$(\underline{\boldsymbol{\phi}}_i)_{i \in [\![1,n_\mathrm{p}]\!]}$ are "space" vectors that belong to $\mathbb{R}^{n_\mathrm{u}}$ and are further constrained to be orthonormal with respect to the usual euclidean scalar product of $\mathbb{R}^{n_\mathrm{u}}$, while $(\gamma_i)_{i \in [\![1,n_\mathrm{p}]\!]}$ are scalar functions of time and parameter. We emphasise here the fact that the spatial basis $\underline{\underline{\boldsymbol{\phi}}}$ is not known *a priori* but is assumed to be independent on time and parameter (i.e.: we perform a separation of variables). The POD essentially delivers a decomposition of the exact solution $\underline{\mathbf{U}}$ into bi-orthonormal modes $\Big((\underline{\boldsymbol{\phi}}_i),\gamma_i\Big)_{i \in [\![1,n_\mathrm{p}]\!]}$ of decreasing importance. The truncation of those modes at order n_p provides the best representation of the solution with a basis of n_p modes in the sense that the sum over the time-parameter domain of all distances between the exact solution and its n_p-order approximation is minimised. Distance $d(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}})$ is expected to decrease quickly with the order of the decomposition.

3.2.2 Snapshot POD

The POD transform (10,13) requires the knowledge of the exact solution over $\tilde{\mathcal{P}}$, which is not compatible with our desired usage. However, one can derive a similar transform that computes an optimal decomposition of the solution $\underline{\mathbf{U}}$ over a discrete subset $\tilde{\mathcal{P}}^{\mathrm{s}} = \mathcal{T}^{\mathrm{h}} \times \mathcal{P}^{\mathrm{s}}$ of $\tilde{\mathcal{P}}$.

$$\forall\,(t,\underline{\boldsymbol{\mu}}) \in \tilde{\mathcal{P}}^{\mathrm{s}}, \qquad \underline{\bar{\mathbf{U}}}(t;\underline{\boldsymbol{\mu}}) = \underline{\bar{\mathbf{U}}}^{\mathrm{s}}(t;\underline{\boldsymbol{\mu}}) + \underline{\boldsymbol{\epsilon}}^{\mathrm{s}}(t;\underline{\boldsymbol{\mu}}) \tag{14}$$

$$\underline{\bar{\mathbf{U}}}^{\mathrm{s}}(t;\underline{\boldsymbol{\mu}}) = \sum_{i=1}^{n_{\mathrm{p}}} \underline{\boldsymbol{\phi}}_i\,\gamma_i(t;\underline{\boldsymbol{\mu}}) = \underline{\underline{\boldsymbol{\phi}}}\,\underline{\boldsymbol{\gamma}}(t;\underline{\boldsymbol{\mu}})\,,$$

such that $\underline{\bar{\mathbf{U}}}^{\mathrm{s}}$ is solution to the optimisation problem:

$$\underline{\bar{\mathbf{U}}}^{\mathrm{s}} = \underset{\underline{\bar{\mathbf{U}}}^{*} \in \{\underline{\mathbf{Z}}\,|\,\underline{\mathbf{Z}}(t;\underline{\boldsymbol{\mu}}) = \underline{\underline{\boldsymbol{\phi}}}\,\underline{\boldsymbol{\gamma}}(t;\mu),\,\forall\,(t;\underline{\boldsymbol{\mu}})\in\tilde{\mathcal{P}}^{\mathrm{s}}\}}{\operatorname{argmin}} d^{\mathrm{s}}(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}^{*}) \tag{15}$$

with d^{s} the metric defined on the space $\bar{\mathcal{U}}^{\mathrm{s}}$ of functions defined over $\tilde{\mathcal{P}}^{\mathrm{s}}$ with values in $\mathbb{R}^{n_{\mathrm{u}}}$:

$$\begin{array}{rccc} d^{\mathrm{s}}: & \bar{\mathcal{U}}^{\mathrm{s}} \times \bar{\mathcal{U}}^{\mathrm{s}} & \rightarrow & \mathbb{R} \\ & (\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}) & \mapsto & d(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}) \end{array} \tag{16}$$

with

$$d^{\mathrm{s}}(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}^{\mathrm{s}}) = \sum_{\underline{\boldsymbol{\mu}}\in\mathcal{P}^{\mathrm{s}}} \sum_{t\in\mathcal{T}^{\mathrm{h}}} \|\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}) - \underline{\bar{\mathbf{U}}}^{\mathrm{s}}(t;\underline{\boldsymbol{\mu}})\|_2^2 \tag{17}$$

$\mathcal{P}^{\mathrm{s}} = \{\underline{\boldsymbol{\mu}}_1, \dots, \underline{\boldsymbol{\mu}}_{n_\mu}\}$ is a discrete subset of the parameter domain $\mathcal{P}$. $\big(\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})\big)_{(t;\underline{\boldsymbol{\mu}})\in\tilde{\mathcal{P}}^{\mathrm{s}}}$ are particular "truth" solutions of problem (2) for some parameters $\underline{\boldsymbol{\mu}} \in \tilde{\mathcal{P}}$, called snapshot. The snapshot POD metric (17) can be viewed as a quadrature rule for its integral counterpart (13).

Optimal reduced spatial space span $\Big((\underline{\boldsymbol{\phi}}_i)_{i\in[\![1,n_{\mathrm{p}}]\!]}\Big)$, with the additional constraint of orthonormality of $(\underline{\boldsymbol{\phi}}_i)_{i\in[\![1,n_{\mathrm{p}}]\!]}$, and corresponding scalar weighting functions $(\gamma_i)_{i\in[\![1,n_{\mathrm{p}}]\!]}$ are given, at any order n_{p}, by

- $\underline{\boldsymbol{\phi}}_i$ is the eigenvector of the POD operator $\underline{\underline{\mathbf{H}}}$ (covariance operators if the snapshot vectors were centred) associated to its i$^{\mathrm{th}}$ largest eigenvalue λ_i. $\underline{\underline{\mathbf{H}}}$ is defined by

$$\underline{\underline{\mathbf{H}}} = \sum_{\underline{\boldsymbol{\mu}}\in\mathcal{P}^{\mathrm{s}}} \sum_{t\in\mathcal{T}^{\mathrm{h}}} \underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})\,\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})^T\,. \tag{18}$$

- $\forall\,(t,\underline{\boldsymbol{\mu}}) \in \tilde{\mathcal{P}}^{\mathrm{s}}, \quad \gamma_i(t;\underline{\boldsymbol{\mu}}) = \underline{\boldsymbol{\phi}}_i^T\,\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}})$

The truncation error of a POD transform of order n_{p} is given by

$$d^{\mathrm{s}}(\underline{\mathbf{U}},\underline{\bar{\mathbf{U}}}^{\mathrm{s}}) = \sum_{i=n_{\mathrm{p}}+1}^{n_{\mathrm{s}}} \lambda_i\,, \tag{19}$$

where $n_{\mathrm{s}} = n_{\mathrm{t}} \times n_\mu$ is the number of snapshot solutions, and therefore the maximum possible rank of operator $\underline{\underline{\mathbf{H}}}$.

The eigenvalue decomposition of $\underline{\underline{\mathbf{H}}}$ is obtained at relatively cheap costs when $n_{\mathrm{t}} \times n_\mu < n_{\mathrm{u}}$ by exploiting the discrete nature of the available information (which is essentially the idea proposed in [2]). One computes the singular value decomposition (SVD) of the snapshot operator $\underline{\underline{\mathbf{S}}} = \Big(\underline{\mathbf{U}}(t_1,\underline{\boldsymbol{\mu}}_1)\ \underline{\mathbf{U}}(t_2,\underline{\boldsymbol{\mu}}_1)\ \dots\ \underline{\mathbf{U}}(t_{n_{\mathrm{t}}},\underline{\boldsymbol{\mu}}_{n_\mu})\Big)$. The SVD reads $\underline{\underline{\mathbf{S}}} = \underline{\underline{\mathbf{Q}}}\,\underline{\underline{\boldsymbol{\Sigma}}}\,\underline{\underline{\mathbf{W}}}^T$ with $\underline{\underline{\mathbf{Q}}}$ and $\underline{\underline{\mathbf{W}}}$ unitary matrices and $\underline{\underline{\boldsymbol{\Sigma}}}$ a rectangular matrix with diagonal upper block. We then have $\underline{\underline{\mathbf{H}}} = \underline{\underline{\mathbf{S}}}\,\underline{\underline{\mathbf{S}}}^T = \underline{\underline{\mathbf{Q}}}\,\underline{\underline{\boldsymbol{\Sigma}}}\,\underline{\underline{\mathbf{W}}}^T\,\underline{\underline{\mathbf{W}}}\,\underline{\underline{\boldsymbol{\Sigma}}}^T\,\underline{\underline{\mathbf{Q}}}^T = \underline{\underline{\mathbf{Q}}}\,\underline{\underline{\boldsymbol{\Sigma}}}\,\underline{\underline{\boldsymbol{\Sigma}}}^T\,\underline{\underline{\mathbf{Q}}}^T$, which is the eigenvalue decomposition of $\underline{\underline{\mathbf{H}}}$ and the eigenvalues are the squares of the singular values of $\underline{\underline{\mathbf{S}}}$. The values of the weighting functions $(\gamma_i)_{i\in[\![1,n_{\mathrm{p}}]\!]}$ over $\tilde{\mathcal{P}}^{\mathrm{s}}$ can be readily extracted from matrix $\underline{\underline{\mathbf{W}}}$ if necessary, but this information is not of particular interest the present context.

3.2.3 Reduced spaces in POD-based model order reduction

The snapshot POD essentially provides an optimal decomposition of the solution in the discrete space $\tilde{\mathcal{P}}^{\mathrm{s}}$. It can be truncated at an order $n_{\mathrm{p}} \leq n_{\mathrm{s}}$ for which the normalised truncation error

$$\nu_{\mathrm{snap}}^2 = d^{\mathrm{s}}(\underline{\mathbf{U}}, \bar{\underline{\mathbf{U}}}^{\mathrm{s}}) = \frac{\displaystyle\sum_{i=n_{\mathrm{p}}+1}^{n_{\mathrm{s}}} \lambda_i}{\displaystyle\sum_{i=1}^{n_{\mathrm{s}}} \lambda_i}, \tag{20}$$

is sufficiently low.

POD-based reduced order modelling proposes to simply discard functions $(\gamma_i)_{i\in[\![1,n_{\mathrm{p}}]\!]}$ (which are only defined for a discrete set of parameter values anyway), and look for the solution of the evolution problem for any value of parameter $\underline{\boldsymbol{\mu}} \in \tilde{\mathcal{P}}$, in the reduced space $\mathrm{span}((\underline{\boldsymbol{\phi}}_i)_{i\in[\![1,n_{\mathrm{p}}]\!]})$. The amplitude associated with the basis vectors are computed optimally by the "online" projection technique given in section 3.1. In this context, it is clear that the snapshot POD is used to define a reduced space for projection-based reduced order modelling (which is therefore independent on time and parameter):

$$\forall\,(t, \underline{\boldsymbol{\mu}}) \in \mathcal{T}^{\mathrm{h}} \times \mathcal{P},\, \forall\, i \in [\![1, n_{\mathrm{c}}]\!] \qquad \underline{\mathbf{C}}_i(t; \underline{\boldsymbol{\mu}}) = \underline{\boldsymbol{\phi}}_i \qquad (n_{\mathrm{c}} = n_{\mathrm{p}}) \tag{21}$$

Remark: *A solution over the initial time-parameter domain $\tilde{\mathcal{P}}$ could be reconstructed by an explicit interpolation of the functions $(\gamma_i)_{i\in[\![1,n_p]\!]}$ (i.e.: interpolation by an arbitrary polynomial basis) or by other implicit interpolation techniques such as Kriging or Moving Least-Squares for instance, as proposed in [35, 46]), which would lead to a decomposition of type* (10). *However, such an explicit interpolation approach in $\tilde{\mathcal{P}}$ is suboptimal as the behaviour of the governing equations between the pre-computed snapshot solutions is unknown. In addition, the Galerkin projection framework defined in section 3.1 permits to reuse the error estimates available in finite element schemes for the certification of the implicitly interpolated solution (see [14, 31, 9, 47] for instance), at least in the linear case. The extension of this idea to nonlinear problems is currently an active area of research and will not be addressed in this contribution.*

An important point to emphasise is the requirement to perform cost-intensive simulations to compute the snapshot in the "offline" phase. We assume in this work that the initial problem of evolution involves a large number of degrees of freedom in space and time and requires high-performance computing for the "truth" solutions to be at reach. In particular, these solutions can be obtained efficiently on parallel architecture by using domain decomposition methods, which are, to date, probably the best parallel solvers for structural mechanics. This requirement will actually serve our needs in the case of fracture, as shown later.

3.3 System approximation

As stated in section 3.1, an approximation of the fully discrete system of equations (2) must be associated with the choice of the reduced space. In order to limit the computational expense due to the evaluation of the nonlinear functions $\underline{\mathbf{F}}_{\mathbf{int}}$, two families of strategies have been intensively studied in the literature.

3.3.1 Linearisation

The first family proposes to linearise [48, 9], or perform a higher-order Taylor expansion [49, 24, 50] of the nonlinear terms in the system of equations governing the "truth" solutions. The reduced linearised operators can be computed once and for all "offline" and reused "online" in the Newton solver. Obviously, the validity of Taylor expansions is only local along the trajectory of the reduced state variables. The authors of [49] proposed an elegant "offline" linearisation of the nonlinear terms of the discrete set of equations that depends on the value of the reduced state variables. In the "online" phase, the nonlinear terms of the discrete set of equations are approximated as a weighted combination of the "offline" trajectory-dependent linearisations.

3.3.2 Evaluation of nonlinear terms on reduced spatial domains

The second family of system approximations proposes to only evaluate the nonlinear function at particular points of the domain. In a first subset of these strategies, the nonlinear function is reconstructed by interpolation over an other POD basis ("gappy" technique) [5, 34, 51, 36]. The expansion of the nonlinear term reads:

$$
\begin{aligned}
&\forall\, t \in \mathcal{T}^h,\ \forall\, \underline{\boldsymbol{\alpha}}^\star \in \mathbb{R}^{n_c}, \\
&\underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}^\star, \left(\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}(\tau,\mu) \right)_{\tau \in \mathcal{T}^h,\, \tau<t} ; \underline{\boldsymbol{\mu}} \right) \;\approx\; \sum_{i=1}^{n_\mathrm{d}} \underline{\mathbf{D}}_i\, \beta_i(\underline{\boldsymbol{\alpha}}^\star, (\underline{\boldsymbol{\alpha}}(\tau,\mu))_{\tau \in \mathcal{T}^h,\, \tau<t}; \underline{\boldsymbol{\mu}}) \\
&\qquad\qquad \approx\; \underline{\underline{\mathbf{D}}}\, \underline{\boldsymbol{\beta}} \left(\underline{\boldsymbol{\alpha}}^\star, (\underline{\boldsymbol{\alpha}}(\tau,\mu))_{\tau \in \mathcal{T}^h,\, \tau<t} ; \underline{\boldsymbol{\mu}} \right) .
\end{aligned}
\tag{22}
$$

The columns of $\underline{\underline{\mathbf{D}}} \in \mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{d}}$ are spatial functions corresponding to a truncated snapshot POD expansion of the image of the reduced space by $\underline{\mathbf{F}}_{\mathbf{int}}$, which is performed "offline". In practice, Newton iterates obtained while solving the reduced model without system approximation are used to define the "static" snapshot space $\{\underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}^\star, (\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}(\tau,\mu))_{\tau \in \mathcal{T}^h,\, \tau<t}; \underline{\boldsymbol{\mu}} \right) \mid t \in \mathcal{T}^h,\ \underline{\boldsymbol{\alpha}}^\star \in \mathbb{R}^{n_c}\}$. Interpolation coefficients $\underline{\boldsymbol{\beta}}$ are found by enforcing that at any point $(t, \underline{\boldsymbol{\mu}})$ of $\tilde{\mathcal{P}}$, the interpolation must be optimal with respect to a limited number n_sa of spatial degrees of freedom:

$$
\underline{\boldsymbol{\beta}} \left(\underline{\boldsymbol{\alpha}}^\star, (\underline{\boldsymbol{\alpha}}(\tau,\mu))_{\tau \in \mathcal{T}^h,\, \tau<t} ; \underline{\boldsymbol{\mu}} \right) = \underset{\underline{\boldsymbol{\beta}}^\star \in \mathbb{R}^{n_\mathrm{d}}}{\operatorname{argmin}} \left(\| \underline{\underline{\mathbf{D}}}\, \underline{\boldsymbol{\beta}}^\star - \underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}^\star, \left(\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}(\tau,\mu) \right)_{\tau \in \mathcal{T}^h,\, \tau<t} ; \underline{\boldsymbol{\mu}} \right) \|_{\underline{\underline{\mathbf{P}}}} \right) \tag{23}
$$

$\underline{\underline{\mathbf{P}}}$ is a boolean diagonal operator with n_sa non-zero entries ($n_\mathrm{sa} \geq n_\mathrm{d}$ and $n_\mathrm{sa} \ll n_\mathrm{u}$) corresponding to the evaluation degrees of freedom of the spatial interpolation of the nonlinear term. $\|\underline{\mathbf{X}}\|_{\underline{\underline{\mathbf{P}}}} = \sqrt{\underline{\mathbf{X}}^T \underline{\underline{\mathbf{P}}}\, \underline{\mathbf{X}}}$ is the semi-norm associated with $\underline{\underline{\mathbf{P}}}$ for an arbitrary vector $\underline{\mathbf{X}} \in \mathbb{R}^{n_\mathrm{u}}$. Substituting this approximation into the full system of equation (2), together with the reduced basis approximation for the displacement vector, the following reduced expression is obtained for the approximation of the "truth" residual (5) at a particular point of the time-parameter domain:

$$
\forall\, \underline{\boldsymbol{\alpha}}^\star \in \mathbb{R}^{n_c}, \qquad \underline{\mathbf{R}}_{\mathbf{gap}}(\underline{\boldsymbol{\alpha}}^\star) \overset{\text{def}}{=} \underline{\underline{\mathbf{D}}} \left(\underline{\underline{\mathbf{D}}}^T \underline{\underline{\mathbf{P}}}\, \underline{\underline{\mathbf{D}}} \right)^{-1} \underline{\underline{\mathbf{D}}}^T \underline{\underline{\mathbf{P}}}\, \underline{\mathbf{F}}_{\mathbf{int}} \left(\underline{\underline{\mathbf{C}}}\, \underline{\boldsymbol{\alpha}}^\star \right) + \underline{\mathbf{F}}_{\mathbf{ext}} \,, \tag{24}
$$

where operator $\underline{\underline{\mathbf{D}}}^T \underline{\underline{\mathbf{P}}}\, \underline{\underline{\mathbf{D}}}$ is assumed to be invertible. The reduced variables can then be obtained in the "offline" phase by minimising the norm of the modified residual, or by solving the Galerkin projection of the governing equations $\underline{\underline{\mathbf{C}}}^T \underline{\mathbf{R}}_{\mathbf{gap}}(\underline{\boldsymbol{\alpha}}) = \underline{\mathbf{0}}$. Only a restriction to the evaluation degrees of freedom of the nonlinear function is calculated to evaluate the residual of the system, which allows the "online" phase of the interpolation scheme to have a numerical complexity that does not depend on the "truth" discretisations.

The second subset of these strategies, comprising the method proposed in [52], the Hyperreduction method [53] and an early version of the Missing Point Estimation technique [54] can be qualified as collocations-based strategies. These methods do not reconstruct the nonlinear function over the domain. They propose instead to look for a solution that is optimal with respect to a few of the equations of the initial system (2). This can be expressed in a least-square approach:

$$
\underline{\boldsymbol{\alpha}} = \underset{\underline{\boldsymbol{\alpha}}^\star \in \mathbb{R}^{n_c}}{\operatorname{argmin}} \left(\| \underline{\mathbf{R}}_{\mathbf{gap}}(\underline{\boldsymbol{\alpha}}^\star) \|_{\underline{\underline{\mathbf{P}}}} \right) , \tag{25}
$$

or in the (Petrov-) Galerkin framework

$$\text{Find } \underline{\boldsymbol{\alpha}} \in \mathbb{R}^{n_c} \text{ such that } \quad \underline{\underline{\mathbf{C}}}^T \underline{\underline{\mathbf{P}}}\, \underline{\mathbf{R}}_{\mathbf{gap}}(\underline{\boldsymbol{\alpha}}) = \underline{\mathbf{0}}\,. \tag{26}$$

The strategies proposed in the literature for this two subset of techniques differ in the way of building operator $\underline{\underline{\mathbf{P}}}$, which requires a critical trade-off between optimality, stability and tractability. In [34], $\underline{\underline{\mathbf{P}}}$ is constructed such that the condition number of operator $\underline{\underline{\mathbf{D}}}^T\, \underline{\underline{\mathbf{P}}}\, \underline{\underline{\mathbf{D}}}$ is minimised. In the hyperreduction method [53], the non-zero entries of $\underline{\underline{\mathbf{P}}}$ correspond to the largest entries (in some sense) of the approximated nonlinear vector function. In [51], the points are selected to limit the growth of the residual error between a solution and its snapshot reconstruction.

3.3.3 Chosen strategy

We will focus in this work on the "gappy" technique, as used in [51] and [36]. Since the main objective of this paper is not the system approximation strategy but the introduction of the partitioned POD technique, this method is selected as the most widely used and studied. We note for the following developments that at a particular point of the time-parameter domain, Newton iteration $i+1$ applied to reduced system (24), in the Galerkin framework, reads:

$$\underline{\boldsymbol{\Delta\alpha}}^{i+1} = -\left(\underline{\underline{\mathbf{C}}}^T \underline{\underline{\mathbf{D}}} (\underline{\underline{\mathbf{D}}}^T \underline{\underline{\mathbf{P}}}\underline{\underline{\mathbf{D}}})^{-1} \underline{\underline{\mathbf{D}}}^T \underline{\underline{\mathbf{P}}}\, \underline{\underline{\mathbf{K}}}\, \underline{\underline{\mathbf{C}}} \right)^{-1} \underline{\underline{\mathbf{C}}}^T \underline{\mathbf{R}}^i_{\mathbf{gap}}\,, \tag{27}$$

where $\underline{\mathbf{R}}^i_{\mathbf{gap}} \overset{\text{def}}{=} \underline{\mathbf{R}}_{\mathbf{gap}}(\underline{\boldsymbol{\alpha}}^i)$.

The application of this technique will be further addressed in the last section of this paper. Meanwhile, we focus on the issue of computing and using relevant POD-based reduced spaces in the particular case of fracture mechanics, using a Partitioned POD approach.

3.4 Example of application of the POD in fracture mechanics

3.4.1 Lattice model

We consider a lattice structure made of n_b damageable bars in uniaxial tension or compression. A bar marked $b \in \mathcal{B} \overset{\text{def}}{=} [\![1, n_\text{b}]\!]$ occupies a 1D linear domain $\Omega^{(b)}$ embedded in $\mathbb{R}^2$, such that $\Omega \overset{\text{def}}{=} \bigcup_{b \in [\![1,n_\text{b}]\!]} \Omega^{(b)}$. We will denote by $\mathcal{P} = \{P_i \,|\, i \in [\![1, n_\text{pt}]\!]\}$ the set of nodes of the lattice structure. Let us define the unit vector $\underline{n}^{(b)}$ attached to bar $b \in \mathcal{B}$ such that if P_i and P_j are the two extremities of $\Omega^{(b)}$ and $i < j$, then $\underline{n}^{(b)} = \frac{\underline{P_i P_j}}{\|\underline{P_i P_j}\|}$. We denote the local coordinate of point $M \in \Omega^{(b)}$ by $s^{(b)} = \|\underline{P_i M}\|$. The global reference frame associated to the physical space is denoted by $\mathcal{R}(0, \underline{e}_x, \underline{e}_y)$.

We look for a two dimensional displacement field $\underline{u}$, and a scalar stress field N defined over Ω that satisfy the system of equations given below. The restriction of these fields to bar $b \in \mathcal{B}$ will be denoted by $\underline{u}^{(b)}$ and $N^{(b)}$ respectively.

Equilibrium. The local mechanical equilibrium of bar $b \in \mathcal{B}$ reads, at any point of domain $\Omega^{(b)}$:

$$\frac{\partial N^{(b)}}{\partial s^{(b)}} + \underline{f} \cdot \underline{n}^{(b)} = 0\,, \tag{28}$$

where the body force $\underline{f}$ is a two-dimensional vector field. At a lattice node $P \in \mathcal{P}$ between a set of bars denoted by $\mathcal{B}_\text{n}^{(i)} \subset \mathcal{B}$, the stresses are required to satisfy the nodal equilibrium, which reads, if no pointwise external force is applied at point P,

$$\sum_{b \in \mathcal{B}_\text{n}^{(i)}} N^{(b)}_{|P}\ \bar{\underline{n}}^{(b)}_{|P} = 0\,, \tag{29}$$

or if P belong to the set of points $\mathcal{P}^{\mathrm{F}} \subset \mathcal{P}$ that are subjected to Neumann boundary conditions,

$$\sum_{b \in \mathcal{B}_{\mathrm{n}}^{(i)}} N_{|P}^{(b)}\ \underline{\bar{n}}_{|P}^{(b)} + \underline{N}_{\mathrm{d}|P} = 0\,. \tag{30}$$

In the previous equation $\underline{N}_{\mathrm{d}|P} \in \mathbb{R}^2$ is a prescribed force. In equilibrium equation (29) and (30), $\underline{\bar{n}}_{|P}^{(b)} = \underline{n}^{(b)}$ if $s_{|P}^{(b)} = 0$ (first extremity of the bar), and $\underline{\bar{n}}_{|P}^{(b)} = -\underline{n}^{(b)}$ otherwise (second extremity of the bar).

Displacement admissibility. We assume that the restriction $\underline{u}^{(b)}$ to beam b of the displacement $\underline{u}$ is linear. Furthermore, at any node $P \in \mathcal{P}$, the continuity of the displacement field between connected beams must be satisfied:

$$\forall (b, b') \in \mathcal{B}_{\mathrm{n}}^{(i)}, \qquad \underline{u}_{|P}^{(b)} = \underline{u}_{|P}^{(b')} = \underline{u}_{|P}\,. \tag{31}$$

The displacement field also satisfies Dirichlet boundary conditions at any node $P \in \mathcal{P}^{\mathrm{u}} \subset \mathcal{P}$ satisfying $\mathcal{P}^{\mathrm{u}} \cap \mathcal{P}^{\mathrm{F}} = \{\}$, which reads

$$\underline{u}_{|P} = \underline{u}_{\mathrm{d}|P}\,, \tag{32}$$

where $\underline{u}_{\mathrm{d}|P} \in \mathbb{R}^2$ is a prescribed displacement.

Constitutive law. The constitutive law relates the stress and displacement fields locally. At time $t \in \mathcal{T}$, and for any $b \in \mathcal{B}$, the constitutive law expressed at an arbitrary point of domain $\Omega^{(b)}$ reads formally

$$N^{(b)} = N^{(b)}\left(\left\{\epsilon^{(b)}\left(\underline{u}_{|\tau}^{(b)}\right)\right\}_{\tau \leq t}\right)\,, \tag{33}$$

where the deformation $\epsilon^{(b)}$ is defined by

$$\epsilon^{(b)}\left(\underline{u}^{(b)}\right) = \frac{\partial \underline{u}^{(b)}}{\partial s^{(b)}} \cdot \underline{n}^{(b)} \tag{34}$$

3.4.2 Damage model

The fracture of the lattice structure is described by classical damage mechanics [55]. We postulate the existence of a free Helmholtz energy at any time $t \in \mathcal{T}$:

$$\psi(\epsilon^{(b)}, d) = \frac{1}{2}\, E(1-d)S\left(\epsilon^{(b)}\right)^2 \tag{35}$$

E is the Young's modulus of bar b, S is its section (assumed constant), and d is a damage variable that ranges from 0 (safe material), to 1 (completely damaged material point). The state equations are obtained by derivation of the free energy with respect to the state variables.

$$N = \frac{\partial \psi}{\partial \epsilon^{(b)}} = E(1-d)S\epsilon^{(b)}\,, \tag{36}$$

$$Y = -\frac{\partial \psi}{\partial d} = \frac{1}{2}\, ES\,\left(\epsilon^{(b)}\right)^2\,. \tag{37}$$

Y is a driving force associated with the damage variable d. To close the system, a simple evolution law is formulated as follows

$$d = \min\left\{\max_{\tau \leq t}\left\{\alpha\left(\frac{Y_{|\tau}}{Y_{\mathrm{c}}}\right)^{\beta}\right\}, 1\right\}\,, \tag{38}$$

where Y_{c} , α and β are parameters of the damage model. Notice that the history dependency in the previous equation (non-reversibility of the damage process) is inherited by the discretised system of equations. Regarding classical localisation issues related to damage models, we note that the lattice model is naturally nonlocal, the length of the beams being a length scale used as a regularisation parameter. Using shorter beams or higher order will provide material models that dissipate less energy when cracks propagate.

Variational form and discrete system of equations. Let us weigh the residual of the local equilibrium (28) by a kinematically admissible displacement field $\underline{u}^\star$, integrate over $\Omega^{(b)}$ and sum over $\mathcal{B}$:

$$\sum_{b\in\mathcal{B}} \int_{\Omega^{(b)}} \frac{\partial N^{(b)}}{\partial s^{(b)}} \, \underline{u}^{(b)^\star} \cdot \underline{n}^{(b)} \, ds^{(b)} + \sum_{b\in\mathcal{B}} \int_{\Omega^{(b)}} \underline{f} \cdot \underline{n}^{(b)} \, \underline{u}^{(b)^\star} . \underline{n}^{(b)} \, ds^{(b)} = 0 \,. \tag{39}$$

Integrating by part the summands of first term of the last equation, and taking into both the continuity of $\underline{u}^\star$ at any node of the lattice structure and the nodal equilibrium, one gets the variational form of the lattice problem

$$-\sum_{b\in\mathcal{B}} \int_{\Omega^{(b)}} N^{(b)} \frac{\partial \underline{u}^{(b)^\star}}{\partial s^{(b)}} \cdot \underline{n}^{(b)} \, ds^{(b)} + \sum_{b\in\mathcal{B}} \int_{\Omega^{(b)}} \underline{f} \cdot \underline{n}^{(b)} \, \underline{u}^{(b)^\star} . \underline{n}^{(b)} \, ds^{(b)} + \sum_{P\in\mathcal{P}^{\mathrm{F}}} \underline{N}_{\mathrm{d}|P} \cdot \underline{u}^\star_{|P} = 0 \,, \tag{40}$$

where we have additionally enforced the condition that test function $\underline{u}^\star$ vanishes at every node belonging to P^{u}. Last, by writing the piecewise linearity of the displacement field of bar $b \in \mathcal{B}$ in the form:

$$\underline{u}^{(b)}(s^{(b)}) = \underline{\underline{\Lambda}}^{(b)} \left(s^{(b)}\right) \underline{\underline{\tilde{\mathbf{A}}}}^{(b)^T} \underline{\mathbf{U}} \quad \text{with} \quad \underline{\underline{\Lambda}}^{(b)} \left(s^{(b)}\right) = \begin{pmatrix} 1-\tilde{s}^{(b)} & 0 & \tilde{s}^{(b)} & 0 \\ 0 & 1-\tilde{s}^{(b)} & 0 & \tilde{s}^{(b)} \end{pmatrix} \tag{41}$$

where $\tilde{s}^{(b)} \overset{\text{def}}{=} \frac{s^{(b)}}{\|\underline{P_i P_j}\|}$ and $\underline{\underline{\tilde{\mathbf{A}}}}^{(b)}$ the assembly operator such that $\underline{\mathbf{U}}^{(b)} = \underline{\underline{\tilde{\mathbf{A}}}}^{(b)^T} \underline{\mathbf{U}}$ with $\underline{\mathbf{U}}^{(b)} = \left(\underline{u}_{|P_i} \cdot \underline{e}_x \quad \underline{u}_{|P_i} \cdot \underline{e}_y \quad \underline{u}_{|P_j} \cdot \underline{e}_x \quad \underline{u}_{|P_j} \cdot \underline{e}_y\right)^T$ the vector of nodal values of the restriction of the displacement to bar b, $\underline{\mathbf{U}}$ the global vector of nodal displacement values and P_i and P_j $(i<j)$ the two extremities of bar b, we get the expression of the semi-discrete problem at time $t \in \mathcal{T}$:

$$\begin{aligned} &\forall\, \underline{\mathbf{U}}^\star \in \mathbb{R}^{\tilde{n}_{\mathrm{u}}} \text{ such that } \left(\underline{\underline{\hat{\mathbf{A}}}}^{(P)^T} \underline{\mathbf{U}}^\star = 0, \forall\, P \in \mathcal{P}^{\mathrm{u}}\right), \quad \underline{\mathbf{U}}^{\star T} \left(\underline{\tilde{\mathbf{F}}}_{\mathbf{int}} \left((\underline{\mathbf{U}}(\tau))_{\tau\in[0,t]}\right) + \underline{\tilde{\mathbf{F}}}_{\mathrm{ext}}(t)\right) = 0 \\ &\text{with} \quad \begin{cases} \underline{\tilde{\mathbf{F}}}_{\mathbf{int}} \left((\underline{\mathbf{U}}(\tau))_{\tau\in[0,t]}\right) = -\sum_{b\in\mathcal{B}} \underline{\underline{\tilde{\mathbf{A}}}}^{(b)} \int_{\Omega^{(b)}} \frac{\partial \underline{\underline{\Lambda}}^{(b)^T}}{\partial s^{(b)}} \underline{n}^{(b)} N^{(b)} \left((\underline{\mathbf{U}}(\tau))_{\tau\in[0,t]}\right) ds^{(b)} \\ \underline{\tilde{\mathbf{F}}}_{\mathrm{ext}}(t) = \sum_{b\in\mathcal{B}} \underline{\underline{\tilde{\mathbf{A}}}}^{(b)} \int_{\Omega^{(b)}} \underline{\underline{\Lambda}}^{(b)^T} \underline{n}^{(b)} \, \underline{n}^{(b)^T} \underline{f}(t) \, ds^{(b)} + \sum_{P\in\mathcal{P}^{\mathrm{F}}} \underline{\underline{\hat{\mathbf{A}}}}^{(P)} \underline{N}_{\mathrm{d}|P}(t) \,, \end{cases} \end{aligned} \tag{42}$$

where $\underline{\underline{\hat{\mathbf{A}}}}^{(P)}$ an assembly defined such that at any node $P \in \mathcal{P}$, we have $\left(\underline{u}_{|P} \cdot \underline{e}_x \quad \underline{u}_{|P} \cdot \underline{e}_y\right)^T = \underline{\underline{\hat{\mathbf{A}}}}^{(P)^T} \underline{\mathbf{U}}$. Variational principle (42) needs to be complemented by the Dirichlet boundary conditions (32), and the resulting problem can be parametrised and discretised in time to obtain the "truth" problem (2).

3.4.3 Parametrised problem of fracture

The leftmost part of the structure is fixed (null Dirichlet boundary conditions) while a prescribed displacement, which puts the structure in tension, is gradually applied on the rightmost part. The direction of the load is controlled by an input parameter $\theta(\equiv \underline{\boldsymbol{\mu}}) \in \mathbb{R}$ which ranges in $\mathcal{P} = [15°, 45°]$. An initial crack (notch) is defined at the top centre of the structure by initially setting the damage fields of the corresponding bars to 1, as illustrated in figure 2. As the load is progressively applied to the damageable structure, the crack propagates. The time evolution of the crack propagation problem is discretised using 10 homogeneous load steps. The lattice structure is built up using 1071 nodes linked by 4070 bars. The Young's moduli, bar sections and and lengths of the horizontal and vertical bars of the regular lattice are set to unity. The body force field is null.

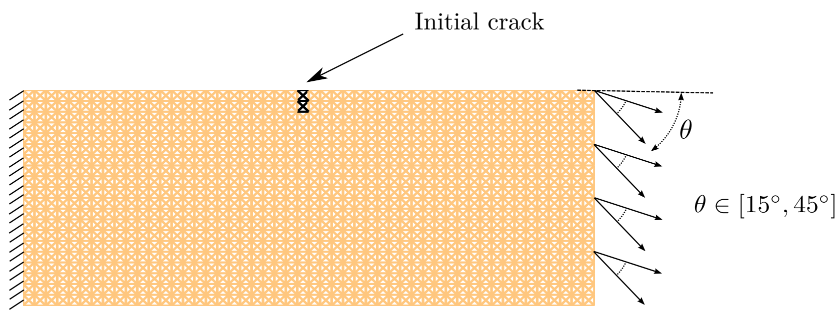


Figure 2: Definition of the nonlinear lattice problem used for the numerical experiments of this paper. The loss of stiffness of each bar while increasing local strain is described by a damage model. The direction of the prescribed displacement on the right-hand edge of the rectangular lattice structure is a parameter of the model. The aim is to predict the propagation of the damage onset (initially damaged bars represented in black) for any angle of the prescribed load.

Our goal is to predict the damage state in the lattice for any arbitrary angle $\theta \in \mathcal{P}$ without solving the "truth" model. The solution will be looked for in a space generated by a spectral analysis of precomputed solutions (Snapshot POD) corresponding to a number n_μ of particular parameters distributed homogeneously in the unidimensional parameter domain and including the two extrema values of θ, 15° and 45°.

Results displayed in figure 3 illustrate the behaviour of the reduced order modelling approach for $n_\mu = 2$. The normalised truncation error ν_{snap} of the snapshot POD as given in equation (20) is arbitrarily set to 10^{-2} (see figure 4), which leads to the definition of a reduced space of dimension.

It is noticed that each load angle θ leads to a crack/damage zone propagating approximately orthogonally to the load direction, as is commonly observed in fracture mechanics. Consequently, each and every load angle leads to a different damage pattern which cannot be well represented by a linear combination of the cracks obtained for a limited number of snapshot solutions (figure 4, bottom). In fact, the solution to parametric problems involving the evolution of topological changes cannot, in general, be obtained efficiently using a method based on the separation of variables (unless one manages to map the physical space to a reference space were correlation in the data can be retrieved [26]). One systematic way to circumvent the problem would be to enrich the snapshot "online" [25, 56], but this leads to difficulties related to the cost of evaluating the projection error.

Despite these apparent difficulties, the topological changes are localised in space. In the regions that are far away from the crack, the solution is indeed well approximated by a linear combination of the pre-computed basis vectors. Consequently, a classical model reduction can still be performed but only over selected regions of the domain. The following section presents a possible strategy to implement this idea based on a domain decomposition method where the subdomains are selectively and independently reduced, based on a criterion described in section 4.3.

Remark: *The initial crack is meant to provide a stress concentration zone from which fracture will initiates. We emphasise here that this is an idealisation of a general situation in realistic engineering components. Cracks initiates from joints, supports, free edges (large shear stresses due to a mismatch between elastic properties in composite laminates for instance), non-smooth parts of the boundary of the component (corner), or from interior regions which are subjected to extreme stress concentration under particular external loading conditions. Therefore, the regions of potential initiations are not arbitrary*

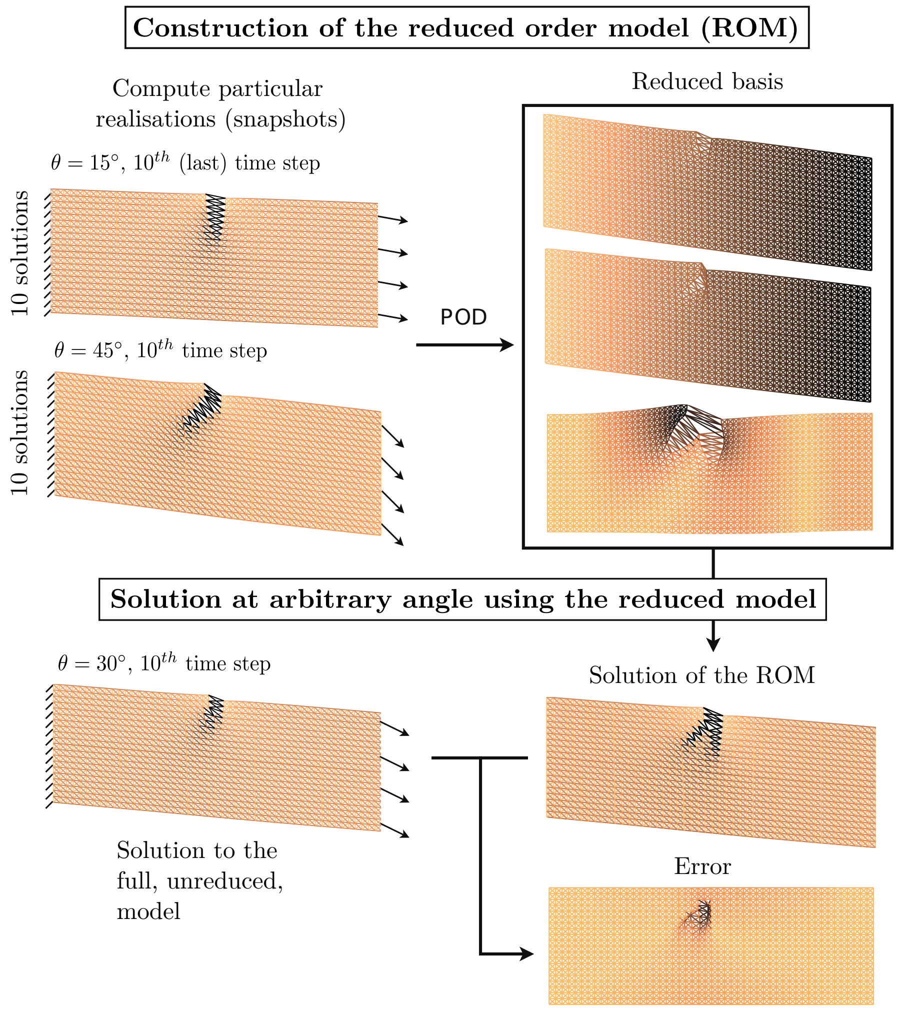


Figure 3: Schematic representation of the Snapshot POD model order reduction technique for the proposed parametrised problem of fracture. The "truth" time evolution of the problem is computed "offline" for a certain number of values of the parameter. A reduced space is generated by performing a spectral analysis of this snapshot (POD). In the "online" phase, the "truth" problem is solved approximately by making use of a Galerkin projection of the governing equations in this reduced space, for any parameter value of interest. In the case of fracture mechanics, the projection error localises in the "process zone" surrounding the crack. Far away from it, a reduced space of small dimension associated to a relatively coarse exploration of the parameter domain is sufficient to capture the solution with a high level of accuracy. The darkest bars correspond to a completely damaged state of the material, while the lightest bars are undamaged.

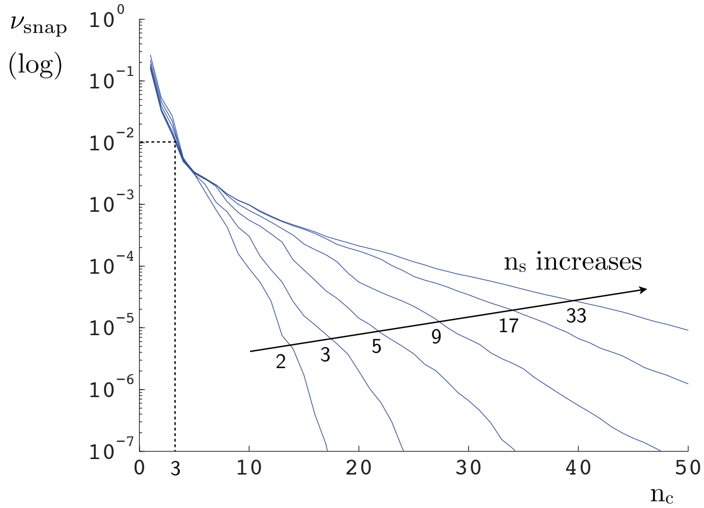


Figure 4: Convergence of the normalised POD error indicator as a function of the order of truncation, for increasing size of the number of parameter values used to build the snapshot. The lack of correlation due to the crack propagation introduces a local error of projection, which appears here as a decrease in the convergence rate of the spectral decomposition below a certain value of the snapshot POD error indicator. This threshold is relatively low due to the global nature of the metric used to evaluate the accuracy of the projection. The numbers displayed on the graph are the number of load angles used to create the snapshot.

for a given parametric problem. In the particular example treated in this paper, fracture propagates from the notch which was introduced in the geometry. However, in all the following developments, we do not make use of the knowledge of the position of this initial defect, which emulates the existence of a priori unknown zones of stress concentration in the structure.

4 Partitioned model order reduction approach

4.1 Principle of the primal Schur-based domain decomposition method

Schur-based non-overlapping domain decomposition methods (see a review in [57]) are dedicated to the solution of large scale linear systems. In our case, we use the primal Schur-based domain decomposition (balancing domain decomposition (BDD) [37, 38, 39])) to calculate successive Newton iterates for the solution of the reference nonlinear time-dependant problem. Schur-based domain decomposition methods propose to condense the linearised balance equations on the interface degrees of freedom (degrees of freedom that are shared by at least two subdomains), by eliminating the interior degrees of freedom. The resulting interface problem is solved by an iterative solver, usually by a preconditioned Krylov subspace algorithm, which is particularly well-suited to parallel computing. The condensation realises a first step of preconditioning, but the derivation of a preconditioner for the condensed interface problem is a key point to obtain an efficient and scalable domain decomposition method.

Let us now give an overview of the domain decomposition method for the solution of the "truth" problems corresponding to parameters $\mu \in \mathcal{P}^{\text{s}}$ (*i.e.:* the snapshot). Domain Ω is split into non-overlapping subdomains $(\Omega^{(e)})_{e\in[\![1,n_{\text{e}}]\!]}$ such that $\bigcup_{e\in[\![1,n_{\text{e}}]\!]}\Omega^{(e)} = \Omega$, as illustrated in Figure 5. Each bar of the lattice structure belongs to one and only one subdomain. Nodes that are shared by two adjacent subdomains are interface nodes. We later refer to the set of subdomain indexes $[\![1, n_{\text{e}}]\!]$ as $\mathcal{E}$.

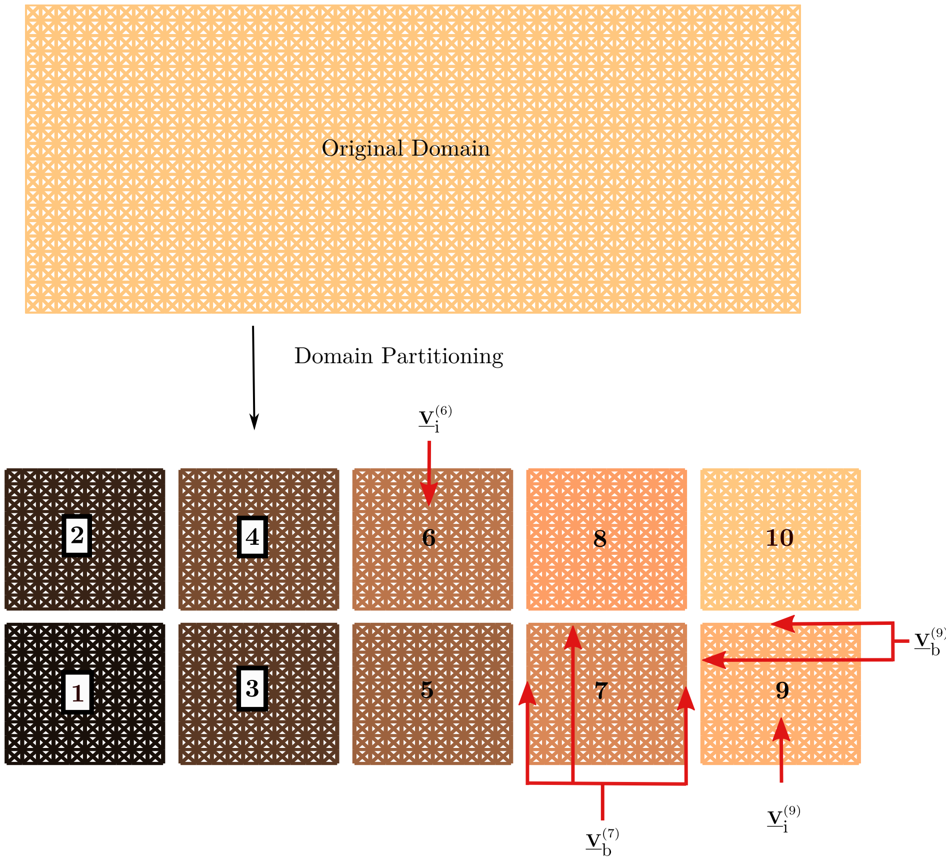


Figure 5: Subdivision of the domain of interest into 10 non-overlapping subdomains. $\underline{\mathbf{\Delta U}}_{\mathrm{i}}^{(e)}$ is the restriction of a vector $\underline{\mathbf{\Delta U}}^{(e)}$ of nodal values of subdomain e to the internal degrees of freedom of the subdomain, while $\underline{\mathbf{\Delta U}}_{\mathrm{b}}^{(e)}$ corresponds to the interface nodal values. The superscript between brackets indicates the number of the subdomain.

Let $\underline{\mathbf{U}}^{(e)}(t;\underline{\boldsymbol{\mu}}) \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}}$ be the vector of nodal displacements of $\Omega^{(e)}$, which is looked for at an arbitrary point $(t,\underline{\boldsymbol{\mu}}) \in \tilde{\mathcal{P}}$ of the time-parameter domain. Each subdomain carries its own nodal unknowns for the interface nodes, which means that, for now, the corresponding kinematic is allowed to jump at the interface.

The local equilibrium of subdomain $\Omega^{(e)}$ is expressed in an algebraic form as follows:

$$\underline{\mathbf{F}}_{\mathbf{int}}^{(e)}\left(\underline{\mathbf{U}}^{(e)}(t;\underline{\boldsymbol{\mu}}), \left(\underline{\mathbf{U}}^{(e)}(\tau;\underline{\boldsymbol{\mu}})\right)_{\tau\in\mathcal{T}^h,\,\tau<t};\underline{\boldsymbol{\mu}}\right) + \underline{\mathbf{F}}_{\mathbf{ext}}^{(e)}(t;\underline{\boldsymbol{\mu}}) = \underline{\underline{\mathbf{t}}}^{(e)\,T}\underline{\boldsymbol{\lambda}}^{(e)}\,, \tag{43}$$

with $\underline{\boldsymbol{\lambda}}^{(e)} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}$ a vector of reaction forces from adjacent subdomains and $\underline{\underline{\mathbf{t}}}^{(e)} \in \{0,1\}^{n_{\mathrm{b}}^{(e)}} \times \{0,1\}^{n_{\mathrm{u}}^{(e)}}$ a trace operator which extracts the entries of vector of local nodal values corresponding to the interface nodes (*i.e.* an output vector $\underline{\mathbf{X}}_{\mathbf{b}}^{(e)} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}$ defined by $\underline{\mathbf{X}}_{\mathbf{b}}^{(e)} = \underline{\underline{\mathbf{t}}}^{(e)}\underline{\mathbf{X}}^{(e)}$, with $\underline{\mathbf{X}}^{(e)} \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}}$ an arbitrary vector of local nodal values). The reaction forces must satisfy the following global interface equilibrium property:

$$\sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)}\underline{\boldsymbol{\lambda}}^{(e)} = \underline{\mathbf{0}} \tag{44}$$

where $\{\underline{\underline{\mathbf{A}}}^{(e)} \in \{0,1\}^{n_{\mathrm{b}}} \times \{0,1\}^{n_{\mathrm{b}}^{(e)}} \,|\, e \in \mathcal{E}\}$ is a set of assembly operators, with n_{b} the number of interface equilibrium conditions, which is equal to the number of interface nodes. The set of subproblems is closed by the condition of equality of nodal displacements at an interface between two-subdomains (*i.e.:* kinematic continuity in a continuous setting), which reads

$$\sum_{e\in\mathcal{E}} \underline{\underline{\bar{\mathbf{A}}}}^{(e)} \underline{\underline{\mathbf{t}}}^{(e)} \underline{\mathbf{U}}^{(e)} = \underline{\mathbf{0}} \tag{45}$$

where $\{\underline{\underline{\bar{\mathbf{A}}}}^{(e)} \in \{0,-1,1\}^{\bar{n}_\mathrm{b}} \times \{0,-1,1\}^{n_\mathrm{b}^{(e)}} \,|\, e \in \mathcal{E}\}$, are signed boolean operators, with $\bar{n}_\mathrm{b}$ the number of independent interface kinematic constraints. (see [57] for more details about the definition of properties of the assembly and trace operators).

In order to give expressions that are closer to the actual implementation of the method, we perform a linearisation of local problems (43) at iteration $i+1$ of the Newton algorithm. We look for iterates $\{(\underline{\mathbf{U}}^{(e),i+1}, \underline{\boldsymbol{\lambda}}^{(e),i+1}) \in \mathbb{R}^{n_\mathrm{u}^{(e)}} \times \mathbb{R}^{n_\mathrm{b}^{(e)}} \,|\, e \in \mathcal{E}\}$ of the local displacements and reaction forces satisfying both the local linearised systems

$$\underline{\underline{\mathbf{K}}}^{(e),i} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e),i+1} = -\underline{\mathbf{R}}^{(e)} + \underline{\underline{\mathbf{t}}}^{(e)^T} \underline{\boldsymbol{\lambda}}^{(e),i+1} \quad ,\forall\, e \in \mathcal{E}\,, \tag{46}$$

and the global interface conditions (44) and (45). In the previous equation, the local tangent stiffness is $\underline{\underline{\mathbf{K}}}^{(e),i} \stackrel{\text{def}}{=} \left.\frac{\partial \underline{\mathbf{F}}^{(e)}_{\mathbf{int}}(\underline{\mathbf{U}}^{(e)})}{\partial \underline{\mathbf{U}}^{(e)}}\right|_{\underline{\mathbf{U}}^{(e)}=\underline{\mathbf{U}}^{(e),i}}$, the residual vector is $\underline{\mathbf{R}}^{(e),i} \stackrel{\text{def}}{=} \underline{\mathbf{F}}^{(e)}_{\mathbf{int}}(\underline{\mathbf{U}}^{(e),i}) + \underline{\mathbf{F}}^{(e)}_{\mathbf{ext}}$, and the increment of displacement is defined by $\underline{\boldsymbol{\Delta}\mathbf{U}}^{(e),i+1} = \underline{\mathbf{U}}^{(e),i+1} - \underline{\mathbf{U}}^{(e),i}$. In the following, we will drop superscripts i and $i+1$.

If we introduce the local operator $\underline{\underline{\mathbf{E}}}^{(e)} \in \{0,1\}^{n_\mathrm{i}^{(e)}} \times \{0,1\}^{n_\mathrm{u}^{(e)}}$ ($n_\mathbf{i}^{(e)} \stackrel{\text{def}}{=} n_\mathbf{u}^{(e)} - n_\mathrm{b}^{(e)}$ is the number of interior degrees of freedom of e) such that the output vector $\underline{\mathbf{X}}_\mathbf{i}^{(e)} = \underline{\underline{\mathbf{E}}}^{(e)} \underline{\mathbf{X}}^{(e)}$, with $\underline{\mathbf{X}}^{(e)} \in \mathbb{R}^{n_\mathrm{u}^{(e)}}$ arbitrary, is the restriction of $\underline{\mathbf{X}}^{(e)}$ to the interior nodes of subdomain $\Omega^{(e)}$, for any $e \in \mathcal{E}$, we can recast the local systems (46) as follows:

$$\begin{bmatrix} \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ii}} & \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ib}} \\ \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi}} & \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bb}} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{i}} \\ \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} -\underline{\mathbf{R}}^{(e)}_{\mathbf{i}} \\ -\underline{\mathbf{R}}^{(e)}_{\mathbf{b}} + \underline{\boldsymbol{\lambda}}^{(e)} \end{bmatrix} \quad ,\forall\, e \in \mathcal{E}\,, \tag{47}$$

where $\underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{i}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{E}}}^{(e)} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}$, $\underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{b}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{t}}}^{(e)} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}$, $\underline{\mathbf{R}}^{(e)}_{\mathbf{i}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{E}}}^{(e)} \underline{\mathbf{R}}^{(e)}$, $\underline{\mathbf{R}}^{(e)}_{\mathbf{b}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{t}}}^{(e)} \underline{\mathbf{R}}^{(e)}$, $\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ii}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{E}}}^{(e)} \underline{\underline{\mathbf{K}}}^{(e)} \underline{\underline{\mathbf{E}}}^{(e)^T}$, $\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ib}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{E}}}^{(e)} \underline{\underline{\mathbf{K}}}^{(e)} \underline{\underline{\mathbf{t}}}^{(e)^T}$, $\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{t}}}^{(e)} \underline{\underline{\mathbf{K}}}^{(e)} \underline{\underline{\mathbf{E}}}^{(e)^T}$ and $\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bb}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{t}}}^{(e)} \underline{\underline{\mathbf{K}}}^{(e)} \underline{\underline{\mathbf{t}}}^{(e)^T}$. The interior degrees of freedom $\underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{i}}$ are eliminated from local systems (47) by static condensation, which is obtained by writing

$$\underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{i}} = \underline{\underline{\mathbf{K}}}^{(e)^{-1}}_{\mathbf{ii}} \left(-\underline{\mathbf{R}}^{(e)}_{\mathbf{i}} - \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ib}} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{b}} \right)\,, \tag{48}$$

where $\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ii}}$ is assumed to be invertible. The condensed local problem is obtained by substitution of expression (48) in the second line of (47):

$$\underline{\underline{\mathbf{S}}}^{(e)}_{\mathbf{p}} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{b}} = \underline{\mathbf{F}}^{(e)}_{\mathbf{c}} + \underline{\boldsymbol{\lambda}}^{(e)}\,, \tag{49}$$

where the primal Schur complement $\underline{\underline{\mathbf{S}}}^{(e)}_{\mathbf{p}}$ is defined by $\underline{\underline{\mathbf{S}}}^{(e)}_{\mathbf{p}} = \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bb}} - \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi}} \underline{\underline{\mathbf{K}}}^{(e)^{-1}}_{\mathbf{ii}} \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ib}}$, and the condensed forces $\underline{\mathbf{F}}^{(e)}_{\mathbf{c}}$ are defined by $\underline{\mathbf{F}}^{(e)}_{\mathbf{c}} = -\underline{\mathbf{R}}^{(e)}_{\mathbf{b}} - \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi}} \underline{\underline{\mathbf{K}}}^{(e)^{-1}}_{\mathbf{ii}} (-\underline{\mathbf{R}}^{(e)}_{\mathbf{i}})$.

We now apply the primal domain decomposition methodology by enforcing the interface kinematic continuity (45) in a strong sense, which is done by writing that the local trace of the unknown displacement vectors $\{\underline{\mathbf{U}}^{(e)} \,|\, e \in \mathcal{E}\}$ are obtained by extraction from a global interface vector $\underline{\mathbf{U}}_\mathbf{b} \in \mathbb{R}^{n_\mathrm{b}}$

$$\underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)}_{\mathbf{b}} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{t}}}^{(e)} \underline{\boldsymbol{\Delta}\mathbf{U}}^{(e)} = \underline{\underline{\mathbf{A}}}^{(e)^T} \underline{\boldsymbol{\Delta}\mathbf{U}}_{\mathbf{b}} \quad ,\forall\, e \in \mathcal{E}\,, \tag{50}$$

which implies the fulfilment of (45) provided that the previous Newton iterate of the underlying displacement field is continuous.

A global assembled interface problem is obtained when left multiplying each of the local condensed systems (equation (49)) by assembly operators $\{\underline{\underline{\mathbf{A}}}^{(e)} \,|\, e \in \mathcal{E}\}$ and summing up over all subdomains, which reads

$$\sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}_{\mathbf{p}}^{(e)} \underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{c}}^{(e)} + \underbrace{\sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\boldsymbol{\lambda}}^{(e)}}_{=\underline{\mathbf{0}} \quad \text{from} \quad (44)} \tag{51}$$

$$\Longleftrightarrow \sum_{e\in\mathcal{E}} \left(\underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}_{\mathbf{p}}^{(e)} \underline{\underline{\mathbf{A}}}^{(e)^T} \right) \underline{\mathbf{\Delta U}}_{\mathbf{b}} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{c}}^{(e)} \,. \tag{52}$$

In a compact form, we look for an interface vector $\underline{\mathbf{\Delta U}}_{\mathbf{b}} \in \mathbb{R}^{n_{\mathrm{b}}}$ satisfying

$$\underline{\underline{\mathbf{S}}}_{\mathbf{p}} \underline{\mathbf{\Delta U}}_{\mathbf{b}} = \underline{\mathbf{F}}_{\mathbf{c}} \quad \text{with} \quad \begin{cases} \underline{\underline{\mathbf{S}}}_{\mathbf{p}} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}_{\mathbf{p}}^{(e)} \underline{\underline{\mathbf{A}}}^{(e)^T} \\ \underline{\mathbf{F}}_{\mathbf{c}} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{c}}^{(e)} \,. \end{cases} \tag{53}$$

Interface problem (53) can be solved iteratively in parallel using a Krylov-subspace method such as the conjugate gradient in a symmetric case or GMRes [58](or BiCGStab [59]) in a non-symmetric case. In this framework, the global Schur complement need not be assembled. Instead, whenever it is needed in a matrix/vector multiplication, the multiplication is performed locally on each subdomain using the local Schur complements. The outcome of these local multiplications is then assembled:

$$\forall \, \underline{\mathbf{X}}_{\mathbf{b}} \in \mathbb{R}^{n_{\mathrm{b}}}, \qquad \underline{\underline{\mathbf{S}}}_{\mathbf{p}} \underline{\mathbf{X}}_{\mathbf{b}} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}_{\mathbf{p}}^{(e)} \underbrace{\underline{\underline{\mathbf{A}}}^{(e)^T} \underline{\mathbf{X}}_{\mathbf{b}}}_{=\underline{\mathbf{X}}_{\mathbf{b}}^{(e)}} \,. \tag{54}$$

The local inversions involved in the computation of the local Schur complements are performed directly (using a Cholesky factorisation for instance). Using this method it is possible to perform the matrix/vector multiplications (computationally the most demanding part of a Krylov-subspace method) in parallel. In a similar way, the dot products involved in the iterative algorithm can be performed in parallel.

$$\forall \, \underline{\mathbf{X}}_{\mathbf{b}} \in \mathbb{R}^{n_{\mathrm{b}}}, \qquad \underline{\mathbf{X}}_{\mathbf{b}}^{T} \underline{\mathbf{X}}_{\mathbf{b}} = \sum_{e\in\mathcal{E}} \underline{\mathbf{X}}_{\mathbf{b}}^{(e)^T} \underline{\underline{\mathbf{D}}}^{(e)} \underline{\mathbf{X}}_{\mathbf{b}}^{(e)} \,, \tag{55}$$

where $\{\underline{\underline{\mathbf{D}}}^{(e)} \,|\, e \in \rceil\}$ are diagonal matrices whose natural entries depend on the geometric multiplicity of the interface nodes.

4.2 Formulation of reduced order modelling in the domain decomposition framework

4.2.1 Local snapshot POD reduced spaces

We propose to use POD-based model order reduction on the interior degrees of freedom of each subdomain. We assume that a snapshot $\{\underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}) \,|\, (t,\underline{\boldsymbol{\mu}}) \in \tilde{\mathcal{P}}^{\mathrm{s}}\}$ is available. This snapshot has been computed by making use the domain decomposition preconditioner described previously. Local POD spatial bases $\left(\underline{\mathbf{C}}_{\mathbf{i},i}^{(e)}\right)_{i\in[\![1,n_{\mathrm{c}}^{(e)}]\!]}$ of dimensions $n_{\mathrm{c}}^{(e)}$ are now computed for the interior degrees of freedom of each subdomain $e \in [\![1, n_{\mathrm{e}}]\!]$ as described in section 3. Accordingly, the normalised truncation error of the local snapshot POD transforms are defined as follows:

$$\left(\nu_{\mathrm{snap}}^{(e)}\right)^2 = \frac{\sum_{\underline{\boldsymbol{\mu}}\in\mathcal{P}^{\mathrm{s}}} \sum_{t\in\mathcal{T}^{\mathrm{h}}} \left\| \underline{\mathbf{U}}_{\mathbf{i}}^{(e)}(t;\underline{\boldsymbol{\mu}}) - \sum_{j=1}^{n_{\mathrm{c}}^{(e)}} \left(\underline{\mathbf{C}}_{\mathbf{i},j}^{(e)^T} \underline{\mathbf{U}}_{\mathbf{i}}^{(e)}(t;\underline{\boldsymbol{\mu}}) \right) \underline{\mathbf{C}}_{\mathbf{i},j}^{(e)} \right\|_2^2}{\sum_{t\in\mathcal{T}^{\mathrm{h}}} \sum_{\underline{\boldsymbol{\mu}}\in\mathcal{P}^{\mathrm{s}}} \|\underline{\mathbf{U}}_{\mathbf{i}}^{(e)}(t;\underline{\boldsymbol{\mu}})\|_2^2} \,, \quad \forall e \in \mathcal{E} \,, \tag{56}$$

where $\underline{\mathbf{U}}_{\mathbf{i}}^{(e)} \overset{\text{def}}{=} \underline{\underline{\mathbf{E}}}^{(e)}\underline{\mathbf{U}}^{(e)}$ for any $e \in \mathcal{E}$. Let us define the local operators $\{\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} \,|\, e \in \mathcal{E}\}$ whose columns are the local POD basis vectors of subdomain e.

4.2.2 Local projection

In the "online" stage, we look for the interior degrees of freedom corresponding to an arbitrary point of the time-parameter domain $\tilde{\mathcal{P}}$ in the local reduced spaces. The reduction technique is here directly described for the linearised problem for the sake of concision, but one could equivalently start from the nonlinear partitioned problem (43), introduce the a local reduced basis approximation and linearise the result.

The kinematic interior approximation for the linearised problem reads:

$$\begin{bmatrix} \underline{\mathbf{\Delta U}}_{\mathbf{i}}^{(e)} \\ \underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} \end{bmatrix} = \begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)}\underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)} \\ \underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} \end{bmatrix}, \quad \forall e \in \mathcal{E}, \tag{57}$$

where $\underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)}$ is a vector of local reduced state variables. Therefore, the local linearised system of equation (47) corresponding to an arbitrary subdomain $e \in \mathcal{E}$ now reads

$$\begin{bmatrix} \underline{\underline{\mathbf{K}}}_{\mathbf{ii}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{ib}}^{(e)} \\ \underline{\underline{\mathbf{K}}}_{\mathbf{bi}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{bb}}^{(e)} \end{bmatrix} \begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)}\underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)} \\ \underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} \end{bmatrix} = \begin{bmatrix} -\underline{\mathbf{R}}_{\mathbf{i}}^{(e)} \\ -\underline{\mathbf{R}}_{\mathbf{b}}^{(e)} + \underline{\boldsymbol{\lambda}}^{(e)} \end{bmatrix}. \tag{58}$$

This is a set of $n_{\mathrm{i}}^{(e)} + n_{\mathrm{b}}^{(e)}$ equations in $n_{\mathrm{c}}^{(e)} + n_{\mathrm{b}}^{(e)}$ unknowns. As we expect that $n_{\mathrm{i}}^{(e)} + n_{\mathrm{b}}^{(e)} \gg n_{\mathrm{c}}^{(e)} + n_{\mathrm{b}}^{(e)}$, this system is overdetermined in general. Consistently with the developments proposed in section 3.1, we perform a Galerkin projection: the residual of local system (58) is required to be orthogonal to the local reduced space, which reads

$$\begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} & \underline{\underline{\mathbf{0}}} \\ \underline{\underline{\mathbf{0}}} & \underline{\underline{\mathbf{I}}}_{d,\mathbb{R}^{n_{\mathrm{i}}^{(e)}}} \end{bmatrix}^T \left(\begin{bmatrix} -\underline{\mathbf{R}}_{\mathbf{i}}^{(e)} \\ -\underline{\mathbf{R}}_{\mathbf{b}}^{(e)} + \underline{\boldsymbol{\lambda}}^{(e)} \end{bmatrix} - \begin{bmatrix} \underline{\underline{\mathbf{K}}}_{\mathbf{ii}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{ib}}^{(e)} \\ \underline{\underline{\mathbf{K}}}_{\mathbf{bi}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{bb}}^{(e)} \end{bmatrix} \begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)}\underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)} \\ \underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} \end{bmatrix} \right) = \underline{\mathbf{0}}. \tag{59}$$

We end up with the following linear, square and symmetric system for the expression of the reduced local equilibria:

$$\left(\underline{\mathbf{F}}_{\mathbf{r}}^{(e)} + \begin{bmatrix} \underline{\mathbf{0}} \\ \underline{\boldsymbol{\lambda}}^{(e)} \end{bmatrix} \right) - \underline{\underline{\mathbf{K}}}_{\mathbf{r}}^{(e)} \begin{bmatrix} \underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)} \\ \underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} \end{bmatrix} = \underline{\mathbf{0}} \quad \text{where} \quad \begin{cases} \underline{\underline{\mathbf{K}}}_{\mathbf{r}}^{(e)} \overset{\text{def}}{=} \begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)T}\underline{\underline{\mathbf{K}}}_{\mathbf{ii}}^{(e)}\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} & \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)T}\underline{\underline{\mathbf{K}}}_{\mathbf{ib}}^{(e)} \\ \underline{\underline{\mathbf{K}}}_{\mathbf{bi}}^{(e)}\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{bb}}^{(e)} \end{bmatrix} \\ \underline{\mathbf{F}}_{\mathbf{r}}^{(e)} \overset{\text{def}}{=} \begin{bmatrix} -\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)T}\underline{\mathbf{R}}_{\mathbf{i}}^{(e)} \\ -\underline{\mathbf{R}}_{\mathbf{b}}^{(e)} \end{bmatrix}, \end{cases} \tag{60}$$

4.2.3 Condensed interface problem

Similarly as described in section 4.1, local systems (60) are condensed on the interface degrees of freedom, and are formally assembled. To do so, the reduced state variables $\underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)}$ are eliminated using the identity

$$\underline{\mathbf{\Delta}\boldsymbol{\alpha}}_{\mathbf{i}}^{(e)} = \underline{\underline{\mathbf{K}}}_{\mathbf{ii,r}}^{(e)\,-1} \left(-\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)T}\underline{\mathbf{R}}_{\mathbf{i}}^{(e)} - \underline{\underline{\mathbf{K}}}_{\mathbf{ib,r}}^{(e)}\underline{\mathbf{\Delta U}}_{\mathbf{b}}^{(e)} \right), \tag{61}$$

where $\underline{\underline{\mathbf{K}}}_{\mathbf{ii,r}} \overset{\text{def}}{=} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)T}\underline{\underline{\mathbf{K}}}_{\mathbf{ii}}^{(e)}\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)}$ is assumed to be invertible and $\underline{\underline{\mathbf{K}}}_{\mathbf{ib,r}} \overset{\text{def}}{=} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)T}\underline{\underline{\mathbf{K}}}_{\mathbf{ib}}^{(e)}$. By making use of interface kinematic and equilibrium conditions, which are not unchanged in our reduced order modelling approach, the assembled condensed reduced system reads:

$$\text{Find } \underline{\mathbf{\Delta U}}_{\mathbf{b}} \in \mathbb{R}^{n_{\mathrm{b}}} \text{ such that } \quad \underline{\underline{\mathbf{S}}}_{\mathbf{p,r}}\underline{\mathbf{\Delta U}}_{\mathbf{b}} = \underline{\mathbf{F}}_{\mathbf{c,r}} \quad \text{with} \quad \begin{cases} \underline{\underline{\mathbf{S}}}_{\mathbf{p,r}} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)}\underline{\underline{\mathbf{S}}}_{\mathbf{p,r}}^{(e)}\underline{\underline{\mathbf{A}}}^{(e)T} \\ \underline{\mathbf{F}}_{\mathbf{c,r}} = \sum_{e\in\mathcal{E}} \underline{\underline{\mathbf{A}}}^{(e)}\underline{\mathbf{F}}_{\mathbf{c,r}}^{(e)}, \end{cases} \tag{62}$$

with the expression of the local condensed operators $\underline{\underline{\mathbf{S}}}^{(e)}_{\mathbf{p,r}} \overset{\text{def}}{=} \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bb}} - \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi,r}} {\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ii,r}}}^{-1} \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ib,r}}$, the local condensed forces $\underline{\mathbf{F}}^{(e)}_{\mathbf{c,r}} \overset{\text{def}}{=} -\underline{\mathbf{R}}^{(e)}_{\mathbf{b}} - \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi,r}} {\underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{ii,r}}}^{-1} (-{\underline{\underline{\mathbf{C}}}^{(e)}_{\mathbf{i}}}^{T} \underline{\mathbf{R}}^{(e)}_{\mathbf{i}})$ and $\underline{\underline{\mathbf{K}}}_{\mathbf{bi,r}} \overset{\text{def}}{=} \underline{\underline{\mathbf{K}}}^{(e)}_{\mathbf{bi}} \underline{\underline{\mathbf{C}}}^{(e)}_{\mathbf{i}}$, for any $e \in \mathcal{E}$. Problem (62) can be solved in parallel (if the snapshot data is distributed in memory) using a Krylov algorithm, as described in section 4.1.

We can now go one step further and choose not to reduce the local problems corresponding to some of the subdomains. Indeed, if localised non-linearities arise (damage in our case), the local reduction based on the separation of variables might be inefficient: a prohibitively large number of spatial basis vectors might be required to obtained the desired accuracy over the whole parameter domain (recall the results of section 3.4.3). This particular issue will be addressed in section 4.3. So far, we will assume that the subdomains are divided into two complementary sets $\mathcal{E}^{\text{red}} \cup \mathcal{E}^{\text{nred}} = \mathcal{E}$, where $\mathcal{E}^{\text{red}}$ is a set of subdomains for which reduction is numerically efficient, while $\mathcal{E}^{\text{nred}}$ is the complementary set of subdomains, for which a direct solution to the corresponding local problem is preferred. The resulting hybrid condensed reduced problem consists in finding $\underline{\mathbf{\Delta U}}_{\mathbf{b}} \in \mathbb{R}^{n_{\text{b}}}$ satisfying

$$\underline{\underline{\mathbf{S}}}_{\mathbf{p,hr}} \underline{\mathbf{\Delta U}}_{\mathbf{b}} = \underline{\mathbf{F}}_{\mathbf{c,hr}} \text{ with } \begin{cases} \underline{\underline{\mathbf{S}}}_{\mathbf{p,hr}} = \sum_{e \in \mathcal{E}^{\text{red}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}^{(e)}_{\mathbf{p,r}} {\underline{\underline{\mathbf{A}}}^{(e)}}^{T} + \sum_{e \in \mathcal{E}^{\text{nred}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}^{(e)}_{\mathbf{p}} {\underline{\underline{\mathbf{A}}}^{(e)}}^{T} \\ \underline{\mathbf{F}}_{\mathbf{c,hr}} = \sum_{e \in \mathcal{E}^{\text{red}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}^{(e)}_{\mathbf{c,r}} + \sum_{e \in \mathcal{E}^{\text{nred}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}^{(e)}_{\mathbf{c}} \end{cases} . \qquad (63)$$

4.3 Local error estimation by Cross-Validation

4.3.1 Principle

The partitioned projection approach described in section 4.2 allows us to construct reduced spaces that are independent for each subdomain. We propose here a simple scheme in order to (i) determine independently the dimension of the local reduced space that is necessary to achieve a predefined accuracy for the solution of each of the subproblems (ii) evaluate whether a subproblem is reducible or not in the sense of the usual separation of variables assumed by the POD.

These two points are addressed while considering that a relevant snapshot is *a priori* available. This relevant snapshot should explore the parameter domain sufficiently. At the same time, one does not want to compute too many snapshot solutions, in order for the “offline/online” strategy to remain affordable. Ultimately, a third point has to be added for the design of a substructured learning strategy: (iii) assess whether the snapshot contains a sufficient quantity of information, and generate additional, well-chosen data if required. This last issue is extremely complicated to address. Some recent propositions have been made in [60, 46, 56], but most of the studies on the POD, or the Principal Component Analysis in the statistics community (a recent review is provided in [61]) consider that a sufficiently rich snapshot is available, and perform the spectral analysis without considering the need, or the possibility, to regenerate data *a posteriori*.

We will here address points (i) and (ii), while point (iii) will be left to the perspectives of this work. The particular technique used in this paper relies heavily on cross-validation (CV, see [62] in the context of the PCA), and more precisely the Leave-One-Out (LOOCV) technique. In order to validate the predictivity of statistical models, one usually divide the available data into a training set and a validation set. In our application, the training set is the snapshot: the set of solutions to the parametric problem of evolution that corresponds to parameter values in $\mathcal{P}^{\text{s}}$. The relevancy of the reduced spaces generated by the snapshot-POD can then be evaluated on a set of additional fine-scale solutions: the training set. Using independent training and validation sets permits to avoid the overfitting behaviour (or “Type-III error” in statistics) that is classically observed in any regression-type model. In our context, the Snapshot POD only minimises the mean square error of projection of the snapshot solutions in the reduced space (17). Therefore, the associated error estimate (19) is expected to underestimate the error of projection associated to a hierarchically enriched snapshot, and in the limit, to underestimate the integral form (13) of the error of projection. Using a different set of solutions to identify the reduced space and to compute the error of projection permits to avoid this effect, but at the cost of additional data, which means further cost-intensive fine-scale solutions in our case.

The cross-validation error estimate avoids these additional computations by emulating the independence of training and validation sets using the same dataset. In order to do so, the summand in equation (56) is calculated using the local reduced basis obtained by a snapshot POD transform of all the available snapshot solutions but the one corresponding to the value of the summation variable. This is the usual LOOCV strategy applied to the POD. This can be written formally, for any subdomain $e \in \mathcal{E}$:

$$\left(\tilde{\nu}^{(e)}_{\text{snap}}\right)^2 = \frac{\displaystyle\sum_{\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\text{s}}} \sum_{t \in \mathcal{T}^{\text{h}}} \left\| \underline{\mathbf{U}}_{\mathbf{i}}(t;\underline{\boldsymbol{\mu}}) - \sum_{j=1}^{n_{\text{c}}^{(e)}} \left(\underline{\tilde{\mathbf{C}}}_{\mathbf{i},j}^{(e),(\underline{\boldsymbol{\mu}})\,T} \, \underline{\mathbf{U}}_{\mathbf{i}}(t;\underline{\boldsymbol{\mu}}) \right) \underline{\tilde{\mathbf{C}}}_{\mathbf{i},j}^{(e),(\underline{\boldsymbol{\mu}})} \right\|_2^2}{\displaystyle\sum_{t \in \mathcal{T}^{\text{h}}} \sum_{\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\text{s}}} \| \underline{\mathbf{U}}_{\mathbf{i}}(t;\underline{\boldsymbol{\mu}}) \|_2^2}, \tag{64}$$

the modified reduced basis vectors $\left(\underline{\tilde{\mathbf{C}}}_{\mathbf{i},j}^{(e),(\underline{\boldsymbol{\mu}})}\right)_{j \in [\![1, n_{\text{c}}^{(e)}]\!]}$, which are parametrised by the summation variable $\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\text{s}}$, are the $n_{\text{c}}^{(e)}$ first eigenvectors of the following modified POD operator:

$$\underline{\underline{\tilde{\mathbf{H}}}}^{(\underline{\boldsymbol{\mu}})} = \sum_{\underline{\boldsymbol{\mu}}^\star \in \left(\mathcal{P}^{\text{s}} \setminus \underline{\boldsymbol{\mu}}\right)} \sum_{t \in \mathcal{T}^{\text{h}}} \underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}^\star) \, \underline{\mathbf{U}}(t;\underline{\boldsymbol{\mu}}^\star)^T . \tag{65}$$

Technically speaking, the computation of this estimate requires to perform an SVD for each of the snapshot solutions (and for each subdomain).

Let us remark that statistical error estimates are commonly used in the context of deterministic parametric problem. For instance, classical Kriging interpolations are based on a randomisation of the field to interpolate. We refer to [35, 46] for recent combinations of Kriging and POD. The later contribution uses the LOOCV both as an error estimate and as a criterion to refine the snapshot space in a hierarchical manner.

4.3.2 Application

The LOOCV error estimate is now applied to the problem of fracture. The parameter domain is sampled using a regular grid of 5 parameter values including the extremities of $\mathcal{P} = [15^\circ \; 45^\circ]$, which is, for now, assumed to be sufficiently fine for our purpose. In figure 6, the corresponding LOOCV estimate is plotted as a function of the dimension of the local reduced spaces for 4 different subdomains: subdomain 6, which is the most affected by the damage propagation, subdomain 4, which contains the "tip of the crack" for a range of parameter angles, and subdomains 2 and 7, which are further away from the source of nonlinearity (or lack of correlation, depending on the point of view). Again, we emphasise that we treat all subdomains in the same manner. We do not make use of an *a priori* knowledge of the spatial distribution of damage. The lack of reducibility of certain parametric subproblems must be an output of the method.

The effect of the localised damage on the error estimates of each subdomain is relatively clear. For subdomains that are far away from the crack, we observe a fast convergence of the LOOCV error estimate with the dimension of the local POD reduced spaces. A satisfyingly level of predictivity, set here to the threshold $\tilde{\nu}^{(e)}_{\text{snap}} \leq 10^{-3}$, is obtained with 4 to 5 reduced basis vectors. It is interesting to notice that we do not obtain a clear "elbow" in the convergence curve, which is often used to define the "dimensionality" of the underlying parametric problem. This is, to our best knowledge, due to the far effect of the crack. The lack of correlation due to the local damage tends to pollute the remote area. Further evidence of this fact can be found in our recent investigations about this particular effect [63]. For the subdomains that contain most of the damage, the observed convergence curves are much flatter. The required accuracy for subdomain 4 is obtained with 7 local POD basis vectors. In the case of subdomain 6, the LOOCV error estimate does not reach the predefined threshold. This indicates that the corresponding subproblem should not be reduced.

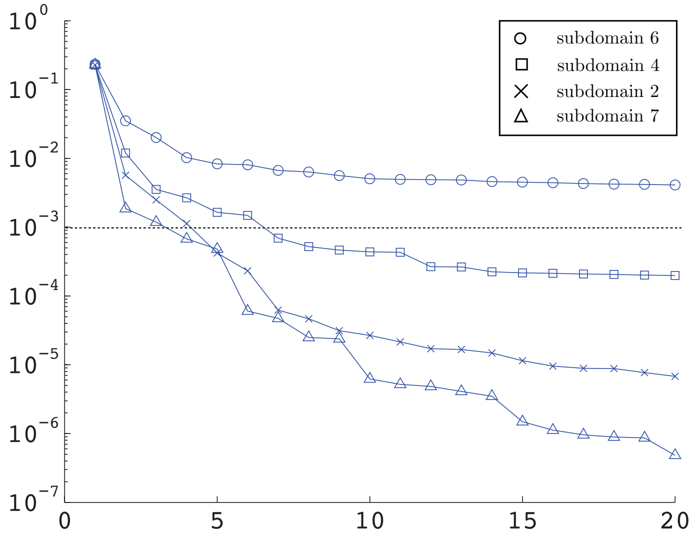


Figure 6: Cross-validation error estimate as a function of the order of the POD transforms for 4 of the 10 subdomains. The snapshot comprises 5 instances of the solution to the parametric problem of evolution. Subdomains are numbered as in Figure 5.

We have now achieved our objective of choosing the dimension of the local reduced spaces based on a CV error estimate, and identifying non-reducible subproblems, based on an assumed sufficiently fine sampling of the parameter domain. The local reduced spaces obtained in this section will be the one used in the following to demonstrate the numerical efficiency of the partitioned model order reduction approach.

5 System approximation in the partitioned model order reduction approach

5.1 Local "gappy" approximations

We propose here to extend the concept of "system approximation" to the partitioned model order reduction introduced in section 4. As mentioned previously, we choose to apply a tailored version of the "gappy" reconstruction technique presented in different contexts in [5, 51, 36]. It is important to realise that the gappy technique approximates the Galerkin projection framework described in section 4. Therefore, the system approximation will systematically be compared, or optimised, with respect to this framework and not with respect to the "truth" modelling. This approach to system approximations is characterised as "consistent" in [36].

The starting point of the gappy technique is to compute local "static" reduced bases $\left\{ \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)} \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}} \times \mathbb{R}^{n_{\mathrm{d}}^{(e)}} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$ to approximate the vectors of internal forces $\left\{ \underline{\mathbf{F}}_{\mathbf{int},\mathbf{i}}^{(e)} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\} \overset{\mathrm{def}}{=} \left\{ \underline{\underline{\mathbf{E}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{int}}^{(e)} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$, as detailed previously in the non-partitioned case (see section 3). Once the local bases are computed, the approximation reads

$$\forall\, e \in \mathcal{E}^{\mathrm{red}}\,, \forall\, t \in \mathcal{T}^{\mathrm{h}}\,, \forall\, \underline{\boldsymbol{\alpha}}^{(e)^\star} \in \mathbb{R}^{n_{\mathrm{c}}^{(e)}}\,,\ \forall\, \underline{\mathbf{U}}_{\mathbf{b}}^{(e)^\star} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}\,,$$
$$\underline{\mathbf{F}}_{\mathbf{int,i}}^{(e)} \left(\begin{pmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} \underline{\boldsymbol{\alpha}}^{(e)^\star} \\ \underline{\mathbf{U}}_{\mathbf{b}}^{(e)^\star} \end{pmatrix}, \left(\underline{\mathbf{U}}^{(e)}(\tau; \underline{\boldsymbol{\mu}}) \right)_{\tau \in \mathcal{T}^{\mathrm{h}}, \tau < t}; \underline{\boldsymbol{\mu}} \right) \approx \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)}\, \underline{\boldsymbol{\beta}}_{\mathbf{i}}^{(e)} \left(\begin{pmatrix} \underline{\boldsymbol{\alpha}}^{(e)^\star} \\ \underline{\mathbf{U}}_{\mathbf{b}}^{(e)^\star} \end{pmatrix}, \left(\underline{\mathbf{U}}^{(e)}(\tau; \underline{\boldsymbol{\mu}}) \right)_{\tau \in \mathcal{T}^{\mathrm{h}}, \tau < t}, \underline{\boldsymbol{\mu}} \right), \tag{66}$$

where $n_{\mathrm{b}}^{(e)}$ is the number of interface degrees of freedom of subdomain e. We assume that the "static" reduced bases are available. In the "online" stage, the "static" interpolation coefficients $\left\{ \underline{\boldsymbol{\beta}}_{\mathbf{i}}^{(e)} \in \mathbb{R}^{n_{\mathrm{d}}^{(e)}} \mid e \in \mathcal{E}^{\mathrm{red}} \right\}$ are obtained at an arbitrary point along the reduced kinematic trajectory by minimisation of a distance between the previous approximation and the exact local vector of internal forces evaluated. This distance is measured at a set of sample spatial points, which yields the partitioned gappy approximation

$$\forall\, e \in \mathcal{E}^{\mathrm{red}}\,, \forall\, \underline{\boldsymbol{\alpha}}^{(e)^\star} \in \mathbb{R}^{n_{\mathrm{c}}^{(e)}}\,,\ \forall\, \underline{\mathbf{U}}_{\mathbf{b}}^{(e)^\star} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}\,,$$
$$\underline{\mathbf{F}}_{\mathbf{int,i}}^{(e)} \left(\begin{pmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} \underline{\boldsymbol{\alpha}}^{(e)^\star} \\ \underline{\mathbf{U}}_{\mathbf{b}}^{(e)^\star} \end{pmatrix} \right) \approx \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)} \left(\underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)^T} \underline{\underline{\mathbf{P}}}_{\mathbf{i}}^{(e)}\, \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)} \right)^{-1} \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)^T} \underline{\underline{\mathbf{P}}}_{\mathbf{i}}^{(e)}\, \underline{\mathbf{F}}_{\mathbf{int,i}}^{(e)} \left(\begin{pmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} \underline{\boldsymbol{\alpha}}^{(e)^\star} \\ \underline{\mathbf{U}}_{\mathbf{b}}^{(e)^\star} \end{pmatrix} \right), \tag{67}$$

The local boolean operator $\underline{\underline{\mathbf{P}}}_{\mathbf{i}}^{(e)}$ operating on the subdomain $e \in \mathcal{E}^{\mathrm{red}}$ is such that only the diagonal entries that correspond to all the degrees of freedom of a small set of internal nodes of subdomain e are set to one. These nodes are called "control points" or "control nodes". We define the local "gappy" operator of subdomain e by $\underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} = \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)} \left(\underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)^T} \underline{\underline{\mathbf{P}}}_{\mathbf{i}}^{(e)} \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)} \right)^{-1} \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)^T} \underline{\underline{\mathbf{P}}}_{\mathbf{i}}^{(e)}$.

Let us explain how this approximation is employed to reduce the "online" numerical complexity of the partitioned Galerkin-POD technique. Upon linearisation of the local nonlinear subproblems (i.e.: derivation of the vector of internal forces with respect to the reduced state variables and interface degrees of freedom), and taking into account the gappy approximation (67), one gets a modified expression of the local tangent systems (compare equation (58)) at Newton iteration $i+1$ of an arbitrary time-parameter point of $\tilde{\mathcal{P}}$, for any subdomain $e \in \mathcal{E}$:

$$\begin{bmatrix} \underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} \underline{\underline{\mathbf{K}}}_{\mathbf{ii}}^{(e)} & \underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} \underline{\underline{\mathbf{K}}}_{\mathbf{ib}}^{(e)} \\ \underline{\underline{\mathbf{K}}}_{\mathbf{bi}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{bb}}^{(e)} \end{bmatrix} \begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} \underline{\boldsymbol{\Delta\alpha}}_{\mathbf{i}}^{(e)} \\ \underline{\boldsymbol{\Delta}\mathbf{U}}_{\mathbf{b}}^{(e)} \end{bmatrix} = \begin{bmatrix} -\underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} \underline{\mathbf{F}}_{\mathbf{int,i}}^{(e)} \left(\underline{\mathbf{U}}^{(e),i} \right) - \underline{\mathbf{F}}_{\mathbf{ext,i}}^{(e)} \\ -\underline{\mathbf{R}}_{\mathbf{b}}^{(e)} + \underline{\boldsymbol{\lambda}}^{(e)} \end{bmatrix}, \tag{68}$$

with $\underline{\mathbf{F}}_{\mathbf{ext,i}}^{(e)} \stackrel{\text{def}}{=} \underline{\underline{\mathbf{E}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{ext}}^{(e)}$

As mentioned in section 3, this system is overdetermined but solutions can be obtained by making use of optimum arguments. We use a Galerkin projection, which, together with the gappy approximation, yields the following matrix formulation of the tangent subproblem corresponding to subdomain $e \in \mathcal{E}^{\mathrm{red}}$:

$$\left(\underline{\mathbf{F}}_{\mathbf{r,sa}}^{(e)} + \begin{bmatrix} \underline{\mathbf{0}} \\ \underline{\boldsymbol{\lambda}}^{(e)} \end{bmatrix} \right) - \underline{\underline{\mathbf{K}}}_{\mathbf{r,sa}}^{(e)} \begin{bmatrix} \underline{\boldsymbol{\Delta\alpha}}_{\mathbf{i}}^{(e)} \\ \underline{\boldsymbol{\Delta}\mathbf{U}}_{\mathbf{b}}^{(e)} \end{bmatrix} = \underline{\mathbf{0}} \quad \text{with} \begin{cases} \underline{\underline{\mathbf{K}}}_{\mathbf{r,sa}}^{(e)} = \begin{bmatrix} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)^T} \underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} \underline{\underline{\mathbf{K}}}_{\mathbf{ii}}^{(e)} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} & \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)^T} \underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} \underline{\underline{\mathbf{K}}}_{\mathbf{ib}}^{(e)} \\ \underline{\underline{\mathbf{K}}}_{\mathbf{bi}}^{(e)} \underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)} & \underline{\underline{\mathbf{K}}}_{\mathbf{bb}}^{(e)} \end{bmatrix} \\ \underline{\mathbf{F}}_{\mathbf{r,sa}}^{(e)} = \begin{bmatrix} -\underline{\underline{\mathbf{C}}}_{\mathbf{i}}^{(e)^T} \left(\underline{\underline{\mathbf{G}}}_{\mathbf{i}}^{(e)} \underline{\mathbf{F}}_{\mathbf{int,i}}^{(e)} (\underline{\mathbf{U}}^{(e),i}) + \underline{\mathbf{F}}_{\mathbf{ext,i}}^{(e)} \right) \\ -\underline{\mathbf{R}}_{\mathbf{b}}^{(e)} \end{bmatrix}. \end{cases} \tag{69}$$

A condensed linearised interface problem is finally obtained as follows. We look for $\underline{\boldsymbol{\Delta}\mathbf{U}}_{\mathbf{b}} \in \mathbb{R}^{n_{\mathrm{b}}}$ satisfying

$$\underline{\underline{\mathbf{S}}}_{\mathbf{p,r,sa}}\, \underline{\boldsymbol{\Delta}\mathbf{U}}_{\mathbf{b}} = \underline{\mathbf{F}}_{\mathbf{c,r,sa}} \quad \text{with} \begin{cases} \underline{\underline{\mathbf{S}}}_{\mathbf{p,r,sa}} = \displaystyle\sum_{e \in \mathcal{E}^{\mathrm{red}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}_{\mathbf{p,r,sa}}^{(e)} \underline{\underline{\mathbf{A}}}^{(e)^T} + \sum_{e \in \mathcal{E}^{\mathrm{nred}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\underline{\mathbf{S}}}_{\mathbf{p}}^{(e)} \underline{\underline{\mathbf{A}}}^{(e)^T} \\ \underline{\mathbf{F}}_{\mathbf{c,r,sa}} = \displaystyle\sum_{e \in \mathcal{E}^{\mathrm{red}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{c,r,sa}}^{(e)} + \sum_{e \in \mathcal{E}^{\mathrm{nred}}} \underline{\underline{\mathbf{A}}}^{(e)} \underline{\mathbf{F}}_{\mathbf{c}}^{(e)}. \end{cases} \tag{70}$$

The method to obtain the expression of the modified primal Schur complement $\underline{\underline{\mathbf{S}}}_{\mathbf{p,r,sa}}$ and the corresponding condensed right-hand side is not detailed for the sake of concision. It follows exactly the method deployed to get their counterparts whereby no system approximation was used (see equation (62)).

Notice that the symmetry of the condensed interface problem is lost when using the gappy technique. This issue can be alleviated by using a GMRes algorithm.

The key benefit in using the gappy technique is that only the components of the local tangents and local residuals that are not filtered out by operators $\left\{ \underline{\underline{\mathbf{P}}}_{\mathbf{i}}^{(e)} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$ need to be computed, the remainder being reconstructed by interpolation in the "static" reduced spaces. In terms of implementation, the assembly of the tangents and residuals is performed via loops over all elements. With the system approximation, only contributions from elements that are connected to one of the "control nodes" are computed, which results in an online complexity that does not depend on the "truth" number of unknowns. The set of elements over which an integration of the internal forces is required is called the reduced integration domain. An example of such a domain is shown in Figure 7. The way this reduced integration domain was obtained is detailed in the following.

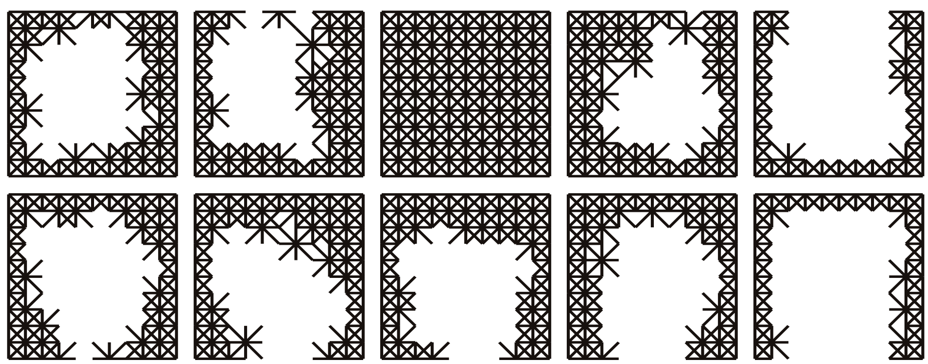


Figure 7: Example of a reduced integration domain. Subdomain 6 is not reduced. Therefore, all the associated elements belong to the integration domain. Since the interface between substructures is not reduced in the proposed primal version of the Schur-based partitioned model order reduction method, all the elements that are connected to the interface also belong to the reduced integration domain. The remaining controlled nodes are obtained by a Partitioned Discrete Empirical Interpolation Method.

5.2 Construction of the system approximation

5.2.1 Static POD bases

To generate the local bases $\left\{ \underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$, we develop a technique that is strongly inspired by the one proposed in [36]. Equation (66) indicates that we would like the system approximation to be optimal for any set of local reduced state variables. However, we can reasonably restrict ourselves to the state variables that are observed on a set of particular solutions to the Galerkin projection of the parametric problem in the kinematic reduced space. In order to do so, we first solve all time evolution problems corresponding to snapshot space $\mathcal{P}^{\mathrm{s}}$ using the Galerkin framework described in section 4, without system approximation. Such computations are expensive, but they are performed "offline". The local solutions that are obtained in this fashion belong to the local POD reduced spaces and are considered as reference for the system approximation. We now want to approximate the spaces spanned by the local vectors of internal forces corresponding to the successive iterations of the Newton algorithm used to compute these reduced solutions. Let us call these spaces the "static" snapshot spaces. They can be represented mathematically, for any subdomain $e \in \mathcal{E}^{\mathrm{red}}$, by the following set:

$$\mathcal{F}^{s,(e)} = \left\{ \underline{\mathbf{F}}^{(e)}_{\mathbf{int},\mathbf{i}} \left(\begin{pmatrix} \underline{\underline{\mathbf{C}}}^{(e)}_{\mathbf{i}} \underline{\boldsymbol{\alpha}}^{(e),i}(t, \underline{\boldsymbol{\mu}}) \\ \underline{\mathbf{U}}^{(e),i}_{\mathbf{b}}(t; \underline{\boldsymbol{\mu}}) \end{pmatrix}, \left(\underline{\mathbf{U}}^{(e)}(\tau; \underline{\boldsymbol{\mu}}) \right)_{\tau \in \mathcal{T}^{\mathrm{h}}, \tau < t} ; \underline{\boldsymbol{\mu}} \right) \,\middle|\, \underline{\boldsymbol{\mu}} \in \mathcal{P}^{\mathrm{s}}, t \in \mathcal{T}^{\mathrm{h}}, \, i \in [\![1, n^{(t),(\underline{\boldsymbol{\mu}})}_{\mathrm{new}}]\!] \right\} . \tag{71}$$

In the previous expression, $n^{(t),(\underline{\boldsymbol{\mu}})}_{\mathrm{new}}$ denotes the number of iterations of the Newton algorithm used to solve the problem of evolution at time $t \in \mathcal{T}^{\mathrm{h}}$ and for parameter $\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\mathrm{s}}$. A singular value decomposition can now be used to compress and hierarchically order the information contained in this set, which is similar to the technique used to obtain the reduced bases for the displacements and constitutes a keystone for the greedy selection of the reduced integration domain proposed in [5, 51]. Technically, for each subdomain $e \in \mathcal{E}^{\mathrm{red}}$, a matrix whose columns are the vectors of set (71) is constructed. This matrix is decomposed by singular value decomposition. The left-singular vectors associated to singular values that are larger than a certain tolerance define the columns of operator $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$.

5.2.2 Selection of the control points

For each subdomain $e \in \mathcal{E}^{\mathrm{red}}$, given the "static" reduced basis $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$, we can now choose which subset of interior nodes will be defined as control nodes. This choice completely defines boolean operator $\underline{\underline{\mathbf{P}}}^{(e)}_{\mathbf{i}}$ and, together with $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$ obtained in the previous subsection, the required gappy reconstruction operator $\underline{\underline{\mathbf{G}}}^{(e)}_{\mathbf{i}}$.

In the context of the DEIM [51], the selection is performed in a greedy manner, for increasing rank of operator $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$, where we recall that the columns of this operator are hierarchically ordered by SVD. More precisely, at iteration $j > 0$ of the greedy algorithm, the degree of freedom for which the gappy interpolation error

$$\underline{\boldsymbol{\epsilon}}^{(e),j}_{\mathbf{i},\mathbf{gap}} = \underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i},[1,j]} \, \underline{\boldsymbol{\beta}}^{j} - \underline{\mathbf{D}}^{(e)}_{\mathbf{i},j+1} \,, \tag{72}$$

is maximum is defined as a "control degree of freedom". Operator $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i},[1,j]}$ is composed of the j first columns of $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$, while $\underline{\mathbf{D}}^{(e)}_{\mathbf{i},j+1}$ is the $j+1^{\mathrm{th}}$ column of $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$. Interpolation coefficient $\underline{\boldsymbol{\beta}}^{j}$ is obtained by solving the following optimisation problem:

$$\underline{\boldsymbol{\beta}}^{j} = \underset{\underline{\boldsymbol{\beta}}^{\star} \in \mathbb{R}^{j}}{\mathrm{argmin}} \left(\left\| \underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i},[1,j]} \, \underline{\boldsymbol{\beta}}^{\star} - \underline{\mathbf{D}}^{(e)}_{\mathbf{i},j+1} \right\|_{\underline{\underline{\mathbf{P}}}^{(e),j}_{\mathbf{i}}} \right) , \tag{73}$$

The rank of the j^{th} greedy iterate $\underline{\underline{\mathbf{P}}}^{(e),j}_{\mathbf{i}}$ is j-times the number of scalar unknowns per interior node of subdomain e. In our implementation of the method, the node carrying the new "control degree of freedom" is added as a new "control point", and all its associated degrees of freedom are controlled, which means that the corresponding entries in $\underline{\underline{\mathbf{P}}}^{(e),j+1}_{\mathbf{i}}$ are set to one. For an arbitrary subdomain e, the application of this method provides a number of "control nodes" equal to the rank of $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$. We refer to reference [51] for more details about this technique, and in particular for a discussion about its optimality (in a greedy sense) and stability.

5.2.3 Dimension of the local POD "static spaces"

One question that now arises is how to choose the order of truncation of the local SVD performed to approximate span($\mathcal{F}^{s,(e)}$), for any subdomain $e \in \mathcal{E}$. In other words, we need to choose the rank of the matrix of left singular vectors $\underline{\underline{\mathbf{D}}}^{(e)}_{\mathbf{i}}$ for each subdomain $e \in \mathcal{E}^{\mathrm{red}}$. The simplest method is to truncate the local SVDs such that the truncation error becomes smaller than a predefined tolerance, or to use a cross-validation estimate, as proposed in section 4.3 when defining the dimension of the local reduced spaces for the displacements. However, we prefer here to link the error generated by the gappy reconstruction technique to an error measured in terms of displacements, such that it can be compared to the error introduced by the truncation of the local snapshot POD performed to generate the local "kinematic" reduced spaces.

In order to implement this idea, we proceed in an iterative manner. For a given truncation of the local "static" SVDs, we evaluate the error introduced by the system approximation directly. This is done by solving the reduced problem when using the system approximation, and comparing the solution obtained in this fashion to the solution obtained when solving the reduced system of equations without system approximation. The error is of course only evaluated for parameter values belonging to the sampled parameter domain $\mathcal{P}^{\mathrm{s}}$. If this error estimate is too large (in a sense to be defined later on), the dimensions of the "static" reduced spaces is increased and the error estimation procedure is repeated.

More specifically, we initiate the iterative process with $n_{\mathrm{d}}^{(e)} = n_{\mathrm{c}}^{(e)}$ for all subdomains $e \in \mathcal{E}$. Local indicators for the total error introduced by the reduced order modelling technique are defined as follows:

$$\forall\, e \in \mathcal{E}, \quad \nu_{\mathrm{tot}}^{(e)} = \sum_{\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\mathrm{s}}} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \left\| \underline{\mathbf{U}}_{\mathbf{ex}}^{(e)}(t;\underline{\boldsymbol{\mu}}) - \underline{\mathbf{U}}_{\mathbf{r,sa}}^{(e)}(t;\underline{\boldsymbol{\mu}}) \right\|_2 , \tag{74}$$

where $\underline{\mathbf{U}}_{\mathbf{ex}}^{(e)}$ is the "truth" solution to the parametric time-dependant problem, which has been computed to build the POD projection space for the displacement, and $\underline{\mathbf{U}}_{\mathbf{r,sa}}^{(e)}$ denotes the solution obtained when using the reduced order model, with the current iterate of the system approximation, which needs to be computed. Performing simple algebraic manipulations, we can recast the expression of these estimates in the following manner:

$$\forall\, e \in \mathcal{E}, \quad \nu_{\mathrm{tot}}^{(e)} = \sum_{\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\mathrm{s}}} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \left\| \underline{\mathbf{U}}_{\mathbf{ex}}^{(e)}(t;\underline{\boldsymbol{\mu}}) - \underline{\mathbf{U}}_{\mathbf{r}}^{(e)}(t;\underline{\boldsymbol{\mu}}) + \underline{\mathbf{U}}_{\mathbf{r}}^{(e)}(t;\underline{\boldsymbol{\mu}}) - \underline{\mathbf{U}}_{\mathbf{r,sa}}^{(e)}(t;\underline{\boldsymbol{\mu}}) \right\|_2 , \tag{75}$$

with $\underline{\mathbf{U}}_{\mathbf{r}}^{(e)}$ the solution to the parametrised problem obtained when using the reduced order model without system approximation, which has been computed to generate the "static" snapshot. We can now use the triangle inequality, which yields the following relationship:

$$\forall\, e \in \mathcal{E}, \quad \nu_{\mathrm{tot}}^{(e)} \leq \nu_{\mathrm{r}}^{(e)} + \nu_{\mathrm{r,sa}}^{(e)} \quad \text{with} \quad \left\{ \begin{aligned} \nu_{\mathrm{r}}^{(e)} &= \sum_{\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\mathrm{s}}} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \left\| \underline{\mathbf{U}}_{\mathbf{ex}}^{(e)}(t;\underline{\boldsymbol{\mu}}) - \underline{\mathbf{U}}_{\mathbf{r}}^{(e)}(t;\underline{\boldsymbol{\mu}}) \right\|_2 \\ \nu_{\mathrm{r,sa}}^{(e)} &= \sum_{\underline{\boldsymbol{\mu}} \in \mathcal{P}^{\mathrm{s}}} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \left\| \underline{\mathbf{U}}_{\mathbf{r}}^{(e)}(t;\underline{\boldsymbol{\mu}}) - \underline{\mathbf{U}}_{\mathbf{r,sa}}^{(e)}(t;\underline{\boldsymbol{\mu}}) \right\|_2 \end{aligned} \right. \tag{76}$$

Now, the term $\nu_{\mathrm{r,sa}}^{(e)}$ measures the local error introduced by the system approximation, while $\nu_{\mathrm{r}}^{(e)}$ measures the local error introduced by the kinematic approximation, which is monitored by the cross-validation estimate defined in section 4.3, and can be decreased by enriching the "kinematic" reduced space. The idea is then to compare these two estimates and to make sure that they are of the same order of magnitude, which can be formulated as follows:

$$\frac{\nu_{\mathrm{r,sa}}^{(e)}}{\nu_{\mathrm{r}}^{(e)}} \leq 1 \tag{77}$$

If this condition is not satisfied with the current iterate of the system approximation, for any subdomain $e \in \mathcal{E}$, the rank $n_{\mathrm{d}}^{(e)}$ of the corresponding "static" operator $\underline{\underline{\mathbf{D}}}_{\mathbf{i}}^{(e)}$ is increased (by one in our current implementation), and the error estimation procedure is repeated.

Notice that this simple strategy to control the accuracy of the gappy technique requires to compute a certain number of solutions to the evolution problem corresponding to parameters in $\mathcal{P}^{\mathrm{s}}$. However, this is performed "offline", and at reduced cost as we make use of the the gappy technique to compute the iterates of $\{\nu_{\mathrm{r,sa}}^{(e)} \,|\, e \in \mathcal{E}\}$, while the set $\{\nu_{\mathrm{r}}^{(e)} \,|\, e \in \mathcal{E}\}$ is computed once and for all and only requires information that is already available.

The reduced integration domain obtained by applying the methodology described in this section is represented in figure 7 and will be the one used in the next section.

6 Results

6.1 Online numerical costs ("speed-up")

We now solve the parametric, time-dependent lattice problem described in section 3.4 using the partitioned model order reduction approach, and report the speed-up in terms of run time. Speed-up is here to be understood as the ratio between the CPU time that is necessary to solve the "truth" model, and the CPU time required to solve the reduced order model. The high numerical costs of the "offline" phase are not considered in this definition.

We propose four different lattice structures, using 121 (figure 8), 256, 441 and 961 (figure 9) nodes for each of the 10 subdomains. The snapshot that is used to compute the local reduced spaces is the one chosen in section 4.3. Let us recall that the cross-validation procedure leads us to omit any reduction in subdomain 6, whose associated subproblem will be solved exactly. The remainder of the subproblems are projected in the appropriate reduced spaces identified in section 4.3, using the Petrov-Galerkin formulation (system approximation) developed in section 5. We present speed-up results for the simulations corresponding to $\theta = 40°$ and $\theta = 27°$. These time solutions are not in the snapshot, and we can reasonably extrapolate that the observed speed-ups are representative of what can be expected for an arbitrary value of the parameter.

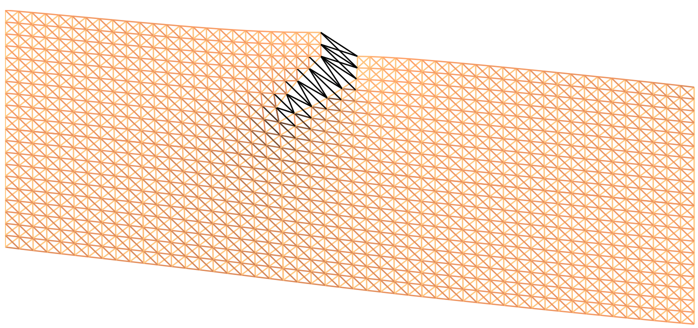


Figure 8: Solution corresponding to the last time step of the fully discrete time-dependent problem for a load angle of 45°. The lattice structure represented here is composed of 121 nodes per subdomain. The darkest bars correspond to a completely damaged state of the material, while the lightest bars are undamaged.

The proposed methodology is implemented in the commercial package Matlab, in a pseudo parallel fashion: the required operations that are local per subdomains are performed sequentially using a single processor. In this setting, we choose to solve the non-symmetric condensed interface problems using a direct LU factorisation. The reason for this is that no reduction of this problem has been developed so far. The number of interface degrees of freedom remains unchanged after the projection of the subproblems in the local reduced spaces. We therefore chose the implementation of the method that favored the observed speed-up, keeping in mind that it is pseudo-parallel. We will come back to this point in the conclusion of this work.

In order to show the performance of the reduced order model, we first compute the "truth" solution of the fully discrete problem that corresponds to the first of the two particular load angles mentioned previously. Note that this fine solution is computed using the partitioned model, but with no reduction. The convergence tolerance for the Newton algorithm used at each time step (euclidean norm of the residual divided by the norm of the vector of external forces) is set to 10^{-7}. This is the reference solution $\underline{\mathbf{U}}_{\mathbf{ex}}$. The accuracy of any approximate solution $\underline{\mathbf{U}}_{\mathbf{app}}$ will be quantified using the following

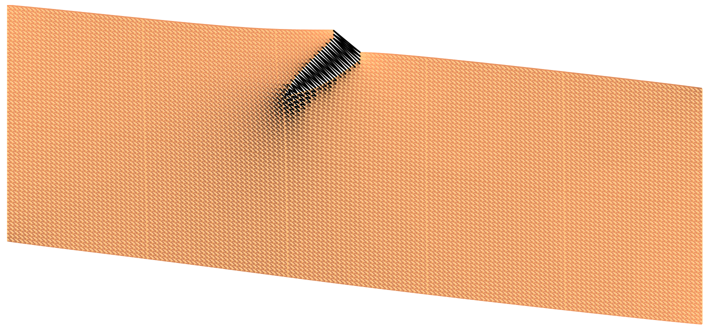


Figure 9: Solution correspoding to the last time step of the fully discrete time-dependent problem for a load angle of 45°, using 961 lattice nodes per subdomain.

normalised error function:

$$\nu_{\mathrm{app}}^{(\underline{\boldsymbol{\mu}})}(\underline{\mathbf{U}}_{\mathbf{app}})^2 = \frac{\sum_{t\in\mathcal{T}^h} \left\|\underline{\mathbf{U}}_{\mathbf{app}}(t;\underline{\boldsymbol{\mu}}) - \underline{\mathbf{U}}_{\mathbf{ex}}(t;\underline{\boldsymbol{\mu}})\right\|_2^2}{\sum_{t\in\mathcal{T}^h} \left\|\underline{\mathbf{U}}_{\mathbf{ex}}(t;\underline{\boldsymbol{\mu}})\right\|_2^2} . \tag{78}$$

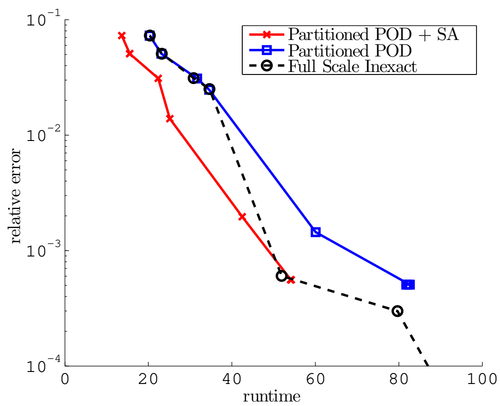
Secondly, an approximate solution $\underline{\mathbf{U}}_{\mathbf{inex}}$ is obtained by a straightforward time-reduction technique: the Newton algorithms are solved to a loose tolerance, and the error $\nu_{\mathrm{app}}^{(\underline{\boldsymbol{\mu}})}(\underline{\mathbf{U}}_{\mathbf{inex}})$ is reported as a function of run time in figure 10. This result is entitled "Full Scale Inexact" (notice that our use of the term "inexact" is not to be confused with the Inexact Newton Method, whereby one loosens the convergence tolerance of an iterative linear solver associated with the successive predictors of a Newton algorithm [64]) .

Finally, we compare the speed-up obtained when using this straightforward approach to the one obtained with the projection-based partitioned reduction approaches. The error between the reference solution $\underline{\mathbf{U}}_{\mathbf{ex}}$ and the one obtained by the Galerkin projection-based partitioned model order reduction (without system approximation), denoted by $\underline{\mathbf{U}}_{\mathbf{r}}$, is the output $\nu_{\mathrm{app}}^{(\underline{\boldsymbol{\mu}})}(\underline{\mathbf{U}}_{\mathbf{r}})$ of the previously defined error function. The corresponding result is labelled "Partitioned POD" in figure 10. The error $\nu_{\mathrm{app}}^{(\underline{\boldsymbol{\mu}})}(\underline{\mathbf{U}}_{\mathbf{r,sa}})$ of solution $\underline{\mathbf{U}}_{\mathbf{r,sa}}$ obtained with the partitioned reduction technique and the system approximation is reported next, under the label "Partitioned POD + System Approximation". All these curves are reproduced for the second test load angle in figure 11.

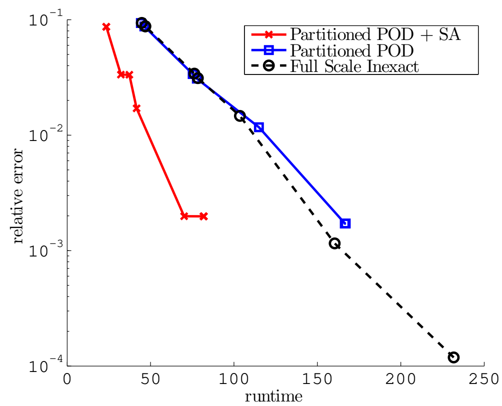
The errors described previously are plotted for different levels of convergence of the Newton algorithms, in both the approximate full-scale case and the reduced cases, which provides a fair comparison ground for the various domain decomposition algorithms.

Observing the two figures of results, the following remarks can be made:

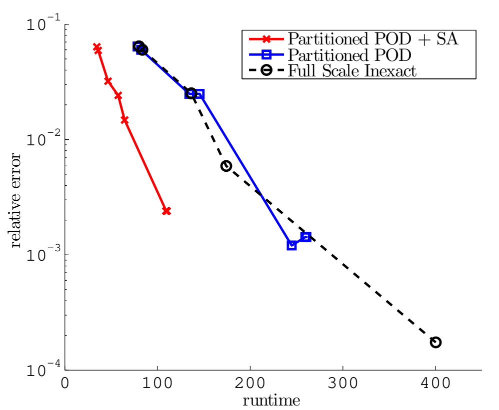
- a significant speed-up is obtained when using the partitioned model order reduction approach together with the system approximation. This observation is only valid for certain range of accuracy. Indeed, the projection-based approach is limited, in terms of reachable accuracy, by the snapshot approximation of the POD, and by its subsequent truncation at a low order. For instance, in the top-right result of figure 10, the error obtained with the reduction method cannot decrease under 2×10^{-3}. This is of course to be expected, and the remedy to this problem, if necessary, is to increase the size of the local reduced spaces. On the contrary, the error versus



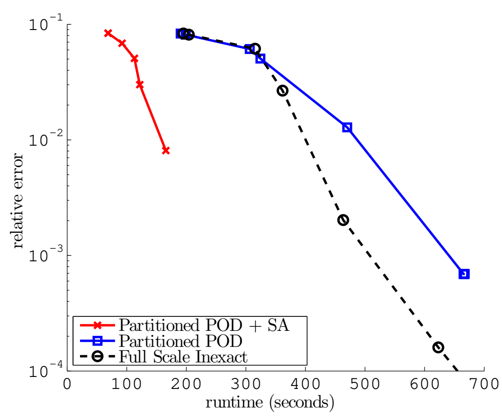
(a) Relative error for the different models using 121 nodes per subdomain



(b) Relative error for the different models using 256 nodes per subdomain

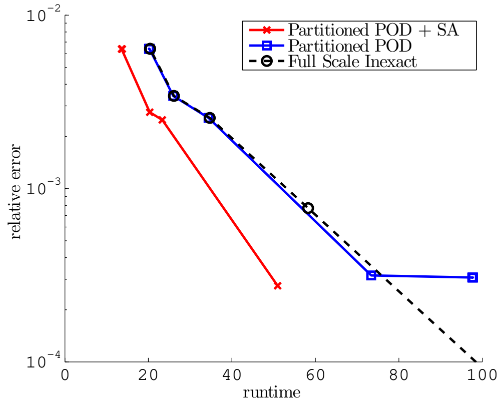


(c) Relative error for the different models using 441 nodes per subdomain

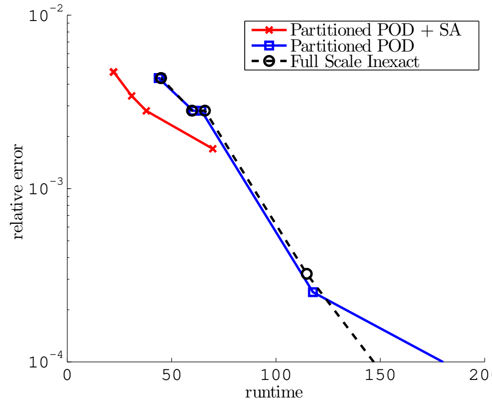


(d) Relative error for the different models using 961 nodes per subdomain

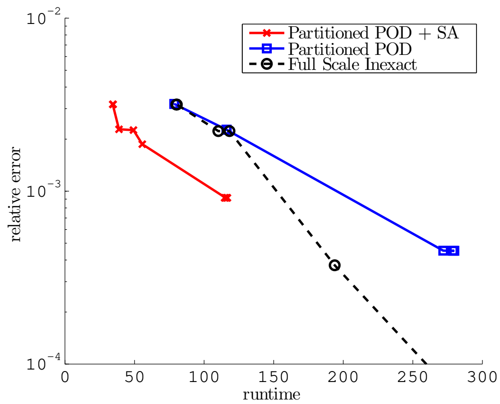
Figure 10: Relative error for the reference model and for the reduced order model as a function of runtime for a load angle $\theta = 40°$. The different points of the curves are generated by loosening the convergence of the Newton algorithms.



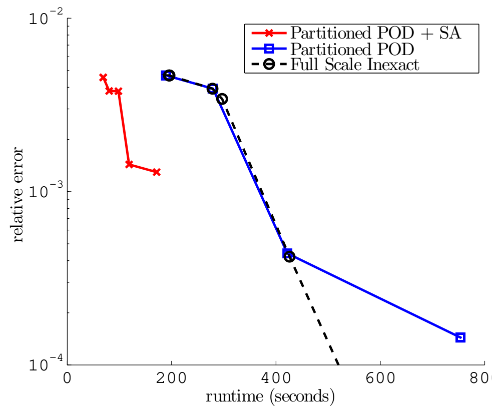
(a) Relative error for the different models using 121 nodes per subdomain



(b) Relative error for the different models using 256 nodes per subdomain



(c) Relative error for the different models using 441 nodes per subdomain



(d) Relative error for the different models using 961 nodes per subdomain

Figure 11: Relative error for the reference model and for the reduced order model as a function of runtime for a load angle $\theta = 27°$

CPU time corresponding to the "truth" problems can reach machine precision when decreasing the convergence tolerance of the Newton solvers.

- the Galerkin version of the partitioned POD approach produces insignificant speed-ups. This is a well-known fact. The number of degrees of freedom is reduced compared to the full-scale system, but the costly integrations of the reduced generalised forces over the spatial domain forbids any benefit in terms of computational gain over the reference model.

- the speed-up, observed in the region of reachable accuracy for the POD-based reduced order models, increases with the number of degrees of freedom of the reference problem. This can be easily explained. The cost of solving the reference problem increases with the number of fine-scale degrees of freedom. However, the dimensions of the local reduced spaces do not depend on this model refinement, but on the statistical properties of the parametric problem. Typically, one would expect that the numerical cost associated with the reduction technique does not increase with the number of degrees of freedom of the "truth" models. In practice, this is not the case as some computational overhead penalises our implementation of the partitioned model order reduction approach, not the least of which is the fact that the condensed interface problem is not reduced. This overhead becomes more important when one increases the number of subdomains while keeping the same mesh size, since the number of degrees of freedom on the interface increases. This will be discussed in the conclusion of the paper.

Notice that in practice, the simulations using the reduced models with system approximation are only performed with the lowest tolerance threshold for the Newton algorithm. The intermediate run times have only been given for demonstration purposes.

6.2 Remarks about the numerical efficiency of the system approximation

We now present the previous speed-up results in a different form. The aim is to show the trend in computational gain as a function of the number of degrees of freedom of the reference problem, when using the proposed reduction approach, in a unique graph. In order to so, the speed-up results reported previously are reported in figure 12 as a function of the ratio between the number of elements of the lattice and the number of elements that are connected to the control nodes of the system approximation. This ratio increases in a roughly linear manner with the number of degrees of freedom of the "truth" problem. The different points of the curve are the one obtained with the lattice models comprising respectively 64, 121, 256, 441 and 961 nodes per subdomain, with an appropriately low tolerance for the nonlinear solution algorithm.

The increase in the speed-up as function of the number of degrees of freedom of the full-scale problem appears clearly in this form. But more importantly, the graph shows that the observed speed-up is directly related to the size of the reduced integration domain. As mentioned previously, this is a clear indication that the main factor that prevents us from obtaining further speed-up with the proposed method is the fact that the interface problem is not reduced, which requires to perform integrations over a large number of elements. This is a path to explore in order to bring the idea of reduced order modelling in a partitioned framework to its full capability in the context of fracture.

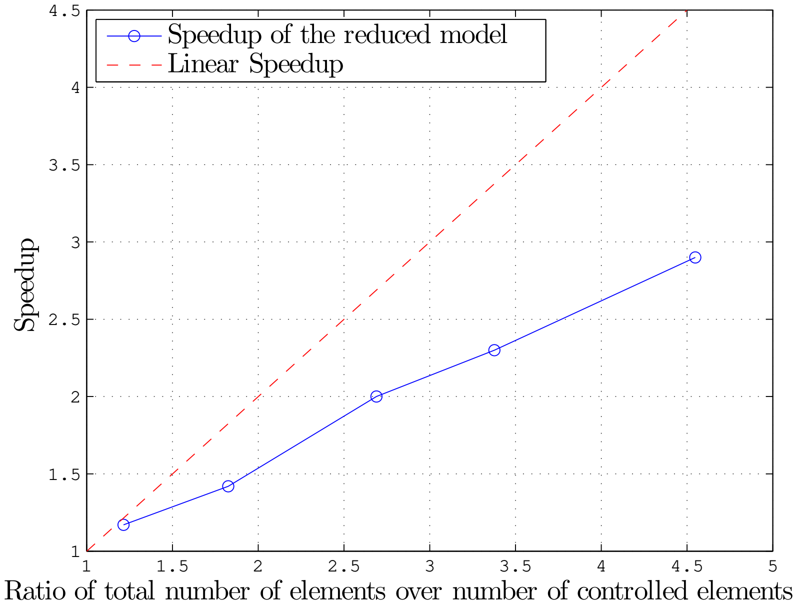


Figure 12: Evolution of the speed-up with the ratio of the number of elements in the structure over the number of elements comprising the reduced integration domain.

7 Conclusion and perspectives

In this paper, we have proposed a partitioned model order reduction strategy for parametrised problems of nonlinear fracture mechanics. The domain coupling has been performed using the tried and tested primal Schur-complement domain decomposition method. The local subproblems have been reduced by projection in low-dimensional subspaces obtained by the snapshot POD. We have shown that this approach permits to reduce, in a flexible manner, the computational cost associated with highly nonlinear problems. In particular:

- the local reduced spaces are generated independently, and have independent dimensions, which allows us to focus the numerical effort where it is most needed. In fracture mechanics, subdomains that are close to highly damaged zones need a richer model to account for the effect of topological changes. The local POD transforms automatically generate local reduced spaces of relatively large dimensions in these zones.

- the domain decomposition framework enables us to switch from reduced local solvers to "truth" local solvers in a transparent manner. This is particularly useful for the subdomains that contain process zones, as a solution obtained by reduced order modelling would become more expensive than a direct solution for a desirable accuracy.

- the transitition between "offline" and "online" computations becomes flexible. The reduced models can be used in the zones where the local reduced spaces converge in a fast manner when enriching the snapshot space, while still computing snapshots and refining the reduced models via a direct local solver in the remaining subdomains.

We have shown that such a flexibility results in a significant speed-up in the case of parametric fracture mechanics problems. This speed-up naturally increases when the size of the highly damaged zone, in which the information is highly uncorrelated, is small compared to the scale of the structure.

This work is a step towards an optimal cost-reduction strategy for parametrised problems of fracture. Further work needs to be done to increase the understanding, robustness and performance of the method. Two main research avenues are particularly interesting from our point of view. Firstly, the interface problem itself was not reduced in our case, to guarantee the interface kinematic compatibility. This results in a suboptimal reduced order model and, in the case of parallel computing, would generate expensive communications through the network. A reduction of the interface problem using the POD can be performed, associated with a system approximation that is similar to the one we have used in this paper. Alternatively, using a dual Schur-complement domain decomposition method would allow the kinematic approximation of the subproblems to include the interface as well. One then needs to identify a relevant Lagrange multiplier space to ensure optimality and stability of the Galerkin projection of the reference equations. This idea is our current direction of research.

An other difficulty is the load balancing mismatch that would occur when using such a strategy in parallel. CPUs which support domains that are not reduced, or domains for which the corresponding subproblem need to be projected in a space of relatively high dimension, would require to perform more operations. Hence, the domain partitioning itself should be performed jointly with the model reduction in order to distribute the load evenly.

Finally, we outlined throughout the paper some points that need further investigations but which are not directly related to the topic of reduced model partitioning addressed in this paper. The optimal choice of the snapshot samples used to construct *a posteriori* reduced order models is currently a very active research area (see for instance the review [47] concerning the reduced basis method, or the new developments proposed in [60] in the case of the snapshot POD). For arbitrary type of nonlinearity, a clear answer to this problem is, to date, not available. We have used a technique based on cross-validation, which, admittedly, requires a decently fine snapshot space in order to provide a relevant error estimate. In addition, our technique does not help find particular zones of non-smoothness in the parameter domain. It only provides a general trend for the projection error. Furthermore, an important point related to this issue is that the error criteria that have been used in this work are all based on global euclidean norms, without consideration for the physical phenomenon of interest. We believe that developing a "goal-oriented" domain-decomposition-based reduced order modelling would help alleviate a certain number of issues related to the certification of reduced order modelling for general nonlinearities.

Acknowledgements

The authors thank the financial support of EPSRC High End Computing Studentship for Mr. Olivier Goury as well as the support of Cardiff and Glasgow Universitys Schools of Engineering. Pierre Kerfriden and Stéphane Bordas would like to also acknowledge the financial support of the Royal Academy of Engineering and of the Leverhulme Trust for S. Bordas' Senior Research Fellowship entitled "Towards the next generation surgical simulator" which funded Pierre Kerfriden's post in 2009-2010 and allowed him to lay the foundations for the work presented in this paper.

References

[1] C. Antoulas and D.C. Sorensen. Approximation of large-scale dynamical systems: an overview. *International Journal of Applied Mathematics and Computer Science*, 11(5):1093–1121, 2001.

[2] L. Sirovich. Turbulence and the dynamics of coherent structures. part I: coherent structures. *Quarterly of Applied Mathematics*, 45:561–571, 1987.

[3] C.A. Beattie, J. Borggaard, S. Gugercin, and T. Iliescu. A domain decomposition approach to pod. *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.

[4] D. Amsallem and C. Farhat. An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.

[5] NC Nguyen, AT Patera, and J. Peraire. A 'best points' interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4):521–543, 2008.

[6] M. Buffoni, H. Telib, and A. Iollo. Iterative methods for model reduction by domain decomposition. *Computers & Fluids*, 38(6):1160–1167, June 2009.

[7] R. Craig and M. Bampton. Coupling of substructures for dynamic analysis. *American Institute of Aeronautics and Astronautics*, 6(7):1313–1319, 1968.

[8] J M Dickens, J M Nakagawa, and M J Wittbrodt. A critique of mode acceleration and modal truncation augmentation methods for modal response analysis. *Computers and Structures*, 62(6):985–998, 1997.

[9] M. Meyer and H.G. Matthies. Efficient model reduction in non-linear dynamics using the Karhunen-Loeve expansion and dual-weighted-residual methods. *Computational Mechanics*, 31(1):179–191, 2003.

[10] P. E. Barbone, D. Givoli, and I. Patlashenko. Optimal modal reduction of vibrating substructures. *International Journal for Numerical Methods in Engineering*, 57:341–369, 2003.

[11] D. Rixen. A dual craig-bampton method for dynamic substructuring. *Journal of Computational and Applied Mathematics*, 168:383–391, 2004.

[12] D. Markovic, K.C. Park, and A. Ibrahimbegovic. Reduction of substructural interface degrees of freedom in flexibility-based component mode synthesis. *International journal for numerical methods in engineering*, 70(2):163–180, 2007.

[13] WC Hurty. Vibrations of Structural Systems by Component Mode Synthesis. *Journal of the Engineering Mechanics Division*, 86(4):51–70, 1960.

[14] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

[15] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus de Mathématiques*, 339(9):667–672, 2004.

[16] P. G. Constantine and Q. Wang. Residual minimizing model interpolation for parameterized nonlinear dynamical systems. *SIAM Journal on Scientific Computing*, 34:118–144, 2012.

[17] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[18] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[19] D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1):75 – 89, 2008.

[20] D Ryckelynck and D M Benziane. Multi-level a priori hyper-reduction of mechanical models involving internal variables. *Computer Methods in Applied Mechanics and Engineering*, 199(17-20):1134–1142, 2010.

[21] P. Ladevèze, J.C. Passieux, and D. Néron. The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21):1287–1296, 2009.

[22] F. Chinesta, A. Ammar, and E. Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering - State of the Art Reviews*, 17(4):327–350, 2010.

[23] A. Nouy. A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1603–1626, 2010.

[24] J Yvonnet and Q.C. He. The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. *Journal of Computational Physics*, 223(1):341–368, 2007.

[25] P. Kerfriden, P. Gosselet, S. Adhikari, and S. Bordas. Bridging proper orthogonal decomposition methods and augmented Newton-Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):850–866, 2011.

[26] F. Galland, A. Gravouil, E. Malvesin, and M. Rochette. A global model reduction approach for 3D fatigue crack growth with confined plasticity. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):699–716, 2011.

[27] KC Park and YH Park. Partitioned component mode synthesis via a flexibility approach. *AIAA journal*, 42(5), 2004.

[28] Christian Rickelt and Stefanie Reese. A simulation strategy for life time calculations of large, partially damaged structures. In *III European Conference on Computational Mechanics*, 2006.

[29] P. Kerfriden, J.C. Passieux, and S. Bordas. Local/global model order reduction strategy for the simulation of quasi-brittle fracture. *International Journal for Numerical Methods in Engineering*, 89(2):154–179, 2011.

[30] S.G. Haryadi, R.K. Kapania, and S.G. Haryadi. Global/local analysis of composite plates with cracks. *Composites Part B*, 29(B):271–276, 1998.

[31] P.A. LeGresley and J.J. Alonso. Dynamic domain decomposition and error correction for reduced order models. *AIAA 41st Aerospace Sciences Meeting*, 2003.

[32] A Ammar, F Chinesta, and E Cueto. Coupling finite elements and proper generalized decompositions. *International Journal for Multiscale Computational Engineering*, 9(1):1–24, 2011.

[33] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM Journal on Numerical analysis*, 40(2):492–515, 2003.

[34] P. Astrid, S. Weiland, K. Willcox, and A.C.P.M. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, 2008.

[35] Manyu Xiao, Piotr Breitkopf, Rajan Filomeno Coelho, Catherine Knopf-Lenoir, Maryan Sidorkiewicz, and Pierre Villon. Model reduction by CPOD and Kriging. *Structural and Multidisciplinary Optimization*, 41:555–574, 2010.

[36] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[37] C. Farhat and F.X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.

[38] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233–241, 1993.

[39] P. Le Tallec. Domain decomposition methods in computational mechanics. In *Computational Mechanics Advances*, volume 1. Elsevier, 1994.

[40] P. Ladevèze and D. Dureisseix. A micro/macro approch for parallel computing of heterogeneous structures. *International Journal for computational Civil and Structural Engineering*, 1:18–28, 2000.

[41] N. Germain, J. Besson, F. Feyel, and P. Gosselet. High-performance parallel simulation of structure degradation using non-local damage models. *International journal for numerical methods in engineering*, 71(3):253–276, 2007.

[42] O. Allix, P. Kerfriden, and P. Gosselet. On the control of the load increments for a proper description of multiple delamination in a domain decomposition framework. *International Journal for Numerical Methods in Engineering*, DOI:10.1002/nme.2884, 2010.

[43] O. Lloberas-Valls, D.J. Rixen, Simone A., and L.J. Sluys. Domain decomposition techniques for the efficient modeling of brittle heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 200:1577–1590, 2011.

[44] P.-A. Guidault, O. Allix, L. Champaney, and S. Cornuault. A multiscale extended finite element method for crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 197(5):381–399, 2008.

[45] O. Allix, P. Kerfriden, and P. Gosselet. A relocalization technique for the multiscale computation of delamination in composite structures. *Computer Modeling in Engineering and Sciences*, 55(3):271–291, 2010.

[46] J-C. Jouhaud T. Braconnier, M. Ferrier and P. Sagaut. Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers and Fluids*, 40(1):195–209, 2011.

[47] A Quarteroni, G Rozza, and A Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(3):1–44, 2011.

[48] Y Chen and J White. A Quadratic Method for Nonlinear Model Order Reduction. In *Proceeding of the international Conference on Modeling and Simulation of Microsystems*, pages 470–480, 2000.

[49] M Rewienski and J White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):155–170, 2003.

[50] S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Real-time deformable models of non-linear tissues by model reduction techniques. *Computer Methods and Programs in Biomedicine*, 91(3):223–231, 2008.

[51] S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32:2737–2764, 2010.

[52] P.A. LeGresley and J.J. Alonso. Investigation of non-linear projection for pod based reduced order models for aerodynamics. *AIAA paper*, 926:2001, 2001.

[53] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *Journal of Computational Physics*, 202(1):346–366, 2005.

[54] P. Astrid. Reduction of process simulation models-a proper orthogonal decomposition approach. *PhD report, Technical University of Eindhoven*, 2004.

[55] J. Lemaître and J.-L. Chaboche. *Mechanics of Solid Materials*. Cambridge University Press, 1990.

[56] D. Ryckelynck, D. Missoum Benziane, S. Cartel, and J. Besson. A robust adaptive model reduction method for damage simulations. *Computational Materials Science*, 50(5):1597–1605, 2011.

[57] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13:515–572, 2006.

[58] Y. Saad and M.H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[59] H.A. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13:631, 1992.

[60] K. Kunisch and S. Volkwein. Optimal snapshot location for computing pod basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):509–529, 2010.

[61] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[62] W. J. Krzanowski. Cross-validation in principal component analysis. *Biometrics*, 43(3):575–584, 1987.

[63] P. Kerfriden, K.M. Schmidt, T. Rabczuk, and Bordas S.P.A. Statistical extraction of process zones and representative subspaces in fracture of random composites. *Accepted for publication in International Journal for Multiscale Computational Engineering, arXiv:1203.2487v2*, 2012.

[64] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact newton methods. *SIAM Journal on Numerical Analyses*, 19(2):400–408, 1982.