Computational time savings in Multiscale Fracture Mechanics using Model Order Reduction

Olivier Goury

Supervisors: Dr. Pierre Kerfriden
Prof. Stéphane Bordas

A thesis submitted to the graduate school
in fulfilment of the requirements for the degree of
Doctor of Philosophy

Advanced Materials and Computational Mechanics Group
Cardiff School of Engineering

Cardiff, Wales, United Kingdom

February 23, 2015

Summary

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. This local nature of fracture leads to large numerical models. Projection-based reduced order modelling is an increasingly popular technique for the fast solution of parametrised problems. However, traditional model order reduction methods are unable to reliably deal with either the initiation or the propagation of a crack or a local zone with high damage concentration.

In this thesis, we look at the general problem of applying model order reduction to fracture/damage mechanics, in the pursuit of rationalising the computational time involved in these kind of simulations. The first contribution of this thesis is the development of a reduced-order modelling for computational homogenisation, which is a general multi-scale method used to take microscopic data into account when deriving an engineering-scale model. A specific strategy is used to reduce the cost of solving the representative element volume (RVE) boundary value problem traditionally formulated in this method.

The second contribution was made by developing a partitioned reduced-order procedure for the case of parametrised nonlinear material deformations involving a local lack of correlation, which typically happens with fracture. The method allows to reduce the regions undergoing little non-linearities whilst computational work can be concentrated on regions of high non-linearity.

Declaration

DECLARATION
This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed.....(candidate) Date:.....

STATEMENT 1
This thesis is being submitted in partial fulfillment of the requirements for the degree of PhD.

Signed.....(candidate) Date:.....

STATEMENT 2
This work is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed.....(candidate) Date:.....

DECLARATION
I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed.....(candidate) Date:.....

Acknowledgements

I would like to acknowledge my supervisors Dr. Pierre Kerfriden and Prof. Stéphane Bordas for their assistance, guidance, and support over the duration of my Phd. I greatly appreciated the informality of our relationship, leading to a relaxed working atmosphere.

I would also like to acknowledge Prof. Wing Kam Liu and his research group, who welcomed me in Chicago for a 9-month research visit.

I am also grateful for the many friends I have made through the research lab in Cardiff, who are too many to name here. The mutual support we gave to each other as well as the good times we shared outside the office was essential to keep motivated and excited for research.

Contents

1 Introduction **1**

1.1 Motivation and strategies 1

1.2 Aims of the thesis and outline 2

2 Reduced-order modelling methods **5**

2.1 Projection-based model order reduction 5

2.1.1 Component-mode synthesis 6

2.1.2 Reduced-basis method 6

2.1.3 Snapshot-POD (Proper Orthogonal decompostion) 8

2.1.4 Proper Generalised decompostion (PGD) 10

2.1.5 Nonlinear Locally Isometric Embedding 12

2.2 System approximation methods 13

2.2.1 Collocation-type strategies 14

2.2.2 Discrete Empirical Interpolation method 15

2.2.3 Energy Conservation and Weighting method (ECWM) 17

3 Development of a reduced order model of nonlinear heterogeneous materials **19**

3.1 Introduction 19

3.2 Computational homogenisation setting 22

3.2.1 RVE boundary value problem 23

3.2.2 Scale coupling 25

3.2.3 Space discretisation and Newton solution algorithm 26

3.2.4 Parametrised RVE problem: description of the macroscopic load 26

3.3 Reduction of the RVE boundary value problem 27

3.3.1 Galerkin projection of the governing equations in a reduced space 30

3.3.2 A first model reduction approach using Snapshot POD on a snapshot randomly generated . . . 31
3.3.3 Model reduction using a POD-greedy algorithm . . . 38
3.4 Example and Numerical Results . . . 43
3.4.1 Snapshot-POD reduced space and numerical results . . . 44
3.4.2 Application of the POD-greedy algorithm . . . 51
3.5 Conclusion and perspectives of this chapter . . . 53

4 Partitioned model order reduction applied to fracture mechanics 55
4.1 General problem statement . . . 60
4.2 Model Order Reduction and Proper Orthogonal Decomposition . . . 62
4.2.1 Projection-based model order reduction . . . 62
4.2.2 Proper Orthogonal Decomposition in projection-based model order reduction . . . 64
4.2.3 System approximation . . . 68
4.2.4 Example of application of the POD in fracture mechanics . . . 71
4.3 Partitioned model order reduction approach . . . 78
4.3.1 Principle of the primal Schur-based domain decomposition method 78
4.3.2 Formulation of reduced order modelling in the domain decomposition framework . . . 82
4.3.3 Local error estimation by Cross-Validation . . . 85
4.4 System approximation in the partitioned model order reduction approach 89
4.4.1 Local "gappy" approximations . . . 89
4.4.2 Construction of the system approximation . . . 91
4.5 Results . . . 95
4.5.1 Online numerical costs ("speed-up") . . . 95
4.5.2 Remarks about the numerical efficiency of the system approximation . . . 101
4.6 Conclusion and perspectives of this chapter . . . 102

5 Conclusion and Discussion 105

A POD using an energy-based scalar product 109

B Partitioned model order reduction in a dual domain decomposition framework **111**

B.1 Local equilibrium 111

B.2 Standard FETI formulation 112

B.3 Reduced FETI formulation 114

B.3.1 Reduction of the displacement 114

B.3.2 Reduction of the traction forces 115

B.4 Computational gain 117

CHAPTER 1

Introduction

1.1 Motivation and strategies

Engineers have an interest in simulating the behaviour of materials under various conditions (loading, temperature, electromagnetic field, etc...). This can be used for different purposes, such as structure design, predicting the failure of materials, or more generally understanding the mechanisms involved in the material deformations. Constitutive laws, which define the physics of the materials, are used to derive equations that monitor the material deformations by establishing a relationship between stresses and strains inside the material. These constitutive laws, different for every material type, depend on various structural parameters (Young's modulus, Poisson's ratio,...) that can be calibrated from experiments. These laws can have various level of complexity (elasticity, hyperelasticity, plasticity, viscoplasticity, damage, etc...) and be implicitly defined. Since an analytical solution is not available in general, those equations are solved numerically with the use of an appropriate method: finite element method, meshless method, finite difference method, molecular dynamics... In principle, these methods produce satisfying results in reasonable computational time, providing that the constitutive law chosen to model the material of interest corresponds to reality and captures well its expected behaviour, and that the underlying discretisation of the material does not lead to a prohibitive number of unknowns.

However, real-life 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. This implies that fracture mechan-

ics for nonlinear microscopically heterogeneous materials is computationally extremely expensive in general:

- Fracture mechanics typically involves sharp local gradients: a very fine spatial discretisation is required to represent those gradients correctly
- A fine time discretisation is required to guarantee the stability of the simulation
- The microscopic nature of the material means that the macroscopic constitutive law is unknown in general. A multiscale method is needed to obtain a coarse-scale stress/strain relationship. For nonlinear material models, stress/strain relationship varies with time and space inside the material, and therefore, this relationship needs to be evaluated a large number of times. This has a formidable computational expense.
- In engineering-design processes, a prohibitively high number of solutions might be of interest, for a range of values of design parameters, or take randomness of the material into account.

Therefore, there is a need for developing efficient computational strategies to tackle the numerical complexity of fracture/damage mechanics in parametric multiscale problems.

Projection-based reduced order modelling is an increasingly popular technique for the fast solution of parametrised problems and is a potential solution to the computational complexity of fracture mechanics. The key idea is to represent the parametric variations of the solution in a low-dimensional subspace. This subspace can be identified using the snapshot-POD [1, 2, 3, 4, 5, 6, 7, 8], which compresses the posterior information contained in an exhaustive sampling of the parameter domain, or the Reduced Basis Method [9, 10, 11, 12, 13], which searches for this attractive subspace in the form of a linear combination of samples chosen quasi-optimally via a Greedy algorithm ("offline stage").

1.2 Aims of the thesis and outline

In this thesis, we look at the general problem of applying model order reduction to fracture/damage mechanics, in the pursuit of rationalising the computational time involved in these kind of simulations. Traditional model order reduction methods are unable to reliably deal with either the initiation or the propagation of a crack or a local

zone with high damage concentration, and this thesis investigates new model order reduction methods to deal with these challenges. We will look at both problems of crack initiation and propagation.

- Simulating the initiation of a crack in a microscopically heterogeneous material with an unknown constitutive law involves a multiscale strategy, which is known for having a formidable computational cost. The first contribution of this thesis is the development of a reduced-order modelling for computational homogenisation. A specific strategy is used to reduce the cost of solving the representative element volume (RVE) boundary value problem traditionally formulated in this method. One of the specific difficulties of that problem when using a history-dependent constitutive law (damage law in this case), is that the parameter space is the ensemble of possible far-field loads applied onto the RVE, which is of very large dimension and hence requires a well-thought sampling. A greedy algorithm was developed to gradually select a loading path that brings the most significant information.

- To simulate accurately the propagation of a crack, fine discretisation in both space and time is necessary, which again implies prohibitive computational efforts. The second contribution of thesis is made by developing a reduced-order strategy for the case of parametrised nonlinear material deformations involving a local lack of correlation, such as fracture. In this context, a standard projection-based reduced order model strategy performs typically poorly over the process zone area. A partitioned reduced-order framework was introduced for tackling this issue, where regions undergoing little non-linearities can be largely reduced and computational work can be concentrated on regions of high non-linearity.

This dissertation is organised as follows: in Chapter 2, a literature review on the various reduced order modelling is given. Chapter 3 deals with the application of model order reduction to the representative element volume (RVE) boundary value problem in the framework of computational homogenisation. In Chapter 4, the focus is made on a partitioned model order reduction strategy. Finally, Chapter 5 contains the global conclusion of this thesis, and possible further developments.

CHAPTER 2

Reduced-order modelling methods

In this chapter, we present various common methods used in the reduced order modelling community. We make a distinction between two main categories: projection-based methods and interpolation methods.

2.1 Projection-based model order reduction

For a general mechanical problem, the equilibrium equation to be solved is:

$$\mathbf{f_{int}}(\mathbf{x}) + \mathbf{f_{ext}} = \mathbf{0}, \tag{2.1}$$

where the vector $\mathbf{x}$ represents the degrees of freedom of the mechanical system (typically representing displacement in solid mechanics). The vector field $\mathbf{f_{int}}$ represents the internal forces created by the displacement field and $\mathbf{f_{ext}}$ the external forces applied on the solid. In some cases, the dimension of $\mathbf{x}$ can be large, and the computational time involved in solving this problem by numerical methods can be tremendous. The concept of reduced-order modelling is to define a surrogate system of smaller dimension:

$$\widetilde{\mathbf{f_{int}}}(\mathbf{x}_\mathrm{r}) + \widetilde{\mathbf{f_{ext}}} = \mathbf{0}, \tag{2.2}$$

that retains the important features of 2.1. $\widetilde{\mathbf{x}_\mathbf{r}}$ is the reduced state variable. The reduced system 2.2 is computationally much easier to solve. In the following, we present various ways of defining the reduced system from the literature.

2.1.1 Component-mode synthesis

In the component-mode synthesis [14, 15], after dividing the system of interest into several pieces or components, an approximation of each component is made by projecting the state variables onto some pre-calculated modes. Those modes are typically the normal modes of the structure (that contains the dynamics of the structure). Those can be enriched by static modes coming from constraints or attachments.

2.1.2 Reduced-basis method

The reduced-basis method is typically used in the context of parametrised partial differential equations [9, 10, 11, 12, 13]. $u(\mu)$ is the unknown field satisfying a partial differential equation parametrised by $\mu \in \mathcal{P}$, the parameter space. A quantity of interest $s(\mu)$ is defined by applying a linear functional l. The domain of interest is Ω. Standard assumptions of continuity and integrability are made: u belongs to the Sobolev space $\mathcal{H}^1(\Omega)$. The problem is discretised using a standard finite element scheme. We introduce v the test function that belongs to $\mathcal{H}^1_0(\Omega)$ (v is 0 on the boundary $\partial\Omega$). The weak form then reads:

$$\begin{cases} \forall v \in \mathcal{H}^1_0(\Omega), \quad a\left(u, v; \mu\right) = f(v; \mu) \\ s(\mu) = l\left(u_h(\mu); \mu\right), \end{cases} \tag{2.3}$$

where $a(.,.;\mu)$ is a bilinear form, and f and l are linear forms. a is also assumed coercive with the coercivity constant $\alpha(\mu) = \frac{a(w,w;\mu)}{\|w\|^2}$. For some specific cases, the advantageous assumption that a has an affine expansion can be made:

$$a\left(u_1, u_2; \mu\right) = \sum_{q=1}^{Q} \Theta^q(\mu) a^q(u_1, u_2). \tag{2.4}$$

The key benefit is that, under this assumption, a^q does not depend on μ, which means it can be computed during an offline stage once and for all. When the partial differential equation is describing the behaviour of a non-linear material, this assumption is typically not true, and another layer of approximation has to be made to enable a computational gain in this method (such as system approximation which is described in section 2.2).

After discretisation following the finite element method, the discretised weak form

reads (with $\mathcal{V} \subset \mathcal{H}^1(\Omega)$ and $\mathcal{V}_0 \subset \mathcal{H}^1_0(\Omega)$ being the discretised finite element space):

$$\begin{cases} \forall v_h \in \mathcal{V}_0, \quad a\left(u_h, v_h; \mu\right) = f(v_h; \mu) \\ s(\mu) = l\left(u_h(\mu); \mu\right), \end{cases} \tag{2.5}$$

Now, the principle of the reduced basis method is to express the solution field $u_h(\mu)$ as linear combination of some precomputed solution $\left(u_h(\mu_k)\right)_{k\in[\![1,N]\!]}$ evaluated for a well-chosen set of parameters $(\mu_k)_{k\in[\![1,N]\!]}$ so that $u_N(\mu) \in W_N = vect\left(u_h(\mu_1), \ldots, u_h(\mu_N)\right)$.

The problem is to find the optimal set of parameters that leads to the most accurate approximation of the original finite element solution. This is done through a greedy algorithm, based on an a posteriori error bound of the output noted $\Delta_N(\mu)$, where N shows the dependency of that error bound on the order of the reduced basis approximation. The derivation of the error estimate Δ_N relies on the coercivity constant $\alpha(\mu)$. It is inexpensive to evaluate in the case a has an affine expansion. When this isn't the case, an error indicator based on an approximated evaluation of the residual $r\left(v_h; \mu\right) = f(v_h; \mu) - a\left(u_N(\mu), v_h; \mu\right)$ has be computed as a surrogate of the error estimate (see for example [**?**]).

Given an arbitrary starting parameter value μ_1, the first vector of the reduced basis $u_h(\mu_1)$ is computed. The basis is then iteratively enriched by finding the parameter value that maximises the error bound, providing what is expected to be the most informative solution. To this purpose, a subset Ξ of the parameter space $\mathcal{P}$ is defined. This subset serves as a discrete surrogate of the continuous space $\mathcal{P}$, and has to be rich. Now, at step i of the greedy algorithm, a new solution u_h is evaluated at:

$$\mu_i = \underset{\mu\in\Xi}{\operatorname{argmax}}\, \Delta_i(\mu), \tag{2.6}$$

and the reduced basis is enriched with the new solution $u_h(\mu_i)$. This procedure carries on until the value of $\max\limits_{\mu\in\Xi} \Delta_i(\mu)$ is below a defined tolerance, at which point the reduced basis is judged rich enough.

For conditioning purposes, the reduced basis $W_{N_{\max}} = \left(u_h(\mu_1), \ldots, u_h(\mu_{N_{\max}})\right)$ is often orthogonalised with a Gram-Schmidt method, which means $W_{N_{\max}}$ can be written as $W_{N_{\max}} = (\zeta_1, \ldots, \zeta_{N_{\max}})$ with $(\zeta_1, \ldots, \zeta_{N_{\max}})$ forming an orthonormal basis. Then, the reduced basis expansion reads $u_r = \sum u_N^k \zeta_k$. Substituting this expansion into equation (2.5), using the affine expansion of a (2.4) and using the bilinearity of a and the linearity

of l, we can write:

$$\forall \zeta_m, m \in [\![1, N]\!], \quad \sum_{k=1}^{N} \left(\sum_{q=1}^{Q} \Theta^q(\mu) a^q \left(\zeta_k, \zeta_m; \right) \right) u_N^k(\mu) = f(\zeta_m), \tag{2.7}$$

where v was substituted by any ζ_m. The quantities $a^q \left(\zeta_k, \zeta_m; \right)$ and $f(\zeta_m)$ can be precomputed in the offline stage, which means the online stage will only consist of building and solving a much smaller version than the original.

2.1.3 Snapshot-POD (Proper Orthogonal decompostion)

The snapshot-POD method can be applied to a wide range of applications in various fields from fluid dynamics to solid mechanics. It is very attractive as it does not require any knowledge of the system studied as it builds the reduced model from an output dataset of the system of interest, generated "offline", i.e. prior to the reduced order simulation. This dataset is called the snapshot. Considering a general parametrised mechanical problem equilibrium:

$$\mathbf{f_{int}} \left(\mathbf{u}(t, \boldsymbol{\mu}) , \boldsymbol{\mu} \right) + \mathbf{f}_{\text{ext}}(\boldsymbol{\mu}) = \mathbf{0}, \tag{2.8}$$

with $\mathbf{u}$ representing the displacement and $\boldsymbol{\mu}$ the parameter. The method is based on the approximation of $\mathbf{u}\left(t, \boldsymbol{\mu}\right)$ by a linear expansion based on a parameter-independent basis $(\boldsymbol{\phi_1}, \boldsymbol{\phi_2}, \ldots, \boldsymbol{\phi_N}) = \boldsymbol{\Phi}$:

$$\mathbf{u}(t, \boldsymbol{\mu}) \approx \widetilde{\mathbf{u}}\left(t, \boldsymbol{\mu}\right) = \sum_{i=1}^{N} \boldsymbol{\phi}_i \alpha_i \left(t, \boldsymbol{\mu}\right) = \boldsymbol{\Phi} \boldsymbol{\alpha} \left(t, \boldsymbol{\mu}\right), \tag{2.9}$$

where $\boldsymbol{\alpha}\left(t, \boldsymbol{\mu}\right)$ is the vector of reduced state variable, of dimension N. The POD looks for an orthogonal basis which on average, best represent the solution of 2.8, as the parameter $(t, \boldsymbol{\mu})$ varies in $\mathcal{T} \times \mathcal{P}$, by minimising the following cost function:

$$J\left(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_N}\right) = \int_{\mu} \int_{t} \left\| \mathbf{u}\left(t; \boldsymbol{\mu}\right) - \sum_{k=1}^{N} \boldsymbol{\phi_k} \boldsymbol{\phi_k}^T \mathbf{u}\left(t; \boldsymbol{\mu}\right) \right\|_2^2 d\boldsymbol{\mu}. \tag{2.10}$$

The cost function evaluates the sum of the differences between the exact solution $\mathbf{u}(t; \boldsymbol{\mu})$ and its projection onto the basis $\boldsymbol{\Phi}$. The basis $\boldsymbol{\Phi}$ is hierarchical in the sense that the

vector $\boldsymbol{\phi_1}$ gives the best [1] approximation of the solution with 1 vector, $[\boldsymbol{\phi_1}, \boldsymbol{\phi_2}]$ the best approximation of the solution with 2 vectors and so on.

This problem requires the exact solution over the entire time-parameter domain $\mathcal{T} \times \mathcal{P}$ and has no practical use. However, it can be approached by calculating a number of solutions for certain values of the time and the parameter. Assume now that a dataset of many output solutions $(\mathbf{s_1} \dots \mathbf{s_p}) = \mathbf{S}$ corresponding to a certain discrete subset of parameters $\mathcal{P}^{\mathrm{s}} \subset \mathcal{P}$ and a discrete subset of timesteps $\mathcal{T}^{\mathrm{s}} \subset \mathcal{T}$ is available. A discrete approximation of the cost function 2.10 reads:

$$J^{\mathrm{s}}(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_N}) = \sum_{i=1}^{p} \sum_{t \in \mathcal{T}^{\mathrm{s}}} \|\mathbf{s}_i - \sum_{k=1}^{N} \boldsymbol{\phi_k} \boldsymbol{\phi_k}^T \mathbf{s}_i\|_2^2 \,. \tag{2.11}$$

Adding the constraint of orthonormality of the basis $\boldsymbol{\Phi}$, the minimisation problem solved by the POD is:

$$\begin{cases} \min\limits_{\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_l}} J^{\mathrm{s}}(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_N}) \\ \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_N} \rangle = \delta_{ij} \,, \end{cases} \tag{2.12}$$

This problem can expressed in an unconstrained manner by recasting it as the minimisation of the Lagrangian:

$$\mathcal{L}(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_N}, \boldsymbol{\mu}) = J^{\mathrm{s}}(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_N}) + \sum_{i,j}^{N} \lambda_{ij} \left(\langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle - \delta_{ij} \right) \,. \tag{2.13}$$

The necessary optimality conditions read:

$$\frac{\partial \mathcal{L}}{\partial \mu_{ij}}(\boldsymbol{\Phi}, \boldsymbol{\mu}) = \boldsymbol{\phi_i}^T \boldsymbol{\phi_j} - \delta_{ij} = 0 \in \mathbb{R}, \quad \forall\, i, j < N \tag{2.14}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi_k}}(\boldsymbol{\Phi}, \boldsymbol{\mu}) = \nabla_{\boldsymbol{\phi_k}} \mathcal{L}(\boldsymbol{\Phi}, \boldsymbol{\mu}) = -2 \sum_{i} \mathbf{s_i} \left(\boldsymbol{\phi_k}^T \mathbf{s_i} \right) + \sum_{i=1}^{l} (\lambda_{ik} + \lambda_{ki}) \boldsymbol{\phi_i}, \quad \forall\, i < N. \tag{2.15}$$

One can show that this is equivalent to:

$$\mathbf{S}\mathbf{S}^T \boldsymbol{\phi_k} = \lambda_{kk} \boldsymbol{\phi_k} \quad \text{and} \quad \lambda_{ij} = 0 \quad \forall i \neq j \,, \tag{2.16}$$

which is a simple symmetric eigenvalue problem. It can be solved using a singular value

[1] In an average sense

decomposition:

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T, \tag{2.17}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma}$ a rectangular matrix containing the (hierarchically ordered) singular values of $\mathbf{S}$ on its diagonal. An approximation of $\mathbf{S}$ of rank i^{th} order is given by $\mathbf{U}\widehat{\mathbf{\Sigma}}\mathbf{W}^T$, where $\widehat{\mathbf{\Sigma}}$ is given by the truncation $\mathbf{\Sigma}$ at order k (only the first k singular values of $\mathbf{S}$ remains on the diagonal). In this decomposition, the first k columns of $\mathbf{U}$ forms the optimal basis of order k, i.e. $\forall\, k$, $\boldsymbol{\phi_k} = \mathbf{u_k}$.

Using the orthonormality of the basis $\mathbf{\Phi}$, one can now determine a simple error formula the POD basis of order l:

$$J^{\mathrm{s}}_{\langle\cdot\rangle}(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_l}) = \sum_i \|\mathbf{s}_i - \sum_{k=1}^{l} \boldsymbol{\phi_k} \langle \boldsymbol{\phi_k}, \mathbf{s}_i \rangle \|^2 = \sum_i \sum_{k=l+1}^{n} \|\boldsymbol{\phi_k} \langle \boldsymbol{\phi_k}, \mathbf{s}_i \rangle\|^2 = \sum_{k=l+1}^{n} \lambda_{kk} \tag{2.18}$$

When normalised, we obtain the following error that represents how well the POD basis of order l approximates the snapshot $\mathbf{S}$:

$$\nu(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_l}) = \sqrt{\frac{\sum_{k=l+1}^{n} \lambda_{kk}}{\sum_{k=1}^{n} \lambda_{kk}}}. \tag{2.19}$$

This error estimate may be used to make a decision on the dimension of the reduced basis $\mathbf{\Phi}$.

2.1.4 Proper Generalised decompostion (PGD)

The PGD [16, 17, 18] is typically used with partial differential equations depending on a high number of parameters. It is based the principle of separation of variables. The solution is approximated by a sum of N functional products involving d functions that each depend only on a single variable:

$$u(x_1, \ldots, x_d) \approx \sum_{i=1}^{N} F_i^1(x_1) \times \ldots \times F_i^d(x_d) \tag{2.20}$$

As such, it can be seen as a generalisation of the Proper Orthogonal Decomposition which only separates the space dependence:

$$u(x,t,\boldsymbol{\mu}) = \sum_{i=1}^{N} \phi_i(x)\alpha_i(t,\boldsymbol{\mu}). \tag{2.21}$$

Here, the x_i's can be representing spatial coordinates, time, or any other parameter involved in the problem, such as boundary conditions, material parameters, etc... The functions $\left(F_i^k\right)_{i=[\![1,N]\!];k=[\![1,d]\!]}$ are unknown a priori and require an iterative procedure to be numerically identified. The method relies on writing the separated expansion in the weak formulation of the PDE. A partial differential equation can be written in the general form:

$$\mathcal{F}\left(x_1,\dots,x_d,u,\frac{\partial u}{\partial x_1},\dots,\frac{\partial u}{\partial x_d},\frac{\partial^2 u}{\partial x_1\partial x_1},\dots,\frac{\partial^2 u}{\partial x_1\partial x_d},\dots\right)=0 \tag{2.22}$$

The weak formulation then reads:

$$\int_{x_1\in\mathcal{X}_1}\dots\int_{x_d\in\mathcal{X}_d} u^\star\,.\,\mathcal{A}\left(x_1,\dots,x_d,u,\frac{\partial u}{\partial x_1},\dots,\frac{\partial u}{\partial x_d},\frac{\partial^2 u}{\partial x_1\partial x_1},\dots\right)dx_1\dots dx_d=0 \tag{2.23}$$

u is sought for in the PGD expansion form defined in equation (2.20). At step k of the PGD enrichment process, the solution is given by:

$$u^k(x_1,\dots,x_d)=\sum_{i=1}^{k} F_i^1(x_1)\times\dots\times F_i^d(x_d). \tag{2.24}$$

The solution at step $k+1$ is searched for as:

$$u^{k+1}(x_1,\dots,x_d)=\underbrace{\left(\sum_{i=1}^{k} F_i^1(x_1)\times\dots\times F_i^d(x_d)\right)}_{=u^k(x_1,\dots,x_d)}+F_{k+1}^1(x_1)\times\dots\times F_{k+1}^d(x_d). \tag{2.25}$$

This expansion can be substituted in the weak form (2.23). The test function $u^\star$ can be defined in various ways. One simple expansion reads:

$$u^\star(x_1,\dots,x_d)={F_{k+1}^1}^\star(x_1)\times\dots\times F_{k+1}^d(x_d)+\dots+F_{k+1}^1(x_1)\times\dots\times {F_{k+1}^d}^\star(x_d). \tag{2.26}$$

From there, a fixed point algorithm is usually used to identify the terms F_{k+1}^i of the

PGD expansion. Starting with an arbitrary value for each term (typically the value from the former enrichment step), the values F_{k+1}^i are updated by solving an associated problem one at a time. Starting by assuming the value of F_{k+1}^i is known for $i = 2, \dots, d$, $u^\star$ reduces to:

$$u^\star(x_1, \dots, x_d) = {F_{k+1}^1}^\star(x_1) \times \dots \times F_{k+1}^d(x_d), \tag{2.27}$$

and we can integrate (2.23) over $\mathcal{X}_2, \dots, \mathcal{X}_d$ which then leads to the 1-dimensional problem:

$$\int_{\mathcal{X}_1} {F_{k+1}^1}^\star(x_1) \cdot \widetilde{\mathcal{A}} \left(x_1, F_{k+1}^1(x_1), \frac{\partial F_{k+1}^1}{\partial x_1}, \dots \right) dx_1 = 0, \tag{2.28}$$

where $\widetilde{\mathcal{A}} = \int_{\mathcal{X}_2} \dots \int_{\mathcal{X}_d} F_{k+1}^2(x_2) \times \dots \times F_{k+1}^d(x_d) \cdot \mathcal{A}(\cdot, x_2, \dots, x_d)\, dx_2 \dots dx_d$. Solving this problem will give the value of F_{k+1}^1 over the set $\mathcal{X}_1$. The same procedure is applied to obtain each F_{k+1}^i from $\left(F_{k+1}^j\right)_{j \in [\![1,d]\!] \setminus i}$. This carries on until reaching convergence.

2.1.5 Nonlinear Locally Isometric Embedding

In the Nonlinear Locally Isometric Embedding (NLIE) [19], the high-dimensional data is approximated by a nonlinear manifold, rather than by a linear combination of basis functions as is the case in standard projection-based methods (such as POD/PCA, reduced basis,...). This means that in this case, the approximation of the solution will have a representation that is not purely a linear expansion. The reduced representation of the state variables has more flexibility and can lead to a representation that is of smaller dimensionality than a corresponding linear representation would be. Figure 2.1 illustrates this idea.

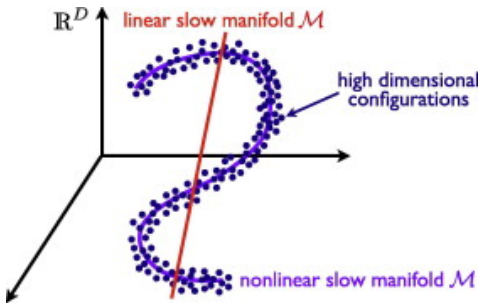


Figure 2.1: Figure taken from [19]. A non-linear representation of the system studied can have a significantly lower dimensionality than a traditional linear approximation

Assuming a set of data $(x^i)_{i \in \mathcal{S}}$ of solution of the problem of interest has been pre-computed, the expansion of the state variables is defined using a nonlinear mapping

$\mathbf{q}$:

$$\mathbf{x} = \sum_{i \in \mathcal{S}} p_i \left(\mathbf{q}(\mathbf{x}) \right) \mathbf{x}^{\mathbf{i}}. \tag{2.29}$$

$\mathbf{q}$ is an operator that maps the high dimensional data to a low-dimensional representation that is found using a partitioned version of the method called Isomap [20]. This method is based on a graph representation of the high-dimensional data based on euclidean distances between nearest neighbours. Shortest paths between 2 snapshots in this graph are then used to estimate a geodesic distance. The functions p_i are local basis functions encountered in meshless methods.

The reduced representation can then be substituted in the problem's equilibrium equation. In principle, this method will lead to a formulation of smaller dimension which has a potential for computational gain. However, the expansion being more complicated than a standard projection-based model order reduction method implies that the reduced equations will also be more complex to solve.

2.2 System approximation methods

In the previous section, various ways of reducing the order of a full order model was described. However, in the case of nonlinear materials, these approximations are inefficient and another layer of approximation is required. For a general parametrised mechanical problem, the equilibrium equation reads:

$$\mathbf{f}_{\text{int}} \left(\mathbf{u} \left(t, \boldsymbol{\mu} \right) ; \boldsymbol{\mu} \right) + \mathbf{f}_{\text{ext}} \left(\boldsymbol{\mu} \right) = \mathbf{0}, \tag{2.30}$$

where $\boldsymbol{\mu}$ is the parameter. When substituting the displacement $\mathbf{u} \left(t, \boldsymbol{\mu} \right)$ with the expansion $\boldsymbol{\Phi} \boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right)$, the residual can be defined as:

$$\mathbf{r} \left(\boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right) \right) = \mathbf{f}_{\text{int}} \left(\boldsymbol{\Phi} \boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right) ; \boldsymbol{\mu} \right) + \mathbf{f}_{\text{ext}} \left(\boldsymbol{\mu} \right), \tag{2.31}$$

where $\boldsymbol{\Phi}$ is the spatial basis of the reduced space (which is independent of time and parameter), and $\boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right)$ is the vector of reduced state variables, that depends on both time and parameter. The traditional corresponding reduced equilibrium equation with Galerkin projection reads:

$$\boldsymbol{\Phi}^T \mathbf{r} \left(\boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right) \right) = \mathbf{0} \Leftrightarrow \boldsymbol{\Phi}^T \mathbf{f}_{\text{int}} \left(\boldsymbol{\Phi} \boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right) ; \boldsymbol{\mu} \right) + \boldsymbol{\Phi}^T \mathbf{f}_{\text{ext}} \left(\boldsymbol{\mu} \right) = \mathbf{0}. \tag{2.32}$$

Despite the reduction of the number of the system unknowns, the complexity of the simulation is still dependent on the fine discretisation of the full model as structural operators defined over the domain still need to be constructed, such as internal forces, stiffness matrix, equilibrium residual. When in a nonlinear setting, these quantities vary and have no affine decomposition which means they can not be precomputed, and have to be evaluated at each step of the reduced simulation, thus leading to an insignificant time gain.

To alleviate this issue, system approximation methods propose to evaluate those structural quantities on only a few parts of the domain (in a finite element setting, these parts are a certain small subset of elements of the full mesh), called the reduced integration domain. Assuming the discretisation method chosen is FEM (though it could, in principle, be any other kind of discretisation), defined over a mesh of elements $\mathcal{E}$, the large and expensive to compute internal forces vector $\mathbf{f}_{\text{int}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)$ is only computed on a reduced integration domain (which is a subset of elements $\widetilde{\mathcal{E}} \subset \mathcal{E}$. This truncated vector of internal forces is denoted $\widehat{\mathbf{f}_{\text{int}}}$. We denote n_{sa} the number of degrees of freedom that are fully described by the subset of elements $\widetilde{\mathcal{E}}$ and n_{u} the number of degrees of freedom of the full model.

A boolean diagonal operator $\mathbf{P}$ is introduced which has n_{sa} non-zero entries only, corresponding to the evaluation degrees of freedom of the reduced integration domain $\widetilde{\mathcal{E}}$. $\mathbf{P}$ can be written $\mathbf{P} = \mathbf{E}^T\,\mathbf{E}$, with $\mathbf{E} \in \mathbb{R}^{n_{\text{sa}}} \times \mathbb{R}^{n_u}$ being the extractor that returns the entries corresponding to the reduced integration domain only, so that we have the equality:

$$\widehat{\mathbf{f}_{\text{int}}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) = \mathbf{E}\,\mathbf{f}_{\text{int}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right). \tag{2.33}$$

The number of controlled elements n_{sa} is usually much smaller than the number of degrees of freedom of the full order model, which allows for significant time gain. Using that reduced integration domain, a reduced system is formulated in different ways depending on the method, using this incomplete information only. Of course, a key part in the success of this approximation is the selection of reduced set of elements $\widetilde{\mathcal{E}}$. Again, the selection strategy vary with the different methods. In the following, we describe some of the main methods found the literature.

2.2.1 Collocation-type strategies

Early versions of Hyperreduction [21] and Missing point estimation [22] fall into this category. They propose to look for a solution that is optimal with respect to a few of

the equations of the initial system (4.2). This can be expressed in a Petrov-Galerkin framework:

$$\mathbf{\Phi}^T \mathbf{P} \left(\mathbf{f}_{\text{int}} \left(\mathbf{\Phi} \boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right) ; \boldsymbol{\mu} \right) + \mathbf{f}_{\text{ext}} \left(\boldsymbol{\mu} \right) \right) = \mathbf{0}. \tag{2.34}$$

Selection of the reduced integration domain In the Hyperreduction method [21], the controlled nodes are iteratively chosen as the largest entries of the average of basis function gradient onto each node. The idea that the basis function should be "observable" from the reduced integration domain, meaning that the restriction of the reduced basis onto the reduced integration domain should be forming a linearly independent set.

In the Missing point estimation [22], the controlled nodes are chosen so that they minimise the norm of the so-called "alias sensitivity operator", which is a measure of the error between a direct projection of the exact solution $\mathbf{u_r}$ onto the reduced space and its corresponding interpolation $\hat{\mathbf{u}}_\mathbf{r}$. This eventually leads to a problem of maximising the conditioning of the matrix $\widehat{\mathbf{\Phi}}^T \widehat{\mathbf{\Phi}}$, where $\widehat{\mathbf{\Phi}}$ is the restriction of the reduced basis to the reduced integration domain. A greedy algorithm is then used to construct the reduced integration domain by iteratively enriching the reduced domain with the node leading to the smallest condition number.

2.2.2 Discrete Empirical Interpolation method

The Discrete Empirical Interpolation method (DEIM) [23], a discrete version of Empirical Interpolation method (EIM) [10]. The nonlinear function $\mathbf{f}_{\text{int}}$ is reconstructed by interpolation over an other POD basis $\mathbf{\Psi}$ ("gappy" technique). The expansion reads:

$$\mathbf{f}_{\text{int}} \left(\mathbf{\Phi} \boldsymbol{\alpha} \left(t, \boldsymbol{\mu} \right) ; \boldsymbol{\mu} \right) = \sum_{i=1}^{n_\psi} \boldsymbol{\psi}_i \beta_i \left(t, \boldsymbol{\mu} \right) \tag{2.35}$$

$$= \mathbf{\Psi} \boldsymbol{\beta} \left(t, \boldsymbol{\mu} \right) \tag{2.36}$$

The columns of $\mathbf{\Psi} \in \mathbb{R}^{n_\text{u}} \times \mathbb{R}^{n_\psi}$ are spatial functions corresponding to a truncated snapshot POD expansion of the image of the reduced space by $\mathbf{f_{int}}$, 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 $\{\mathbf{f_{int}} \left(\mathbf{\Phi}\, \boldsymbol{\alpha} \right), \boldsymbol{\alpha}^\star \in \mathbb{R}^{n_\text{c}}\}$. Interpolation coefficients $\boldsymbol{\beta}$ are found by enforcing that at any point $(t, \boldsymbol{\mu})$, the interpolation must be optimal with respect to the limited number n_{sa} of spatial degrees

of freedom from the reduced integration domain $\widetilde{\mathcal{E}}$:

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}^\star \in \mathbb{R}^{n_\mathrm{d}}}{\operatorname{argmin}} \left(\| \boldsymbol{\Psi}\, \boldsymbol{\beta}^\star - \mathbf{f_{int}} \left(\boldsymbol{\Phi}\, \boldsymbol{\alpha}; \boldsymbol{\mu} \right) \|_{\mathbf{P}} \right) \tag{2.37}$$

$\|\mathbf{x}\|_{\mathbf{P}} = \sqrt{\mathbf{x}^T \mathbf{P}\, \mathbf{x}}$ is the semi-norm associated with $\mathbf{P}$ for an arbitrary vector $\mathbf{x} \in \mathbb{R}^{n_\mathrm{u}}$. Solving equation (2.37) leads to the value of $\boldsymbol{\beta} = \left(\boldsymbol{\Psi}^T \mathbf{P} \boldsymbol{\Psi} \right)^{-1} \boldsymbol{\Psi}^T \mathbf{f_{int}} \left(\boldsymbol{\Phi}\, \boldsymbol{\alpha}; \boldsymbol{\mu} \right)$. Substituting this approximation into the expansion (2.36), together with the reduced basis approximation for the displacement vector, the following reduced expression is obtained for the surrogate of the full vector of internal forces at any point of the time-parameter domain:

$$\mathbf{f_{int}} \left(\boldsymbol{\Phi}\, \boldsymbol{\alpha} \right) \approx \boldsymbol{\Psi} \left(\boldsymbol{\Psi}^T\, \mathbf{P}\, \boldsymbol{\Psi} \right)^{-1} \boldsymbol{\Psi}^T \mathbf{P}\, \mathbf{f_{int}} \left(\boldsymbol{\Phi}\, \boldsymbol{\alpha} \right), \tag{2.38}$$

where operator $\boldsymbol{\Psi}^T\, \mathbf{P}\, \boldsymbol{\Psi}$ is assumed to be invertible. One can now define the residual of the DEIM approximation:

$$\mathbf{r}_{\mathrm{gap}}(\boldsymbol{\alpha}) = \boldsymbol{\Psi} \left(\boldsymbol{\Psi}^T\, \mathbf{P}\, \boldsymbol{\Psi} \right)^{-1} \boldsymbol{\Psi}^T \mathbf{P}\, \mathbf{f_{int}} \left(\boldsymbol{\Phi}\, \boldsymbol{\alpha} \right) + \mathbf{f}_{\mathrm{ext}}. \tag{2.39}$$

Note that in the case the number of degrees of freedom in the reduced integration domain matches the dimension of the "static" basis $\boldsymbol{\Psi}$, this expression reduces to:

$$\mathbf{r}_{\mathrm{gap}}(\boldsymbol{\alpha}) = \boldsymbol{\Psi} \left(\mathbf{E}\, \boldsymbol{\Psi} \right)^{-1} \mathbf{E}\, \mathbf{f_{int}} \left(\boldsymbol{\Phi}\, \boldsymbol{\alpha} \right) + \mathbf{f}_{\mathrm{ext}}, \tag{2.40}$$

where we used the extractor $\mathbf{E}$, which verifies $\mathbf{E}^T\, \mathbf{E} = \mathbf{P}$, defined in (2.33).

The reduced variables can then be obtained in the "offline" phase by minimising the norm of this residual, or by solving the Galerkin projection of the governing equations $\boldsymbol{\Phi}^T \mathbf{R_{gap}}(\boldsymbol{\alpha}) = \mathbf{0}$. 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.

Selection of the reduced integration domain This method finds a set of degrees of freedom $\mathcal{I}$ in a greedy manner from the internal forces basis $\boldsymbol{\Psi}$. We quickly describe the method.

At iteration j of the greedy algorithm, $j-1$ points have been already selected. We define the extractor $\mathbf{E^j}$ that extracts those j selected degrees of freedom (i.e. for any vector $\mathbf{v}$, $\mathbf{E^j v}$ is a smaller vector containing only the j entries of $\mathbf{v}$ corresponding to the

selected degrees of fredom). The residual $\mathbf{r}^{\mathbf{j}}_{\text{gap}} = \left|\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}\boldsymbol{\beta}^{\mathbf{j}} - \boldsymbol{\psi}_{\mathbf{j+1}}\right|$ is evaluated, where $\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}$ is the matrix containing the first j vectors of the basis $\boldsymbol{\Psi}$ and $\boldsymbol{\psi}_{\mathbf{j+1}}$ is the $j+1^{th}$ vectors in that basis. $\boldsymbol{\beta}$ is the solution of the minimisation problem

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}^\star}{\operatorname{argmin}} \left\|\mathbf{E}^{\mathbf{j}}\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}\boldsymbol{\beta}^\star - \mathbf{E}^{\mathbf{j}}\boldsymbol{\psi}_{\mathbf{j+1}}\right\|_2 . \tag{2.41}$$

The solution is easily found: $\boldsymbol{\beta} = \left(\mathbf{E}^{\mathbf{j}}\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}\right)^{-1}\mathbf{E}^{\mathbf{j}}\boldsymbol{\psi}_{\mathbf{j+1}}$. The greedy procedure then selects the index of the highest entry in $\mathbf{r}^{\mathbf{j}}_{\text{gap}}$ as the $j+1^{th}$ control degree of freedom.

The motivation of this selection process is based on the following inequality, measuring the error between the exact vector of internal forces and its interpolation using DEIM:

$$\left\|\mathbf{f_{int}}\left(\boldsymbol{\Phi}\,\boldsymbol{\alpha}\right) - \widetilde{\mathbf{f_{int}}}\left(\boldsymbol{\Phi}\,\boldsymbol{\alpha}\right)\right\|_2 \leq \left|\left|\left|\left(\mathbf{E}\boldsymbol{\Psi}\right)^{-1}\right|\right|\right|_2 \left\|\left(\mathbf{I} - \boldsymbol{\Psi}\,\boldsymbol{\Psi}^T\right)\mathbf{f_{int}}\left(\boldsymbol{\Phi}\,\boldsymbol{\alpha}\right)\right\|_2 , \tag{2.42}$$

where $\widetilde{\mathbf{f_{int}}}\left(\boldsymbol{\Phi}\,\boldsymbol{\alpha}\right) = \boldsymbol{\Psi}\left(\mathbf{E}\,\boldsymbol{\Psi}\right)^{-1}\mathbf{E}\,\mathbf{f_{int}}\left(\boldsymbol{\Phi}\,\boldsymbol{\alpha}\right)$ is the "cheap" DEIM approximation of $\mathbf{f_{int}}\left(\boldsymbol{\Phi}\,\boldsymbol{\alpha}\right)$. It can be proven that the DEIM selection controls the growth of the matrix norm $\left|\left|\left|\left(\mathbf{E}\boldsymbol{\Psi}\right)^{-1}\right|\right|\right|_2$ at each iteration of the procedure. At the end of the greedy algorithm, the number of control degrees of freedom chosen equals the number of basis vectors $\left(\boldsymbol{\psi}_{\mathbf{i}}\right)_{n_{\text{gap}}}$ which makes system (3.26) well defined.

2.2.3 Energy Conservation and Weighting method (ECWM)

Rather than approximating the full order version of some internal material quantity (such as the internal forces), the ECWM [24] focuses on approximating the reduced version of these forces which is actually the quantity that is used when solving the reduced order model equations.

The reduced internal forces reads:

$$\mathbf{f}^{\text{r}}_{\text{int}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) = \boldsymbol{\Phi}^T\mathbf{f}_{\text{int}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right). \tag{2.43}$$

In a finite element context, the reduced internal forces can be reformulated at the element level:

$$\mathbf{f}^{\text{r}}_{\text{int}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) = \sum_{e\in\mathcal{E}}\boldsymbol{\Phi}_e^T\mathbf{f}_{\text{int}}\left(\boldsymbol{\Phi}_e\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right), \tag{2.44}$$

where, $\mathcal{E}$ is the set of elements, and $\boldsymbol{\Phi}_e$ is the restriction of $\boldsymbol{\Phi}$ to the nodal entries in contact with element e. Now, each row of the reduced forces $\mathbf{f}^{\text{r}}_{\text{int}}$ can be interpreted

as the virtual work performed by $\mathbf{f}_{\text{int}}$ along each column of $\boldsymbol{\Phi}$, seen as a virtual displacement. Under this light, approximating the reduced forces $\mathbf{f}_{\text{int}}^{\text{r}}$ can be interpreted as approximating a sum of energy contributions (from (2.45)). The idea is then to approximate that global energy by the sum of the energy contributions of a small subset of elements $\widetilde{\mathcal{E}}$ only, weighted by a positive scalar ξ_e.

The ECWM approximation then reads:

$$\widetilde{\mathbf{f}_{\text{int}}^{\text{r}}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) \approx \mathbf{f}_{\text{int}}^{\text{r}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) = \sum_{e\in\widetilde{\mathcal{E}}} \xi_e \boldsymbol{\Phi}_e^T \mathbf{f}_{\text{int}}\left(\boldsymbol{\Phi}_e\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right). \tag{2.45}$$

Note that this formulation implies that the computation of the ECWM approximation $\widetilde{\mathbf{f}_{\text{int}}^{\text{r}}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t,\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)$ involves looping over the reduced set of elements $\widetilde{\mathcal{E}}$ which will lead to significant time savings.

Now the reduced set of elements $\widetilde{\mathcal{E}}$ and their associated weights $(\xi_e)_{e\in\widetilde{\mathcal{E}}}$ need to be determined. Assuming the internal forces quantities are precomputed over the entire set of elements $\mathcal{E}$ for a certain set $\mathcal{S}$ of pair of parameter values and timesteps $\left(\left(t^i,\boldsymbol{\mu}^{\boldsymbol{i}}\right)\right)_{i\in\mathcal{S}}$, the vector of weights can searched for as the solution of the following non-negative least square problem:

$$\boldsymbol{\xi} = \underset{\boldsymbol{\xi}^\star>\mathbf{0}}{\operatorname{argmin}} \left\|\mathbf{G}\boldsymbol{\xi}^\star - \mathbf{b}\right\|_2, \tag{2.46}$$

where $\mathbf{G} = \left(\mathbf{G}_{ie}\right)_{\left(i\in\mathcal{S},e\in\widetilde{\mathcal{E}}\right)}$ is a block matrix with $\mathbf{G}_{ie} = \boldsymbol{\Phi}_e^T\mathbf{f}_{\text{int}}\left(\boldsymbol{\Phi}_e\boldsymbol{\alpha}\left(t^i,\boldsymbol{\mu}^{\boldsymbol{i}}\right);\boldsymbol{\mu}^{\boldsymbol{i}}\right)$ and $\mathbf{b} = \left(\mathbf{b}_i\right)_{i\in\mathcal{S}}$ with $\mathbf{b}_i = \mathbf{f}_{\text{int}}^{\text{r}}\left(\boldsymbol{\Phi}\boldsymbol{\alpha}\left(t^i,\boldsymbol{\mu}^{\boldsymbol{i}}\right);\boldsymbol{\mu}^{\boldsymbol{i}}\right) = \sum_{e\in\mathcal{E}}\mathbf{G}_{ie}$.

Of course, an exact solution for minimisation problem 2.46 is given by a weight vector $\boldsymbol{\xi}$ identically equal to 1 for all entries. This solution present no interest as it would imply that the reduced integration domain $\widetilde{\mathcal{E}}$ just matches the full domain $\mathcal{E}$. Instead, the ECWM proposes to find a weight vector with a minimal amount of non-zeros entries so that

$$\left\|\mathbf{G}\boldsymbol{\xi} - \mathbf{b}\right\|_2 \leq \tau\|b\|_2, \tag{2.47}$$

with τ a tolerance defined by the user.

Finding the vector $\boldsymbol{\xi}$ with the minimal amount of non-zero entries that satisfy this inequality is extremely complicated, and in practice, the weights are found following a greedy procedure. The weight vector is initialised to be zero, and is iteratively "enriched" until inequality 2.47 is verified, providing a suboptimal but usually acceptable solution.

CHAPTER 3

Development of a reduced order model of nonlinear heterogeneous materials

3.1 Introduction

Multiscale modelling permits to take into account partial microscopic data when deriving engineering-scale working models. In solid mechanics, homogenisation is routinely used to obtain coarse-scale stress/strain relationships that are consistent with some statistical knowledge of the microstructure [25, 26, 27, 28]. This is particularly useful when modelling complex phenomena that would require cumbersome heuristic inference if the subscale physics was ignored. In more advanced applications of upscaling concepts, the conservation laws of the coarse-scale medium themselves may be obtained from lower-scale data [29, 30]. Homogenisation can be seen as one particular class of upscaling technique, whereby coarse-scale models approximate the limit of the underlying microscale model when the scale ratio tends to zero [25, 31]. In the classical setting of micromechanics (see for instance [27, 28, 32]), the homogenisation process leads to two interlinked problems: a macroscale mechanical problem with homogeneous constitutive relations, and a microscale problem set over a representative volume element (RVE) of the microstructure, which is often interpreted as a material point of the homogeneous continuum. The solution to the macroscale problem defines a far-field loading for the RVE, usually in the form of boundary conditions. In turns, the solution of the RVE problem permits to find the homogenised coefficients of the coarse-scale constitutive relations, for instance by using micro/macro energy equivalence.

RVE problems were traditionally solved approximately using analytical or semi-analytical approaches [33, 34, 26, 35, 27]. In the last 20 years, computational ho-

mogenisation has emerged as an interesting alternative approach [36, 37, 38, 39, 40], whereby the RVE problem is solved using direct numerical simulation. In linear elasticity, the homogenised constitutive relation can be pre-computed by performing a small set of material tests. The results of these tests are then assembled in the form of a homogenised Hooke tensor that can be readily used at the coarse-scale. In a nonlinear setting, a "naive" implementation of computational homogenisation requires to solve the RVE problem at every (quadrature) point of the macroscopic domain, which, although attractive due to its generality, may render the approach prohibitively expensive. A considerable amount of recent work aims at providing an answer to this dilemma.

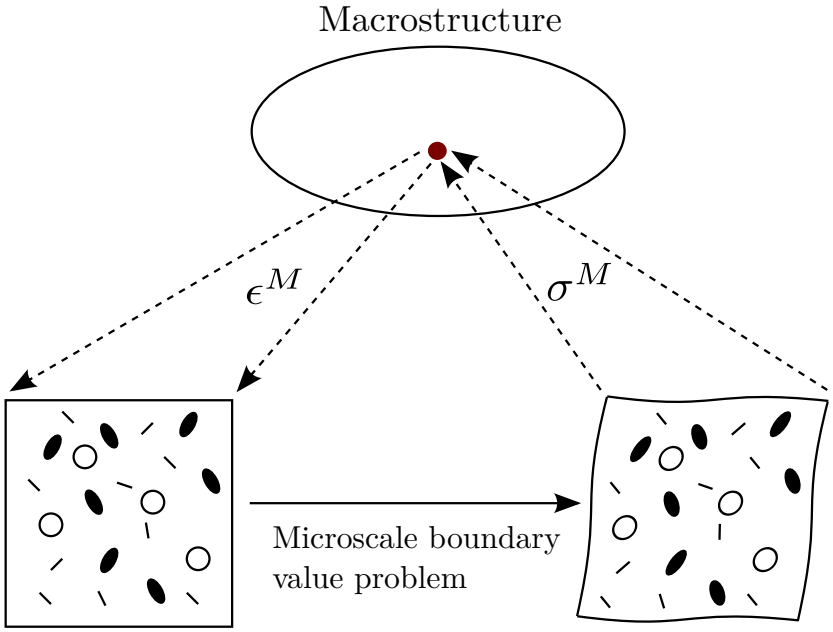


Figure 3.1: Semi-concurrent homogenisation procedure. At each macrostructural quadrature point, an RVE boundary value poblem can be stated with boundary conditions dictated by the macrostrain at this point. Once the boundary value problem solved, the corresponding macrostress is evaluated as a spatial average of the microsstress over the RVE.

On the one hand, the community that relied heavily on semi-analytical approaches to solve RVE problems has developed methods to circumvent the limitations due to the restrictive assumptions upon which these approaches were traditionally based, at the cost of increased computational requirements. The (non-) uniform transformation analysis [41, 42, 43] (see also [44, 45]) and the Voronoi cell approach developed in [46] are remarkable instances of such developments. On the other hand, the community that relied primarily on computational homogenisation methods has tried to reduce the

amount of RVE computations by using meta-modelling, often called meso-modelling in this context. Such developments include the R3M [47, 48] and the method developed in [49], which both rely on a combination of Snapshot-POD expansion for the solution field, and a surface response approach to interpolate the coefficients of this expansion over the space of admissible loading conditions. Our proposed approach is a further step in this direction, which bypasses the need for the surface response step and replaces it by reduced order modelling (ROM).

Projection-based reduced order modelling is an increasingly popular technique for the fast solution of parametrised boundary-value problems. The key idea is to represent the parametric variations of the solution in a low-dimensional subspace. This subspace can be identified using the snapshot-POD [1, 2, 3, 4, 5, 6, 7, 8], which compresses the posterior information contained in an exhaustive sampling of the parameter domain, or the Reduced Basis Method [9, 10, 11, 12, 13], which searches for this attractive subspace in the form of a linear combination of samples chosen quasi-optimally via a Greedy algorithm ("offline stage"). In a second stage, the boundary value problem is projected into this subspace, for instance by a Galerkin method, resulting in a reduced model of number of unknowns equal to the dimension of the attractive space. This reduced model is used to deliver an approximation of the solution to the parametric BVP for any set of parameters, and as such can be seen as an implicit interpolation method over the parameter domain ("online stage"). Early contributions concerning these type of methods have shown an increased accuracy compared to traditional response surface methods, for a given sampling of the parameter domain. Perhaps more importantly, these methods are based on approximation theories, and therefore "naturally" incorporate reliability estimates (e.g. [9, 2, 12, 8]).

In this chapter, we propose to reformulate the nonlinear RVE problem as a parametrised boundary value problem, and subsequently to approximate it using Projection-based ROM. Without loss of generality, we will consider an elastic damageable material represented by a network of damageable beams, with non-homogeneous material properties representing a random distribution of stiff inclusions into a softer matrix. The RVE problem will be parametrised by its far-field loading, represented by homogeneous Dirichlet conditions that belong to a vector space of dimension 6 (3 in two 2D), the time evolution of the coefficients of the associated linear combination being a function of the macroscopic material point. Therefore, our aim is to characterise the solution of the RVE problem for any history of the far-field load, within the restriction of ellipticity (which implicitly define the bounds of the parameter domain).

In a first attempt to approximate this parametrised solution, we will generate random loadings, enforcing some dissipation at each timestep and deploy the Galerkin-POD methodology to derive a reliable ROM. In a more advanced approach, we will avoid relying on randomness, and deploy a Reduced Basis Approach to sample more exhaustively the high-dimensional space. Our strategy is inspired by the work presented in [11] which uses a gradient-based optimisation method in each step of the Greedy algorithm meant to find the parametric case of worst prediction. In our case, this optimisation problem will be solved using an simplifying heuristic, whereby the worst load history will be found as a sequence of worst time steps, which reduces its dimensionality.

We will pay particular attention in the efficiency of the proposed strategy. In particular, projection-based ROM in the nonlinear setting is known to require an additional level of approximation to remain efficient, known as "hyperreduction" or "system approximation" [10, 50, 51, 4, 23, 5, 6]. We will make use of tailored version of the discrete empirical interpolation method (DEIM) [10, 23], which is, to date, the most widely used system approximation methodology. The original DEIM will be slightly modified to allow for the approximation of a vanishing nonlinear term in the balance equations of the discrete RVE problem. We will also propose a way to choose a good ratio between level of approximations in the truncation of attractive subspace versus system approximation.

The chapter is organised as follows. In section 3.2, we define the class of nonlinear homogenisation problems that we want to reduce, and explain how these problems can be parametrised. In section 3.3, we develop specific model order reduction approaches based on the Snapshot-POD and the Reduce Basis methodologies. We highlight the pros and cons of these two distinct approaches in the context of nonlinear homogenisation. In section 3.4, we present our numerical examples and discuss the results. Conclusions are drawn in section 3.5.

3.2 Computational homogenisation setting

We consider a generic representative volume element (RVE) occupying domain Ω (Figure 3.2), corresponding to a microscopically heterogeneous structure. The computational homogenisation approach that is considered in this work is a classical FE^2 scheme [36]: the RVE problem is to be solved numerically, under homogeneous Dirichlet boundary conditions, at every quadrature point of the macroscopic domain, which implicitly

defines the nonlinear constitutive law at the macroscopic level. We will work under the assumption of small perturbations and isothermal mechanical evolution. The material studied in this chapter is damageable elastic, but the methodology is general. In this chapter, the RVE will be modelled by a 2D network of damageable beams (see for instance [52, 53] for more details), whose mechanical properties materialise heterogeneities (random distribution of stiff inclusions in our case). However, for the sake of simplicity, the idea of the approach will first be exposed in the context of continuum mechanics and then discretised, the formulation of the spatially discretised continuum-based or lattice-based model being similar.

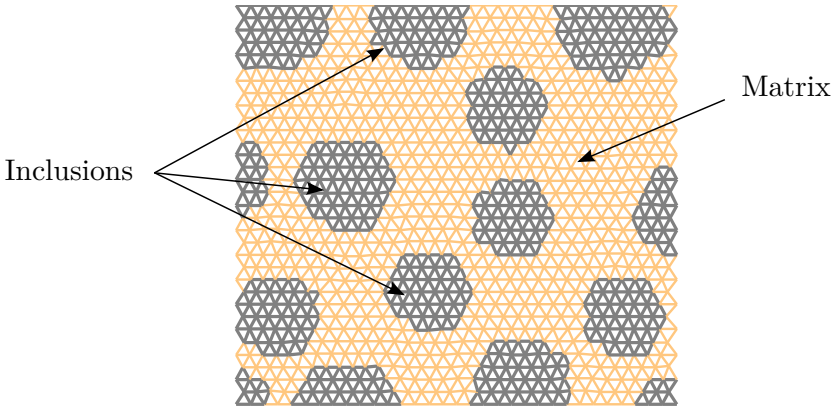


Figure 3.2: Lattice model of the computational representative volume element. Beams have different mechanical properties that depend on their location with respect to the distribution of heterogeneities in the computational domain. An arbitrary distribution of inclusions is chosen as a test case for this chapter.

3.2.1 RVE boundary value problem

At the RVE level, the displacement field is additively split into a fluctuation $\tilde{\mathbf{u}}$ and a smooth (or "macroscopic") part $\bar{\mathbf{u}}$:

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) + \bar{\mathbf{u}}(\mathbf{x}, t) \tag{3.1}$$

where the fluctuation $\tilde{\mathbf{u}}$ vanishes on the boundary $\partial\Omega$ of RVE domain Ω, t denotes time, and the smooth part of the displacement belongs to a 3-dimensional vector space,

$$\bar{\mathbf{u}}(t) = \boldsymbol{\epsilon}^M(t)\,(\mathbf{x} - \bar{\mathbf{x}}) \tag{3.2}$$

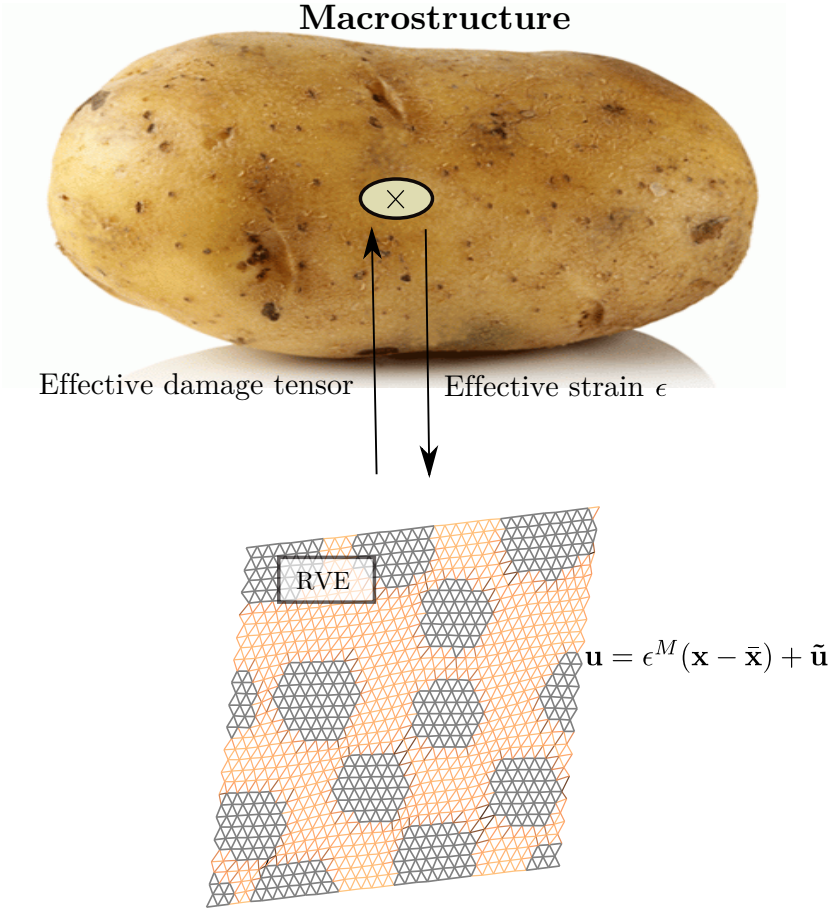


Figure 3.3: Schematic representation of computational homogenisation. The constitutive law of the macro-structure is defined implicitly. The macroscopic strain is applied as boundary condition to the RVE boundary value problem. In turn, the macroscopic stress field is extracted from the solution of the RVE problem using duality principles.

where $\mathbf{x}$ is the position of a material point of the RVE, while $\bar{\mathbf{x}}$ is its centroid and $\boldsymbol{\epsilon}^{\boldsymbol{M}}(t)$ gathers three scalar load coordinates that depend on the position of the corresponding material point of the macroscopic structure:

$$\boldsymbol{\epsilon}^M(t) = \begin{pmatrix} \epsilon^M_{xx}(t) & \epsilon^M_{xy}(t) \\ \epsilon^M_{xy}(t) & \epsilon^M_{yy}(t) \end{pmatrix} \tag{3.3}$$

The mechanical equilibrium of the RVE is expressed by the principal of virtual work:

$$\int_\Omega \boldsymbol{\sigma}^{\mathbf{m}} : \boldsymbol{\epsilon}(\boldsymbol{\delta}\mathbf{u})\, d\Omega = 0\,, \qquad \forall\, \boldsymbol{\delta}\mathbf{u} \text{ s.t. } \boldsymbol{\delta}\mathbf{u}_{|\partial\Omega} = 0 \tag{3.4}$$

where $\boldsymbol{\sigma}^{\mathbf{m}}$ is the microscopic Cauchy stress $\boldsymbol{\epsilon}$ is the strain operator that extracts the symmetric part of the gradient of a displacement vector, while $\boldsymbol{\delta u}$ is a virtual fluctuation field.

The (damageable elastic) constitutive relation of the different micro-constituents of the material is assumed to be known at any time t of the analysis:

$$\boldsymbol{\sigma}^{\mathbf{m}} = \boldsymbol{\sigma}^{\mathbf{m}}((\boldsymbol{\epsilon}(\mathbf{u}(\tau)))_{\tau \leq t}) \tag{3.5}$$

where rate independence, causality and locality are assumed. The history dependence that appears in the previous expression is due to non-reversible damage processes such as plasticity or damage. For the sake of clarity, explicit history-dependance of the variables may be omitted in the remainder of the chapter.

3.2.2 Scale coupling

Following the classical computational homogenisation approach, the relationship between the macroscopic stress $\boldsymbol{\sigma}^M$ and the macroscopic strain at time t and at an arbitrary macroscopic material point can be obtained by using the Hill-Mandel micro-macro energy consistency condition, which reads in the present context

$$\boldsymbol{\sigma}^M \left((\boldsymbol{\epsilon}(\mathbf{u}^M(\tau)))_{\tau<t}\right) : \boldsymbol{\epsilon}^{M^\star} = \frac{1}{|\Omega|} \int_{\partial\Omega} (\boldsymbol{\sigma}^m \cdot \mathbf{n}) \cdot \boldsymbol{u}^\star \, d\Gamma \tag{3.6}$$

for any microscopic displacement $\boldsymbol{u}^\star$ and any macroscopic strain $\boldsymbol{\epsilon}^{M^\star}$ related by the "strain averaging" ansatz $\boldsymbol{u}^\star(\mathbf{x}) = \boldsymbol{\epsilon}^{M^\star}(\mathbf{x} - \bar{\mathbf{x}})$. In the previous expression, $\mathbf{u}^M$ denotes the value of the macroscopic displacement field, and $\boldsymbol{\sigma}^m$ is the microscopic stress field that is the solution of the RVE problem under far field load $\bar{\mathbf{u}}(\tau) = \boldsymbol{\epsilon}(\mathbf{u}^M(\tau))(\mathbf{x} - \bar{\mathbf{x}})$ for any time $\tau < t$.

Equation (3.6) leads directly to the definition of the macroscopic stress as a function of the macroscopic strain history $(\boldsymbol{\epsilon}(\mathbf{u}^M(\tau)))_{\tau<t}$:

$$\boldsymbol{\sigma}^M \left((\boldsymbol{\epsilon}(\mathbf{u}^M(\tau)))_{\tau<t}\right) = \frac{1}{|\Omega|} \int_{\partial\Omega} (\boldsymbol{\sigma}^m \cdot \mathbf{n}) \otimes (\mathbf{x} - \bar{\mathbf{x}}) \, d\Gamma \tag{3.7}$$

which is subsequently used as constitutive equation for the macroscopic problem. Note that in the case of the lattice-based model, a special care is taken for the definition of the normal $\mathbf{n}$.

3.2.3 Space discretisation and Newton solution algorithm

Equilibrium equation (3.4), after substitution of the microscopic constitutive relation, is discretised in space using for instance the finite element method (FEM):

$$\forall\, t,\ \forall\, \boldsymbol{\delta}\mathbf{u} \text{ s.t. } \mathbf{C}\,\boldsymbol{\delta}\mathbf{u} = 0, \qquad \boldsymbol{\delta}\mathbf{u}^T\, \mathbf{f}_{\text{int}}((\mathbf{u}(\tau))_{\tau\leq t}) = 0\ . \tag{3.8}$$

This equation is complemented by the kinematic admissibility condition $\mathbf{u}(t) = \tilde{\mathbf{u}}(t) + \bar{\mathbf{u}}(t)$, where $\mathbf{u}$ denotes the vector of degrees of freedom of the FEM solution of the RVE problem at time t, the vector $\bar{\mathbf{u}}(t)$ of degrees of freedom corresponding to the smooth "macroscopic" continuous field is known, and the vector of degrees of freedom $\tilde{\mathbf{u}}(t)$ corresponding to the continuous fluctuation field satisfies the discrete version of the vanishing boundary condition $\mathbf{C}\,\tilde{\mathbf{u}}(t) = 0$.

We will use a classical implicit time stepping procedure to discretise the RVE problem in time (i.e. integrate the history dependance in the microscopic constitutive relation). This will be further justified in next paragraph. The continuous time interval $\mathcal{T}$ is discretised into n_t subintervals ($[t_n\, t_{n+1}]$). Equilibrium and kinematic relations are enforced at successive discrete times t_n, while the continuous history dependency appearing in the constitutive relation is replaced by its discrete counterpart. The fully discrete version, non-linear version of system of equations (3.8) arising at time t_{n} is solved using a Newton-Raphson algorithm (NR). At each iteration of this algorithm, the following linearisation is computed and solved:

$$\forall\, \boldsymbol{\delta}\mathbf{u} \text{ s.t. } \mathbf{C}\,\boldsymbol{\delta}\mathbf{u} = 0, \qquad \boldsymbol{\delta}\mathbf{u}^T \left(\mathbf{K}^i \boldsymbol{\Delta}\tilde{\mathbf{u}}^{i+1} + \mathbf{r}^i\right) = 0\,, \tag{3.9}$$

where $\mathbf{K}^i = \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{u}}|_{\mathbf{u}^i}$ is the tangent stiffness matrix, $\mathbf{r}^i = \mathbf{f}_{\text{int}}(\tilde{\mathbf{u}}^i + \bar{\mathbf{u}})$ is the residual vector and $\boldsymbol{\Delta}\tilde{\mathbf{u}}^{i+1} = \tilde{\mathbf{u}}^{i+1} - \tilde{\mathbf{u}}^i = \mathbf{u}^{i+1} - \mathbf{u}^i$ (the second equality is only true if the smooth field is used as an initialisation for the NR algorithm, i.e. $\mathbf{u}^0 = \bar{\mathbf{u}}$) is the variation in the fluctuation vector.

3.2.4 Parametrised RVE problem: description of the macroscopic load

In a FE^2 setting, the RVE problem is solved independently for every quadrature point of the macroscopic mesh. In order to apply our ROM technique, we recast the RVE problem as a family of boundary value problems subject to parameter dependency.

The parameters are the three independent components of the far field load tensor $\boldsymbol{\epsilon}^M$ (ϵ^M_{xx}, ϵ^M_{yy} and ϵ^M_{xy}). Physically, they correspond to scalar descriptors of the loading history applied to the macroscopic material point. We emphasise the fact that these parameters are three functions of time, which is not a classical setting for Model Order Reduction. This high (theoretically infinite) dimensionality is a challenge. Some realisations of the loading functions are depicted in figure 3.4.

The next step is to define the parameter domain, or in other words the space in which the three load functions can vary freely. This seems to be a largely problem-dependent issue, and we will focus the discussion on the class of rate-independent, damageable elastic materials. In this case, the first remark is that homogenisation loses its meaning once ellipticity is lost at the macroscopic level. Therefore, bounds are implicitly and collectively defined on the values of the loading functions by enforcing that the macroscopic tangent should remain positive definite. A second remark is that the speed at which the load is applied has no influence on the RVE solution; only the load path matters, which eliminates the need to describe loads that would be applied at different speeds but would essentially result in the same path.

We finally define a time integration scheme for the load history by forcing the macroscopic load to vary by a given amount between two successive time steps. More precisely,

$$\forall\, n \in [\![1, n_t]\!], \quad ||\boldsymbol{\epsilon}^{\mathbf{M}}(t_n) - \boldsymbol{\epsilon}^{\mathbf{M}}(t^\star_{n-1})||_2 = \Delta l \qquad \text{and} \qquad \boldsymbol{\epsilon}^{\mathbf{M}}(t_0) = 0 \tag{3.10}$$

Load parameter Δl should be sufficiently small for the constitutive equations of the RVE to be correctly integrated and for the nonlinear solutions algorithms to converge.

Note that in this time-discrete setting, the number of parameters is 2[1] times the number of pseudo-time steps n_t, which highlights the high-dimensionality of the problem.

3.3 Reduction of the RVE boundary value problem

Our goal is to solve the balance equations of the RVE problem for any history of the macroscopic strain at reduced costs whilst retaining the accuracy of the computed macroscopic stress field. In order to do so, we postulate that for any load applied to the RVE, the fluctuation part of the displacement field can be approximated with an

[1]It is not 3 since we fixed the value of the load between two successive time steps.

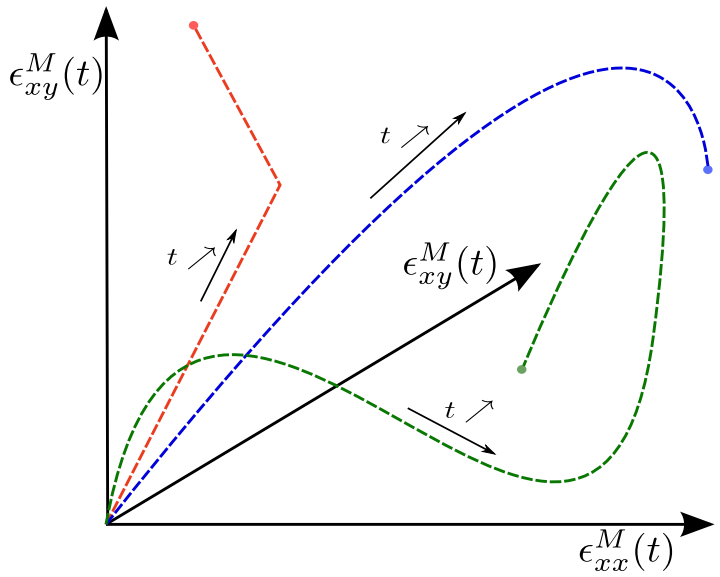


Figure 3.4: Representation of the parameter domain for the nonlinear RVE problem.

acceptable level of accuracy in a vector space of small dimension, called reduced space. This space being identified, we will find an approximation of the displacement field by looking for the amplitude (*i.e.* generalised coordinates) associated to the (few) basis vectors of this space. In this context, three questions arise:

- How can we identify the reduced space?
- How can we find the generalised coordinates in an efficient and stable manner?
- How can we evaluate the reliability of the approach?

The answer to the second question is now relatively well established in the literature. We will make use of a Petrov-Galerkin projection of the discrete set of balance equations (3.8) into the reduced space. More precisely, we will proceed in two stages: a first "ideal" Galerkin projection[2], followed by a second stage of approximation, called "system approximation" [5] of "hyperreduction" [51] to make the solution of the projected system computationally tractable.

The answers to the first and third questions are strongly intertwined, and we describe in the following paragraphs two different manners to approach the problem.

A POD-based approach looks for the best reduced space, in the sense of the minimisation of the projection error on average over the parameter domain. In practice,

[2]Some authors advocate the need for a residual minimising approach to ensure the optimality of this step [5, 13]

this optimisation problem is reduced to a problem of minimum projection error over a representative set of solutions to the parametrised problem, the so-called snapshots [54]. In the case of large parametric dimensions, the sampling of the parameter domain needs to be done in such a way that it overcomes the "curse of dimensionality", for instance by using quasi-random sampling techniques. The reliability of the approach can then be evaluated by resampling (cross-validation, bootstrap, ...) or other statistical tools. This approach suffers from two major drawbacks. Firstly, the optimality of the reduced space is established in an average sense over the parameter domain, which potentially results in inaccurate representation of outliers even for large dimensions of the reduced model. Secondly, the exhaustive sampling of the parameter domain might be prohibitively expensive, and is, in any case, inefficient if performed in a (statistically) uniform manner. The interested reader can find possible ways to tackle this difficult in [55]. Nonetheless, the POD-based methodology remains attractive because the optimisation problem associated with the search of the reduced space can be solved using standard linear algebra tools, namely singular value decomposition.

The Reduced Basis [9] methodology aims at minimising the maximum projection error over the parameter domain. In practice, this is performed in a suboptimal manner using a Greedy algorithm: the reduced order model is constructed iteratively by enriching the reduced space in order to decrease the error at the point of the parameter domain where some measure of projection error is at its largest. When reliable error estimates are available for the projection, the search for the highest level of error over the parameter domain is very efficient, which makes the approach very attractive. The sampling of the parameter domain is performed in a rational manner, which ensures that the construction of the ROM remains affordable. When error estimates are not available, the approach remains attractive in the context of large parametric dimensions. Indeed, the point of the parameter domain that corresponds to the largest level of projection error can be found using gradient-based optimisation, whose numerical complexity may be made independent of the parametric dimension by using the adjoint methodology [11] to compute the sensitivities. In this setting, the "curse" of dimensionality can be overcome whilst retaining reliability of the ROM over the entire parameter domain[3].

In the remainder of this section, we explore these two different possibilities for the reduction of the nonlinear RVE problem. We first propose a Snapshot-POD approach,

[3]This is arguable as the gradient-based optimiser will converge to a local minimum in the parameter domain, see [11] for a more detailed discussion and the proposition of a remedy.

where the sampling is performed randomly, enforcing the random samples to undergo a minimum dissipation at each time step. This setting will allow us to explain in details how the Petrov-Galerkin ROM is constructed from the knowledge of the reduced space. In a second stage, we will develop a Reduced Basis approach for general loading, and propose specific ideas to overcome the "curse of dimensionality".

3.3.1 Galerkin projection of the governing equations in a reduced space

The fluctuating part of the displacement over the RVE[4] is searched for in a reduced space $\mathcal{U}_{\text{MOR}} = \text{span}\left((\boldsymbol{\phi}_i)_{i=1,N}\right)$ of dimension N (se figure 3.5). The displacement is parametrised by the history of the far field load $\left(\boldsymbol{\epsilon}^{\mathbf{M}}(t)\right)_{t\in[0,T]}$, which will subsequently be denoted by $\boldsymbol{\epsilon}^{\mathbf{M}}$ for simplicity. Mathematically, the surrogate for the displacement can be expressed at any time t as

$$\mathbf{u}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) = \bar{\mathbf{u}}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) + \tilde{\mathbf{u}}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) \approx \bar{\mathbf{u}}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) + \sum_{i=1}^{N} \boldsymbol{\phi}_i\, \alpha_i(t;\boldsymbol{\epsilon}^{\mathbf{M}}) = \bar{\mathbf{u}}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) + \boldsymbol{\Phi}\,\boldsymbol{\alpha}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) \tag{3.11}$$

The degrees of freedom of the surrogate are the components of the vector of generalised coordinates $\boldsymbol{\alpha}$. In the previous equation, operator $\boldsymbol{\Phi}$ is the matrix whose columns are the basis vectors of the reduced space $\mathcal{U}_{\text{MOR}}$.

Substituting the trial and test vectors of balanced equation (3.8) by surrogate (3.11) leads to the Galerkin formulation

$$\forall\, t, \forall\, \boldsymbol{\delta\alpha}, \qquad \boldsymbol{\delta\alpha}^T\, \boldsymbol{\Phi}^T\, \mathbf{f}_{\text{int}}\left((\bar{\mathbf{u}}(t;\boldsymbol{\epsilon}^{\mathbf{M}}) + \boldsymbol{\Phi}\,\boldsymbol{\alpha}(t;\boldsymbol{\epsilon}^{\mathbf{M}})\right) = 0\,. \tag{3.12}$$

This reduced nonlinear system of equations can be solved using a Newton-Raphson algorithm. At iteration i of this algorithm, we solve the linear system

$$\boldsymbol{\Phi}^T\left(\widetilde{\mathbf{K}}^i \boldsymbol{\Phi}\,\boldsymbol{\Delta\alpha}^{i+1} + \tilde{\mathbf{r}}^i\right) = 0\,, \tag{3.13}$$

where $\widetilde{\mathbf{K}}^i = \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{u}}_{|\bar{\mathbf{u}}+\boldsymbol{\Phi}(x)\,\boldsymbol{\alpha}^i}$ is the tangent operator, $\tilde{\mathbf{r}}^i = \mathbf{f}_{\text{int}}(\bar{\mathbf{u}} + \boldsymbol{\Phi}\,\boldsymbol{\alpha}^i)$ is the residual vector. It is important to recall that although the number of degrees of freedom of this

[4] We work at a fully discrete level with vectors of degrees of freedom corresponding to continuous fields that belong to FE spaces, but we will refer to such quantities as "fields" or simply "displacements" to avoid unecesseray complication of the explanations.

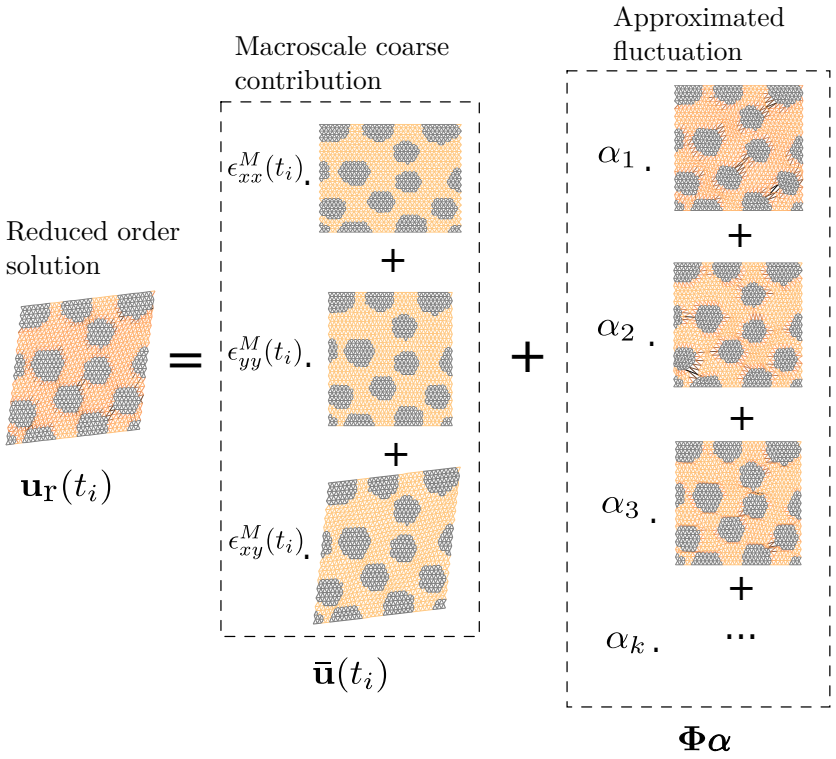


Figure 3.5: Surrogate model for the displacement field in the RVE. The surrogate is the sum of a macroscopic contribution (known *a priori*) and a fluctuation that is represented as a linear combination of basis vectors and obtained through ROM.

system, N, may be small, the cost of assembling the tangent operators and residuals remains expensive. The reduced model cannot be used "online" in this form, which is why an additional "system approximation" is necessary, which will be detailed in section 3.3.2. For now, we will focus on our first proposition to construct a reduced space using the Snapshot-POD methodology.

3.3.2 A first model reduction approach using Snapshot POD on a snapshot randomly generated

Random sampling of the parameter domain

The parameter space is sampled randomly by iteratively generating random load increments $\widetilde{\boldsymbol{\Delta\epsilon^M}}(t_n) = \begin{bmatrix} \widetilde{\epsilon_{xx}}(t_n) & \widetilde{\epsilon_{xy}}(t_n) \\ \widetilde{\epsilon_{xy}}(t_n) & \widetilde{\epsilon_{yy}}(t_n) \end{bmatrix}$ of predefined norm Δl. Initialising the loading

path to be generated by $\boldsymbol{\epsilon}^{\boldsymbol{M}}(t_0) = \mathbf{0}$, the path is iteratively incremented as:

$$\boldsymbol{\epsilon}^{\boldsymbol{M}}(t_{n+1}) = \boldsymbol{\epsilon}^{\boldsymbol{M}}(t_n) + \widetilde{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}(t_n), \text{ with } \|\widetilde{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}(t_n)\| = \Delta l, \tag{3.14}$$

where $\|\widetilde{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}(t_n)\| = \sqrt{\widetilde{\epsilon_{xx}}^2(t_n) + \widetilde{\epsilon_{yy}}^2(t_n) + \widetilde{\epsilon_{xy}}^2(t_n)}$. The random load increments $\widetilde{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}(t_n)$ are forced to create dissipation, by ensuring that at least one of the following inequalities is true at each timestep:

$$\langle\widetilde{\epsilon_{xx}}(t_n)\rangle^+ > \max_{k\in[\![0,n-1]\!]} \widetilde{\epsilon_{xx}}(t_k) \tag{3.15}$$

$$\langle\widetilde{\epsilon_{yy}}(t_n)\rangle^+ > \max_{k\in[\![0,n-1]\!]} \widetilde{\epsilon_{yy}}(t_k) \tag{3.16}$$

$$|\widetilde{\epsilon_{xy}}(t_n)| > \max_{k\in[\![0,n-1]\!]} |\widetilde{\epsilon_{xy}}(t_k)|, \tag{3.17}$$

where $\langle x\rangle^+$ is the positive part of x:

$$\langle x\rangle^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0\ . \end{cases} \tag{3.18}$$

These conditions mean that either the tension in x direction, in y direction or shear has to increase at each timestep. When no dissipation is created, the damage law behave essentially linearly and do not add to the complexity of the snapshot space. An example of a few loading paths generated using this method is displayed in Figure 3.6. It is hoped that the randomness of this procedure will allow to explore the parameter space exhaustively, as long as the number of paths generated is high enough.

Snapshot POD

Once the snapshot is computed, an optimisation problem can be solved to identify the reduced space that minimises a measure of the projection error of the samples. We define the snapshot matrix $\mathbf{S} = \left[\mathbf{s}_1(t_1)\ \mathbf{s}_1(t_2)\dots\mathbf{s}_1(t_{n_t})\ \mathbf{s}_2(t_1)\ \dots\ \mathbf{s}_{n_\mu}(t_{n_t})\right]$, whose columns correspond to the computed samples in various far-field load cases over n_t time steps[5].

[5]Note that in practice, n_t is different between different load cases. Here we try to keep simple notations.

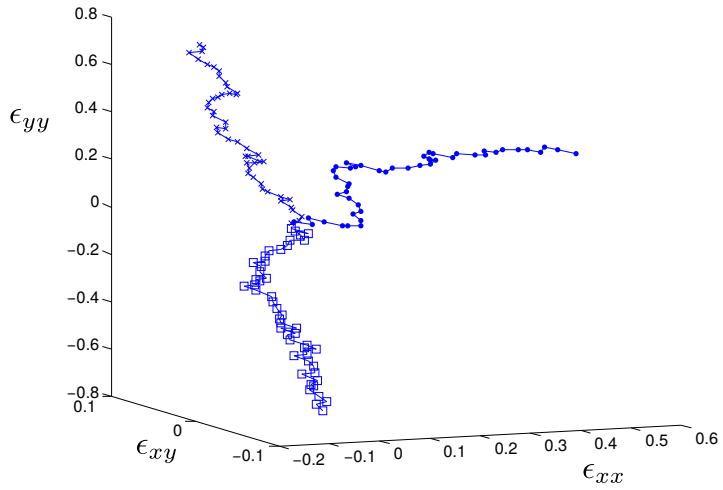


Figure 3.6: Example of loading paths obtained using the random procedure.

The POD minimisation problem reads:

$$\begin{cases} \min\limits_{\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}} J^{\mathrm{s}}_{\langle . \rangle}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) \\ \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle = \delta_{ij} \,, \end{cases} \tag{3.19}$$

where the scalar product $\langle . \rangle$ remains to be defined and $\forall \, \mathbf{x}, \|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. The cost function is defined as:

$$J^{\mathrm{s}}_{\langle . \rangle}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) = \sum_{t_j=t_1}^{t_{n_t}} \sum_{i=1}^{n_\mu} \|\mathbf{s}_i(t_j) - \sum_{k=1}^{N} \langle \boldsymbol{\phi}_\mathbf{k}, \mathbf{s}_i(t_j) \rangle \, \boldsymbol{\phi}_\mathbf{k}\|^2 \,. \tag{3.20}$$

Now, we need to define the scalar product $\langle . \rangle$. The most common choice is the canonical scalar product (i.e. $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \, \mathbf{y}$) which induces the $\mathcal{L}_2$-norm. In our case, the $\mathcal{L}_2$-norm of the displacement field has little interest. Since we are interested in the energy output of the RVE, we choose a scalar product induced by the initial structure stiffness $\mathbf{K_0}$: $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{K_0}} = \mathbf{x}^T \, \mathbf{K_0} \, \mathbf{y}$. This gives a structure specific measure of the displacement quantities. One can then show that solving 3.19 is equivalent to solve the eigenvalue problem:

$$\mathbf{S}\mathbf{S}^T \mathbf{K_0} \boldsymbol{\phi_i} = \lambda_i \boldsymbol{\phi_i}. \tag{3.21}$$

This will provide a set of $\mathbf{K_0}$-orthogonal vectors that best represent the snapshot space in terms of energy. We then have the following error (which represents how well the

POD basis of order N approximates the snapshot $\mathbf{S}$:

$$\nu_{\text{POD}}(\boldsymbol{\phi_1}, ..., \boldsymbol{\phi_N}) = \sqrt{\frac{\sum_{k=N+1}^{n_u} \lambda_k}{\sum_{k=1}^{n_u} \lambda_k}}. \tag{3.22}$$

See Appendix A for more details.

Estimation of the maximal POD error by cross-validation (CV)

One important point in projection-based reduced order modelling is the estimation of the error made by the reduced order model depending on the dimension of the basis $\boldsymbol{\Phi}$. Given a predefined accuracy, this information allows the user to choose the optimal size of the reduced space, i.e. the smallest size that achieve that accuracy. The error 3.22 given by the POD is mean square error that represent the average projection error between the snapshot and its projection onto the reduced space. This estimate is not satisfactory for two reasons:

- This estimate is biased as it is based as the reduced space is based on the snapshot that also serves as a validation set. It underestimates the actual error.

- It is an average, which means that for some specific choice of the loading path, the error may be much higher. There is no guarantee on how large the maximum error may be (i.e. the error linked to the load path leading to the solution the least well approximated).

In our context, we are interested in the maximal error that the reduced model may lead to, which means we want to exhibit an error bound for which it is guaranteed that no loading path will lead to an error that is larger than this bound.

Cross-validation (CV) is a statistical way to determine the predictability of a model according to a set of data. The data set is partitioned into a training set and a validation set, which need to be mutually exclusive for the estimator to be unbiased. The model is built based on the training set, and an error is evaluated by comparing its predictions on the validation set. The leave-one-out CV (LOOCV) considers successively, and in turn, each single data value to be the validation set while the rest of the data forms the training set. The combined errors are then averaged to evaluate the level of predictability of the model based on the data given.

In our context, assuming we have a snapshot at hand that corresponds to the parameter set denoted $\mathcal{P}_{\boldsymbol{\epsilon}}$, the LOOCV estimate of the maximum error reads:

$$\nu^{p}_{\text{LOOCV}} = \max_{\boldsymbol{\epsilon}^{M}_{j} \in \mathcal{P}_{\boldsymbol{\epsilon}}} \left(\sum_{t_i = t_0}^{t_{n_t}} \frac{\left\| \mathbf{u}_{\text{ex}}\left(t_i; \boldsymbol{\epsilon}^{M}_{j}\right) - \boldsymbol{\Phi}^{\boldsymbol{p}}_{\boldsymbol{\mathcal{P}_{\epsilon}} \setminus \boldsymbol{\epsilon}^{M}_{j}} \left\langle \boldsymbol{\Phi}^{\boldsymbol{p}}_{\boldsymbol{\mathcal{P}_{\epsilon}} \setminus \boldsymbol{\epsilon}^{M}_{j}}, \mathbf{u}_{\text{ex}}\left(t_i; \boldsymbol{\epsilon}^{M}_{j}\right) \right\rangle_{\mathbf{K_0}} \right\|^{2}_{\mathbf{K_0}}}{\left\| \mathbf{u}_{\text{ex}}\left(t_i; \boldsymbol{\epsilon}^{M}_{j}\right) \right\|^{2}_{\mathbf{K_0}}} \right), \tag{3.23}$$

where $\boldsymbol{\Phi}^{\boldsymbol{p}}_{\boldsymbol{\mathcal{P}_{\epsilon}} \setminus \boldsymbol{\epsilon}^{M}_{j}}$ is the reduced space obtained when using the training set $\mathcal{P}_{\boldsymbol{\epsilon}} \setminus \boldsymbol{\epsilon}^{M}_{j}$, which is the global set $\mathcal{P}_{\boldsymbol{\epsilon}}$ but excluding the j^{th} loading path $\boldsymbol{\epsilon}^{M}_{j}$.

System Approximation

Constraining the displacement in a low-dimensional space does not provide a significant computational gain, even if the systems to be solved are of smaller dimension. This is because the material of study is nonlinear and history-dependent, and its stiffness varies not only in different areas of the material but also with time. This requires to evaluate the stiffness everywhere in the material and this at each time step of the simulation. This means that the numerical complexity remains despite the simplification on the displacement. Hence, to decrease the numerical complexity, the domain itself need to be approximated. Several authors have looked into that. Notable contributions include the Hyperreduction method [51], the missing point estimation [50], or system approximation [5]. Those methods share the idea that the material properties will be evaluated only at a small set of points or elements within the material domain. They differ in the way of selecting those points and in the treatment of that reduced information. In this chapter, we will use the "gappy" method, very much like in [5, 56].

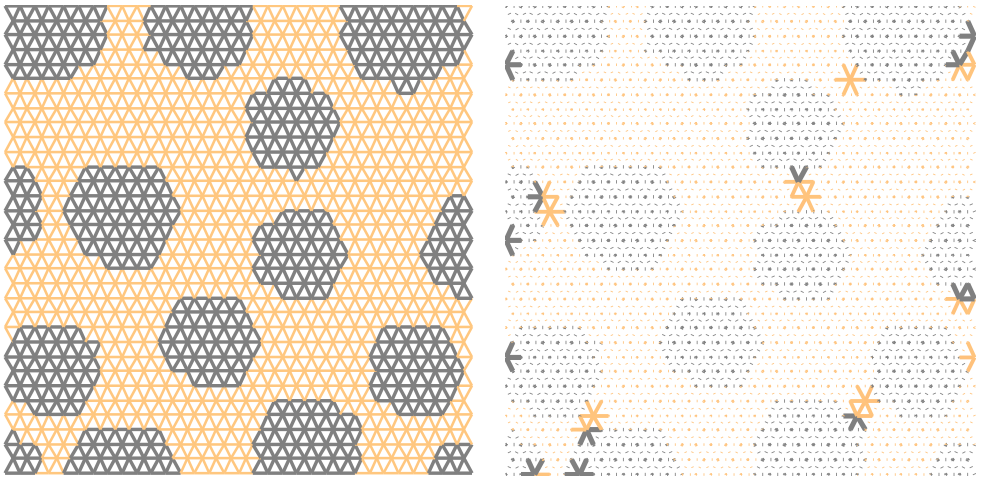
Gappy Method

The internal forces generated by the reduced displacement $\mathbf{f}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha})$ will be evaluated only in a small subset of the degrees of freedom $\mathcal{I}$ of the domain Ω. A procedure to select $\mathcal{I}$ will be described later on. All the elements in contact with those degrees of freedom have to be considered. We refer to those as the controlled elements. The internal forces will then be reconstructed by writing the internal forces as a linear com-

bination of a few basis vectors themselves (just like it was made for the displacement).

$$\mathbf{f}_{\text{int}}(\mathbf{\Phi}\,\boldsymbol{\alpha}) \approx \sum_{1}^{n_{\text{gap}}} \boldsymbol{\psi}_{\mathbf{i}}\beta_i = \mathbf{\Psi}\boldsymbol{\beta}, \tag{3.24}$$

where $\left[\boldsymbol{\psi}_{\mathbf{1}}, \cdots, \boldsymbol{\psi}_{\mathbf{n_{\text{gap}}}}\right] = \mathbf{\Psi}$ is the forces basis of size n_{gap} and $\boldsymbol{\beta}$ the associated scalar coefficients.



(a) Original structure (b) Example of a surrogate structure

Figure 3.7: Example of a surrogate structure. The stiffness of the structure is evaluated on controlled elements only, while the other ones are just like ghosts

The coefficients $\boldsymbol{\beta}$ of the expansion are found so that to minimise the norm of the difference between the linear expansion and the nonlinear term over the subset $\mathcal{I}$:

$$\underset{\boldsymbol{\beta}^\star}{\text{argmin}} \|\mathbf{f}_{\text{int}}(\mathbf{\Phi}\,\boldsymbol{\alpha}) - \mathbf{\Psi}\boldsymbol{\beta}^\star\|_{\mathbf{P}}, \tag{3.25}$$

with $\mathbf{P}$ being a matrix so that $\mathbf{P}ij = \begin{cases} 1 & \text{if} \quad i \in \mathcal{I} \quad \text{and} \quad i = j \\ 0 & \text{otherwise} \end{cases}$ and $\|\mathbf{x}\|_{\mathbf{P}} = \|\mathbf{P}^T\,\mathbf{x}\,\mathbf{P}\|_2$. $\mathbf{P}$ can be written $\mathbf{E}\,\mathbf{E}^T$ with $\mathbf{E}$ being an extractor matrix so that $\mathbf{E}^T\mathbf{x}$ is the restriction of $\mathbf{x}$ to the set $\mathcal{I}$. If the number of points in $\mathcal{I}$ is identical to the number of basis vectors $(\boldsymbol{\psi_i})_{i=1,n_{\text{gap}}}$, $\boldsymbol{\beta}^\star$ can be found by solving the equation:

$$\mathbf{E}^T\,\mathbf{\Psi}\boldsymbol{\beta} = \mathbf{E}^T\,\mathbf{f}_{\text{int}}(\mathbf{\Phi}\,\boldsymbol{\alpha}), \tag{3.26}$$

which implies:

$$\boldsymbol{\beta} = (\mathbf{E}^T\boldsymbol{\Psi})^{-1}\mathbf{E}^T\mathbf{f}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha}). \tag{3.27}$$

Note that this works only assuming that $\mathbf{E}^T\boldsymbol{\Psi}$ is invertible, which is almost always the case if the subset $\mathcal{I}$ is well-chosen. At a Newton iteration of our POD-Galerkin framework, this reduces equation (3.13) to (where we dropped Newton iteration indices for the sake of clarity):

$$\boldsymbol{\Phi}^T\boldsymbol{\Psi}(\mathbf{E}^T\boldsymbol{\Psi})^{-1}\mathbf{E}^T\,\mathbf{K}\boldsymbol{\Phi}\boldsymbol{\Delta\alpha} = -\boldsymbol{\Phi}^T\boldsymbol{\Psi}(\mathbf{E}^T\boldsymbol{\Psi})^{-1}\mathbf{E}^T\,\mathbf{r}. \tag{3.28}$$

This can be rewritten in the form:

$$\mathbf{G}\mathbf{E}^T\,\mathbf{K}\boldsymbol{\Phi}\boldsymbol{\Delta\alpha} = -\mathbf{G}\,\mathbf{E}^T\,\mathbf{r}, \tag{3.29}$$

where we define the gappy operator $\mathbf{G} = \boldsymbol{\Phi}^T\boldsymbol{\Psi}(\mathbf{E}\boldsymbol{\Psi})^{-1}$.

Remark: *Note that once the "offline" stage operations are done, the bases* $\boldsymbol{\Phi}$ *and* $\boldsymbol{\Psi}$ *are calculated and the set of control points* $\mathcal{I}$ *is selected and the gappy operator is evaluated. In the "online" stage, all that remains to do is build a system of dimension equal to the size of the displacement basis and solve it which is computationally much cheaper. In particular, the evaluation of* $\mathbf{K}$ *will be substituted by the evaluation of* $\mathbf{E}^T\,\mathbf{K}$*, which allows great time savings.*

Selection of the controlled elements

The selection of the control elements will be done using the discrete empirical interpolation method (DEIM) [56]. This method finds a set of degrees of freedom $\mathcal{I}$ in a greedy manner from the internal forces basis $\boldsymbol{\Psi}$. We quickly describe here the method.

At iteration j of the greedy algorithm, $j-1$ points have been already selected. We define the extractor $\mathbf{E^j}$ that extracts those j selected degrees of freedom (i.e. for any vector $\mathbf{v}$, $\mathbf{E^j v}$ is a smaller vector containing only the j entries of $\mathbf{v}$ corresponding to the selected degrees of fredom). The residual $\mathbf{r}_{\text{gap}} = \left|\boldsymbol{\psi}_{[\mathbf{1,j}]}\boldsymbol{\beta^j} - \boldsymbol{\psi}_{\mathbf{j+1}}\right|$ is evaluated, where $\boldsymbol{\psi}_{[\mathbf{1,j}]}$ is the matrix containing the first j vectors of the basis $\boldsymbol{\Psi}$ and $\boldsymbol{\psi}_{\mathbf{j+1}}$ is the $j+1^{th}$

vectors in that basis. $\boldsymbol{\beta}$ is the solution of the minimisation problem

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}^\star}{\operatorname{argmin}} \left\| \mathbf{E}^{\mathbf{j}} \boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]} \boldsymbol{\beta}^\star - \mathbf{E}^{\mathbf{j}} \boldsymbol{\psi}_{\mathbf{j+1}} \right\|_2 . \tag{3.30}$$

The solution is easily found: $\boldsymbol{\beta} = \left(\mathbf{E}^{\mathbf{j}} \boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]} \right)^{-1} \mathbf{E}^{\mathbf{j}} \boldsymbol{\psi}_{\mathbf{j+1}}$. The greedy procedure then selects the index of the highest entry in $\mathbf{r}_{\text{gap}}$ as the $j+1^{th}$ control degree of freedom. This procedure essentially selects the set of degrees of freedom that maximises the conditioning of the system (3.26). At the end of the greedy algorithm, the number of control degrees of freedom chosen equals the number of basis vectors $\left(\boldsymbol{\psi}_{\mathbf{i}}\right)_{n_{\text{gap}}}$ which makes system (3.26) well defined.

3.3.3 Model reduction using a POD-greedy algorithm

As said in the previous section, it may not be satisfactory to use an arbitrary sampling method, since some important information could be unwittingly dropped out. The accuracy of the reduced model greatly depends on the snapshot space and how well it samples the parameter space. Here, the parameter space contains any load path (based on the macro-strain $\boldsymbol{\epsilon}^{\boldsymbol{M}}(t)$) over a certain period of time until fracture is reached. After time discretisation, the parameter space is of dimension $3 \times n_{\text{t}}$, since in 2 dimensions the load can be uniaxial in the x or y direction or in shear. n_{t} stands for the number of time steps required to reach fracture. To ensure an exhaustive sampling, we will follow a greedy iterative procedure. In [11], a method for sampling a high-dimensional parameter space based on model-constrained optimisation was proposed. In our context, this approach is intractable. Indeed, evaluating a numerical derivative with respect to the all the parameters at once involves $3 \times n_t$ full order evaluation that each has a cost of the order of n_t, which means the cost of this procedure grows quadratically with the number of timesteps ($\mathcal{O}\left(n_t^2\right)$). We will present in the following a greedy method that will have a cost growing only linearly with the number of timesteps (($\mathcal{O}\left(n_t\right)$)).

In our case, we take advantage of the fact that our parameter space is naturally hierarchical (it is ordered according to the time discretisation) to use a less computationally-intensive method.The idea is to enrich the snapshot space with the solution $\mathbf{u}_{\max}$ obtained from a certain discrete loading path $\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_0, t_1, \ldots)$ that leads to the worst approximation by the current POD basis. This "path of worst approximation" is built up in a greedy manner, incrementally at each timestep. This is the loading path that will intuitively provide the most information to the POD basis.

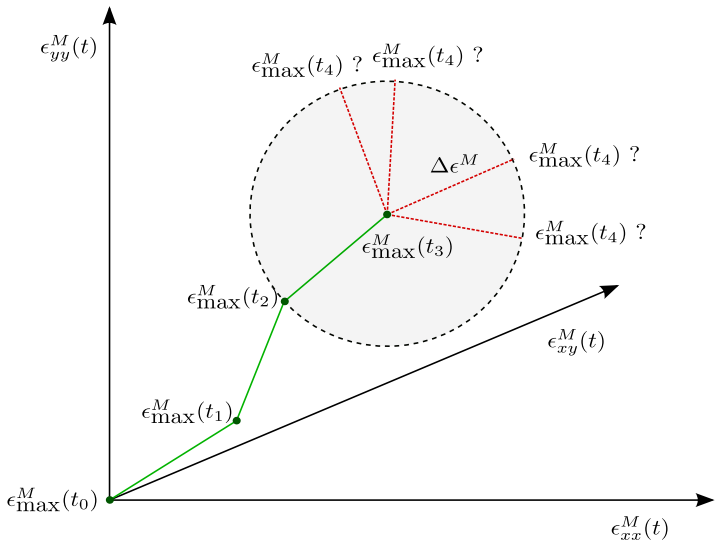


Figure 3.8: Illustration of the greedy procedure to find the path of worst prediction. At step 4 of the procedure, we are dealing with the problem of finding the load increment $\boldsymbol{\delta\epsilon^M}$ that leads to the worst approximation at time t_4. This load increment is constrained to have a predefined norm. This optimisation problem is solved numerically using a standard gradient method. Once the optimal load increment is found, the load path is updated up to step 4, and the procedure iterates in the same way for all the subsequent timesteps until the structure fails.

Construction of the path of "worst prediction"

The greedy procedure is as follows: assuming $\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_0, t_1, \ldots, t_i)$ is the "path of worst approximation" up to timestep t_i, the procedure looks for the load increment $\Delta\epsilon^i_{\max}$ that maximises the error between the full order model and the reduced model at time t_{i+1} having followed the loading path $\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_0, t_1, \ldots, t_i)$ up to time t_i. The loading path is then extended to time t_{i+1} as:

$$\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_{i+1}) = \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_i) + \Delta\epsilon^i_{\max}. \tag{3.31}$$

The procedure iterates until reaching fracture. The maximisation process to find the load increment $\Delta\epsilon^i_{\max}$, based on gradient descent, is described in the following paragraphs.

A sequence of maximisation problems

We define a sequence of error functions parametrised by the current timestep t_i and

the value of the load up to that time $\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau<t_i}}$:

$$f_{\text{err}}\left(t_i, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}, \boldsymbol{\Phi^i}\right) = \|\mathbf{u}_{\text{ex}}\left(t_i; \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}\right) - \mathbf{u}_{\text{r}}\left(t_i; \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}, \boldsymbol{\Phi^i}\right)\|_{\mathbf{K_0}}, \tag{3.32}$$

where $\mathbf{u}_{\text{ex}}$ stands for the exact solution, parametrised by a loading path $\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}$ up to time t_i and $\mathbf{u}_{\text{r}}$ stands for the reduced order solution using the kinematic basis $\boldsymbol{\Phi^i}$ and the same loading path. No system approximation is considered. Indeed, in this case, we are interested in sampling exhaustively the parameter space to obtain the best kinematic basis $\boldsymbol{\Phi}$, without having the system approximation to pollute the data. The function f_{err} simply evaluates the error between the exact solution and the reduced order solution at timestep t_i. Now, at step $i+1$ of the greedy procedure, we are looking for the load increment that maximises the reduced order model error. To this purpose, we solve the following maximisation problem:

$$\begin{cases} \max\limits_{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}}} f_{\text{err}}\left(t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{t_i}} + \boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}}\right) \\ \left\|\boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}}\right\| = \delta_{\text{step}}, \end{cases} \tag{3.33}$$

where δ_{step} is a predefined load increment value that we keep constant during the whole greedy procedure. $\boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}} = \begin{bmatrix} \Delta\epsilon_{xx} & \Delta\epsilon_{xy} \\ \Delta\epsilon_{xy} & \Delta\epsilon_{yy} \end{bmatrix} = \begin{pmatrix} \Delta\epsilon_{xx} \\ \Delta\epsilon_{yy} \\ \Delta\epsilon_{xy} \end{pmatrix}$ is the load increment from timestep t_i through to t_{i+1}.

Problem (3.33) can be rewritten to become unconstrained:

$$\max_{\theta^\star,\varphi^\star} f_{\text{err}}\left(t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{t_i}} + \boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}(\theta^\star, \varphi^\star)\right) = \widehat{f_{\text{err}}}\left(\theta^\star, \varphi^\star; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}\right), \tag{3.34}$$

where $\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}(\theta^\star, \varphi^\star) = \begin{pmatrix} \Delta\epsilon_{xx}(\theta^\star, \varphi^\star) \\ \Delta\epsilon_{yy}(\theta^\star, \varphi^\star) \\ \Delta\epsilon_{xy}(\theta^\star, \varphi^\star) \end{pmatrix} = \begin{pmatrix} \delta_{\text{step}} \cos\theta^\star \cos\varphi^\star \\ \delta_{\text{step}} \sin\theta^\star \cos\varphi^\star \\ \delta_{\text{step}} \sin\varphi^\star \end{pmatrix}$ which guarantees $\left\|\widehat{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}\right\| = \delta_{\text{step}}$. Here, we used standard spherical coordinates with θ being the polar angle and φ the azimuthal angle, as shown in Figure 3.9. δ_{step} represents the radial distance which is fixed in our case.

Gradient descent algorithm

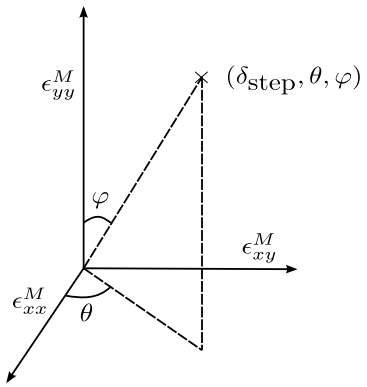


Figure 3.9: Standard spherical coordinates to identify values of macro-strain.

To solve problem (3.34), we use a gradient descent method (also known as steepest descent). This is a first order optimisation method. From an initial guess (θ_0, φ_0), a sequence of iterates is found using the following update:

$$\begin{pmatrix} \theta^{k+1} \\ \varphi^{k+1} \end{pmatrix} = \begin{pmatrix} \theta^{k} \\ \varphi^{k} \end{pmatrix} + \gamma \frac{\partial \widehat{f_{\text{err}}}}{\partial\,(\theta, \varphi)}(\theta^k, \varphi^k; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}}), \tag{3.35}$$

where γ is a scalar whose value is found through a linesearch using a bisection algorithm. The derivative $\frac{\partial \widehat{f_{\text{err}}}}{\partial(\theta,\varphi)}(\theta^k, \varphi^k; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}})$ is evaluated numerically by taking small variations (of size ν) around the value $\boldsymbol{\Delta \epsilon^k}$:

$$\frac{\partial \widehat{f_{\text{err}}}}{\partial\,(\theta, \varphi)}(\theta^k, \varphi^k; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}}) \approx \frac{1}{\nu} \begin{bmatrix} \widehat{f_{\text{err}}}\left(\begin{pmatrix} \theta^k + \nu \\ \varphi^k \end{pmatrix}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}} \right) - \widehat{f_{\text{err}}}\left(\theta^k, \varphi^k; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}} \right) \\ \widehat{f_{\text{err}}}\left(\begin{pmatrix} \theta^k \\ \varphi^k + \nu \end{pmatrix}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}} \right) - \widehat{f_{\text{err}}}\left(\theta^k, \varphi^k; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau \leq t_i}} \right) \end{bmatrix}. \tag{3.36}$$

In theory, the gradient descent algorithm would run until reaching some convergence criterion. In practice, only a few iterations are performed to reach a decent optimum, as each step involves solving the full order model several times and is hence costly. The results are displayed in Figure 3.15.

Remark: *Trying to find the path of worst approximation using a standard optimisation method similar to what was done in [11] quickly becomes intractable when considering the number of timesteps n_t in our context. Indeed, computing the gradient of cost*

function with respect to the parameter involves $3 \times n_t$ evaluations of the full order model that itself is at a cost of n_t timesteps. This makes a gradient evaluation of the order of $3 \times n_t^2$: the optimisation algorithm grows quadratically with the number of timesteps. However, in the context of our greedy optimisation procedure, computing the gradient only requires 3 evaluations of one timestep of the full order model, and that for each timestep, which means the computational cost grows only linearly with n_t, making it a feasible algorithm.

POD-greedy and stopping criterion

The POD-greedy algorithm consists in iteratively enriching the reduced basis with the solution generated by the path of worst prediction. More precisely, at stage k of the POD-greedy algorithm, the current reduced basis is denoted $\mathbf{\Phi^k}$. The solution generated by the path of worst prediction is denoted $\mathbf{u}^{\mathbf{k}}_{\max}(t)$. We define the projection error $\mathbf{p}_{\text{err}}$ as the following:

$$\mathbf{p}^{\mathbf{k}}_{\text{err}} = \mathbf{u}^{\mathbf{k}}_{\max} - \mathbf{\Phi^k}\,\mathbf{\Phi^k}^T\mathbf{u}^{\mathbf{k}}_{\max}. \tag{3.37}$$

$\mathbf{p}_{\text{err}}$ has the interesting property to be orthogonal to $\mathbf{\Phi^k}$. It can be interpreted as the part of $\mathbf{u}^{\mathbf{k}}_{\max}$ that the basis $\mathbf{\Phi^k}$ is unable to approximate. Therefore, it is meaningful to enrich the current basis using the error $\mathbf{p}^{\mathbf{k}}_{\text{err}}$ rather than $\mathbf{u}^{\mathbf{k}}_{\max}$ itself, since it is in the span of $\mathbf{p}^{\mathbf{k}}_{\text{err}}$ that the "new" information lies. $\mathbf{\Phi^k}$ is enriched by applying a POD transform to the projection error $\mathbf{p}^{\mathbf{k}}_{\text{err}}$ and adding the eigenvector $\boldsymbol{\phi}^{\boldsymbol{k}}_{\text{enrich}}$ corresponding to the largest eigenvalue . The new basis is then $\mathbf{\Phi^{k+1}} = [\mathbf{\Phi^k}, \boldsymbol{\phi}^{\boldsymbol{k}}_{\text{enrich}}]$. The procedure goes on enrichment after enrichment until the prediction error generated by the last enrichment, averaged over the number of timesteps necessary to reach fracture, goes below a certain tolerance, chosen by the user:

$$\frac{1}{n_t}\sqrt{\sum_{t_i=t_0}^{t_{n_t}} \left\| \mathbf{u}_{\text{ex}}\left(t_i, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(\mathbf{\Phi^k})\right) - \mathbf{u}_{\text{r}}\left(t_i; \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(\mathbf{\Phi^k}), \mathbf{\Phi^k}\right) \right\|^2_{\mathbf{K_0}}} \leq \nu_{\max} \tag{3.38}$$

This error criterion is intuitive as it insures that the solution the "least well" predicted by the reduced basis still is within controlled distance to the full order model.

In our specific setting, constructing the path of worst approximation is a very expensive process (compared to the cheap evaluation of the error bound in the standard reduced-basis method), and it is of interest to minimise the amount of enrichment steps

of the greedy procedure by storing more than the first eigenvector from the snapshot POD on $\mathbf{p}^{\mathbf{k}}_{\text{err}}$ to make the overall procedure faster by enlarging the reduced basis faster. This leads to a suboptimal basis size in practice (which means similar accuracies may be reached for a given tolerance by using fewer basis vectors), but allows for a shorter and more feasible offline computational time. To decide the number N^k of eigenvectors to add to the current reduced basis, the eigenvectors are iteratively added one by one until the error $\nu^k_{\text{err}}(N)$ between the exact and reduced solution generated using the path of worst prediction reaches the same tolerance ν_{enrich} that is required by the user for the POD-greedy algorithm:

$$\begin{cases} \text{Find } N^k = \underset{N^\star}{\text{argmin}} \quad \nu^k_{\text{err}}(N^\star) \quad \text{such that: } \nu^k_{\text{err}} \leq \nu_{\max} \\ \text{with } \nu^k_{\text{err}}(N) = \frac{1}{n_t}\sqrt{\sum_{t_i=t_0}^{t_{n_t}} \| \mathbf{u}_{\text{ex}}\left(t_i, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(\boldsymbol{\Phi}^{\boldsymbol{k}})\right) - \mathbf{u}_{\text{r}}\left(t_i; \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(\boldsymbol{\Phi}^{\boldsymbol{k}}), [\boldsymbol{\Phi}^{\boldsymbol{k}}, \boldsymbol{\Phi}^{\mathbf{k},[1,N]}_{\text{enrich}}]\right) \|^2_{\mathbf{K_0}}}, \end{cases} \tag{3.39}$$

where $\boldsymbol{\Phi}^{\mathbf{k},[1,N]}_{\text{enrich}}$ is the basis obtained by storing the first N eigenvectors of a POD transform applied on $\mathbf{p}^{\mathbf{k}}_{\text{err}}$. $\mathbf{u}_{\text{r}}\left(t_i; \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(\boldsymbol{\Phi}^{\boldsymbol{k}}), [\boldsymbol{\Phi}^{\boldsymbol{k}}, \boldsymbol{\Phi}^{\mathbf{k},[1,N]}_{\text{enrich}}]\right)$ is the reduced solution obtained when using a kinetic basis enriched with the first N vectors of the POD transform of the previous projection error and it needs to be recomputed for each value of N. It may still be expensive as we do not apply any system approximation at this point. However, it is faster than going through the procedure of the maximisation problems (3.33). We later refer to this particular POD-greedy strategy as the adaptive strategy, the former strategy being referred to as standard.

3.4 Example and Numerical Results

We consider the sample RVE as presented in section 3.2, and we proceed to apply the reduced-order modelling methods described in the previous sections. Here we are interested in comparing the performance of the ROM when applying POD on a set generated by an arbitrary sampling of the parameter space (presented in section 3.3.2) versus when applying the Reduced basis approach assorted with the POD-greedy procedure (described in section 3.3.3). We present the results in two chunks:

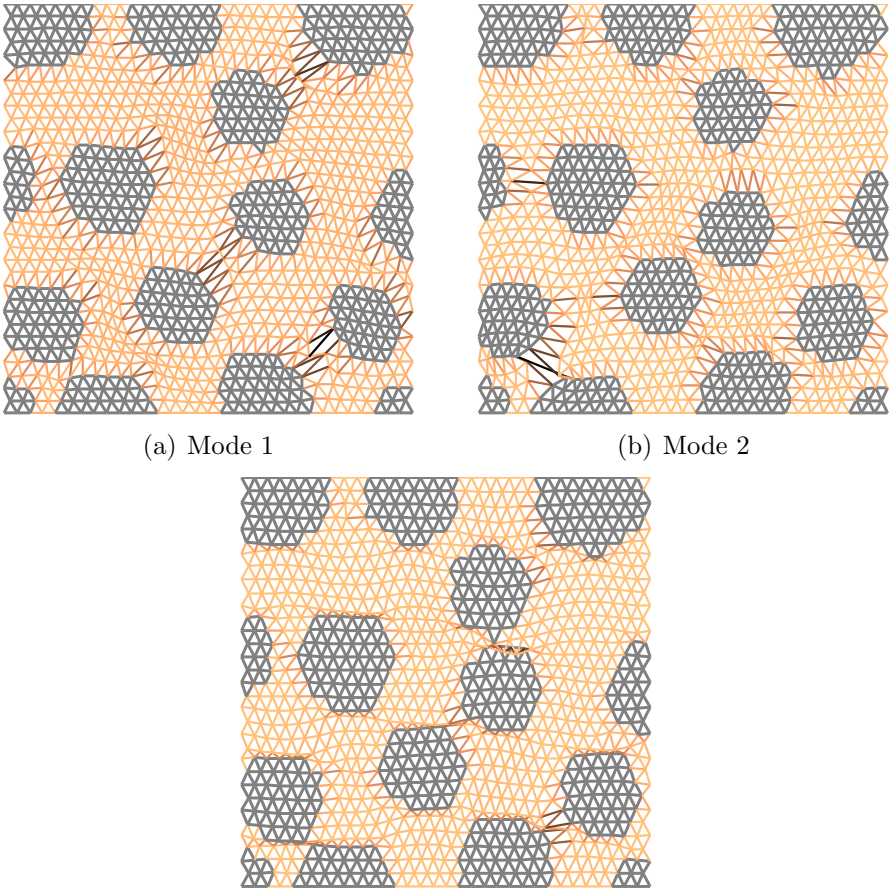
- The first section shows the basis obtained when using the POD procedure from section 3.3.2, together with a mechanical interpretation of the location of the controlled elements generated using the DEIM method inside the RVE. It also

presents the error and the speedup obtained with the POD method while varying the dimension of the displacement and static bases.

- The second section is devoted to the results obtained with the reduced basis method. It shows the far-field load paths generated by the Reduced basis method presented in section 3.3.3 as well as their associated error when using the two kinds of POD-greedy procedure presented in the paper (standard and adaptive). A comparison of the performance of POD against Reduced basis method concludes the results display.

3.4.1 Snapshot-POD reduced space and numerical results

Displacement basis. We proceed to apply the snapshot-POD procedure with random snapshot selection described in section 3.3.2. 36 load paths are randomly generated. The first few vectors of the POD expansion are displayed in Figure 3.10.



(a) Mode 1 (b) Mode 2

(c) Mode 3

Figure 3.10: Vectors associated to the 3 largest eigenvalues obtained using the snapshot-POD procedure with random snapshot selection. The damage localises between pairs of inclusions.

System approximation. We follow the procedure described in 3.3.2. The basis $\mathbf{\Psi}$ is extracted from the snapshot space generated by the same loading paths used for the displacement basis $\mathbf{\Phi}$. The set of controlled elements is selected using the DEIM [56]. The amount of vectors in the basis $\mathbf{\Psi}$ is chosen so that the error generated by the system approximation is of the same order than the global error of the reduced order model. The error ν_{tot} between the exact solution and the reduced model solution with system approximation can be decomposed in the following way (with $\mathbf{u}_{\text{ex}}(t)$ the exact solution, $\mathbf{u}_{\text{r}}(t;\mathbf{\Phi})$, the reduced order solution without the system approximation using the displacement basis $\mathbf{\Phi}$, and $\mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi})$ the complete reduced order model with

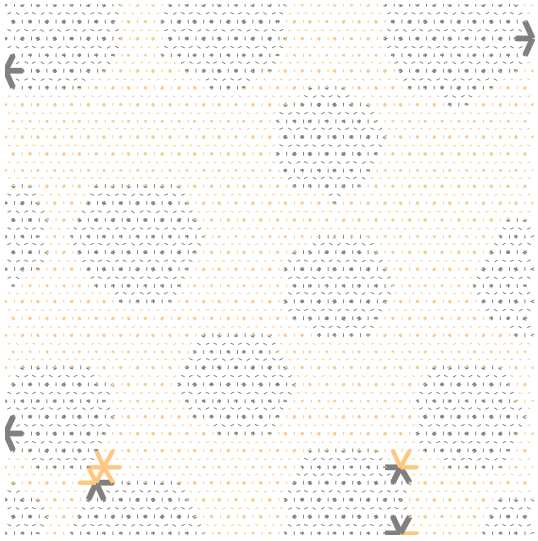
system approximation using the displacement basis $\mathbf{\Phi}$ and the static basis $\mathbf{\Psi}$):

$$\begin{aligned}\nu_{\text{tot}}(t)^2 &= \|\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi})\|^2_{\mathbf{K_0}} && (3.40)\\ &= \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r}}(t;\mathbf{\Phi})) + (\mathbf{u}_{\text{r}}(t;\mathbf{\Phi}) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi}))\|^2_{\mathbf{K_0}} && (3.41)\\ &\leq \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r}}(t;\mathbf{\Phi}))\|^2_{\mathbf{K_0}} + \|(\mathbf{u}_{\text{r}}(t;\mathbf{\Phi}) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi}))\|^2_{\mathbf{K_0}}\,. && (3.42)\end{aligned}$$

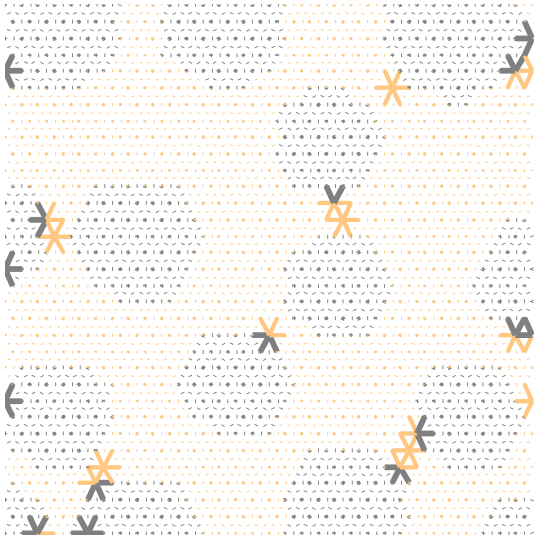
Taking this in consideration, the basis $\mathbf{\Psi}$ is chosen to be the smallest (i.e. the one with the least amount of vectors) that verifies the inequality:

$$\|(\mathbf{u}_{\text{r}}(t;\mathbf{\Phi}) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi}))\|_{\mathbf{K_0}} \leq \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r}}(t;\mathbf{\Phi}))\|_{\mathbf{K_0}}\,. \tag{3.43}$$

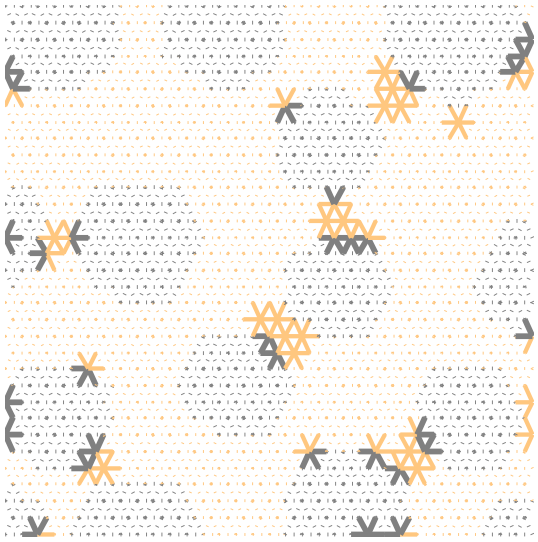
This guarantees that the error generated by the system approximation is controlled by the error generated by approximating the displacement. The location of controlled elements (which are all the elements in contact with the control degrees of freedom) is shown in Figure 3.11 for various basis sizes. It is interesting to remark that the controlled elements gather around inclusions where damage is the highest. Figure 3.12 illustrate this effect.



(a) Controlled elements with 3 displacement basis vectors. 10 “static“ basis vectors are needed to achieve the minimal accuracy condition (3.43)



(b) Controlled elements with 5 displacement basis vectors. 28 “static“ basis vectors are needed to achieve the minimal accuracy condition (3.43)



(c) Controlled elements with 15 ”dynamic” basis vectors. 60 “static“ basis vectors are needed to achieve the minimal accuracy condition (3.43)

Figure 3.11: Controlled elements selected using various basis sizes. The bigger the basis, the more controlled elements are needed. The elements tend to gather around the regions where the variation of displacement is the highest, hence where the variation of the internal forces will be high.

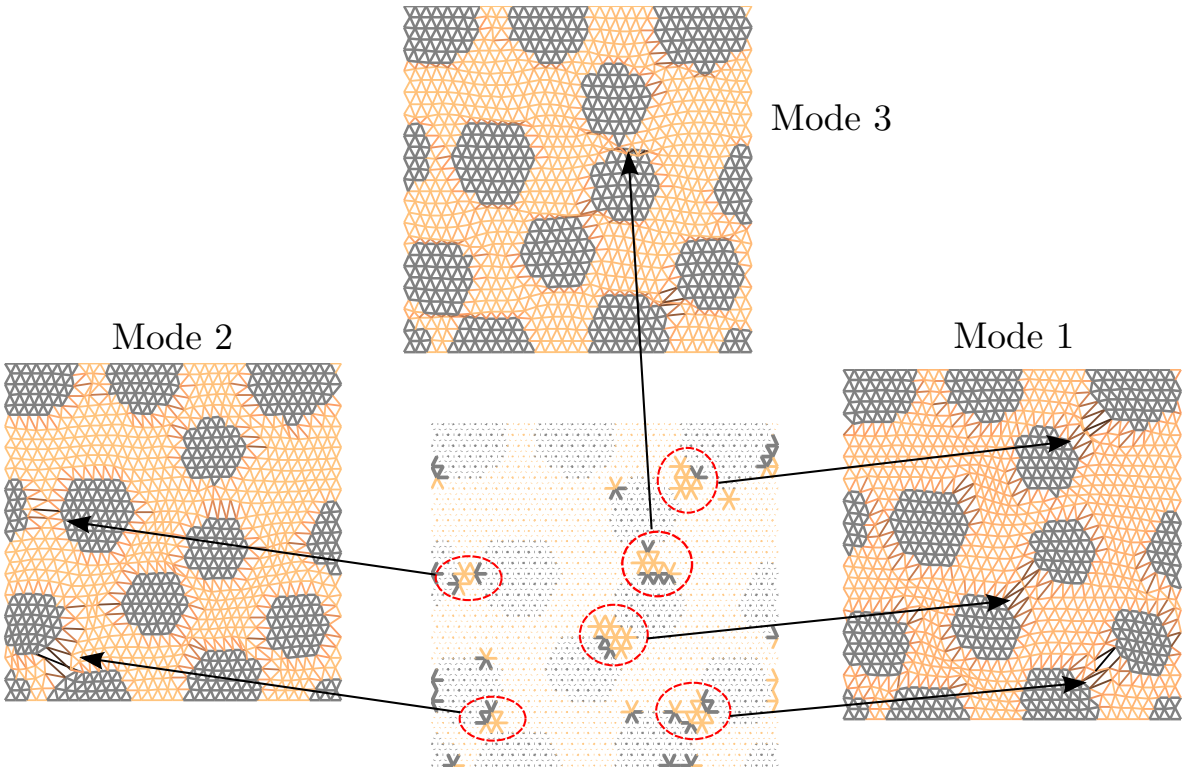
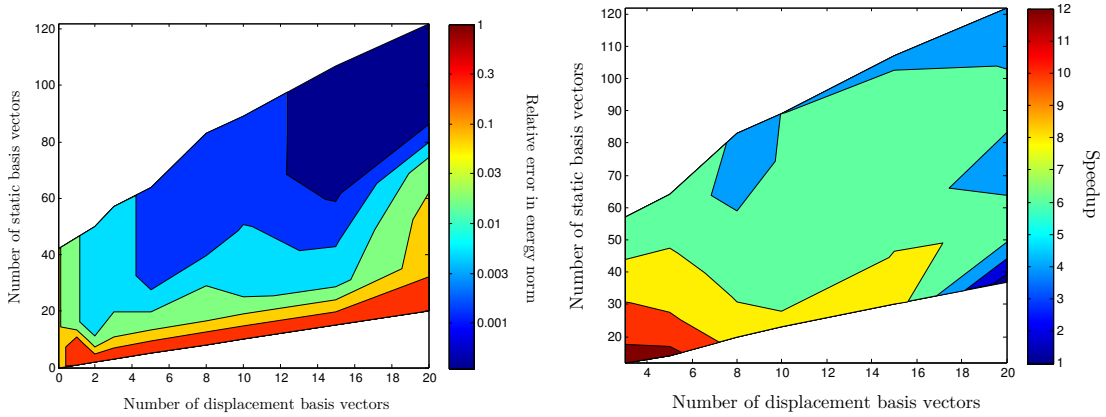


Figure 3.12: Regions of interest selected by the system approximation procedure. Those regions (circled in the Figure) are matching the areas of higher displacement found in the POD bases. This is intuitively good, since those elements have to give enough information to be able to reconstruct the internal forces over the entire domain. Those are the elements whose behaviour vary the most when changing the loading path (which is the parameter of the reduced model), hence containing the core information necessary to build up an accurate reconstruction.

Numerical savings In this section, we will test the performance of the method by comparing the relative error between the "truth" solution of the RVE problem, which is the solution obtained when using the full order model, and the reduced order model.

The following load path considered for testing the efficiency of the model is set using the following effective strain: $\boldsymbol{\epsilon}^{\mathbf{M}}(t) = \frac{t}{T} . \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Note that this case is not in the snapshot.

We then proceed to solve the RVE boundary value problem subjected to this loading path using both the full order model and the reduced-order model while varying the sizes of the displacement and static bases. Induced errors and time gained are displayed in Figure 3.13.



(a) Evolution of the error varying the number of displacement and static basis vectors.

(b) Ratio of time time gained from running the full order model and the reduced model.

Figure 3.13: Numerical results tested on a loading path not included in the snapshot space. Here, the snapshot selection was arbitrary and relatively fine which allows to consider various number of basis sizes.

Several remarks can be made:

- As expected, the error decreases when the number of either the displacement or static bases vectors increases. A higher dimensional representation of the solution leads unsurprisingly to more accuracy.
- The time gained using the reduced model becomes more and more important when the number of vectors in the bases decreases.
- Looking at Figure 3.13(b), it can be seen that the speedup is roughly dependent on the size of the static bases, rather than on the displacement basis. Indeed, the number of controlled elements, which is linked to the amount of computations to be done, is directly linked to the dimension of the static basis $\mathbf{\Psi}$.
- To have a well defined reduced order model, the dimension of the static basis $\mathbf{\Psi}$ should at least match the dimension of the displacement basis $\mathbf{\Phi}$. However, it can be seen that to achieve a reasonable tolerance on the error, the dimension of the static basis should actually be relatively larger.

The error with respect to the speedup is displayed in Figure 4.12. What we call speedup here is the ratio of the elapsed time of the full order simulation over the elapsed time of the reduced model. It represents how many times faster is the reduced order model compared to the full order model.

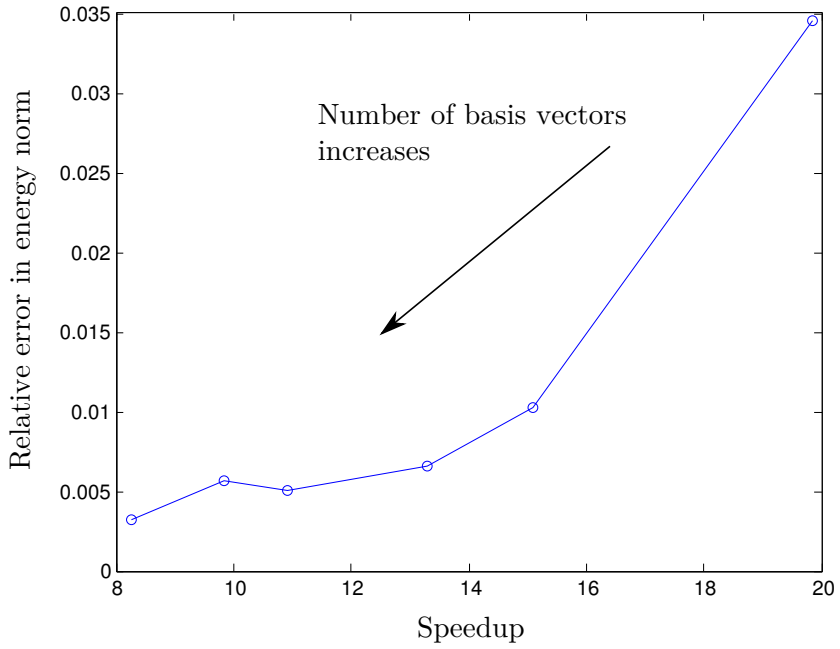


Figure 3.14: Evolution of the error with respect to the speedup while increasing the number of basis vectors. The number of vectors in the static basis is chosen according to the rule defined in section 3.4.1.

It can be seen that there is a proportional relation between speedup and error: as the number of basis functions increases, the speedup and the error decrease. The user can reduce the error with the price of having a slower simulation. What makes the reduced model faster is purely the bypassing of most of the elements when computing the internal forces or the tangent stiffness (this bypassing is possible thanks to the system approximation technique). Note that the speedup is not purely equal to the ratio between controlled elements and total number of elements since the Newton-Raphson procedure requires more steps to converge in the reduced order model scheme than in the full order model. Another remark is that beyond a certain dimension of the reduced space, the error does not decrease very much and reaches a plateau. This means that no matter how many vectors in the basis, a maximum accuracy is achieved. This can be explained by the fact that the loading path tested is not part of the snapshot. The only way to decrease this residual error is to enrich the snapshot space. We define $\mathbf{u}_{\text{ex,snap}}(t)$ as the projection of the exact solution onto the snapshot space. Using the same principle than equation (3.42), we can decompose the error further (dropping

parameters for clarity):

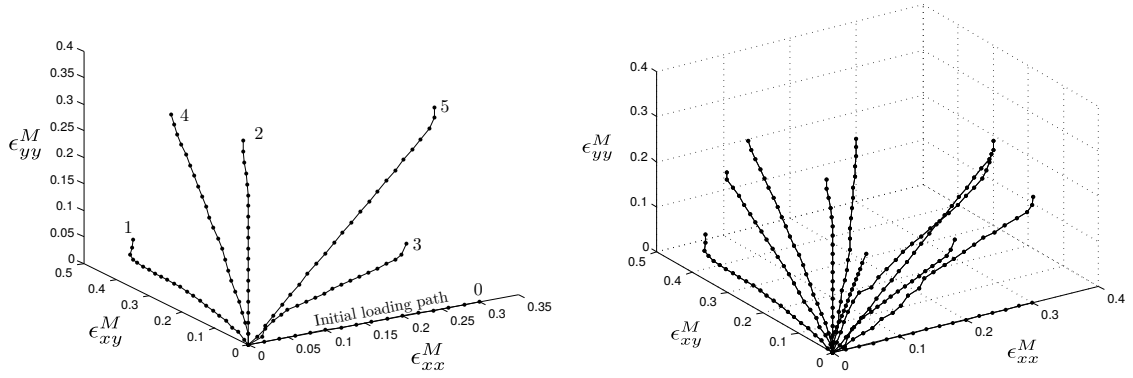
$$\nu_{\text{tot}}(t)^2 = \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{ex,snap}}(t)) + (\mathbf{u}_{\text{ex,snap}}(t) - \mathbf{u}_{\text{r}}(t)) + (\mathbf{u}_{\text{r}}(t) - \mathbf{u}_{\text{r,sa}}(t))\|^2 \tag{3.44}$$

$$= \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{ex,snap}}(t))\|^2 + \|(\mathbf{u}_{\text{ex,snap}}(t) - \mathbf{u}_{\text{r}}(t)) + (\mathbf{u}_{\text{r}}(t) - \mathbf{u}_{\text{r,sa}}(t))\|^2, \tag{3.45}$$

$\|(\mathbf{u}_{\text{ex,snap}} - \mathbf{u}_{\text{r}}) + (\mathbf{u}_{\text{r}} - \mathbf{u}_{\text{r,sa}})\|^2$ can be made as small as desired by taking high dimensional bases $\mathbf{\Phi}$ and $\mathbf{\Psi}$. The residual error that remains is $\|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{ex,snap}}(t))\|^2$, which entirely depends on the richness of the snapshot space.

3.4.2 Application of the POD-greedy algorithm

In this section, we show the results of the POD-greedy procedure, initialising the first load path to be pure tension in the $\boldsymbol{\epsilon_{xx}}$ direction. Successive loading paths generated by the POD-greedy method are displayed in Figure 3.15.



(a) First 5 loading paths generated by the greedy algorithm, starting arbitrarily from a uniaxial loading in ϵ_{xx}.

(b) First 10 loading paths generated by the greedy algorithm.

Figure 3.15: POD-Greedy sampling results. One can see that generated paths spread very well inside the parameter space of all possible loading paths.

The loading paths are spatially well spread, which is intuitively a necessary condition for generating the maximum of information with a minimum of snapshots. In Figure 3.16, we display the error of the paths of worst prediction (using the formula from equation (3.38)) generated successively during the standard POD-greedy and the adaptive POD-greedy procedure with an underlying tolerance $\nu_{\max} = 10^{-2}$.

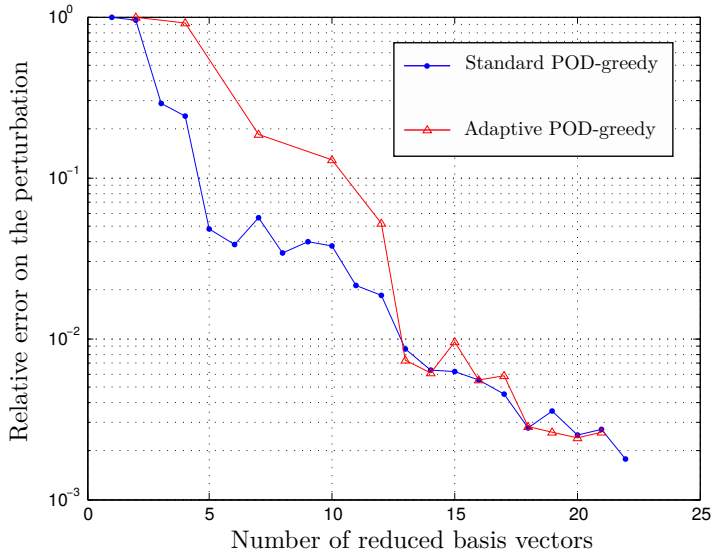


Figure 3.16: Evolution of the error of the path of worst prediction during both POD-greedy procedures. The underlying tolerance used in the adaptive procedure is $\nu_{\max} = 10^{-2}$. For a given size of the reduced basis, the standard POD-greedy strategy achieves a better prediction than the adaptive procedure. However, it has to be noted that the adaptive method requires much fewer snapshots. Note that in this example, it is the squared error on the perturbation $\widetilde{\mathbf{u}}$ that is plotted.

Several remarks can be made:

- The error generally decreases as the enrichment progresses. It is very much expected as when the size of the basis increases, the error on the solution that is the least well approximated by that basis decreases.
- The adaptive POD-greedy procedure converges slower than the standard one but reaches the target tolerance $\nu_{\max}$ requiring fewer iterations of the POD-greedy procedure. This means, the offline stage was sped up to the price of having a reduced basis of slightly larger dimension.

Comparative results between the Reduced basis and the Snapshot-POD strategies We now investigate the comparative performance of the POD versus the reduced basis strategy. Note that the Reduced basis method naturally provides a "worst case scenario" error, by its iterative procedure, which is based on the enrichment of the current basis with the solution the least well approximated. To make a fair comparison with snapshot-POD, the LOOCV error estimate 3.23 defined in section 3.3.2, which give an estimate of that maximal error based on the random snapshot, is used. Results are

displayed in Figure 3.17. The Reduced basis estimate of the maximal error is about 10 times smaller than the one predicted for the snapshot-POD model. An interpretation of this result is that the random selection of the snapshot inside the parameter domain is not exhaustive enough and leads to redundant information, which implies that some "outcast" load paths can generate a solution that is poorly represented by the reduced space. This result should be mitigated by the fact that the POD-greedy procedure used in the reduced basis approach forced loadings to remain positive in each component, which may reduce artificially the dimension of the parameter space.

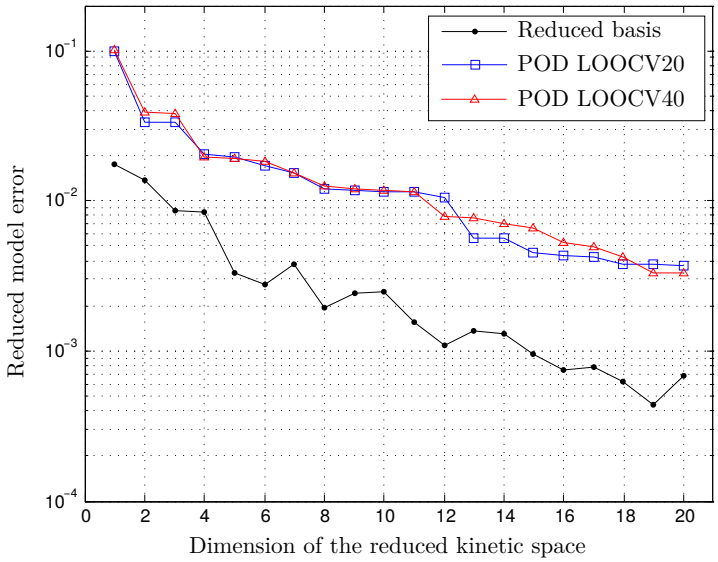


Figure 3.17: Comparison of the estimate of maximal error between the RB strategy and the POD strategy for various reduced space dimensions. The RB error estimate is coming from a direct computation of the error between reduced and exact solution on the path of worst approximation. The POD error estimate is computed performing a LOOCV cross-validation from a set 20 random snapshots and a set of 40 random snapshots. We can see that the RB strategy leads to a better accuracy. The relative similarity between the estimates of the POD strategy with a snapshot set with 20 and 40 solutions can let think that a richer snapshot may not lead to a more accurate reduced space.

3.5 Conclusion and perspectives of this chapter

In this chapter, we looked at the problem of reducing the computational cost of solving the RVE boundary value problem involved in computational homogenisation using projection-based model order reduction. A damage law was used to model the material

microstructure. This problem is parametrised by the history of strain coming from a simulation of a macrostructure (not described here) that is applied as boundary condition onto the RVE. The main challenge is coming from the very high-dimensionality of the parameter domain which consists of all possible far-field loadings over time until reaching loss of ellipticity in the structure. This makes the sampling of the parameter space a very complicated task, when in the framework of snapshot-POD model order reduction. In this chapter, we proposed to compare two strategies:

- The sampling is done randomly, step by step, enforcing that a minimum dissipation occurs at each random timestep. The reduced space is found using proper orthogonal decomposition (POD). A LOOCV cross-validation error estimate is evaluated for the user to decide on the appropriate dimension of the reduced space according to a desired accuracy.

- The problem is solved using a reduced-basis type of method. The sampling is done by iteratively enriching the reduced basis with a solution generated using what was defined as the loading path of "worst approximation" (PWA). As in a POD-greedy setting, a POD is performed on the enriched solution. The user then chooses a global tolerance that stops the enrichment. Since computing the path of worst approximation is computationally quite expensive (though performed offline), a development was explained where not only the eigenvector with highest magnitude may be concatenated to the reduced basis but as many as required so that the reduced solution from the PWA matches the truth one up to the global tolerance.

Numerical results show that the RB strategy is more efficient at finding a relevant set of sample solutions than a more "brute force" random sampling. Indeed, with the same dimension of the reduced space, the maximum error obtained using the RB method is smaller than when using the standard POD. As the problem is non-linear, a system approximation was used (DEIM) in order to have a get speedup out of our reduced model. It was shown that in this setting, the control elements generated by the DEIM algorithm have a physical sense as they tend to gather in the regions of high damage.

Future work in this area would involve using a quantity of interest relevant to a numerical code on the coarse scale. In this work we used an energy-norm.

CHAPTER 4

Partitioned model order reduction applied to fracture mechanics

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 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 chapter 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 [57]), fluid mechanics [54, 58, 59, 60, 61] and structural dynamics [62, 63, 2, 64, 65, 15]. The theory and applicability of various projection-based model order reduction methods such as component mode synthesis [14, 62], the reduced basis method [9, 10, 13], the proper orthogonal decomposition [66, 67, 54] which will be used in this work, the *a priori* hyperreduction method [51, 68] or the proper generalised decomposition [16, 17, 18] 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 [47, 51, 4, 69]. 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 chapter). 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. An early contribution comes from the work of Craig and Bampton [62], who proposed a reduction by projection on a modal basis defined over predefined subdomains. This idea has been explored and

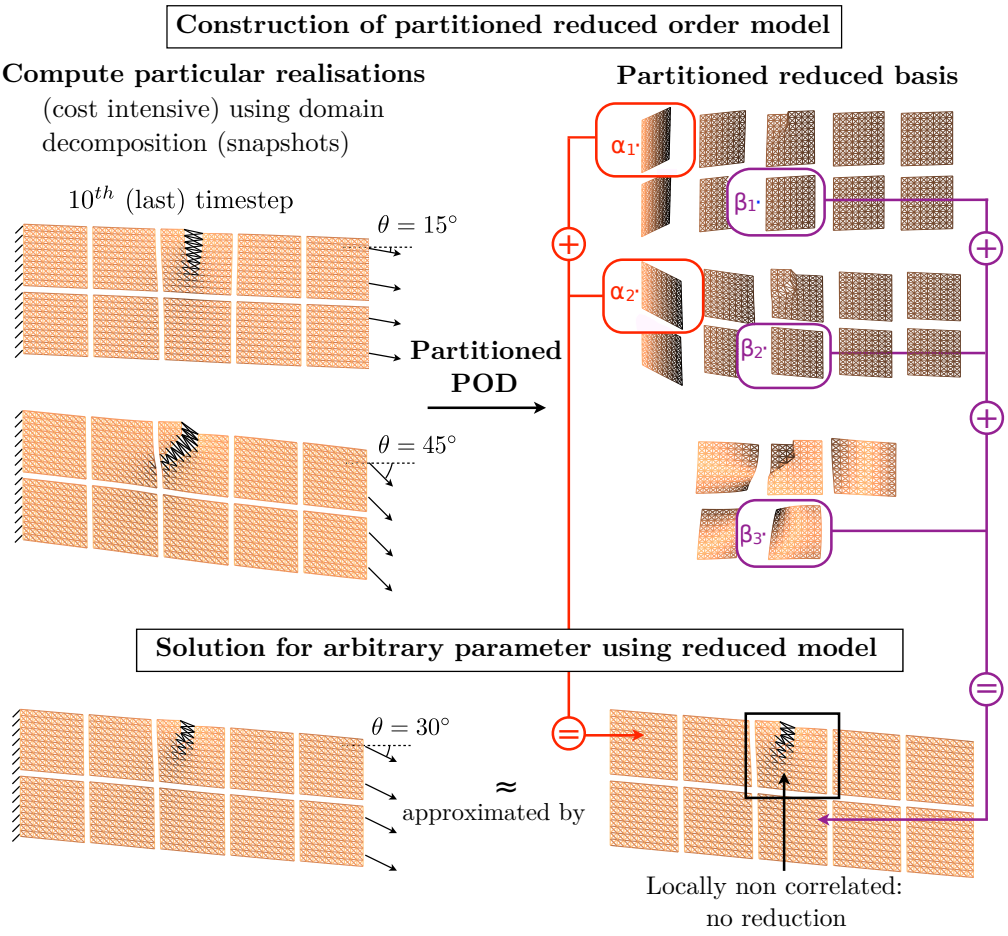


Figure 4.1: Schematic representation of the partitioned POD-based model order reduction strategy. A Snapshot POD is performed locally for each subdomain in an "offline" phase, which requires the "truth" solution corresponding to a set of particular parameter values. In the"online" phase, the solution 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

improved in [70, 65, 15], or coupled with other reduction methods, as in the case of the proper generalised decomposition [16]. 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) [64, 71, 72, 61], or the global reduced model is locally enriched by a fine-scale description [73, 74, 75] (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 [72]). The work presented here is original 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 [3, 74, 50, 76, 5]) are powerful tools to reduce the computational burden associated with the repetitive analysis of parametrised nonlinear problems. These can be applied to both static and dynamic behaviours. 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 conception of the design process is that domain decomposition methods [77, 78, 79, 80], 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 [81, 82]. 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 chapter can be extended to Schwartz-based

methods, as done for the proper generalised decomposition in the LaTin framework [16], or to other Schur-dual based domain decomposition methods, as presented in [65] for component mode synthesis. We will focus in this work on the primal Schur-based domain decomposition method proposed in [77, 78]. 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 4.1) 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 [54] can be prohibitively expensive to compute. This issue was treated in [58] 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 [83], 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 [84] and [85], 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 chapter is organised as follows. In section 4.1, we give the general assumptions regarding the class of nonlinear problems which are addressed in this chapter. Section

4.2 introduces classical model order reduction by projection. We focus on the snapshot POD methodology. 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 4.3, 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 4.4. Finally, we propose results in terms of running time in section 4.5 (as a first step, the partitioned POD is used in a serial computing approach), and discuss further improvements for the proposed strategy.

4.1 General problem statement

We consider the evolution of a general 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 $\boldsymbol{\mu}$ that evolves in the parameter domain denoted $\mathcal{P} \subset \mathbb{R}^{n_\mu}$.

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

$$\forall\, (t, \boldsymbol{\mu}) \in \mathcal{T} \times \mathcal{P}, \qquad \mathbf{f_{int}} \left(\left(\mathbf{u}(\tau; \boldsymbol{\mu}) \right)_{\tau \in [0,t]} ; \boldsymbol{\mu} \right) + \mathbf{f_{ext}}(t; \boldsymbol{\mu}) = \mathbf{0}\,. \tag{4.1}$$

The vector of internal forces, $\mathbf{f_{int}} \in \mathbb{R}^{n_\mathrm{u}}$, is a non-linear function of the current state variables $\mathbf{u}(t; \boldsymbol{\mu})$ (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 (4.1). As we model structural damage, the vector of internal forces at time t also depends on the history of the state variables $(\mathbf{u}(\tau; \boldsymbol{\mu}))_{\tau \in [0,t[}$ over the past time interval $[0, t[$. Typically, the dependence of $\mathbf{f_{int}}$ to the history of the displacement is due to non-reversible material processes. In the context of parametric problems, $\mathbf{f_{int}}$

may additionally depend on the design variables (design-dependent elastic constants for instance). $\mathbf{f_{ext}} \in \mathbb{R}^{n_\mathrm{u}}$ 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 (4.1) is performed. We search for a sequence of solutions $(\mathbf{u}(t;\boldsymbol{\mu}))_{t\in\mathcal{T}^\mathrm{h}}$, 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

$$\forall\, (t,\boldsymbol{\mu}) \in \mathcal{T}^\mathrm{h} \times \mathcal{P}, \qquad \mathbf{f_{int}}\left(\mathbf{u}(t;\boldsymbol{\mu}), (\mathbf{u}(\tau;\boldsymbol{\mu}))_{\tau\in\mathcal{T}^h,\tau<t}\,;\boldsymbol{\mu}\right) + \mathbf{f_{ext}}(t;\boldsymbol{\mu}) = \mathbf{0} \tag{4.2}$$

System (4.2) 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 $(\mathbf{u}(\tau;\boldsymbol{\mu}))_{\tau\in\mathcal{T}^h,<t}$ is known, which allows to compute vector $\mathbf{u}(t;\boldsymbol{\mu})$. 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, $(\mathbf{u}(t;\boldsymbol{\mu}))_{t\in\mathcal{T}^\mathrm{h}}$ is referred to as the "truth" solution as it is the one that will be approximated in the reduced order modelling approach.

Discrete system (4.2) 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:

$$\text{Find } \boldsymbol{\Delta}\mathbf{u}^{i+1} \in \mathbb{R}^{n_\mathrm{u}} \text{ such that } \quad \mathbf{K}^i\,\boldsymbol{\Delta}\mathbf{u}^{i+1} = -\mathbf{R}^i\,, \tag{4.3}$$

where $\boldsymbol{\Delta}\mathbf{u}^{i+1} = \mathbf{u}^{i+1} - \mathbf{u}^i$ is an increment in the displacement vector ($\mathbf{u}^{i+1}$ is the actual solution of linear prediction $i+1$), $\mathbf{K}^i = \left.\frac{\partial \mathbf{f_{int}}(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^i}$ is the tangent operator and $\mathbf{R}^i = \mathbf{f_{int}}(\mathbf{u}^i) + \mathbf{f_{ext}}$ 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 $i+1$, $\frac{\|\mathbf{R}^{i+1}\|_2}{\|\mathbf{f_{ext}}\|_2}$, is lower than a chosen tolerance ϵ_new.

4.2 Model Order Reduction and Proper Orthogonal Decomposition

Let us recall that our goal in this chapter is to solve problem (4.2) 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 $\mathbb{R}^{n_\mathrm{u}}$. 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 (4.2) 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).

4.2.1 Projection-based model order reduction

Let us write that the solution of (4.2) can be approximated, at any time $t \in \mathcal{T}^\mathrm{h}$ and for any value of the parameter $\mu \in \mathcal{P}$, in a subspace of $\mathbb{R}^{n_\mathrm{u}}$ spanned by (a few) identified basis vectors $(\mathbf{c}_i(t;\mu))_{i\in[\![1,n_\mathrm{c}]\!]}$ belonging to $\mathbb{R}^{n_\mathrm{u}}$:

$$\forall\,(t,\boldsymbol{\mu}) \in \mathcal{T}^\mathrm{h} \times \mathcal{P}, \qquad \mathbf{u}(t;\boldsymbol{\mu}) \approx \sum_{i=1}^{n_\mathrm{c}} \mathbf{c}_i(t;\boldsymbol{\mu})\,\alpha_i(t,\boldsymbol{\mu}) = \mathbf{C}(t;\boldsymbol{\mu})\,\boldsymbol{\alpha}(t,\boldsymbol{\mu})\,. \tag{4.4}$$

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

Injecting this approximation into (4.2) at a particular point $(t,\boldsymbol{\mu})$ of the time-parameter domain $\mathcal{T}^\mathrm{h} \times \mathcal{P}$, one obtains an over-constrained set of equations in the n_c reduced state variables $\boldsymbol{\alpha}$ ($n_\mathrm{c} \ll n_\mathrm{u}$). Let us define the residual of (4.2) by

$$\forall\,\boldsymbol{\alpha}^\star \in \mathbb{R}^{n_\mathrm{c}}, \quad \widetilde{\mathbf{r}}(\boldsymbol{\alpha}^\star) \overset{\mathrm{def}}{=} \mathbf{r}(\mathbf{C}\,\boldsymbol{\alpha}) = \mathbf{f_{int}}\,(\mathbf{C}\,\boldsymbol{\alpha}^\star) + \mathbf{f_{ext}} \tag{4.5}$$

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 (4.5) and its least-square minimisation. The latter reads:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}^* \in \mathbb{R}^{n_c}}{\operatorname{argmin}} \left(\|\widetilde{\mathbf{r}}(\boldsymbol{\alpha}^*)\|_{\boldsymbol{\Theta}} \right) , \tag{4.6}$$

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

$$\mathbf{C}^T \widetilde{\mathbf{r}}(\boldsymbol{\alpha}) = \mathbf{0} . \tag{4.7}$$

We use the Galerkin approach. Nonlinear problem (4.7) can be solved by a classical Newton algorithm. The linearisation of reduced problem (4.7) at iteration $i+1$ of a Newton solver (see for instance [4] for more details) leads to the following problem:

$$\mathbf{C}^T \left(\widetilde{\mathbf{r}}^i + \mathbf{K}^i \, \mathbf{C} \, \boldsymbol{\Delta\alpha}^{i+1} \right) = \mathbf{0} \tag{4.8}$$

where $\boldsymbol{\Delta\alpha}^{i+1} = \boldsymbol{\alpha}^{i+1} - \boldsymbol{\alpha}^i$ is the unknown quantity of the linear prediction and $\widetilde{\mathbf{r}}^i \overset{\text{def}}{=} \widetilde{\mathbf{r}}(\boldsymbol{\alpha}^i)$. Linearised system (4.8) is a Galerkin reduction of linearised equation (4.3) with the kinematic constraint $\boldsymbol{\Delta}\mathbf{u}^{i+1} = \mathbf{C} \, \boldsymbol{\Delta\alpha}^{i+1}$. It can also be seen as a least-square reduction as these two approaches are equivalent for the linearised problem when using a $\mathbf{K}^{-1}$-norm. Indeed, expanding the norm of the residual, we get:

$$\begin{aligned} \left\| \widetilde{\mathbf{r}}^i + \mathbf{K}^i \, \mathbf{C} \, \boldsymbol{\Delta\alpha}^{i+1} \right\|^2_{(\mathbf{K}^i)^{-1}} &= \widetilde{\mathbf{r}}^{iT} \left(\mathbf{K}^i \right)^{-1} \mathbf{r}^i + \boldsymbol{\Delta\alpha}^{i+1^T} \mathbf{C}^T \mathbf{K}^{iT} \left(\mathbf{K}^i \right)^{-1} \mathbf{K}^i \mathbf{C} \boldsymbol{\Delta\alpha}^{i+1} \\ &\qquad + 2 \boldsymbol{\Delta\alpha}^{i+1^T} \mathbf{C}^T \mathbf{K}^{iT} \left(\mathbf{K}^i \right)^{-1} \widetilde{\mathbf{r}}^i \qquad (4.9) \\ &= \widetilde{\mathbf{r}}^{iT} \left(\mathbf{K}^i \right)^{-1} \mathbf{r}^i + \boldsymbol{\Delta\alpha}^{i+1^T} \mathbf{C}^T \mathbf{K}^{iT} \mathbf{C} \boldsymbol{\Delta\alpha}^{i+1} + 2 \boldsymbol{\Delta\alpha}^{i+1^T} \mathbf{C}^T \widetilde{\mathbf{r}}^i , \qquad (4.10) \end{aligned}$$

where we used the fact that $\mathbf{K}^i$ is symmetric. Now, we look for the state variable increment $\boldsymbol{\Delta\alpha}$ that minimises this expression:

$$\begin{aligned} & \frac{\partial \left\| \widetilde{\mathbf{r}}^i + \mathbf{K}^i \, \mathbf{C} \, \boldsymbol{\Delta\alpha} \right\|^2_{(\mathbf{K}^i)^{-1}}}{\partial \boldsymbol{\Delta\alpha}}_{|\boldsymbol{\Delta\alpha} = \boldsymbol{\Delta\alpha}^{i+1}} = \mathbf{0} && (4.11) \\ \iff & 2 \, \mathbf{C}^T \mathbf{K}^{iT} \mathbf{C} \boldsymbol{\Delta\alpha}^{i+1} + 2 \, \mathbf{C}^T \widetilde{\mathbf{r}}^i = \mathbf{0} && (4.12) \\ \iff & \mathbf{C}^T \left(\widetilde{\mathbf{r}}^i + \mathbf{K}^i \, \mathbf{C} \, \boldsymbol{\Delta\alpha}^{i+1} \right) = \mathbf{0}, && (4.13) \end{aligned}$$

which matches the Galerkin projection equation (4.8). Note that a different type of

projection could be chosen, such as a Petrov-Galerkin projection [5], where the residual is projected onto a space that may differ from the unknown field space, i.e in our case, some basis different than $\mathbf{C}$. We choose the Galerkin projection as it is the most commonly used in the reduced-order community and is known to give accurate results.

The solution to (4.8) reads

$$\boldsymbol{\Delta\alpha}^{i+1} = -\left(\mathbf{C}^T\,\mathbf{K}^i\,\mathbf{C}\right)^{-1}\mathbf{C}^T\widetilde{\mathbf{r}}^i\,, \tag{4.14}$$

providing that the reduced linearised operator $\mathbf{C}^T\,\mathbf{K}^i\,\mathbf{C}$ (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 (4.8) 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 $\mathbf{f_{int}}$ 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.

4.2.2 Proper Orthogonal Decomposition in projection-based model order reduction

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 $\tilde{\mathcal{P}} \stackrel{\text{def}}{=} \mathcal{T}^{\text{h}} \times \mathcal{P}$ as

$$\forall\, (t, \boldsymbol{\mu}) \in \tilde{\mathcal{P}}, \qquad \mathbf{u}(t;\boldsymbol{\mu}) = \bar{\mathbf{u}}(t;\boldsymbol{\mu}) + \boldsymbol{\epsilon}(t;\boldsymbol{\mu}) \tag{4.15}$$

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

such that $\bar{\mathbf{u}}$ is the function of separable form (4.15) that is the closest to the exact solution,

$$\bar{\mathbf{u}} = \underset{\bar{\mathbf{u}}^* \in \{\mathbf{Z}\,|\,\mathbf{Z}(t;\boldsymbol{\mu}) = \boldsymbol{\phi}\,\boldsymbol{\gamma}(t,\boldsymbol{\mu}),\, \forall\,(t,\boldsymbol{\mu}) \in \tilde{\mathcal{P}}\}}{\operatorname{argmin}} d(\mathbf{u}, \bar{\mathbf{u}}^*)\,, \tag{4.16}$$

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

$$\begin{array}{rccc} d: & \bar{\mathcal{U}} \times \bar{\mathcal{U}} & \to & \mathbb{R} \\ & (\mathbf{u}, \bar{\mathbf{u}}) & \mapsto & d(\mathbf{u}, \bar{\mathbf{u}}) \end{array} \tag{4.17}$$

$$d(\mathbf{u}, \bar{\mathbf{u}}) = \int_{\boldsymbol{\mu} \in \mathcal{P}} \sum_{t \in \mathcal{T}^{\text{h}}} \|\mathbf{u}(t;\boldsymbol{\mu}) - \bar{\mathbf{u}}(t;\boldsymbol{\mu})\|_2^2 \; d\boldsymbol{\mu}\,. \tag{4.18}$$

$(\boldsymbol{\phi}_i)_{i \in [\![1, n_{\text{p}}]\!]}$ are "space" vectors that belong to $\mathbb{R}^{n_{\text{u}}}$ and are further constrained to be orthonormal with respect to the usual euclidean scalar product of $\mathbb{R}^{n_{\text{u}}}$, while $(\gamma_i)_{i \in [\![1, n_{\text{p}}]\!]}$ are scalar functions of time and parameter. We emphasise here the fact that the spatial basis $\boldsymbol{\phi}$ 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 $\mathbf{u}$ into bi-orthonormal modes $((\boldsymbol{\phi}_i), \gamma_i)_{i \in [\![1, n_{\text{p}}]\!]}$ 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 $d(\mathbf{u}, \bar{\mathbf{u}})$ is expected to decrease quickly with the order of the decomposition.

Snapshot POD

The POD transform (4.15,4.18) 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 $\mathbf{u}$ over a discrete

subset $\tilde{\mathcal{P}}^{\mathrm{s}} = \mathcal{T}^{\mathrm{h}} \times \mathcal{P}^{\mathrm{s}}$ of $\tilde{\mathcal{P}}$.

$$\forall\, (t, \boldsymbol{\mu}) \in \tilde{\mathcal{P}}^{\mathrm{s}}, \qquad \bar{\mathbf{u}}(t;\boldsymbol{\mu}) = \bar{\mathbf{u}}^{\mathrm{s}}(t;\boldsymbol{\mu}) + \boldsymbol{\epsilon}^{\mathrm{s}}(t;\boldsymbol{\mu}) \tag{4.19}$$

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

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

$$\bar{\mathbf{u}}^{\mathrm{s}} = \underset{\bar{\mathbf{u}}^* \in \{\mathbf{Z}\,|\,\mathbf{Z}(t;\boldsymbol{\mu}) = \boldsymbol{\phi}\,\boldsymbol{\gamma}(t;\mu),\, \forall\,(t;\boldsymbol{\mu}) \in \tilde{\mathcal{P}}^{\mathrm{s}}\}}{\operatorname{argmin}} d^{\mathrm{s}}(\mathbf{u}, \bar{\mathbf{u}}^*) \tag{4.20}$$

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 $\mathbb{R}^{n_{\mathrm{u}}}$:

$$\begin{array}{rccc} d^{\mathrm{s}}: & \bar{\mathcal{U}}^{\mathrm{s}} \times \bar{\mathcal{U}}^{\mathrm{s}} & \to & \mathbb{R} \\ & (\mathbf{u}, \bar{\mathbf{u}}) & \mapsto & d(\mathbf{u}, \bar{\mathbf{u}}) \end{array} \tag{4.21}$$

with

$$d^{\mathrm{s}}(\mathbf{u}, \bar{\mathbf{u}}^{\mathrm{s}}) = \sum_{\boldsymbol{\mu} \in \mathcal{P}^{\mathrm{s}}} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \|\mathbf{u}(t;\boldsymbol{\mu}) - \bar{\mathbf{u}}^{\mathrm{s}}(t;\boldsymbol{\mu})\|_2^2 \tag{4.22}$$

$\mathcal{P}^{\mathrm{s}} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{n_\mu}\}$ is a discrete subset of the parameter domain $\mathcal{P}$. $(\mathbf{u}(t;\boldsymbol{\mu}))_{(t;\boldsymbol{\mu}) \in \tilde{\mathcal{P}}^{\mathrm{s}}}$ are particular "truth" solutions of problem (4.2) for some parameters $\boldsymbol{\mu} \in \tilde{\mathcal{P}}$, called snapshot. The snapshot POD metric (4.22) can be viewed as a quadrature rule for its integral counterpart (4.18).

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

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

$$\mathbf{H} = \sum_{\boldsymbol{\mu} \in \mathcal{P}^{\mathrm{s}}} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \mathbf{u}(t;\boldsymbol{\mu})\, \mathbf{u}(t;\boldsymbol{\mu})^T\,. \tag{4.23}$$

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

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

$$d^{\mathrm{s}}(\mathbf{u}, \bar{\mathbf{u}}^{\mathrm{s}}) = \sum_{i=n_{\mathrm{p}}+1}^{n_{\mathrm{s}}} \lambda_i\,, \tag{4.24}$$

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

The eigenvalue decomposition of $\mathbf{H}$ 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 [54]). One computes the singular value decomposition (SVD) of the snapshot operator $\mathbf{S} = \Big(\mathbf{u}(t_1, \boldsymbol{\mu}_1)\ \mathbf{u}(t_2, \boldsymbol{\mu}_1)\ \ldots\ \mathbf{u}(t_{n_\mathrm{t}}, \boldsymbol{\mu}_{n_\mu})\Big)$. The SVD reads $\mathbf{S} = \mathbf{Q}\,\boldsymbol{\Sigma}\,\mathbf{W}^T$ with $\mathbf{Q}$ and $\mathbf{W}$ unitary matrices and $\boldsymbol{\Sigma}$ a rectangular matrix with diagonal upper block. We then have $\mathbf{H} = \mathbf{S}\,\mathbf{S}^T = \mathbf{Q}\,\boldsymbol{\Sigma}\,\mathbf{W}^T\,\mathbf{W}\,\boldsymbol{\Sigma}^T\,\mathbf{Q}^T = \mathbf{Q}\,\boldsymbol{\Sigma}\,\boldsymbol{\Sigma}^T\,\mathbf{Q}^T$, which is the eigenvalue decomposition of $\mathbf{H}$ and the eigenvalues are the squares of the singular values of $\mathbf{S}$. 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 $\mathbf{W}$ if necessary, but this information is not of particular interest in the present context.

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

$$\nu^2_\mathrm{snap} = d^\mathrm{s}(\mathbf{u}, \bar{\mathbf{u}}^\mathrm{s}) = \frac{\displaystyle\sum_{i=n_\mathrm{p}+1}^{n_\mathrm{s}} \lambda_i}{\displaystyle\sum_{i=1}^{n_\mathrm{s}} \lambda_i}\,, \tag{4.25}$$

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 $\boldsymbol{\mu} \in \tilde{\mathcal{P}}$, in the reduced space $\mathrm{span}((\boldsymbol{\phi}_i)_{i\in[\![1,n_\mathrm{p}]\!]})$. The amplitude associated with the basis vectors are computed optimally by the "online" projection technique given in section 4.2.1. 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):

$$\forall\,(t, \boldsymbol{\mu}) \in \mathcal{T}^\mathrm{h} \times \mathcal{P},\ \forall\, i \in [\![1, n_\mathrm{c}]\!] \qquad \mathbf{C}_i(t; \boldsymbol{\mu}) = \boldsymbol{\phi}_i \qquad (n_\mathrm{c} = n_\mathrm{p}) \tag{4.26}$$

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 [76, 55]), which would lead to a decomposition of type* (4.15)*. 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 4.2.1 permits to reuse the error estimates available in finite element schemes for the certification of the implicitly interpolated solution (see [9, 74, 2, 12] 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.

4.2.3 System approximation

As stated in section 4.2.1, an approximation of the fully discrete system of equations (4.2) 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 $\mathbf{f_{int}}$, two families of strategies have been intensively studied in the literature.

Linearisation

The first family proposes to linearise [86, 2], or perform a higher-order Taylor expansion [87, 47, 88] 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 [87] 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 "on-

line" phase, the nonlinear terms of the discrete set of equations are approximated as a weighted combination of the "offline" trajectory-dependent linearisations.

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) [60, 50, 23, 5]. The expansion of the nonlinear term reads:

$$\begin{aligned} &\forall\, t \in \mathcal{T}^h,\, \forall\, \boldsymbol{\alpha}^\star \in \mathbb{R}^{n_\mathrm{c}}, \\ &\qquad \mathbf{f_{int}}\left(\mathbf{C}\,\boldsymbol{\alpha}^\star, (\mathbf{C}\,\boldsymbol{\alpha}(\tau,\mu))_{\tau\in\mathcal{T}^h,\,\tau<t}\,;\boldsymbol{\mu}\right) &&\approx \sum_{i=1}^{n_\mathrm{d}} \mathbf{D}_i\,\beta_i(\boldsymbol{\alpha}^\star, (\boldsymbol{\alpha}(\tau,\mu))_{\tau\in\mathcal{T}^h,\,\tau<t};\boldsymbol{\mu}) \\ &&&\approx \mathbf{D}\,\boldsymbol{\beta}\left(\boldsymbol{\alpha}^\star, (\boldsymbol{\alpha}(\tau,\mu))_{\tau\in\mathcal{T}^h,\,\tau<t}\,;\boldsymbol{\mu}\right). \end{aligned} \tag{4.27}$$

The columns of $\mathbf{D} \in \mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{d}}$ are spatial functions corresponding to a truncated snapshot POD expansion of the image of the reduced space by $\mathbf{f_{int}}$, 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 $\{\mathbf{f_{int}}\left(\mathbf{C}\,\boldsymbol{\alpha}^\star, (\mathbf{C}\,\boldsymbol{\alpha}(\tau,\mu))_{\tau\in\mathcal{T}^h,\,\tau<t};\boldsymbol{\mu}\right) \,|\, t \in \mathcal{T}^h,\, \boldsymbol{\alpha}^\star \in \mathbb{R}^{n_\mathrm{c}}\}$. Interpolation coefficients $\boldsymbol{\beta}$ are found by enforcing that at any point $(t,\boldsymbol{\mu})$ of $\tilde{\mathcal{P}}$, the interpolation must be optimal with respect to a limited number n_sa of spatial degrees of freedom:

$$\boldsymbol{\beta}\left(\boldsymbol{\alpha}^\star, (\boldsymbol{\alpha}(\tau,\mu))_{\tau\in\mathcal{T}^h,\,\tau<t}\,;\boldsymbol{\mu}\right) = \underset{\boldsymbol{\beta}^\star\in\mathbb{R}^{n_\mathrm{d}}}{\operatorname{argmin}} \left(\|\mathbf{D}\,\boldsymbol{\beta}^\star - \mathbf{f_{int}}\left(\mathbf{C}\,\boldsymbol{\alpha}^\star, (\mathbf{C}\,\boldsymbol{\alpha}(\tau,\mu))_{\tau\in\mathcal{T}^h,\,\tau<t}\,;\boldsymbol{\mu}\right)\|_\mathbf{P}\right) \tag{4.28}$$

$\mathbf{P}$ 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. $\|\mathbf{X}\|_\mathbf{P} = \sqrt{\mathbf{X}^T\mathbf{P}\,\mathbf{X}}$ is the semi-norm associated with $\mathbf{P}$ for an arbitrary vector $\mathbf{X} \in \mathbb{R}^{n_\mathrm{u}}$. Substituting this approximation into the full system of equation (4.2), together with the reduced basis approximation for the displacement vector, the following reduced expression is obtained for the approximation of the "truth" residual (4.5) at a particular point of the time-parameter domain:

$$\forall\, \boldsymbol{\alpha}^\star \in \mathbb{R}^{n_\mathrm{c}}, \qquad \mathbf{R_{gap}}(\boldsymbol{\alpha}^\star) \overset{\mathrm{def}}{=} \mathbf{D}\left(\mathbf{D}^T\,\mathbf{P}\,\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{P}\,\mathbf{f_{int}}\left(\mathbf{C}\,\boldsymbol{\alpha}^\star\right) + \mathbf{f_{ext}}\,, \tag{4.29}$$

where operator $\mathbf{D}^T\,\mathbf{P}\,\mathbf{D}$ 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 $\mathbf{C}^T\mathbf{R}_{\mathbf{gap}}(\boldsymbol{\alpha}) = \mathbf{0}$. 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 [1], the Hyperreduction method [21] and an early version of the Missing Point Estimation technique [22] 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 (4.2). This can be expressed in a least-square approach:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}^\star \in \mathbb{R}^{n_c}}{\operatorname{argmin}} \left(\|\mathbf{R}_{\mathbf{gap}}(\boldsymbol{\alpha}^\star)\|_{\mathbf{P}} \right) , \tag{4.30}$$

or in the (Petrov-) Galerkin framework

$$\text{Find } \boldsymbol{\alpha} \in \mathbb{R}^{n_c} \text{ such that } \quad \mathbf{C}^T\mathbf{P}\,\mathbf{R}_{\mathbf{gap}}(\boldsymbol{\alpha}) = \mathbf{0}\,. \tag{4.31}$$

The strategies proposed in the literature for this two subset of techniques differ in the way of building operator $\mathbf{P}$, which requires a critical trade-off between optimality, stability and tractability. In [50], $\mathbf{P}$ is constructed such that the condition number of operator $\mathbf{D}^T\,\mathbf{P}\,\mathbf{D}$ is minimised. In the hyperreduction method [21], the non-zero entries of $\mathbf{P}$ correspond to the largest entries (in some sense) of the approximated nonlinear vector function. In [23], the points are selected to limit the growth of the residual error between a solution and its snapshot reconstruction.

Chosen strategy

We will focus in this work on the "gappy" technique, as used in [23] and [5]. 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

(4.29), in the Galerkin framework, reads:

$$\boldsymbol{\Delta\alpha}^{i+1} = -\left(\mathbf{C}^T\mathbf{D}(\mathbf{D}^T\mathbf{P}\mathbf{D})^{-1}\mathbf{D}^T\mathbf{P}\,\mathbf{K}\,\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{R}^i_{\mathbf{gap}}\,, \tag{4.32}$$

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

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.

4.2.4 Example of application of the POD in fracture mechanics

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 $\mathbf{n}^{(b)}$ 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 $\mathbf{n}^{(b)} = \frac{\mathbf{P_iP_j}}{\|\mathbf{P_iP_j}\|}$. We denote the local coordinate of point $M \in \Omega^{(b)}$ by $s^{(b)} = \|\mathbf{P_iM}\|$. The global reference frame associated to the physical space is denoted by $\mathcal{R}(0, \mathbf{e}_x, \mathbf{e}_y)$.

We look for a two dimensional displacement field $\mathbf{u}$, 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 $\mathbf{u}^{(b)}$ and $N^{(b)}$ respectively.

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

$$\frac{\partial N^{(b)}}{\partial s^{(b)}} + \mathbf{f}\cdot\mathbf{n}^{(b)} = 0\,, \tag{4.33}$$

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

$$\sum_{b\in\mathcal{B}^{(i)}_{\text{n}}} N^{(b)}_{|P}\ \bar{\mathbf{n}}^{(b)}_{|P} = 0\,, \tag{4.34}$$

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

$$\sum_{b \in \mathcal{B}_{\mathrm{n}}^{(i)}} N_{|P}^{(b)} \; \bar{\mathbf{n}}_{|P}^{(b)} + \mathbf{N}_{\mathrm{d}|P} = 0 \,. \tag{4.35}$$

In the previous equation $\mathbf{N}_{\mathrm{d}|P} \in \mathbb{R}^2$ is a prescribed force. In equilibrium equation (4.34) and (4.35), $\bar{\mathbf{n}}_{|P}^{(b)} = \mathbf{n}^{(b)}$ if $s_{|P}^{(b)} = 0$ (first extremity of the bar), and $\bar{\mathbf{n}}_{|P}^{(b)} = -\mathbf{n}^{(b)}$ otherwise (second extremity of the bar).

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

$$\forall (b, b') \in \mathcal{B}_{\mathrm{n}}^{(i)}, \qquad \mathbf{u}_{|P}^{(b)} = \mathbf{u}_{|P}^{(b')} = \mathbf{u}_{|P} \,. \tag{4.36}$$

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

$$\mathbf{u}_{|P} = \mathbf{u}_{\mathrm{d}|P} \,, \tag{4.37}$$

where $\mathbf{u}_{\mathrm{d}|P} \in \mathbb{R}^2$ 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

$$N^{(b)} = N^{(b)} \left(\left\{ \epsilon^{(b)} \left(\mathbf{u}_{|\tau}^{(b)} \right) \right\}_{\tau \leq t} \right) , \tag{4.38}$$

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

$$\epsilon^{(b)} \left(\mathbf{u}^{(b)} \right) = \frac{\partial \mathbf{u}^{(b)}}{\partial s^{(b)}} \cdot \mathbf{n}^{(b)} \tag{4.39}$$

Damage model

The fracture of the lattice structure is described by classical damage mechanics [89]. We postulate the existence of a free Helmholtz energy at any time $t \in \mathcal{T}$:

$$\psi(\epsilon^{(b)}, d) = \frac{1}{2} E(1-d) S \left(\epsilon^{(b)}\right)^2 \tag{4.40}$$

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 differentiating the free energy with respect to the state variables:

$$N = \frac{\partial \psi}{\partial \epsilon^{(b)}} = E(1-d) S \epsilon^{(b)} , \tag{4.41}$$

$$Y = -\frac{\partial \psi}{\partial d} = \frac{1}{2} ES \left(\epsilon^{(b)}\right)^2 . \tag{4.42}$$

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

$$d = \min \left\{ \max_{\tau \leq t} \left\{ \alpha \left(\frac{Y_{|\tau}}{Y_{\mathrm{c}}} \right)^{\beta} \right\} , 1 \right\} , \tag{4.43}$$

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 (4.33) by a kinematically admissible displacement field $\mathbf{u}^\star$, integrate over $\Omega^{(b)}$ and sum over $\mathcal{B}$:

$$\sum_{b \in \mathcal{B}} \int_{\Omega^{(b)}} \frac{\partial N^{(b)}}{\partial s^{(b)}} \, \mathbf{u}^{(b)\star} \cdot \mathbf{n}^{(b)} \, ds^{(b)} + \sum_{b \in \mathcal{B}} \int_{\Omega^{(b)}} \mathbf{f} \cdot \mathbf{n}^{(b)} \, \mathbf{u}^{(b)\star} . \mathbf{n}^{(b)} \, ds^{(b)} = 0 . \tag{4.44}$$

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

$$-\sum_{b\in\mathcal{B}}\int_{\Omega^{(b)}} N^{(b)}\frac{\partial \mathbf{u}^{(b)^\star}}{\partial s^{(b)}}\cdot\mathbf{n}^{(b)}\,ds^{(b)}+\sum_{b\in\mathcal{B}}\int_{\Omega^{(b)}}\mathbf{f}\cdot\mathbf{n}^{(b)}\;\mathbf{u}^{(b)^\star}.\mathbf{n}^{(b)}\,ds^{(b)}+\sum_{P\in\mathcal{P}^{\mathrm{F}}}\mathbf{N}_{\mathrm{d}|P}\cdot\mathbf{u}^\star_{|P}=0\,, \tag{4.45}$$

where we have additionally enforced the condition that test function $\mathbf{u}^\star$ 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:

$$\mathbf{u}^{(b)}(s^{(b)})=\mathbf{\Lambda}^{(b)}\left(s^{(b)}\right)\;\tilde{\mathbf{A}}^{(b)^T}\mathbf{u}\quad\text{with}\quad\mathbf{\Lambda}^{(b)}\left(s^{(b)}\right)=\begin{pmatrix}1-\tilde{s}^{(b)} & 0 & \tilde{s}^{(b)} & 0\\ 0 & 1-\tilde{s}^{(b)} & 0 & \tilde{s}^{(b)}\end{pmatrix} \tag{4.46}$$

where $\tilde{s}^{(b)}\overset{\text{def}}{=}\frac{s^{(b)}}{\|\mathbf{P_iP_j}\|}$ and $\tilde{\mathbf{A}}^{(b)}$ the assembly operator such that $\mathbf{u}^{(b)}=\tilde{\mathbf{A}}^{(b)^T}\mathbf{u}$ with $\mathbf{u}^{(b)}=\begin{pmatrix}\mathbf{u}_{|P_i}\cdot\mathbf{e}_x & \mathbf{u}_{|P_i}\cdot\mathbf{e}_y & \mathbf{u}_{|P_j}\cdot\mathbf{e}_x & \mathbf{u}_{|P_j}\cdot\mathbf{e}_y\end{pmatrix}^T$ the vector of nodal values of the restriction of the displacement to bar b, $\mathbf{u}$ 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 $t\in\mathcal{T}$:

$$\forall\,\mathbf{u}^\star\in\mathbb{R}^{\tilde{n}_{\mathrm{u}}}\text{ such that }\left(\hat{\mathbf{A}}^{(P)^T}\mathbf{u}^\star=0,\,\forall\,P\in\mathcal{P}^{\mathrm{u}}\right),\;\mathbf{u}^{\star T}\left(\tilde{\mathbf{f}}_{\mathbf{int}}\left((\mathbf{u}(\tau))_{\tau\in[0,t]}\right)+\tilde{\mathbf{f}}_{\mathrm{ext}}(t)\right)=0$$
$$\text{with}\quad\begin{cases}\tilde{\mathbf{f}}_{\mathbf{int}}\left((\mathbf{u}(\tau))_{\tau\in[0,t]}\right)=-\displaystyle\sum_{b\in\mathcal{B}}\tilde{\mathbf{A}}^{(b)}\int_{\Omega^{(b)}}\frac{\partial\mathbf{\Lambda}^{(b)^T}}{\partial s^{(b)}}\,\mathbf{n}^{(b)}N^{(b)}\left((\mathbf{u}(\tau))_{\tau\in[0,t]}\right)\,ds^{(b)}\\ \tilde{\mathbf{f}}_{\mathrm{ext}}(t)=\displaystyle\sum_{b\in\mathcal{B}}\tilde{\mathbf{A}}^{(b)}\int_{\Omega^{(b)}}\mathbf{\Lambda}^{(b)^T}\mathbf{n}^{(b)}\,\mathbf{n}^{(b)^T}\mathbf{f}(t)\,ds^{(b)}+\displaystyle\sum_{P\in\mathcal{P}^{\mathrm{F}}}\hat{\mathbf{A}}^{(P)}\mathbf{N}_{\mathrm{d}|P}(t)\,,\end{cases} \tag{4.47}$$

where $\hat{\mathbf{A}}^{(P)}$ is an assembly operator defined such that at any node $P\in\mathcal{P}$, we have $\begin{pmatrix}\mathbf{u}_{|P}\cdot\mathbf{e}_x & \mathbf{u}_{|P}\cdot\mathbf{e}_y\end{pmatrix}^T=\hat{\mathbf{A}}^{(P)^T}\mathbf{u}$. Variational principle (4.47) needs to be complemented by the Dirichlet boundary conditions (4.37), and the resulting problem can be parametrised and discretised in time to obtain the "truth" problem (4.2).

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 $\theta(\equiv\boldsymbol{\mu})\in\mathbb{R}$ 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 4.2. 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.

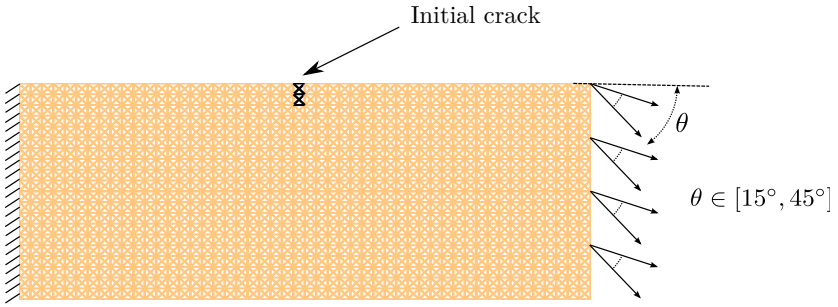


Figure 4.2: Definition of the nonlinear lattice problem used for the numerical experiments of this paper. The loss of stiffness of each bar while increasing local strain is described by a damage model. The direction of the prescribed displacement on the right-hand edge of the rectangular lattice structure 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 4.3 illustrate the behaviour of the reduced order modelling approach for $n_\mu = 2$. A normalised truncation error ν_{snap} of the snapshot POD as given in equation (4.25) arbitrarily set to 10^{-2} for example (see figure 4.4) would lead to the definition of a reduced space of dimension 3.

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 4.4). In fact, the solution to parametric problems involving the evolution of topological changes cannot, in general, be obtained

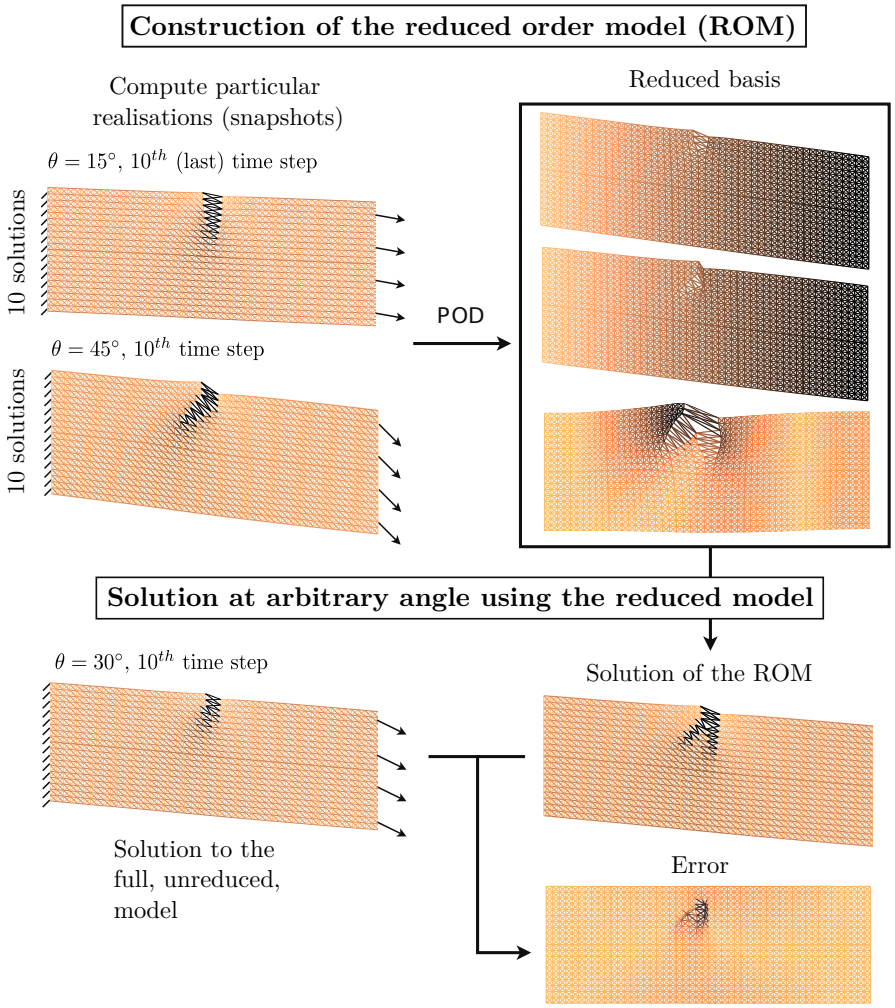


Figure 4.3: Schematic representation of the Snapshot POD model order reduction technique for the proposed parametrised problem of fracture. The "truth" time evolution of the problem is computed "offline" for a certain number of values of the parameter. A reduced space is generated by performing a spectral 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.

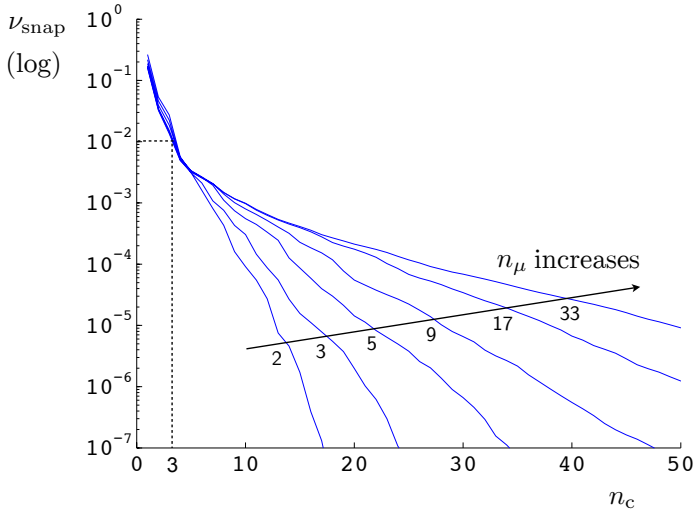


Figure 4.4: Convergence of the normalised POD error indicator ν_{snap} as a function of the order of truncation, for increasing size of the number of parameter values used to build the snapshot. The lack of correlation due to the crack propagation introduces a local error of projection, which appears here as a decrease in the convergence rate of the spectral decomposition (i.e. a decrease in the "slope" of the spectrum curve) 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.

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 [69]). One systematic way to circumvent the problem would be to enrich the snapshot "online" [4, 90], 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 4.3.3.

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 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.

4.3 Partitioned model order reduction approach

4.3.1 Principle of the primal Schur-based domain decomposition method

Schur-based non-overlapping domain decomposition methods (see a review in [91]) 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) [77, 78, 79])) 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 $\mu \in \mathcal{P}^{\mathrm{s}}$ (*i.e.:* the snapshot). Domain Ω is split into non-overlapping subdomains $(\Omega^{(e)})_{e\in[\![1,n_{\mathrm{e}}]\!]}$ such that $\bigcup_{e\in[\![1,n_{\mathrm{e}}]\!]} \Omega^{(e)} = \Omega$, as illustrated in Figure 4.5. Each bar of the lattice structure belongs to one and only one

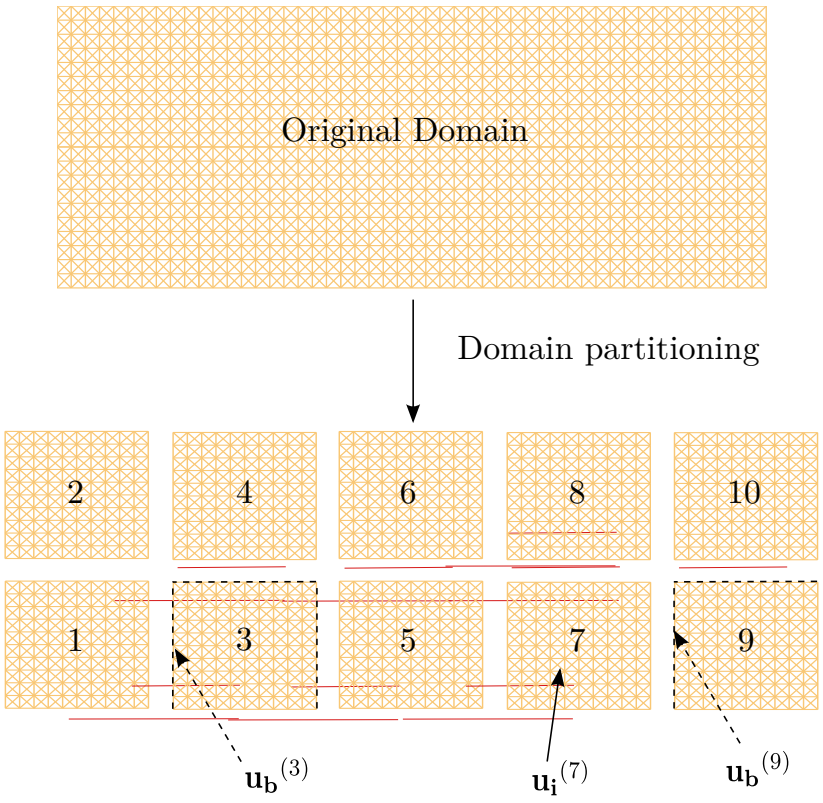


Figure 4.5: Subdivision of the domain of interest into 10 non-overlapping subdomains. $\boldsymbol{\Delta}\mathbf{u}_{\mathrm{i}}^{(e)}$ is the restriction of a vector $\boldsymbol{\Delta}\mathbf{u}^{(e)}$ of nodal values of subdomain e to the internal degrees of freedom of the subdomain, while $\boldsymbol{\Delta}\mathbf{u}_{\mathrm{b}}^{(e)}$ corresponds to the interface nodal values. The superscript between brackets indicates the number of the subdomain.

subdomain. Nodes that are shared by two adjacent subdomains are interface nodes. We later refer to the set of subdomain indexes $[\![1, n_{\mathrm{e}}]\!]$ as $\mathcal{E}$. Let $\mathbf{u}^{(e)}(t;\boldsymbol{\mu}) \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}}$ be the vector of nodal displacements of $\Omega^{(e)}$, which is looked for at an arbitrary point $(t, \boldsymbol{\mu}) \in \tilde{\mathcal{P}}$ 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:

$$\mathbf{f}_{\mathbf{int}}^{(e)}\left(\mathbf{u}^{(e)}(t;\boldsymbol{\mu}), \left(\mathbf{u}^{(e)}(\tau;\boldsymbol{\mu})\right)_{\tau\in\mathcal{T}^h,\,\tau<t}; \boldsymbol{\mu}\right) + \mathbf{f}_{\mathbf{ext}}^{(e)}(t;\boldsymbol{\mu}) = {\mathbf{T}^{(e)}}^T \boldsymbol{\lambda}^{(e)}\,, \tag{4.48}$$

with $\boldsymbol{\lambda}^{(e)} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}$ a vector of reaction forces from adjacent subdomains and $\mathbf{T}^{(e)} \in \{0,1\}^{n_{\mathrm{b}}^{(e)}} \times \{0,1\}^{n_{\mathrm{u}}^{(e)}}$ a trace operator which extracts the entries of vector of local nodal values corresponding to the interface nodes (*i.e.* an output vector $\mathbf{x}_{\mathbf{b}}^{(e)} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}$ defined

by $\mathbf{x}_{\mathbf{b}}^{(e)} = \mathbf{T}^{(e)}\mathbf{x}^{(e)}$, with $\mathbf{x}^{(e)} \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}}$ an arbitrary vector of local nodal values). The reaction forces must satisfy the following global interface equilibrium property:

$$\sum_{e\in\mathcal{E}} \mathbf{A}^{(e)}\boldsymbol{\lambda}^{(e)} = \mathbf{0} \tag{4.49}$$

where $\{\mathbf{A}^{(e)} \in \{0,1\}^{n_{\mathrm{b}}} \times \{0,1\}^{n_{\mathrm{b}}^{(e)}} \,|\, e \in \mathcal{E}\}$ 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

$$\sum_{e\in\mathcal{E}} \bar{\mathbf{A}}^{(e)}\mathbf{T}^{(e)}\mathbf{u}^{(e)} = \mathbf{0} \tag{4.50}$$

where $\{\bar{\mathbf{A}}^{(e)} \in \{0,-1,1\}^{\bar{n}_{\mathrm{b}}} \times \{0,-1,1\}^{n_{\mathrm{b}}^{(e)}} \,|\, e \in \mathcal{E}\}$, are signed boolean operators, with $\bar{n}_{\mathrm{b}}$ the number of independent interface kinematic constraints. (see [91] 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 (4.48) at iteration $i+1$ of the Newton algorithm. We look for iterates $\{(\mathbf{u}^{(e),i+1}, \boldsymbol{\lambda}^{(e),i+1}) \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}} \times \mathbb{R}^{n_{\mathrm{b}}^{(e)}} \,|\, e \in \mathcal{E}\}$ of the local displacements and reaction forces satisfying both the local linearised systems

$$\mathbf{K}^{(e),i}\boldsymbol{\Delta}\mathbf{u}^{(e),i+1} = -\mathbf{r}^{(e)} + {\mathbf{T}^{(e)}}^{T}\boldsymbol{\lambda}^{(e),i+1} \quad ,\forall\, e \in \mathcal{E}\,, \tag{4.51}$$

and the global interface conditions (4.49) and (4.50). In the previous equation, the local tangent stiffness is $\mathbf{K}^{(e),i} \overset{\text{def}}{=} \left.\frac{\partial \mathbf{f}_{\mathbf{int}}^{(e)}(\mathbf{u}^{(e)})}{\partial \mathbf{u}^{(e)}}\right|_{\mathbf{u}^{(e)}=\mathbf{u}^{(e),i}}$, the residual vector is $\mathbf{r}^{(e),i} \overset{\text{def}}{=} \mathbf{f}_{\mathbf{int}}^{(e)}(\mathbf{u}^{(e),i}) + \mathbf{f}_{\mathbf{ext}}^{(e)}$, and the increment of displacement is defined by $\boldsymbol{\Delta}\mathbf{u}^{(e),i+1} = \mathbf{u}^{(e),i+1} - \mathbf{u}^{(e),i}$. In the following, we will drop superscripts i and $i+1$.

If we introduce the local operator $\mathbf{E}^{(e)} \in \{0,1\}^{n_{\mathrm{i}}^{(e)}} \times \{0,1\}^{n_{\mathrm{u}}^{(e)}}$ ($n_{\mathrm{i}}^{(e)} \overset{\text{def}}{=} n_{\mathrm{u}}^{(e)} - n_{\mathrm{b}}^{(e)}$ is the number of interior degrees of freedom of e) such that the output vector $\mathbf{x}_{\mathbf{i}}^{(e)} = \mathbf{E}^{(e)}\mathbf{x}^{(e)}$, with $\mathbf{x}^{(e)} \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}}$ arbitrary, is the restriction of $\mathbf{x}^{(e)}$ to the interior nodes of subdomain $\Omega^{(e)}$, for any $e \in \mathcal{E}$, we can recast the local systems (4.51) as follows:

$$\begin{bmatrix} \mathbf{K}_{\mathbf{ii}}^{(e)} & \mathbf{K}_{\mathbf{ib}}^{(e)} \\ \mathbf{K}_{\mathbf{bi}}^{(e)} & \mathbf{K}_{\mathbf{bb}}^{(e)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{u}_{\mathbf{i}}^{(e)} \\ \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_{\mathbf{i}}^{(e)} \\ -\mathbf{r}_{\mathbf{b}}^{(e)} + \boldsymbol{\lambda}^{(e)} \end{bmatrix} \quad ,\forall\, e \in \mathcal{E}\,, \tag{4.52}$$

where $\boldsymbol{\Delta}\mathbf{u}_{\mathbf{i}}^{(e)} \overset{\text{def}}{=} \mathbf{E}^{(e)}\boldsymbol{\Delta}\mathbf{u}^{(e)}$, $\boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)} \overset{\text{def}}{=} \mathbf{T}^{(e)}\boldsymbol{\Delta}\mathbf{u}^{(e)}$, $\mathbf{r}_{\mathbf{i}}^{(e)} \overset{\text{def}}{=} \mathbf{E}^{(e)}\mathbf{r}^{(e)}$, $\mathbf{r}_{\mathbf{b}}^{(e)} \overset{\text{def}}{=} \mathbf{T}^{(e)}\mathbf{r}^{(e)}$, $\mathbf{K}_{\mathbf{ii}}^{(e)} \overset{\text{def}}{=} \mathbf{E}^{(e)}\mathbf{K}^{(e)}\mathbf{E}^{(e)T}$, $\mathbf{K}_{\mathbf{ib}}^{(e)} \overset{\text{def}}{=} \mathbf{E}^{(e)}\mathbf{K}^{(e)}\mathbf{T}^{(e)T}$, $\mathbf{K}_{\mathbf{bi}}^{(e)} \overset{\text{def}}{=} \mathbf{T}^{(e)}\mathbf{K}^{(e)}\mathbf{E}^{(e)T}$ and $\mathbf{K}_{\mathbf{bb}}^{(e)} \overset{\text{def}}{=} \mathbf{T}^{(e)}\mathbf{K}^{(e)}\mathbf{T}^{(e)T}$. The interior degrees of freedom $\boldsymbol{\Delta}\mathbf{u}_{\mathbf{i}}^{(e)}$ are eliminated from local systems (4.52) by static condensation, which is obtained by writing

$$\boldsymbol{\Delta}\mathbf{u}_{\mathbf{i}}^{(e)} = \mathbf{K}_{\mathbf{ii}}^{(e)^{-1}} \left(-\mathbf{r}_{\mathbf{i}}^{(e)} - \mathbf{K}_{\mathbf{ib}}^{(e)} \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)} \right) , \tag{4.53}$$

where $\mathbf{K}_{\mathbf{ii}}^{(e)}$ is assumed to be invertible. The condensed local problem is obtained by substitution of expression (4.53) in the second line of (4.52):

$$\mathbf{S}_{\mathbf{p}}^{(e)} \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)} = \mathbf{f}_{\mathbf{c}}^{(e)} + \boldsymbol{\lambda}^{(e)} , \tag{4.54}$$

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

We now apply the primal domain decomposition methodology by enforcing the interface kinematic continuity (4.50) in a strong sense, which is done by writing that the local trace of the unknown displacement vectors $\{\mathbf{u}^{(e)} \,|\, e \in \mathcal{E}\}$ are obtained by extraction from a global interface vector $\mathbf{u}_{\mathbf{b}} \in \mathbb{R}^{n_{\mathrm{b}}}$

$$\boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)} \overset{\text{def}}{=} \mathbf{T}^{(e)}\boldsymbol{\Delta}\mathbf{u}^{(e)} = \mathbf{A}^{(e)^T} \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}} \quad , \forall\, e \in \mathcal{E} , \tag{4.55}$$

which implies the fulfilment of (4.50) 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 (4.54)) by assembly operators $\{\mathbf{A}^{(e)} \,|\, e \in \mathcal{E}\}$ and summing up over all subdomains, which reads

$$\sum_{e\in\mathcal{E}} \mathbf{A}^{(e)} \mathbf{S}_{\mathbf{p}}^{(e)} \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)} = \sum_{e\in\mathcal{E}} \mathbf{A}^{(e)} \mathbf{f}_{\mathbf{c}}^{(e)} + \underbrace{\sum_{e\in\mathcal{E}} \mathbf{A}^{(e)} \boldsymbol{\lambda}^{(e)}}_{= \mathbf{0} \quad \text{from} \quad (4.49)} \tag{4.56}$$

$$\Longleftrightarrow \sum_{e\in\mathcal{E}} \left(\mathbf{A}^{(e)} \mathbf{S}_{\mathbf{p}}^{(e)} \mathbf{A}^{(e)^T} \right) \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}} = \sum_{e\in\mathcal{E}} \mathbf{A}^{(e)} \mathbf{f}_{\mathbf{c}}^{(e)} . \tag{4.57}$$

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

$$\mathbf{S_p \Delta u_b} = \mathbf{f_c} \quad \text{with} \quad \begin{cases} \mathbf{S_p} = \sum_{e \in \mathcal{E}} \mathbf{A}^{(e)} \mathbf{S}_\mathbf{p}^{(e)} {\mathbf{A}^{(e)}}^T \\ \mathbf{f_c} = \sum_{e \in \mathcal{E}} \mathbf{A}^{(e)} \mathbf{f}_\mathbf{c}^{(e)} \,. \end{cases} \tag{4.58}$$

Interface problem (4.58) can be solved iteratively in parallel using a Krylov-subspace method such as the conjugate gradient in a symmetric case or GMRes [92](or BiCGStab [93]) 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:

$$\forall\, \mathbf{x_b} \in \mathbb{R}^{n_\mathrm{b}}, \qquad \mathbf{S_p x_b} = \sum_{e \in \mathcal{E}} \mathbf{A}^{(e)} \mathbf{S}_\mathbf{p}^{(e)} \underbrace{{\mathbf{A}^{(e)}}^T \mathbf{x_b}}_{=\mathbf{x}_\mathbf{b}^{(e)}} \,. \tag{4.59}$$

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.

$$\forall\, \mathbf{x_b} \in \mathbb{R}^{n_\mathrm{b}}, \qquad {\mathbf{x_b}}^T \mathbf{x_b} = \sum_{e \in \mathcal{E}} {\mathbf{x}_\mathbf{b}^{(e)}}^T \mathbf{D}^{(e)} \mathbf{x}_\mathbf{b}^{(e)} \,, \tag{4.60}$$

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

4.3.2 Formulation of reduced order modelling in the domain decomposition framework

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 $\{\mathbf{u}(t;\boldsymbol{\mu}) \,|\, (t,\boldsymbol{\mu}) \in \tilde{\mathcal{P}}^\mathrm{s}\}$ is available. This snapshot has been computed by making use of the domain decomposition preconditioner described previously. Local POD spatial bases $\left(\mathbf{C}_{\mathbf{i},i}^{(e)}\right)_{i \in [\![1, n_\mathrm{c}^{(e)}]\!]}$ 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 4.2. Accordingly, the normalised truncation error of the local snapshot POD transforms are defined as follows:

$$\left(\nu_{\mathrm{snap}}^{(e)}\right)^2 = \frac{\displaystyle\sum_{\boldsymbol{\mu}\in\mathcal{P}^{\mathrm{s}}}\sum_{t\in\mathcal{T}^{\mathrm{h}}}\left\|\mathbf{u}_{\mathbf{i}}^{(e)}(t;\boldsymbol{\mu}) - \sum_{j=1}^{n_{\mathrm{c}}^{(e)}}\left(\mathbf{C}_{\mathbf{i},j}^{(e)^T}\mathbf{u}_{\mathbf{i}}^{(e)}(t;\boldsymbol{\mu})\right)\mathbf{C}_{\mathbf{i},j}^{(e)}\right\|_2^2}{\displaystyle\sum_{t\in\mathcal{T}^{\mathrm{h}}}\sum_{\boldsymbol{\mu}\in\mathcal{P}^{\mathrm{s}}}\|\mathbf{u}_{\mathbf{i}}^{(e)}(t;\boldsymbol{\mu})\|_2^2}, \quad \forall e \in \mathcal{E}\,, \tag{4.61}$$

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

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 (4.48), introduce the a local reduced basis approximation and linearise the result.

The kinematic interior approximation for the linearised problem reads:

$$\begin{bmatrix}\boldsymbol{\Delta}\mathbf{u}_{\mathbf{i}}^{(e)}\\ \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)}\end{bmatrix} = \begin{bmatrix}\mathbf{C}_{\mathbf{i}}^{(e)}\boldsymbol{\Delta}\boldsymbol{\alpha}_{\mathbf{i}}^{(e)}\\ \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)}\end{bmatrix}, \quad \forall e \in \mathcal{E}\,, \tag{4.62}$$

where $\boldsymbol{\Delta}\boldsymbol{\alpha}_{\mathbf{i}}^{(e)}$ is a vector of local reduced state variables. Therefore, the local linearised system of equation (4.52) corresponding to an arbitrary subdomain $e \in \mathcal{E}$ now reads

$$\begin{bmatrix}\mathbf{K}_{\mathbf{ii}}^{(e)} & \mathbf{K}_{\mathbf{ib}}^{(e)}\\ \mathbf{K}_{\mathbf{bi}}^{(e)} & \mathbf{K}_{\mathbf{bb}}^{(e)}\end{bmatrix}\begin{bmatrix}\mathbf{C}_{\mathbf{i}}^{(e)}\boldsymbol{\Delta}\boldsymbol{\alpha}_{\mathbf{i}}^{(e)}\\ \boldsymbol{\Delta}\mathbf{u}_{\mathbf{b}}^{(e)}\end{bmatrix} = \begin{bmatrix}-\mathbf{r}_{\mathbf{i}}^{(e)}\\ -\mathbf{r}_{\mathbf{b}}^{(e)} + \boldsymbol{\lambda}^{(e)}\end{bmatrix}. \tag{4.63}$$

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 4.2.1, we perform a Galerkin projection: the residual of local system (4.63) is required to be orthogonal to the local reduced space, which

reads

$$\begin{bmatrix} \mathbf{C}_\mathbf{i}^{(e)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d,\mathbb{R}^{n_\mathbf{i}^{(e)}}} \end{bmatrix}^T \left(\begin{bmatrix} -\mathbf{r}_\mathbf{i}^{(e)} \\ -\mathbf{r}_\mathbf{b}^{(e)} + \boldsymbol{\lambda}^{(e)} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_\mathbf{ii}^{(e)} & \mathbf{K}_\mathbf{ib}^{(e)} \\ \mathbf{K}_\mathbf{bi}^{(e)} & \mathbf{K}_\mathbf{bb}^{(e)} \end{bmatrix} \begin{bmatrix} \mathbf{C}_\mathbf{i}^{(e)} \boldsymbol{\Delta\alpha}_\mathbf{i}^{(e)} \\ \boldsymbol{\Delta u}_\mathbf{b}^{(e)} \end{bmatrix} \right) = \mathbf{0}\,. \tag{4.64}$$

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

$$\left(\mathbf{f}_\mathbf{r}^{(e)} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\lambda}^{(e)} \end{bmatrix} \right) - \mathbf{K}_\mathbf{r}^{(e)} \begin{bmatrix} \boldsymbol{\Delta\alpha}_\mathbf{i}^{(e)} \\ \boldsymbol{\Delta u}_\mathbf{b}^{(e)} \end{bmatrix} = \mathbf{0} \quad \text{where} \quad \begin{cases} \mathbf{K}_\mathbf{r}^{(e)} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{K}_\mathbf{ii}^{(e)} \mathbf{C}_\mathbf{i}^{(e)} & \mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{K}_\mathbf{ib}^{(e)} \\ \mathbf{K}_\mathbf{bi}^{(e)} \mathbf{C}_\mathbf{i}^{(e)} & \mathbf{K}_\mathbf{bb}^{(e)} \end{bmatrix} \\ \mathbf{f}_\mathbf{r}^{(e)} \overset{\text{def}}{=} \begin{bmatrix} -\mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{r}_\mathbf{i}^{(e)} \\ -\mathbf{r}_\mathbf{b}^{(e)} \end{bmatrix}, \end{cases} \tag{4.65}$$

Condensed interface problem

Similarly as described in section 4.3.1, local systems (4.65) are condensed on the interface degrees of freedom, and are formally assembled. To do so, the reduced state variables $\boldsymbol{\Delta\alpha}_\mathbf{i}^{(e)}$ are eliminated using the identity

$$\boldsymbol{\Delta\alpha}_\mathbf{i}^{(e)} = \mathbf{K}_\mathbf{ii,r}^{(e)^{-1}} \left(-\mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{r}_\mathbf{i}^{(e)} - \mathbf{K}_\mathbf{ib,r}^{(e)} \boldsymbol{\Delta u}_\mathbf{b}^{(e)} \right), \tag{4.66}$$

where $\mathbf{K}_\mathbf{ii,r} \overset{\text{def}}{=} \mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{K}_\mathbf{ii}^{(e)} \mathbf{C}_\mathbf{i}^{(e)}$ is assumed to be invertible and $\mathbf{K}_\mathbf{ib,r} \overset{\text{def}}{=} \mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{K}_\mathbf{ib}^{(e)}$. 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:

$$\text{Find } \boldsymbol{\Delta u}_\mathbf{b} \in \mathbb{R}^{n_\mathrm{b}} \text{ such that } \quad \mathbf{S}_\mathbf{p,r}\, \boldsymbol{\Delta u}_\mathbf{b} = \mathbf{f}_\mathbf{c,r} \quad \text{with} \quad \begin{cases} \mathbf{S}_\mathbf{p,r} = \sum_{e \in \mathcal{E}} \mathbf{A}^{(e)} \mathbf{S}_\mathbf{p,r}^{(e)}\, \mathbf{A}^{(e)^T} \\ \mathbf{f}_\mathbf{c,r} = \sum_{e \in \mathcal{E}} \mathbf{A}^{(e)} \mathbf{f}_\mathbf{c,r}^{(e)}, \end{cases} \tag{4.67}$$

with the expression of the local condensed operators $\mathbf{S}_\mathbf{p,r}^{(e)} \overset{\text{def}}{=} \mathbf{K}_\mathbf{bb}^{(e)} - \mathbf{K}_\mathbf{bi,r}^{(e)}\, \mathbf{K}_\mathbf{ii,r}^{(e)^{-1}} \mathbf{K}_\mathbf{ib,r}^{(e)}$, the local condensed forces $\mathbf{f}_\mathbf{c,r}^{(e)} \overset{\text{def}}{=} -\mathbf{r}_\mathbf{b}^{(e)} - \mathbf{K}_\mathbf{bi,r}^{(e)}\, \mathbf{K}_\mathbf{ii,r}^{(e)^{-1}} (-\mathbf{C}_\mathbf{i}^{(e)^T} \mathbf{r}_\mathbf{i}^{(e)})$ and $\mathbf{K}_\mathbf{bi,r} \overset{\text{def}}{=} \mathbf{K}_\mathbf{bi}^{(e)} \mathbf{C}_\mathbf{i}^{(e)}$, for any $e \in \mathcal{E}$. Problem (4.67) can be solved in parallel (if the snapshot data is distributed in memory) using a Krylov algorithm, as described in section 4.3.1.

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 4.2.4). This particular issue will be addressed in section 4.3.3. 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 $\boldsymbol{\Delta u_b} \in \mathbb{R}^{n_b}$ satisfying

$$\mathbf{S_{p,hr}}\,\boldsymbol{\Delta u_b} = \mathbf{f_{c,hr}} \text{ with } \begin{cases} \mathbf{S_{p,hr}} = \displaystyle\sum_{e\in\mathcal{E}^{\text{red}}} \mathbf{A}^{(e)}\mathbf{S}^{(e)}_{\mathbf{p,r}}\,\mathbf{A}^{(e)^T} + \sum_{e\in\mathcal{E}^{\text{nred}}} \mathbf{A}^{(e)}\mathbf{S}^{(e)}_{\mathbf{p}}\,\mathbf{A}^{(e)^T} \\ \mathbf{f_{c,hr}} = \displaystyle\sum_{e\in\mathcal{E}^{\text{red}}} \mathbf{A}^{(e)}\mathbf{f}^{(e)}_{\mathbf{c,r}} + \sum_{e\in\mathcal{E}^{\text{nred}}} \mathbf{A}^{(e)}\mathbf{f}^{(e)}_{\mathbf{c}} \end{cases} . \tag{4.68}$$

4.3.3 Local error estimation by Cross-Validation

Principle

The partitioned projection approach described in section 4.3.2 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 [94, 55, 90], but most of the studies on the POD,

or the Principal Component Analysis in the statistics community (a recent review is provided in [95]) 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 [96] 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 $\mathcal{P}^{\mathrm{s}}$. 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 (4.22). Therefore, the associated error estimate (4.24) is expected to underestimate the error of projection associated to a hierarchically enriched snapshot, and in the limit, to underestimate the integral form (4.18) 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 (4.61) 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 $e \in \mathcal{E}$:

$$\left(\tilde{\nu}_{\mathrm{snap}}^{(e)}\right)^2 = \frac{\displaystyle\sum_{\boldsymbol{\mu}\in\mathcal{P}^{\mathrm{s}}} \sum_{t\in\mathcal{T}^{\mathrm{h}}} \left\| \mathbf{u}_{\mathbf{i}}(t;\boldsymbol{\mu}) - \sum_{j=1}^{n_{\mathrm{c}}^{(e)}} \left(\tilde{\mathbf{C}}_{\mathbf{i},j}^{(e),(\boldsymbol{\mu})\,T} \mathbf{u}_{\mathbf{i}}(t;\boldsymbol{\mu}) \right) \tilde{\mathbf{C}}_{\mathbf{i},j}^{(e),(\boldsymbol{\mu})} \right\|_2^2}{\displaystyle\sum_{t\in\mathcal{T}^{\mathrm{h}}} \sum_{\boldsymbol{\mu}\in\mathcal{P}^{\mathrm{s}}} \|\mathbf{u}_{\mathbf{i}}(t;\boldsymbol{\mu})\|_2^2}, \tag{4.69}$$

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

$$\tilde{\mathbf{H}}^{(\boldsymbol{\mu})} = \sum_{\boldsymbol{\mu}^\star \in (\mathcal{P}^{\mathrm{s}} \setminus \boldsymbol{\mu})} \sum_{t \in \mathcal{T}^{\mathrm{h}}} \mathbf{u}(t;\boldsymbol{\mu}^\star)\, \mathbf{u}(t;\boldsymbol{\mu}^\star)^T . \tag{4.70}$$

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 [76, 55] 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.

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 4.6, 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 $\tilde{\nu}_{\mathrm{snap}}^{(e)} \leq 10^{-3}$, 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 recent

investigations about this particular effect [53]. 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.

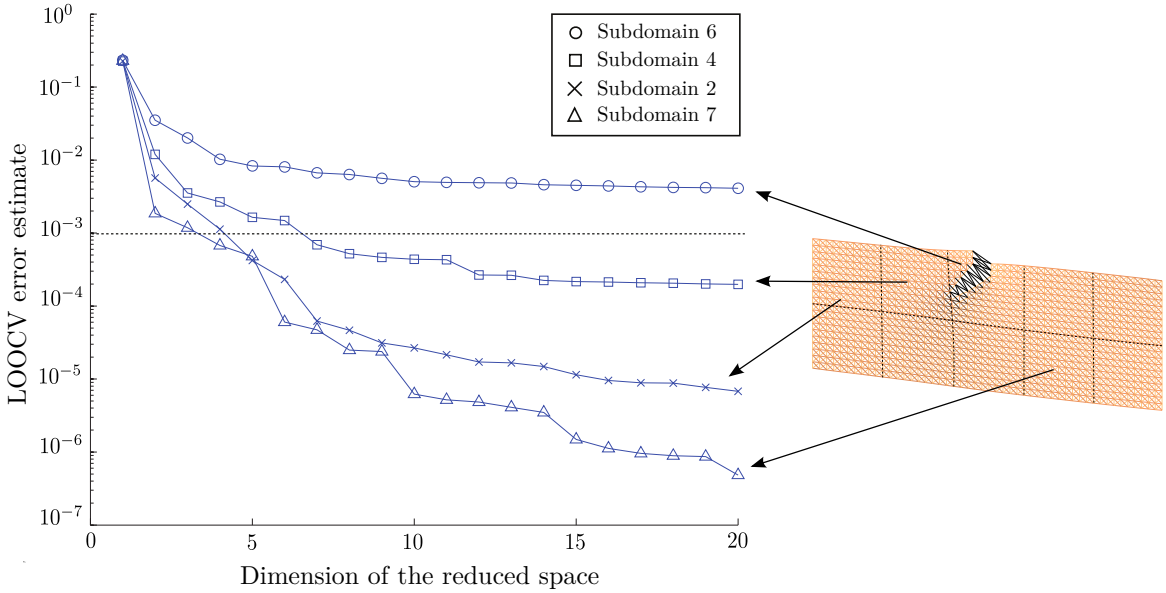


Figure 4.6: Cross-validation error estimate as a function of the order of the POD transforms for 4 of the 10 subdomains. The snapshot comprises 5 instances of the solution to the parametric problem of evolution. Subdomains are numbered as in Figure 4.5.

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.

4.4 System approximation in the partitioned model order reduction approach

4.4.1 Local "gappy" approximations

We propose here to extend the concept of "system approximation" to the partitioned model order reduction introduced in section 4.3. As mentioned previously, we choose to apply a tailored version of the "gappy" reconstruction technique presented in different contexts in [60, 23, 5]. It is important to realise that the gappy technique approximates the Galerkin projection framework described in section 4.3. 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 [5].

The starting point of the gappy technique is to compute local "static" reduced bases $\left\{ \mathbf{D}_{\mathbf{i}}^{(e)} \in \mathbb{R}^{n_{\mathrm{u}}^{(e)}} \times \mathbb{R}^{n_{\mathrm{d}}^{(e)}} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$ to approximate the vectors of internal forces $\left\{ \mathbf{f}_{\mathbf{int},\mathbf{i}}^{(e)} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\} \stackrel{\mathrm{def}}{=} \left\{ \mathbf{E}^{(e)} \mathbf{f}_{\mathbf{int}}^{(e)} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$, as detailed previously in the non-partitioned case (see section 4.2). Once the local bases are computed, the approximation reads

$$\begin{aligned} &\forall\, e \in \mathcal{E}^{\mathrm{red}}, \forall\, t \in \mathcal{T}^{\mathrm{h}}, \forall\, \boldsymbol{\alpha}^{(e)\star} \in \mathbb{R}^{n_{\mathrm{c}}^{(e)}}, \forall\, \mathbf{u}_{\mathbf{b}}^{(e)\star} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}, \\ &\quad \mathbf{f}_{\mathbf{int},\mathbf{i}}^{(e)} \left(\begin{pmatrix} \mathbf{C}_{\mathbf{i}}^{(e)} \boldsymbol{\alpha}^{(e)\star} \\ \mathbf{u}_{\mathbf{b}}^{(e)\star} \end{pmatrix}, \left(\mathbf{u}^{(e)}(\tau;\boldsymbol{\mu}) \right)_{\tau \in \mathcal{T}^{\mathrm{h}}, \tau < t}; \boldsymbol{\mu} \right) \approx \\ &\qquad \mathbf{D}_{\mathbf{i}}^{(e)} \boldsymbol{\beta}_{\mathbf{i}}^{(e)} \left(\begin{pmatrix} \boldsymbol{\alpha}^{(e)\star} \\ \mathbf{u}_{\mathbf{b}}^{(e)\star} \end{pmatrix}, \left(\mathbf{u}^{(e)}(\tau;\boldsymbol{\mu}) \right)_{\tau \in \mathcal{T}^{\mathrm{h}}, \tau < t}, \boldsymbol{\mu} \right), \end{aligned} \tag{4.71}$$

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 $\left\{ \boldsymbol{\beta}_{\mathbf{i}}^{(e)} \in \mathbb{R}^{n_{\mathrm{d}}^{(e)}} \,|\, e \in \mathcal{E}^{\mathrm{red}} \right\}$ 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

$$\begin{aligned} &\forall\, e \in \mathcal{E}^{\mathrm{red}}, \forall\, \boldsymbol{\alpha}^{(e)\star} \in \mathbb{R}^{n_{\mathrm{c}}^{(e)}}, \forall\, \mathbf{u}_{\mathbf{b}}^{(e)\star} \in \mathbb{R}^{n_{\mathrm{b}}^{(e)}}, \\ &\quad \mathbf{f}_{\mathbf{int},\mathbf{i}}^{(e)} \left(\begin{pmatrix} \mathbf{C}_{\mathbf{i}}^{(e)} \boldsymbol{\alpha}^{(e)\star} \\ \mathbf{u}_{\mathbf{b}}^{(e)\star} \end{pmatrix} \right) \approx \mathbf{D}_{\mathbf{i}}^{(e)} \left(\mathbf{D}_{\mathbf{i}}^{(e)T} \mathbf{P}_{\mathbf{i}}^{(e)} \mathbf{D}_{\mathbf{i}}^{(e)} \right)^{-1} \mathbf{D}_{\mathbf{i}}^{(e)T} \mathbf{P}_{\mathbf{i}}^{(e)} \mathbf{f}_{\mathbf{int},\mathbf{i}}^{(e)} \left(\begin{pmatrix} \mathbf{C}_{\mathbf{i}}^{(e)} \boldsymbol{\alpha}^{(e)\star} \\ \mathbf{u}_{\mathbf{b}}^{(e)\star} \end{pmatrix} \right), \end{aligned} \tag{4.72}$$

The local boolean operator $\mathbf{P}_{\mathbf{i}}^{(e)}$ operating on the subdomain $e \in \mathcal{E}^{\text{red}}$ 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 $\mathbf{G}_{\mathbf{i}}^{(e)} = \mathbf{D}_{\mathbf{i}}^{(e)} \left(\mathbf{D}_{\mathbf{i}}^{(e)^T} \mathbf{P}_{\mathbf{i}}^{(e)} \mathbf{D}_{\mathbf{i}}^{(e)} \right)^{-1} \mathbf{D}_{\mathbf{i}}^{(e)^T} \mathbf{P}_{\mathbf{i}}^{(e)}$.

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 (4.72), one gets a modified expression of the local tangent systems (compare equation (4.63)) at Newton iteration $i+1$ of an arbitrary time-parameter point of $\tilde{\mathcal{P}}$, for any subdomain $e \in \mathcal{E}$:

$$\begin{bmatrix} \mathbf{G}_{\mathbf{i}}^{(e)} \mathbf{K}_{\mathbf{ii}}^{(e)} & \mathbf{G}_{\mathbf{i}}^{(e)} \mathbf{K}_{\mathbf{ib}}^{(e)} \\ \mathbf{K}_{\mathbf{bi}}^{(e)} & \mathbf{K}_{\mathbf{bb}}^{(e)} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{\mathbf{i}}^{(e)} \boldsymbol{\Delta\alpha}_{\mathbf{i}}^{(e)} \\ \boldsymbol{\Delta} \mathbf{U}_{\mathbf{b}}^{(e)} \end{bmatrix} = \begin{bmatrix} -\mathbf{G}_{\mathbf{i}}^{(e)} \mathbf{f}_{\mathbf{int,i}}^{(e)} \left(\mathbf{u}^{(e),i} \right) - \mathbf{f}_{\mathbf{ext,i}}^{(e)} \\ -\mathbf{R}_{\mathbf{b}}^{(e)} + \boldsymbol{\lambda}^{(e)} \end{bmatrix}, \tag{4.73}$$

with $\mathbf{f}_{\mathbf{ext,i}}^{(e)} \overset{\text{def}}{=} \mathbf{E}^{(e)} \mathbf{f}_{\mathbf{ext}}^{(e)}$

As mentioned in section 4.2, 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 $e \in \mathcal{E}^{\text{red}}$:

$$\left(\mathbf{f}_{\mathbf{r,sa}}^{(e)} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\lambda}^{(e)} \end{bmatrix} \right) - \mathbf{K}_{\mathbf{r,sa}}^{(e)} \begin{bmatrix} \boldsymbol{\Delta\alpha}_{\mathbf{i}}^{(e)} \\ \boldsymbol{\Delta} \mathbf{U}_{\mathbf{b}}^{(e)} \end{bmatrix} = \mathbf{0}, \tag{4.74}$$

with

$$\begin{cases} \mathbf{K}_{\mathbf{r,sa}}^{(e)} = \begin{bmatrix} \mathbf{C}_{\mathbf{i}}^{(e)^T} \mathbf{G}_{\mathbf{i}}^{(e)} \mathbf{K}_{\mathbf{ii}}^{(e)} \mathbf{C}_{\mathbf{i}}^{(e)} & \mathbf{C}_{\mathbf{i}}^{(e)^T} \mathbf{G}_{\mathbf{i}}^{(e)} \mathbf{K}_{\mathbf{ib}}^{(e)} \\ \mathbf{K}_{\mathbf{bi}}^{(e)} \mathbf{C}_{\mathbf{i}}^{(e)} & \mathbf{K}_{\mathbf{bb}}^{(e)} \end{bmatrix} \\ \mathbf{f}_{\mathbf{r,sa}}^{(e)} = \begin{bmatrix} -\mathbf{C}_{\mathbf{i}}^{(e)^T} \left(\mathbf{G}_{\mathbf{i}}^{(e)} \mathbf{f}_{\mathbf{int,i}}^{(e)} (\mathbf{u}^{(e),i}) + \mathbf{f}_{\mathbf{ext,i}}^{(e)} \right) \\ -\mathbf{R}_{\mathbf{b}}^{(e)} \end{bmatrix}. \end{cases}$$

A condensed linearised interface problem is finally obtained as follows. We look for

$\boldsymbol{\Delta}\mathbf{U_b} \in \mathbb{R}^{n_\mathrm{b}}$ satisfying

$$\mathbf{S_{p,r,sa}}\,\boldsymbol{\Delta}\mathbf{U_b} = \mathbf{f_{c,r,sa}} \quad \text{with} \begin{cases} \mathbf{S_{p,r,sa}} = \displaystyle\sum_{e\in\mathcal{E}^{\mathrm{red}}} \mathbf{A}^{(e)}\mathbf{S}^{(e)}_{\mathbf{p,r,sa}}\,\mathbf{A}^{(e)^T} + \sum_{e\in\mathcal{E}^{\mathrm{nred}}} \mathbf{A}^{(e)}\mathbf{S}^{(e)}_{\mathbf{p}}\,\mathbf{A}^{(e)^T} \\ \mathbf{f_{c,r,sa}} = \displaystyle\sum_{e\in\mathcal{E}^{\mathrm{red}}} \mathbf{A}^{(e)}\mathbf{f}^{(e)}_{\mathbf{c,r,sa}} + \sum_{e\in\mathcal{E}^{\mathrm{nred}}} \mathbf{A}^{(e)}\mathbf{f}^{(e)}_{\mathbf{c}}\,. \end{cases} \tag{4.75}$$

The method to obtain the expression of the modified primal Schur complement $\mathbf{S_{p,r,sa}}$ 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 (4.67)).

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 $\left\{\mathbf{P}^{(e)}_{\mathbf{i}} \,|\, e \in \mathcal{E}^{\mathrm{red}}\right\}$ 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 4.7. The way this reduced integration domain was obtained is detailed in the following.

4.4.2 Construction of the system approximation

Static POD bases

To generate the local bases $\left\{\mathbf{D}^{(e)}_{\mathbf{i}} \,|\, e \in \mathcal{E}^{\mathrm{red}}\right\}$, we develop a technique that is strongly inspired by the one proposed in [5]. Equation (4.71) 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

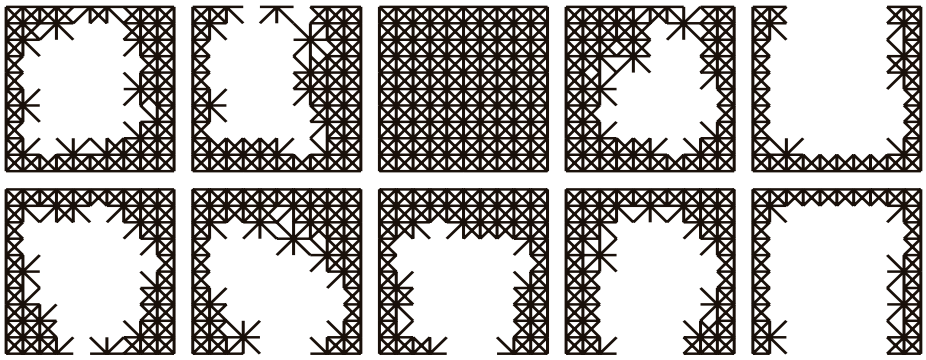


Figure 4.7: Example of a reduced integration domain. Subdomain 6 is not reduced. Therefore, all the associated elements belong to the integration domain. Since the interface between substructures is not reduced in the proposed primal version of the Schur-based partitioned model order reduction method, 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.

in section 4.3, 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 $e \in \mathcal{E}^{\text{red}}$, by the following set:

$$\mathcal{F}^{s,(e)} = \left\{ \mathbf{f}^{(e)}_{\mathbf{int},\mathbf{i}} \left(\begin{pmatrix} \mathbf{C}^{(e)}_{\mathbf{i}} \boldsymbol{\alpha}^{(e),i}(t,\boldsymbol{\mu}) \\ \mathbf{u}^{(e),i}_{\mathbf{b}}(t;\boldsymbol{\mu}) \end{pmatrix}, \left(\mathbf{u}^{(e)}(\tau;\boldsymbol{\mu})\right)_{\tau\in\mathcal{T}^{\text{h}},\tau<t} ; \boldsymbol{\mu} \right) \right\}_{\left| \boldsymbol{\mu}\in\mathcal{P}^{\text{s}}, t\in\mathcal{T}^{\text{h}}, i\in[\![1,n^{(t),(\boldsymbol{\mu})}_{\text{new}}]\!]\right.} . \tag{4.76}$$

In the previous expression, $n^{(t),(\boldsymbol{\mu})}_{\text{new}}$ denotes the number of iterations of the Newton algorithm used to solve the problem of evolution at time $t \in \mathcal{T}^{\text{h}}$ and for parameter $\boldsymbol{\mu} \in \mathcal{P}^{\text{s}}$. 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 [60, 23]. Technically, for each subdomain $e \in \mathcal{E}^{\text{red}}$, a matrix whose columns are the vectors of set (4.76) 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 $\mathbf{D}_{\mathbf{i}}^{(e)}$.

Selection of the control points

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

In the context of the DEIM [23], the selection is performed in a greedy manner, for increasing rank of operator $\mathbf{D}_{\mathbf{i}}^{(e)}$, 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

$$\boldsymbol{\epsilon}_{\mathbf{i,gap}}^{(e),j} = \mathbf{D}_{\mathbf{i},[1,j]}^{(e)} \boldsymbol{\beta}^j - \mathbf{D}_{\mathbf{i},j+1}^{(e)} , \tag{4.77}$$

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

$$\boldsymbol{\beta}^j = \underset{\boldsymbol{\beta}^\star \in \mathbb{R}^j}{\operatorname{argmin}} \left(\left\| \mathbf{D}_{\mathbf{i},[1,j]}^{(e)} \boldsymbol{\beta}^\star - \mathbf{D}_{\mathbf{i},j+1}^{(e)} \right\|_{\mathbf{P}_{\mathbf{i}}^{(e),j}} \right) , \tag{4.78}$$

The rank of the j^{th} greedy iterate $\mathbf{P}_{\mathbf{i}}^{(e),j}$ 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 $\mathbf{P}_{\mathbf{i}}^{(e),j+1}$ 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 $\mathbf{D}_{\mathbf{i}}^{(e)}$. We refer to reference [23] for more details about this technique, and in particular for a discussion about its optimality (in a greedy sense) and stability.

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 $\text{span}(\mathcal{F}^{s,(e)})$, for any subdomain $e \in \mathcal{E}$. In other words, we need to choose the rank of the matrix of left singular vectors $\mathbf{D}_{\mathbf{i}}^{(e)}$ for each subdomain

$e \in \mathcal{E}^{\text{red}}$. 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 4.3.3 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 $\mathcal{P}^{\text{s}}$. 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 $n_{\text{d}}^{(e)} = n_{\text{c}}^{(e)}$ for all subdomains $e \in \mathcal{E}$. Local indicators for the total error introduced by the reduced order modelling technique are defined as follows:

$$\forall\, e \in \mathcal{E}, \quad \nu_{\text{tot}}^{(e)} = \sum_{\boldsymbol{\mu} \in \mathcal{P}^{\text{s}}} \sum_{t \in \mathcal{T}^{\text{h}}} \left\| \mathbf{u}_{\mathbf{ex}}^{(e)}(t;\boldsymbol{\mu}) - \mathbf{u}_{\mathbf{r,sa}}^{(e)}(t;\boldsymbol{\mu}) \right\|_2 , \tag{4.79}$$

where $\mathbf{u}_{\mathbf{ex}}^{(e)}$ is the "truth" solution to the parametric time-dependant problem, which has been computed to build the POD projection space for the displacement, and $\mathbf{u}_{\mathbf{r,sa}}^{(e)}$ 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:

$$\forall\, e \in \mathcal{E}, \quad \nu_{\text{tot}}^{(e)} = \sum_{\boldsymbol{\mu} \in \mathcal{P}^{\text{s}}} \sum_{t \in \mathcal{T}^{\text{h}}} \left\| \mathbf{u}_{\mathbf{ex}}^{(e)}(t;\boldsymbol{\mu}) - \mathbf{u}_{\mathbf{r}}^{(e)}(t;\boldsymbol{\mu}) + \mathbf{u}_{\mathbf{r}}^{(e)}(t;\boldsymbol{\mu}) - \mathbf{u}_{\mathbf{r,sa}}^{(e)}(t;\boldsymbol{\mu}) \right\|_2 , \tag{4.80}$$

with $\mathbf{u}_{\mathbf{r}}^{(e)}$ 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:

$$\forall\, e \in \mathcal{E}, \quad \nu_{\mathrm{tot}}^{(e)} \leq \nu_{\mathrm{r}}^{(e)} + \nu_{\mathrm{r,sa}}^{(e)} \quad \text{with} \quad \begin{cases} \nu_{\mathrm{r}}^{(e)} = \displaystyle\sum_{\boldsymbol{\mu}\in\mathcal{P}^{\mathrm{s}}} \sum_{t\in\mathcal{T}^{\mathrm{h}}} \left\| \mathbf{u}_{\mathbf{ex}}^{(e)}(t;\boldsymbol{\mu}) - \mathbf{u}_{\mathbf{r}}^{(e)}(t;\boldsymbol{\mu}) \right\|_2 \\ \nu_{\mathrm{r,sa}}^{(e)} = \displaystyle\sum_{\boldsymbol{\mu}\in\mathcal{P}^{\mathrm{s}}} \sum_{t\in\mathcal{T}^{\mathrm{h}}} \left\| \mathbf{u}_{\mathbf{r}}^{(e)}(t;\boldsymbol{\mu}) - \mathbf{u}_{\mathbf{r,sa}}^{(e)}(t;\boldsymbol{\mu}) \right\|_2 \end{cases} \tag{4.81}$$

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 4.3.3, 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:

$$\frac{\nu_{\mathrm{r,sa}}^{(e)}}{\nu_{\mathrm{r}}^{(e)}} \leq 1 \tag{4.82}$$

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 $\mathbf{D}_{\mathbf{i}}^{(e)}$ 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 4.7 and will be the one used in the next section.

4.5 Results

4.5.1 Online numerical costs ("speed-up")

We now solve the parametric, time-dependent lattice problem described in section 4.2.4 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 4.8), 256, 441 and 961 (figure 4.9) nodes for each of the 10 subdomains. The snapshot that is used to compute the local reduced spaces is the one chosen in section 4.3.3. 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 4.3.3, using the Petrov-Galerkin formulation (system approximation) developed in section 4.4. 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.

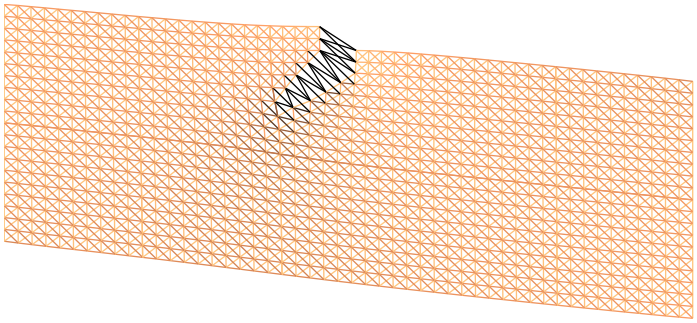


Figure 4.8: Solution corresponding to the last time step of the fully discrete time-dependent problem for a load angle of 45°. The lattice structure represented here is composed of 121 nodes per subdomain. The darkest bars correspond to a completely damaged state of the material, while the lightest bars 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.

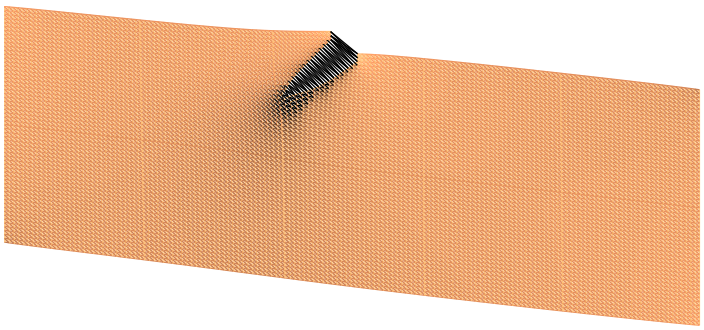


Figure 4.9: Solution correspoding to the last time step of the fully discrete time-dependent problem for a load angle of 45°, using 961 lattice nodes per subdomain.

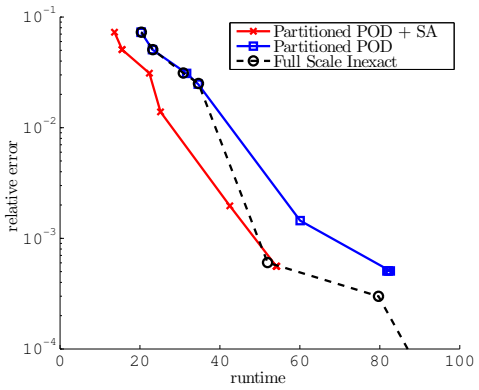
We will come back to this point in the conclusion of this work.

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 $\mathbf{u_{ex}}$. The accuracy of any approximate solution $\mathbf{u_{app}}$ will be quantified using the following normalised error function:

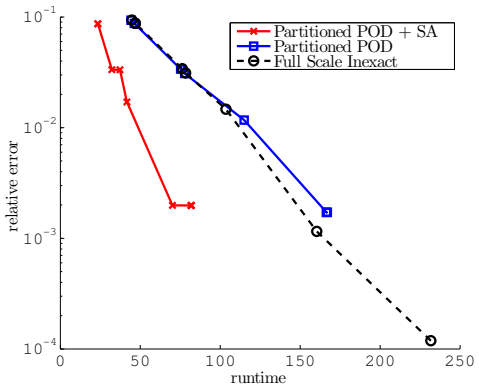
$$\nu_{\text{app}}^{(\boldsymbol{\mu})}(\mathbf{u_{app}})^2 = \frac{\sum_{t\in\mathcal{T}^h} \|\mathbf{u_{app}}(t;\boldsymbol{\mu}) - \mathbf{u_{ex}}(t;\boldsymbol{\mu})\|_2^2}{\sum_{t\in\mathcal{T}^h} \|\mathbf{u_{ex}}(t;\boldsymbol{\mu})\|_2^2} \,. \tag{4.83}$$

Secondly, an approximate solution $\mathbf{u_{inex}}$ is obtained by a straightforward time-reduction technique: the Newton algorithms are solved to a loose tolerance, and the error $\nu_{\text{app}}^{(\boldsymbol{\mu})}(\mathbf{u_{inex}})$ is reported as a function of run time in figure 4.10. 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 [97]) .

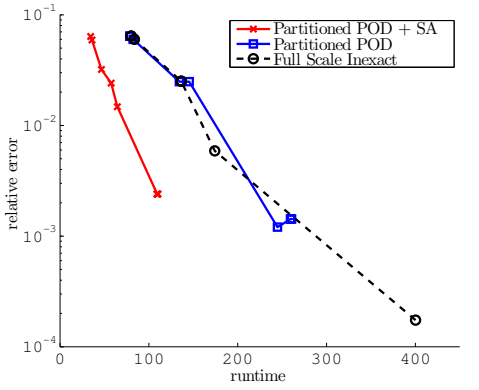
Finally, we compare the speed-up obtained when using this straightforward approach



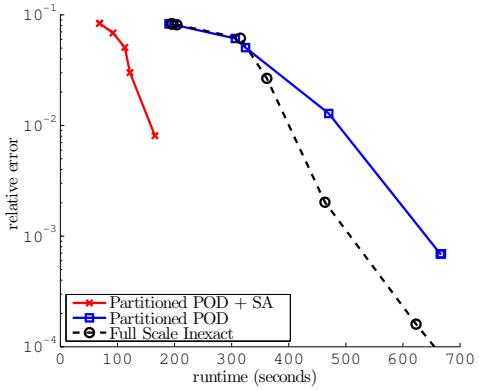
(a) Relative error for the different models using 121 nodes per subdomain



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

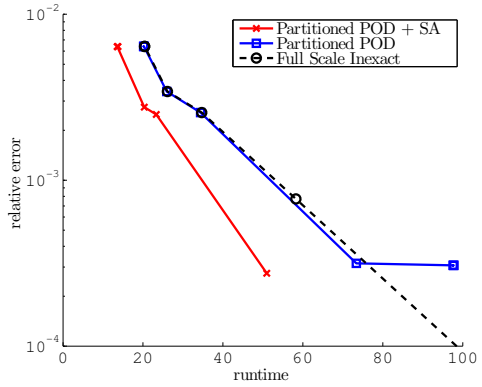


(c) Relative error for the different models using 441 nodes per subdomain

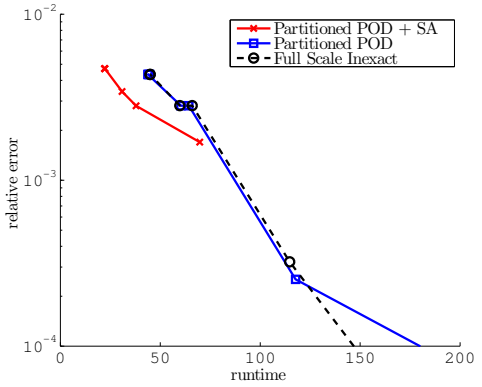


(d) Relative error for the different models using 961 nodes per subdomain

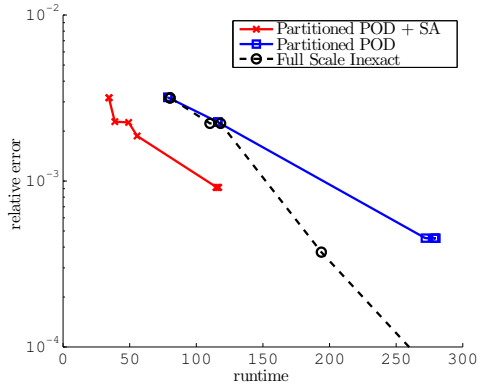
Figure 4.10: Relative error for the reference model and for the reduced order model as a function of runtime for a load angle $\theta = 40°$. The different points of the curves are generated by loosening the convergence of the Newton algorithms.



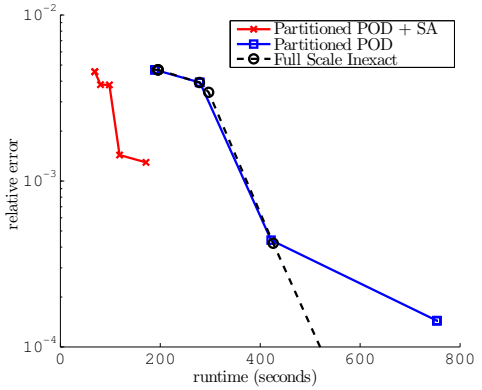
(a) Relative error for the different models using 121 nodes per subdomain



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

to the one obtained with the projection-based partitioned reduction approaches. The error between the reference solution $\mathbf{u_{ex}}$ and the one obtained by the Galerkin projection-based partitioned model order reduction (without system approximation), denoted by $\mathbf{u_r}$, is the output $\nu_{\text{app}}^{(\boldsymbol{\mu})}(\mathbf{u_r})$ of the previously defined error function. The corresponding result is labelled "Partitioned POD" in figure 4.10. The error $\nu_{\text{app}}^{(\boldsymbol{\mu})}(\mathbf{u_{r,sa}})$ of solution $\mathbf{u_{r,sa}}$ 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 4.11.

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 4.10, 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 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.

4.5.2 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 4.12 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. In the next section, a partitioned reduced order formulation using a dual domain decomposition will be presented, which can potentially solve that issue.

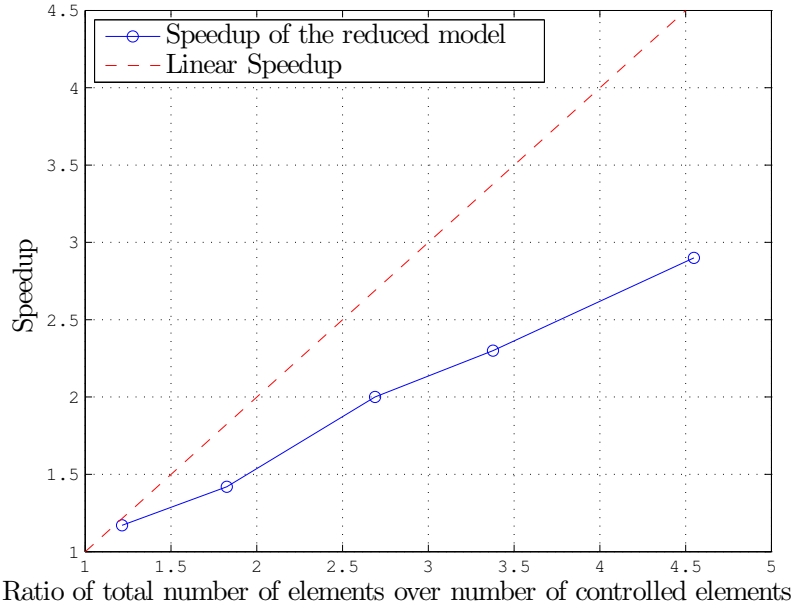


Figure 4.12: Evolution of the speed-up with the ratio of the number of elements in the structure over the number of elements comprising the reduced integration domain.

4.6 Conclusion and perspectives of this chapter

In this chapter, 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 developed in the appendix B.

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 [12] concerning the reduced basis method, or the new developments proposed in [94] 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.

CHAPTER 5

Conclusion and Discussion

In this thesis, we investigated the application of model order reduction to nonlinear multiscale mechanical problems. Two main points were focused on:

- In the context of computational homogenisation, a strategy for efficiently reducing the cost of solving the RVE boundary value problem using a projection-based model order reduction approach was presented. The RVE was modelled using a structure made of damageable beams. Sampling the parameter space, which is made of all possible far-field loading history applied on the RVE and has therefore a high-dimensionality, was one of the main challenges. It was treated by using 2 strategies:

  - a random selection of the snapshot assorted with a POD cross-validation estimate to predict the accuracy of the reduced model
  - in a Reduced basis(RB) fashion, a greedy algorithm is used to iteratively enrich the snapshot set with the loading path leading to the worst approximation by the current reduced space.

  Numerical tests show that the RB approach allows for a more efficient selection of the snapshot set by selecting the few most relevant, and therefore informative, paths. This approach generally allows for the definition of a reduced space of smaller dimension that can achieve a tolerance chosen by the user, as well as a less computationally intense offline stage. Indeed, following the more "brute force" approach of randomly selecting the loading paths, which might statistically require a high number of sampling values to explore exhaustively enough the parameter

space, may take a lot of computational effort to reach just the same accuracy reached by the RB approach proposed.

- In the second part of the thesis, the focus was made on developing a reduced-order strategy for the case of parametrised nonlinear material deformations involving a local lack of correlation, such as fracture. In this context, a standard projection-based reduced order model strategy performs typically poorly over the process zone area (i.e. the area where the fracture occurs), as a change in the parameter lead to a topologically different solution in that area. To alleviate this problem, a partitioned model order reduction built upon Schur-based domain decomposition method was proposed (both primal and dual). This approach, based on a preliminary arbitrary partition of the structure studied (but in practice may be chosen according to its geometry), allows for a definition of various independent reduced spaces of various dimension defined on each partition. A LOOCV cross-validation estimate can be used to choose the appropriate dimension of each reduced space to achieve a given tolerance, and in the particular case of a problem with a fractured area, to identify the partition containing that fracture, and use a full-order model on that area. The strategy was combined with a system approximation scheme (DEIM). Numerical results using the primal domain decomposition method show that the methods allows for a significant computational gain, while keeping a reasonable accuracy. However, a limit on the potential computational gain comes from the primal formulation which leaves the degrees of freedom lying on the interfaces unreduced. A formulation of a similar model order reduction scheme in the dual DDM was done. This formulation involves reduced spaces covering the entire subdomains, which can lead to further computational gain.

Discussion. Although reduced order modelling methods that may look attractive, in principle, were developed, it is interesting to reflect on their usefulness for an actual real-life application. In the following, we point out some weaknesses of the methods developed in this thesis, and suggest new potential improvements that could be pursued.

- In Chapter 3, we built a reduced model on a RVE that would be able to represent, in principle, any far-field load generated by a strain coming from a macroscale simulation. In our case, we considered absolutely any possible load path inside a fictitious "3× Number of timesteps" dimensional space that may well be unrealistically large. In an actual application, the set of macrostrains actually generated

by the macroscale material may be contained within a space of much smaller dimension. This means that our method may try to represent a space unnecessary too large and therefore may lead to a reduced basis of larger dimension that is really needed. In a case of a RVE geometry more complex than our test-example, the reduced basis greedy algorithm may not even converge below a reasonable tolerance.

- In Chapter 4, we presented a partitioned model order reduction technique which assumes an evenly fine discretisation across all the subdomains defined on the material. However, in a real application, the discretisation is typically uneven: it is refined along the regions of high nonlinearity, and relatively coarse on the regions far away the process zone and undergoing mild gradients. One can wonder if this method is useful when compared to an adaptive mesh refinement strategy. These two approaches are different and may each have advantages and drawbacks. Indeed, one could argue that adaptive mesh refinement techniques will essentially lead to a fine mesh in regions of high gradient and a coarse mesh in regions of mild variations while the partitioned MOR technique proposed could in principle, have a low dimensional representation of high gradients (if those vary smoothly with the parameter and time) but a higher dimensional representation of low gradients (if those vary a lot when changing the parameter). These two approaches may handle the problem differently and hence perform differently depending the problem studied.

- It may not be reasonnable to try to represent every solution with a unique basis independent of time and parameter. Indeed, in some cases, there may well just not exist a low dimensional representation of solution for any parameter and time value. However, rather than a spatial partition, one could consider a time-parameter partition, that would lead to a different reduced space depending on the value of the time and parameter.

APPENDIX A

POD using an energy-based scalar product

In this section, we assume we are dealing with a nonlinear parametrised mechanical structure with a (time and parameter dependent) stiffness matrix $\mathbf{K}(t;\boldsymbol{\mu})$. We assume we have a snapshot matrix $\mathbf{S} = [\mathbf{s}_1 \ \dots \mathbf{s}_p]$ of sample displacements of the structure available and we are looking for the orthonormal basis $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \ \dots \boldsymbol{\phi}_N]$ that best represent this snapshot. One definition of energy $\mathcal{E}(t)$ over the structure can be evaluated as:

$$\mathcal{E}(t) = \mathbf{u}(t;\mu)^T \mathbf{K}(t;\mu)\mathbf{u}(t;\mu), \tag{A.1}$$

with $\mathbf{u}(t;\mu)$ the displacement and $\mathbf{K}(t;\mu)$ the stiffness matrix. This motivates the use of the following scalar product:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{K}(t;\mu)} \stackrel{\text{def}}{=} \mathbf{x}^T \mathbf{K}(t;\mu)\mathbf{y}. \tag{A.2}$$

With that definition, the basis vectors $\boldsymbol{\phi_i}$ are now defined so that they minimise the reduced-order model error in terms of energy over the structure, which may be meaningful in some applications. However, the material non-linear behaviour and history-dependence mean that the stiffness matrix evolves during time and changes with μ. This scalar product would hence vary too, which is impractical since we are looking for a basis that is fixed in both the time and parameter space. An alternative is to pick a simplified version:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{K_0}} \stackrel{\text{def}}{=} \mathbf{x}^T \mathbf{K_0} \mathbf{y}, \tag{A.3}$$

where $\mathbf{K_0}$ is the stiffness of the structure at the initial time (which does not depend on the parameter μ). The POD minimisation problem to solve is:

$$\begin{cases} \min\limits_{\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}} J^{\mathrm{s}}_{\langle.\rangle_{\mathbf{K}_0}}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_N}) \\ \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle_{\mathbf{K}_0} = {\boldsymbol{\phi_i}}^T \mathbf{K}_0 \boldsymbol{\phi_j} = \delta_{ij}, \end{cases} \tag{A.4}$$

with

$$J^{\mathrm{s}}_{\langle.\rangle_{\mathbf{K}_0}}(\boldsymbol{\Phi}) = \sum_i^p \| \mathbf{s}_i - \sum_k^N \boldsymbol{\phi}_{\mathbf{k}}. \langle \boldsymbol{\phi}_{\mathbf{k}}, \mathbf{s}_i \rangle \|^2. \tag{A.5}$$

The Lagrangian is:

$$\mathcal{L}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_N},\boldsymbol{\mu}) = J^{\mathrm{s}}_{\langle.\rangle_{\mathbf{K}_0}}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_N}) + \sum_{i,j}^N \lambda_{ij} \left({\boldsymbol{\phi_i}}^T \mathbf{K}_0 \boldsymbol{\phi_j} - \delta_{ij} \right). \tag{A.6}$$

The necessary optimality conditions read:

$$\frac{\partial \mathcal{L}}{\partial \mu_{ij}}(\boldsymbol{\Phi},\boldsymbol{\mu}) = {\boldsymbol{\phi_i}}^T \mathbf{K}_0 \boldsymbol{\phi_j} - \delta_{ij} = 0 \in \mathbb{R}, \quad \forall\, i,j < N \tag{A.7}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi_k}}(\boldsymbol{\Phi},\boldsymbol{\mu}) = \nabla_{\boldsymbol{\phi_k}} \mathcal{L}(\boldsymbol{\Phi},\boldsymbol{\mu}) = -2 \sum_i^p \mathbf{K}_0\, \mathbf{s_i} \left({\boldsymbol{\phi_k}}^T \mathbf{K}_0\, \mathbf{s_i} \right) + \sum_{i=1}^N (\lambda_{ik} + \lambda_{ki}) \mathbf{K}_0 \boldsymbol{\phi_i}, \quad \forall\, i < N. \tag{A.8}$$

One can show that this is equivalent to:

$$\mathbf{K_0 S S}^T \mathbf{K_0} \boldsymbol{\phi_i} = \lambda_{ii} \mathbf{K_0} \boldsymbol{\phi_i}. \tag{A.9}$$

Assuming that $\mathbf{K_0}$ is not singular or that $\boldsymbol{\phi_i}$ does not excite the kernel of $\mathbf{K_0}$), equation (A.9) reduces to the eigenvalue problem:

$$\mathbf{S S}^T \mathbf{K_0} \boldsymbol{\phi_i} = \lambda_{ii} \boldsymbol{\phi_i}. \tag{A.10}$$

Solving that equation will provide a set of $\mathbf{K_0}$-orthogonal vectors that best represent the snapshot space in terms of energy. It is not symmetric anymore, so a singular value decomposition can not be used to solve that problem.

APPENDIX B

Partitioned model order reduction in a dual domain decomposition framework

In this section, we present how a projection-based reduced-order modelling technique could be applied in a dual Schur-complement domain decomposition framework, namely the finite-element tearing and interconnecting (FETI) method described in [77]. The key advantage of using a dual method over a primal one is that the interface of each subdomain can be included in the kinematic basis approximation, which allows to extend the system approximation onto the interfaces as well, resulting in a further time gains. However, this goes at the price that the original Lagrange multiplier space need to be modified to ensure the stability of the method. This matter is discussed in the following sections.

B.1 Local equilibrium

We consider a mechanical structure (possibly with a non-linear constitutive law) defined over a domain Ω. Domain Ω is now split into non-overlapping subdomains $(\Omega^{(e)})_{e\in[\![1,n_\mathrm{e}]\!]}$ such that $\bigcup_{e\in[\![1,n_\mathrm{e}]\!]}\Omega^{(e)}=\Omega$. Let $\mathbf{u}^{(e)}$ be the vector of local degrees of freedom corresponding to $\Omega^{(e)}$, $\mathbf{K}^{(e)}$ the corresponding local stiffness matrix and $f^{(e)}$ the forces applied on subdomain (e). The local equilibrium on $\Omega^{(e)}$ reads (where, in the case of a non-linear material, this equation would be a step of a Newton-Raphson iterative procedure)

$$\mathbf{K}^{(e)}\mathbf{u}^{(e)}=\mathbf{f}^{(e)}+{\mathbf{T}^{(e)}}^{T}\boldsymbol{\lambda}^{(e)}\,, \tag{B.1}$$

where $\boldsymbol{\lambda}^{(e)}$ corresponds to the contact forces applied on the boundary of $\Omega^{(e)}$, and $\mathbf{T}^{(e)}$ is the trace operator that extracts boundary degrees of freedom within subdomain (e) ($\mathbf{T}^{(e)^T}\boldsymbol{\lambda}^{(e)}$ is hence a vector with non-zero values only on the boundary of $\Omega^{(e)}$).

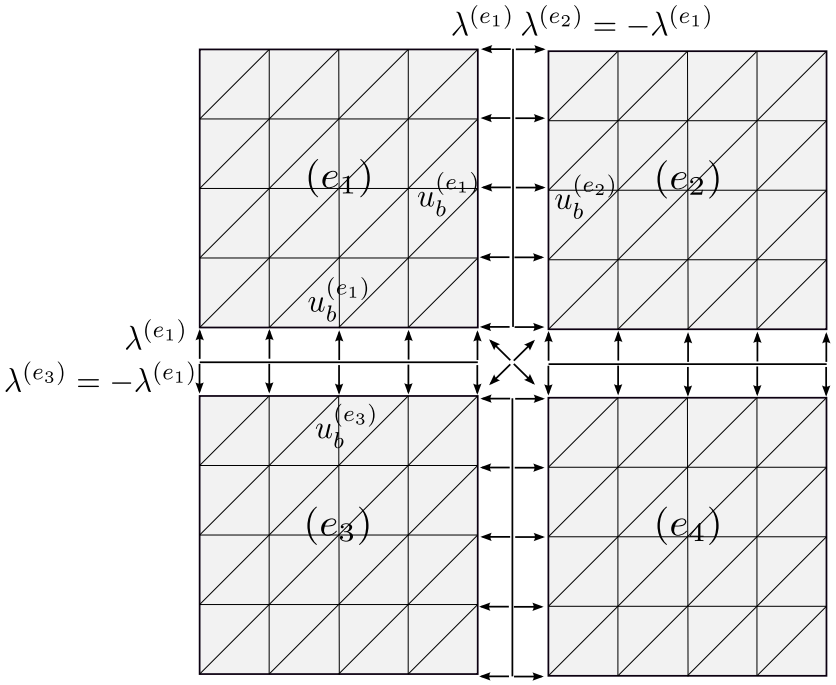


Figure B.1: Partitioning of a square structure into 4 substructures, with local nodal displacement, and nodal reaction forces. In the FETI setting, the reactions forces are uniquely defined using a single vector $\boldsymbol{\lambda}$. Note that some information is redundant at crosspoints, i.e. points that are shared by more than 2 subdomains.

B.2 Standard FETI formulation

We explain briefly here the FETI method. The reader is invited to have a look at [77] for more details about the method or at [91] for an general overview of domain decomposition methods. Unlike the Balanced domain decomposition (BDD) method which defines a unique vector of displacement on the subdomains' interfaces, the FETI method is based on the definition of a unique vector of contact forces $\boldsymbol{\lambda}$, which guarantees the compatibility of the local $\boldsymbol{\lambda}^{(e)}$'s with each other. The continuity of the displacement at the interfaces has then to be forced as:

$$\sum_{e\in\mathcal{E}} \mathbf{B}^{(e)}\mathbf{T}^{(e)}\mathbf{u}^{(e)} = \mathbf{0}\,, \tag{B.2}$$

where $\mathbf{B}^{(e)}$ is a signed boolean operator. $\mathbf{B}^{(e)}$ maps local degrees of freedom to global ones, with appropriate signs so that 2 degrees of freedom sharing an interface have opposite signs. In place where more than 2 sudomains are in contact, $\mathbf{B}^{(e)}$ gives redundant information. The local contact forces can be retrieved from the global vector of forces through:

$$\boldsymbol{\lambda}^{(e)} = \mathbf{B}^{(e)^T} \boldsymbol{\lambda}. \tag{B.3}$$

Now, the unknown in the FETI method are the contact forces rather than the displacement. The next step is to eliminate the displacement u from the local equilibrium equations (B.1). A problem arises by the fact the local stiffness matrix $\mathbf{K}^{(e)}$ is singular if the domain $\Omega^{(e)}$ does not intersect any Dirichlet boundary condition. Equations (B.1) can be solved for each subdomain $\Omega^{(e)}$ with the use of a generalised inverse. The generalised inverse is noted $\mathbf{K}^{(e)^+}$ and verifies the property $\mathbf{K}^{(e)}\,\mathbf{K}^{(e)^+}\,\mathbf{y} = \mathbf{y}$ for $\mathbf{y} \in \text{range}\left(\mathbf{K}^{(e)}\right)$. The solution to (B.1) is then expressed as:

$$\mathbf{u}^{(e)} = \mathbf{K}^{(e)^+} \left(\mathbf{F}^{(e)} + \boldsymbol{\lambda}^{(e)}\right) + \mathbf{R}^{(e)} \boldsymbol{\alpha}^{(e)}\,, \tag{B.4}$$

where $\mathbf{R}^{(e)}$ is a basis of the kernel of $\mathbf{K}^{(e)}$, which represents the rigid body motion of the subdomain (e). $\boldsymbol{\alpha}^{(e)}$ is a scalar vector that contains the magnitude of each rigid body modes. This holds only if $\left(\mathbf{F}^{(e)} + \boldsymbol{\lambda}^{(e)}\right)$ does not excite the kernel $\mathbf{R}^{(e)}$ of $\mathbf{K}^{(e)}$, which is insured through the following equation:

$$\mathbf{R}^{(e)^T} \left(\mathbf{F}^{(e)} + \boldsymbol{\lambda}^{(e)}\right) = \mathbf{0}. \tag{B.5}$$

Local equilibrium (B.4) is condensed on the interface of (e) with the use of the trace operator $\mathbf{t}^{(e)}$ and then left multiplied by the signed boolean operator $\mathbf{B}^{(e)}$. Summing over all subdomains has the effect of cancelling out the displacement (thanks to equation (B.2)) and we obtain a mixed system depending only on the tractions forces $\boldsymbol{\lambda}$ and the coefficients of the rigid body modes associated:

$$\begin{bmatrix} \mathbf{S_d} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} -\mathbf{F} \\ -\mathbf{e} \end{bmatrix} \quad \text{with} \quad \begin{cases} \mathbf{S_d} = \displaystyle\sum_{e \in \mathcal{E}} \mathbf{B}^{(e)}\, \mathbf{t}^{(e)}\, \mathbf{K}^{(e)^+}\, \mathbf{t}^{(e)^T}\, \mathbf{B}^{(e)^T} \\ \mathbf{G} = \left(\ldots\, \mathbf{B}^{(e)}\, \mathbf{t}^{(e)}\, \mathbf{R}^{(e)}\, \ldots\right) \\ \mathbf{F} = \displaystyle\sum_{e \in \mathcal{E}} \mathbf{B}^{(e)} \mathbf{t}^{(e)} \mathbf{K}^{(e)^+} \mathbf{F}^{(e)} \\ \mathbf{e} = \left(\ldots\, \mathbf{F}^{(e)^T}\, \mathbf{t}^{(e)^T}\, \mathbf{B}^{(e)}\, \mathbf{t}^{(e)}\, \mathbf{R}^{(e)}\, \ldots\right)^T \end{cases} \tag{B.6}$$

The displacement can then be computed from equation (B.4). In practice, this system is never fully assembled and it is instead solved iteratively with a projected preconditioned conjugate gradient, very much like it is done in the primal case.

B.3 Reduced FETI formulation

In this section, we develop a reduced formulation of the FETI method presented above. It will make use of two main ingredients:

- Bases $C^{(e)}$ will be used to approximate the displacement onto each subdomain.
- A basis $\mathbf{\Lambda}$ will be used to define a reduced space for the Lagrange multipliers and get a well-defined system to solve.

B.3.1 Reduction of the displacement

The displacement is expressed in the following form:

$$\mathbf{u}^{(e)} = \mathbf{C}^{(e)}\,\boldsymbol{\beta}^{(e)} + \mathbf{R}^{(e)}\,\boldsymbol{\alpha}^{(e)}, \tag{B.7}$$

where $\mathbf{R}^{(e)}$ is a basis of the kernel of $\mathbf{K}^{(e)}$ (as defined in section B.2), and $\mathbf{C}^{(e)}$ is a basis of the displacement that is chosen orthogonal to the kernel $\mathbf{R}^{(e)}$ (i.e. ${\mathbf{C}^{(e)}}^T\mathbf{R}^{(e)} = \mathbf{0}$). In this way, the displacement is explicitly described as a direct sum of a body deformation ($\mathbf{C}^{(e)}\,\boldsymbol{\beta}^{(e)}$) and a rigid body motion ($\mathbf{R}^{(e)}\,\boldsymbol{\alpha}^{(e)}$). This means also that a reduction will only be performed on the deformations of the body but not on the rigid body motions. Those can be considered irreducible and all the reduction will be concentrated onto the deformations. Substituting equation (B.7) into equation (B.1), and using Galerkin orthogonality one obtains the following equation:

$${\mathbf{C}^{(e)}}^T\mathbf{K}^{(e)}\,\mathbf{C}^{(e)}\,\boldsymbol{\beta}^{(e)} = {\mathbf{C}^{(e)}}^T\left(\mathbf{F}^{(e)} + \boldsymbol{\lambda}^{(e)}\right), \tag{B.8}$$

where we used the fact that $\mathbf{K}^{(e)}\mathbf{R}^{(e)}\,\boldsymbol{\alpha}^{(e)} = \mathbf{0}$. ${\mathbf{C}^{(e)}}^T\mathbf{K}^{(e)}\,\mathbf{C}^{(e)}$ is invertible since $\mathbf{C}$ has been chosen in a space orthogonal to the kernel of $\mathbf{K}^{(e)}$. We hence have $\boldsymbol{\beta}^{(e)} = \left({\mathbf{C}^{(e)}}^T\mathbf{K}^{(e)}\,\mathbf{C}^{(e)}\right)^{-1}{\mathbf{C}^{(e)}}^T\left(\mathbf{F}^{(e)} + \boldsymbol{\lambda}^{(e)}\right)$ and we obtain a reduced version of equation (B.4):

$$\mathbf{u}^{(e)} = \mathbf{C}^{(e)}\left({\mathbf{C}^{(e)}}^T\mathbf{K}^{(e)}\,\mathbf{C}^{(e)}\right)^{-1}{\mathbf{C}^{(e)}}^T\left(\mathbf{F}^{(e)} + \boldsymbol{\lambda}^{(e)}\right) + \mathbf{R}^{(e)}\boldsymbol{\alpha}^{(e)}\,. \tag{B.9}$$

From there, we deduce the global reduced interface problem in a similar fashion than for equation (B.6):

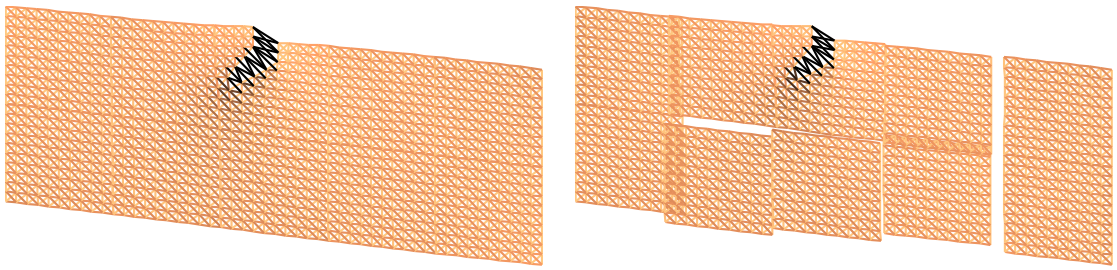
$$\begin{bmatrix} \mathbf{S}_{\mathrm{d,r}} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} -\mathbf{F}_{\mathrm{r}} \\ -\mathbf{e} \end{bmatrix}, \tag{B.10}$$

with

$$\begin{cases} \mathbf{S}_{\mathbf{dr}} = \sum_{e \in \mathcal{E}} \mathbf{B}^{(e)}\, \mathbf{t}^{(e)}\, \mathbf{C}^{(e)} \left(\mathbf{C}^{(e)^T} \mathbf{K}^{(e)}\, \mathbf{C}^{(e)} \right)^{-1} \mathbf{C}^{(e)^T}\, \mathbf{t}^{(e)^T}\, \mathbf{B}^{(e)^T} \\ \mathbf{G} = \left(\dots\, \mathbf{B}^{(e)}\, \mathbf{t}^{(e)}\, \mathbf{R}^{(e)}\, \dots \right) \\ \mathbf{F}_{\mathrm{r}} = \sum_{e \in \mathcal{E}} \mathbf{B}^{(e)} \mathbf{t}^{(e)} \mathbf{C}^{(e)} \left(\mathbf{C}^{(e)^T} \mathbf{K}^{(e)}\, \mathbf{C}^{(e)} \right)^{-1} \mathbf{C}^{(e)^T} \mathbf{F}^{(e)} \\ \mathbf{e} = \left(\dots\, \mathbf{F}^{(e)^T}\, \mathbf{t}^{(e)^T}\, \mathbf{B}^{(e)}\, \mathbf{t}^{(e)}\, \mathbf{R}^{(e)}\, \dots \right)^T \end{cases}.$$

B.3.2 Reduction of the traction forces

Reducing the displacement only, while keeping all the traction forces leads to a problem that is not well defined with too many continuity conditions on the interfaces between subdomains. This means that equation (B.10) may not have a solution or at least that the convergence of the Krylov solver will be very slow. The contact forces $\boldsymbol{\lambda}_b$ need hence to be reduced to obtain a well defined problem. Furthermore, this will have the effect of speeding up the solving of the system since it will have fewer unknowns. However, the drawback will be that the strict continuity of the displacement between two neighbouring subdomains won't be guaranteed anymore. An illustration of that is shown in Figure B.2.



(a) Solution with a basis $\boldsymbol{\Lambda}$ enriched enough

(b) Solution with a basis $\boldsymbol{\Lambda}$ too small. The contact forces are poorly approximated by $\boldsymbol{\Lambda}$: the solution is now obviously discontinuous.

Figure B.2: A damageable lattice structure is put under tension in incremental time steps. A crack at its top-centre propagates. The domain is divided into 10 subdomains and solved with a reduction of the contact forces. A basis $\boldsymbol{\Lambda}$ of different sizes is used.

The principle is to reduce the dimensionality of the global vector of contact forces by expressing it as a linear expansion, just like what was done for the displacement. Assume a basis $\boldsymbol{\Lambda} = (\boldsymbol{\Lambda}_1, \cdots, \boldsymbol{\Lambda}_N)$ that represents the Lagrange multiplier space is known. The global traction force vector $\boldsymbol{\lambda}$ can be approximated as:

$$\boldsymbol{\lambda} \approx \boldsymbol{\Lambda}\,\boldsymbol{\gamma}, \tag{B.11}$$

with $\boldsymbol{\gamma}$ a scalar vector. To make more sense from that expansion, $\boldsymbol{\Lambda}$ can be ordered in a block structure with each block representing a basis of the traction forces for a different interface between 2 subdomains (we assume there are n_b of them):

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda^1} & & & \\ & \boldsymbol{\Lambda^2} & & \\ & & \ddots & \\ & & & \boldsymbol{\Lambda^{n_\mathrm{b}}} \end{bmatrix}. \tag{B.12}$$

$\boldsymbol{\gamma}$ is ordered in the same way:

$$\boldsymbol{\gamma} = \begin{bmatrix} {\gamma^1}^T & {\gamma^2}^T & \cdots & {\gamma^{n_\mathrm{b}}}^T \end{bmatrix}^T. \tag{B.13}$$

In this fashion, $\boldsymbol{\Lambda}^i\,\boldsymbol{\gamma}^i$ is the approximation of the traction forces onto interface number i. The size of the basis of each local interface can vary, depending on the number of basis vectors used for the displacement of the neighbouring subdomains. For the special case of the crosspoints (i.e. points that share an interface with 3 or more subdomains), no reduction is performed and the associated contact values can be all grouped together as one block identity matrix on the diagonal of the global contact forces basis $\boldsymbol{\Lambda}$. Now, substituting $\boldsymbol{\lambda}$ by $\boldsymbol{\Lambda}\,\boldsymbol{\gamma}$ in equations (B.9) and (B.5), and using Galerkin orthogonality for the upper equations, we obtain the fully reduced symmetric interface problem:

$$\begin{bmatrix} \boldsymbol{\Lambda}^T\mathbf{S_d}\,\boldsymbol{\Lambda} & \boldsymbol{\Lambda}^T\mathbf{G} \\ \left(\boldsymbol{\Lambda}^T\mathbf{G}\right)^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\Lambda}^T\mathbf{F} \\ -\mathbf{e} \end{bmatrix} \tag{B.14}$$

This system has now a reduced dimension compared to that of the original one (B.10). In practice, the basis $\boldsymbol{\Lambda}$ can be determined by applying a POD decomposition from the same set of snapshots, calculated in the offline stage, that was used for the displacement, which means that it comes at no additional offline cost. Equation (B.14) can still be

solved using a Krylov subspace method.

B.4 Computational gain

Just like in the previous section, a system approximation method need to be used to insure any significant computational gain in the case the material of study is non-linear. The key advantage that this dual decomposition has over the primal decomposition is that the local displacement bases now cover the entire subdomain, including its interface. This gives room for a system approximation covering the entire subdomain rather than leaving the interface fully integrated.

The price to pay for this computational gain is that the dimensions of each basis $\mathbf{\Lambda^i}$ is an extra parameter in the reduced formulation that is non-trivial to fix. Its size has a direct influence on the continuity of the solution across subdomains and therefore its accuracy.

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] Pierre Kerfriden, Pierre Gosselet, Sondipon Adhikari, and Stephane Pierre-Alain 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):850–866, 2011.

[5] 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.

[6] Pierre Kerfriden, Olivier Goury, Timon Rabczuk, and Stephane Pierre-Alain Bordas. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. *Computer methods in applied mechanics and engineering*, 2012.

[7] A. Radermacher and S. Reese. Model reduction in elastoplasticity: proper orthogonal decomposition combined with adaptive sub-structuring. *Computational Mechanics*, 54(3):677–687, 2014.

[8] P. Kerfriden, J. J. Ródenas, and S. P.-A. Bordas. Certification of projection-based reduced order modelling in computational homogenisation by the constitutive relation error. *International Journal for Numerical Methods in Engineering*, 97(6):395–422, 2014.

[9] 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.

[10] 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.

[11] Tan Bui-Thanh, Karen Willcox, and Omar Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Daniel Millán and Marino Arroyo. Nonlinear manifold learning for model reduction in finite elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 261:118–131, 2013.

[20] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

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

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

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

[24] Charbel Farhat, Philip Avery, Todd Chapman, and Julien Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 2014.

[25] E. Sanchez-Palencia. Non homogeneous media and vibration theory. *Lecture Notes in Physics*, 127, 1980.

[26] P. Suquet. *Homogenization techniques for composite media*, chapter Elements of Homogenization for Inelastic Solid Mechanics. E. Sanchez-Palencia and A. Zaoui, Springer Verlag, 1987.

[27] Sia Nemat-Nasser and Muneo Hori. *Micromechanics: overall properties of heterogeneous materials*, volume 2. Elsevier Amsterdam, 1999.

[28] Graeme W Milton. *The theory of composites*, volume 6. Cambridge University Press, 2002.

[29] Jacob Fish and Wen Chen. Higher-order homogenization of initial/boundary-value problem. *Journal of engineering mechanics*, 127(12):1223–1230, 2001.

[30] S. Forest, F. Pradel, and K. Sab. Asymptotic analysis of heterogeneous cosserat media. *International Journal of Solids and Structures*, 38(26-27):4585–4608, 2001.

[31] G. Allaire. Homogenisation and two-scale convergence. *SIAM J. Math. Anal.*, 23(6):1482–1518, 1992.

[32] V. Buryachenko. *Micromechanics of Heterogeneous Materials*. Springer, 2007.

[33] T Mori and K Tanaka. Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta metallurgica*, 21(5):571–574, 1973.

[34] JR Willis. Bounds and self-consistent estimates for the overall properties of anisotropic composites. *Journal of the Mechanics and Physics of Solids*, 25(3):185–202, 1977.

[35] T Zohdi, M Feucht, D Gross, and P Wriggers. A description of macroscopic damage through microstructural relaxation. *International journal for numerical methods in engineering*, 43(3):493–506, 1998.

[36] F. Feyel and J.-L. Chaboche. FE^2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3-4):309–330, 2000.

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

[38] Christian Miehe. Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. *International Journal for Numerical Methods in Engineering*, 55(11):1285–1322, 2002.

[39] Tarek J. Zohdi and Peter Wriggers. *Introduction to Computational Micromechanics*, volume 20 of *lecture notes in applied and computational mechanics*. Springer, 2005.

[40] M.G.D. Geers, V.G. Kouznetsova, and W.A.M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *J. Computational Applied Mathematics*, 234(7):2175–2182, 2010.

[41] George J Dvorak. Transformation field analysis of inelastic composite materials. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 437(1900):311–327, 1992.

[42] Jean-Claude Michel and Pierre Suquet. Nonuniform transformation field analysis. *International journal of solids and structures*, 40(25):6937–6955, 2003.

[43] Felix Fritzen and T Böhlke. Three-dimensional finite element implementation of the nonuniform transformation field analysis. *International Journal for Numerical Methods in Engineering*, 84(7):803–829, 2010.

[44] Caglar Oskay and Jacob Fish. Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1216–1243, 2007.

[45] Jacob Fish, Vasilina Filonova, and Zheng Yuan. Hybrid impotent–incompatible eigenstrain based homogenization. *International Journal for Numerical Methods in Engineering*, 2013.

[46] Somnath Ghosh. *Micromechanical analysis and multi-scale modeling using the Voronoi cell finite element method*. Taylor & Francis US, 2011.

[47] 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.

[48] E Monteiro, J Yvonnet, and QC He. Computational homogenization for nonlinear conduction in heterogeneous materials using model reduction. *Computational Materials Science*, 42(4):704–712, 2008.

[49] P. Breitkopf L. Xia. A reduced multiscale model for nonlinear structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 280:117–134, 2014.

[50] 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.

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

[52] A. Arslan, R. Ince, and B. L. Karihaloo. Improved lattice model for concrete fracture. *Journal of Engineering Mechanics*, 128(157-65), 2002.

[53] Pierre Kerfriden, Karl Michael Schmidt, Timon Rabczuk, and Stephane Pierre-Alain Bordas. Statistical extraction of process zones and representative subspaces in fracture of random composites. *International Journal for Multiscale Computational Engineering*, 11(3), 2013.

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

[55] 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.

[56] S. Chaturantabut and D.C. Sorensen. Discrete empirical interpolation for nonlinear model reduction. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 4316–4321. Ieee, 2004.

[57] 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.

[58] 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.

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

[60] 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.

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

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

[63] 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.

[64] 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.

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

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

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

[68] 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.

[69] 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.

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

[71] 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.

[72] 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.

[73] 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.

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

[75] 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.

[76] 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.

[77] 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.

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

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

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] 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.

[88] 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.

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

[90] 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.

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

[92] 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.

[93] 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.

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

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

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

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