Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to Bayesian optimization

Olivier Goury, Pierre Kerfriden, Stéphane Bordas

DRAFT: Bridging analytical and computational homogenisation for nonlinear multiscale problems: a reduced order modelling approach for a damage problem

O. Goury[1], P. Kerfriden[1], S.P.A. Bordas[1]

[1] Cardiff University, School of Engineering
Queen's Buildings, The Parade, Cardiff CF24 3AA, Wales, UK

May 21, 2014

Abstract

This paper aims at bridging analytical and computational homogenisation. This is done by applying model order reduction on the representative volume element (RVE) boundary value problem. We show that this allows for great time savings, proportional to the size of the RVE discretisation, while keeping an accuracy controlled by the user. This method thus keep the versatility of computational homogenisation while being as computationally achievable as analytical homogenisation.

Keywords: model order reduction, proper orthogonal decomposition (POD), nonlinear fracture mechanics, system approximation, computational homogenisation

1 Introduction

Many natural or engineered materials are heterogeneous. More precisely, they are heterogeneous at a microscopic length scale, but homogeneous at a macroscopic length scale. For a better understanding of the material behaviour, it is necessary to modelise the macroscale while taking the microscale into account. At the microscale, the material can be made of inclusions, voids, fibers, etc.

Using supercomputers, a direct numerical simulation of the material might be at reach but at tremendous computational cost, since the microscale features need to be meshed within the coarse scale material. There is hence a need for various homogenisation techniques. Those are based on the definition of a representative volume element(RVE) [1], also called unit cell in the literature. That element has to be small enough to be considered as a point from the macroscale point of view, but large enough so that it statistically contains enough heterogeneities to be representative of the microstructure. From there, approaches can essentially be hierarchical or concurrent (or semi-concurrent), analytical of computational.

Analytical homogenisation (See for example [2]) assumes a certain constitutive law of the macrostructure that depends on a certain number of parameters that can be found by studying the behaviour of the RVE under certain loading conditions. This approach is limited to simple inclusion geometries, as it involves solving the RVE problem analytically, or at least guessing the form of its solution. This approach is not always applicable, especially in the case of materials with complicated geometries and behaving nonlinearly. The advantage is that obtaining the material parameters is then very fast. The other way is computational homogenisation. Unlike an analytical approach, it does not require any a priori guess of the solution of the RVE problem since it involves solving it anytime the material properties are needed [3]. It is computationally more expensive since the RVE boundary value problem has to be solved repeatedly at any point of the macro-mesh where material properties are needed.

This paper will aim at reducing the cost of solving the RVE boundary value problem with the use of a model order reduction method. This can be seen as a bridge between analytical and computational homogenisation. Model order reduction provides a surrogate of a full order model that retains its behaviour but can be evaluated at reduced cost. Several authors have looked at ways to alleviate the cost of solving the RVE problem. This includes [4, 5] where low order approximations of eigenstrains were used or [6] with the use of the Voronoi cell finite element method.

In this paper, a projection-based method will be used. The solution will be searched for in a space of small dimension spanned by a few basis vectors. This basis will be obtained by using the popular proper orthogonal decomposition (POD) [7]. Solutions of the RVE boundary value problem when loaded with various boundary conditions (tensile, mixed mode,...) will be stored in an "offline" stage. This provides the so-called snapshot. With the use of a POD transform, few relevant modes capturing the necessary information will be selected. In the "online" stage, the solution of the RVE will be obtained from solving the RVE reduced problem. Various applications of POD can be found in [8, 9, 10, 11]. This method hence combines both features of the analytical and the computational homogenisation, making it a accurate and versatile method. Note that the concept of applying POD-based model order reduction to computational homogenisation was already proposed in [12] for hyperelastic solid or in [13] for a conduction problem, though no system approximation was applied.

Within the POD-System approximation framework, the computations are now of the order of the reduced model, instead of the one of the full scale model. To solve the RVE boundary value problem, all that is needed now is to store the gappy operator. The approach is still computational, since the RVE boundary value problem still needs to be solved, but it is solved at very reduced costs. The "offline" stage can be thought of as a "learning" stage, where some information about the behaviour of the RVE under different scenarios of loading is stored and treated with the output being the most "informative " degrees of freedom and the corresponding gappy operator that enables a reconstruction of the missing information. In the "online" stage, the form of the solution of the RVE boundary value problem is partly restricted by the fact it needs to fit in the POD bases. This can be seen as a partly analytical method as well.

The paper is organised as follows. In section 2, we define the type of problem we want to solve and define the representative volume element equations. In section 3, we define the model order reduction approach and incorporate it in the RVE boundary value problem formulation. Section 4 presents the system approximation technique that allows a fast integration of the material properties. Section 5 presents an example and the numerical results associated. Conclusion are drawn in section 6.

2 Problem statement

We are considering a heterogeneous material with a damage constitutive law at the microscale. We assume that to correctly model the full strucuture, it is necessary to take the microstructure into account. Hence a multiscale approach will be used, namely a semi-concurrent approach (FE^2 type, see [3]). A representative volume element (RVE) will be used to numerically evaluate the constitutive law of the macrostructure. In this paper, the RVE will be modelled with a lattice structure, made of damageable beams with various Young's moduli materialising heterogeneities (inclusions in this case). This model is chosen for its simplicity and may not be very realistic, but this is just a tool for testing the reduced order model and is thus not the focus of this paper. An illustration of the RVE is displayed in Figure 1.

2.1 RVE boundary value problem

Over the RVE, the following balance equation is verified:

$$\int_{\Omega} \boldsymbol{\sigma}^{\mathbf{m}}(\boldsymbol{\delta u}) : \boldsymbol{\epsilon}^{\mathbf{m}}(\boldsymbol{\delta u}) d\Omega = \int_{\partial\Omega} \mathbf{t}.\boldsymbol{\delta u} d\Gamma \,, \tag{1}$$

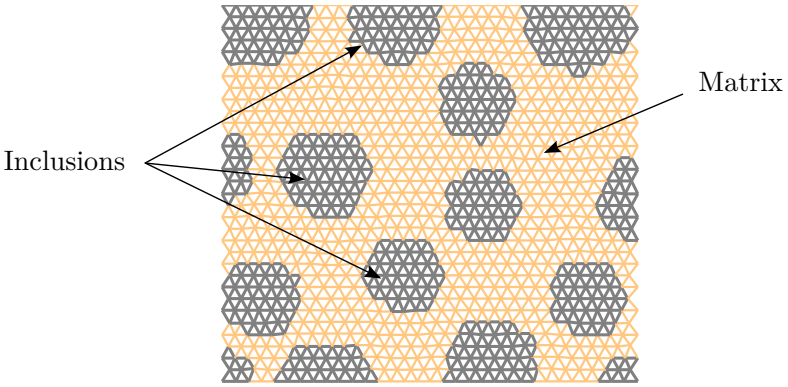


Figure 1: Lattice model of the RVE. Bars have different properties depending if they represent the matrix or the inclusions

where $\boldsymbol{\sigma}^{\mathbf{m}}$ is the Cauchy stress at the micro-level and $\boldsymbol{\epsilon}^{\mathbf{m}}$ the micro-strain. Unlike the macro-problem, the constitutive law is know. In this paper, we consider a model based on standard damage mechanics applied to a lattice structure made of beams. The model is described in [14].

2.2 Scale coupling

The Hill-Mandel principle states that the energy should be conserved from the micro to the macro scale:

$$\boldsymbol{\sigma}^M : \boldsymbol{\epsilon}^M = \int_\Omega \boldsymbol{\sigma}^m : \boldsymbol{\epsilon}^m d\Omega \tag{2}$$

The strain at the macro-level is taken as the average of micro-strain over the RVE:

$$\boldsymbol{\epsilon}^M = \frac{1}{|\Omega|} \int_\Omega \boldsymbol{\epsilon}^m d\Omega \tag{3}$$

Equations 2 and 3 imply the following relationship between microstress and macrostress:

$$\boldsymbol{\sigma}^M = \frac{1}{|\Omega|} \int_\Omega \boldsymbol{\sigma}^m d\Omega \tag{4}$$

2.3 Boundary conditions

The micro problem is defined with a damageable lattice model. The displacement over the RVE is represented as a sum of a coarse contribution $\bar{\mathbf{u}}$ given by the macro-model and a fluctuation $\tilde{\mathbf{u}}$ throughout the RVE at the micro-level:

$$\mathbf{u}(x;\mu) = \bar{\mathbf{u}}(x;\mu) + \tilde{\mathbf{u}}(x;\mu) \tag{5}$$

The fluctuation is assumed to be null over the RVE domain on average, which guarantees the fact no factitious energy is created:

$$\int_\Omega \tilde{\mathbf{u}}(x;\mu) dx = 0 \tag{6}$$

The coarse contribution is fixed by the macro-strain tensor: $\bar{\mathbf{u}}(t) = \boldsymbol{\epsilon}^{\mathbf{M}}(t)(\mathbf{x} - \bar{\mathbf{x}})$, where $\bar{\mathbf{x}}$ is the barycentre of the RVE. In this framework, the true unknown of the RVE problem is the fluctuation $\tilde{\mathbf{u}}$.

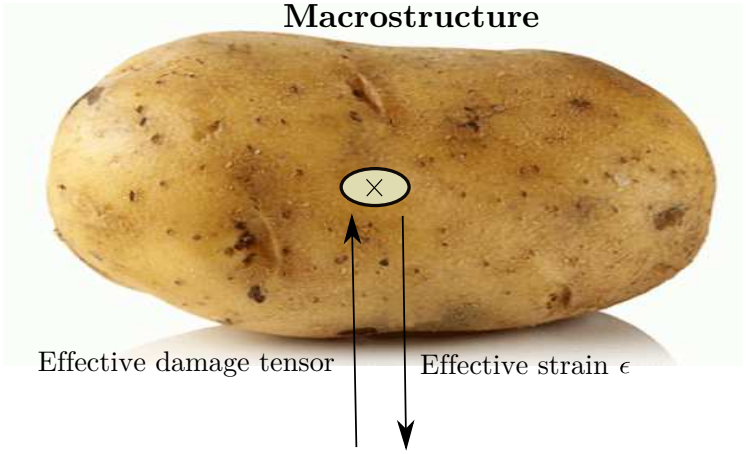


Figure 2: Schematic representation of semi-concurrent computational homogenisation. The constitutive law of the macro-structure is unknown. An effective strain is applied as boundary condition to the RVE boundary value problem. Once solved, the local material properties are obtained.

Parameter space Within our model order reduction framework, the parameter will be the effective macro-strain $\boldsymbol{\epsilon}^{\mathbf{M}} = \begin{bmatrix} \epsilon_{xx}(t) & \epsilon_{xy}(t) \\ \epsilon_{xy}(t) & \epsilon_{yy}(t) \end{bmatrix}$, or more precisely, all its values on the discretised time line $[0, T]$: $\boldsymbol{\epsilon}^{\mathbf{M}}(0), \boldsymbol{\epsilon}^{\mathbf{M}}(t_1), \ldots, \boldsymbol{\epsilon}^{\mathbf{M}}(T)$. This space of a dimension theoretically infinite, but in practice its dimension depends on the time discretisation. In any case, it is of high dimension and building a reduced model that can handle any parameter in that high-dimensional space presents a challenge.

2.4 Newton-Raphson formulation

Inserting equation (5) into equation (1), the problem can be expressed in a more general form with the unknown being the fluctuation, and the effective macro-strain being a parameter:

$$\mathbf{F}_{\text{int}}(\tilde{\mathbf{u}}; \boldsymbol{\epsilon}^{\mathbf{M}}) + \mathbf{F}_{\text{ext}}(\boldsymbol{\epsilon}^{\mathbf{M}}) = 0. \tag{7}$$

It is a non-linear system that is solved using a Newton-Raphson algorithm. At each iteration, one has to solve the following system until convergence:

$$\mathbf{K}^i \boldsymbol{\Delta}\tilde{\mathbf{u}}^{i+1} = -\mathbf{r}^i, \tag{8}$$

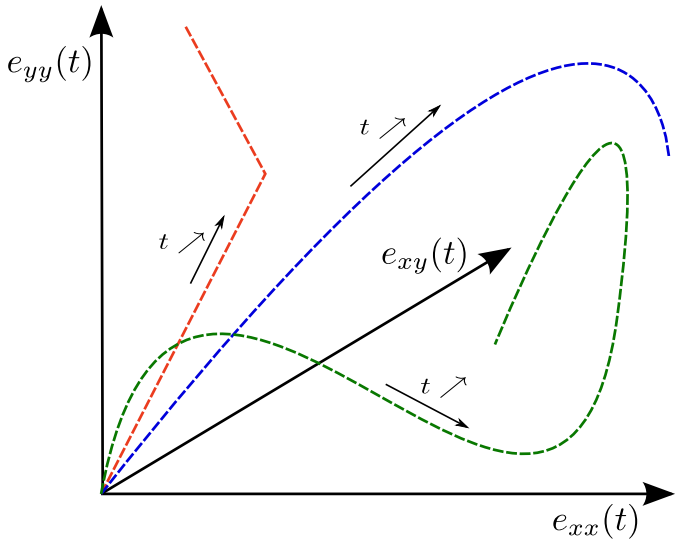


Figure 3: Representation of the parameter space of the reduced model. It is essentially any 3-dimensional curves parametrised by the time. This space is theoretically of infinite dimension. Some sample loading paths are represented.

where $\mathbf{K}^i = \frac{\partial \mathbf{F}_{\text{int}}}{\partial \tilde{\mathbf{u}}}(\tilde{\mathbf{u}}^i)$ is the tangent stiffness matrix, $\mathbf{r}^i = \mathbf{F}_{\text{int}}(\tilde{\mathbf{u}}^i) + \mathbf{F}_{\text{ext}}$ is the residual and $\boldsymbol{\Delta}\tilde{\mathbf{u}}^{i+1} = \tilde{\mathbf{u}}^{i+1} - \tilde{\mathbf{u}}^i$ is the variation in the fluctuation.

3 Model Order Reduction

Our goal is to solve the balance equations for any macro-strain $\left(\boldsymbol{\epsilon}^{\mathbf{M}}(t)\right)_{t\in[0,T]}$ at a cheap cost while keeping the accuracy of a quantity of interest to be defined. At the RVE level, we assume that for any boundary conditions applied on the RVE, the displacement field lies in a subspace of the admissible displacement space that is of small dimension, and can be span by a few basis vectors.

3.1 Projection-based model order reduction

The displacement fluctuation on the RVE is searched for in a vector space of small dimension $\mathcal{U}_{\text{MOR}} = \text{span}\left((\phi_i)_{i=1,N}\right)$:

$$\mathbf{u}(x;\boldsymbol{\epsilon}^{\mathbf{M}}) = \bar{\mathbf{u}}(x;\boldsymbol{\epsilon}^{\mathbf{M}}) + \tilde{\mathbf{u}}(x;\boldsymbol{\epsilon}^{\mathbf{M}}) = \bar{\mathbf{u}}(x;\boldsymbol{\epsilon}^{\mathbf{M}}) + \sum_1^N \boldsymbol{\phi}_i\,\alpha_i = \bar{\mathbf{u}}(x;\boldsymbol{\epsilon}^{\mathbf{M}}) + \boldsymbol{\Phi}\,\boldsymbol{\alpha} \tag{9}$$

The unknown of the RVE boundary value problem is now the state variable α rather than the displacement fluctuation $\tilde{\mathbf{u}}$.

Plugging this into the i^{th} Newton iteration (8) leads to:

$$\mathbf{K}^i\boldsymbol{\Phi}\,\boldsymbol{\Delta\alpha}^{i+1} = -\mathbf{r}^i, \tag{10}$$

The system is now under-determined, since the fluctuation and hence its variation is now restricted to a reduced space. Instead, at each step of a Newton algorithm, the state variable variation $\boldsymbol{\Delta\alpha}$ has to minimise the expression: $\|\mathbf{K}^i\boldsymbol{\Phi}\boldsymbol{\Delta\alpha}^{i+1} + \mathbf{r}^i\|$. In a Galerkin framework, this becomes:

$$\boldsymbol{\Phi}^T\mathbf{K}^i\boldsymbol{\Phi}\boldsymbol{\Delta\alpha}^{i+1} = -\boldsymbol{\Phi}^T\mathbf{r}^i \tag{11}$$

This system is of much smaller dimension than the original equation (8).

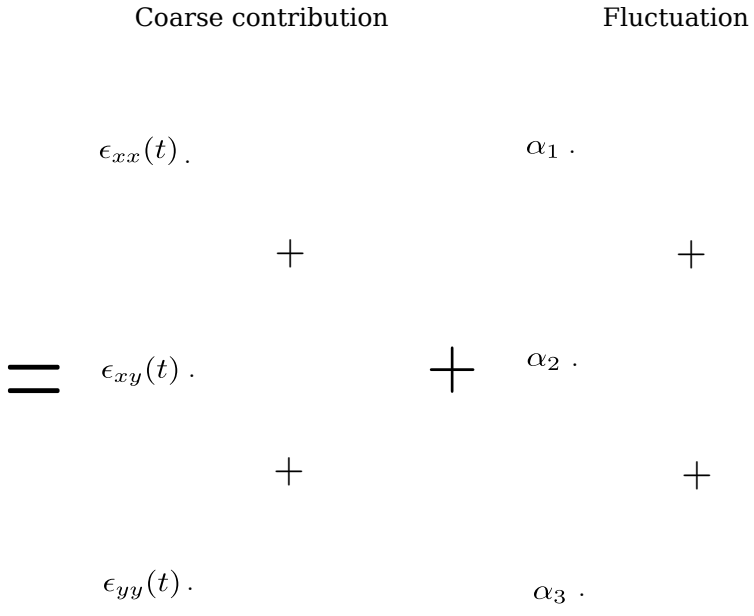


Figure 4: Representation of a RVE boundary value problem solution in the reduced order model setting: it is a sum of the coarse scale contribution and a fluctuation that is represented as a linear combination of basis vectors

3.2 Proper Orthogonal Decomposition

3.2.1 Principle

The proper orthogonal decomposition (POD) is a popular method for generating a set of orthogonal vectors representing well a high-dimensional space. It is used to define a hierarchical basis that best represent a set of data for a given amount of modes. In our case, it provides an orthonormal set of basis vectors $[\boldsymbol{\phi_1}, \boldsymbol{\phi_2}, \ldots] = \boldsymbol{\Phi}$ that minimises the following cost function for any number n_{POD} of modes:

$$J_{\langle .,. \rangle}(\boldsymbol{\Phi}) = \int_\mu \sum_t \| \mathbf{u}_{\text{ex}}(t;\mu) - \sum_k^{n_{\text{POD}}} \boldsymbol{\phi}_\mathbf{k} . \langle \boldsymbol{\phi}_\mathbf{k}, \mathbf{u}_{\text{ex}}(t;\mu) \rangle \|^2, \tag{12}$$

where $\langle ., . \rangle$ is a scalar product that remains to be defined and $\|.\| = \sqrt{\langle ., . \rangle}$ is the induced norm. The parameter μ is should be written $\mu(\{\tau | \tau \leq t\})$ as it varies with t but we denote it μ for simplifying the notations. The cost function evaluates the sum of the differences between the exact solution $\mathbf{u}_{\text{ex}}(t;\mu)$ 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 of the RVE problem with 1 vector, $[\boldsymbol{\phi_1}, \boldsymbol{\phi_2}]$ the best approximation of the solution of the RVE problem with 2 vectors and so on.

This problem requires the exact solution over the entire parameter domain and can't hence be solved. However, it can be approached by calculating a number of solutions for certain values of the parameter. This is presented in the following section.

[1] In the sense of the $\|.\|$ norm

3.2.2 Snapshot POD

Let us now assume that, for a discrete set of values $\mathcal{P}^{\mathrm{s}}$ of the parameter μ at some specific timesteps, the exact solution is available. The set of those solutions is called the snapshot. We define the snapshot matrix $\mathbf{S} = \left[\mathbf{s}_1, \ldots, \mathbf{s}_{n_\mu}\right] = (\mathbf{u}_{\mathrm{ex}}(t;\mu))_{t\in[0,T],\mu\in\mathcal{P}^{\mathrm{s}}}$. The discrete version of the cost function (12) defined over $\mathcal{P}^{\mathrm{s}}$ reads:

$$J^{\mathrm{s}}_{\langle.\rangle}(\mathbf{\Phi}) = \sum_i \|\mathbf{s}_i - \sum_k^{n_{\mathrm{POD}}} \boldsymbol{\phi}_{\mathbf{k}} . \langle \boldsymbol{\phi}_{\mathbf{k}}, \mathbf{s}_i\rangle \|^2. \tag{13}$$

The minimisation problem associated to this cost function is:

$$\begin{cases} \min\limits_{\boldsymbol{\phi_1},\ldots,\boldsymbol{\phi_l}} J^{\mathrm{s}}_{\langle.\rangle}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) \\ \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j}\rangle = \delta_{ij}. \end{cases} \tag{14}$$

This problem can be solved analytically by defining the Lagrangian:

$$\mathcal{L}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l},\boldsymbol{\mu}) = J^{\mathrm{s}}_{\langle.\rangle_{\mathbf{K}_0}}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) + \sum_{i,j}^{l} \lambda_{ij}\left(\left\langle \boldsymbol{\phi}_{\mathbf{i}}, \boldsymbol{\phi}_{\mathbf{j}}\right\rangle - \delta_{ij}\right). \tag{15}$$

Canonical scalar product

The next step is to define a scalar product $\langle.\rangle$. One common choice is to chose the canonical scalar product, that is: $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^T\mathbf{y}$.

In this case, the necessary optimality conditions read:

$$\frac{\partial \mathcal{L}}{\partial \mu_{ij}}(\mathbf{\Phi},\boldsymbol{\mu}) = \boldsymbol{\phi_i}^T\boldsymbol{\phi_j} - \delta_{ij} = 0 \in \mathbb{R}, \quad \forall\, i,j < l \tag{16}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi_k}}(\mathbf{\Phi},\boldsymbol{\mu}) = \nabla_{\boldsymbol{\phi_k}}\mathcal{L}(\mathbf{\Phi},\boldsymbol{\mu}) = -2\sum_i \mathbf{s_i}\left(\boldsymbol{\phi_k}^T\mathbf{s_i}\right) + \sum_{i=1}^{l}(\lambda_{ik}+\lambda_{ki})\boldsymbol{\phi_i}, \quad \forall\, i < l. \tag{17}$$

One can show that this is equivalent to:

$$\mathbf{S}\mathbf{S}^T\boldsymbol{\phi_k} = \lambda_{kk}\boldsymbol{\phi_k} \quad \text{and} \quad \lambda_{ij} = 0 \quad \forall i \neq j\,, \tag{18}$$

which is a simple symmetric eigenvalue problem.

It can be solve using a singular value decomposition:

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T, \tag{19}$$

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 know determine a simple error formula the POD basis of order l:

$$\left(J^{\mathrm{s}}_{\langle.\rangle}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l})\right)^2 = \sum_i \|\mathbf{s}_i - \sum_{k=1}^{l} \boldsymbol{\phi}_{\mathbf{k}}\langle \boldsymbol{\phi}_{\mathbf{k}}, \mathbf{s}_i\rangle\|^2 = \sum_i \sum_{k=l+1}^{n} \|\boldsymbol{\phi}_{\mathbf{k}}\langle \boldsymbol{\phi}_{\mathbf{k}}, \mathbf{s}_i\rangle\|^2 = \sum_{k=l+1}^{n}\lambda_{kk} \tag{20}$$

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

$$\nu(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) = \sqrt{\frac{\sum_{k=l+1}^{n}\lambda_{kk}}{\sum_{k=1}^{n}\lambda_{kk}}}. \tag{21}$$

3.2.3 POD using a structure-dependent scalar product

Using a canonical scalar product means that what is being minimised is the average error of the displacement over the entire RVE domain (this is according to the definition of the cost function $J^{s}_{\langle.\rangle}$). This may not be the best choice. In the multiscale framework of computational homogenisation, the reduced model should be able to predict the average stress on the RVE, since it is the output quantity of the RVE simulation that is then used in the macro-simulation. The energy $\mathcal{E}(t)$ over the RVE is hence a quantity of interest. It is evaluated as:

$$\mathcal{E}(t) = \mathbf{u}(t;\mu)^T \mathbf{K}(t;\mu)\mathbf{u}(t;\mu), \tag{22}$$

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

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{K}(t;\mu)} \overset{\text{def}}{=} \mathbf{x}^T \mathbf{K}(t;\mu)\mathbf{y}. \tag{23}$$

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 RVE, which is meaningful. 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:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{K_0}} \overset{\text{def}}{=} \mathbf{x}^T \mathbf{K_0}\mathbf{y}, \tag{24}$$

where $\mathbf{K_0}$ is the stiffness of the structure at the initial time (which does not depend on the parameter μ). This scalar product, though symmetric since $\mathbf{K_0}$ is, is not well-defined: $\mathbf{K_0}$ is singular since rigid body motions create no forces. However, the basis $\mathbf{\Phi}$ is a representation of the displacement fluctuation only (which has zero displacement on the RVE boundary), it has hence no rigid body modes contribution. Therefore, on the displacement fluctuation space, the operator $\langle ., . \rangle_{\mathbf{K_0}}$ is indeed a well-defined scalar product. In this case, the new minimisation problem to solve is:

$$\begin{cases} \min\limits_{\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}} J^{s}_{\langle.\rangle_{\mathbf{K}_0}}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) \\ \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle_{\mathbf{K}_0} = {\boldsymbol{\phi_i}}^T \mathbf{K}_0 \boldsymbol{\phi_j} = \delta_{ij}. \end{cases} \tag{25}$$

The Lagrangian becomes:

$$\mathcal{L}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l},\boldsymbol{\mu}) = J^{s}_{\langle.\rangle_{\mathbf{K}_0}}(\boldsymbol{\phi_1},...,\boldsymbol{\phi_l}) + \sum_{i,j}^{l} \lambda_{ij} \left({\boldsymbol{\phi_i}}^T \mathbf{K}_0 \boldsymbol{\phi_j} - \delta_{ij} \right). \tag{26}$$

The necessary optimality conditions read:

$$\frac{\partial \mathcal{L}}{\partial \mu_{ij}}(\mathbf{\Phi},\boldsymbol{\mu}) = {\boldsymbol{\phi_i}}^T \mathbf{K}_0 \boldsymbol{\phi_j} - \delta_{ij} = 0 \in \mathbb{R}, \quad \forall\, i,j < l \tag{27}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi_k}}(\mathbf{\Phi},\boldsymbol{\mu}) = \nabla_{\boldsymbol{\phi_k}} \mathcal{L}(\mathbf{\Phi},\boldsymbol{\mu}) = -2 \sum_i \mathbf{K}_0\, \mathbf{s_i} \left({\boldsymbol{\phi_k}}^T \mathbf{K}_0\, \mathbf{s_i} \right) + \sum_{i=1}^{l} (\lambda_{ik} + \lambda_{ki}) \mathbf{K}_0 \boldsymbol{\phi_i}, \quad \forall\, i < l. \tag{28}$$

One can show that this is equivalent to:

$$\mathbf{K_0 S S}^T \mathbf{K_0} \boldsymbol{\phi_i} = \lambda_{ii} \mathbf{K_0} \boldsymbol{\phi_i}. \tag{29}$$

Taking into account the fact that $\boldsymbol{\phi_i}$ is a basis of the fluctuation on the RVE and hence has no component of a rigid body mode ($\boldsymbol{\phi_i} \in \mathrm{Ker}(\mathbf{K_0})^{\perp}$), equation (29) reduces to the eigenvalue problem:

$$\mathbf{S S}^T \mathbf{K_0} \boldsymbol{\phi_i} = \lambda_{ii} \boldsymbol{\phi_i}. \tag{30}$$

Solving that equation will provide us a set of $\mathbf{K_0}$-orthogonal vectors that best represent our snapshot space in terms of energy. It is not symmetric anymore, so a singular value decomposition can not be used to solve that problem, and we have to use a more costly eigenvalue decomposition. This is not a big concern since all those computations are done in the offline stage and are done once and for all. One can prove that the error formula (21) obtained with the canonical scalar product remains true with this special scalar product.

4 System Approximation

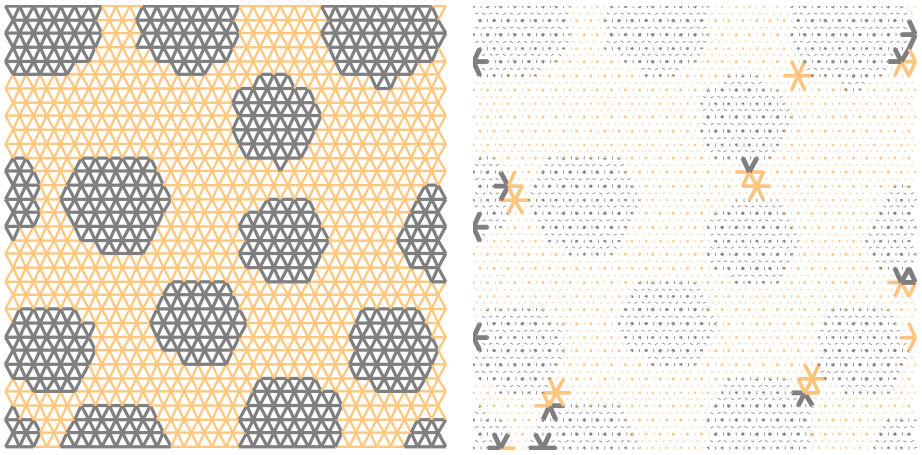
Constraining the displacement in a low-dimensional space does not provide a significant computational gain, even if the systems to be solved are of smaller dimension. This is because the material of study is nonlinear and history-dependent, and its stiffness varies not only in different areas of the 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 [15], the missing point estimation [16], or system approximation [17]. 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 paper, we will use the "gappy" method, very much like in [17, 18].

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

$$\mathbf{F}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha}) \approx \sum_{1}^{n_{\text{gap}}} \boldsymbol{\psi}_{\mathbf{i}} \beta_i = \boldsymbol{\Psi}\boldsymbol{\beta}, \tag{31}$$

where $\left[\boldsymbol{\psi}_{\mathbf{1}}, \cdots, \boldsymbol{\psi}_{\mathbf{n}_{\text{gap}}}\right] = \boldsymbol{\Psi}$ 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 5: 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 $\mathcal{I}$:

$$\underset{\boldsymbol{\beta}^\star}{\operatorname{argmin}} \|\mathbf{F}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha}) - \boldsymbol{\Psi}\boldsymbol{\beta}^\star\|_{\mathbf{P}}, \tag{32}$$

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:

$$\mathbf{E}^T\,\boldsymbol{\Psi}\boldsymbol{\beta} = \mathbf{E}^T\,\mathbf{F}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha}) \tag{33}$$

which implies:

$$\boldsymbol{\beta} = (\mathbf{E}^T\boldsymbol{\Psi})^{-1}\mathbf{E}^T\mathbf{F}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha}) \tag{34}$$

At a newton iteration of our POD-Galerkin framework, this reduces equation (11) to:

$$\boldsymbol{\Phi}^T\boldsymbol{\Psi}(\mathbf{E}^T\boldsymbol{\Psi})^{-1}\mathbf{E}^T\,\mathbf{K}\boldsymbol{\Phi}\boldsymbol{\Delta}\boldsymbol{\alpha} = -\boldsymbol{\Phi}^T\mathbf{r}. \tag{35}$$

This can be rewritten in the form:

$$\mathbf{G}\mathbf{E}^T\,\mathbf{K}\boldsymbol{\Phi}\boldsymbol{\Delta}\boldsymbol{\alpha} = -\left(\mathbf{G}\,\mathbf{E}^T\,\mathbf{F}_{\text{int}}(\boldsymbol{\Phi}\,\boldsymbol{\alpha}) + \boldsymbol{\Phi}^T\mathbf{F}_{\text{ext}}\right), \tag{36}$$

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, all that the bases $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are calculated, 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 small dimension and solve it which is computationally much cheaper.*

4.2 Selection of the controlled elements

The selection of the control elements will be done using the discrete empirical interpolation method (DEIM) [18]. This method finds a set of degrees of freedom $\mathcal{I}$ in a greedy manner from the internal forces basis $\mathbf{\Psi}$. 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 $\mathbf{r}_{\text{gap}} = \left|\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}\boldsymbol{\beta}^{\mathbf{j}} - \boldsymbol{\psi}_{\mathbf{j+1}}\right|$ is evaluated, where $\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}$ is the matrix containing the first j vectors of the basis $\mathbf{\Psi}$ and $\boldsymbol{\psi}_{\mathbf{j+1}}$ is the $j+1^{th}$ vectors in that basis. $\boldsymbol{\beta}$ is the solution of the minimisation problem

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}^\star}{\operatorname{argmin}} \left\| \mathbf{E^j}\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}\boldsymbol{\beta}^\star - \mathbf{E^j}\boldsymbol{\psi}_{\mathbf{j+1}} \right\|_2 . \tag{37}$$

The solution is easily found: $\boldsymbol{\beta} = \left(\mathbf{E^j}\boldsymbol{\psi}_{[\mathbf{1},\mathbf{j}]}\right)^{-1} \mathbf{E^j}\boldsymbol{\psi}_{\mathbf{j+1}}$. 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 (33). At the end of the greedy algorithm, the number of control degrees of freedom chosen equals the number of basis vectors $(\boldsymbol{\psi}_{\mathbf{i}})_{n_{\text{gap}}}$ which makes system (33) well defined.

5 Example and Numerical Results

We consider a sample RVE made of damageable lattice bars. The RVE is composed of a matrix with inclusions. Bars with materialising inclusions are given a high Young's modulus compared to the matrix bars. This model is simple but the focus of this study is not the multiscale model itself but the application of model order reduction.

5.1 Snapshot selection

The parameter space, made of all possible values of the macro-strain $\boldsymbol{\epsilon}^{\mathbf{M}}(t)$ for $t \in [0, T]$, is of very high dimension.

5.1.1 Arbitrary regular sampling

We simplify the snapshot selection to monotonic increase of the strain with time in the following form:

$$\boldsymbol{\epsilon}^{\mathbf{M}}(t) = \frac{t}{N} \cdot \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{xy} & \epsilon_{yy} \end{bmatrix} . \tag{38}$$

The snapshots are then chosen so that the macro-strains have an equivalent "norm": $\epsilon_{xx}^2+\epsilon_{yy}^2+\epsilon_{xy}^2 = C$, where C is a constant. They are equally distributed on the 3-dimensional sphere parametrised by ϵ_{xx}, ϵ_{yy} and ϵ_{xy}. 100 snapshots are taken on that sphere. This is schematically represented in Figure 6.

5.1.2 Greedy enrichment of the snapshot space

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 $\boldsymbol{\epsilon}^{\boldsymbol{M}}$) 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. The parameter space dimension grows with the number of timesteps and can hence be quite large. To ensure an exhaustive

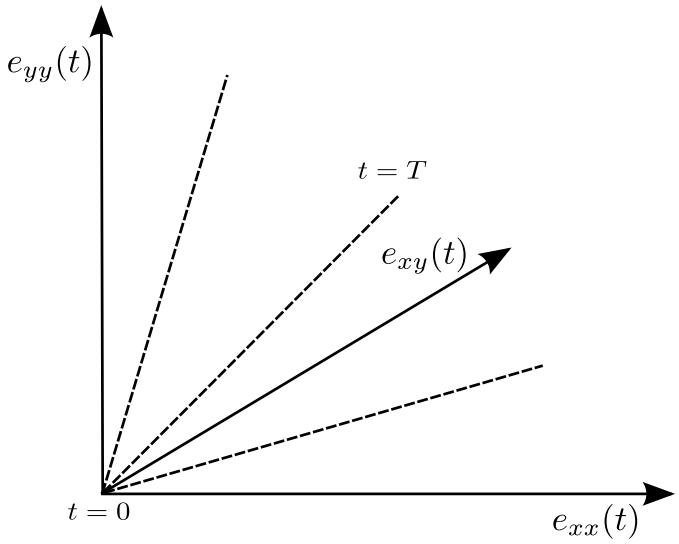


Figure 6: Sample loading paths taken used to generate the snapshot space. The loading paths are linear and monotonic.

sampling, it is necessary to enrich the arbitrary sampling defined in the previous section. We will do so following a greedy iterative procedure.

The idea is to enrich the snapshot space with the solution $\mathbf{u_{enrich}}$ 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.

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:

$$\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_{i+1}) = \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\max}(t_i) + \Delta\epsilon^i_{\max}. \tag{39}$$

The procedure iterates until reaching fracture. The maximisation process to find the the load increment $\Delta\epsilon^i_{\max}$ is described in the following paragraphs. Check Figure 7 to see load paths samples generated by this algorithm.

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 $\boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau<t_i}}$:

$$f_{\text{err}}\left(t_i, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}\right) = \|\mathbf{u_{ex}}\left(t_i, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}\right) - \mathbf{u_r}\left(t_i, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}\right)\|_{\mathbf{K_0}}. \tag{40}$$

This function 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. In this purpose, we solve the following maximisation problem:

$$\begin{cases} \max\limits_{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}}} f_{\text{err}}\left(t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau\leq t_i}}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{t_i}} + \boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}}\right) \\ \left\|\boldsymbol{\Delta\epsilon}^{\boldsymbol{M\star}}\right\| = \delta_{\text{step}}, \end{cases} \tag{41}$$

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 (41) can be rewritten to become unconstrained:

$$\max_{\Delta\epsilon^\star_{xx},\Delta\epsilon^\star_{yy}} f_{\text{err}}\left(t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{t_i}} + \widehat{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}(\Delta\epsilon^\star_{xx}, \Delta\epsilon^\star_{yy})\right) = \widehat{f_{\text{err}}}\left(\Delta\epsilon^\star_{xx}, \Delta\epsilon^\star_{yy}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}\right), \tag{42}$$

where $\widehat{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}(\Delta\epsilon^\star_{xx}, \Delta\epsilon^\star_{yy}) = \begin{pmatrix} \Delta\epsilon^\star_{xx} \\ \Delta\epsilon^\star_{yy} \\ \sqrt{\delta^2_{\text{step}} - {\Delta\epsilon^\star_{xx}}^2 - {\Delta\epsilon^\star_{yy}}^2} \end{pmatrix}$ which guarantees $\left\|\widehat{\boldsymbol{\Delta\epsilon}^{\boldsymbol{M}}}\right\| = \delta_{\text{step}}$. In this case we assume $\Delta\epsilon_{xy} \geq 0$ so we keep a positive sign in front of the square root.

Gradient descent algorithm

To solve problem (42), we use a gradient descent method (also famously known as steepest descent). This is a first order optimisation method. From an initial guess $\boldsymbol{\Delta\epsilon}^{\boldsymbol{0}} = \begin{pmatrix} \Delta\epsilon^0_{xx} \\ \Delta\epsilon^0_{yy} \end{pmatrix}$, a sequence of iterates is found using the following update:

$$\boldsymbol{\Delta\epsilon}^{\boldsymbol{k+1}} = \boldsymbol{\Delta\epsilon}^{\boldsymbol{k}} + \gamma \frac{\partial\widehat{f_{\text{err}}}}{\partial\left(\Delta\epsilon_{xx}, \Delta\epsilon_{yy}\right)}(\boldsymbol{\Delta\epsilon}^{\boldsymbol{k}}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}), \tag{43}$$

where γ is a scalar whose value is found through a linesearch using a bisection algorithm.

The derivative $\frac{\partial\widehat{f_{\text{err}}}}{\partial(\Delta\epsilon_{xx},\Delta\epsilon_{yy})}(\boldsymbol{\Delta\epsilon}^{\boldsymbol{k}}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}})$ is evaluated numerically by taking small variations (of size ν) around the value $\boldsymbol{\Delta\epsilon}^{\boldsymbol{k}}$:

$$\frac{\partial\widehat{f_{\text{err}}}}{\partial\left(\Delta\epsilon_{xx}, \Delta\epsilon_{yy}\right)}(\boldsymbol{\Delta\epsilon}^{\boldsymbol{k}}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}) \approx \frac{1}{\nu}\begin{bmatrix} \widehat{f_{\text{err}}}\left(\begin{pmatrix} \Delta\epsilon^k_{xx}+\nu \\ \Delta\epsilon^k_{yy} \end{pmatrix}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}\right) - \widehat{f_{\text{err}}}\left(\boldsymbol{\Delta\epsilon}^{\boldsymbol{k}}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}\right) \\ \widehat{f_{\text{err}}}\left(\begin{pmatrix} \Delta\epsilon^k_{xx} \\ \Delta\epsilon^k_{yy}+\nu \end{pmatrix}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}\right) - \widehat{f_{\text{err}}}\left(\boldsymbol{\Delta\epsilon}^{\boldsymbol{k}}; t_{i+1}, \boldsymbol{\epsilon}^{\boldsymbol{M}}_{\boldsymbol{\tau}\leq\boldsymbol{t_i}}\right) \end{bmatrix}, \tag{44}$$

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.

5.2 POD basis

The basis $\boldsymbol{\Phi}$ is selected using the POD method explained in section 3.2. The first few modes are displayed in Figure 8. We will test the results using various number of basis vectors. Procedures to select the optimal number of modes according to robust cross-validation procedures can be found in [11].

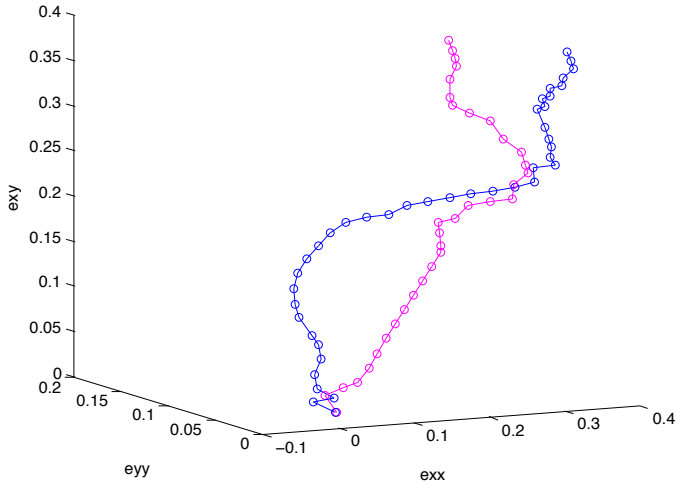
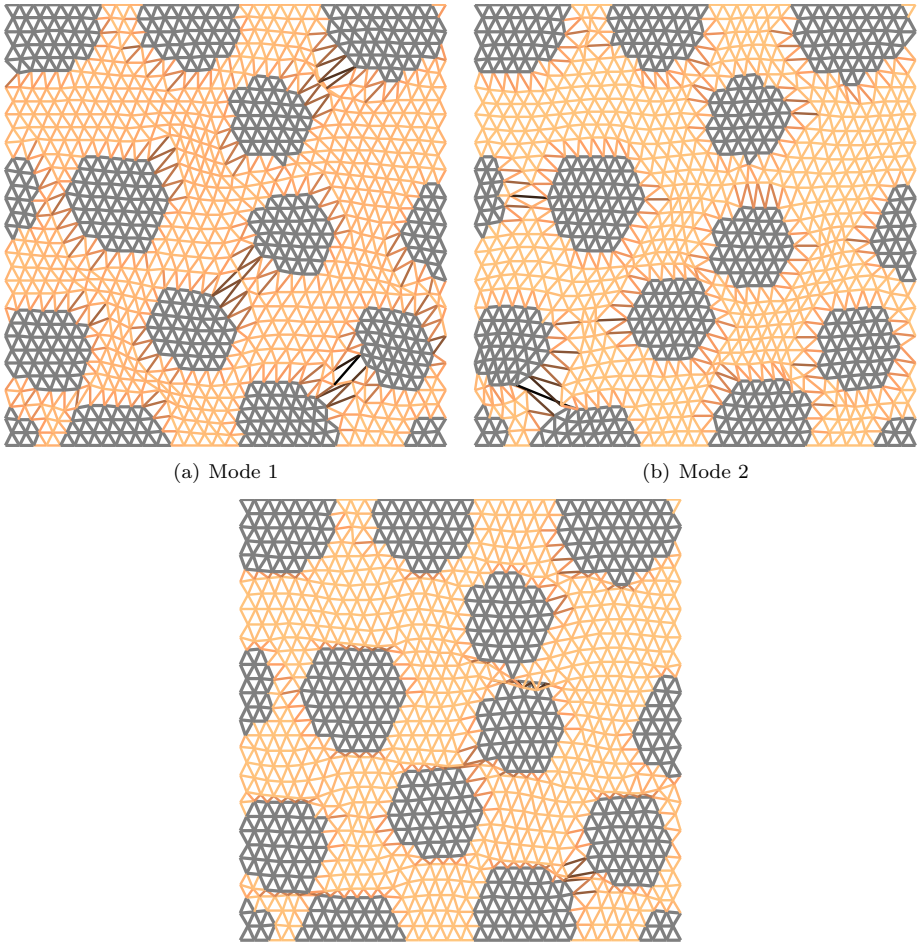


Figure 7: Sample loading paths generated by the algorithm



(a) Mode 1

(b) Mode 2

(c) Mode 3

Figure 8: First 3 modes obtained through the POD. The damage localises between pairs of inclusions.

5.3 System approximation

We follow the procedure described in 4. The basis $\mathbf{\Psi}$ is extracted from the same snapshot space as used for the displacement basis $\mathbf{\Phi}$. The set of controlled elements is selected using the DEIM [18]. The amount of vectors in the basis $\mathbf{\Psi}$ is chosen so that the system approximation does not increase 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 dynamic basis $\mathbf{\Phi}$, and $\mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi})$ the complete reduced order model with system approximation using the dynamic basis $\mathbf{\Phi}$ and the static basis $\mathbf{\Psi}$):

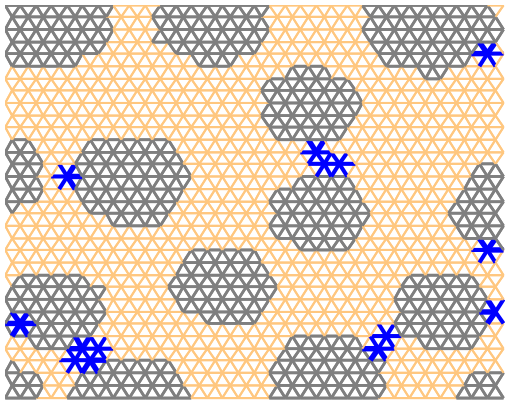
$$\nu_{\text{tot}}(t)^2 = \|\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi})\|_2^2 = \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r}}(t;\mathbf{\Phi})) + (\mathbf{u}_{\text{r}}(t;\mathbf{\Phi}) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi}))\|_2^2 \tag{45}$$

$$\leq \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r}}(t;\mathbf{\Phi}))\|_2^2 + \|(\mathbf{u}_{\text{r}}(t;\mathbf{\Phi}) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi}))\|_2^2 . \tag{46}$$

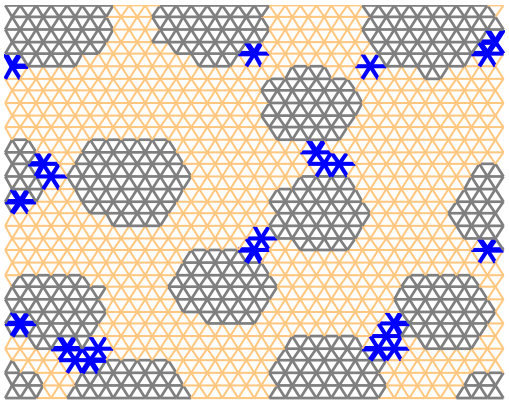
Taking this in consideration, the basis $\mathbf{\Psi}$ is chosen to be the smallest that verifies the inequality:

$$\|(\mathbf{u}_{\text{r}}(t;\mathbf{\Phi}) - \mathbf{u}_{\text{r,sa}}(t;\mathbf{\Phi},\mathbf{\Psi}))\|_2 \leq \frac{1}{10}\|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{r}}(t;\mathbf{\Phi}))\|_2 . \tag{47}$$

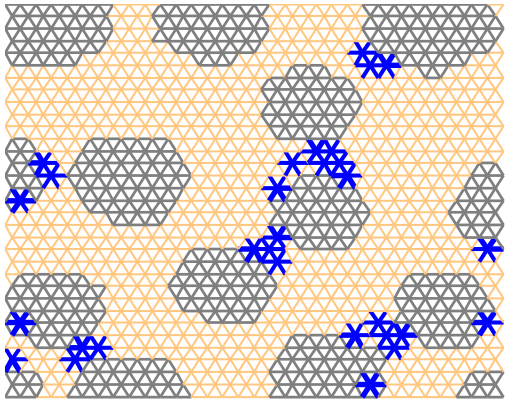
This guarantees that error generated by the system approximation remains insignificant compared to the error generated by approximating the displacement. More details can be found in [11]. The location of controlled elements (which are all the elements in contact with the control degrees of freedom) is shown in Figure 9 for various basis sizes.



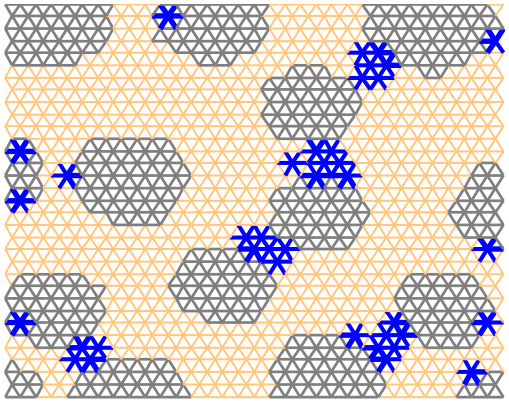
(a) Controlled elements with 2 "dynamic" basis vectors. 16 "static" basis vectors are needed to achieve the minimal accuracy



(b) Controlled elements with 4 "dynamic" basis vectors. 28 "static" basis vectors are needed to achieve the minimal accuracy



(c) Controlled elements with 10 "dynamic" basis vectors. 34 "static" basis vectors are needed to achieve the minimal accuracy



(d) Controlled elements with 15 "dynamic" basis vectors. 48 "static" basis vectors are needed to achieve the minimal accuracy

Figure 9: 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.

5.4 Numerical results

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 model. We will focus on the simulation of the RVE problem only, and not consider the complete computational homogenisation framework (the macroscale problem will not be solved). For the first test, the load path is set using the following effective strain: $\boldsymbol{\epsilon}^{\mathbf{M}}(t) = \frac{t}{T} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Note that this case is not in the snapshot. First, the relative error is plotted with various numbers of POD basis vectors. See Figure 11.

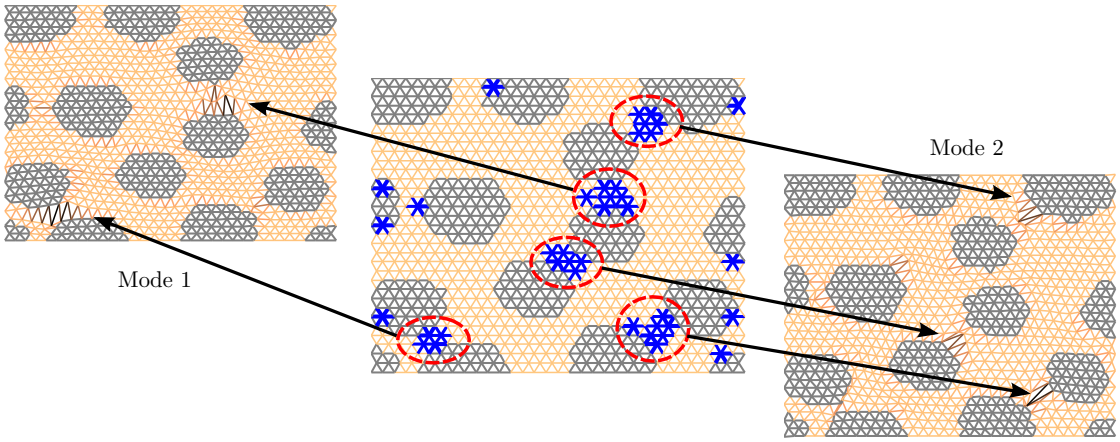


Figure 10: Regions of interest selected by the system approximation procedure. Those regions (circled in the Figure) are matching the areas of higher displacement found in the POD bases. This is intuitively good, since those elements have to give enough information to be able to reconstruct the internal 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.

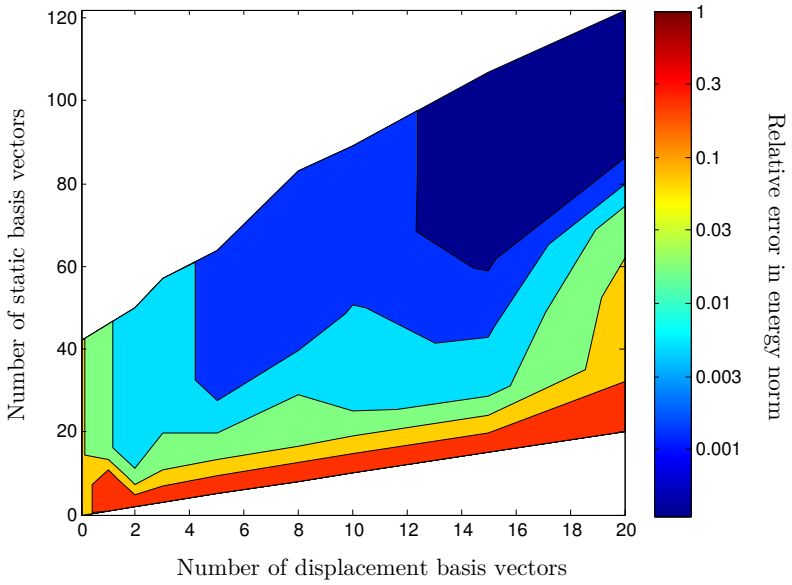


Figure 12: Evolution of the error varying the number of displacement and static basis vectors.

Several remarks can be made:

- As expected, the error decreases when the number of basis vectors increases.
- However, after about 4 vectors in the basis, 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

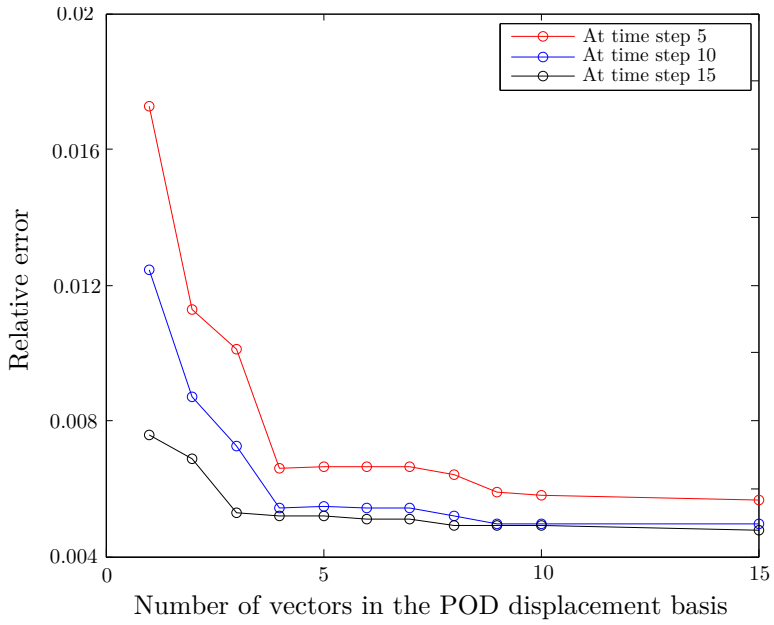


Figure 11: Relative error when using the reduced model without system approximation. As the number of POD basis vectors increases, the error decreases. Using more than 4 basis vectors does not have a strong effect. The error reaches a threshold. That threshold corresponds to the projection error of the exact solution onto the snapshot space.

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 (46), we can decompose the error further (dropping parameters for clarity):

$$\nu_{\text{tot}}(t)^2 = \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{ex,snap}}(t)) + (\mathbf{u}_{\text{ex,snap}}(t) - \mathbf{u}_{\text{r}}(t)) + (\mathbf{u}_{\text{r}}(t) - \mathbf{u}_{\text{r,sa}}(t))\|^2 \tag{48}$$

$$= \|(\mathbf{u}_{\text{ex}}(t) - \mathbf{u}_{\text{ex,snap}}(t))\|^2 + \|(\mathbf{u}_{\text{ex,snap}} - \mathbf{u}_{\text{r}}) + (\mathbf{u}_{\text{r}} - \mathbf{u}_{\text{r,sa}})\|^2, \tag{49}$$

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

The error with respect to the speedup is displayed in Figure 13. 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.

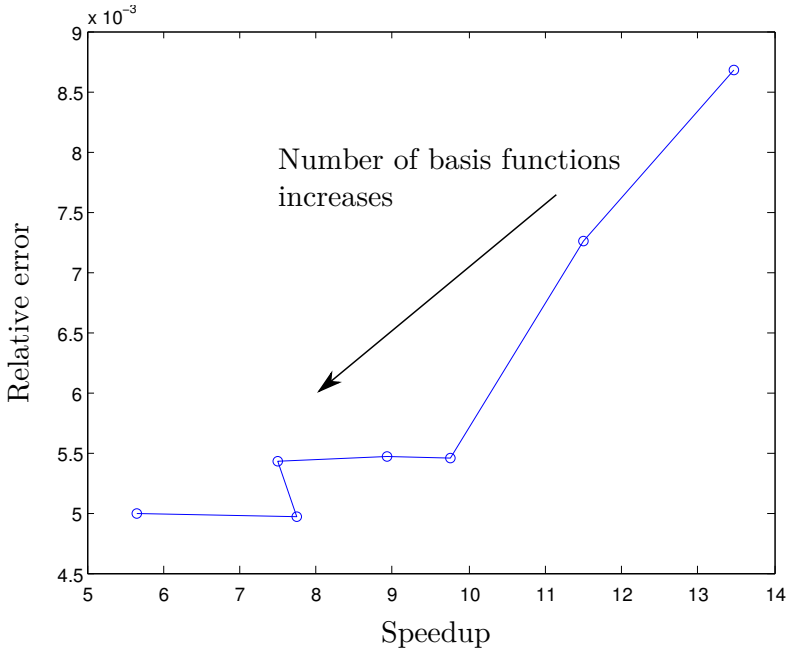


Figure 13: Evolution of the error with respect to the speedup while increasing the number of basis vectors.

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. 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. However that speedup is still significant.

6 Conclusion and perspectives

This paper aimed at reducing the computational burden of solving the representative volume element (RVE) boundary problems arising in semi-concurrent computational homogenisation the context of damage. A POD-based reduced order modelling method was applied, combined with a system approximation technique. One big challenge in trying to apply this method is to select an appropriate snapshot space. The snapshot space was selected using various monotonic loadings paths on the RVE. The example shows a good speedup and a reasonable accuracy. The controlled elements (which are the key elements that monitor information used in the reduced order model) were selected using the discrete empirical interpolation method (DEIM). Those elements gather in the area of high damage and of high variation of displacement, intuitively being the most informative areas. This approach can be though as a bridge between analytical and computational homogenisation: the reduced bases are pseudo-analytical solution of the RVE problem that is still computationally solved at reduced cost. An interesting development to the method would be to have adaptive reduced bases changing depending on the loading applied and the state of the material.

Acknowledgements

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

References

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

[2] 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.

[3] F. Feyel. A multilevel finite element (fe2) to describe the response of highly non-linear structures using generalized continua. *Computer Methods in Applied Mechanics and Engineering*, 192:3233–3244, 2003.

[4] 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.

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

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

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.